Psychometrics 101

Ronald P. Cody, Ed.D., Robert Wood Johnson Medical School, Piscataway, NJ

INTRODUCTION

SAS® software can be used to perform a variety of psychometric tasks, including test scoring, grade reporting, item analysis, and test reliability estimation. This short paper will help you get started in using SAS software to perform these tasks.

SCORING A MULTIPLE CHOICE TEST

Let's start out with the basics: how to score a multiple choice test, using SAS software. For this example, we will assume that the answer key and student responses are in a raw data file, with the answer key occupying the first line of data. This is a fairly typical data structure. For this example, as well as all the examples that follow, we will use a very small test file, consisting of 20 student responses to a ten item multiple choice test. You can see the actual file in the appendix of this paper.

Many schools use optical mark sense readers to transfer student test answers into machine readable form. We will assume that a student ID is located in columns 1-9 and that the test responses, consisting of the letters A through E, are in columns 11-20. To make the program more general, we use a macro variable to hold the number of test items, so that the program can be modified to score tests of any length, by changing only a single line of code.

The basic test scoring program follows:

```
*------------------------------------*
| PROGRAM NAME: SCORE.SAS            |
| PURPOSE: SCORE A MULTIPLE CHOICE   |
|         TEST                       |
| PROGRAMMER: RON CODY               |
| DATE: July 6, 1999                 |
*------------------------------------*;
%LET NITEMS = 10; ***NUMBER OF ITEMS; <1>

DATA SCORE;
  ARRAY KEY[&NITEMS] $; ***ANS KEY; <2>
  ARRAY ANS[&NITEMS] $; ***STUDENT ANS; <3>
  ARRAY S[&NITEMS];    ***ITEM SCORES; <4>
  RETAIN KEY1-KEY&NITEMS; <5>

  INFILE "C:\SASDATA\TESTDATA.TXT";

  ***READ THE ANSWER KEY;
  IF _N_ = 1 THEN DO; <6>
    INPUT @11 (KEY1-KEY&NITEMS)($1.);
    DELETE;
  END;

  ***READ IN STUDENT ANSWERS;
  INPUT @1   ID 9.
        @11 (ANS1-ANS&NITEMS)($1.); <7>
  FORMAT ID SSN11.;

  ***SCORE THE TEST;
  DO ITEM = 1 TO &NITEMS;
    S[ITEM] = KEY[ITEM] EQ ANS[ITEM]; <8>
  END;

  RAW = SUM(OF S[*]); <9>
  PERCENT = 100*RAW/&NITEMS; <10>

  LABEL ID      = "Social Security Number"
        RAW     = "Raw Score"
        PERCENT = "Percent Score";

  FORMAT ID SSN11.;
```

```
  DROP ITEM;
RUN;
```

EXPLANATION OF THE PROGRAM

To make this program somewhat general, we are using a macro variable (NITEMS) to represent the number of items on the test <1>. We define three arrays, KEY, ANS, and S to hold the answer key, the student answers, and the scored items (0 = incorrect, 1 = correct), respectively <2>,<3>,<4>. Since the answer key is the first line of data, we read in the answer key when _N_ is equal to one <6>. Also, since we do not want the answer key variable to be set to missing for subsequent iterations of the data step, they are retained <5>.

For this example, the student social security number is located in columns 1 to 9 and the student answers start in column 11, as indicated in the INPUT statement <7>.

Scoring the test is accomplished in a single line <8> using a somewhat unusual statement. The array element s[n] is set equal to the equality test between the answer key (KEY[n]) and the student answer (ANS[n]). If the student answer matches the answer key, S[n] is equal to one (true), otherwise, it is set equal to zero (false).

Finally, the raw score is the sum of the S[I]'s. Statement <9> Is equivalent to
```
RAW = SUM(OF S1-S&NITEMS);
```
The term S[*] is equivalent to all the elements in the array.

SOME SIMPLE REPORTS

Once the test is scored, you will probably want to print a student roster (in SS order), some summary statistics (such as the class mean), a histogram of test scores, and some item statistics. The following procedures will get this process started:

```
PROC SORT DATA=SCORE;
  BY ID;
RUN;

PROC PRINT DATA=SCORE LABEL;
  TITLE "Listing of Student Scores";
  ID ID;
  VAR RAW PERCENT;
RUN;

PROC MEANS DATA=SCORE MAXDEC=2 N MEAN STD
               MIN MAX;
  TITLE "Class Statistics";
  VAR RAW PERCENT;
RUN;

PROC FREQ DATA=SCORE;
  TITLE "Distribution of Student Answers";
  TABLES ANS1-ANS&NITEMS / NOCUM;
RUN;

PROC GCHART DATA=SCORE;
  TITLE "Histogram of Student Scores";
  VBAR PERCENT / MIDPOINTS=0 TO 100 BY 5;
RUN;
```

All of these procedures are straightforward and we do not need to spend any time on them. The output (edited) from all of these procedures is shown below:

Listing of Student Scores

Security Number	Social Raw Score	Percent Score
001-34-1212	9	90
002-13-2345	9	90
011-87-1234	5	50
022-54-9393	3	30
033-87-1323	6	60
101-34-5768	7	70
101-76-3848	2	20
102-36-4757	3	30
102-67-9876	2	20
109-87-6543	5	50
110-87-7676	6	60

. . .

Class Statistics

The MEANS Procedure

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
RAW	Raw Score	24	6.50	2.90	1.00	10.00
PERCENT	Percent Score	24	65.00	29.04	10.00	100.00

Distribution of Student Answers

The FREQ Procedure

ANS1	Frequency	Percent
A	15	62.50
B	3	12.50
C	1	4.17
D	3	12.50
E	2	8.33

ANS2	Frequency	Percent
A	2	8.33
B	15	62.50
C	1	4.17
D	2	8.33
E	4	16.67

ANS3	Frequency	Percent
A	2	8.33
B	3	12.50
C	16	66.67
D	1	4.17
E	2	8.33

. . .

Histogram of Student Scores
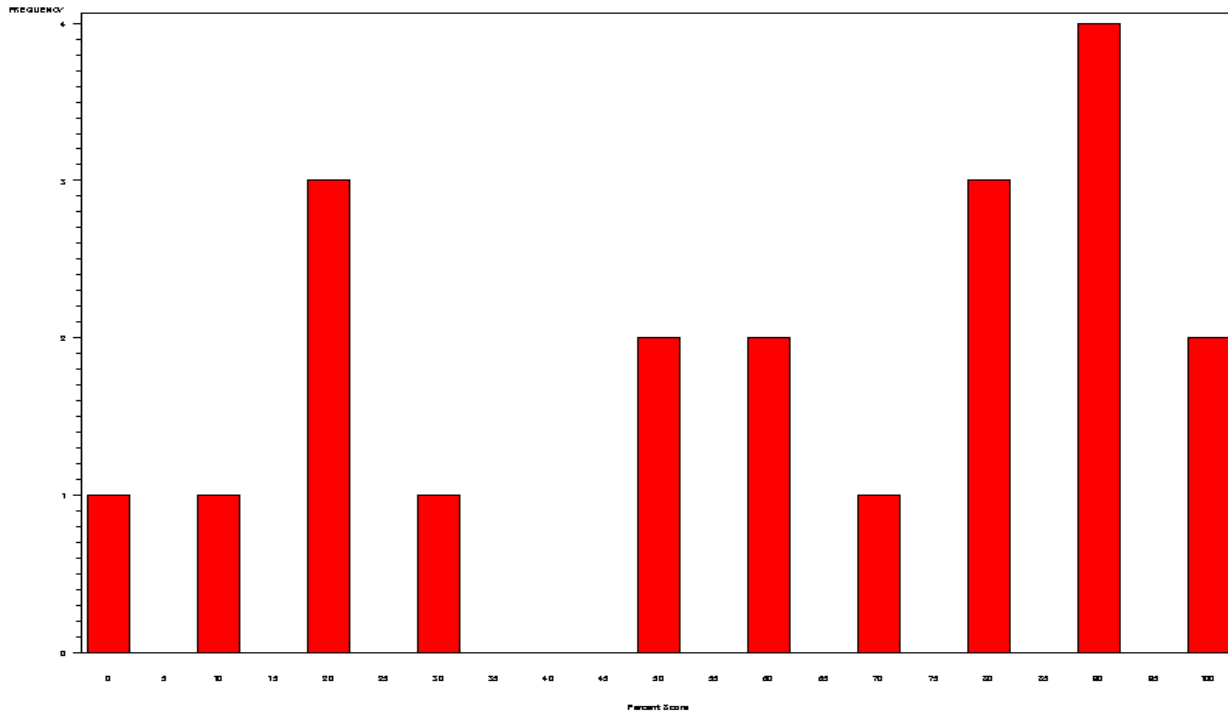
ITEM ANALYSIS

Besides looking at the number and percent of the students choosing each of the multiple choices to the 10 items, we can gain some more insight into the items if we know the mean test score of all of the students choosing each of the multiple choices for each item. For example, if the mean test score of the students answering an item correctly is lower than the mean test score for students choosing the wrong answers for that item, we might be suspicious that the answer key is incorrect or that there was some problem with that item. To create a nice looking table of answer frequencies along with the mean score of the students choosing each of the items, we can restructure the data set a bit and use PROC TABULATE to produce a very nice looking table. Here is the code:

```
PROC FORMAT;  <1>
  PICTURE PCT LOW - HIGH = '00000%';
RUN;

%LET NITEMS = 10;

DATA SCORE;
  INFILE 'C:\SASDATA\TESTDATA.TXT';
  ARRAY ANS[&NITEMS] $ 2 ANS1-ANS&NITEMS;  <2>
  ARRAY KEY[&NITEMS] $ 1 KEY1-KEY&NITEMS;
  ARRAY S[&NITEMS] 3 S1-S&NITEMS;

  RETAIN KEY1-KEY&NITEMS;

  IF _N_ = 1 THEN
     INPUT @11 (KEY1-KEY&NITEMS)($1.);

  INPUT @1  ID 1-9
        @11 (ANS1-ANS&NITEMS)($1.);

  DO I = 1 TO &NITEMS;
    IF KEY[I] EQ ANS[I] THEN DO;
      S[I] = 1;
```

```
      SUBSTR(ANS[I],2,1) = '*';  <3>
    END;
    ELSE S[I] = 0;
  END;
  RAW = SUM(OF S1-S&NITEMS);
  PERCENT = 100*RAW / &NITEMS;

  KEEP ANS1-ANS&NITEMS ID RAW PERCENT;

  LABEL ID      = "Social Security Number"
        RAW     = "Raw Score"
        PERCENT = "Percent Score";

  FORMAT ID SSN11.;
RUN;
DATA TEMP;  <4>
  SET SCORE;
  ARRAY ANS[*] $ 2 ANS1-ANS&NITEMS;
  DO Question = 1 TO &NITEMS;
    Choice = ANS[QUESTION];
    OUTPUT;
  END;
  KEEP QUESTION CHOICE PERCENT;
RUN;
PROC TABULATE DATA=TEMP;
  TITLE "Mean Score for Each Item Choice";
  TITLE2 "Note: * indicates correct answer";
  CLASS QUESTION CHOICE;
  VAR PERCENT;
  TABLE QUESTION*CHOICE,
        PERCENT=' '*(PCTN<CHOICE>*F=PCT.
                     N*F=3.
                     MEAN*F=PCT.) /
        RTS=20 MISSTEXT=' ';
  KEYLABEL ALL = 'Total'
           MEAN = 'Mean Score'
           PCTN = 'Freq.' ;
RUN;
```

The picture format <1>, places a percent sign at the end of the frequencies and mean score percentages. The scoring data step is similar to the previous scoring program except that the item response variables in the array ANS are given two bytes instead of just one <2>. The reason for this is that we will place an asterisk in the second byte of the item choice corresponding to the correct answer. Line <3> of the program accomplishes this task, using the SUBSTR function on the left side of the equal sign. Finally, the last data step <4> restructures the data set so that each observation contains a question number, the student choice, and the mean score of the students choosing that answer choice. The output (edited) from this program is shown next:

```
Mean Score for Each Item Choice
Note: * indicates correct answer
```

Question	Choice	Freq.	N	Mean Score
1	A*	62%	15	76%
	B	12%	3	43%
	C	4%	1	30%
	D	12%	3	53%
	E	8%	2	50%
2	A	8%	2	20%
	B*	62%	15	81%
	C	4%	1	80%
	D	8%	2	30%
	E	16%	4	40%
3	A	8%	2	20%
	B	12%	3	30%
	C*	66%	16	79%
	D	4%	1	30%
	E	8%	2	65%

```
    . . .
```

Look at item one for example. The correct answer was A (as indicated by the asterisk). 62% of the students chose A and the students who chose A averaged 76% on the test, higher than the average test score of students selecting any of the other answer choices.

MORE ITEM STATISTICS (POINT BISERIAL CORRELATIONS)

Classical item analysis typically includes a point-biserial correlation coefficient, also known as a discrimination index. This is actually no different than a Pearson correlation computed between each item (scored 0 or 1) and the test score for each student. Because one of the variables (the item score) is dichotomous the correlation coefficient is given this special name. If you were to compute this correlation by hand (ugh!), the Pearson formula simplifies somewhat because one of the variables can only take on values of 0 or 1. However, since we are using SAS software, we can use PROC CORR to compute the point-biserial correlations. In addition, we can use the ALPHA option of PROC CORR to compute these correlations as well as Cronbach's Alpha (which we will discuss next). For now, let's use PROC CORR to compute the correlations between each item and the total test score. We have:

```
PROC CORR DATA=SCORE NOSIMPLE;
  VAR S1-S&NITEMS;
  WITH RAW;
RUN;
```

Output from this procedure is shown next:

The CORR Procedure

1 With Variables: RAW
10   Variables: S1   S2   S3   S4   S5   S6   S7
S8   S9   S10

Pearson Correlation Coefficients, N = 24
Prob > |r| under H0: Rho=0

	S1	S2	S3	S4	S5
RAW	0.49948	0.74166	0.71504	0.85443	0.62873
Raw Score	0.0130	<.0001	<.0001	<.0001	0.0010

Pearson Correlation Coefficients, N = 24
Prob > |r| under H0: Rho=0

	S6	S7	S8	S9	S10
RAW	0.71139	0.71343	0.65398	0.23691	0.22704
Raw Score	<.0001	<.0001	0.0005	0.2650	0.2860

The point-biserial correlation for items 1, 2, and 3 are .49948, .74166, and .71504, respectively.   To be slightly more rigorous in this calculation, we should actually compute the correlation between each item and the total test score, computed with that item removed. The reason for this is that each item is naturally correlated with the total test score since each item makes up 1/nth of the total. The ALPHA option of PROC CORR does this calculation for us. Since the ALPHA option does not allow a WITH statement, we need to modify the procedure as follows:

```
PROC CORR DATA=SCORE NOSIMPLE ALPHA;
   VAR S1-S&NITEMS;
RUN;
```

Below, is a portion of the PROC CORR output generated by this code:

Cronbach Coefficient Alpha with Deleted Variable

Deleted Variable	Raw Variables		Standardized Variables	
	Correlation with Total	Alpha	Correlation with Total	Alpha
S1	0.355220	0.798950	0.355854	0.796723
S2	0.648451	0.764315	0.646843	0.762815
S3	0.617784	0.768546	0.618639	0.766242
S4	0.800750	0.747645	0.802747	0.743306
S5	0.516348	0.780672	0.515152	0.778556
S6	0.610059	0.769042	0.607562	0.767580
S7	0.610426	0.768729	0.609298	0.767371
S8	0.536206	0.777874	0.543102	0.775271
S9	0.086763	0.823679	0.082465	0.825763
S10	0.058183	0.830738	0.053993	0.828637

Notice that the correlation between each of the 10 items and the total are reduced somewhat. For example, the item 1 point-biserial coefficient was reduced from .49948 to .35522 and item 2 coefficient was reduced from .74166 to .648451. On longer tests, this difference would be smaller.

INTERPRETATION OF THE POINT-BISERIAL CORRELATIONS

What do these correlations tell us about the items? A positive point-biserial correlation indicates that students who scored well on the test were more likely to get the item right than students who did poorly. Suppose you were careless in writing a test question and some of the smarter students "read into" the question in a way that you did not intend, and got it wrong.   What would happen to the point-biserial coefficient? Since a group of the smarter students would get the item wrong the coefficient might be close to zero or even negative. What about a mistake in the answer key? When this happens, the result are usually dramatic. The weaker students will be more likely to get the item right (by guessing) while the smarter students will be more likely to get the item wrong (at least according to your wrong answer key).   The result is a negative point-biserial correlation coefficient. Checking the answer key should be the first step when any point-biserial coefficient is negative.

You should be aware that items that are very easy or very hard will tend to have coefficients near zero. Why? If, for example, every student answered an item correctly, the correlation between the item and the test score would be zero. The same goes for items where almost everyone gets it wrong. You may want to keep items with close to zero point-biserial coefficients that are on the easy side. These are not

necessarily bad items, just that they are not useful in discriminating between the good and poor students.

ITEM CHARACTERISTIC CURVES

Another way to see how well your items are performing, is to divide the class into quartiles (based on the total test score) and determine the proportion of each quartile that answered the item correctly. For example, suppose that 25% of the bottom quartile, 45% of the second quartile, 65% of the third quartile, and 95% of the top quartile answered an item correctly. This would be considered a good discriminating item. We can write some SAS code that will produce these percentages by quartile. The key will be to use PROC RANK (with the GROUPS= option) to create the quartiles. Then, we can use PROC MEANS to compute the proportion correct for each of the four quartiles (remember that PROC RANKS will number the quartiles 0, 1, 2, and 3). Finally, a simple PROC PRINT will display the proportions correct by quartiles. Here is the SAS code:

```
PROC RANK DATA=SCORE
          OUT=QUART(KEEP=RAW S1-S&NITEMS
                   QUARTILE)

          GROUPS=4;
   VAR RAW;
   RANKS QUARTILE;
RUN;
```

```
PROC MEANS DATA=QUART NWAY NOPRINT;
   CLASS QUARTILE;
   VAR S1-S&NITEMS;
   OUTPUT OUT  = Q_OUT
          MEAN = PROP1-PROP&NITEMS;
RUN;

PROC PRINT DATA=Q_OUT NOOBS;
   TITLE "ITEM CHARACTERISTIC CURVE";
   ID QUARTILE;
   VAR PROP1 - PROP&NITEMS;
   FORMAT PROP1-PROP&NITEMS PERCENT7.1;
RUN;
```

The GROUPS=4 option of PROC RANK creates a variable called QUARTILE which has values of 0 to 3, depending on which quartile the raw score lies in. Remember that the S-variables are the scored items (0 = incorrect, 1 = correct), so the mean of the S's by QUARTILE gives us exactly what we want. Finally, a PROC PRINT lists out the results (see below):

ITEM CHARACTERISTIC CURVE

QUARTILE	PROP1	PROP2	PROP3	PROP4	PROP5
0	16.7%	0.0%	16.7%	0.0%	16.7%
1	80.0%	80.0%	60.0%	80.0%	80.0%
2	80.0%	60.0%	80.0%	100%	100%
3	75.0%	100%	100%	100%	87.5%

QUARTILE	PROP6	PROP7	PROP8	PROP9	PROP10
0	16.7%	16.7%	16.7%	66.7%	50.0%
1	40.0%	20.0%	20.0%	60.0%	60.0%
2	100%	80.0%	80.0%	60.0%	60.0%
3	87.5%	100%	100%	100%	75.0%

Since we are using fake, made up data, many of the items are not performing too well. However, look at item 7 (PROP7). Sixteen percent of the lowest quartile got this item correct while 100% of the top quartile got it correct. If you have a large group taking your test, you may want to divide the class into more than four groups. For a very large class, six groups are frequently used.

COMBINING ALL THE ITEM STATISTICS INTO A SINGLE TABLE

It would be nice to have all the information about each item (the proportion of the class choosing A, B, C, etc., the item difficulty, the point-biserial correlation coefficient, and the proportion correct by quartile) in a single table. By using output data sets, doing come restructuring in a data step, and using PROC TABULATE, we can do just that. We present a complete item analysis program below, without much comment:

```
*----------------------------------------*
| Program Name: SCORE4.SAS  in C:\APPLIED |
| Purpose: To score a multiple-choice exam |
*----------------------------------------*;
%LET NITEMS = 10;
***The number of items on the test;
```

```
DATA SCORE; <1>
   INFILE 'C:\SASDATA\TESTDATA.TXT';
   ARRAY ANS[&NITEMS] $ 1 ANS1-ANS&NITEMS;
   ***Student answers;
   ARRAY KEY[&NITEMS] $ 1 KEY1-KEY&NITEMS;
   ***Answer key;
   ARRAY S[&NITEMS] S1-S&NITEMS;
   ***Score array 1=right,0=wrong;
   RETAIN KEY1-KEY&NITEMS;

   IF _N_ = 1 THEN
      INPUT @11 (KEY1-KEY&NITEMS)($1.);

   INPUT @1 ID 1-9
         @11 (ANS1-ANS&NITEMS)($1.);

   DO I = 1 TO &NITEMS;
      S[I] = KEY[I] EQ ANS[I];
   END;

   RAW = SUM (OF S1-S&NITEMS);
   PERCENT = 100*RAW / &NITEMS;
```

```
   KEEP ANS1-ANS&NITEMS S1-S&NITEMS
       KEY1-Key&NITEMS
       ID RAW PERCENT;
  ***Note: ANS1-ANSn, S1-Sn, KEY1-KEYn
               are needed later on;

  LABEL ID = 'Social Security Number'
        RAW = 'Raw Score'
        PERCENT  = 'Percent Score';
RUN;

***Section to prepare data sets for PROC
   TABULATE;

***Write correlation coefficients to a data set;
PROC CORR DATA=SCORE NOSIMPLE NOPRINT  <2>
  OUTP=CORROUT(WHERE = (_TYPE_='CORR'));
  VAR S1-S&NITEMS;
  WITH RAW;
RUN;

***Reshape the data set;
DATA CORR;  <3>
  SET CORROUT;
  ARRAY S[*] 3 S1-S&NITEMS;
  DO I=1 TO &NITEMS;
    CORR=S[I];
    OUTPUT;
  END;
  KEEP I CORR;
RUN;

***Compute quartiles;
PROC RANK DATA=SCORE GROUPS=4  <4>
          OUT=QUART(DROP=PERCENT ID);
  RANKS QUARTILE;
  VAR RAW;
RUN;

***Create ITEM variable and reshape again;
DATA TAB;  <5>
  SET QUART;
  LENGTH ITEM $ 5 QUARTILE CORRECT I 3
         CHOICE $ 1;
  ARRAY S[*] S1-S&NITEMS;
  ARRAY ANS[*] $ 1 ANS1-ANS&NITEMS;
  ARRAY KEY[*] $ 1 KEY1-KEY&NITEMS;
  QUARTILE=QUARTILE+1;

  DO I = 1 TO &NITEMS;
    ITEM = RIGHT(PUT(I,3.)) || " " ||
           KEY[I];
    CORRECT = S[I];
    CHOICE = ANS[I];
    OUTPUT;
  END;
```

```
  KEEP I ITEM QUARTILE CORRECT CHOICE;
RUN;

PROC SORT DATA=TAB;
  BY I;
RUN;

***Combine correlations and quartile
information;
DATA BOTH;  <6>
  MERGE CORR TAB;
  BY I;
RUN;

***Print out a pretty table;
OPTIONS LS=72;
PROC TABULATE FORMAT=7.2 DATA=BOTH
ORDER=INTERNAL
     NOSEPS;  <7>
  TITLE 'Item Statistics';

  LABEL QUARTILE = 'Quartile'
        CHOICE   = 'Choices';

  CLASS ITEM QUARTILE CHOICE;
  VAR CORRECT CORR;
  TABLE ITEM='#  Key'*F=6.,
    CHOICE*(PCTN<CHOICE>)*F=3.
    CORRECT=' '*MEAN='Diff.'*F=PERCENT5.2
    CORR=' '*MEAN='Corr.'*F=5.2
    CORRECT=' '*QUARTILE*MEAN='Prop.
      Correct'*F=PERCENT7.2 /
      RTS=8;
      KEYLABEL PCTN='%' ;
RUN;
```

A BRIEF EXPLANATION

The first Data Step <1> is the same as we have used
previously in this paper. Point-biserial correlation coefficients
are written to a data set (CORROUT) with the PROC CORR
statements <2>, so that they can be merged with other item
statistics later. The output data set from PROC CORR is
contained in a single observation. We need to restructure this
data set so that it contains an observation for each item for
each student. This is accomplished in the Data Step starting
on line <3>. PROC RANK <4>, is used to compute the class
quartiles. We then restructure the data set created by PROC
RANK so that it can be used with PROC TABULATE to create
the final table <5>. The correlations are then merged with the
other item statistics in a final Data Step <6>. Finally, we
write out monster PROC TABULATE code to produce a very
functional table (see below):

Item Statistics

# Key	Choices A %	B %	C %	D %	E %	Diff.	Corr.	Quartile 1 Prop. Correct	Quartile 2 Prop. Correct	Quartile 3 Prop. Correct	Quartile 4 Prop. Correct
1 A	63	13	4	13	8	63%	0.50	16.7%	80.0%	80.0%	75.0%
2 B	8	63	4	8	17	63%	0.74	0.00%	80.0%	60.0%	100%
3 C	8	13	67	4	8	67%	0.72	16.7%	60.0%	80.0%	100%
4 D	8	13	4	71	4	71%	0.85	0.00%	80.0%	100%	100%
5 E	13	4	4	8	71	71%	0.63	16.7%	80.0%	100%	87.5%
6 C	17	4	63	13	4	63%	0.71	16.7%	40.0%	100%	87.5%
7 D	13	17	8	58	4	58%	0.71	16.7%	20.0%	80.0%	100%
8 E	17	8	4	13	58	58%	0.65	16.7%	20.0%	80.0%	100%
9 B	17	75	.	4	4	75%	0.24	66.7%	60.0%	60.0%	100%
10 A	63	25	8	.	4	63%	0.23	50.0%	60.0%	60.0%	75.0%

OVERALL TEST STATISTICS

There isn't too much left to do. Besides the test statistics already produced earlier in this paper (the class mean and standard deviation and a histogram of test scores), we need a measure of overall test reliability. A test is reliable if a retest of the same test at a later time (usually about three weeks later) results in similar scores. That is, if a student received a 90% on the first administration, we would expect a high score on the retest. A measure of the reliability is the correlation between the pairs of scores obtain from the two administrations of the test. How do we estimate the test-retest correlation if we only give the test once?

In the "old days" before computers were in every office, test-retest reliability was computed by a statistics called the "split-half" correlation. What was done was to divide the test into two equal parts, one consisting of the odd numbered items and the other of the even numbered items. Scores on the odd and even numbered items were then correlated and the correlation was "corrected" by the Spearman Brown formula so that it appeared that you were correlating two tests, each with a length of the original test. The odd and even test scores were treated as if they were a test and a retest.

A more robust statistical procedure developed by Cronbach, is very similar to a split-half correlation. Cronbach's Coefficient Alpha is an estimate of test reliability and can be used even when the item scores are not dichotomous. For example, each item on a test might be scored on a 5-point Likert scale. When the items are dichotomous as in our earlier example, coefficient alpha can be computed by a simpler formula call the Kuder-Richardson Formula 20 (in a book by Kuder and Richardson, that formula was numbered 20 and the name "formula-20" stuck). Some time around SAS version 6.08, the ALPHA option was added to PROC CORR. We already saw that this could be used to generate item-test correlations. It also outputs Cronbach's Coefficient Alpha. Below, is another piece of the output from running PROC CORR with the ALPHA option:

Cronbach Coefficient Alpha

Variables	Alpha
Raw	0.802310
Standardized	0.800705

Coefficients above about .8 are considered quite good by most psychometricians.

INTERRATER RELIABILITY

Another task that falls under the psychometric umbrella, is comparing the reliability of two raters. Suppose you have x-rays on 10 subjects. Two radiologists read each of the x-rays and determine if a fracture is present. You would like a way of measuring how strong the agreement is between the two raters. You might be tempted to draw a 2 by 2 table and compute a McNemar's test of significance. However, you need to take into account that, by chance alone, the two raters could agree. As a matter of fact, if each rater flipped a coin to determine if a fracture was present, they would agree half the time! A statistic called Coefficient Kappa is a measure of interrater reliability that takes into account that some agreement between two raters on a dichotomous score results from chance. The KAPPA option on the TABLES statement of PROC FREQ will compute Kappa and a significance test for Kappa. To see how this works, look at the data from two radiologists examining 10 x-rays:

```
                  X-ray
            1  2  3  4  5  6  7  8  9  10
Rater 1     Y  Y  Y  N  N  Y  Y  N  N
Rater 2     Y  Y  N  Y  N  Y  Y  N  N   Y
```

To compute Kappa, we have:

```
DATA X_RAY;
  LENGTH RATER_1 RATER_2 $ 1;
  INPUT RATER_1 RATER_2 @@;
DATALINES;
Y Y  Y Y  Y N  N Y  N N  Y Y  Y Y  N N  N
N  N Y
;
PROC FREQ DATA=X_RAY;
  TABLE RATER_1 * RATER_2 / NOCUM KAPPA;
RUN;
```

Output follows:

The FREQ Procedure

Table of RATER_1 by RATER_2

RATER_1   RATER_2

Frequency,
Percent   ,
Row Pct ,
Col Pct ,N   ,Y   , Total
ƒƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
N   ,   3 ,   2 ,   5
        , 30.00 , 20.00 , 50.00
        , 60.00 , 40.00 ,
        , 75.00 , 33.33 ,
ƒƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Y   ,   1 ,   4 ,   5
        , 10.00 , 40.00 , 50.00
        , 20.00 , 80.00 ,
        , 25.00 , 66.67 ,
ƒƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆƒƒƒƒƒƒƒƒˆ
Total   4   6   10
               40.00   60.00 100.00

Statistics for Table of RATER_1 by RATER_2

McNemar's Test

Statistic (S)   0.3333
DF   1
Pr > S   0.5637

Simple Kappa Coefficient

Kappa   0.4000
ASE   0.2840
95% Lower Conf Bound   -0.1566
95% Upper Conf Bound   0.9566

Sample Size = 10

We see that Kappa is only .4 and that the p-value from McNemar's test is not even close to significant (p = .5637). In this example, the two raters agreed in 7 out of 10 cases. This seems pretty good until you remember that by chance alone, they would agree 5 times out of 10. So, 7 out of 10 is not much an improvement over chance.

CONCLUSION
You have seen some ways to use SAS software to conduct a very complete psychometric evaluation of a test. At the very least, you can use the scoring program at the beginning of this paper to score your exams (especially if your used a mark-sense reader and have machine readable test responses). By regularly analyzing your tests, you can determine if your answer key is correct, weed out bad items (and perhaps rescore the test), and produce more reliable tests, a very worthy goal.

APPENDIX

FILE TESTDATA.TXT
```
          ABCDECDEBA
101345768 ABCDECCDEA
102364757 CDCEACCABC
102679876 BABABABABA
240781234 AAAAAAAAAA
011871234 DEEDEDAEBA
110877676 ABCDEBBBBB
112098989 EBCDECDEBA
202458776 ABEDECAEBA
101763848 BEABEABEAB
459238765 ABCDECDEBA
541121111 AECDECDEBB
109876543 ABCDEABCDE
022549393 DDDBADDDBA
033871323 ABBCDCDABA
131445555 ABCDEDDEBA
121212121 BBCDECDEBA
220022222 ABCDECDDAA
112233444 ABCDECDEBB
451012121 ABCDECDEBA
118235746 EEBBCEEBBC
001341212 ABCDECDEBB
002132345 ABCDDCDEBA
456221234 ACCDECDEBB
564853456 DBCDECDEAA
```

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:

Ronald Cody
Robert Wood Johnson Medical School
Dept of Environmental and Community Medicine
675 Hoes Lane
Piscataway, NJ 08854
Work Phone: (732)235-4490
Fax: (732)235-4569
Email: CODY@UMDNJ.EDU