ANALYZING C-TEST DATA IN WINSTEPS

Todd H. McKay and Young A Son

1. Welcome! The purpose of this guide is to walk you through setting up and analyzing your C-test data. Along the way, we will discuss aspects of both theory and practice that will (hopefully) help you support ways to interpret your C-test data and justify particular uses of your C-test as an assessment tool.

First, we must give credit where credit is due. This how-to guide was inspired (both conceptually and aesthetically) by the creator of Winsteps and educational measurement expert, Mike Linacre. Mike is not only a Jedi master of measurement but is generally one of the most helpful people we have ever encountered in our academic careers. So, while this guide is meant to help analyze C-test data, for more in-depth coverage of Winsteps, including other data it can be used to analyze, visit the Winsteps website at www.winsteps.com. Homage paid, we now turn to the matter at hand: the C-test guide.

By the end of this guide (provided you read through it carefully and did not doze off [incidentally, we wouldn't blame you]), you will be able to do the following:

- Download and install a version of Winsteps on your computer. (#2)
- Prepare your C-test data in Excel for analysis in Winsteps. (#10)
- Import data to Winsteps by creating a "control file" (more about what this is later). (#14)
- Check assumptions for both rating-scale and partial credit models. (#62)
- Analyzing C-test data-Part 1 (#77)
- What is this 'fit' business? (#83)
- Analyzing C-test data-Part 2 (#109)
- A possibility for placement or screening (#215)

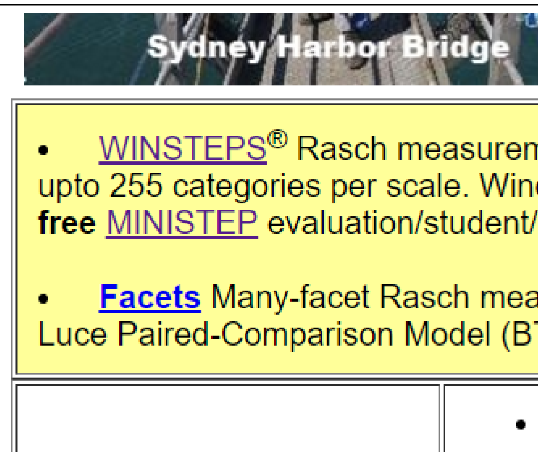
2. Downloading and Installing Ministep

3. To get started, let's begin by downloading the software we will need to analyze C-test data. You may notice that the title of this guide is "Analyzing C-test Data in Winsteps." However, we are going to be analyzing data using the free version of Winsteps, which is Ministep (Linacre, 2016). With Ministep, we can analyze up to 60 cases, or data from 60 test takers. The full version is a great deal at only $149 for life (plus updates, a user manual, etc.). Nowadays, with some software packages costing upwards of $500 for a year, this is a steal.

Copy and paste or enter the following link into your browser:

www.winsteps.com

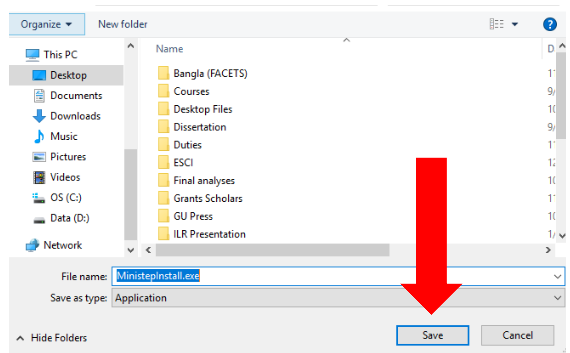
Scroll down to the yellow box (beneath Mike on the bridge) and click on "**MINISTEP**."



<table>
<tr><td>4.</td><td>In the page that opens up, you'll notice that you have two download options, "Ministep: Standard Download" and "Ministep: Secure Download."

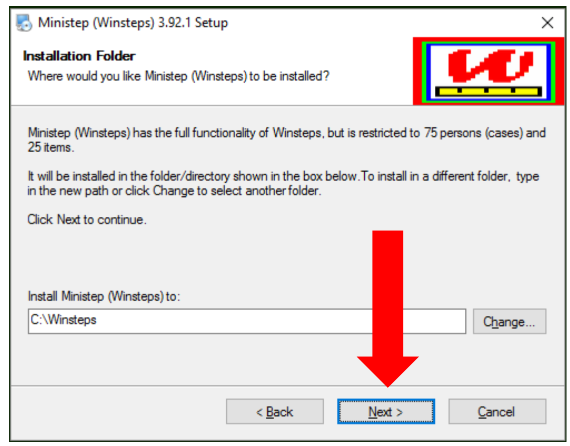
We'll assume you want to be as safe as possible. Click on "Ministep: Secure Download."</td><td>
</td></tr>
<tr><td>5.</td><td>A window will pop up, prompting you to download a file called "MinistepInstall.exe."

Save the file wherever you like. I usually save installation files on my desktop for quick access.

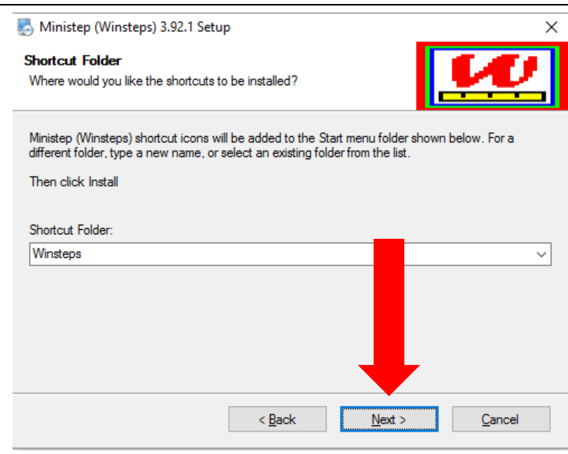
Click on "Save."</td><td>
</td></tr>
<tr><td>6.</td><td>Navigate to where you saved the installation file and double-click on it to begin the installation process.

A Ministep setup window should open in your browser.

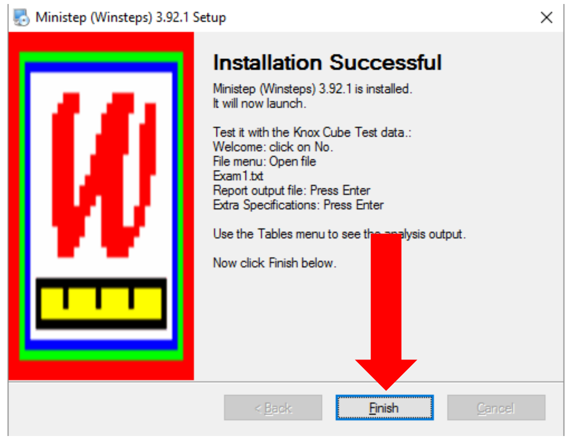
In the "Install Ministep (Winsteps) to:" field, feel free to click "Change" to change the installation path.

Click "Next."</td><td>
</td></tr>
<tr><td>7.</td><td>Another window will appear, asking you where you would like the Ministep shortcut folder installed.

Click "Next."

The installation process begins and should only take a few seconds to complete.</td><td>
</td></tr>
</table>

<table>
<tr><td>8.</td><td>Another window pops up, stating “Installation Successful” (what a relief!).

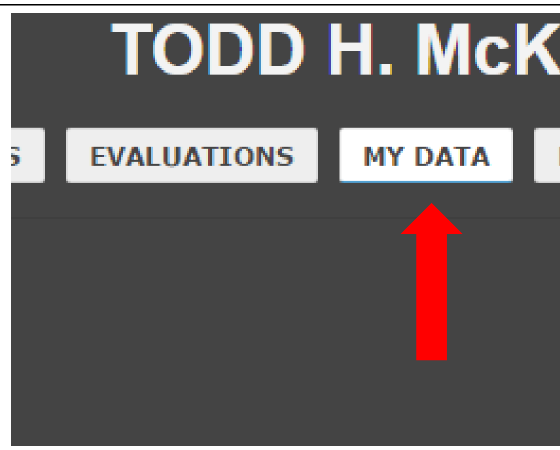
Click on “Finish” to end the installation process.</td><td>
</td></tr>
<tr><td>9.</td><td>Ministep then opens automatically on your computer.

We will come back to Ministep after we have set up our C-test data in Excel for analysis.

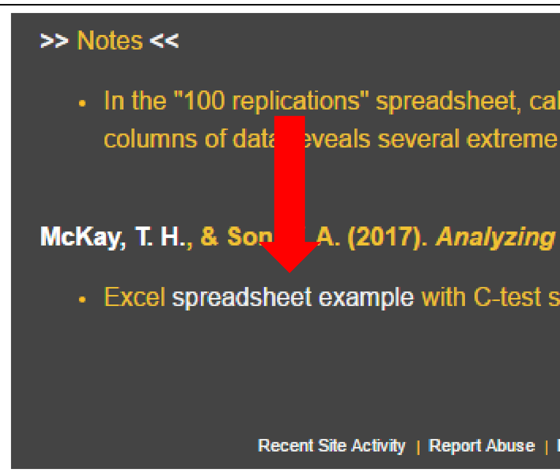
For now, close Ministep.</td><td>
</td></tr>
<tr><td>10.</td><td colspan="2">Preparing your C-test Data in Excel</td></tr>
<tr><td>11.</td><td>Your spreadsheet containing the scores of C-test test takers should have several columns, including one for participant ID numbers, one column for each C-test text, and columns for any other variables of interest.

To give you an example, copy and paste or type the following link into your browser:

https://sites.google.com/site/toddhavilandmckay/

This is a personal website that I use to store and share data. Click on “My Data” tab in the horizontal navigation bar.</td><td>
</td></tr>
<tr><td>12.</td><td>Scroll to the bottom of the “My Data” page. Under the “Analyzing C-test data in Winsteps: A practical how-to guide” section, click on the white “spreadsheet example.”

This will open a spreadsheet containing data from the Bangla C-test.

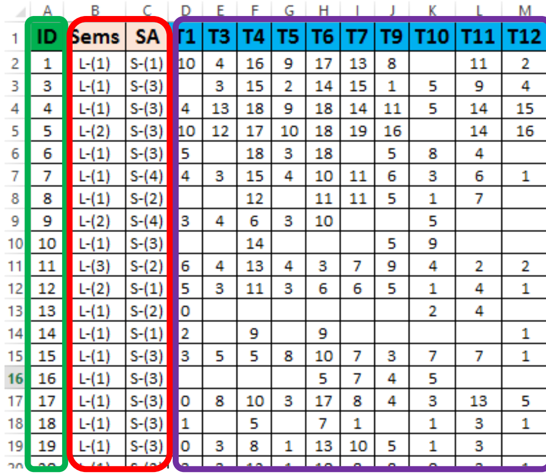
Download and open the spreadsheet.</td><td>
</td></tr>
</table>

13. The spreadsheet you open should like the one in the image to the right.

Green box: The values in this column are the ID numbers of the test takers. For the Bangla C-test, there were 35 test takers. One test taker (#2) was removed before analysis.

Red box: These two columns contain additional variables that we may want to look at later; they were interesting to the developers of the Bangla C-test. The “Sems” column indicates the semester of study of a test taker. For example, if a student was in her fourth semester of Bangla study, the value is “L-(4).” In the Bangla programs from which test takers were recruited, there were many ‘beginning-level’ students but few ‘advanced’ ones. The “SA” column contains test takers’ self-assessment ratings. Test takers were asked to indicate, on a scale of 1–5 (1 = horrendous, 5 = absolutely amazing), their proficiency across the four skills in addition to an “Overall” self-assessment rating. Values in the “SA” column contain the “Overall” rating.

Purple box: These are the total scores for each test taker for each text. Here, there are ten texts. Texts 2 (T2), 8 (T8), and 13 (T13) were eliminated from further development in an earlier piloting phase wherein texts were given to native Bangla speakers—hence ten texts, T1–T12. Each column contains the total scores for test takers out of 25. Recall that each C-test text has 25 blanks. Test taker #17 correctly restored 17/25 blanks on text 6; hence #17’s score on T6 = 17. You’ll also notice that when a test taker did not attempt to complete a text at all (maybe they ran out of time, had an emergency, or simply could not restore blanks), no score was recorded. Rasch analysis is very robust against missing data; *leave missing scores blank unless you have good reason to score them 0.*



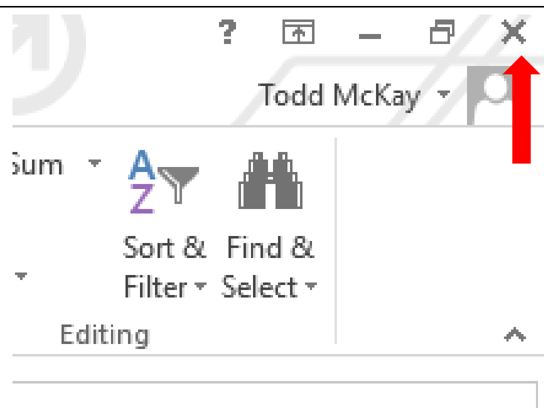
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ID	Sems	SA	T1	T3	T4	T5	T6	T7	T9	T10	T11	T12
2	1	L-(1)	S-(1)	10	4	16	9	17	13	8		11	2
3	3	L-(1)	S-(3)		3	15	2	14	15	1	5	9	4
4	4	L-(1)	S-(3)	4	13	18	9	18	14	11	5	14	15
5	5	L-(2)	S-(3)	10	12	17	10	18	19	16		14	16
6	6	L-(1)	S-(3)	5		18	3	18		5	8	4	
7	7	L-(1)	S-(4)	4	3	15	4	10	11	6	3	6	1
8	8	L-(1)	S-(2)			12		11	11	5	1	7	
9	9	L-(2)	S-(4)	3	4	6	3	10			5		
10	10	L-(1)	S-(3)			14				5	9		
11	11	L-(3)	S-(2)	6	4	13	4	3	7	9	4	2	2
12	12	L-(2)	S-(1)	5	3	11	3	6	6	5	1	4	1
13	13	L-(1)	S-(2)	0							2	4	
14	14	L-(1)	S-(1)	2		9		9					1
15	15	L-(1)	S-(3)	3	5	5	8	10	7	3	7	7	1
16	16	L-(1)	S-(3)					5	7	4	5		
17	17	L-(1)	S-(3)	0	8	10	3	17	8	4	3	13	5
18	18	L-(1)	S-(3)	1		5		7	1		1	3	1
19	19	L-(1)	S-(3)	0	3	8	1	13	10	5	1	3	

14. **Import Data to Winsteps**

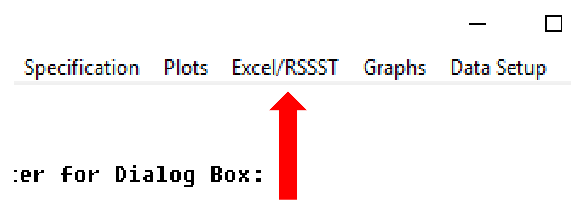
15. Next, we are going to import data from the Excel spreadsheet into Winsteps for analysis.

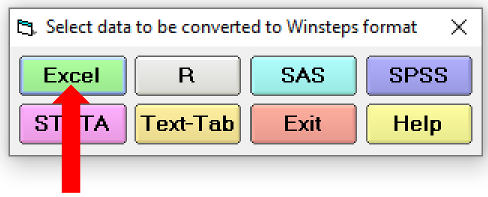
If you would like to continue using your own data, feel free to follow along using your own spreadsheet of C-test data.

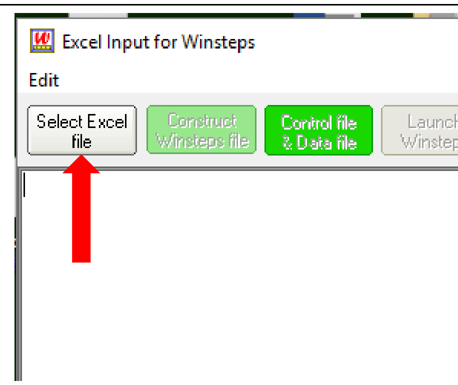
For now, we are done with the Excel spreadsheet. Click “**X**” to close the spreadsheet.



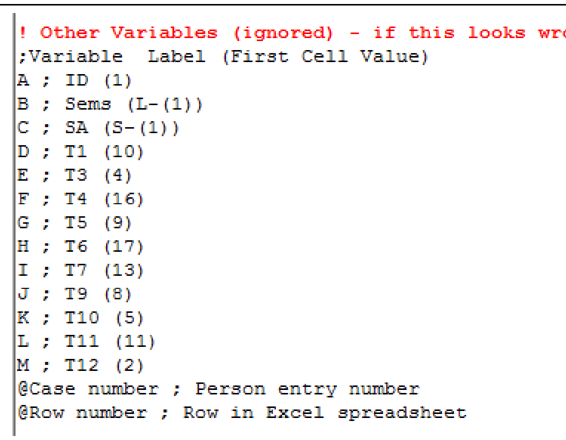
<table>
<tr><td>16.</td><td>Open Ministep.

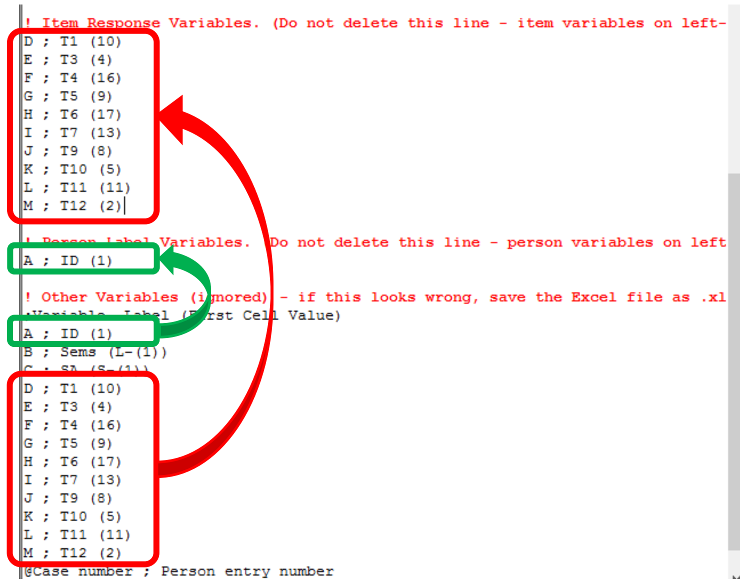
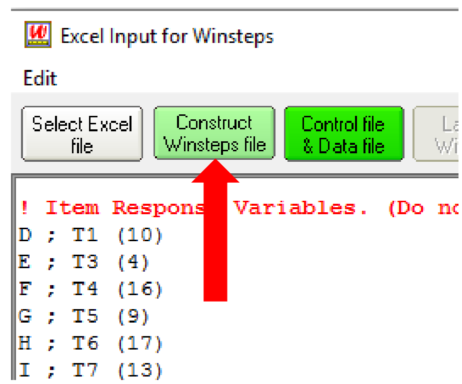
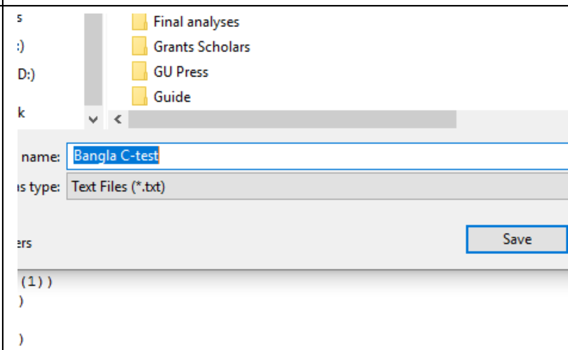
In the Ministep window that opens, click on “Excel /RSSST.”</td><td>
</td></tr>
<tr><td>17.</td><td>A small window pops up with many differently colored buttons, showing us the many different ways that we can import data into Winsteps for analysis.

Click on the green “Excel” button.</td><td>
</td></tr>
<tr><td>18.</td><td>Another window pops up, called “Excel Input for Winsteps.”

Click on the “Select Excel file” in the top left.</td><td>
</td></tr>
<tr><td>19.</td><td>In the window that pops up, navigate to wherever you saved the “C-test guide (10 texts)” spreadsheet or wherever you have your own spreadsheet.

Click “Open.”

Back in the Excel input window, Winsteps populates the window with information from the columns in the spreadsheet. You can see there is an “A; ID (1)” label for the ID column in the spreadsheet, labels for the “Sems” (B; Sems (L-(1)) and “SA” (C; SA (S-(1)) columns, and labels for text columns (T1–T12). The numbers and values in parentheses remind us what the data are in those columns; these values come from the first row of data in the spreadsheet. At this point, everything looks okay.</td><td>
</td></tr>
</table>

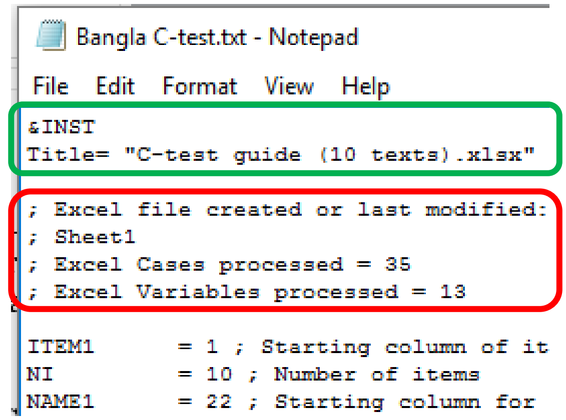
20.	You'll notice three rows of red text in the Excel input window: "! Item Response Variables," "! Person Label Variables," and "! Other Variables." All of our data are beneath the "! Other Variables" line, but we'll move them around so that Winsteps knows which labels correspond to items and which corresponds to the test takers. Green: Copy and paste the ID label (A; ID (1)) from "! Other Variables" beneath the "! Person Label Variables" line. Red: Copy and paste all item labels from "! Other Variables" beneath the "! Item Variables" line.	
21.	Winsteps uses what are called "control files," which are text (.txt) files, to run analyses. Click on the green "**Construct Winsteps file**" button at the top of the Excel input window.	
22.	A window opens up asking us where we would like to save the control file that we are creating from the Excel data. Let's call the control file "Bangla C-test." Type "**Bangla C-test**" into the "File name:" field. Click "**Save**."	

23.

A text file opens in Notepad containing a whole bunch of information. When I opened my first ever control file (like the one we have open here), I was slightly overwhelmed by the information shown and what it all meant. However, it's all fairly straightforward. Let's break it down.

Green box: The "Title =" shows us the title of the Excel spreadsheet from which we created the current control file. This is indeed the title!

Rex box: In the first line is listed the date we created the control file. "Sheet 1" is the sheet from the Excel spreadsheet; we only had one data sheet, and so "Sheet 1" is what appears. "Excel Cases processed =" shows us how many rows of data (test-taker data) there are in the file; for Bangla, there were 35 test takers, and so the "35" is what we expect. The "Excel Variables processed =" tells us how many columns of data were grabbed from the Excel spreadsheet; we had 13 columns (one for IDs, one for Sems, one for SA, and ten text columns). This looks great.



```
Bangla C-test.txt - Notepad
File  Edit  Format  View  Help
&INST
Title= "C-test guide (10 texts).xlsx"
; Excel file created or last modified:
; Sheet1
; Excel Cases processed = 35
; Excel Variables processed = 13

ITEM1      = 1 ; Starting column of it
NI         = 10 ; Number of items
NAME1      = 22 ; Starting column for
```

24.

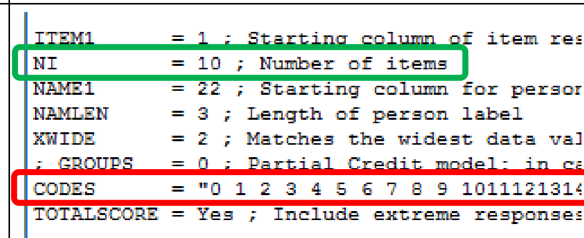
The next set of labels are specifications that tell Winsteps more about the data being analyzed. Note that I added spaces to make the file visually easier to parse.

Green box: The "NI =" means 'number of items' or questions. There are ten texts or super-items in the Bangla C-test, so NI = 10 is correct.

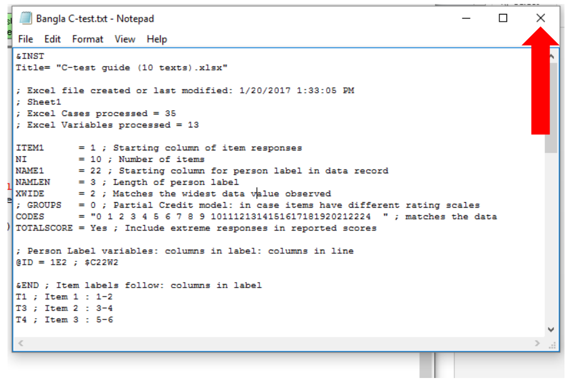
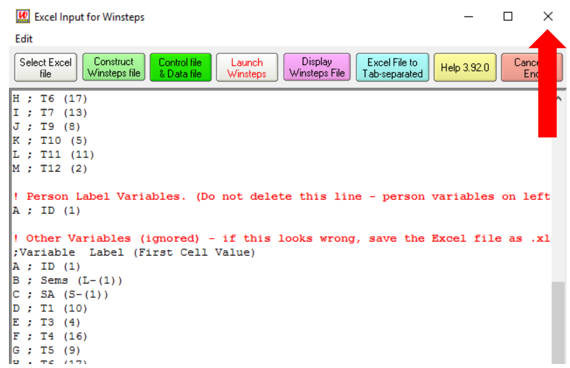
Red box: The "CODES =" refers to the possible values for scores on an item or question. Because each Bangla text is scored out of 25, possible codes or scores are 0–25. However, no test taker scored 25/25 on a text; the highest score was 24/25. Thus, codes are from 0–24. Missing data were left blank in the Excel spreadsheet, so there is a blank space after the last value of 24. We will discuss missing data later.

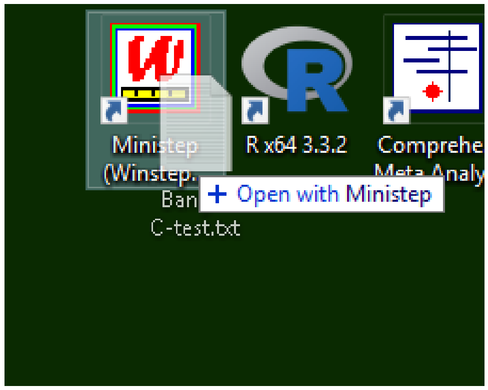
Notice that explanations for each specification (ITEM1=, NI=, NAME1=, etc.) are provided after the semicolon. Linacre (in his usual helpful fashion) adds this information to all control files that are created.

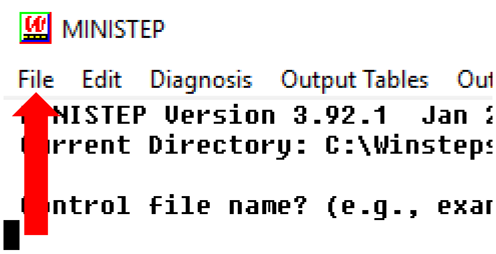
To understand what some of the other specifications are, we will have to look at our data in a different way.

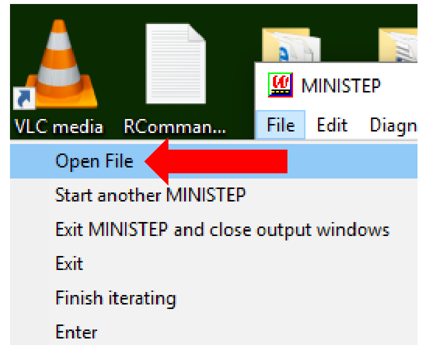


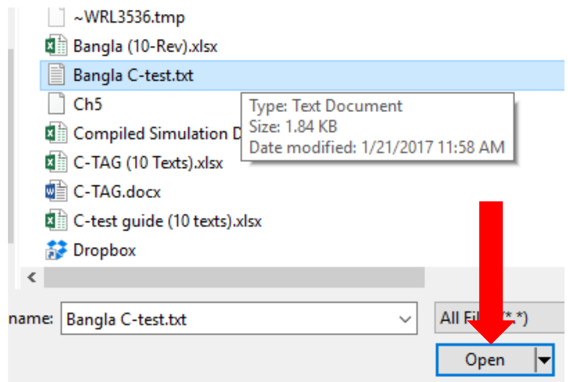
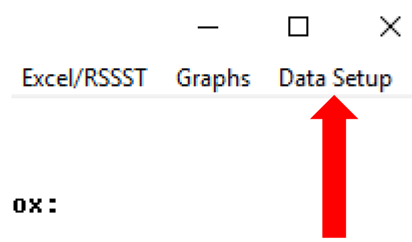
```
ITEM1       = 1 ; Starting column of item re
NI          = 10 ; Number of items
NAME1       = 22 ; Starting column for persor
NAMLEN      = 3 ; Length of person label
XWIDE       = 2 ; Matches the widest data val
; GROUPS    = 0 ; Partial Credit model: in ca
CODES       = "0 1 2 3 4 5 6 7 8 9 1011121314
TOTALSCORE = Yes ; Include extreme response
```

<table>
<tr>
<td>25.</td>
<td>Close the “Bangla C-test.txt” control file.</td>
<td>
</td>
</tr>
<tr>
<td>26.</td>
<td>Close the “Excel Input for Winsteps” window.</td>
<td>
</td>
</tr>
<tr>
<td>27.</td>
<td>Let’s open the data in Ministep as though we are going to begin the analysis.

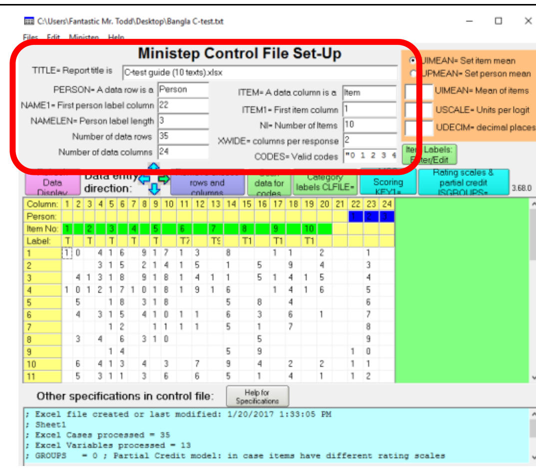
To open a control file in Ministep, drag and drop the control file over the Ministep desktop shortcut.</td>
<td>
</td>
</tr>
<tr>
<td>28.</td>
<td>Alternatively, double click on the Ministep shortcut icon to open Ministep.

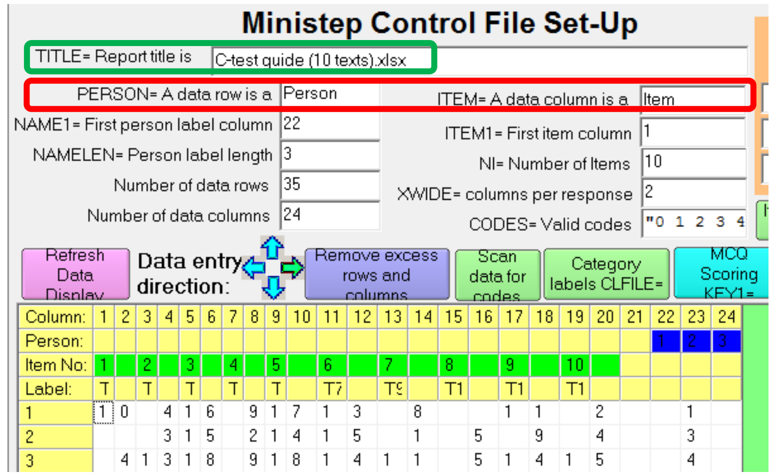
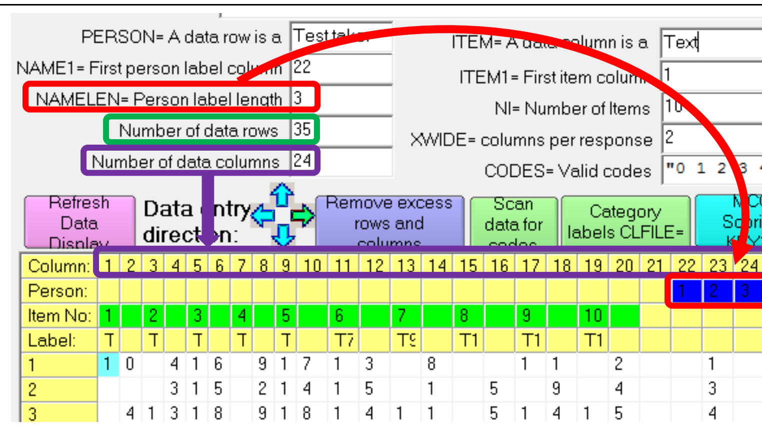
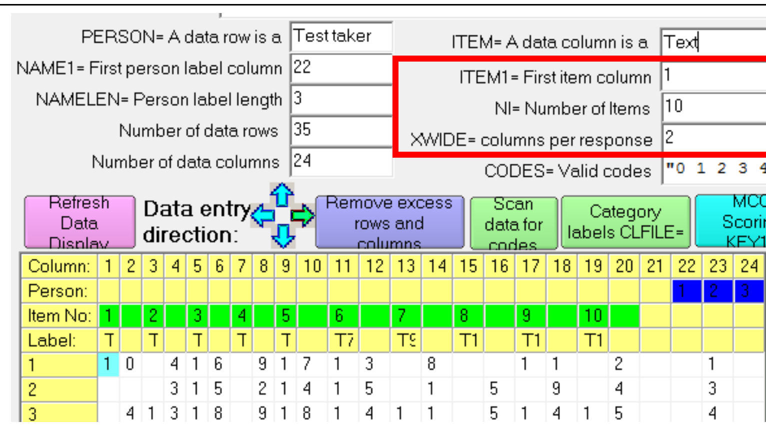
Click “File.”</td>
<td>
</td>
</tr>
</table>

<table>
<tr>
<td>29.</td>
<td>Click “Open File.”</td>
<td>
</td>
</tr>
<tr>
<td>30.</td>
<td>Select the “Bangla C-test.txt” file (wherever you have that saved).

Click “Open.”</td>
<td>
</td>
</tr>
<tr>
<td>31.</td>
<td>In the Ministep window that opens, click “Data Setup” in the top right.</td>
<td>
</td>
</tr>
<tr>
<td>32.</td>
<td>The “Ministep Control File Set-Up” window appears on the screen.

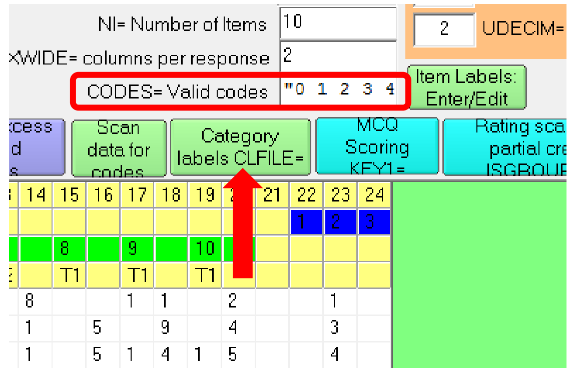
There are several ways to open data in Ministep. You can import data from Excel, SPSS, SAS, R, etc. (like we did with the Excel spreadsheet) to create a control file for the analysis, but we can also manually enter data in this nice interface and then export the data to a control file. Opening data in this way can serve as a nice check to make sure that the information in the control file looks right.

Red box: Look at that! Some of those funky specification lines that we saw in the text file also appear here. With the spreadsheet of data right below these specifications, we are in a better place to examine them and see what they mean. Let’s look at these now.</td>
<td>
</td>
</tr>
</table>

33.	Green box: "TITLE=" This line shows the title of the Excel spreadsheet we pulled data from; this is the same as the information in the text file we looked at and looks okay. Red box: "PERSON=" We can call the people whose scores we are going to analyze whatever we want. We can also change what we call the items ("ITEM="). Let's leave these as "Person and	
34.	Red box: "NAMELEN=" tells us how many cells are needed for the test-taker ID #s. You need one cell per digit. So, if you have 125 participants, you need 3 cells (NAMELEN=3). Here, we have 3 specified even though there are only 35 test takers (2 cells needed). Green box: The 35 in this field means we have a total of 35 cases (test takers) or 35 rows of data. Purple box: The 24 in this field means we have a total of 24 columns in the spreadsheet on the screen. The box outlines columns 1–24.	
35.	Red box: These three specifications tell Ministep more about the items or texts in the spreadsheet. "ITEM1=1" means that the first item starts in column 1. Ministep can then grab the remaining items because it knows items start in the first column. "NI=10" means we have 10 items in total, which we do. "XWIDE=2" means we need two cells for items because scores on texts are anywhere from 1 (1 cell) to 25 (2 cells). We need one cell per digit.	

36. Red box: The “CODES=Valid codes” field contains the possible scores for C-test texts. If you click the green “Scan data for codes” button, Ministep will do a quick scan of the values in the white part of the spreadsheet to see what the scores are.

Click the green “**Category labels CLFILE=**” button.



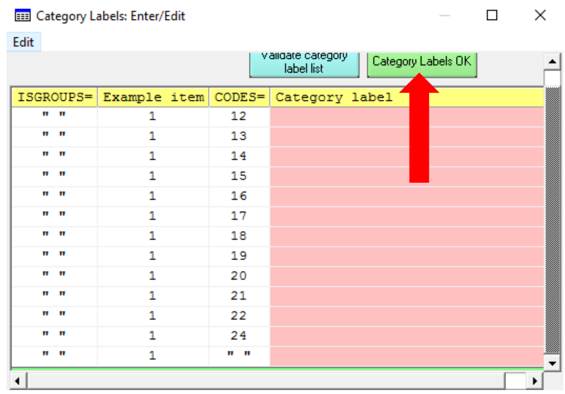
37. A window pops up, which contains more information about the scores or codes in the Bangla C-test.

Do not worry about “ISGROUPS=” for now; this specification tells Ministep whether we want all texts to have the same, 0–25 scale (RSM) or to allow texts to be there own scales (more on this later).

The “CODES=” column contains the range of C-test scores. As we noted earlier, the highest score on any one Bangla text was 24/25, so there is no 25 in this column. The final “ ” is blank because missing scores were left as blank cells in the Excel spreadsheet.

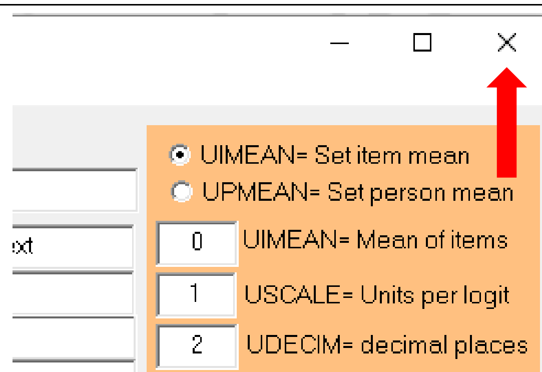
In the “Category label” column, we can add a label to each score or code. This could be useful if you are analyzing scores from Likert items, where a “1” = ‘strongly disagree’ and a “5” = ‘strongly agree,’ for example. In this case, we could type “strongly disagree,” “slightly disagree,” “neither agree nor disagree,” etc. in the red cell that corresponds to the value. We don’t want to assign a label to each C-test blank, so we leave these red cells blank.

Click “**Category Labels OK**.”



38. Now that we have a better sense of what those specifications that we saw in the .txt control file are about, let’s go ahead and close the “Data Setup” window and return to the text file.

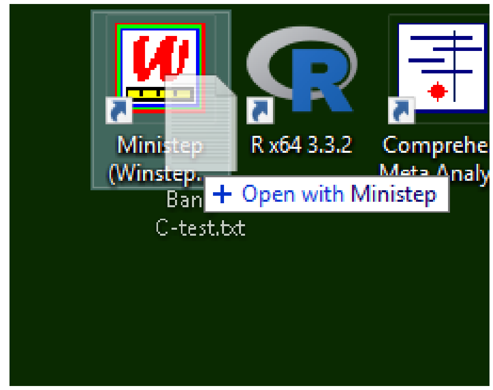
Close the “Ministep Control File Set-Up.”



39. Navigate to the “Bangla C-test.txt” file.

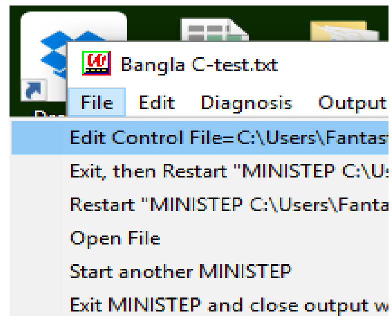
Drag and drop the file on the Ministep desktop icon to open it.

We can double-click the .txt file to open it, but let’s open it another way (so that you know your options).



40. Click “**File**” in the top left.

Click “**Edit Control File=....**”



41. We’re now back in the control file we started with.

Now, however, the different specifications (ITEM=, NI=, NAME1=, etc.) should make more sense to you. Additional commentary is added after the “;” in each line. Recall that Ministep ignores anything after a semicolon.

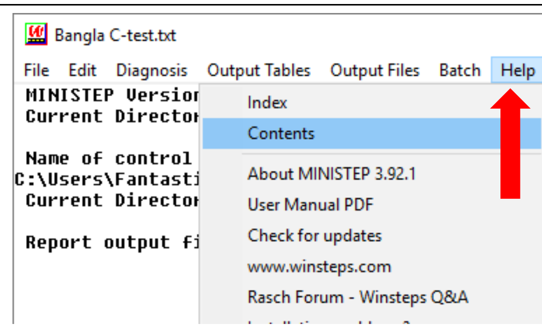
The one specification that we did not see in the nice “Data Setup” page was “TOTALSCORE=."

```
; Excel file created or last modified:
; Sheet1
; Excel Cases processed = 35
; Excel Variables processed = 13

ITEM1      = 1 ; Starting column of ite
NI         = 10 ; Number of items
NAME1      = 22 ; Starting column for p
NAMLEN     = 3 ; Length of person label
XWIDE      = 2 ; Matches the widest dat
; GROUPS   = 0 ; Partial Credit model:
CODES      = "0 1 2 3 4 5 6 7 8 9 10111
TOTALSCORE = Yes ; Include extreme resp
```

42. Let’s look up what “TOTALSCORE=” means.

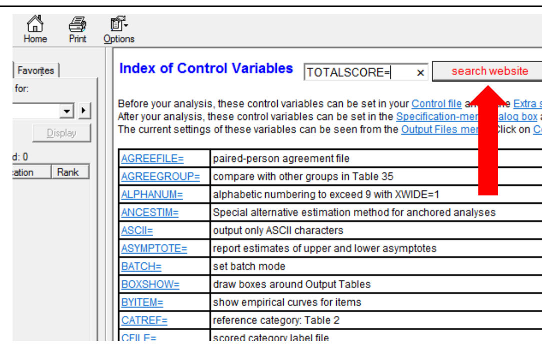
Navigate back to the main Ministep screen and click “**Help**” from the menu at the top.

Click “**Contents**.”



43. A “WINSTEPS Help for Rasch Analysis” window pops up. The main page displays a list of specifications or control variables that we can add to our control file to do different things. From this handy-dandy help menu, we can learn to do just about anything on Ministep/Winsteps.

We search for “TOTALSCORE=” by typing this in the search field at the top, or we can scroll down to it page. Scroll down to “**TOTALSCORE=**” and click on it.

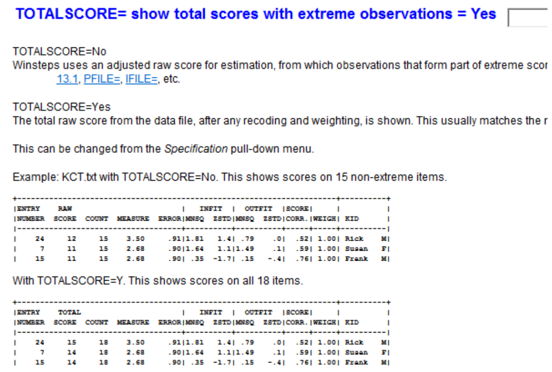


44.

From the screen that pops up, we can read more about this control variable or specification.

With TOTALSCORE=Yes specified in the control file, we get reports on all C-test texts, regardless of whether or not they are 'extreme.' Extreme scores can do funky things to analyses. With TOTALSCORE=Yes, we can get reports on our data with and without potentially pesky extreme test takers and texts.

Close the help menu.



45.

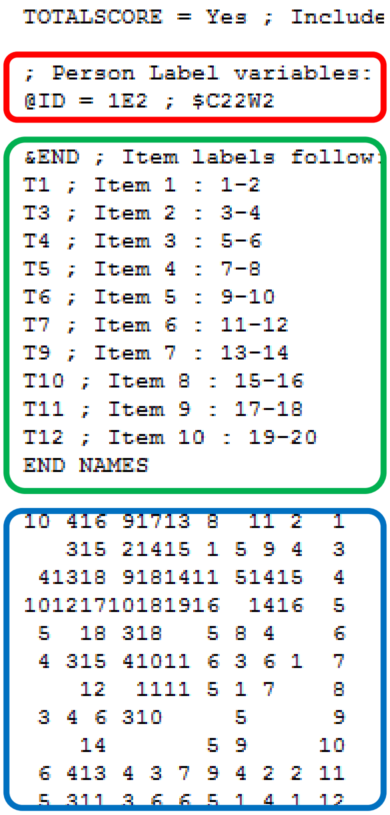
Pull up the control file again.

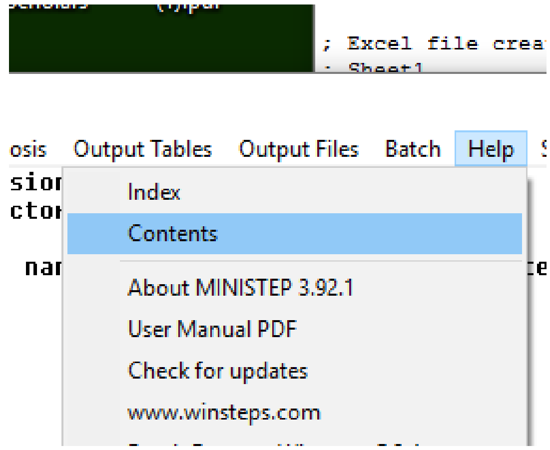
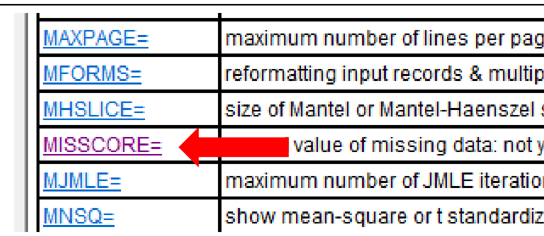
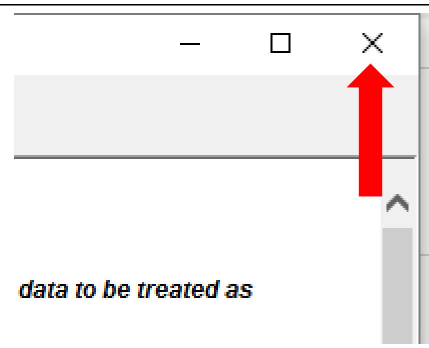
Red box: The "@ID =" line tells Ministep where test takers' IDs are in the data set. They are in column 22 (C22) and are two columns wide (W2).

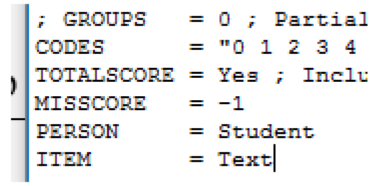
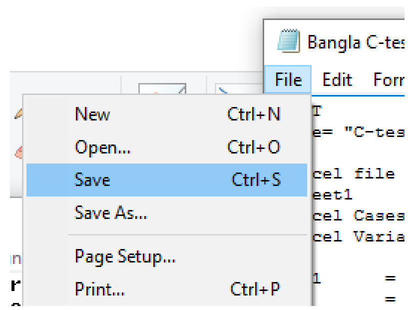
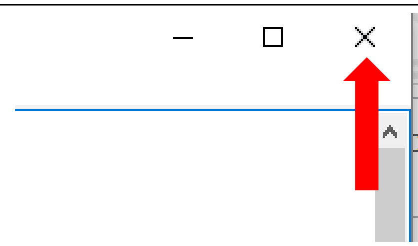
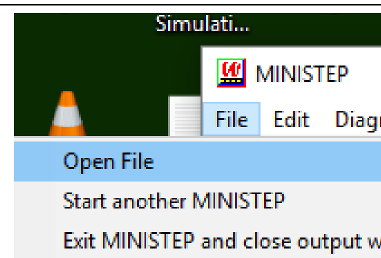
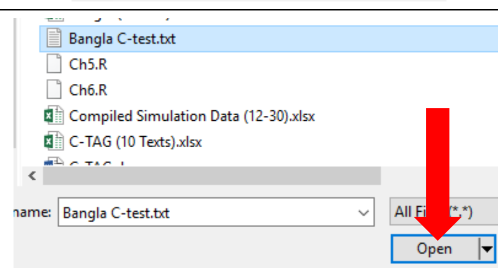
Green box: The information between "&END" and "END NAMES" is about our C-test texts or items. The first text, which we called "T1," is "Item 1" and is in columns "1-2." The last text is "T12," "Item 10," and is in columns "19-20" in the Data Setup page.

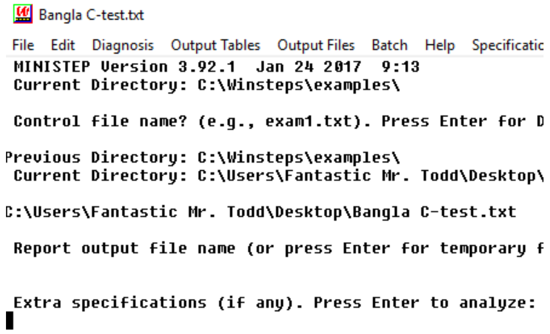
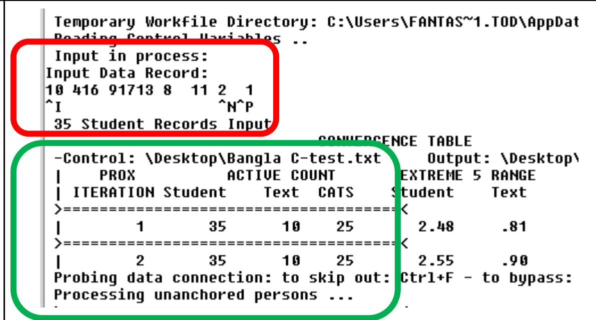
Blue box: These are the actual data for the analysis, followed by the test taker ID numbers. Each row corresponds to one test taker. For example, the first test taker "1" scored a 10 on T1, a 4 on T3, (note that a blank is added before the value if we specified that the data are two columns wide but the value is one digit), a 16 on T4, and so on.

That covers just about all the variables and specifications in the control file! We covered one way (the quick way) of readying the control file for analysis in Ministep, but remember that you can also create a control file by manually entering data in the "Data Setup" interface. Also, once you get better at Ministep and IRT, you can create control files by typing directly into a text document. We think the more IRT-proficient you become, the easier it will be to make changes directly into the text file.

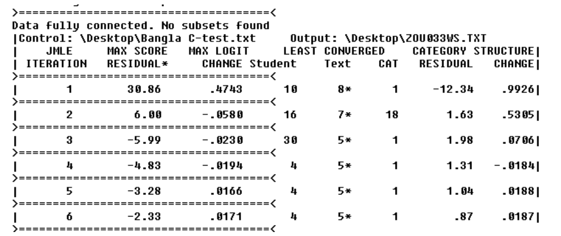


46.	Before closing the control file, two things: First, we will talk more about the “GROUPS=” equals variable later on. For now, remember that “GROUPS=0” means “Hey, Ministep! Run the partial credit model!” This allows texts to create their own rating structure. Again, more on this to come. But, by adding a semicolon before this variable, we are telling Ministep to ignore this variable. Ministep thus ignores the “GROUPS=0” variable and runs the rating-scale model. Second, we want to tell Ministep how to treat missing data. Let’s take a look at our options (options are nice). Return to the Ministep screen. Click on “**Help**” and then “**Contents**.”	
47.	In the “Index of Control Variables” page, scroll down to “MISSCORE=.” Click on the “**MISSCORE=**” link.	
48.	Take a moment to read up on how to treat missing data. (These help screens are pure gold and taught me half of what I know about IRT. Thanks, Mike!) We have a couple options for how we can treat our missing data. We have two options that are viable for the Bangla C-test data. We can either enter “MISSCORE = 0” and tell Ministep to treat missing data as 0’s, or we can enter “MISSCORE = -1” and tell Ministep to ignore missing data and trudge on with the analysis without it. At this point, you might want to think back on how you decided to score responses to C-test texts. Deciding how to score C-test data can be a tricky affair, for we often don’t know if a blank is blank because the test taker didn’t know the word or because they didn’t even try to restore the word (i.e., they ran out of time, they got lazy and skipped around, etc.). In the case of Bangla, a whole text was treated as missing if the test taker did not try to complete any blanks in a text. If a test taker tried to complete a few blanks, but they were all wrong, the text was scored 0. Because scores are summed across texts to create super-items for C-tests, it doesn’t matter how you score individual blanks. If a test taker gets 2/25 blanks	

	correct, the total score is 2/25 regardless of whether or not the other blanks were wrong or missing. Because we scored Bangla C-test texts in the manner above, let's return to the control file to enter in a few more specifications. Close the help window.	
49.	Navigate back to the control file. Add "MISSCORE = -1" to the list of variables to tell Ministep to ignore missing data. Also, to make reading the output easier, enter "PERSON = Student" (to report participants as "Students" in the output) and "ITEM = Text" (to report items as texts).	
50.	Click "**File**" and then "**Save**" to save changes to the control file.	
51.	We are now ready to run the analysis! Close the control file.	
52.	Open Ministep if you closed it. On the main Ministep page, click "**File**" and then "**Open File**."	
53.	Select the "Bangla C-test.txt" file (wherever you have that saved." Click "**Open**."	

54.	"Report output file name." Press "**Enter**." "Extra specifications:" We can add any addition variables here before running the analysis (e.g., ITEM=, MISSCORE=, etc.) There are two places we can add specifications. I find these places to be particularly useful for removing items and people from the analysis once we determine who/what to remove. Press "**Enter**." And, we're off!	
55.	A whole bunch of text appears in the Ministep analysis window, ending in a table of very useful information. Scroll back to the top of the screen so that we can go through the output bit by bit. Red box: This information is key and is the first place we stop to make sure the analysis went okay. The string of text that reads "10 416 91713 8 11 2 1" is an example row of data (from the matrix of data at the end of the control file). The ^I indicates where the text scores begin, and the ^N shows us where the scores end. The ^P shows us where the labels for test takers (ID numbers) begin. Our first person was " 1." There is a blank because the first cell in the two-cell column of test-taker labels is blank. Green box: Ministep performs the analysis multiple times on the data, each time becoming more and more accurate; it does a number of iterations of the analysis until the accuracy reaches a certain level. The values in "PROX ITERATION" mean "analysis round 1," "analysis round 2," and so on. The 35 in the "Student" column means data from 35 students were analyzed, and the 10 in the "Text" column means 10 texts were analyzed. Each text has 25 blanks or categories, hence "CATS" are 25. The "Probing data connection" line is useful when we are analyzing multiple data sets, consisting of multiple files, and we tell Ministep to analyze the lot of them. Here, we are not connecting any datasets, though, so we can skip this line.	

56. Rasch analysis uses a process called joint maximum likelihood estimation (JMLE) to determine scores for students *and* the texts that make up the Bangla C-test. What this means is that Rasch sifts through the data and tries to determine patterns it can use to predict students' scores and text difficulty values. It analyzes the data again and again until it can predict fairly accurately. Therefore, you'll notice that the values in this table become smaller and smaller and smaller. (For those so inclined, Rasch is a special type of regression model, wherein the slope of the regression equation is either constrained to 1 or is an estimate of the population variance, a^2. This is what allows for the 'specific objectivity' in the Rasch philosophy because slope estimates are all the same and only intercept differ.)



```
>=======================================<
Data fully connected. No subsets found
|Control: \Desktop\Bangla C-test.txt     Output: \Desktop\ZOU033WS.TXT
|    JMLE     MAX SCORE   MAX LOGIT     LEAST CONVERGED    CATEGORY STRUCTURE|
| ITERATION   RESIDUAL*     CHANGE Student    Text    CAT   RESIDUAL   CHANGE|
>=======================================<
|      1        30.86       .4743     10       8*      1     -12.34     .9926|
>=======================================<
|      2         6.00      -.0580     16       7*     18       1.63     .5305|
>=======================================<
|      3        -5.99      -.0230     30       5*      1       1.98     .0706|
>=======================================<
|      4        -4.83      -.0194      4       5*      1       1.31    -.0184|
>=======================================<
|      5        -3.28       .0166      4       5*      1       1.04     .0188|
>=======================================<
|      6        -2.33       .0171      4       5*      1        .87     .0187|
>=======================================<
```

57. The table that appears at the end of the analysis provides a summary of the student and text measures. The neat thing about IRT is that it tells you how students compare with one another in terms of language ability (or some other construct) and how difficult texts (or other items) are relative to one another. Student ability and text difficulty are referred to as "measures." This information is contained in Table 0.

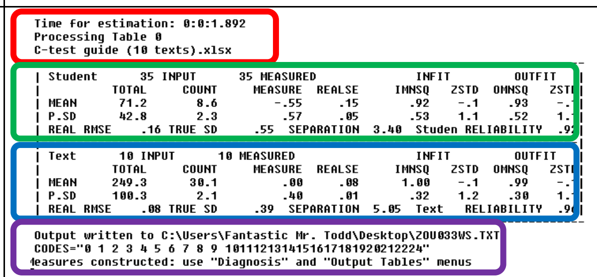
Red box: The analysis took 1.892 seconds on my laptop! With this small data set, the analysis shouldn't take too long. However, we'll see that it takes longer and longer when we start to make modifications to our analysis.

Green box: The top half of Table 0 displays a summary of the analysis for Students.

Blue box: The bottom half of Table 0 shows a summary of the analysis for Texts.

Purple box: This last bit of the analysis shows us the path where Ministep wrote an output text file of the analysis. It also shows us the codes or scores, again, that were in the data.

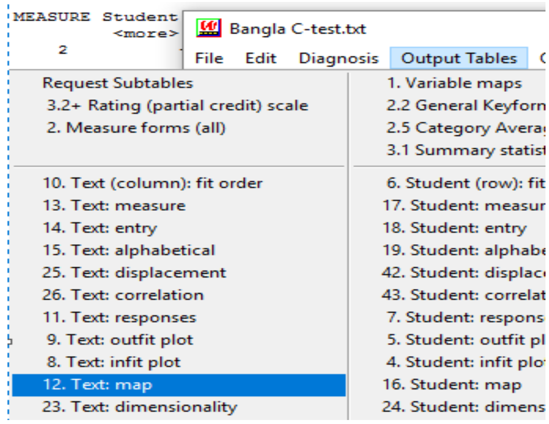
We'll come back to this table later. Let's check out a few nifty graphs.



```
Time for estimation: 0:0:1.892
Processing Table 0
C-test guide (10 texts).xlsx

| Student      35 INPUT       35 MEASURED              INFIT         OUTFIT
|             TOTAL      COUNT      MEASURE  REALSE     IMNSQ   ZSTD  OMNSQ   ZST
| MEAN        71.2        8.6         -.55     .15        .92    -.1    .93    -.
| P.SD        42.8        2.3          .57     .05        .53    1.1    .52    1.
| REAL RMSE    .16 TRUE SD     .55  SEPARATION  3.40  Studen RELIABILITY  .9

| Text        10 INPUT      10 MEASURED              INFIT         OUTFIT
|             TOTAL      COUNT      MEASURE  REALSE     IMNSQ   ZSTD  OMNSQ   ZST
| MEAN       249.3       30.1          .00     .08       1.00    -.1    .99    -.
| P.SD       100.3        2.1          .40     .01        .32    1.2    .30    1.
| REAL RMSE    .08 TRUE SD     .39  SEPARATION  5.05  Text   RELIABILITY  .9

Output written to C:\Users\Fantastic Mr. Todd\Desktop\ZOU033WS.TXT
CODES="0 1 2 3 4 5 6 7 8 9 101112131415161718192021224"
Measures constructed: use "Diagnosis" and "Output Tables" menus
```

<table>
<tr>
<td>58.</td>
<td>In the main Ministep page, click on “Output Tables.”

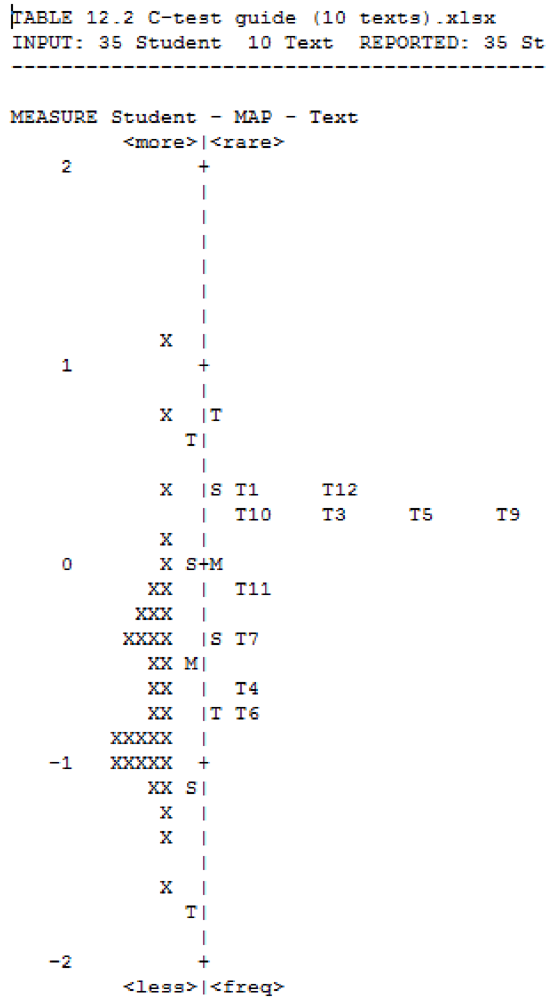
Click on “12. Text: map.”</td>
<td>



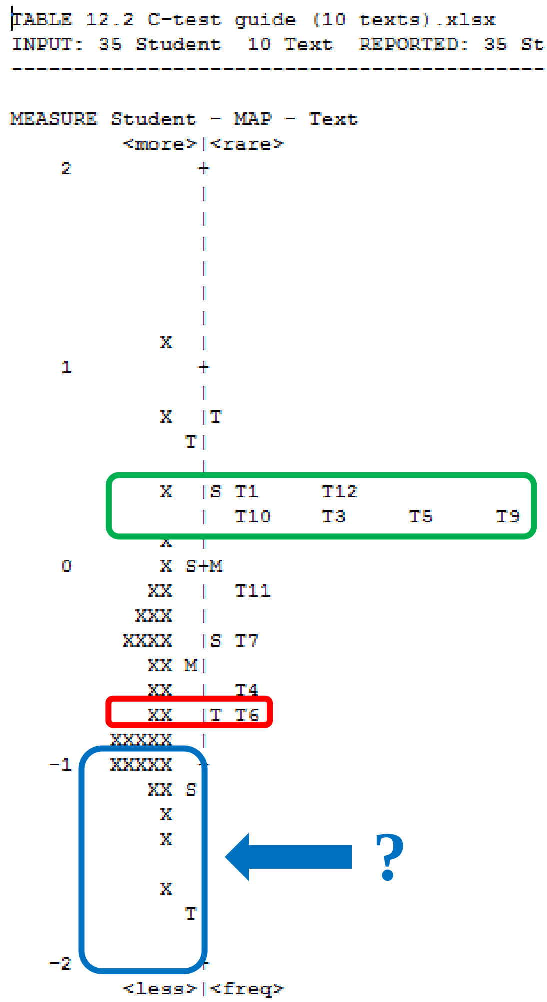
</td>
</tr>
<tr>
<td>59.</td>
<td>This is by far one of the coolest maps you will ever see in your life (tattoo possibilities abound). This is what is called a Wright map (named after Ben Wright, another measurement guru) or variable map. It shows us the distribution of students by their ability levels and the distribution of texts according to how difficult they are!

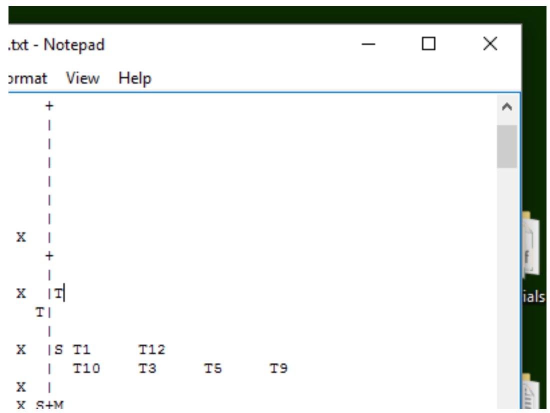
What does this mean? Well, suppose you have two English users for whom English is not their first language: one is a first-semester English learner, and the other is a professor of English literature who translates English literary classics into Hindi/Urdu. Also suppose we have two C-test texts: one taken from a first-grade textbook, and the other is a clipping from the Baltimore Sun (a newspaper). If both scored 24/25 on the textbook text, we would not know who the more ‘able’ English user is; we would think, “They both got 24/25, so they’re equally ‘able’ English users.” However, by adding the newspaper clipping, we can see that the student got a high score on the textbook text and a low score on the newspaper one. We can also see that the professor got a higher score on both. With this information, we can determine that the newspaper text is more difficult than the textbook one, and the student is less ‘able’ than the professor (all other things being equal). We can also see how the texts and our two test takers compare on the same scale!

On the left, in the “MEASURE” column, are the measures for both students and texts (they’re on the same scale). This is just like a ruler. Rasch analysis takes the data and transforms it into what are called “logits.” Logits are neat because the distance between logits -1 and 0 is the same as the distance between 0 and 1. This is a crucial difference between Rasch analysis with IRT and just using percentages. For the above example with our textbook and newspaper texts and two English users,</td>
<td>

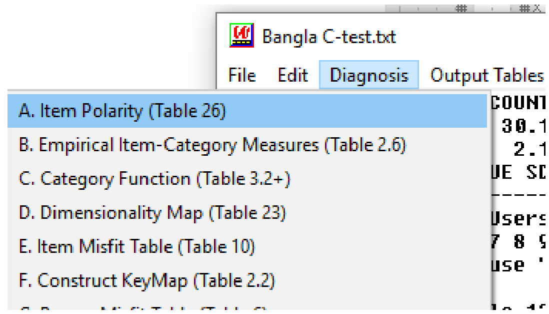


</td>
</tr>
</table>

	if we give a total score on texts using percentages, we cannot say that a score difference of 0 and 5% is the same as the difference from 95-100%. To use a non-language example, imagine we have three math problems on a test: (1) 1+1, (2) 6 x 3, and (3) $\log_2 24 - \log_2 3 = \log_5 x$. Someone might get (1) and (2) (66%), and another person might get all three (100%). But, given the different difficulties of the three math problems can we say that the percent difference between 1 and 2 is the same as 2 and 3? If someone answers (1) and (3) correctly (66%), are they as good at math as someone who gets (1) and (2) correct (also 66%)? Probably not!	
60.	We now know what the values are in the "MEASURE" column; these are the measures (for the students and texts), transformed into logits. Each X represents one student, and the T1, T3, T11, etc. are the texts. The map is oriented such that the more proficient Bangla students are towards the top and the less proficient students are towards the bottom. Also, for texts, more difficult texts are towards the top, and easier ones are near the bottom. Red box: For these two students, T6 is right at their ability level. When students and texts line up like this on the map, there is good 'targeting,' which means we have goodly amounts of statistical information that can be used to calculate the positions of the students and the texts. Thus, good targeting equals less error. Think of targeting like a two-way faucet. If we turn the faucet on and place a cup right beneath it, we get more water in the cup. However, if the faucet and cup aren't lined up (maybe it's early in the morning), some water might miss the cup and end up in the sink. Bad targeting is when you turn the water on and then remember to get a cup from the cupboard. In this analogy, the water is the statistical information. The better the targeting, the more water in the cup that we can use to determine who is more 'able' and which text is more difficult. Blue box: There are no items that are close to these students. We are therefore less sure of their positions with respect to each other, and they will have larger standard errors; this is good motivation to maybe try to incorporate an easier text into the C-test in the future. Green box: We have multiple texts at the same level of difficulty, so, if our goal is to eliminate five texts to create a five-text C-test, we may choose to eliminate a	

	few from this group. However, also note that there is only student at the level of these texts; this means we don't have much statistical information from the student to tell us about these texts. We are trying to use one faucet to fill several cups at the same time. Lastly, note that the group of students on the whole seems to be a bit below the group of texts; this is a sign the texts were relatively difficult on the whole for the group of students. Because many of the Bangla students in this group are low-level learners, this makes sense. Also, students' scores or measures range from about -1.6 to 1.2 logits (more on this in a bit).	
61.	Close the Wright map.	
62.	**Checking Assumptions**	
63.	In the same way that we would check to make sure certain assumptions are met before performing any inferential statistics, we should also check to make sure that certain assumptions are met during our Rasch analysis of the Bangla C-test data. There are three assumptions to remember when analyzing C-test data. First, data should be independent (this should be familiar to you if you have background in statistics). Because a student's response on one C-test blank can influence their responses on other blanks, C-test data are not independent. The same can be said for cloze tests. To get around this problem, we treat each C-test text as a super-item; we add up the scores of individual blanks to get a total score for each text. Therefore, treated as super-items, C-test texts are independent. There is some degree of dependency, but researchers have shown the amount is not too terribly problematic (see Schroeders, Robitzsch, & Schipolowski, 2014). Second, performing a Rasch analysis is like performing a one-factor factor analysis. We want to see that all the items we are analyzing are measuring the same construct (unidimensionality); in the case of the Bangla C-test, we want to assess the degree to texts are measuring Bangla language proficiency. Third, an increase in scores on the C-test should parallel an increasing in Bangla language proficiency. In other words, thinking back on the Wright map, the greater the score, the higher up the Wright map we expect students to be. The lower the score, the lower down on the Wright map the student should be, and so on. Let's have a look at these assumptions for the Bangla C-test that we are analyzing using RSM.	

64. On the main Ministep screen, click "**Diagnosis**" and then click on "**A. Item Polarity (Table 26)**."



65. Red box: This is the reference number for the text in the analysis. If the text labels in the Excel spreadsheet are ordered 1–10, then the entry number and text number will be the same. In our case, the Bangla C-test is missing texts 2 and 8, so the entry number and the text number will be offset a bit. We will refer to these values when removing students or texts from the analysis.

Green box: These are the total scores and counts for each text. Scores are high because each text has a possible score of 25, and there are 10 texts. Total scores are highest for text number 5 and lowest for text number 10.

Blue box: This is the 'score' that the text receives. Remember that Rasch gives us the score for students and texts! The lower the score, or "Measure," the lower down on the Wright map, and the higher the measure, the higher up on the Wright map. Text number 5, our easier text, is lower down with a measure of -.72, and text numbers 1 and 10 are more difficulty (higher up the map), with measures of .39 and .37. These values correspond to texts' positions on the Wright map above.

Purple box: These are the standard errors (SEs) for each text. We want these to be low. Here, all SE values are low, which is good news for our texts.

ENTRY NUMBER	TOTAL SCORE	TOTAL COUNT	MEASURE	MODEL S.E.
7	203	30	.20	.07
1	161	30	.39	.08
3	423	33	-.64	.06
5	438	33	-.72	.06
8	186	32	.26	.07
2	173	27	.28	.08
6	301	29	-.33	.07
10	157	27	.37	.08
4	189	29	.26	.07
9	262	31	-.07	.07
MEAN	249.3	30.1	.00	.07
P.SD	100.3	2.1	.40	.01

66. Red box: We will talk more about Infit and Outfit statistics later when we talk about fit.

Green box: This is the first assumption we want to check. We want to make sure all texts are measuring the same thing. All point-measure correlations (PTMEASUR-AL) should be positive. A negative correlation would mean a text is not measuring what the others are and could mean that an item was coded in the wrong dimension (if you're working with a Likert scale, for example) or that there was a data-entry problem. All these are positive,

INFIT MNSQ	INFIT ZSTD	OUTFIT MNSQ	OUTFIT ZSTD	PTMEASUR-AL CORR.	PTMEASUR-AL EXP.	EXACT MATCH OBS%	EXACT MATCH EXP%	Text
1.46	1.6	1.52	1.8	.74	.84	16.7	16.0	T9
1.48	1.6	1.48	1.6	.75	.84	30.0	18.4	T1
.79	-.8	.79	-.8	.81	.81	6.1	14.6	T4
1.29	1.1	1.20	.8	.83	.81	18.2	15.0	T6
.97	.0	.92	-.2	.84	.83	21.9	17.2	T10
.91	-.2	.85	-.5	.86	.85	33.3	17.0	T3
.66	-1.4	.75	-1.0	.86	.84	20.7	14.1	T7
1.20	.8	1.01	.1	.89	.85	18.5	17.9	T12
.52	-2.1	.58	-1.8	.89	.84	41.4	16.1	T5
.73	-1.1	.74	-1.0	.91	.85	16.1	15.4	T11

Blue boxes: Blue boxes indicate two texts, T1 and T5, for which responses were better than the model expected.

which is great. The observed correlation, “CORR,” should not be too different from the expected correlation, “EXP.” Big gaps between observed and expected values are signs that the data are not fitting the Rasch model properly.

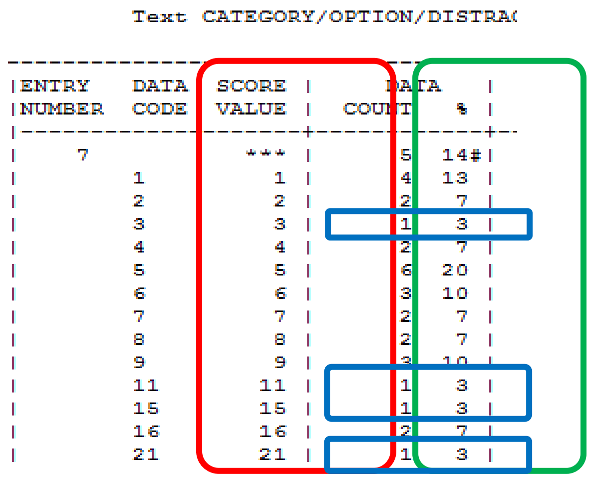
The “EXACT OBS%” is the percentage of scores that are within 0.5 points of the expected scores. The “MATCH EXP%” is the percentage of scores predicted to be within 0.5 points.

67. Scroll down to Table 26.3. This table gives us a look at how the group of students performed on individual blanks for individual C-test texts. This is where we will check the assumption that an increase in scores corresponds to an increase in ability.

The image to the right is for the text whose entry number is 7 but is T9 in the Excel column.

Red box: The “DATA CODE” and “SCORE VALUE” columns go from a blank to 21. The highest score on this text was 21. Note that there are several missing values. No student scored 12, 13, or 14 on this text. The “SCORE VALUE” is the value, meaning someone who got 21/25 was given a score of 21. There were several missing scores, too, which are indicated in the “SCORE VALUE” column by “***.”

Green box: The “DATA COUNT” and “%” columns show the number of students were received a particular score on a text. For instance, 5 students did not attempt this text (“***”), 6 students got a score of 5/25, and only 1 student got a score of 21/25. The “%” is the proportion of students who received that score. Because 5 students did not attempt the text, this proportion is out of 30.

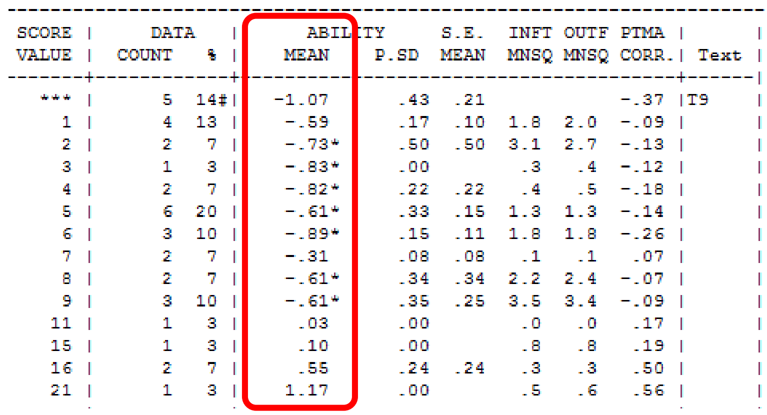


Text CATEGORY/OPTION/DISTRA(

ENTRY NUMBER	DATA CODE	SCORE VALUE	DATA COUNT	DATA %
7		***	5	14#
	1	1	4	13
	2	2	2	7
	3	3	1	3
	4	4	2	7
	5	5	6	20
	6	6	3	10
	7	7	2	7
	8	8	2	7
	9	9	3	10
	11	11	1	3
	15	15	1	3
	16	16	2	7
	21	21	1	3

Blue boxes: This is where things can get a bit hairy when analyzing the data from a small sample of test takers with Rasch. Ideally, we should have 10 responses for each response category when analyzing data with PCM. If our texts have 25 categories (25 blanks), then we should have in the neighborhood of 250 responses for the best possible PCM analysis. Our Bangla C-test sample size (n = 35) is nowhere near this number. For RSM, which we are doing here, we can have less than 10 responses per category, but, for both RSM and PCM, fewer responses mean less information to determine test and student parameters. “Parameters” in Rasch refers to the discrimination ability of texts and students and the positions of texts and students relative to other texts and students.

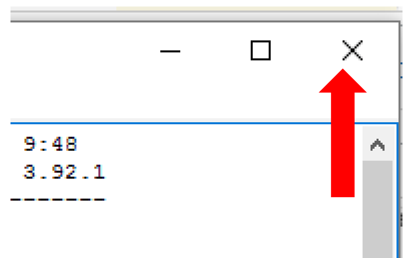
68. Red box: These are the students' scores or the student ability measures ("ABILITY MEAN"). As the scores from the "SCORE VALUE" column go up, we expect students' ability measures to go up. Recall that a negative measure means lower down on the map and a positive value means higher up on the map (i.e., more ability). Students who scored 1/25 have a measure of -.59, and the one who scored 21/25 has a measure of 1.17. So far, so good! However, when a nice progression does not happen, we get the "*'s." These mean, "Wait… A student who got a score of 20/25 has a lower ability measure than a student who got 18/25?"



SCORE VALUE	DATA COUNT	%	ABILITY MEAN	P.SD	S.E. MEAN	INFT MNSQ	OUTF MNSQ	PTMA CORR.	Text
***	5	14#	-1.07	.43	.21			-.37	T9
1	4	13	-.59	.17	.10	1.8	2.0	-.09	
2	2	7	-.73*	.50	.50	3.1	2.7	-.13	
3	1	3	-.83*	.00		.3	.4	-.12	
4	2	7	-.82*	.22	.22	.4	.5	-.18	
5	6	20	-.61*	.33	.15	1.3	1.3	-.14	
6	3	10	-.89*	.15	.11	1.8	1.8	-.26	
7	2	7	-.31	.08	.08	.1	.1	.07	
8	2	7	-.61*	.34	.34	2.2	2.4	-.07	
9	3	10	-.61*	.35	.25	3.5	3.4	-.09	
11	1	3	.03	.00		.0	.0	.17	
15	1	3	.10	.00		.8	.8	.19	
16	2	7	.55	.24	.24	.3	.3	.50	
21	1	3	1.17	.00		.5	.6	.56	

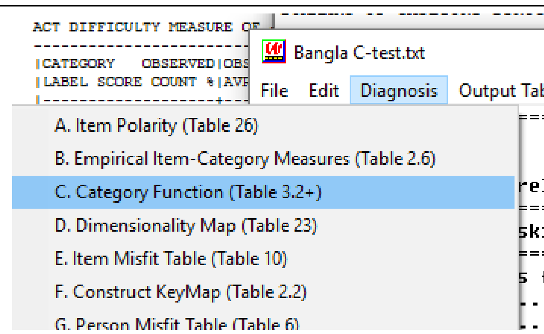
A nice progression is expected by the Rasch model—our second assumption. The model's ability to determine how students stack up is a function of how many students we have in the analysis to make this determination. With few students, we get some disordering! Not great news, but we'll see how things work out. All texts have some disordering.

69. Close out of "A. Item Polarity (Table 26)."



70. Instead of looking at whether an increase in scores matched up with an increase in student ability for each individual text, let's look at scores across all texts.

On the main "Ministep" page, click on "**Diagnosis**" and then click on "**C. Category Function (Table 3.2+)**."



71. In table 3.2, we can look at scores on the whole and how they compare to the range of student ability.

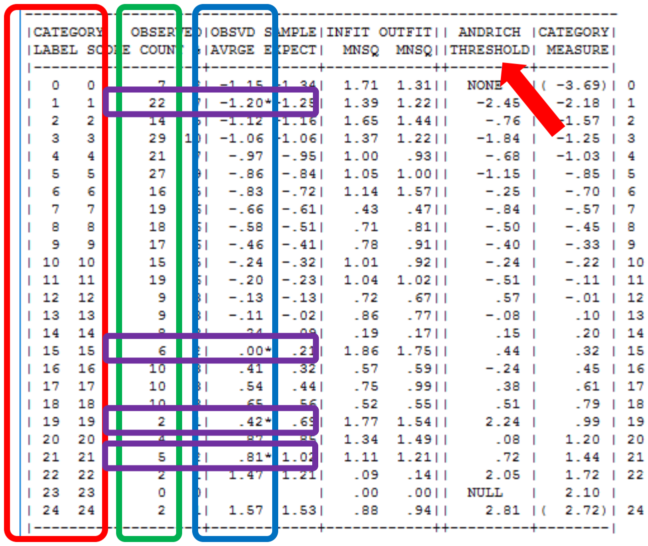
Red box: Notice again how scores go from 0–24.

Green box: We have a lot more students scoring in the 1–11/25 range on the Bangla C-test overall, which is not surprising given the number of low-level learners.

Blue box: Note that the ability measures range from -1.15 to 1.57. There are four disordered categories (*'s) across our 0-24 scale.

Purple boxes: These disordered categories have fewer responses. Recall that about 10 per category is ideal for getting the student ordering right. For "CATEGORY LABEL 1," despite 22 responses for this category, the * is likely due to those 7 students who scored 0/25; we probably assigned scores of 0/25 on several texts for students who simply ran out of time on a given text (or for whatever other reason), resulting in the disordering we see here.

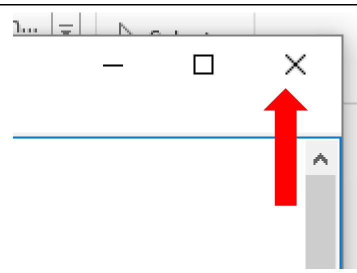
The overall parallel between scores and students' ability measures looks good!



CATEGORY LABEL	SCORE	OBSERVED COUNT	%	OBSVD AVRGE	SAMPLE EXPECT	INFIT MNSQ	OUTFIT MNSQ	ANDRICH THRESHOLD	CATEGORY MEASURE	
0	0	7	[illegible]	-1.15	-1.34	1.71	1.31	NONE	(-3.69)	0
1	1	22	[illegible]	-1.20*	-1.25	1.39	1.22	-2.45	-2.18	1
2	2	14	[illegible]	-1.12	-1.16	1.65	1.44	-.76	-1.57	2
3	3	29	10	-1.06	-1.06	1.37	1.22	-1.84	-1.25	3
4	4	21	[illegible]	-.97	-.95	1.00	.93	-.68	-1.03	4
5	5	27	[illegible]	-.86	-.84	1.05	1.00	-1.15	-.85	5
6	6	16	[illegible]	-.83	-.72	1.14	1.57	-.25	-.70	6
7	7	19	[illegible]	-.66	-.61	.43	.47	-.84	-.57	7
8	8	18	[illegible]	-.58	-.51	.71	.81	-.50	-.45	8
9	9	17	[illegible]	-.46	-.41	.78	.91	-.40	-.33	9
10	10	15	[illegible]	-.24	-.32	1.01	.92	-.24	-.22	10
11	11	19	[illegible]	-.20	-.23	1.04	1.02	-.51	-.11	11
12	12	9	[illegible]	-.13	-.13	.72	.67	.57	-.01	12
13	13	9	[illegible]	-.11	-.02	.86	.77	-.08	.10	13
14	14	[illegible]	[illegible]	[illegible]	[illegible]	.19	.17	.15	.20	14
15	15	6	[illegible]	.00*	.21	1.86	1.75	.44	.32	15
16	16	10	[illegible]	.41	.32	.57	.59	-.24	.45	16
17	17	10	[illegible]	.54	.44	.75	.99	.38	.61	17
18	18	[illegible]	[illegible]	[illegible]	[illegible]	.52	.55	.51	.79	18
19	19	2	[illegible]	.42*	.69	1.77	1.54	2.24	.99	19
20	20	[illegible]	[illegible]	[illegible]	[illegible]	1.34	1.49	.08	1.20	20
21	21	5	[illegible]	.81*	1.02	1.11	1.21	.72	1.44	21
22	22	2	[illegible]	1.47	1.21	.09	.14	2.05	1.72	22
23	23	0	[illegible]			.00	.00	NULL	2.10	
24	24	2	[illegible]	1.57	1.53	.88	.94	2.81	(2.72)	24

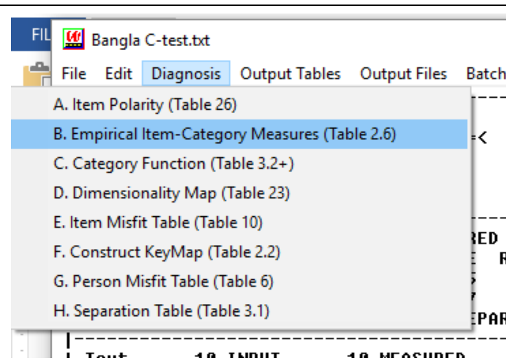
The "ANDRICH THRESHOLD" is an important concept when analyzing rating-scale data that we conveniently gloss over when analyzing C-test data. The values in this column are the point at which—in terms of student ability—students have a 50% chance of endorsing a higher or lower category. For 5-point Likert items, say, it can be very nice to know how respondents will endorse an item given a certain amount of a construct of interest. Because C-tests have 25 categories, knowing at which point a student is 50% likely to score 11/25 versus 12/25 is less useful, particularly when analyzing data with small samples of students. This is undoubtedly an area of contention when using Rasch to analyze C-test data, and one we readily recognize.

72. Close table 3.2.



73. So far, to address the third assumption, we have been looking at a lot of tables. Let's take a look at one more chart to better understand what we mean by increase in scores corresponds to increase in measures.

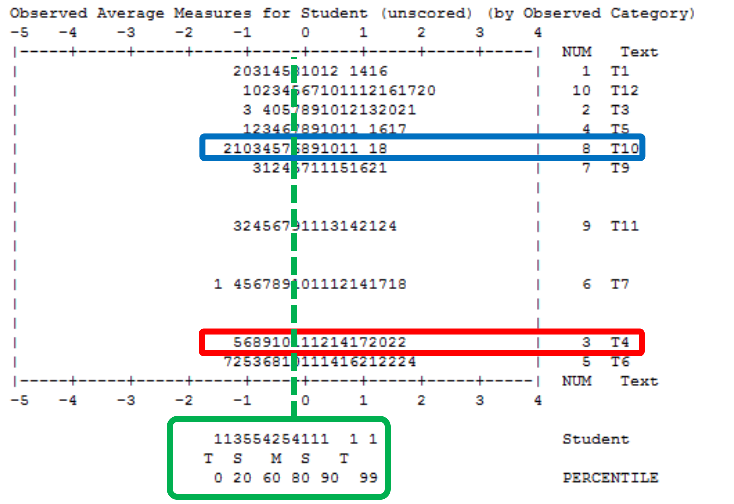
On the main "Ministep" page, click on "**Diagnosis**" at the top and then "**B. Empirical Item-Category Measures (Table 2.6)**."



74. Table 2.6 is similar to the Wright map in some ways. Running left to right, we have the logit scale. Each text has one row of values. Text numbers and their entry numbers ("NUM") are shown on the right-hand side of the map.

Red box: Ideally, we have something like this. Scores are nicely aligned such that lower values are to the left (lower down on the logit scale) and higher scores are to the right. Values here are "5, 6, 8, 9, 10, 11, 12, 14, 17, 20, and 22." In other words, completing more blanks corresponds to a higher score on the logit scale. Fewer blanks corresponds with lower logit scores.

Blue box: Text 10 (T10) has some disordered blanks. The disordering, as we saw above, is probably due to the fact that we had too few test takers who were able to complete the number of blanks indicated by the disordered categories. The fact that they are not *too* disordered, however, is a good sign!



Green box: The top row of numbers represent our Bangla test takers. In the second row, "M" stands for 'mean,' and "S" and "T" indicate 1 and 2 standard deviations from mean "M." If we draw a dashed line at roughly the mean of the text scores (Ministep anchors the overall text difficulty at 0 logits), most test takers appear below this line. Again, this makes sense, knowing that the test-taker sample has a lot of low-level learners.

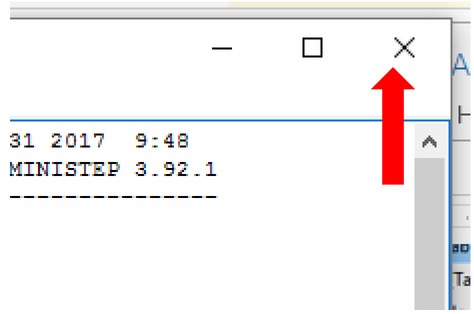
75. We've looked at a lot of tables so far. Let's do a quick recap of for checking assumptions.

(1) Independence of data: This is addressed by treating C-test texts as super-items. Research shows there is still some dependency in C-test texts, but the amount does not mess with the analysis too much.

(2) Dimensionality: Check to make sure all texts have positive point-measure correlations and keep an eye out for (a) negative correlations and (b) big gaps between observed and expected values. By eliminating some misfitting students and texts (this is coming up shortly), values in (b) may improve. If you have negative correlations, check how the item was scored, if there was a data entry error, or consider eliminating. We will discuss how to remove the item from the analysis in Ministep shortly.

(3) Increase in scores = increase in measures: Look at the category frequencies in tables 26.3 and table 3.2 as well as the observed average measures in table 2.6. You should see a parallel between increasing scores and measures. Depending on the size of your dataset, you will likely have more or less disordered categories (marked by *'s).

76. Close table 2.6.



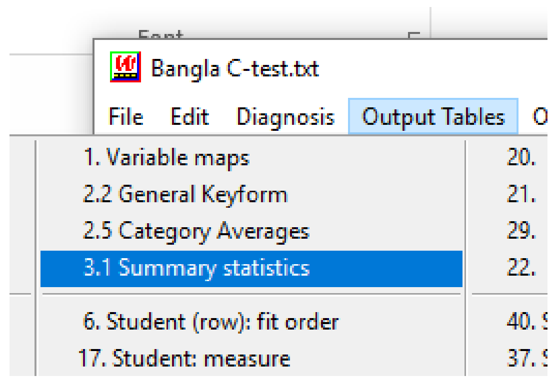
77. **Analyzing C-test Data-Part 1**

78. Now that we've looked at the assumptions for analyzing data with the rating-scale model (RSM), let's start to analyze some data. You may have noted that, at the beginning of this practical guide, I said we were going to analyze our C-test data with both RSM and the partial credit model (PCM), and we will! As we go through the analysis, we will hop back and forth between RSM and PCM analyses so that we can see—in action—what's different between the two models.

79. Let's start by looking at the overall model fit of the Bangla C-test with ten texts before removing five texts to optimize the test.

In the main Ministep window, click "**Output Tables**" and then click on "**3.1 Summary statistics**."

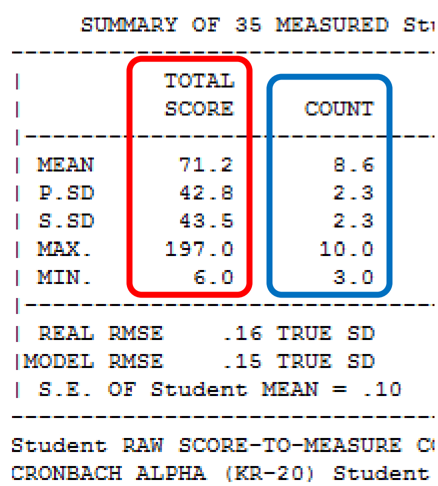
Another text file opens up.



80. In the text file that opens up, you'll see two tables. The first table is table for the student (or person) information, and the second table is for the text (or item) information. This information is similar to that reported in table 0 on the main analysis page. Let's start by looking at the first table with student information.

Red box: The "Total Score" is given in the first column. The mean number of correctly restored blanks on the Bangla C-test was 71.2. The highest score was 197.0, and the lowest was 6.0.

Blue box: The values in the "Count" column indicate the number of responses made. Because our items are whole texts, these values indicate the number of texts. The mean number of attempted texts was 8.6, the most was 10.0, and the lowest was 3.0.



81.

Moving to the right, we have two more columns, a "Measure" column and a "Model S.E." column.

Red box: The average student ability measure is -.55 on the logit scale. The high is 1.17, and the low is -1.56. If you recall, the overall set of texts appeared to be slightly above the group of test takers. The -.55 value confirms our eyeballing earlier. Because texts are anchored at 0 on the logit scale, the fact that the mean student measure is -.55 indicates that, on the whole, our test taker sample is slightly below the mean of the texts.

Blue box: These values are the standard errors (SEs) associated with each measurement. For the most part, these are on the low end, which is great. The higher .37 for the high-ability student is likely because we have few texts at the ability level of this student. Think back on the Wright map above. If students and texts line up on the Wright map, we have more statistical information with which to estimate that student's or text's position relative to other students and texts on the map. We are less sure about the high-ability student because there is no text at his or her level. A look at the map in #60 shows a lone X at about 1.17—and, sure enough, there is no text at the student's level.

Student

MEASURE	MODEL S.E.
-.55	.14
.57	.05
.58	.05
1.17	.37
-1.56	.11

82.

The last two columns in table 3.1 are the model-fit indices. These values indicate how well the overall Rasch model is working giving the data we are analyzing. We have here the "Infit" and "Outfit" statistics, along with their mean square ("MNSQ") and standardized *z*-score values ("ZSTD"). Infit statistics tell us about the extent of the misfit when a text is targeted at a student's ability level (i.e., they are at about the same location on the Wright map). Outfit statistics tell us about the extent of the misfit when a text is not targeted at a student's ability level (i.e., the text is above or below the student's location on the Wright map).

Red boxes: We expect the MNSQ values for both infit and outfit to be about 1.0. Values < 1.0 mean there is some redundancy in the items or dependency in the data (overfit). Values > 1.0 mean there is noise in the data (underfit). MNSQ values tell us how much misfit (over or under) there is in the data.

Blue boxes: These ZSTD values tell us not how much misfit there is but the likelihood of the misfit. Think of these as indicating how worried you need to be about the

INFIT MNSQ	INFIT ZSTD	OUTFIT MNSQ	OUTFIT ZSTD
.92	-.1	.93	-.1
.53	1.1	.52	1.1
.53	1.1	.53	1.1
2.31	2.4	2.41	2.5
.14	-2.6	.19	-2.6

degree of misfit. We want these values to be around 0 (not worried at all), but values > 2.0 mean "Here's a red flag; check it out." ZSTD values are sample-size dependent (like *p*-values), however, so the more test takers we have, the higher the ZSTD values will be. ZSTD values are not too terribly helpful. If we have a large enough group of test takers, ZSTD values will always be > 2.0.

83. **What is this "fit" business?**

Remember that just about everything in statistics boils down to one, simple action: putting a line on a bunch of points. How we get the line is what folks (annoyingly) call the 'model.' If we put a flat line on a set of points, our model is the mean. For correlation and regression analyses, we are plotting a diagonal line on a bunch of points. In IRT, we don't plot a flat line or a diagonal line but an S-shaped line, also called a "sigmoid" or an "ogive." Also in IRT, we don't care about the points individually but the overall response patterns. We'll come back to this in a moment. Let's look at each of these in a bit more detail to understand what we mean by model fit.

84. **The Mean**

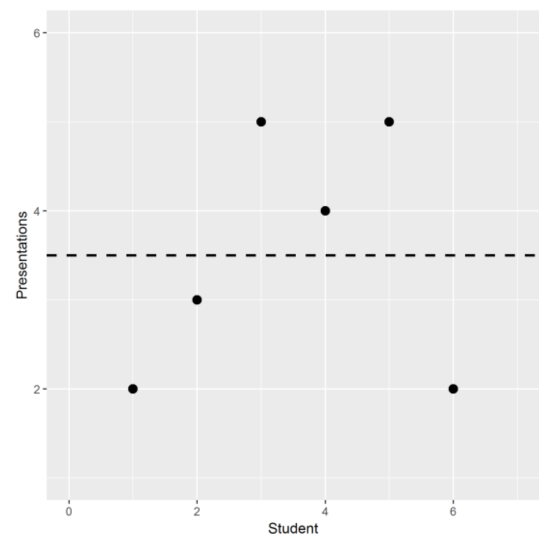
First, the mean is our most basic model. Again, the mean represents a particular line—a flat one—that we put on a bunch of points (our data). The more snug the mean line is on our set of data points, the better the fit.

Imagine that we want to know on average how many presentations graduate students give per semester in the course of their program (I always think we give too many, so I am venting through this example). In the scatterplot to the right, each number on the horizontal *x*-axis corresponds to one graduate student. There are six students represented here. On the vertical *y*-axis are shown the number of presentations each student gives. Student 1 gives 2 presentations per semester, student 2 gives 3, student 3 gives 5 (the nerve), student 4 gives 4, and so on. We can add up the total number of presentations and then divide by the number of students to figure out the average number of presentations given by our group of students each semester (the mean).

Using the formula to the right, we get the following:

$$\bar{x} = \frac{\sum(2+3+5+4+5+2)}{6}$$

$$\bar{x} = \frac{21}{6}$$



The formula for the mean looks like this:

$$\bar{x} = \frac{\sum x}{n}$$

The "$\bar{x}$" stands for 'mean.' The big "Σ" thing just means 'add up.' The "x" represents the values for the points on the scatterplot, and the "n" stands for 'the number of people.' Put together, this means 'add up all the points and divide by the number of people.'

$\bar{x} = 3.5$

The mean (our model) is 3.5! On average, students give 3.5 presentations per semester. The mean is a model because no one actually gives 3.5 presentations. (Well, you might panic halfway through and stop, or you might only finish half your content, I suppose. Your advisor might say, “Todd, that was really only half of what I was expecting.” You get the point.) For the data in our plot, the mean is a good representation of the data; the model fits.

85. Now, as you probably know (or can remember an instance of), people freak out when there are outliers in the dataset, particularly when the dataset is small. We were talking about infit and outfit statistics above. Well, outliers mess with the fit. In other words, outliers make our lines (our models) jump around more than we want them to.

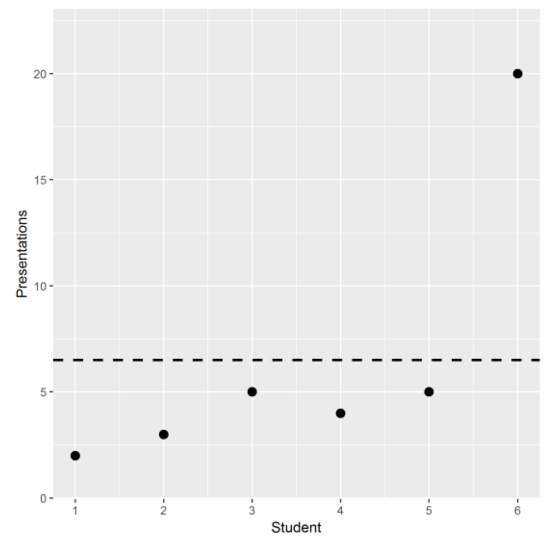
Take another look at the plot to the right. Instead of giving 2 presentations, this data tells us that student 6 gave 20 presentations. The student is either overly ambitious and views PowerPoints as the path to enlightenment, or the undergrad we paid to enter data into Excel messed up somewhere in the data-entry process. We’ll reluctantly assume the latter.

If we revisit our mean equation from above, we get the following:

$$\bar{x} = \frac{\sum(2 + 3 + 5 + 4 + 5 + 20)}{6}$$

$$\bar{x} = \frac{39}{6}$$

$$\bar{x} = 6.5$$



Our mean is now 6.5 instead of 3.5, and we can clearly see in the above plot that the line is well above most of the data points. The line (our model) doesn’t fit the data so well.

Violations like this are what prompt people to use non-parametric statistical analyses, which often use the mode or the median to represent the data. The mode for our dataset above would be 5 and the median would be 4.5 (picture the line moving around); both the median and the mode would represent our data better, too.

86. Great, so we've covered the mean as a model and talked a little bit about how outliers mess things up. The next couple of concepts to talk about are deviation, standard deviation, and variance. I know this isn't a stats course, but knowing these few concepts do make what I talk about later a bit easier to digest.

The dashed red lines indicate the different between the observed point (the black dots) and the mean (our model). How far each point is away from the mean is the *deviation*. Student 1 only gives 2 presentations per semester, but the average is 3.5. Therefore, the deviation for this student from the mean is (2 – 3.5 = -1.5). Student 3 gives 5 presentations per semester, which is 1.5 more presentations than our model 3.5 (5 – 3.5 = 1.5). We can add up all the deviations to get the *total error* for our mean model, like so:

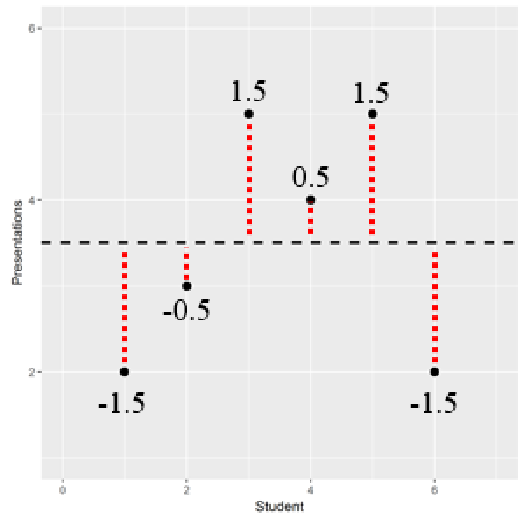
$$\sum(x-\bar{x}) = (-1.5)+(-0.5)+(1.5)+(0.5)+(1.5)+(-1.5) = 0$$

So, by adding up all the deviances, we learn that our total error for the model is 0. Now, we know this can't be true, because clearly the points on the plot are spread out around the mean. The problem is that some points are negative and some are positive and therefore add up to 0. To get around this problem, we square all the deviances and then add them up, getting the *sum of squares* (SS) or the *sum of squared errors* (SSE).

Sum of Squares

$$\text{SS} = \sum(x-\bar{x})^2$$

The above equation for SS should be fairly easy to decipher at this point. It means subtract each point from the mean $(x-\bar{x})$, which gives you the deviation, square them $(^2)$, and then add them up (Σ). The SS for your mean model of student presentations is thus 9.5!



$$SS = \sum(x-\bar{x})^2$$

$$= \sum(-1.5)^2 + (-0.5)^2 + (1.5)^2 + (0.5)^2 + (1.5)^2 + (-1.5)^2$$

$$= \sum(2.25)+(0.25)+(2.25)+(0.25)+(2.25)+(2.25)$$

$$= 9.5!$$

87. The problem with SS is that it depends on how many data points you have; the more data points or observations you have, the larger your SS becomes. Therefore, it's not really a statistic that we can use to make comparisons across models or analyses. But, we can take the average of the SS, which is the *variance*. "Variance" means 'On

Using the equation to the left, our model variance is…

$$variance = \frac{\text{SS}}{(\text{N}-1)} = \frac{9.5}{5} = 1.9$$

average, how much do my points vary off the mean line.' The formula for the variance is given below.

$$variance = \frac{SS}{(N-1)} = \frac{\sum(x-\bar{x})^2}{(N-1)}$$

The top half of the equation is the one for SS. The "N-1" at the bottom is for *degrees of freedom*, which is the total number of groups or datapoints minus 1. Explaining degrees of freedom is always awkward for everyone, so just Google it. It basically limits the number of comparisons that you are allowed to make. We have six students, so our degrees of freedom is *df* = (6 – 1) = 5.

We can now say that our average error in our model (our mean line) is 1.9 presentations. However, because we squared the deviations of each point, the 1.9 really means '1.9 presentations squared.' To generally avoid having to explain what that means to anyone, we take the square root of the variance, which gives us the *standard deviation (SD)*.

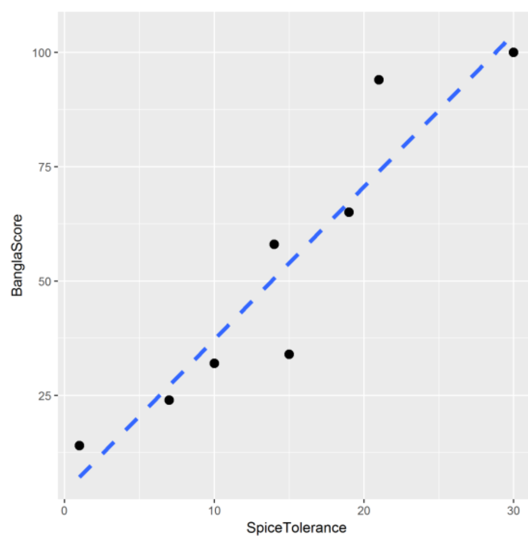
$$SD = \frac{SS}{(N-1)} = \frac{9.5}{5} = 1.38$$

88. Most statistics boils down to putting a line or lines on points. The 'model' is an equation that gives us a line that we can then put on our points. The most basic model is the mean. How well our model fits the data points is what we call "model fit." How far an observed value or data point strays or varies from the mean is called a deviation. We can't add up all the deviations to get a sense of the 'average deviation' because deviations are usually both positive and negative. To get around this issue, we square all the points and then add them up, giving us the sum of squares (SS) or sum of squared errors (SSE). The average of the SS is the variance. The problem with variance is that it is in squared units. To get around this issue, we take the square root of the variance, which gives us the standard deviation (SD). Outliers make the line (our model) jump around, affecting the fit, so we don't like them.

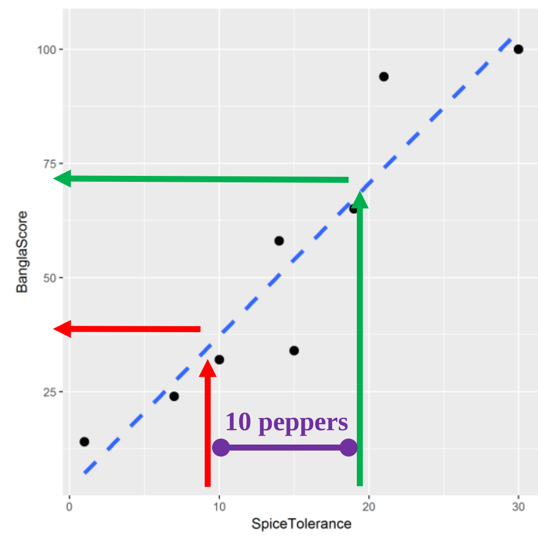
89. **Correlation and Regression**

Let's move on from our mean as the model of interest and look at a regression model. Recall that, in regression, we are not plotting a flat line on the data but a diagonal one. Again, things like "regression" and "logistic regression" still boil down to one thing: putting a line on points (don't let yourself get overly bogged down in the jargon); our line is just a diagonal one. We are talking about mean and regression as models because knowing what's happening with them will make understanding IRT models a bit easier, and Rasch is a type of logistic regression.

90. In the graph to the right, we have yet another model—a regression model (the diagonal line). In regression, we are interested in making predictions. Let's take an example. The more time I spend in Bangladesh, both the better my Bangla becomes (more or less) and the more Bangladeshi food I eat. Bangladeshi food, like most subcontinent food, is very spicy. So, as a researcher and test developer, I think, "Maybe there is a relationship between spice tolerance and scores on a Bangla proficiency test. Just maybe, the more spice-tolerant a Bangla learner is, the better they score on a Bangla test." I therefore *predict* that higher spice tolerance corresponds to higher Bangla scores.



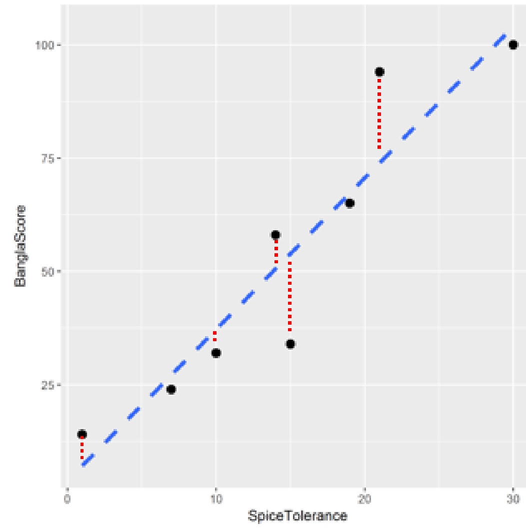
In the graph to the right, my x-axis or predictor variable is "SpiceTolerance" (measured in number of peppers a

	learner is able to consume in a day) and my y-axis or outcome variable is "BanglaScore" (scored out of 100).	
91.	Red arrows: In the graph to the right, the learner with a spice tolerance of 10 has a score of about 38 on the Bangla test. Purple line: Now, what score would a learner get who has a spice tolerance of 20, or who can eat 20 peppers a day? Green arrows: We can use our model to make predictions. If a learner has a higher or lower spice tolerance, what will their corresponding Bangla score be? Here, we can use our model to predict that a learner with a spice tolerance of 20 will obtain a Bangla score of about 72! Useful stuff!	
92.	But, where does this line come from? We can just draw a line across our set of points to get a very crude snapshot of the relationship between spice tolerance and Bangla scores, but, where students are concerned, we want to be as accurate as possible and use the best possible model. So, instead of drawing a line across the points, we use a mathematical formula to get our diagonal line (our model). The formula is shown to the right. The slope of the line ("a") determines the gradient of the line, and the incercept ("b") is the point at which the line crosses the y-axis. Both slope and intercept are what we call "parameters." The slope of the line for the above data is about 3.8, and the line looks like it crosses the y-axis at about 3.5. Therefore, our model/line/equation is… **y = 3.8x + 3.3** You may have noticed that I cheated slightly and said the "slope of the line for the above data is about 3.8." Thankfully, we have statistical software to give us this information. I plugged the data into R, and R told me the slope is 3.8. It also told me the intercept, but I ignored it and used by eyeballs. In summary, we plug the data into software, we let the software tell us what the slope and intercept parameters	$\boldsymbol{y = ax + b}$ y = "BanglaScore" a = slope x = "SpiceTolerance" b = intercept We can use our y = 3.8x + 3.3 model to make predictions. Recall that "x" is spice tolerance. So, plugging in 10 for x (10 peppers), we get… y = 3.8(10) + 3.3 = 38 + 3.3 = 41.3 Someone who has a spice tolerance of 10 will get about 41.3 on the Bangla test! Looking at the graph with our line on it, that number looks about right.

are for our line, and we use the line/equation/model to make predictions. That's regression, folks.

93. Next, we used the spice tolerance and Bangla score data to generate a model to make some predictions. But, how do we know that this model is a 'good' one? The answer is: We compare it to another model. The other model that we have readily available is the mean. So, we use the computer to tell us whether our diagonal line is better than the flat line for making predictions. Whenever someone who uses regression reports statistical significance, they usually mean that their regression model is significantly better than the mean for making predictions.

Before we go into details, the regression model, just like the mean one, has error in it (just about everything we do has error); our data points never exactly fall on the model line. For the mean, how far a point is off the line is called a deviation. We have the same thing in regression, but statisticians decided to call them *residuals*.



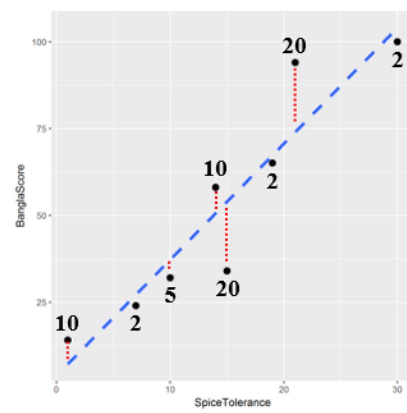
In the above graph, residuals are shown with the dotted red lines.

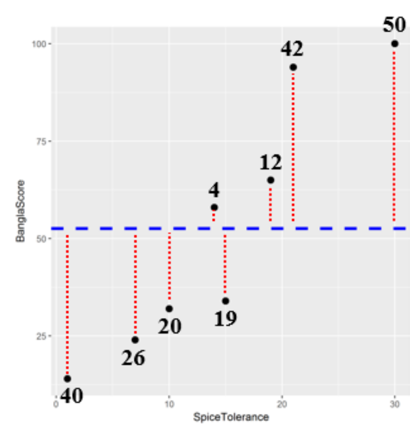
94. Just like for the mean model, we can see that our data points in this example fall both above and below the line. Therefore, if we add them all up, values cancel each other out, and we end up with something close to 0. However, also like the mean model, we get around this by squaring each value and then adding them up. For regression, this is know as the *sum of squared residuals* (SS_R).

Now, we fit a regression line to the model, but the line might not be a very good one, so, as noted above, we can compare it to the mean to see if it's much better than just using the mean.

We take the same set of data points, but this time calculate the SS for the mean (shown in the middle graph). Each red line represents the deviation of the point from the mean. We will call this SS_X where "x" stands for 'mean.'

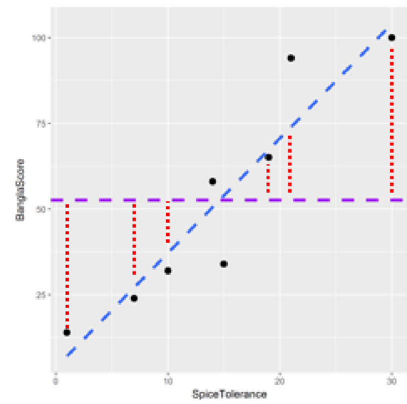
The next step is to calculate the difference between the SS_X and SS_R. If there is a big difference, then the model is better than the mean. If there is little difference between the SS of the models, then our regression line is just about as good as using the mean to make predictions about the data—and so we could just use that. The difference between SS_X and SS_R is the *sum of squares of the model* (SS_M). See the bottom graph.





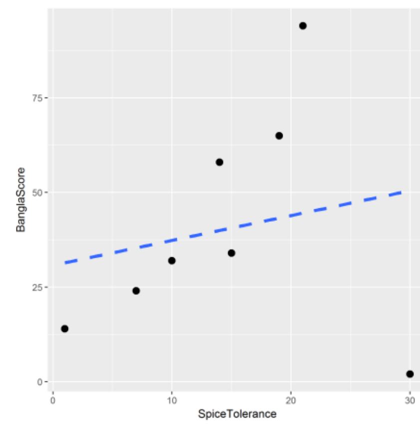
Just how much better the regression model is than the mean for capturing the data can be expressed as R^2. R^2 tells us how much variance is explained by the model, and this value is a percentage. The R^2 formula is thus…

$$R^2 = \frac{SS_M}{SS_X}$$



95. Also like the mean model, outliers mess with the fit and make the line jump around.

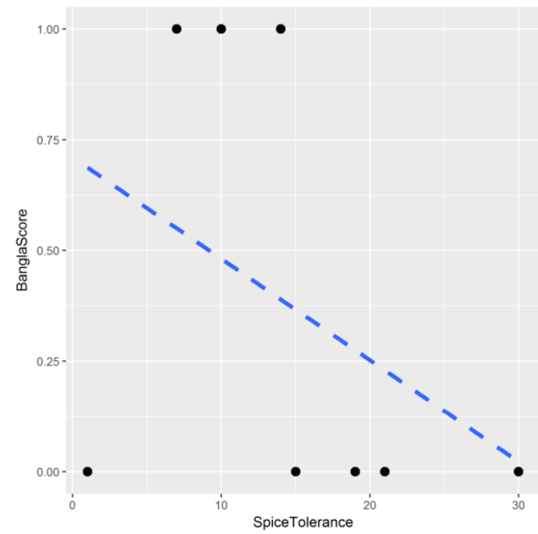
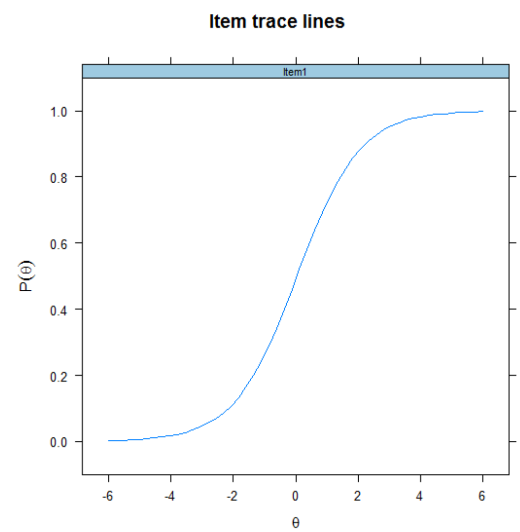
If you look at the graph to the right, we have one outlier at the bottom right. This person has an insane spice tolerance, and yet they got a very low score on the Bangla test. This one outlier pulls the line down, makes the slope flatter, and makes the line fit the data much worse. Now think about the comparison between the mean and regression as models for this data set. Do you think there would be that much of a difference? Would the R^2 be large or small?



96. To quickly summarize this section, when we talk about correlations or regression, we are talking about fitting not a flat line to the data but a diagonal one. We calculate the diagonal line using the *y* = ax + b equation, where *a* is the slope and *b* is the intercept. The computer looks at the data we input and then returns the line that best fits the data—in other words, it fills in the *a* and *b* values for our data. We can then use this line to make predictions. To determine whether or not the regression model is really any good, we compare it to the mean. We calculate SS for both mean (SS_X) and regression (SS_R) models and then look at the difference between the two. The difference can be expressed as SS for the model (SS_M). We divide SS_M by SS_X to get R^2, which tells us how much variance (dots off the line) is 'explained' by the model. Outliers mess with how nicely the line fits the data points.

97. **Rasch Analysis and IRT**

Now let's talk about our last model or analysis for this series of three models—the Rasch model. As noted, the Rasch model is a type of logistic regression. Knowing what we know about regression and the mean as models, making sense of Rasch data should be a lot easier. There are two things that we would like to point out to get things starts. First, the y-axis or outcome variable in Rasch analysis is not a Bangla score but a probability. Second, Rasch analysis/IRT does not use the data from individual points to calculate models—it uses patterns of responses for individuals across all items. So, instead of using a line to predict the likelihood of a probably response (a single data point), we use Rasch to predict the probability that a person would get a series of items correct or incorrect (i.e., a person's response pattern).

98.	The y-axis or outcome variable in Rasch analysis (and IRT, but from now on we just talk about Rasch) is not a score but a probability. Rasch is a form of *logistic* regression. Think about this: For regression, a crucial assumption that you have to meet before you can perform the analysis is linearity; when you correlate your two variables (BanglaScore and SpiceTolerance), the data need to correlate so that you can put a line on the data points at some angle. If we correlate data between spice tolerance (a continuous variable) and a series of right/wrong or 0/1 answers for responses to questions on the Bangla test, we get the graph to the right. The data points are either at the top of the graph (the 1's) or the bottom (the 0's); they do not create a linear set of data points, and so the assumption of linearity is violated.	
99.	We get around the linearity problem in Rasch by transforming the y-axis or outcome data into *probabilities*. In other words, we want to know the likelihood that someone will get a response correct or incorrect, and that likelihood can only range from 0 (no chance in hell) to 1 (a sure-fire guarantee). In the graph to the right, this is what the Rasch model looks like. Notice that the "PΘ" on the y-axis only goes from 0 to 1. The values on the x-axis are the 'ability' of learners, such as Bangla ability or pepper-eating ability. Because the probability can only range from 0 to 1, the curve kind of tapers off close to 0 and 1 and bends inward or outward, making our model line an S-shaped one, which is called an "ogive" or a "sigmoid." If this is one question, the further to the right we move on the x-axis (the more able/proficient/spice-tolerant a learner is), the more likely a learner is to get this item correct (closer to 1). The more we move to the less (the less able/proficient/spice-tolerant a learner is), the chances of a learner getting this item correct decrease substantially. People are never 100% likely to get an item right or wrong, which is why the curve is S-shaped.	 In Rasch analysis and IRT, this type of curve for an item is called an "item characteristic curve" (ICC) or an "item trace line." We can get a trace line for the entire test, too, and this is called a "test characteristic curve" (TCC), which is the sum of all the ICCs in a test.

100. As we noted above, what counts as data for Rasch analysis is not the individual data points but rather the response patterns (even though we need the data, of course, to get the response patterns). What we want is a line not that we can use to predict individual scores but a line that can tell us the likelihood of a string of responses being observed.

Consider the information to the right for a test made up of four dichotomous items. There is one column for each item. For four items, all sixteen possible "Response Patterns" are shown—in other words, there is no other possible combination of right/wrong answers a four-item, dichotomous test. In the "Count" column, these values are for the number of test takers who obtained the corresponding response pattern. For instance, 44 people got every item wrong on the test (0000), while 75 people got every item right (1111).

We feed this information into the Rasch analysis to get the S-shaped curve that describes the response *patterns* of the data in our dataset.

Response Patterns	Count
0000	44
0001	65
0010	25
0011	40
0100	4
0101	6
0110	4
0111	12
1000	7
1001	9
1010	25
1011	29
1100	2
1101	8
1110	38
1111	75

101. Though the actual math is beyond me (can't remember calculus from high-school days), we feed the response-pattern data into the Ministep software to estimate the best possible line. In our linear regression example above, we compare our regression model with the mean to figure out whether or not the regression model is any better than the mean. In Rasch, genius mathematicians and statisticians came up with algorithms to fit the line to the data. After one Rasch, S-curve model is fit to the data, another one is fit, and the first model is compared with the second. This process continues until we obtain the line that best predicts the response patterns in our data.

The actual Rasch model is given below:

$$\frac{1}{1 - \exp\left[-\left(a_j \Theta + \mathrm{d}\right)\right]}$$

In the above model (how we get the line), "a" is the slope and "d" is the intercept; we still use slope and item parameters for Rasch analysis, just like the linear regression model. The "Θ" or 'theta' is the ability, proficiency, or spice tolerance (etc.) of a test taker.

102. Now that we know how Rasch generates a model for the data, let's conclude this section by talking a little bit about fit, which is where we left off in the analysis of the Bangla C-test data.

To the right, we have the responses of 19 test takers to a 10-item test. This is called a "scalogram." Test takers are sorted with high-ability test takers at the top, and low-ability test takers towards the bottom. There is one column for each item. Easier items are towards the left, and the most difficult items are towards the right. Young

Young A	1111111111
Walsh	1011111111
Zoe	1111111101
Mal	1111111110
River	1111111100
Charlie	1111111000
Youssef	1111101010
Soujin	1110101000
Zhizhou	1110110100
Liz	1110010100

A is the best test taker, since we can see that she got all items right (1111111111). Jane (not surprisingly) is one of the worst, getting all items wrong (0000000000). In between Young A and Jane, we have other test takers with a range of ability levels.

Red box: We have a lot of 1's in this corner; the test takers with the highest ability levels are getting all of the easier items correct.

Blue box: We have a lot of 0's in this corner; the test takers with the lowest ability levels are getting the most difficulty items wrong.

This is what we would expect in the response patterns of test takers when we have test takers with a range of ability levels and items that span a range of difficulties.

Sudipta	1101010000
Carly	1110010000
Todd	0000000011
TJ	1010100000
Harrison	1001000000
Sean	1100000000
Nick	0100000000
Simon	1000000000
Jane	0000000000

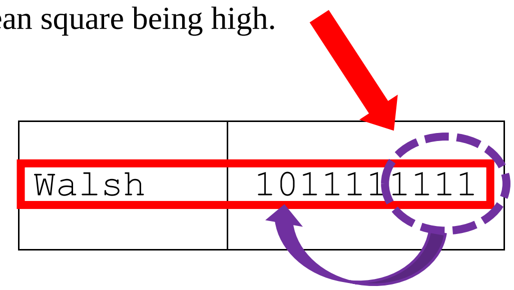
103. The fit of the Rasch model is affected by funky items and funky response patterns. Often, misfitting items are a result of misfitting responses.

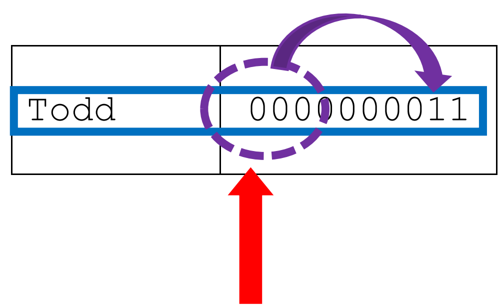
In the response patterns for the 10-item test, there are several unexpected response patterns. Depending on where these patterns are located in the response string, we get high infit or outfit (i.e., misfit) values in the Ministep output (see [82] above).

Red box: Walsh's response pattern is strange. He is a high-ability test taker, so we would expect him to get the easier items correct. However, he scored a 0 on the second item. Given his ability, this is an unexpected response. Walsh maybe got this item incorrect because he dozed off or perhaps there was a data-entry error. Thinking back on the Wright map, this item would be below Walsh, meaning there is a higher probability Walsh would get this item right. Therefore, Walsh would have a high Outfit mean square value.

Blue box: Todd's response pattern is just as weird. Todd is (true to form) a low-ability test taker; accordingly, he got all the easy items and most of the middle-difficulty items wrong. However, for whatever reason, he got the two most difficulty items right. Recalling the Wright map, these items would be well above Todd, and Todd would have a high Outfit mean square value as well. Todd maybe cheated on these items, which is why he got them right, or they were lucky guesses.

Walsh's ability is about here in relation to the test, but we are seeing a strange response pattern *outside* this range—hence the *outfit* mean square being high.





Todd's ability is about here in relation to the test, yet we are seeing strange response patterns well outside this range—again, hence the high outfit mean square value.

104. That's for outfit mean square values, but what about infit mean square values? What does a high infit score mean?

Well, let's take a step back. In an ideal test, we would see a response pattern like the one to the right. This is called a "Guttman" response pattern. A test taker for whom the test is well targeted got the first five items right and the last five items wrong. This means that, right there in the middle between the "1" and "0," is the point at which the difficulty of items begin to exceed the test taker's ability; in other words, it is precisely at this point that the probability of the applicant obtaining correct answers begins to decline. Below this point, however, the items are not too difficult for the test taker, and she gets them right.

```
Guttman    1111100000
```

105. However, response patterns like the one above are rare. There is usually some variability right at the point where items begin to challenge a test taker's ability; maybe they get a few in that range correct and a few incorrect.

Red box: Soujin's response pattern shows variability in the middle of the response string. The test is overall well-targeted for a test taker like Soujin; she gets the first three items correct and the last three items incorrect. In the middle, there is some variability; it is at this point that Soujin's ability begins to be challenged by test items—this variability is exactly what we need for measurement purposes. When response patterns in this middle range are unexpected, we see high infit mean square values.

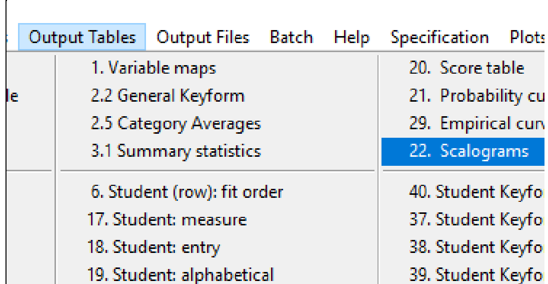
Unexpected responses *outside* this range result in high outfit mean square values. Carly's response pattern would yield a high infit mean square. Unexpected response *patterns* affect the fit of the Rasch model, as opposed to individual, outlying data points.

```
Soujin     1110101000

Carly      1110010000
```

106. So far, we have talked about fit and response patterns for Rasch analysis using dichotomously scored items. However, what about super-items like C-test texts? Well, the basic principle is the same.

To see a scalogram of Bangla C-test scores, return to the main Ministep window.

Click "**Output Tables**" and then click on "**22. Scalograms**."

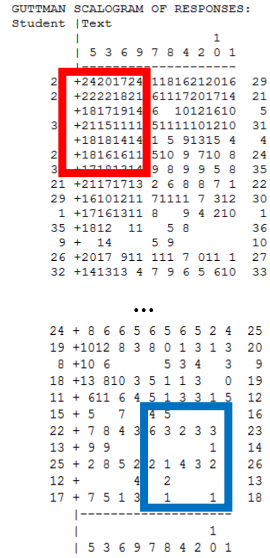


107. In table 22, the scalogram resembles the one for dichotomous items above.

Rex box: We are seeing the more proficient Bangla learners getting easier items correct at the top, left-hand corner of the scalogram.

Blue box: Likewise, we are seeing the lower-ability Bangla learners getting the more difficult items incorrect at the bottom, right-hand portion of the scalogram.

For the Bangla C-test, the Rasch model—our S-shaped curve—is applied such that the probability of obtaining these response patterns is obtained.



```
GUTTMAN SCALOGRAM OF RESPONSES:
Student |Text
        |                   1
        | 5 3 6 9 7 8 4 2 0 1
        |--------------------
     2  +24201724 11816212016  29
     2  +22221821 61117201714  21
        +18171914 6  10121610   5
     3  +21151111 51111101210  31
        +18181414 1 5 91315 4   4
     2  +18161611 510 9 710 8  24
     3  [illegible] 9 8 9 9 5 8  35
    21  +21171713 2 6 8 8 7 1  22
    29  +16101211 71111 7 312  30
     1  +17161311 8   9 4 210   1
    35  +1812  11   5 8        36
     9  +  14     5 9          10
    26  +2017 911 111 7 011 1  27
    32  +141313 4 7 9 6 5 610  33

                ...

    24  + 8 6 6 5 6 5 6 5 2 4  25
    19  +1012 8 3 8 0 1 3 1 3  20
     8  +10 6       5 3 4   3   9
    18  +13 810 3 5 1 1 3   0  19
    11  + 611 6 4 5 1 3 3 1 5  12
    15  + 5   7   4 5          16
    22  + 7 8 4 3 6 3 2 3 3    23
    13  + 9 9             1    14
    25  + 2 8 5 2 2 1 4 3 2    26
    12  +       4   2          13
    17  + 7 5 1 3   1     1    18
        |--------------------
        |                 1
        | 5 3 6 9 7 8 4 2 0 1
```

108. To briefly summarize this section, the Rasch model is a type of logistic regression. To overcome the linearity problem, y-axis or outcome data is transformed into a probability. Because a probability can only range from 0 to 1, the Rasch model is not a diagonal line but an S-shaped curve, called a "sigmoid" or "ogive." For items, the curve is called an "item characteristic curve." For the whole test, the curve is called a "test characteristic curve." Data for Rasch analysis is not individual data points but response patterns. The model is fit to the data so that the likelihood of obtaining the observed response patterns is maximized. Unexpected response patterns affect the fit of the Rasch model. Unexpected response patterns within the productive measurement range (the point at which we start obtaining 0's and 1's for a test taker) result in high infit mean square values. Unexpected responses outside the productive measurement range result in high outfit mean square values. Variability in responses—that set of 0's and 1's—is needed for statistical measurement; without variability, our test isn't doing any 'measuring.'

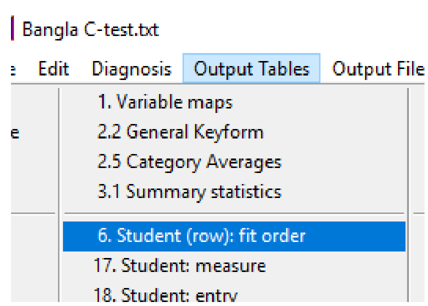
109.

Analyzing C-test Data-Part 2

110. Let's get back to analyzing some data! To select a final set of five texts for the Bangla C-test, we are going to be spending time in the following areas/screens: the item ICC plots, student fit indices, item fit indices, and our original .txt control file.

111. Let's start by seeing if we have any really strange response patterns by students in our data to begin with.

In the main Minstep window, click "**Output Tables**" and then "**6. Student (row): fit order**."



112. In the window that pops up, look at table 6.1 at the top.

Red box: Bangla test takers have been sorted according to fit values. Those who most underfit the data—those with higher infit/outfit mean square values—are towards the top. Those who overfit the data (and don't contribute much to measurement) are at the bottom. We are more concerned with outfit mean square values. The ideal outfit mean square is about 1.00, but the 0.50–1.50 range is productive for measurement. Anything about 2.00 really messes with the measurement properties of the test.

Blue box: We could very well have selected a different option in the "Output Tables" menu to order students in a different way. For instance, by selecting "17. Student: measure," students' are ordered by their ability measure, or their position on the Wright/variable map.

```
-------------------------
L|    INFIT    |   OUTFIT   ||
 |MNSQ   ZSTD|MNSQ   ZSTD|
 |-----------+-----------
1|2.31    2.4|2.41    2.5|
3|2.08    1.9|1.97    1.8|
3|2.00    1.8|1.76    1.5|
2|1.52    1.2|1.58    1.3|
5|1.48    1.1|1.57    1.2|
1|1.45    1.1|1.48    1.1|
```

```
6. Student (row): fit order
17. Student: measure
18. Student: entry
19. Student: alphabetical
42. Student: displacement
43. Student: correlation
7. Student: responses
```

113. If you look to the left of table 6.1, you can see that the student with a high outfit mean square value is "Entry Number 26." However, if you look in the rightmost column, the student is "Student 27." Values in the rightmost column correspond to entries in the Excel spreadsheet. Because there was one missing student, these values are offset.

```
----------------------------------------------------------------------------------------------------
|ENTRY   TOTAL  TOTAL               MODEL|   INFIT    |  OUTFIT   |PTMEASUR-AL|EXACT MATCH|        |
|NUMBER  SCORE  COUNT  MEASURE  S.E. |MNSQ  ZSTD|MNSQ  ZSTD|CORR.  EXP.| OBS%  EXP%| Student|
|-----------------------------------+----------+----------+-----------+-----------+--------|
|   26     88     10    -.38      .11|2.31   2.4|2.41   2.5|A .77   .78|   .0  11.2| 27     |
|   31     54     10    -.86      .13|2.08   1.9|1.97   1.8|B .35   .75| 30.0  17.8| 32     |
|   10     54     10    -.86      .13|2.00   1.8|1.76   1.5|C .37   .75| 30.0  17.8| 11     |
|   27     77     10    -.52      .12|1.52   1.2|1.58   1.3|D .87   .77|   .0  15.2| 28     |
|   28    197     10    1.17      .15|1.48   1.1|1.57   1.2|E .43   .62|   .0  14.2| 29     |
```

When it comes time to remove any misfitting students' or students' responses, as well as to cut down on texts, we need to remember to do in referring to entry numbers.

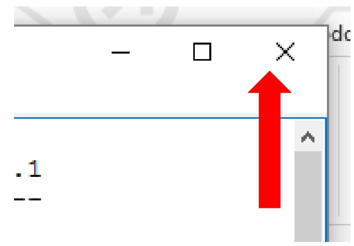
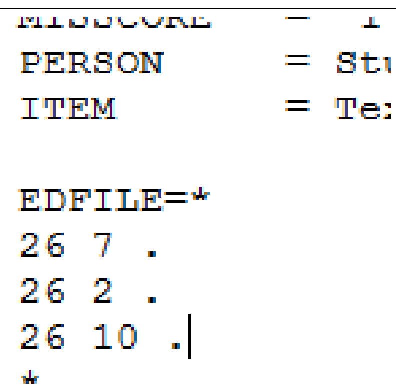
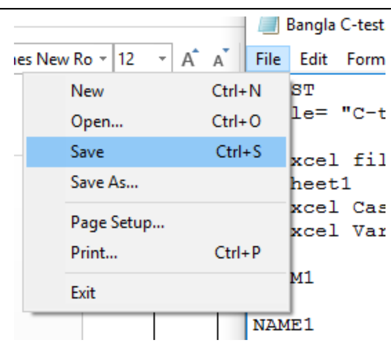
114. Scroll down to table 6.4, "Most Misfitting Response Strings."

This table is very handy; it shows us precisely which scores on which C-test text(s) were unexpected given students' Bangla proficiency levels.

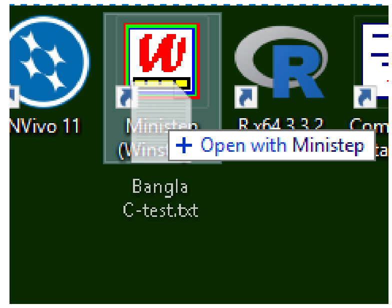
Red box: In the first "Student" column, students' entry numbers are shown. In the second column, student ID numbers (from the Excel spreadsheet are shown).

Blue box: These numbers correspond to text entry numbers. When there are 2+ digits, numbers are stacked vertically (as in text 10, for example).

```
MOST MISFITTING RESPONSE STRINGS
Student OUTMNSQ   |Text
                  |              1
                  | 5 3 6 7 8 2 0 1
               high----------------
26 27      2.41 A| . . . 1 . 011 .
31 32      1.97 B| 3 . . 9 . . . .
10 11      1.76 C| 3 . . 9 . . . .
27 28      1.58 D| . . . . . 0 . .
28 29      1.57 E| . .17 . . . . .
21 22      1.48 F| . . . 2 . . . 1
25 26      1.38 G| . . . . . . . 6
29 30      1.30 H| .10 . . . . . .
```

	Purple box: Ministep flagged student 26's responses to text 7 (1/24 blanks), text 2 (0/24 blanks), and text 10 (11/24 blanks) as misfitting. Before we begin to reduce our number of Bangla C-test texts from 10 to 5, let's begin by removing these misfitting responses. Now, in removing misfitting responses, we will be increasing the overall model fit. However, because we are working with such a small sample size, we want to preserve as much data as possible. So, instead of deleting an entire test taker's row of data (and their responses to all texts), we will only remove responses to individual texts.	
115.	Close out of the "6. Student (row): fit order" window.	
116.	Close Ministep.	
117.	Double click on the "Bangla C-test" control file. In the control file, somewhere below the specification lines, enter in the "EDFILE=*" information to the right. The "*'s" indicate the start and finish of a list of information. "27 7 ." means 'Mark person entry number 26's resposne to text #7 as missing. We repeat this information to mark 26's response to texts 2 and 10 as missing. Make sure to add a space between each value and the missing period.	
118.	Click "**File**" and "**Save**" to save the "Bangla C-test" control file.	

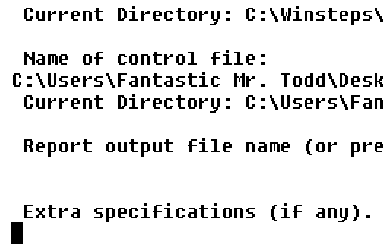
119. Drag and drop the "Bangla C-test" control file over the Ministep desktop icon to open it.



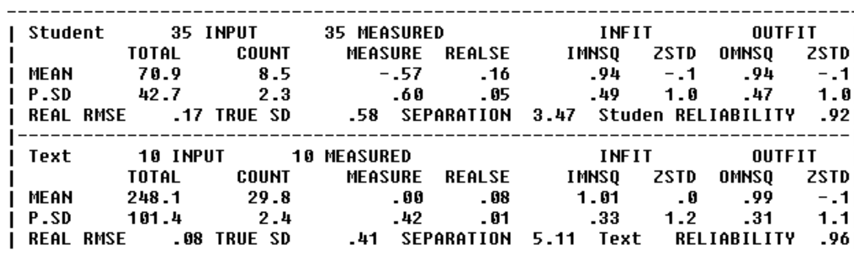
120. "Report output file name?" select "**Enter**."

"Extra specifications?" select "**Enter**."

The analysis runs.



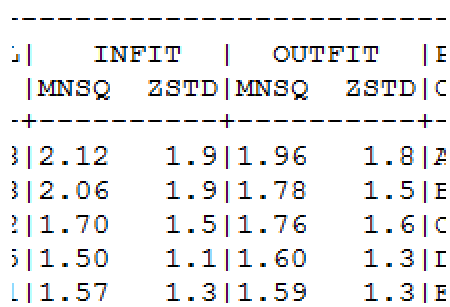
121. In the summary table at the bottom, notice now that the student and text separation have improved. The overall model fit has also improved, with outfit mean square values being closer to 1.00, which is ideal.



122. Let's return to table 6.1 to re-assess student fit.

In the main Minstep window, click "**Output Tables**" and then "**6. Student (row): fit order**."

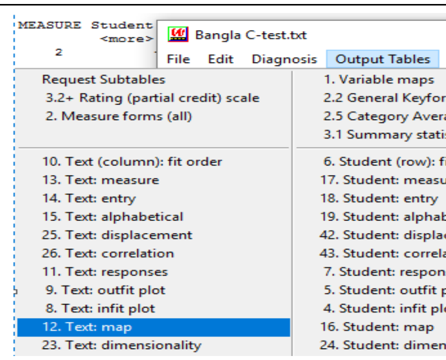
All student outfit mean square values are < 2.0, which is what we want. We can now begin to start removing a few texts!

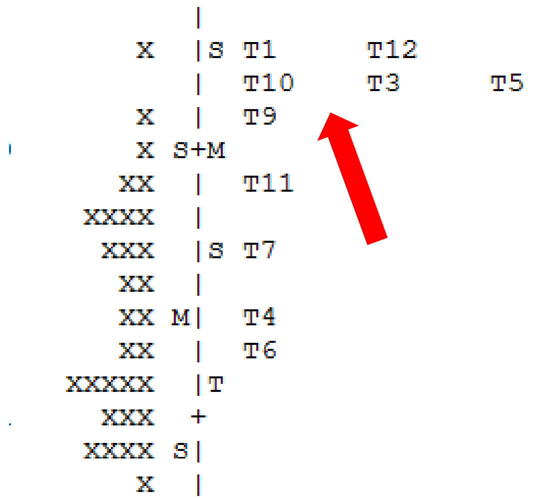
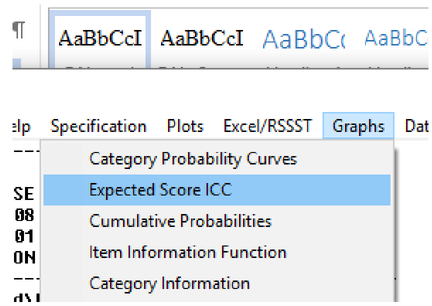
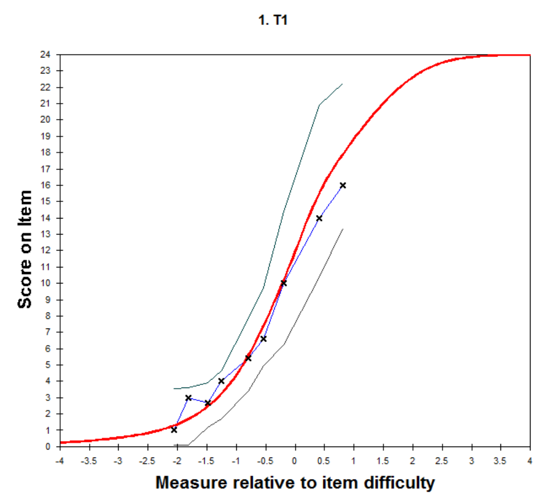


123. Let's remind ourselves of the hierarchy of texts in relation to students.

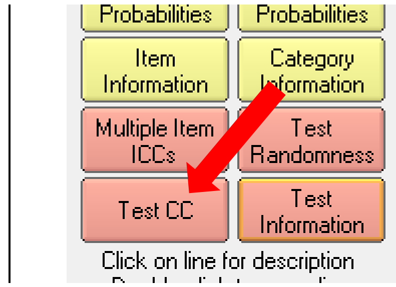
In the main Ministep page, click on "**Output Tables**."

Click on "**12. Text: map**."

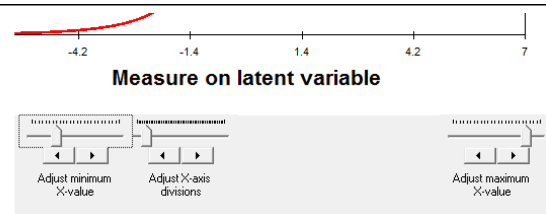


124.	We have quite a few texts bunched up together at the top of the Wright map, meaning they are at roughly the same level of difficulty. If we identify any problematic texts from among this group, they are prime candidates for removal as we look to optimize the C-test by reducing the overall number of texts from 10 to 5.	
125.	In the main Minstep window, click on "**Graphs**" and then "**Expected Score ICC**."	
126.	This is the item-characteristic curve, or ICC, for text #1. Now that we know what the Rasch model is and what it looks like, this graph should make more sense to you. Note that the values on the y-axis are not probabilities; they have been rescaled to correspond to blanks on the test. Gives us a very detailed view of resposne patterns to this particular item. The X's and blue line are the actual, observed data, while the red line is the model idea. Ideally, the X's (our data) fall right on the red line. The blue-gray lines around the red and blue lines are the confidence intervals. Any points that fall outside the confidence intervals are cause for concern. When we have misfitting response patterns, the blue line and series of X's gets pulled outside the confidence interval band. Each X on the blue line represents the group of test takers with ability measures at that level. For instance, Bangla learners with an ability of 1 on the x-axis have a 50-50 get about 16/24 blanks correct on text #1. Those with an ability of -0.5 score about 7/24 blanks on text #1. So far, text #1 looks okay. The observed data follows the red line, and there are no breaks outside of the confidence interval band.	

127. Click the red "**Test CC**" buttom at the bottom right of the screen.



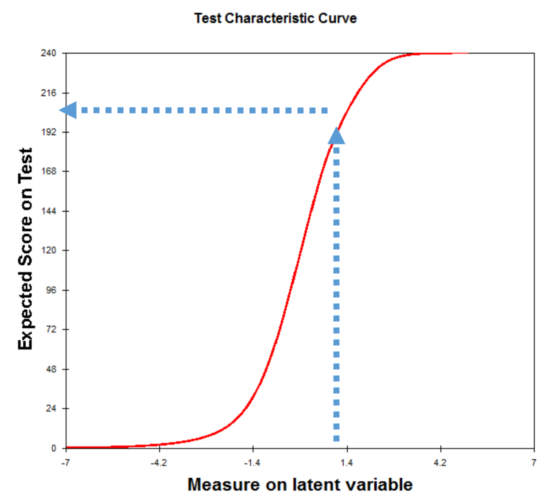
128. At the bottom of the plot screen, use the sliders to adjust the "maximum X-value" to 7 and the "minimum X-value" to -7.



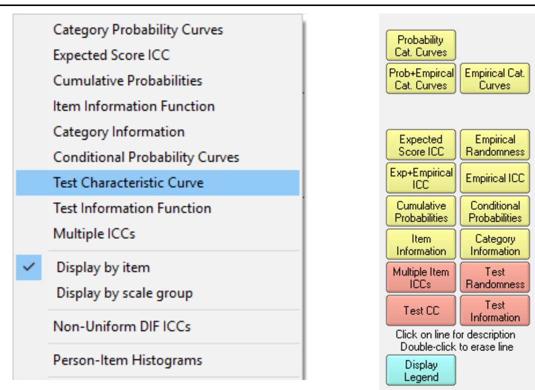
129. The test characteristic curve or TCC is shown to the right. Instead of looking at individual item ICCs like we did above, we can add them all up to get a sense of how students are scoring on the test as a whole given their ability level or their "measure on latent variable."

From the TCC curve on the right, we can see that students with an ability of 1.4 on the Bangla C-test are scoring in the neighborhood of about 200/250 blanks on the Bangla C-test. Again, scores on the y-axis have been rescale from probabilities to C-test blanks.

Since we are going to reduce the number of texts from 10 to 5, the TCC will change quite a bit. However, it's a very informative tool for test development purposes.

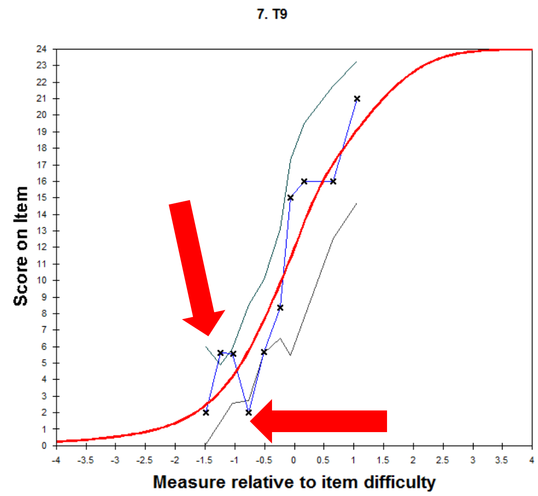


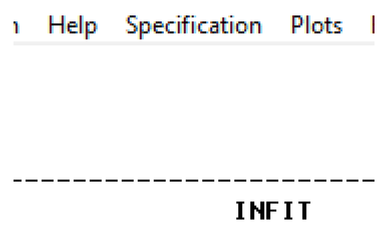
130. I want to point out that most of the information/plots that you can call for in the "Graphs" menu from the main Ministep page can also be accessed using the different colored buttons to the right of the plot screens that open up.



<table>
<tr><td>131.</td><td>Click the yellow “Exp+Empirical ICC” button to return to the ICCs.</td><td>
</td></tr>
<tr><td>132.</td><td>Click the green “Next Curve” button until you reach text 9 (“7. T9”).

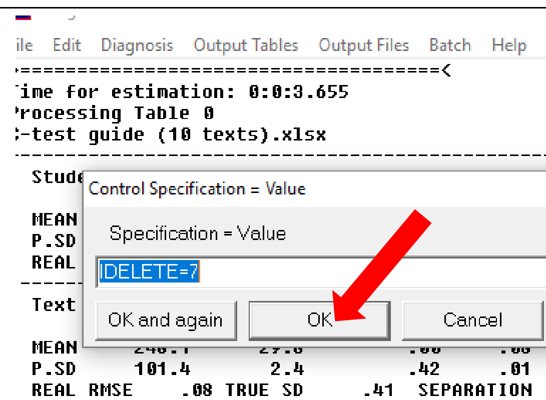
The empirical blue line for text 9 is problematic; it doesn’t follow the expected-model red line very closel. At two points, it jumps outside of the confidence-interval band.

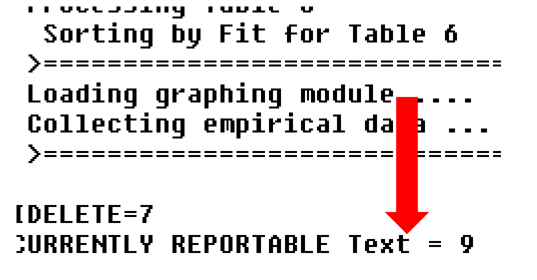
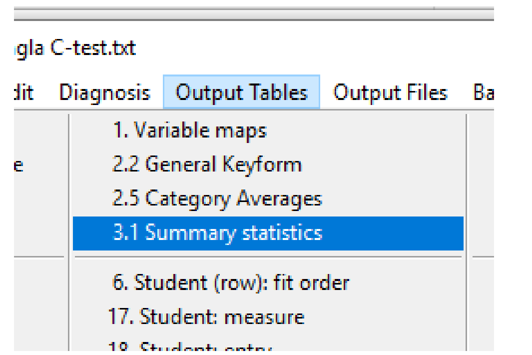
Let’s go ahead and remove text 9.</td><td>
</td></tr>
<tr><td>133.</td><td>There are two places where we can instruct Ministep to remove either a student or a text from the analysis.

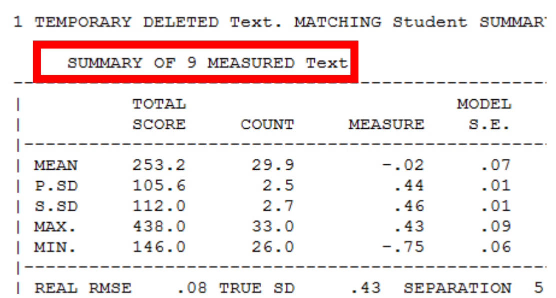
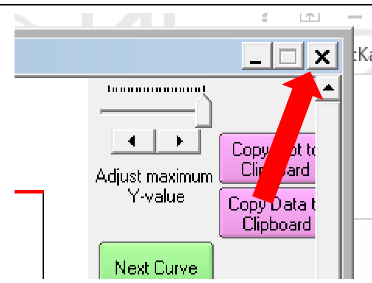
In the main Ministep page, click on “Specification.”</td><td>
</td></tr>
<tr><td>134.</td><td>We can enter additional commands in the “Control Specification = Value” window that pops up, just like entering specifications in our control file.

To delete a text or item, type “IDELETE=”. To remove a person from the analysis, type “PDELETE=.” You will have to specify the text or person you wish to remove by entering their entry number. “I” stands for ‘item,’ and “P” stands for ‘person.’ (You may have guessed.)

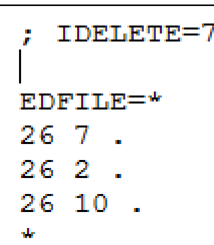
Let’s remove text 7. Type “IDELETE=7”.

Click “OK.”</td><td>
</td></tr>
</table>

<table>
<tr><td>135.</td><td>At the bottom left of the main window, you will notice that two new lines appear. The bottom-most lin now reads “Currently Reportable Text = 9,” meaning 9 texts are in the analysis.</td><td>
</td></tr>
<tr><td>136.</td><td>From the top menu bar, select “Output Tables” and then “3.1 Summary Statistics.”</td><td>
</td></tr>
<tr><td>137.</td><td>In table 3.1 that opens up, you can now see the summary text table reads “Summary of 9 Measured Text.”

The information in the summary table at the bottom of the main analysis window does not change. To change that, we need to close Ministep and enter the “IDELETE=” information elsewhere.</td><td>
</td></tr>
<tr><td>138.</td><td>Close the “Expected Score ICC” window.</td><td>
</td></tr>
<tr><td>139.</td><td>Close Ministep.</td><td></td></tr>
<tr><td>140.</td><td>Double click on the “Bangla C-test” control file.

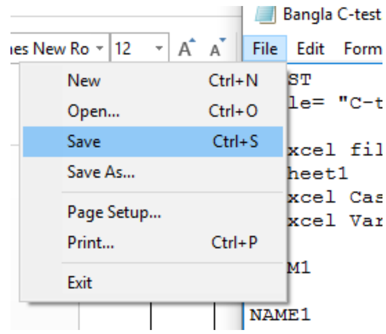
Somewhere in the control file, enter “; IDELETE=7” with no spaces. I like to put this line above the “EDFILE=*” line.

It’s always a good idea to keep a record of what responses/students you are removing as well as which</td><td>
</td></tr>
</table>

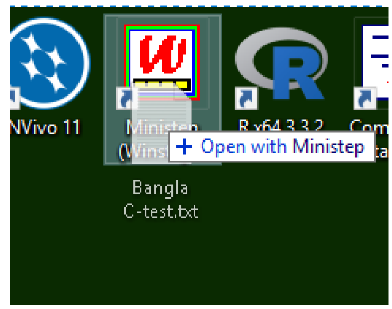
texts you are pulling out of the analysis. The control file is a good place to do just that. We have now noted that we are going to remove text 9 (entry #7), and we have ‘commented out’ this note with a semicolon.

Copy and paste the “**IDELETE=7**” part of the line to your clipboard.

141. Click “**File**” and “**Save**” to save the “Bangla C-test” control file.



142. Drag and drop the “Bangla C-test” control file over the Ministep desktop icon to open it.



143. “Report output file name?” select “**Enter**.”

“Extra specifications?” paste the “**IDELETE=7**” here and then select “Enter” on your keyboard. This is the second place where we can remove students and texts from our analysis.

The analysis runs.

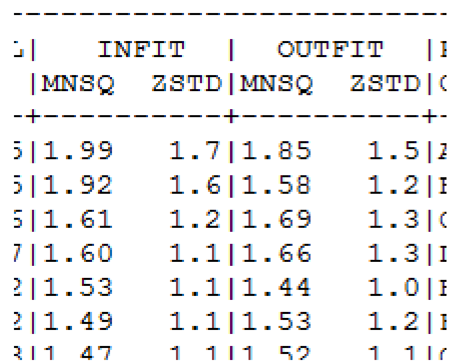
```
 Current Directory: C:\Winsteps\

 Name of control file:
C:\Users\Fantastic Mr. Todd\Desk
 Current Directory: C:\Users\Fan

 Report output file name (or pre

 Extra specifications (if any).
```

144. In the summary table at the bottom, we now see “9 Measured” in the text box. We successfully removed text 9 from the analysis.

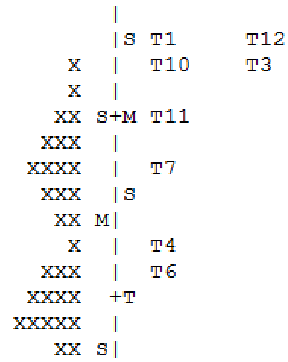
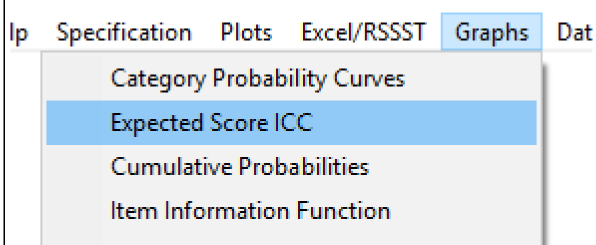
Person and item separation have gone down slightly, and model-fit indices are about the same. Person separation will often go down as we remove texts since we are removing statistical information used to calculate students’ abilities.

```
----------------------------------------------------------------------------
| Student     35 INPUT      35 MEASURED               INFIT         OUTFIT |
|          TOTAL      COUNT       MEASURE  REALSE    IMNSQ   ZSTD  OMNSQ   ZSTD|
| MEAN      65.1        7.7          -.58     .18      .95    -.1    .95    -.1|
| P.SD      38.8        2.1           .65     .05      .52    1.1    .50    1.1|
| REAL RMSE    .18 TRUE SD     .63  SEPARATION  3.42  Studen RELIABILITY  .92|
|----------------------------------------------------------------------------|
| Text       10 INPUT      9 MEASURED                INFIT         OUTFIT |
|          TOTAL      COUNT       MEASURE  REALSE    IMNSQ   ZSTD  OMNSQ   ZSTD|
| MEAN     253.2       29.9           .00     .08     1.02     .0    .99    -.1|
| P.SD     105.6        2.5           .47     .01      .35    1.2    .32    1.1|
| REAL RMSE    .08 TRUE SD     .46  SEPARATION  5.54  Text   RELIABILITY  .97|
----------------------------------------------------------------------------
```

<table>
<tr>
<td>145.</td>
<td>In the main Minstep window, click “Output Tables” and then “6. Student (row): fit order.”

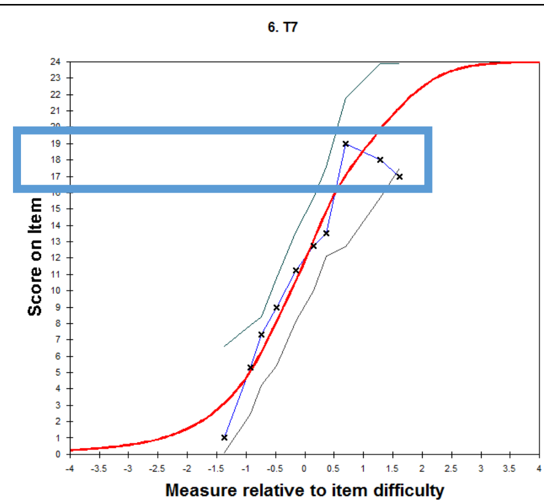
All student outfit mean square values are now < 2.0. Let’s press on to removing another text; 1 down and 4 more to go.</td>
<td>
</td>
</tr>
<tr>
<td>146.</td>
<td>Click on “Output Tables” and then “12. Text: map.” Let’s remind ourselves of the ‘lay of the land.’

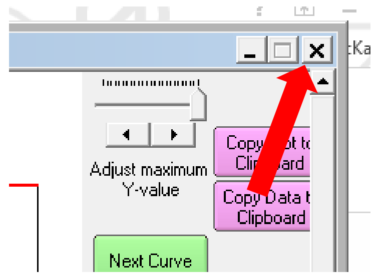
Texts 4 and 6 are still close together, as well as texts 1, 12, 10, 3, and 5.

Let’s revisit the ICC curves.</td>
<td>
</td>
</tr>
<tr>
<td>147.</td>
<td>Click on “Graphs” and then “Expected Score ICC.”</td>
<td>
</td>
</tr>
<tr>
<td>148.</td>
<td>Click the green “Next Curve” button until you reach text 7 (entry #6).

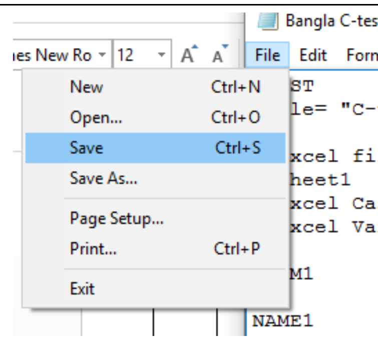
The empirical blue ine is doing two things we don’t want it to. It jumps outside the confidence-interval band at about 1.5 logits on the x-axis, and it hooks down towards the top. As ability increases, we want scores to increase, too. However, towards the top, as ability increases, scores decrease.

Blue box: The decrease that we’re observing only occurs over the range of 16-19/24 blanks. Something funny could be going on with these blanks, or more Bangla C-test takers could straighten the blue line out in this range. Regardless, in the spirit of informed, systematic text reduction, we select text 7 for removal.

Let’s remove text 7 from the analysis by adding information to the control file.</td>
<td>
</td>
</tr>
</table>

149.	Close the “Expected Score ICC” window.	
150.	Close Ministep.	

151.	Double click on the “Bangla C-test” control file to open it.
	
	Add “6” to the “; IDELETE=” line and then copy and paste “IDELETE=6,7” (again, with no spaces) to your clipboard.
		`; IDELETE=6,7`
		
		`EDFILE=*`
		`26 7 .`
		`26 2 .`
		`26 10 .`
		`*`

152.	Click “**File**” and “**Save**” to save the “Bangla C-test” control file.	
153.	Drag and drop the “Bangla C-test” control file over the Ministep desktop icon to open it.	

154.	“Report output file name?” select “**Enter**.”
	
	“Extra specifications?” paste the “**IDELETE=6,7**” here and then select “Enter” on your keyboard. This is the second place where we can remove students and texts from our analysis.
	
	The analysis runs.
		`Current Directory: C:\Winsteps\`
		
		`Name of control file:`
		`C:\Users\Fantastic Mr. Todd\Desk`
		`Current Directory: C:\Users\Fan`
		
		`Report output file name (or pre`
		
		`Extra specifications (if any).`

155. Make note of the changes in the summary table at the bottom of the analysis window. Student and text separation are still good. Fit has improved slightly.

```
-----------------------------------------------------------------------------------
| Student       35 INPUT       35 MEASURED                 INFIT        OUTFIT    |
|              TOTAL      COUNT      MEASURE  REALSE     IMNSQ  ZSTD  OMNSQ  ZSTD|
| MEAN          56.5       6.9         -.61     .19        .97   -.1    .96   -.1|
| P.SD          34.4       1.8          .66     .06        .59   1.1    .56   1.1|
| REAL RMSE      .20 TRUE SD      .63  SEPARATION  3.18  Studen RELIABILITY  .91|
|---------------------------------------------------------------------------------|
| Text          10 INPUT        8 MEASURED                 INFIT        OUTFIT    |
|              TOTAL      COUNT      MEASURE  REALSE     IMNSQ  ZSTD  OMNSQ  ZSTD|
| MEAN         247.2      30.0          .00     .08       1.02    .0    .99   -.1|
| P.SD         110.6       2.6          .47     .02        .37   1.3    .35   1.2|
| REAL RMSE      .08 TRUE SD      .47  SEPARATION  5.56  Text   RELIABILITY  .97|
-----------------------------------------------------------------------------------
```

156. In the main Minstep window, click "**Output Tables**" and then "**6. Student (row): fit order**."

We have one student with an outfit mean square > 2.0.

```
------------------------
 |   INFIT   |  OUTFIT  |E
 |MNSQ  ZSTD|MNSQ  ZSTD|C
-+-----------+----------+-
 |2.39   1.9|2.12   1.7|A
 |2.30   1.8|1.78   1.3|B
 |1.40    .7|1.94   1.1|C
 |1.82   1.4|1.85   1.5|D
 |1.65   1.3|1.68   1.3|E
```

157. Scroll down to table 6.4.

The misfit is due to student 32's responses to text 5 and 2.

Let's remove them from the analysis.

```
MOST MISFITTING RESPONSE STRINGS
Student OUTMNSQ   |Text
                  |             1
                  | 5 3 2 4 8 1 0
              high---------------
31 32       2.12 A| 3 . 7 . . . .
10 11       1.78 B| 3 . . . . 6 .
25 26       1.85 D| . . . . . 6 .
29 30       1.68 E| .10 . . . . .
```

158. Close out of table 6.1.

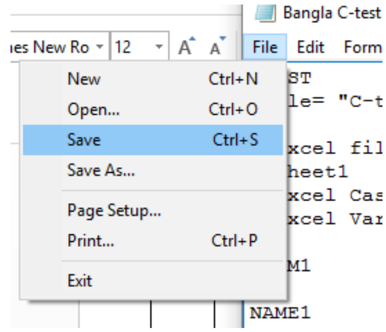
159. Close Ministep.

160. Double click on the "Bangla C-test" control file.

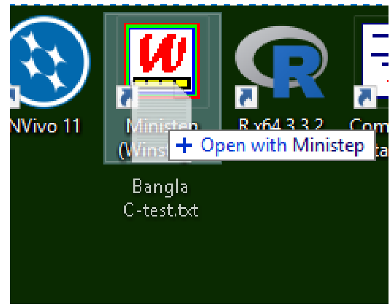
Add the removal information for student 32 (entry #31) to the "EDFILE=*" list.

```
; IDELETE=6,7

EDFILE=*
26 7 .
26 2 .
26 10 .
31 5 .
31 2 .
*
```

161. Click "**File**" and "**Save**" to save the "Bangla C-test" control file.



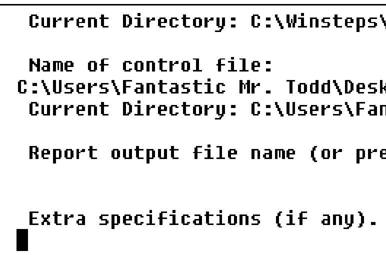
162. Drag and drop the "Bangla C-test" control file over the Ministep desktop icon to open it.



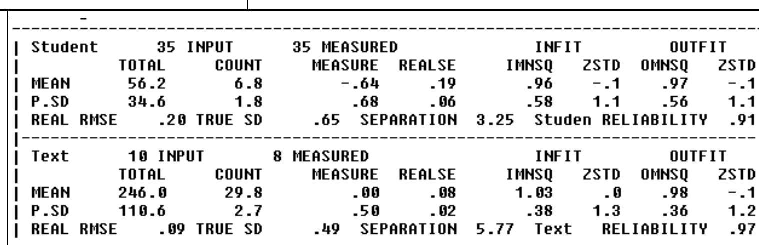
163. "Report output file name?" select "**Enter**."

"Extra specifications?" paste the "**IDELETE=6,7**" here and then select "Enter" on your keyboard. This is the second place where we can remove students and texts from our analysis.

The analysis runs.



164. Note the improved separation values and fit indices in the summary table. It's a good idea to monitor what's happening as we proceed.

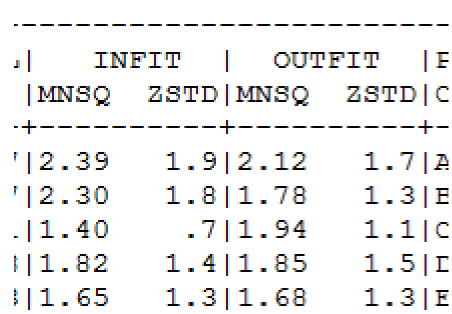


Student	35 INPUT		35 MEASURED		INFIT		OUTFIT	
	TOTAL	COUNT	MEASURE	REALSE	IMNSQ	ZSTD	OMNSQ	ZSTD
MEAN	56.2	6.8	-.64	.19	.96	-.1	.97	-.1
P.SD	34.6	1.8	.68	.06	.58	1.1	.56	1.1
REAL RMSE	.20	TRUE SD	.65	SEPARATION	3.25	Studen	RELIABILITY	.91

Text	10 INPUT		8 MEASURED		INFIT		OUTFIT	
	TOTAL	COUNT	MEASURE	REALSE	IMNSQ	ZSTD	OMNSQ	ZSTD
MEAN	246.0	29.8	.00	.08	1.03	.0	.98	-.1
P.SD	110.6	2.7	.50	.02	.38	1.3	.36	1.2
REAL RMSE	.09	TRUE SD	.49	SEPARATION	5.77	Text	RELIABILITY	.97

165. In the main Minstep window, click "**Output Tables**" and then "**6. Student (row): fit order**."

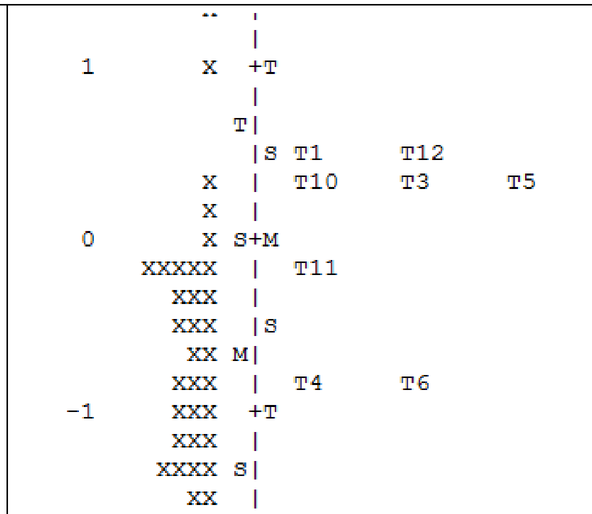
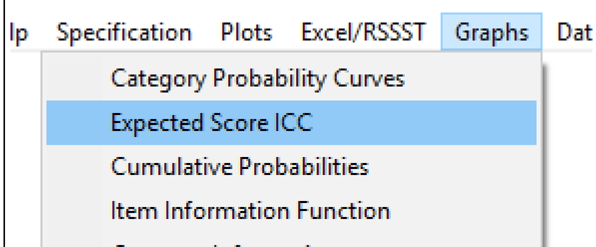
We have one student (entry #15) with an outfit mean square > 2.0, but that student's responses were not flagged as misfitting in table 6.4. So, we press on.



INFIT MNSQ	INFIT ZSTD	OUTFIT MNSQ	OUTFIT ZSTD
2.39	1.9	2.12	1.7
2.30	1.8	1.78	1.3
1.40	.7	1.94	1.1
1.82	1.4	1.85	1.5
1.65	1.3	1.68	1.3

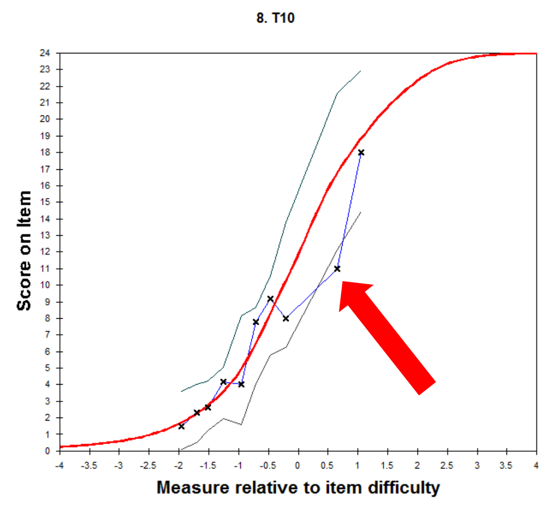
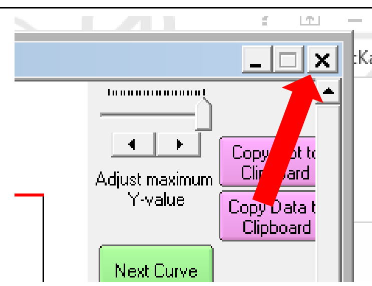
<table>
<tr><td>166.</td><td>Click on “Output Tables” and then “12. Text: map.” Let’s see what’s happened.

Texts 4 and 6 are now at the same difficulty level. Texts 1, 12, 10, 3, and 5 still form a cluster at about 0.5 logits at the top of the Wright map.

Let’s revisit the ICC curves to remove an additional text.</td><td>
</td></tr>
<tr><td>167.</td><td>Click on “Graphs” and then “Expected Score ICC.”</td><td>
</td></tr>
<tr><td>168.</td><td>Click the green “Next Curve” button until you get to text 10 (entry #8).

For text 10, we see a clear jump of the blue empirical line outside of the confidence-interval band.

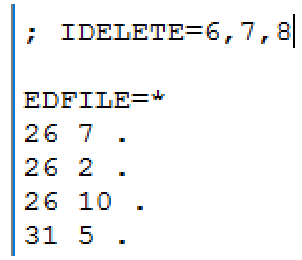
Let’s remove text 10 next.

By now, hopefully the process for going about removing misfitting responses and culling texts is becoming clearer!</td><td>
</td></tr>
<tr><td>169.</td><td>Close the “Expected Score ICC” window.</td><td>
</td></tr>
</table>

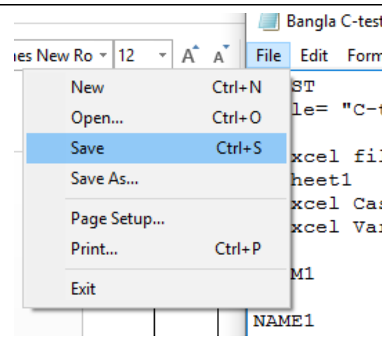
170. Close Ministep.

171. Double click on the “Bangla C-test” control file.

Add text 8 to the “; IDELETE=” specification line and copy and paste the “IDELETE=6,7,8” bit to your clipboard.



172. Click “**File**” and “**Save**” to save the “Bangla C-test” control file.

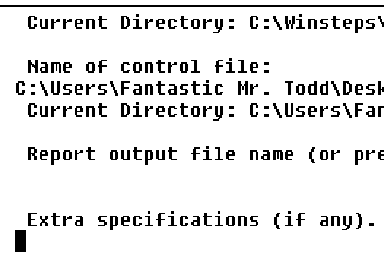


173. Drag and drop the “Bangla C-test” control file over the Ministep desktop icon to open it.

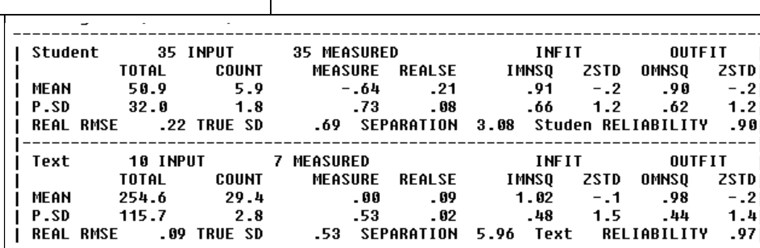
174. “Report output file name?” select “**Enter**.”

“Extra specifications?” paste the “**IDELETE=6,7,8**” here and then select “Enter” on your keyboard.

The analysis runs.



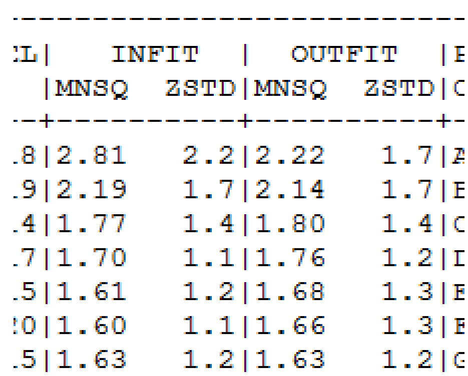
175. Take stock of the changes in the summary table at the end of the analysis. Student separation has gone down. The Student fit is worse, seeing an outfit mean square of .90. This means we probably have some misfitting response patterns given our new 7-text C-test.



```
| Student      35 INPUT       35 MEASURED               INFIT        OUTFIT   |
|             TOTAL    COUNT    MEASURE  REALSE     IMNSQ  ZSTD  OMNSQ  ZSTD|
| MEAN         50.9      5.9       -.64     .21       .91   -.2    .90   -.2|
| P.SD         32.0      1.8        .73     .08       .66   1.2    .62   1.2|
| REAL RMSE     .22 TRUE SD      .69  SEPARATION  3.08  Studen RELIABILITY  .90|
|-----------------------------------------------------------------------------|
| Text       10 INPUT        7 MEASURED               INFIT        OUTFIT   |
|             TOTAL    COUNT    MEASURE  REALSE     IMNSQ  ZSTD  OMNSQ  ZSTD|
| MEAN        254.6     29.4        .00     .09      1.02   -.1    .98   -.2|
| P.SD        115.7      2.8        .53     .02       .48   1.5    .44   1.4|
| REAL RMSE     .09 TRUE SD      .53  SEPARATION  5.96  Text   RELIABILITY  .97|
```

<table>
<tr>
<td>176.</td>
<td>In the main Minstep window, click "Output Tables" and then "6. Student (row): fit order."

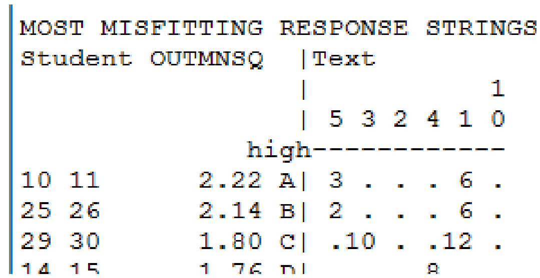
There are two students with outfit mean square values > 2.0. Note, however, that the standardized z-scores (which indicate the likelihood of the misfit) are < 2.0.

Here, with a small dataset, I would be inclinded to let an outfit mean square of 2.14 pass. The last thing we want to do is be blindly dichotomous in our decision-making when it comes to cutoff values. 2.0, while a value indicating noise in the measurement system, is nevertheless an arbitrary cutoff.</td>
<td>
<pre>

:L| INFIT | OUTFIT |E
  |MNSQ ZSTD|MNSQ ZSTD|C
--+----------+----------+-
.8|2.81   2.2|2.22   1.7|A
.9|2.19   1.7|2.14   1.7|E
.4|1.77   1.4|1.80   1.4|C
.7|1.70   1.1|1.76   1.2|I
.5|1.61   1.2|1.68   1.3|E
:0|1.60   1.1|1.66   1.3|E
.5|1.63   1.2|1.63   1.2|G
</pre></td>
</tr>
<tr>
<td>177.</td>
<td>However, if you scroll down to table 6.4, we see that the two misfitters, students 11 and 26 (entry #'s 10 and 25) have unexpected responses to texts 5 and 1.

Up until this point, the empirical blue line in the expected ICC plot of text 1 has been borderline, falling just outside the confidence-interval band at points.

Let's remove these misfitting responses to be consistent with decisions made so far. However, let's keep an eye on the ICCs for texts 5 and 6 to see if they continue to be problematic.

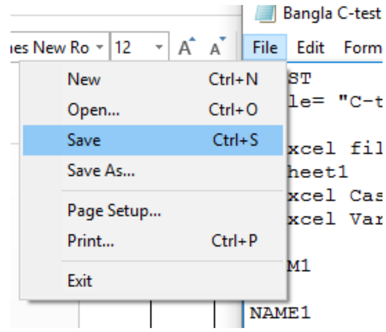
Also note that we have removed quite a few responses at this point. In a large dataset, removing a few misfitting student responses is not a big deal; however, we need to be able to carefully justify and understand the decisions we are making when it comes to removing responses in an alread sparse data set.</td>
<td>
<pre>
MOST MISFITTING RESPONSE STRINGS
Student OUTMNSQ |Text
                  | 1
                  | 5 3 2 4 1 0
              high------------
10 11   2.22 A| 3 . . . 6 .
25 26   2.14 B| 2 . . . 6 .
29 30   1.80 C| .10 . .12 .
</pre></td>
</tr>
<tr>
<td>178.</td>
<td>Close out of table 6.1.</td>
<td></td>
</tr>
<tr>
<td>179.</td>
<td>Close Ministep.</td>
<td></td>
</tr>
</table>

180. Double click on the "Bangla C-test" control file.

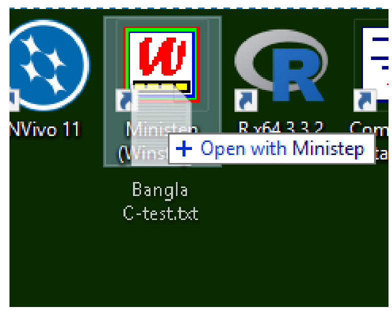
Red box: In the "EDFILE=*" list, entering the information to mark responses to texts 5 and 1 as missing for students 10 and 25.

```
; IDELETE=6,7,8

EDFILE=*
26 7 .
26 2 .
26 10 .
31 5 .
31 2 .
10 5 .
10 1 .
25 5 .
25 1 .
*
```

181. Click "**File**" and "**Save**" to save the "Bangla C-test" control file.



182. Drag and drop the "Bangla C-test" control file over the Ministep desktop icon to open it.



183. "Report output file name?" select "**Enter**."

"Extra specifications?" paste the "**IDELETE=6,7,8**" here and then select "Enter" on your keyboard.

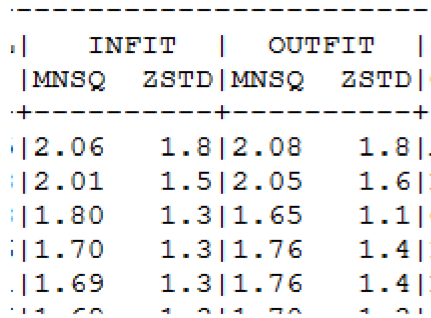
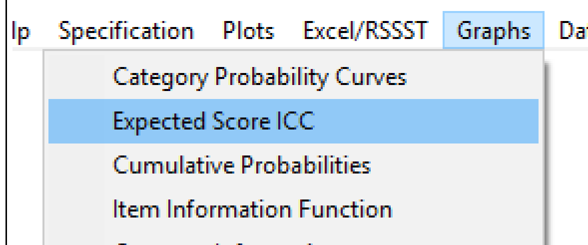
The analysis runs.

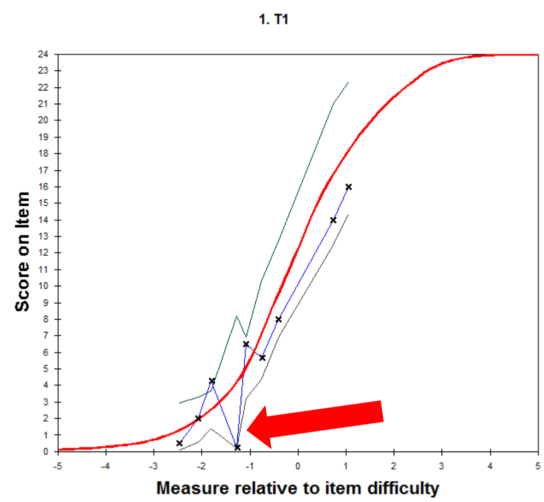
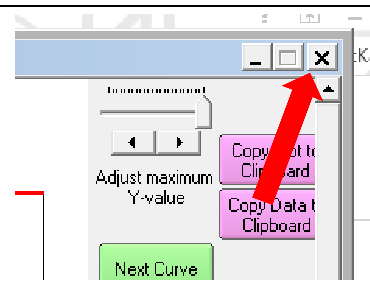
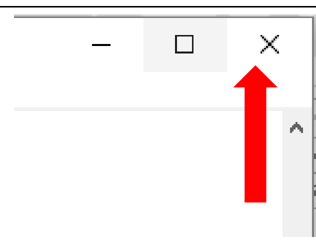
```
Current Directory: C:\Winsteps\

Name of control file:
C:\Users\Fantastic Mr. Todd\Desk
Current Directory: C:\Users\Fan

Report output file name (or pre

Extra specifications (if any).
```

184. As always, see what changed in the summary table at the end of the analysis. Student separation went up, but fit is still lower with an outfit mean square of .90.

```
-------------------------------------------------------------------------------
| Student      35 INPUT      35 MEASURED              INFIT          OUTFIT   |
|             TOTAL     COUNT       MEASURE  REALSE     IMNSQ   ZSTD  OMNSQ   ZSTD|
| MEAN         50.4       5.8         -.73     .22       .89    -.2    .90    -.2|
| P.SD         32.3       1.8          .82     .07       .59    1.1    .59    1.1|
| REAL RMSE     .24 TRUE SD      .78  SEPARATION  3.33  Studen RELIABILITY  .92|
|------------------------------------------------------------------------------|
| Text        10 INPUT      7 MEASURED               INFIT          OUTFIT   |
|             TOTAL     COUNT       MEASURE  REALSE     IMNSQ   ZSTD  OMNSQ   ZSTD|
| MEAN        252.1      28.9          .00     .09      1.05     .0    .99    -.1|
| P.SD        116.0       2.6          .63     .02       .51    1.5    .41    1.3|
| REAL RMSE     .10 TRUE SD      .62  SEPARATION  6.52  Text   RELIABILITY  .98|
-------------------------------------------------------------------------------
```

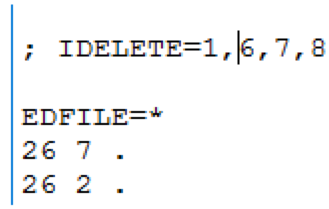
<table>
<tr><td>185.</td><td>In the main Minstep window, click “Output Tables” and then “6. Student (row): fit order.”

There are two students with outfit mean square values > 2.0. However, values are very close to 2.0. Having removed as many responses as we have so far, let’s keep these responses in the analysis and inspect the item ICCs.</td><td>
</td></tr>
<tr><td>186.</td><td>Click on “Graphs” and then “Expected Score ICC.”</td><td>
</td></tr>
<tr><td>187.</td><td>Right away, text 1 shows its ugly head. There’s a lot of zig-zagging amongst lower-ability students, and the empirical blue line jumps outside the confidence-interval band at about -2.0 logits.

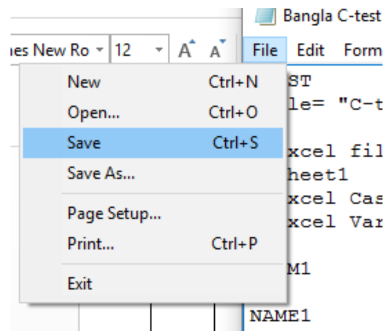
Some zigzagging in the empirical line is not altogether mind-blowing given the type of data we’re analyzing and the small test-taker sample size. However, we must proceed in as principled a manner as possible. We have the confidence-interval breach, some noted misfitting responses to text 1 in the last round of removed responses, and text 1 is at the same difficulty level as a few other texts. Let’s chuck it.</td><td>
</td></tr>
<tr><td>188.</td><td>Close the “Expected Score ICC” window.</td><td>
</td></tr>
<tr><td>189.</td><td>Close Ministep.</td><td></td></tr>
</table>

190. Double click on the "Bangla C-test" control file.

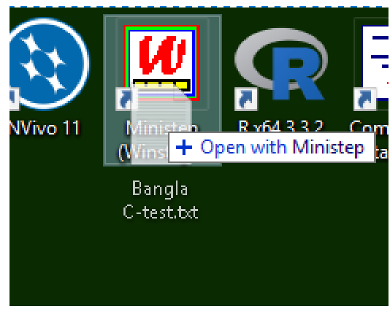
Add "1" to the "; IDELETE=" specification line and then copy and paste "IDELETE=1,6,7,8" to your clipboard.



191. Click "**File**" and "**Save**" to save the "Bangla C-test" control file.



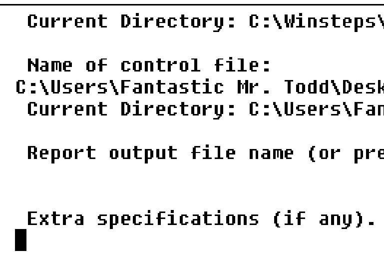
192. Drag and drop the "Bangla C-test" control file over the Ministep desktop icon to open it.



193. "Report output file name?" select "**Enter**."

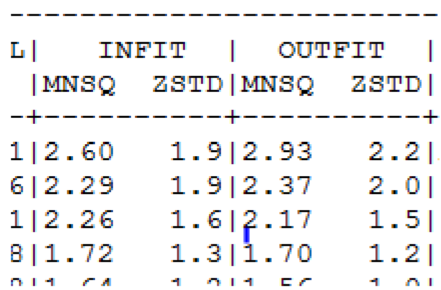
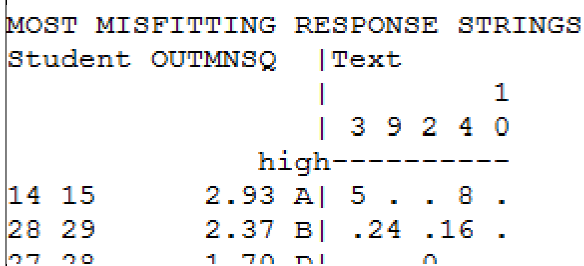
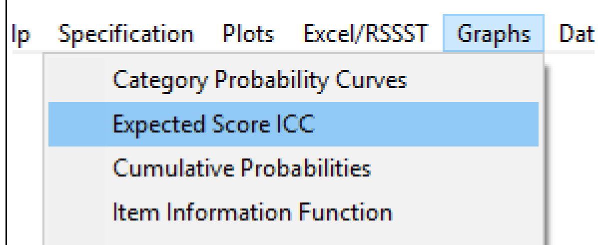
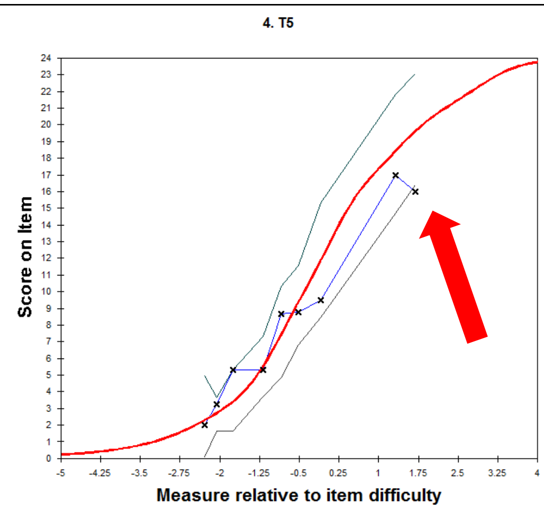
"Extra specifications?" paste the "**IDELETE=1,6,7,8**" here and then select "Enter" on your keyboard.

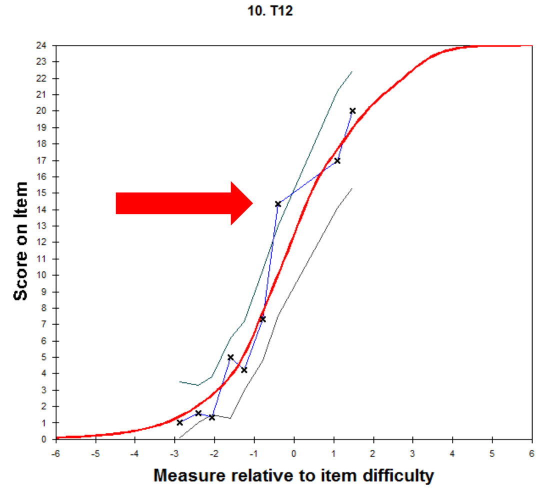
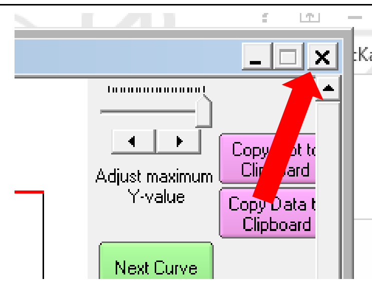
The analysis runs.



194. Student and text separation values have gone up. Text fit is slightly worse (again, having deleted yet another text), with an outfit mean square of .85. Note also that the overall error in the model is going up; the student "Real RMSE", or 'root mean squared error,' is at .27—up from .16 in our first, 10-text run!

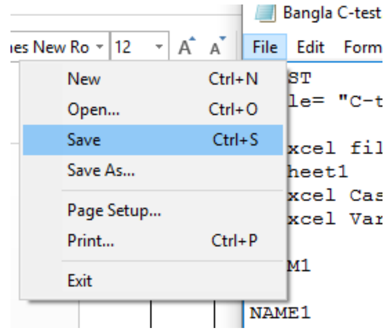
```
---------------------------------------------------------------------------
| Student      35 INPUT      35 MEASURED               INFIT        OUTFIT   |
|            TOTAL     COUNT      MEASURE  REALSE      IMNSQ  ZSTD  OMNSQ  ZSTD|
| MEAN        46.2       5.0       -.73     .26        .85   -.2    .85   -.3|
| P.SD        29.0       1.5        .99     .08        .68   1.1    .69   1.1|
| REAL RMSE    .27 TRUE SD        .96  SEPARATION  3.54  Studen RELIABILITY  .93|
|--------------------------------------------------------------------------|
| Text       10 INPUT      6 MEASURED                  INFIT        OUTFIT   |
|            TOTAL     COUNT      MEASURE  REALSE      IMNSQ  ZSTD  OMNSQ  ZSTD|
| MEAN       269.3      29.0        .00     .10        .96   -.1    .94   -.2|
| P.SD       116.8       2.8        .74     .01        .25    .9    .24    .9|
| REAL RMSE    .10 TRUE SD        .73  SEPARATION  7.56  Text   RELIABILITY  .98|
---------------------------------------------------------------------------
```

195.	In the main Minstep window, click "**Output Tables**" and then "**6. Student (row): fit order**." Alas, we have two more students with outfit mean square values > 2.0. Notably, the standardized *z*-scores are also > 2.0. For a small test-taker sample size, the *z*-scores are a bit more trustworthy than if we had had 500 test takers for the Bangla C-test, in which case all *z*-score values would have been > 2.0!	
196.	Scroll down to table 6.4. The most unexpected resposne patterns for students 15 and 19 (entries 14 and 28) are to texts 3, 4, and 9. Let's how these texts ICC curves stack up against one another and one of the other candidate texts for deletion before we remove any more data.	
197.	Click on "**Graphs**" and then "**Expected Score ICC**."	
198.	Click the green "Next Curve" button until you reach text 5 (entry #4). Scanning ahead, several texts are candidates for deletion at this point. There is one point at which the empirical blue line of text 5 shoots outside the grey confidence-interval band. The misfitting responses we identified in table 6.4 could account for the deviation from the red, Rasch-model line.	

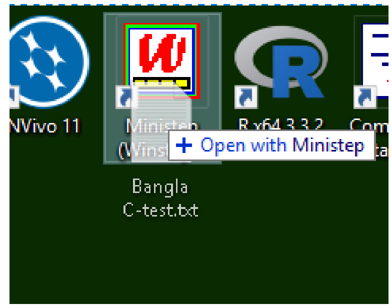
199.	Click the green "Next Curve" button until you reach text 12 (entry #10). This is perhaps the other candidate text for deletion. The empirical blue line jumps outside the confidence interval band at about -0.3 on the x-axis. We have two options at this point. We can continue to remove misfitting response patterns for students 15 and 19, and perhaps the ICC for text 5 will improve. Or, we refrain from removing any more data from the analysis and remove text 12 from the analysis. Crucial to note is that both texts 5 and 12 are at the same level of difficulty; they are both part of that cluster we identified at the top of our Wright map. Knowing this, the second option seems like the way to go: we can avoid deleting any more data, and we are still removing one of the initial texts we identified for removal. Let's remove text 12. After removing text 12, we have our final, 5-text C-test!	
200.	Close the "Expected Score ICC" window.	
201.	Close Ministep.	

202.	Double click on the "Bangla C-test" control file.
	
	Add text 12 (entry #10) to the "; IDELETE=" specification line in the control file and then copy and paste the "IDELETE=1,4,6,7,10" portion to your clipboard.
		`; IDELETE=1,4,6,7,10`
		
		`EDFILE=*`
		`26 7 .`
		`26 2 .`
		`26 10 .`
		`31 5`

203. Click "**File**" and "**Save**" to save the "Bangla C-test" control file.



204. Drag and drop the "Bangla C-test" control file over the Ministep desktop icon to open it.



205. "Report output file name?" select "**Enter**."

"Extra specifications?" paste the "**IDELETE= 1,6,7,8,10**" here and then select "Enter" on your keyboard.

The analysis runs.

```
 Current Directory: C:\Winsteps\

 Name of control file:
C:\Users\Fantastic Mr. Todd\Desk
 Current Directory: C:\Users\Fan

 Report output file name (or pre


 Extra specifications (if any).
█
```

206. Note the changes in the summary table at the bottom of the analysis page. Student separation is a bit lower, but it's still > 3.0 (with a reliability > .90) and we avoided additional data loss. Text separation and reliability are just fine. The fit is not as close to 1.00 as we would like, but it is not horrible. At about .90, the outfit mean squares for both students and texts indicate that our C-test slightly overfits the data, meaning it is slightly less informative overall.

```
-------------------------------------------------------------------------------
| Student      35 INPUT      35 MEASURED              INFIT           OUTFIT   |
|          TOTAL     COUNT    MEASURE  REALSE     IMNSQ   ZSTD  OMNSQ   ZSTD|
| MEAN      41.9       4.3      -.50     .26       .90    -.1    .91    -.1|
| P.SD      23.0       1.0       .87     .07       .65    1.0    .66    1.0|
| REAL RMSE    .27 TRUE SD   .83  SEPARATION  3.12  Studen RELIABILITY  .91|
|-------------------------------------------------------------------------------|
| Text       10 INPUT       5 MEASURED              INFIT           OUTFIT   |
|          TOTAL     COUNT    MEASURE  REALSE     IMNSQ   ZSTD  OMNSQ   ZSTD|
| MEAN     293.4      30.2       .00     .09       .95    -.2    .92    -.3|
| P.SD     113.3       2.8       .66     .01       .30    1.2    .32    1.3|
| REAL RMSE    .09 TRUE SD   .66  SEPARATION  7.34  Text   RELIABILITY  .98|
-------------------------------------------------------------------------------
```

207. Click on "**Output Tables**" and then "**12. Text: map**."

In the portion of the Wright map shown to the right, several points are worth noting.

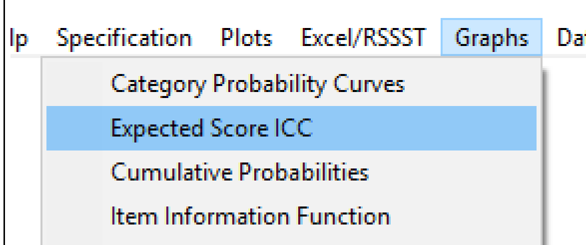
Texts 10 and 3 are close together at the top of the map; they are at the same level of difficulty. Future Bangla C-test developers might try to incorporate a more difficult text. Likewise, texts 4 and 6 are close together at the bottom of the map; they are also at roughly the same level of difficulty. Given the ability of the test-taker sample in this group (being on the lower end), it would be worthwhile to also try to incorporate an even easier text into the Bangla C-test.

Lastly, while texts 10 and 3 as well as 4 and 6 are close together, we are also working with few Bangla learners. One way to increase item separation, apart from selecting new items, is to also sample (more) test takers from a wider ability range. With a wider ability range of test takers, we have more statistical information the length of the Wright map to determine the location and hierarchy of texts. While including more difficult and easier texts would be prudent, it is also likely that future work with more 'middle' and 'advanced' learners could tell us with more certainty how the two sets of stack up against one another and whether they are, in fact, at such similar difficulty levels.

```
                |
          X     |
                |T
               T|
  1             +
                |
                |S  T10     T3
          X     |
        XXX    S|
        XXX     |   T11
  0       X     +M
      XXXXX     |
          X     |
         XX    M|
                |S  T4
          X     |   T6
 -1   XXXXX     +
         XX     |
       XXXX    S|T
       XXXX     |
                |
          X     |
```

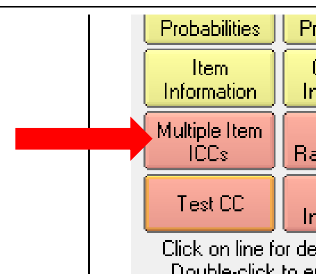
208. Click on "**Graphs**" and then "**Expected Score ICC**."

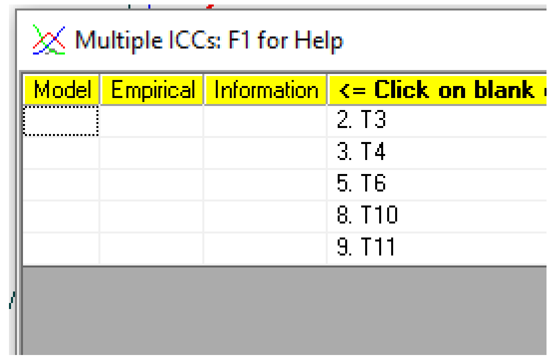
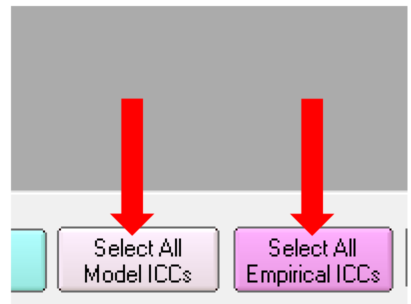
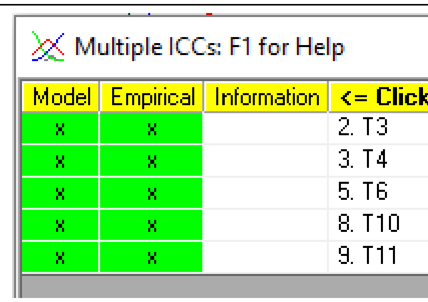
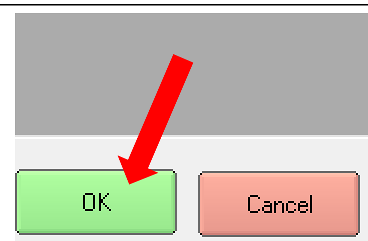
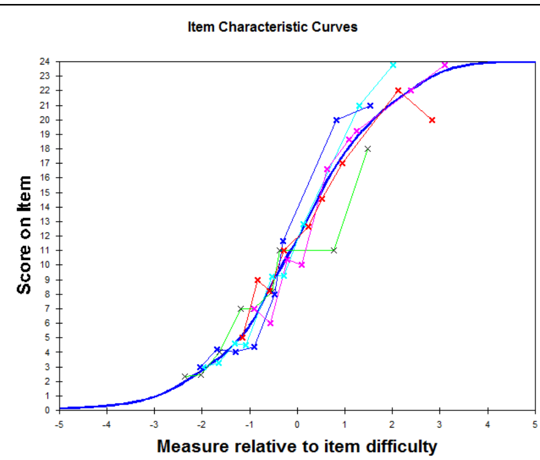
We've already looked at the individual item ICCs. There is some wonkiness in the ICCs here and there, and cleaning up misfitting responses would make the empirical blue lines fit the Rasch-model red lines better. However, in the interest of not deleting any more data, we're willing to accept some imperfection.

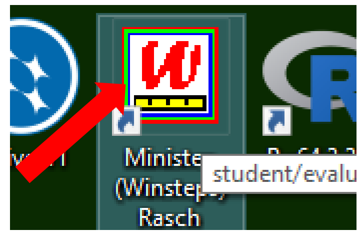
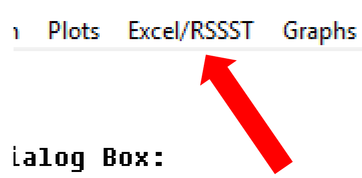
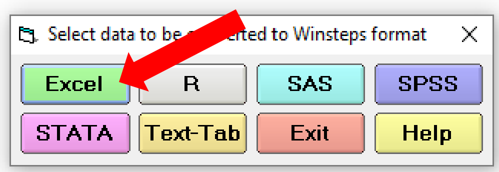
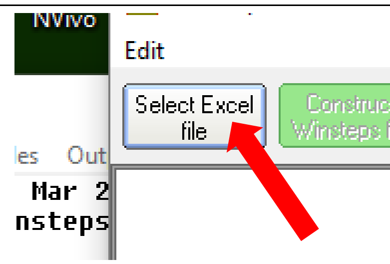
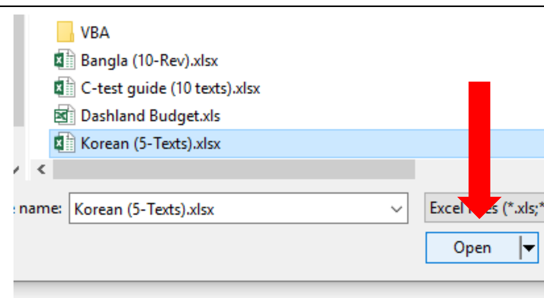


209. Click on the red "**Multiple Item ICCs**" button.

Let's look at all of our ICCs at the same time.

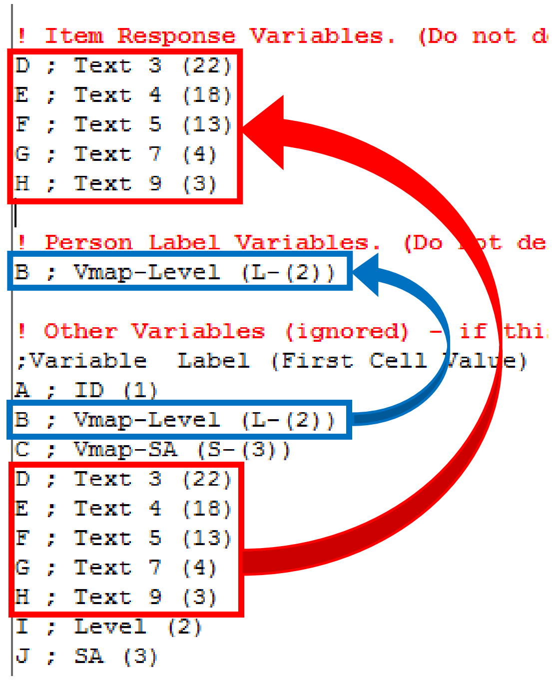
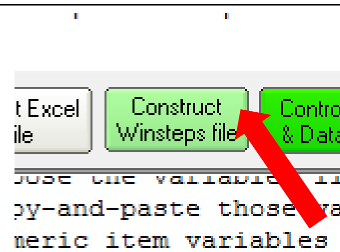
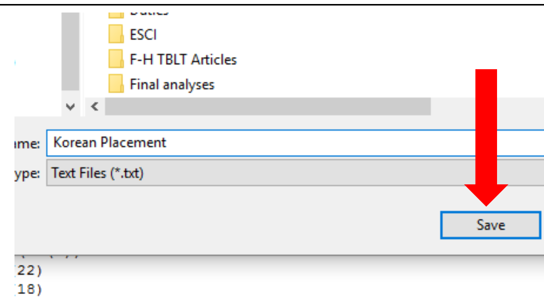
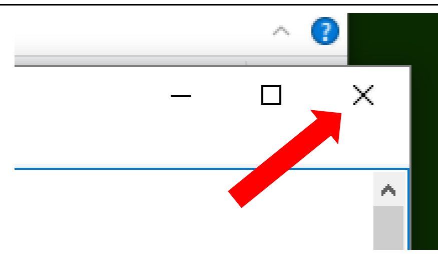


210.	In the window that pops up, you can call for item ICCs individually by selecting cells in the “Model” column for the expected model red lines that we’ve been looking at and the cells in the “Empirical” column to call for the empirical blue lines.	
211.	Click on the white “**Select All Model ICCs**” button and the purple “**Select All Empirical ICCs**” button.	
212.	All cells in the “Model” and “Empirical” columns should now be green.	
213.	Click “**OK**.”	
214.	The “Item Characteristic Curves” plot shows all of the text ICCs together along with the model curve (the blue line). This plot gives us a nice, birds-eye view on how well response patterns across texts fit the Rasch model. You can click on individual colored lines to investigate individual texts. The red line is for text 4, for example.	

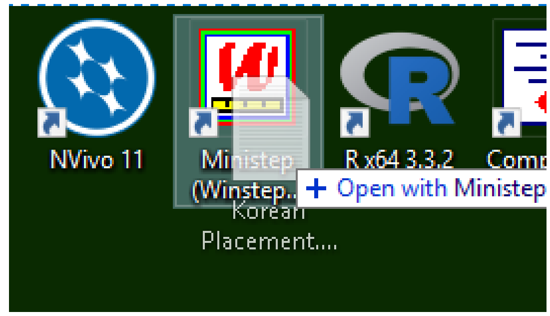
215.	**Possibility for Placement or Screening**	
216.	Let's do one more thing before we call it a day on the Bangla C-test. C-tests can be useful tools for placement and screening purposes, and the Wright or variable map available in the Ministep software can help in this regard. Now, unfortunately, the Bangla C-test data is not the best data set to illustrate how Ministep and Rasch could be used to help inform placement/screening decisions because of the number of heritage-language learners among test takers. So, we will use the Korean data set.	
217.	**Download from website: Check to make sure okay to use the Korean data.**	
218.	Double click on the "Ministep" icon to open it.	
219.	Click "**Excel/RSSST**" at the top of the Ministep window.	
220.	Click the green "**Excel**" button to create a control file for the "Korean (5-Texts)" data.	
221.	Click the "**Select Excel file**" button at the top left of the window that pops up to navigate to where you have the "Korean (5-Texts)" data saved.	
222.	Select the "Korean (5-Texts)" file and select "**Open**."	

<table>
<tr>
<td>223.</td>
<td>In the Excel spreadsheet for the “Korean (5-Texts)” data, just like in the Bangla data spreadsheet, there is a column in which the students’ institutional level was indicated (i.e., first-year Korean learners = “1,” second-year Korean learners = “2,” etc.). This column’s heading is “Vmap-Level” under the “! Other Variables” heading in Ministep window.

Red boxes: Copy-paste “Vmap-Level” under the “! Person Lable Variables” line (in the Bangla analysis, we copy-pasted the “ID” variable.

Blue boxes: Copy-paste the five text lines beneath the “! Item Response Variables” line. The “Korean (5-Texts)” spreadsheet contains columns not for 10 texts but 5.</td>
<td>
</td>
</tr>
<tr>
<td>224.</td>
<td>Click the green “Construct Winsteps file” button.</td>
<td>
</td>
</tr>
<tr>
<td>225.</td>
<td>Type “Korean Placement” into the “File name:” field and then click “Save.”</td>
<td>
</td>
</tr>
<tr>
<td>226.</td>
<td>Close the .txt window that opens up.</td>
<td></td>
</tr>
</table>

227. Drag and drop the "Korean Placement" file on the Ministep icon to open it.

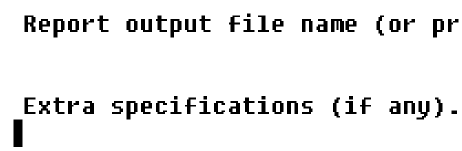


228. "Report output file name"

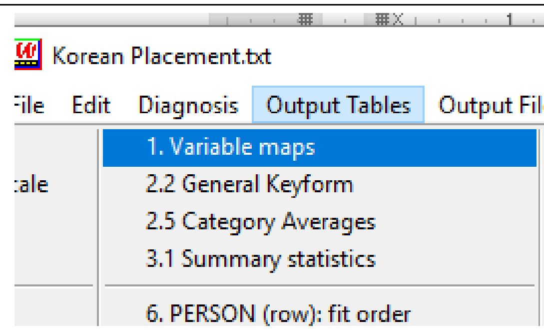
Press **Enter**.

"Extra specifications"

Press **Enter**.



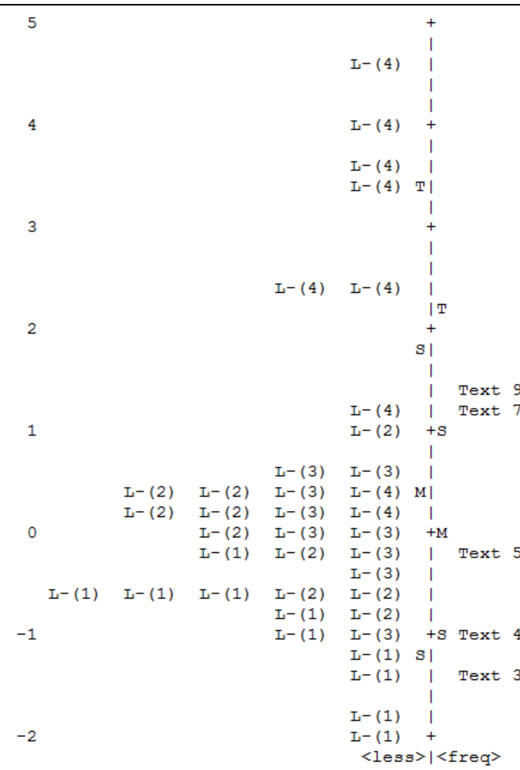
229. In the main Ministep window, select "**Output Tables**" and the select "**1. Variable maps**."



230. A variable or Wright map opens up. However, notice that, this time, instead of displaying X's for test takers, the test takers insitutional levels are shown. Again, "L-(4)" means 'level 4,' "L-(2)" means 'level 2,' etc.

The information contained in this chart speaks to the concurrent validity of the Korean C-test. At the bottom of the Wright map are test takers with lower Korean proficiency levels, and this is the portion of the map where we see many L-(1)'s. As we move up the map, we slowly see more L-(2)'s and L-(3)'s. At the top of the map are the highest proficiency learners, those in their fourth year of Korean study at their institutions. These test takers are shown by L-(4)'s. This array of test takers is what we would expect; those with more years of Korean study are scoring better on the Korean C-test.

How this information could be used to make placement or screening decisions is hopefully becoming apparent. If a Korean program knows that students in their fourth year of study typically score between 1–5 on the logit scale (on the left), then an incoming student who takes the Korean C-test and receives a logit score of 3 could reasonably be



placed in level 4, all things being equal. Likewise, an incoming student who receives a logit score of -1.5 would most likely be a candidate for first-year Korean, and so on.

Although we used the variable map and knowledge of students' institutional level to better understand the predictive validity of the Korean C-test, you could also plug self-assessment ratings beneath the "! Person Label Variables" line when setting up the control file to better understand the concurrent validity of your C-test. Those with higher self-assessment ratings, ideally, would be towards the top of the Wright map, while those with lower ratings would be towards the bottom.

231. **Summary**

That's it, folks! We have covered a lot of information in this guide; hopefully, you'll find the information helpful as you learn to analyze your own C-test data. With what you know about Rasch analysis and the Ministep software, you should be in a good position to understand what most of the output means and to reduce your own set of C-test texts from 10 (or however many you have to begin with) to 5.

We used the Bangla C-test data for this example, mainly because this is a data set we're familiar with; we know what the data look like, and we also are keenly aware of how C-tests were administered, who the test takers are, and what the individual texts are. The Bangla C-test data set, however, definitely has some shortcomings; the Bangla proficiency of learners is overall at the low end of the spectrum, there were few test takers overall (we commented a number of times on the small test-taker sample size), and we had a lot of heritage language learners. We can't emphasize enough that, regardless of the analysis being performed on the data, text selection and performance should be continuously monitored.

We also analyzed data using the rating-scale model, but we would encourage you to consider several different points of consideration when choosing between rating-scale and partial credit models to analyze your data. You should consider the following:

- The audience to whom you will need to communicate findings from your C-test analysis
- The structure of your C-test items, and how the structure is affected by the test-taker sample size
- The usefulness of the analysis

This last point is particularly poignant. In our experience, in most cases analyzing C-test data with rating-scale and partial credit models yields separation values that are sufficient for placement or screening purposes, or to get a quick-and-dirty 'snapshot' of test takers' proficiency levels. When all is said and done, Rasch analysis should be *useful* for you and your development team. If one model is more useful than the other, even though there might be small differences in person separation, use the more useful set of findings.