The Use of Matrix Variables in Examining DIF

Alejandra Osses and Ray Adams, 19 August 2015

The purpose of this tutorial is to illustrate the use of matrix variables. Matrix variables are internal (matrix valued) objects that can be created by various ConQuest procedures, or read into ConQuest and then manipulated. For example the `estimate` command can create matrix variables that store the outcomes of the estimation[1]. Matrix variables can be manipulated, saved or plotted.

In this Tutorial we show how subsets of the data can be analysed to evaluate differential item functioning. In this case we analyse differences between male and female students. We show how the results can be stored as matrix variables and how those matrices can be manipulated and plotted.

The files used in this sample analysis are:

`ex12.cqc`	The command statements.
`ex5.dat`	The data.
`ex6.lab`	The variable labels for the items on the multiple choice test.

The `ex5.dat` file contains achievement for 6800 students. Each line in the file represents one tested student. The first 19 columns of the data set contain identification and demographic information for each student. Columns 20 to 176 contain student responses to multiple-choice, and short and extended answer items. For the multiple-choice items, the codes 1, 2, 3, 4 and 5 are used to indicate the response alternatives to the items. For the short answer and extended response items, the codes 0, 1, 2 and 3 are used to indicate the student's score on the item. If an item was not presented to a student, the code `.` (dot/period) is used; if the student failed to attempt an item and that item is part of a block of non-attempts at the end of a test, then the code R is used. For all other non-attempts, the code M is used. More information about the `ex5.dat` file can be found in Adams & Wu (2010). An extract from the data file is shown in Figure 1.

[1] For a list of commands that can produce matrix variables and the content of those variables see the section "Matrix Objects Created by Analysis Commands" in the ConQuest Command Reference (https://www.acer.edu.au/files/Command-Reference.pdf).

```
         1         2         3
1234567890123456789012345678901234567 (column numbers)[2]

27       2201  711004313154134331232333......142M13
30       2201  85111121133434514..................
......   ......  ......
42       2201  82010232243......143421532132......
......   ......  ......
44       2201  85111134141411531..................
45       2201  86010532333......M41221............
47       2201  88010534345423221..................
```

Figure 1. Extract from the Data File `ex5.dat`

In this example, only data from columns 16 to 25 are used. Column 16 contains de code for the booklet that each student responded; the range is 1 to 8. Column 17 contains the code 0 for male students and 1 for female students. Column 18 contains the code 0 for lower grade (first year of secondary school) students and 1 for upper grade (second year of secondary school) students. Column 19 contains the product of columns 17 and 18, that is, it contains 1 for upper grade female students and 0 otherwise. Columns 20 to 25 contain the student responses to the first six items in the database. These six items are dichotomously scored.

In this sample analysis, the simple logistic model will be fitted to the data to analyse differences in item difficulty between boys and girls using graphic displays. The contents of a command file that can be used to analyse these data are shown in Figures 2, 5, 8 and 9. The command file is separated in three parts to clearly explain the steps to analyse the data in the proposed way.

[2] In each of the listings of the data file, column so you can easily identify the data column. The actual ConQuest data file does not have any column labels.

```
1.[3]  datafile ex5.dat;
2.     title TIMSS Mathematics--First Six Items;
3.     set constraint=cases;
4.     format book 16 gender 17 level 18 gbyl 19 responses 20-25;
5.     labels << ex6.lab;
6.     key 134423 ! 1;
7.     model item;
8.     keepcases 0! gender;
9.     estimate!matrixout=male;
10.    reset;
11.    datafile ex5.dat;
12.    title TIMSS Mathematics--First Six Items;
13.    set constraint=cases;
14.    format book 16 gender 17 level 18 gbyl 19 responses 20-25;
15.    labels << ex6.lab;
16.    key 134423 ! 1;
17.    model item;
18.    keepcases 1! gender;
19.    estimate!matrixout=female;
```

Figure 2. Sample Command File for a Dichotomous Test

1. The `datafile` statement indicates the name and location of the data file. Any file name that is valid for the operating system you are using can be used here.

2. The `title` statement specifies the title that is to appear at the top of any printed ConQuest output.

3. The `set` statement specifies new values for a range of ConQuest system variables. In this case, the use of the `constraints` argument is setting the identification constraints to `cases.` Therefore, the constraints will be set through the population model by forcing the means of the latent variables to be set to zero and allowing all item parameters (difficulty and discrimination) to be free.

4. The `format` statement describes the layout of the data in the file `ex5.dat.` This `format` statement indicates the fields name and their location in the data file. Thus, a field called `book` is located in column 16, a field called `gender` is located in column 17, and so on. Responses to the six items used in this tutorial are in columns 20 through 25 of the data file.

5. The `labels` statement indicates that a set of labels for the variables (in this case, the items) is to be read from the file `ex6.lab`. An extract of `ex6.lab` is shown in Figure 3. (This file must be text only; if you create or edit the file with a word processor, make sure that you save it using the text only option.)

[3] In each of the listings of the command file, each statement is labelled so that it can be easily referred to in the text. The actual ConQuest command file does not have any statement labels.

The first line of the file contains the special symbol ===> (a string of three equals signs and a greater than sign) followed by one or more spaces and then the name of the variable to which the labels are to apply (in this case, `item`). The subsequent lines contain two pieces of information separated by one or more spaces. The first value on each line is the level of the variable (in this case, `item`) to which a label is to be attached, and the second value is the label. If a label includes spaces, then it must be enclosed in double quotation marks `(" ")`. In this sample analysis, the label for item 1 is `BSMMA01`, the label for item 2 is `BSMMA02,` and so on.

```
===> book
1      book1
2      book2
3      book3
.      .
8      book8
===> gender
0      male
1      female
===> level
0      "lower grade"
1      "upper grade"
===> item
1      BSMMA01
2      BSMMA02
3      BSMMA03
.      .
.      .
```

Figure 3. Contents of the Label File `ex6.lab`

6. The `key` statement identifies the correct response for each of the multiple choice test items. In this case, the correct answer for item 1 is `1`, the correct answer for item 2 is `3`, the correct answer for item 3 is `4`, and so on. The length of the argument in the `key` statement is 6 characters, which is the length of the response block given in the `format` statement.

   If a `key` statement is provided, ConQuest will recode the data so that any response `1` to item 1 will be recoded to the value given in the key statement option (in this case, `1`). All other responses to item 1 will be recoded to the value of the `key_default` (in this case, `0`). Similarly, any response `3` to item 2 will be recoded to `1`, while all other responses to item 2 will be recoded to `0`; and so on.

7. The `model` statement must be provided before any traditional or item response analyses can be undertaken. In this example, the argument for the `model` statement is the name of the variable that identifies the response data that are to be analysed (in this case, item). By omitting the option statement we are fitting a rasch model where scores for each item are fixed.

8. The `keepcases` statement specifies a list of values for explicit variables that if not matched will be dropped from the analysis. The `keepcases` command can use two

possible types of matching. EXACT matching occurs when a code in the data is compared to a keep code value using an exact string match. A code will be treated as a keep value if the code string matches the keep string exactly, including leading or trailing blank characters. Values placed in double quotes are matched with this approach. The alternative is TRIM matching, which first trims leading and trailing spaces from both the keep string and the code string and then compares the results. Values not in quotes are matched with this approach. To ensure TRIM matching of a blank or a period character, the words `blank` and `dot` are used. The list of codes should be followed by the name of the explicit variables where these codes are to be found. If there is more than one variable, they should be comma separated.

In this case, we are keeping the code `0` for the variable `gender`, therefore modelling only males' responses. All cases with value `1` in this variable will be excluded from the analysis. By using the `keepcases` command we estimate separate item parameters for these two groups of students, producing separate matrix variables for males and females. We then use these matrix variables to evaluate DIF.

9. The `estimate` statement initiates the estimation of the item response model. The `matrixout` option indicates that a set of matrices with prefix `male_` will be created to hold the results. This matrix will be stored in the temporary workspace. Any existing matrices with matching names will be overwritten without warning.

   The Matrices produced by `estimate` depend upon the options chosen. The list of matrices is found in Figure 4 and their content is described in the section Matrix Objects Created by Analysis Commands in the 'ConQuest Command Reference'[4]. You can see these matrices using the `print` command or using the workspace menu in the GUI mode.

Default	itemparams	history	
If `method=jml` or `abilities=yes` in conjunction with an MML method	mle	wle	
If `abilities=yes` in conjunction with an MML method	pvs		
If `ifit=yes`	Itemfit		
If `pfit=yes`	casefit		
If `stderr=empirical`	estimatecovariances		
If `stderr=quick`	Itemerrors	regressionerrors	covarianceerrors

Figure 4. Matrices created by the `estimate` command

10. The `reset` command resets ConQuest system values to their default values, except for tokens and variables. The command is used here to erase the effects of previously issued commands.

[4] The latest ConQuest Command Reference can be downloaded at https://www.acer.edu.au/files/Command-Reference.pdf

11. The next set of commands is exactly the same to that mentioned above, with the exception of the last two (`estimate` and `keepcases`). In this part of the `ex12.cqc` file, we are modelling responses for females. Therefore, the `keepcases` statement instructs ConQuest to keep in the analysis only those cases where the value of the variable `gender` equals `1`. A set of matrices named with the prefix `female_` will hold the results of the estimated model (`estimate` statement).

The second part of the ex12.cqc file, showed in Figure 5, extracts data from the two matrices created above with the estimate statement. The data is used to create an identity line and then plotted to show differences in item difficulty for males and females.

```
22. /* create data to plot an identity line */
23. compute itemparams=male_itemparams->female_itemparams;
24. let identityx=matrix(2:1);
25. let identityy=matrix(2:1);
26. compute identityx[1,1]=min(itemparams);
27. compute identityy[1,1]=min(itemparams);
28. compute identityx[2,1]=max(itemparams);
29. compute identityy[2,1]=max(itemparams);
30.
31.
32. /* plot the relationship */
33.
34. scatter identityx, identityy! join=yes, seriesname=identity;
35. scatter male_itemparams, female_itemparams! overlay=yes,
36.                         legend=yes,
37.                         xmax=1,
38.                         xmin=-2,
39.                         ymax=1,
40.                         ymin=-2,
41.                         seriesname=male vs female,
42.                         title=Comparison of Item Parameter Estimates,
43.                         subtitle=Male versus Female;
```

Figure 5. Sample Command File for a Dichotomous Test (part II)

23. The compute command takes the male_itemparams and the female_itemparams object from the matrices created with the estimate statements. By using the -> operator these two matrices are concatenated in a new matrix named itemparams. The new matrix contains six rows and two columns. The rows, one for each item, contain the estimated item location parameters (difficulty) and the columns correspond to student gender, male and female. For a list of compute command operators and functions see the ConQuest Command Reference[5].

24. The two let statements define two empty matrices, identityx and identityy, each with two rows and one column. These matrices allow us to draw the identity line in the scatter plot created below.

26. The compute statements fill the two newly created matrices with the minimum and maximum values observed in the matrix itemparams. Both matrices are filled with the same values.

[5] The latest ConQuest Command Reference can be downloaded at https://www.acer.edu.au/files/Command-Reference.pdf

34. The `scatter` statement produces a scatter plot of two variables. In this case, `identityx` and `identityy`. The `join` option indicates that the two points are to be joined by a line; in this case, the identity line. The `seriesname` option defines the text to be used as a series name. The plot is displayed as a separate window in the screen and is shown in Figure 6.

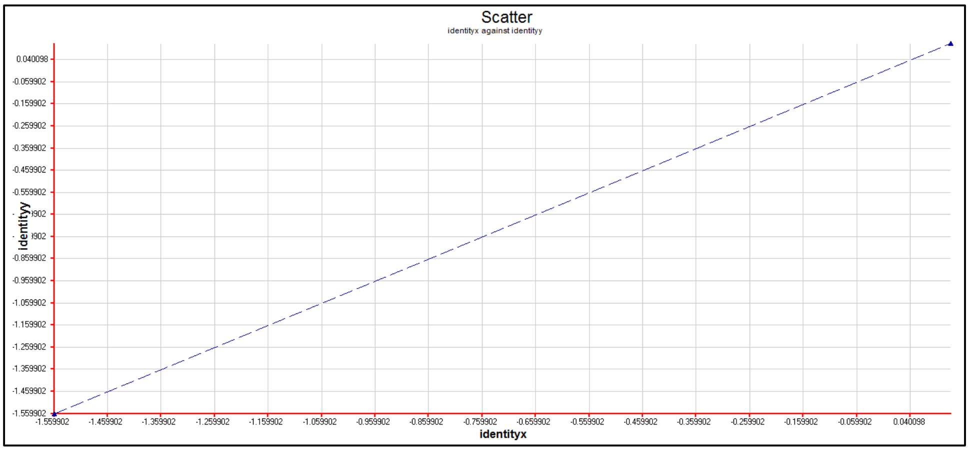

Figure 6. Scatter plot for the identity line

35. The second `scatter` statement produces a scatter plot of the item parameters for males and females (Figure 7). The `overlay` option allows the resulting plot to be overlayed on the existing active plot. In this case, results will be overlayed with the identity line shown in Figure 6. The option `legend` indicates that legend is displayed. The `xmax`, `xmin`, `ymax` and `ymin` options set the maximum and minimum values for the horizontal and vertical axes of the plot, respectively and overwrite the values on the previous plot. The seriesname option specifies the text to be used as series name. The `title` and `subtitle` options specify the text to be used as title and subtitle of the plot.

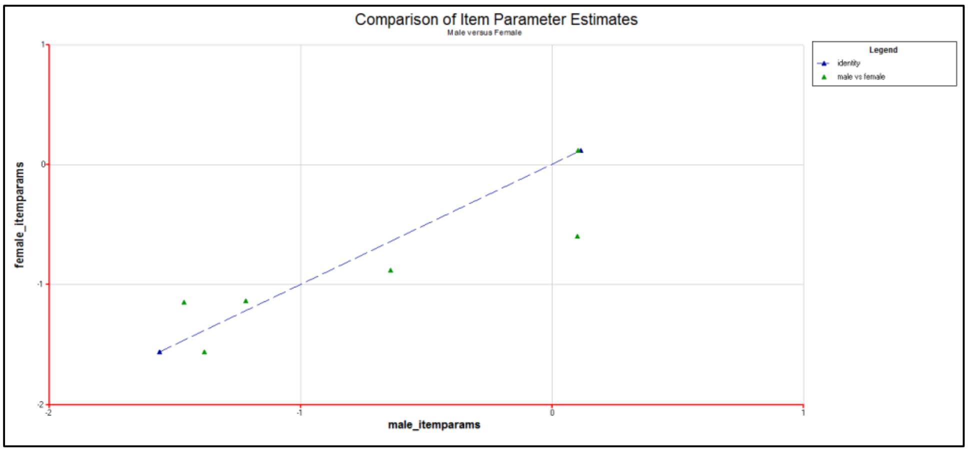

Figure 7. Scatter plot of item parameters for males and females

The third and fourth parts of the ex12.cqc file are displayed in Figures 8 and 9. This set of statements centres the item parameters for both groups on zero and computes the difference between them for each item. With these results and the standard errors from the covariance matrix, a scatter plot is produced to display the Wald test of differences between the two groups (Engle, 1984). The plot also includes 95% confident levels for the Wald test.

```
46.  /*centre the item parameter estimates for both groups on zero
47.   and compute differences */
48.
49.
50.  compute male_itemparams=male_itemparams-
     sum(male_itemparams)/rows(male_itemparams);
51.  compute female_itemparams=female_itemparams-
     sum(female_itemparams)/rows(female_itemparams);
52.  compute difference=male_itemparams-female_itemparams;
53.
54.
55.  /*extract the standard errors from the error covariance matrix */
56.
57.  let var_male=matrix(6:1);
58.  let var_female=matrix(6:1);
59.  for (i in 1:6)
60.  {
61.      compute var_male[i,1]=male_estimatecovariances[i,i];
62.      compute var_female[i,1]=female_estimatecovariances[i,i];
63.  };
64.
65.
66.
67.  /* create data to plot upper and low 95% CI on Wald test */
68.
69.
70.  let upx=matrix(2:1);
71.  let upy=matrix(2:1);
72.  let downx=matrix(2:1);
73.  let downy=matrix(2:1);
74.  compute upx[1,1]=1;
75.  compute upy[1,1]=1.96;
76.  compute upx[2,1]=rows(difference);
77.  compute upy[2,1]=1.96;
78.  compute downx[1,1]=1;
79.  compute downy[1,1]=-1.96;
80.  compute downx[2,1]=rows(difference);
81.  compute downy[2,1]=-1.96;
82.
83.  compute item=counter(rows(difference));
```

Figure 8. Sample Command File for a Dichotomous Test (part III)

50. The compute statement centres the item parameters (eg male_itemparams) by subtracting the mean of the item difficulties (eg

`sum(male_itemparams)/rows(male_itemparams)`) to each item. A matrix with the centred values of item parameters is computed for each group. The difference of item difficulties between the two groups is also computed and stored in a new matrix named `difference`.

57. The `let` statements create two 6 by 1 empty matrices — one for each group.

59. The `for` statement fills the above created matrices with the values of the estimate error variance for each item. These values are found in the diagonal of the estimates error variance-covariance matrix that is produced in the `estimate` statement (rows 9 and 18 in Figure 2).

70. The `let` statements create four 2 by 1 empty matrices, `upx`, `upy`, `downx`, and `downy` so we can plot the confidence interval lines in the plot.

74. The `compute` statements fill the matrices with the following values. The element in the first row and column (i.e., `1,1]`) of the matrices `upx` and `downx` with the number `1`. The element in the second row and first column (i.e., `[2,1]`) of the matrices `upx` and `downx` with the number of rows of the `difference` matrix (i.e., 6). The first and second rows of the matrices `upy` and `downy` with the number 1.96 and -1.96, respectively.

83. The `compute` statement creates a variable named item. The function `counter` creates a matrix with dimensions equal to the number of rows in the `difference` matrix (i.e., 6) by 1 filled with integers running from 1 to 6. This serves for producing the horizontal axis in the scatter plot described in Figure 9.

```
84.
85.
86.   /* calculate SE of difference and Wald test */
87.   compute se_difference=sqrt(var_male+var_female);
88.   compute wald=difference//se_difference;
89.
90.
91.   /* plot standard differences */
92.
93.   scatter upx, upy! join=yes, seriesname=95 PCT CI Upper;
94.   scatter downx, downy! join=yes, overlay=yes, seriesname=95 PCT CI
      Lower;
95.   scatter item, wald! join=yes,
96.                       overlay=yes,
97.                       legend=yes,
98.                       seriesname=Wald Values,
99.                       title=Wald Tests by Item,
100.                      subtitle=Male versus Female;
```

Figure 9. Sample Command File for a Dichotomous Test (part IV)

87. The `compute` statements define two 6 by 1matrices, `se_difference` and `wald`. The row values in the first of these matrices correspond to the square root (`sqrt`) of the sum of variances for each item between groups (`var_male+var_female`). By using the `//`

operator, the values in the Wald matrix are computed as the division of each element in the `difference` matrix by the matching element in the `se_difference` matrix. The Wald test can be used to test for standard differences in item parameters between two groups, males and females in this case.

93. The `scatter` statement produces a scatter plot of the `upx` and `upy` matrix variables. The plot is displayed on a new window. The values 1 and 6 in the horizontal axis and the value 1.96 in the vertical axis. The option `join` specifies a line that joins the points in the horizontal axis. The `seriesname` option defines the text to be used as series name.

94. The `scatter` statement produces a scatter plot of the `downx` and `downy` matrix variables. The values 1 and 6 in the horizontal axis and the value -1.96 in the vertical axis. The option `join` specifies a line that joins the points in the horizontal axis. The `overlay` option indicates that the resulting plot is overlayed with the active plot produced by the previous `scatter` statement. The `seriesname` option defines the text to be used as series name.

95. The last `scatter` statement produces a scatter plot of the item and Wald matrix variables (Figure 10). The `item` matrix, with values from 1 to 6 is displayed in the horizontal axis. And the Wald matrix in the vertical axis. The plot is overlayed with the active plot produced by the two previous `scatter` statements by using the option `overlay`. The legend is set to be displayed by using the option `legend`. The name of the new series added to the plot is set with the `seriesname` option. The title and subtitle are also specified with the corresponding options.

To avoid having a large number of decimal places in the values of the Wald test you have two options. One is to specify the upper and lower values of the vertical axis using the `ymax` and `ymin` options in the scatter statement. Another is to manipulate the graph via the PlotQuest window menus. The second approach is the one we used in Figure 10.

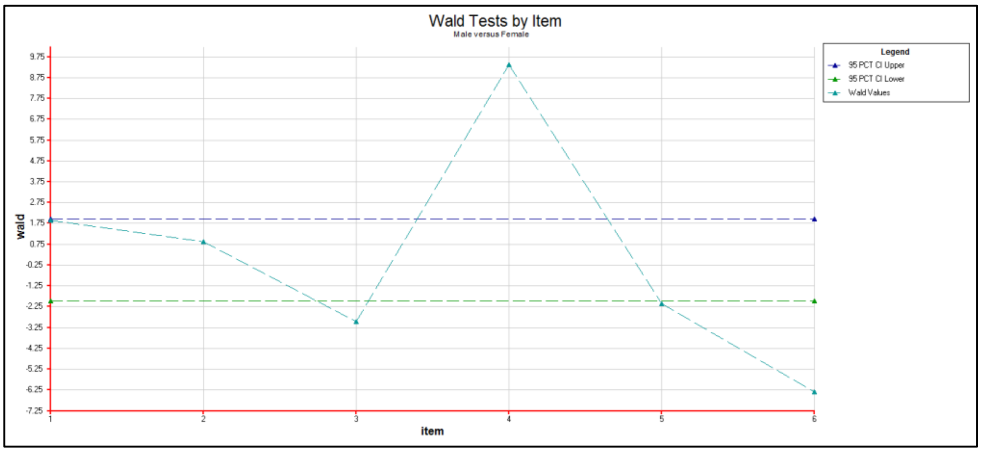

Figure 10. Wald test for standardised differences in item estimates between males and females

RUNNING THE ANALYSIS

To run this sample analysis, start the GUI version. Open the file `ex12.cqc` and choose Run➔Run All. ConQuest will begin executing the statements that are in the cqc file; and as they are executed they will be echoed in the Output Window. When it reaches the estimation command ConQuest will begin fitting the two-parameter model to the data. This analysis will converge in 31 iterations.

After the estimation is completed, the `scatter` statements will produce two plots that will be displayed in new windows. The first of these plots contains a comparison of the item parameter estimates for males and females, and also displays the identity line. The second plot contains the Wald test of standardized differences in item parameters for these two groups, along with the 95% confidence intervals.

As mentioned above, the first plot produced by the `ex12.cqc` file contains a comparison of the item estimates for males and females, along with the identity line. The plot is shown in Figure 7. According to the plot, there seems to be some variation in item difficulties for these two groups of students. An item where difference is more noticeable and thus of particular interest would be item four (the one in the low right corner). Other items showing some degree of variability between the two groups are items three and six (the two on the left bottom corner).

The plot in Figure 10 allows us to determine whether the differences observed in the previous plot are statistically significant. In fact, items three, four and six are those where the Wald values fall considerable outside of the confidence interval, showing presence of DIF between the males and females. Wald values for items one and two are within the confidence interval, which indicates that although these items have different difficulty parameters for males and females, the difference is not statistically significant. Wald value of item five is just outside of the confidence interval; a close inspection of the item to investigate DIF is recommended.

SUMMARY

This tutorial shows how ConQuest matrix variables can be used to evaluate Differential Item Functioning — DIF — between two groups. Some key points covered in this tutorial are:

- the use of the `keepcases` command allows the estimation of item parameters separately for different groups.
- the use of the `matrixout` option in the `estimate` statement allows holding the results for each group in separate matrix variables.
- the use of operators and functions associated to the `compute` statement provide the opportunity to manipulate matrix variables created through the `estimate` command and compute new variables.
- the `scatter` statement allows the graphical comparison of the item parameters for different groups of students.

REFERENCES

Adams, R. & Wu, M. (2010). Unidimensional Latent Regression. Tutorial 5. ACER ConQuest, Notes and Tutorials. Retrieved from http://www.acer.edu.au/files/Conquest-Tutorial-5-UnidimensionalLatentRegression.pdf.

Engle, R.F. (1984). Wald, Likelihood Ratio, and Lagrange Multiplier Tests in Econometrics. In Intriligator, M.D. & Griliches, Z. *Handbook of Econometrics II*. Elsevier. pp. 775-826. Retrieved from http://www.stern.nyu.edu/rengle/LagrangeMultipliersHandbook_of_Econ__II___Engle.pdf.