Proc Tabulate using a Categorical and Numerical Variable

Julie W. Pepe, University of Central Florida, Orlando

Abstract

Variables are usually categorical or numerical, yet in this example, a variable is both. Likert scale data from faculty evaluation surveys is recorded for each student in every class at our university. PROC Tabulate is used to produce a report that includes:

- class identification information
- enrollment data for each class
- percent of students responding per class
- percent of students responding in each category (1 to 5) based on a Likert scale
- calculation of means based on a 1 to 5 scale

The Likert scale values must be used as categorical and numeric in order to compute the required information. The response variable is duplicated (with a new name) ,then defined as categorical on the CLASS statement and as numeric on the VAR statement.

Introduction

Survey forms for evaluating faculty performance recently changed to accommodate statewide criteria. PROC Tabulate is used to produce a one-page printout that displays the information in a faculty friendly format. Students respond to each of 16 questions with a value of 1 - excellent, 2 - very good, 3 - average, 4 - good or 5 - poor. For each value 1 to 5, the number of responses and percent responding is computed. The percentage is calculated based on the number of responses to that question. In addition, a mean score for each question is computed using values (1 to 5).

To get the percent of students enrolled that responded to the survey, enrollment data is merged with response data. The percent responding is listed with the title information for each class.

Methods

The two data files used are: enrollment data (**enroll**) containing class name, section and student enrollment. The Enroll file setup is one input line per class. The other data file (**resp**) has a header record for each class followed by records containing student responses. The header records contain information on college, department, course, section and instructor. The header record separates classes. Each student record contains a value for each of 16 questions. There is one record for each student that filled out a survey.

resp.dat (partial listing)

```
Business Administration Accounting ADAMS
ACG2021 0001
24344543354.5554
2211122222111121
1211122122221222
1122222323222322

Business Administration Accounting BRIDELL
ACG2021 0002
4422135343533444
1111111111111111
1223111111113311
2211122111221221
44333433353345355
Business Administration Accounting TRAINOR
ACG2021 0003
3341243332322433
1133342233111311
2222122222332322
4533455435445555
```

Enroll.csv (partial listing)

```
ACG,2021,001,40
ACG,2021,002,41
ACG,2021,003,47
```

The program splits the raw data into header and response data sets. Variables in the header record are combined so the key information can be printed by specifying one variable name. Incorrect scanner values in the response data set are coded to missing. The response values for each question are assigned variable names**ans1** to **ans16**.

To calculate percent responding the data is sorted by classes. First.var and last.var are used to count the number of students responding per class. Percent responding is rounded to the nearest integer value and merged with all other information.

The data structure needs to be modified so each question's score has a variable identification (**Question**) and score value (**Score**). The value **Score** must be categorical in order to print summary output for each value 1 to 5. A categorical variable is defined using the class statement. The value also must be numerical so that means and standard deviations can be printed for each question. The variable statement is used to identify these variables. The variable **Score** is copied and called **Avg** so all required information can be requested in one Table statement.

Labels for score and question are added to enhance table appearance and readability. The scale is inverted, due to administrative preferences, so that larger values would indicate a better score. PROC Tabulate is then used to produce a one page synopsis of the class results. Formatting the variables to take up a minimum amount of space completed the program. An attached example of the program and output follows.

References

SAS Institute Inc. (1990), *SAS Language: Reference, Version 6, First Edition*., Cary NC:SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/STAT Users Guide Vol. 1 and 2, Version 6, Fourth Edition*., Cary NC:SAS Institute Inc.

The author may be contacted at:
University of Central Florida
Department of Statistics
Post Office Box 162370
Orlando, Florida 32816-2370
or pepe@pegasus.cc.ucf.edu

```
options ls= 90 ps=90 nodate nonumber;
filename sample 'a:RESP.dat';

data header (keep=dept college inst course classes lineinfo keyinfo)
        response (keep=classes ans1-ans16 keyinfo);
        retain course sect classes keyinfo;

INFILE sample;           /*******************************/
length type $ 1;         /* Identify header variables */
length dept $ 34;        /*******************************/
length college $ 23;
length inst $ 15;
length course $ 8;        /*****************************/
length sect $ 4;          /*Identify response variables*/
length classes $ 12;      /*****************************/
length lineinfo $ 90;
length keyinfo $ 27;

INPUT
  @1 type $ @;
select;
  when (type in (’A’,’B’,’C’,’E’,’H’,’I’,’J’,’M’))
        do;
          input @1 college $ 1-23 dept $ 25-34 inst $ course $ sect $;
          classes = course||sect;
          lineinfo = ’CLASS:’||classes||’ ’||’INST:’||inst;
          keyinfo = classes||inst;
      output header;
    end;
  otherwise
        do;
          input @1 ans1 1. ans2 1. ans3 1. ans4 1. ans5 1. ans6 1. ans7 1.
                           ans8 1. ans9 1. ans10 1. ans11 1. ans12 1. ans13 1.
                           ans14 1. ans15 1. ans16 1.;
      if ans1 > 5 then ans1 = .;
      if ans2 > 5 then ans2 = .;
      if ans3 > 5 then ans3 = .;
      if ans4 > 5 then ans4 = .;
      if ans5 > 5 then ans5 = .;
      if ans6 > 5 then ans6 = .;
      if ans7 > 5 then ans7 = .;
      if ans8 > 5 then ans8 = .;
      if ans9 > 5 then ans9 = .;
      if ans10 > 5 then ans10 = .;
      if ans11 > 5 then ans11 = .;
      if ans12 > 5 then ans12 = .;
      if ans13 > 5 then ans13 = .;
      if ans14 > 5 then ans14 = .;
      if ans15 > 5 then ans15 = .;
      if ans16 > 5 then ans16 = .;
          output response;
    end;
end;

data enroll;
infile ’a:ENROLL.csv’delimiter=’,’;
length prefix $ 3;
length number $ 4;        /***********************/
length section $ 4;       /* Identify enrollment */
length course $ 8;        /* variables           */
length classes $ 12;      /************************/
length enrolled $ 4;
input prefix $ number $ section $ roll;
enrolled = roll;
sect = ’0’||section;
course = prefix||number;
classes = course||sect;
drop prefix number section course;
proc sort data=header;  by classes;
proc sort data=enroll;  by classes;

data newhead;
merge header enroll;
by classes;
proc sort data=newhead;  by keyinfo;
proc sort data=response; by keyinfo;

Data classp;                       /******************************/
merge newhead response;                  /* Produce percent responding */
by  keyinfo;                       /* for each class             */
length  pctresp $ 3;                 /******************************/
if first.keyinfo then do;
n = 0;
end;
n + 1;
if last.keyinfo then do;
maxn=n;
pctresp = round(100*(maxn /roll),1);
output;
end;
keep dept keyinfo roll enrolledmaxn pctresp;
proc sort data=classp;  by keyinfo;
proc sort data=newhead; by keyinfo;

Data allclass;
merge newhead classp;
by keyinfo;
proc sort data=allclass;  bykeyinfo;

Data combined;
merge allclass response;
by keyinfo;
length compinfo $ 132;
compinfo= lineinfo||’  ENROLLED: ’||enrolled||’  Percent Responding: ’||pctresp;
proc sort data=combined;
by college dept keyinfo lineinfo;
```

```
data last;
set combined;
score=ans1;   question=1;   output;
score=ans2;   question=2;   output;
score=ans3;   question=3;   output;
score=ans4;   question=4;   output;
score=ans5;   question=5;   output;
score=ans6;   question=6;   output;
score=ans7;   question=7;   output;
score=ans8;   question=8;   output;
score=ans9;   question=9;   output;
score=ans10;   question=10;  output;
score=ans11;   question=11;  output;
score=ans12;   question=12;  output;
score=ans13;   question=13;  output;
score=ans14;   question=14;  output;
score=ans15;   question=15;  output;
score=ans16;   question=16;  output;
keep score question keyinfo college dept inst lineinfo compinfo ;

data convert;                  /***********************/
set last;                   /*Scores are flipped  so*/
conscore = score*6;                /* a 5 is excellent    */
if conscore = 6 then score = 5;       /***********************/
else if conscore = 12 then score = 4;
else if conscore = 18 then score = 3;
else if conscore = 24 then score = 2;
else if conscore = 30 then score = 1;
drop conscore;
avg = score;

proc format;
  value respfmt 5='Excellent'
                  4='Very Good'
                  3='Good'
                  2='Fair'
                  1='Poor';
  value questfmt 1='Feedback to student for performance in course'
                     2='Instructor interested in your learning'
                     3='Use of class time'
                     4='Instructor overall organization of course'
                     5='Continuity between lectures'
                     6='Pace of course'
                     7='Instructor assessment of your progress in the class'
                     8='Text and related materials used'
                     9='Descr. of course objectives and assignments'
```

```
                           10=’Communication of ideas and information’
                           11=’Expression of expectations for performance’
                           12=’Available to assist students outside class’
                           13=’Respect and concern for students’
                           14=’Stimulation of interest’
                           15=’Facilitation of learning’
                           16=’Overall assessment of instructor’;
run;
/*-------Class Report-----------------------------------------------------*/
proc tabulate data=convert format=4.2;
class compinfo question score;
var avg;
table  compinfo=’ ’,question=’ITEM’,
             score=’ ’*(n*F=3.0 pctn<score all>= ’% ’*F=4.1)
              ( avg=’SUMMARY’*(mean std n*F=3.0 ))
         /  misstext = ’0’printmiss condense rts=25;
 format  score respfmt.  question questfmt.  ;
Title1 ’Student Perception of Instruction’;
Title2 ’Fall 1996 Semester: Individual Course Report’;
run;
```