SESUG Paper 180-2019

# Computing Classical Item Statistics in SAS® Using the Long vs. Wide Form of the Data

Imelda C. Go, Questar Assessment, Inc.

## ABSTRACT

The wide form of data is intuitive to people because it resembles data as they are presented in a spreadsheet with each column representing a variable. There are advantages to storing and processing data in their long form versus in their wide form. This paper illustrates one way to simplify and shorten the code needed to compute classical item statistics when the data are in their long form.

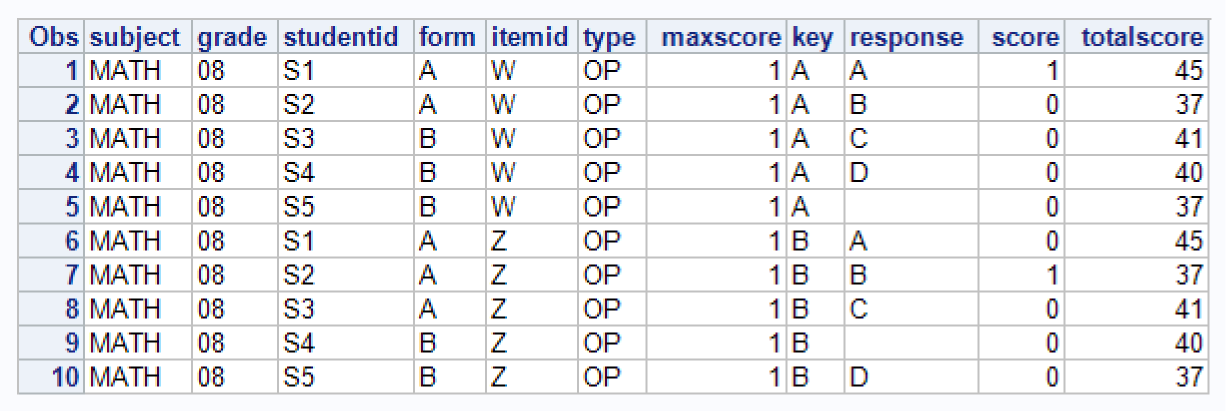
---

Since there are so many ways to do the same thing in SAS, this paper will use a simplified example to illustrate the technique with the following scenario:

- The test consists of single-select multiple choice items with four options (A, B, C, D). Each item has a maximum score of 1.
- There are different ways in which data can be presented in wide form. Let us suppose the operational items for an 8th grade mathematics test are in the following wide form. The example contains the following:
  - Student ID
  - Test-item-related information
    - Item ID
    - Student response to item (A, B, C, D, or blank in this example when there is no response from the student)
    - Item key (correct answer to the item)
    - Maximum number of points for the item when the student correctly answered the item
    - Total score (This is the criterion for calculating the point-biserial correlations.)

| Studentid | ItemID1 | Response1 | Key1 | MaxScore1 | Score1 | ItemID2 | Response2 | Key2 | MaxScore2 | Score2 | TotalScore |
|---|---|---|---|---|---|---|---|---|---|---|---|
| S1 | W | A | A | 1 | 1 | Z | A | B | 1 | 0 | 45 |
| S2 | W | B | A | 1 | 0 | Z | B | B | 1 | 1 | 37 |
| S3 | W | C | A | 1 | 0 | Z | C | B | 1 | 0 | 41 |
| S4 | W | D | A | 1 | 0 | Z |  | B | 1 | 0 | 40 |
| S5 | W |  | A | 1 | 0 | Z | D | B | 1 | 0 | 37 |

Let us suppose we also have the long form of the data as shown below:



| Obs | subject | grade | studentid | form | itemid | type | maxscore | key | response | score | totalscore |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | MATH | 08 | S1 | A | W | OP | 1 | A | A | 1 | 45 |
| 2 | MATH | 08 | S2 | A | W | OP | 1 | A | B | 0 | 37 |
| 3 | MATH | 08 | S3 | B | W | OP | 1 | A | C | 0 | 41 |
| 4 | MATH | 08 | S4 | B | W | OP | 1 | A | D | 0 | 40 |
| 5 | MATH | 08 | S5 | B | W | OP | 1 | A |  | 0 | 37 |
| 6 | MATH | 08 | S1 | A | Z | OP | 1 | B | A | 0 | 45 |
| 7 | MATH | 08 | S2 | A | Z | OP | 1 | B | B | 1 | 37 |
| 8 | MATH | 08 | S3 | A | Z | OP | 1 | B | C | 0 | 41 |
| 9 | MATH | 08 | S4 | B | Z | OP | 1 | B |  | 0 | 40 |
| 10 | MATH | 08 | S5 | B | Z | OP | 1 | B | D | 0 | 37 |

We will then add several indicator variables that will be useful for validating the results and computing the results. The additional indicator variables are defined as follows:

|  |  |
|---|---|
| nOmit | If the response is blank (i.e. there was no response), then the value is 1, otherwise it is a 0. |
| n1 | If the response is A then the value is 1, otherwise it is a 0. |
| n2 | If the response is B then the value is 1, otherwise it is a 0. |
| n3 | If the response is C then the value is 1, otherwise it is a 0. |
| n4 | If the response is D then the value is 1, otherwise it is a 0. |
| np | If the response is equal to the key, then the value is 1, otherwise it is a 0. |

The two following tables show how, by getting the mean and sum of the indicators, we can calculate the classical item statistics and five biserial correlations:

Table 1. Statistics from PROC MEANS

| Variable | MEAN of Variable | SUM of Variable |
|---|---|---|
| nOmit | % selected Omit | Number who did not respond |
| n1 | % selected A | Number who selected A |
| n2 | % selected B | Number who selected B |
| n3 | % selected C | Number who selected C |
| n4 | % selected D | Number who selected D |
| np | % selected correct answer (a.k.a p-value) | Number who answered correctly |
| score | average item score | |

Table 2. Statistics from PROC CORR

| Variable1 | Variable2 | CORRelation between Variable1 and Variable2 |
|---|---|---|
| np | Pbtotal | Point biserial correlation between the score and the criterion for students who answered the item correctly |
| n1 | pbtotal | Point-biserial correlation between the score and the criterion for students who chose response of A |
| n2 | pbtotal | Point-biserial correlation between the score and the criterion for students who chose response of B |
| n3 | Pbtotal | Point-biserial correlation between the score and the criterion for students who chose response of C |
| n4 | Pbtotal | Point-biserial correlation between the score and the criterion for students who chose response of D |

**SAS PROGRAMMING STATEMENTS** | **DESCRIPTION**

```
proc format;
invalue num ''=0 A=1 B=2 C=3 D=4;
```

This format statement allows us to map the response to a number useful for the ARRAY in the DATA step below.

```
data indicators;
set thedata;
if type='OP' then pbtotal=totalscore-score;
        else if type='FT' then pbtotal=totalscore;
```

We get the data and create the indicators described above.

The pbtotal is the criterion for the point biserial correlations. In this example, the operational score (totalscore) is the sum of the operational test items. For an operational test item, pbtotal is the total score minus the item score. For a field test item, it is the total score.

```
    array n{0:4} nomit n1-n4;
    do i=0 to 4;
        n{i}=0;
    end;
    nomit=0;
    np=0;
```

Set all the indicator variables to 0 first.

```
    if response='' then nomit=1;
        else if strip(response) ne '' then
            do;
               n{input(response,num.)}=1;
```

If there is no response (it is blank), then set nomit to 1.

Set the indicator variable that corresponds to the response to 1 (e.g., if the response is B, then nB=1).

```
               np=n{input(key,num.)};
            end;
    drop i;
```

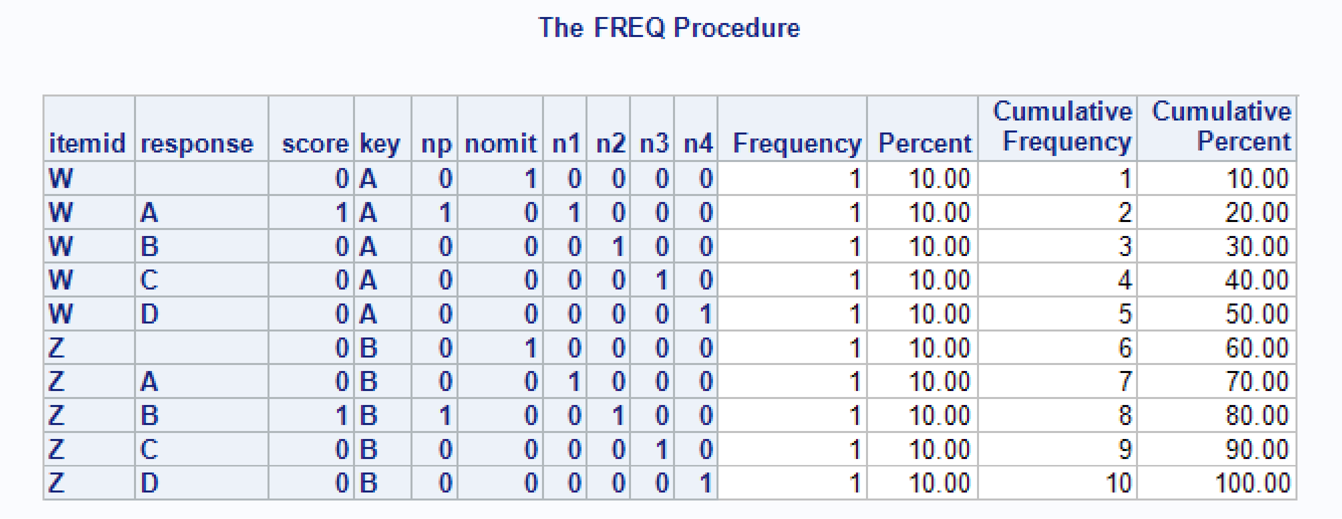
Set np to be equal to the indicator variable that corresponds to the key (e.g., if key is A, then indicator variable is nA). If they responded correctly, the value of nP is 1 and 0 otherwise.

```
proc freq data=processing; tables
itemid*response*score*key*np*nomit*n1*n2*n3*n4/list
missing;
```

This frequency table helps you make sure that the indicator variables were set up correctly.

| SAS PROGRAMMING STATEMENTS | DESCRIPTION |
|---|---|
| `proc means noprint data=processing ;`<br>`class subject grade type itemid form ;`<br>`var nomit n1-n4 np score;`<br>`output out=stats (rename=(_freq_=n)  where=(_type_ in`<br>`(30,31)) )`<br>`sum(nomit n1-n4)=x a b c d`<br>`mean=per_omit per_a per_b per_c per_d p_value item_mean;`<br>`run;` | This code produces the Table 1 statistics.<br>When the CLASS statement is used, we do not need to sort the data.<br>From this single PROC MEANS, we can compute the statistics for all the students and by test form. The CLASS statement and _type_ variable offer powerful tools for disaggregated statistics.<br>When _type_ = 30, the statistics are by *subject*, *grade*, *type*, and *itemid*. That is, we do not disaggregate by *form*. The *form* value is blank for this row of statistics in the output data set. In binary representation, this is 11110 for each of the variables in the order that they appear in the CLASS statement. The 1 is when we want to disaggregate by the variable, and the 0 is when we do not want to disaggregate by the variable. The decimal system equivalent of binary 11110 is 30.<br>When _type_ = 31, the statistics are by *subject*, *grade*, *type* *itemid*, and *form*. In binary, 11111 is equivalent to the decimal value of 31.<br>If you use the CHARTYPE option, PROC MEANS will create a character variable with the binary strings instead of the decimal values.<br>`proc means noprint data=processing chartype;`<br>The WHERE condition will become:<br>`where=(_type_ in ('11110','11111')` |
| `proc sort data=processing; by subject grade itemid form;`<br><br>`proc corr data=processing out=pbbyform`<br>`(where=(_type_='CORR')`<br>`rename= (np=pbis n1=pbis_a n2=pbis_b n3=pbis_c`<br>`n4=pbis_d) ) noprint;`<br>`by subject grade itemid form;`<br>`var np n1-n4;`<br>`with pbtotal;`<br><br>`proc corr data=processing out=pball (where=(_type_='CORR')`<br>`rename= (np=pbis n1=pbis_a n2=pbis_b n3=pbis_c`<br>`n4=pbis_d) ) noprint;`<br>`by subject grade itemid;`<br>`var np n1-n4;`<br>`with pbtotal;`<br>`run;` | This code produces the Table 2 statistics.<br>We must sort the data first because, we are going to use the BY statement with PROC CORR.<br>The first PROC CORR computes the statistics by test form.<br>The second PROC CORR computes the statistics across all test forms.<br>The output data set only keeps the correlations (_type_='CORR').<br>By using the WITH statement, we limit the correlations to those between the variables listed in the VAR statement and those listed in the WITH statement. |
| `data pb; set pbbyform pball; drop _type_;`<br>`run;` | The two point biserial data sets are now in one data set. |
| `proc sort data=pb (drop=_name_); by subject itemid form;`<br>`proc sort data=stats out=stats(drop=_type_); by subject`<br>`itemid form;`<br><br>`data combined; merge stats pb; by subject itemid form;`<br><br>`proc print data=combined;`<br>`format per_omit--pbis_d 4.2;`<br>`run;` | The Table 1 and Table 2 statistics are combined. |

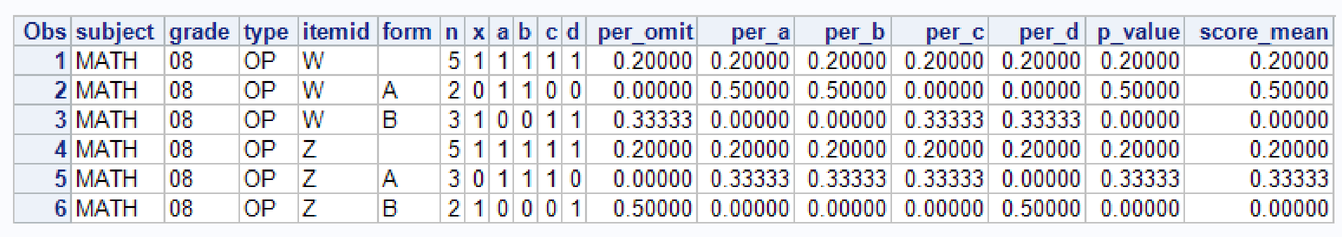
The following output from PROC FREQ, can help you determine if the indicator variables were created correctly. If yes, we can be confident that resulting sums, means, and correlations produce the classical item statistics.



### The FREQ Procedure

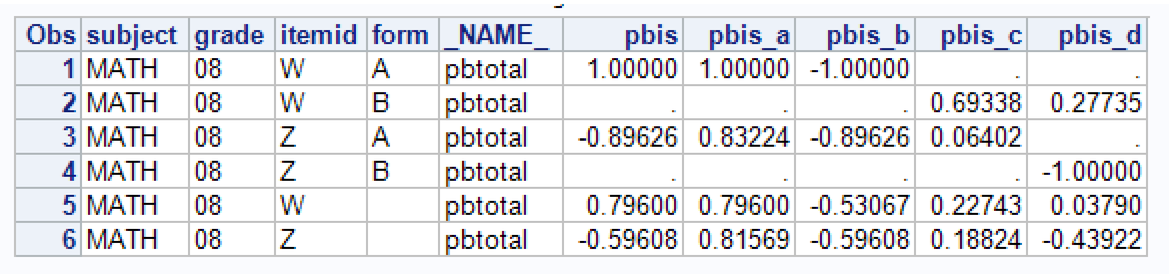
| itemid | response | score | key | np | nomit | n1 | n2 | n3 | n4 | Frequency | Percent | Cumulative Frequency | Cumulative Percent |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| W | | 0 | A | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 10.00 | 1 | 10.00 |
| W | A | 1 | A | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 10.00 | 2 | 20.00 |
| W | B | 0 | A | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 10.00 | 3 | 30.00 |
| W | C | 0 | A | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 10.00 | 4 | 40.00 |
| W | D | 0 | A | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 10.00 | 5 | 50.00 |
| Z | | 0 | B | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 10.00 | 6 | 60.00 |
| Z | A | 0 | B | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 10.00 | 7 | 70.00 |
| Z | B | 1 | B | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 10.00 | 8 | 80.00 |
| Z | C | 0 | B | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 10.00 | 9 | 90.00 |
| Z | D | 0 | B | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 10.00 | 10 | 100.00 |

These are the Table 1 statistics by test form in the data set from PROC MEANS: Our sample data had *form* values for each of the 10 records. When the *form* value is blank below, that row is for the statistics regardless of form. That is the record from PROC MEANS output when _type_=32.



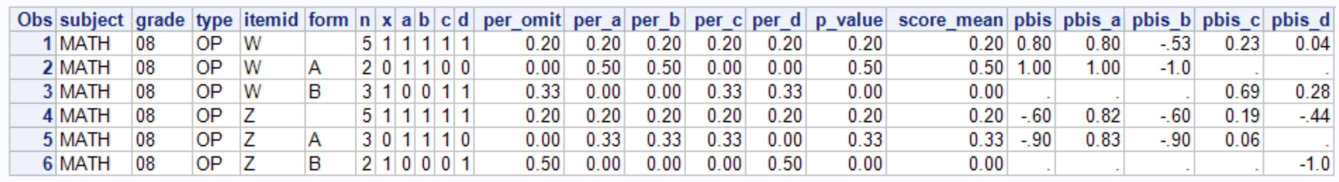
| Obs | subject | grade | type | itemid | form | n | x | a | b | c | d | per_omit | per_a | per_b | per_c | per_d | p_value | score_mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | MATH | 08 | OP | W | | 5 | 1 | 1 | 1 | 1 | 1 | 0.20000 | 0.20000 | 0.20000 | 0.20000 | 0.20000 | 0.20000 | 0.20000 |
| 2 | MATH | 08 | OP | W | A | 2 | 0 | 1 | 1 | 0 | 0 | 0.00000 | 0.50000 | 0.50000 | 0.00000 | 0.00000 | 0.50000 | 0.50000 |
| 3 | MATH | 08 | OP | W | B | 3 | 1 | 0 | 0 | 1 | 1 | 0.33333 | 0.00000 | 0.00000 | 0.33333 | 0.33333 | 0.00000 | 0.00000 |
| 4 | MATH | 08 | OP | Z | | 5 | 1 | 1 | 1 | 1 | 1 | 0.20000 | 0.20000 | 0.20000 | 0.20000 | 0.20000 | 0.20000 | 0.20000 |
| 5 | MATH | 08 | OP | Z | A | 3 | 0 | 1 | 1 | 1 | 0 | 0.00000 | 0.33333 | 0.33333 | 0.33333 | 0.00000 | 0.33333 | 0.33333 |
| 6 | MATH | 08 | OP | Z | B | 2 | 1 | 0 | 0 | 0 | 1 | 0.50000 | 0.00000 | 0.00000 | 0.00000 | 0.50000 | 0.00000 | 0.00000 |

These are the Table 2 statistics by test form in the data set from two PROC CORR runs: When the form value is A or B, that is from the first run of PROC CORR. When the form value is blank, that is from the second run of PROC CORR.



| Obs | subject | grade | itemid | form | _NAME_ | pbis | pbis_a | pbis_b | pbis_c | pbis_d |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | MATH | 08 | W | A | pbtotal | 1.00000 | 1.00000 | -1.00000 | . | . |
| 2 | MATH | 08 | W | B | pbtotal | . | . | . | 0.69338 | 0.27735 |
| 3 | MATH | 08 | Z | A | pbtotal | -0.89626 | 0.83224 | -0.89626 | 0.06402 | . |
| 4 | MATH | 08 | Z | B | pbtotal | . | . | . | . | -1.00000 |
| 5 | MATH | 08 | W | | pbtotal | 0.79600 | 0.79600 | -0.53067 | 0.22743 | 0.03790 |
| 6 | MATH | 08 | Z | | pbtotal | -0.59608 | 0.81569 | -0.59608 | 0.18824 | -0.43922 |

The Table 1 and 2 statistics are in the table below.

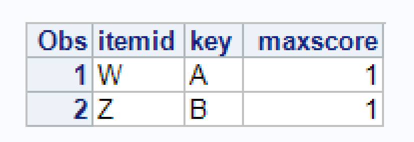


| Obs | subject | grade | type | itemid | form | n | x | a | b | c | d | per_omit | per_a | per_b | per_c | per_d | p_value | score_mean | pbis | pbis_a | pbis_b | pbis_c | pbis_d |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | MATH | 08 | OP | W | | 5 | 1 | 1 | 1 | 1 | 1 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.80 | 0.80 | -.53 | 0.23 | 0.04 |
| 2 | MATH | 08 | OP | W | A | 2 | 0 | 1 | 1 | 0 | 0 | 0.00 | 0.50 | 0.50 | 0.00 | 0.00 | 0.50 | 0.50 | 1.00 | 1.00 | -1.0 | . | . |
| 3 | MATH | 08 | OP | W | B | 3 | 1 | 0 | 0 | 1 | 1 | 0.33 | 0.00 | 0.00 | 0.33 | 0.33 | 0.00 | 0.00 | . | . | . | 0.69 | 0.28 |
| 4 | MATH | 08 | OP | Z | | 5 | 1 | 1 | 1 | 1 | 1 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | -.60 | 0.82 | -.60 | 0.19 | -.44 |
| 5 | MATH | 08 | OP | Z | A | 3 | 0 | 1 | 1 | 1 | 0 | 0.00 | 0.33 | 0.33 | 0.33 | 0.00 | 0.33 | 0.33 | -.90 | 0.83 | -.90 | 0.06 | . |
| 6 | MATH | 08 | OP | Z | B | 2 | 1 | 0 | 0 | 0 | 1 | 0.50 | 0.00 | 0.00 | 0.00 | 0.50 | 0.00 | 0.00 | . | . | . | . | -1.0 |

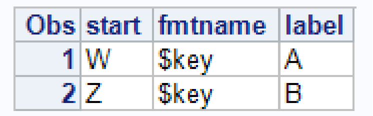
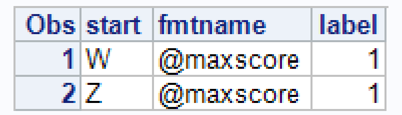
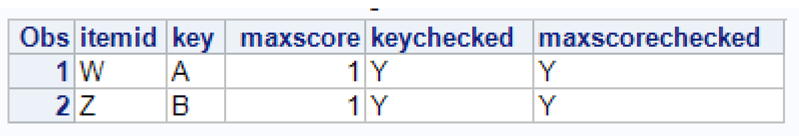
Disadvantages of using the long form of the data:

- The format is not that intuitive and the data may have to be transformed back to its wide form whenever laypeople need to use the data.
- Depending on the form of the original data, it might take some work to get the data into its long form.
- The data sets can get very large since the itemIDs and other item-related values (e.g., key and maxscore in this example) appear several times. If you are storing the long form in a permanent SAS data set, you can store it without the item metadata and just combine the item metadata together easily at the right time:
  - Use a hash object to bring the data you need into the processing.
  - You can make user-defined format and informats for the itemmetadata. Here is an example of using the CNTLIN= option with PROC FORMAT.

  Let's start with this itemmetadata data set with the itemid, key, and maxscore for the two items in the example.



| Obs | itemid | key | maxscore |
|---|---|---|---|
| 1 | W | A | 1 |
| 2 | Z | B | 1 |

<table>
<tr><th>SAS PROGRAMMING STATEMENTS</th><th>DESCRIPTION</th></tr>
<tr><td>proc sort data=itemmetadata; by itemid;<br><br>data key;<br>retain fmtname='$key';<br>set itemmetadata;<br>keep itemid fmtname label;<br>label=key;<br>rename itemid=start;</td><td><table><tr><th>Obs</th><th>start</th><th>fmtname</th><th>label</th></tr><tr><td>1</td><td>W</td><td>$key</td><td>A</td></tr><tr><td>2</td><td>Z</td><td>$key</td><td>B</td></tr></table>This is the data set created.</td></tr>
<tr><td>data maxscore;<br>retain fmtname='@maxscore';<br>set itemmetadata;<br>keep itemid fmtname label;<br>label=maxscore;<br>rename itemid=start;</td><td><table><tr><th>Obs</th><th>start</th><th>fmtname</th><th>label</th></tr><tr><td>1</td><td>W</td><td>@maxscore</td><td>1</td></tr><tr><td>2</td><td>Z</td><td>@maxscore</td><td>1</td></tr></table>This is the data set created. The fmtname value that starts with @ is for creating an INFORMAT.</td></tr>
<tr><td>proc format cntlin=key;<br>proc format cntlin=maxscore;</td><td>By using the two data sets above with the PROC FORMAT CNTLIN= option, we create the user-defined format and informat without any hard-coding.These two statements perform the equivalent of:<br>proc format;<br>value $key W=A Z=B;<br>invalue maxscore W=1 Z=1;</td></tr>
<tr><td>data test; set itemmetadata;<br>if key=put(itemid,$key.) then keychecked='Y';<br>if maxscore=input(itemid,maxscore.) then maxscorechecked='Y';<br>run;</td><td>This just illustrates how you can access the key and maxscore for each item at will and use it in your data processing.<br><table><tr><th>Obs</th><th>itemid</th><th>key</th><th>maxscore</th><th>keychecked</th><th>maxscorechecked</th></tr><tr><td>1</td><td>W</td><td>A</td><td>1</td><td>Y</td><td>Y</td></tr><tr><td>2</td><td>Z</td><td>B</td><td>1</td><td>Y</td><td>Y</td></tr></table></td></tr>
</table>

Advantages of using the long form of the data:

- There is the potential to shorten/simplify the coding.
  - If you were to program this using the wide form of the data, you will have many more variables to refer to in your program. These PROCs may have to be invoked many times, which can lengthen program run time. When the WIDE FORM is used, many programmers will use PROC MEANS and PROC FREQ. Once these get the frequency distribution from PROC FREQ, these may have to use PROC TRANSPOSE and/or other coding depending on the final form of the statistics you are after.
  - This technique can be easily modified to accommodate other types of test items.
- The indicator variables provide an easy way to check that the calculations will be done correctly. As long as we set up the indicator variables correctly, we are confident that PROC MEANS and PROC CORR are producing the correct values.
- There is the potential to shorten the program run time. It takes time for SAS to switch from PROC to PROC. In this example, we wanted the statistics by test form and regardless of test form.
  - These statistics were calculated using a single PROC MEANS statement. When we harness the power of using PROC MEANS, its CLASS statement, and _TYPE_ variables, we can produce disaggregated statistics easily.
  - For Table 2 statistics, PROC CORR was used twice.
- The code is short, simple, and easy to explain. It is highly reusable as long as the input data set in the long form has the same structure.

## CONCLUSION

There are a number of different ways to do anything in SAS. Calculating classical item analysis statistics using the long form of the data has advantages and disadvantages. For those seeking simplicity in coding and the potential to reduce the processing time, processing the long form of the data is quite attractive.

## CONTACT INFORMATION

Imelda C. Go, Ph.D.
igo@questarai.com
Working remotely from Columbia, SC

## TRADEMARK NOTICE