

Paper PO-15

Practical Approaches to Counting in SAS®: How to Get Started When You Don't Know Where to Begin

Sharon Avrunin-Becker, Westat, Rockville, MD

ABSTRACT

One of the wonderful things about SAS is that there are several ways to approach a programming problem with no one way being right or wrong. Some ways take more time to keyboard out all the steps and some ways take up less lines of programming but take up more of your brain energy trying not to type a few extra lines of code. One of the things SAS is frequently used for is to create reports which have calculations of numbers. The biggest question is how do I get my number totals. This paper will show you how the same answer can be derived in totally different ways.

INTRODUCTION

The teachers at the local preschool have asked their students what they want to be when they grow up. The teachers in the 4's class want to see the breakdown of careers in their class to see what extra topics of interest they could add to their curriculum throughout the year. They are especially interested in knowing not only the overall totals but also the totals broken down by gender so they can find the appropriate materials to reinforce their students' interests. Unfortunately, the teachers have no idea how to best approach counting the data. Let's take a look at a couple different approaches and see which best gets us to where we want to be.

The Data in Question...

```
DATA kids;
  LENGTH StudentID $6 Gender $3 Career $15 Age 8.;
  INPUT StudentID Gender Career Age;
  CARDS;
04-001      F      SASProgrammer      3
04-002      F      Doctor              4
04-003      M      Chef                4
04-004      M      Police              5
04-005      M      SASProgrammer      4
04-006      F      Police              4
04-007      M      SASProgrammer      4
04-008      M      Chef                4
04-009      M      Firefighter         5
04-010      M      Doctor              3
04-011      F      Chef                5
04-012      F      SASProgrammer      4
04-013      F      SASProgrammer      4
04-014      M      Firefighter         4
04-015      F      Doctor              3
```

```
RUN;
```

USING PROC FREQ

Probably the easiest solution would be to throw together a quick PROC FREQ to see what the children are interested in. The problem with using just a PROC FREQ is that you are then limited if you want to use these results for other purposes. The nice advantage is that it gives you quick results without much code and even the missing zero is filled in.

HERE IS SOME BASIC PROC FREQ CODE...

```
PROC FREQ DATA=kids ;
  TABLES Career*Gender/ nopercnt nocol norow;
RUN;
```

HERE IS THE OUTPUT...

Table of Career by Gender			
Career	Gender		
	F	M	Total
Chef	1	2	3
Doctor	2	1	3
Firefighter	0	2	2
Police	1	1	2
SASProgrammer	3	2	5
Total	7	8	15

USING PROC TABULATE

Another easy solution would be to throw together a quick PROC TABULATE to see what the children are interested in. Unfortunately, when you read the books there are so many possibilities and ways to use the TABULATE procedure that it can paralyze you in fear, so that you don't even know how to begin. It also requires a bit of additional code to fill in the missing with a zero.

HERE IS SOME QUICK PROC TABULATE CODE...

```
PROC TABULATE DATA=kids ;
  CLASS Career Gender;
  TABLE Career (all), Gender (all);
run;
```

HERE IS THE OUTPUT...

	Gender		
	F	M	
	N	N	N
Career			
Chef	1	2	3
Doctor	2	1	3
Firefighter	0	2	2
Police	1	1	2
SASProgrammer	3	2	5
All	7	8	15

USING PROC SQL WITH DATA STEPS

Sometimes programmers have learned SAS after years of programming in SQL. Their comfort level with SQL leads them to first think of a solution that primarily relies on PROC SQL. They may then take "baby steps" with DATA step programming at the end. Even though this solution requires a significant amount of code, it then enables you to use the results for other purposes, like combining several class totals into one overall report.

THE PROC SQL CODE...

Start with the SQL code...

```
PROC SQL;
  CREATE TABLE gencnt AS
  SELECT Career, Gender, count(*) AS numgender
  FROM kids
  GROUP BY Career, Gender ;
```

```

CREATE TABLE gencnt99 AS
SELECT 'Total' AS Career, Gender, count(*) AS numgender
FROM kids
GROUP BY Gender;

CREATE TABLE gencnttot AS
SELECT Career, count(*) AS total
FROM kids
GROUP BY Career ;

CREATE TABLE gencnt99tot AS
SELECT 'Total' AS Career, count(*) AS total
FROM kids ;

QUIT;

```

then use several data steps including a PROC TRANSPOSE to put all the results together in the order you want them displayed...

```

DATA allgen;
  SET gencnt gencnt99 ;
RUN;

DATA alltots;
  SET gencnttot gencnt99tot;
RUN;

PROC SORT DATA=allgen OUT=sallgen; BY Career; RUN;
PROC SORT DATA=alltots OUT=salltots; BY Career; RUN;

PROC TRANSPOSE DATA=sallgen OUT=allgentrns ;
  BY Career;
  ID Gender;
RUN;

DATA allgentots;
  MERGE allgentrns (DROP=_NAME_)
        salltots;
  BY Career;
  LABEL F = 'Female'
        M = 'Male'
        ;
RUN;

PROC PRINT DATA=allgentots NOOBS LABEL;
RUN;

```

HERE IS THE OUTPUT...

Career	Female	Male	Total
Chef	1	2	3
Doctor	2	1	3
Firefighter	.	2	2
Police	1	1	2
SASProgrammer	3	2	5
Total	7	8	15

USING PROC MEANS WITH DATA STEPS

Sometimes SAS programmers have more of a comfort level with PROC MEANS, since it is less complex than PROC TABULATE, and that leads them to a solution that fits in more with their strengths. By sending the data to an output file, it gives us the same advantage as the PROC SQL step to be able to manipulate the data and use it for several different purposes.

THE PROC MEANS CODE...

Start with the MEANS procedure code...

```
PROC MEANS DATA=kids NOPRINT;
  CLASS Gender Career;
  VAR age;
  OUTPUT OUT=mkids n= ;
RUN;
```

then use several data steps to put all the results together in the order you want them displayed...

```
DATA fixit_FC fixit_MC fixit_FT fixit_MT fixit_TC fixit_TT;
  SET mkids (DROP=_TYPE_ age);
  IF Gender = 'F' and Career ne '' THEN OUTPUT fixit_FC;
  ELSE IF Gender = 'M' and Career ne '' THEN OUTPUT fixit_MC;
  ELSE IF Gender = 'F' and Career = '' THEN DO;
    Career = 'Total';
    OUTPUT fixit_FT;
  end;
  ELSE IF Gender = 'M' and Career = '' THEN DO;
    Career = 'Total';
    OUTPUT fixit_MT;
  end;
  ELSE IF Gender = '' and Career ne '' THEN OUTPUT fixit_TC;
  ELSE IF Gender = '' and Career = '' THEN DO;
    Career = 'Total';
    OUTPUT fixit_TT;
  END;
RUN;

DATA allF (drop=Gender rename=(_FREQ_=Female));
  SET fixit_FC
    fixit_FT ;
RUN;

DATA allM (drop=Gender rename=(_FREQ_=Male));
  SET fixit_MC
    fixit_MT ;
RUN;

DATA allT (drop=Gender rename=(_FREQ_=Total));
  SET fixit_TC
    fixit_TT ;
RUN;

DATA allofit;
  MERGE allF
        allM
        allT
    ;
  BY Career;
RUN;

PROC PRINT DATA=allgentots NOOBS LABEL;
RUN;
```

HERE IS THE OUTPUT...

Career	Female	Male	Total
Chef	1	2	3
Doctor	2	1	3
Firefighter	.	2	2
Police	1	1	2
SASProgrammer	3	2	5
Total	7	8	15

CONCLUSION

As you can see there is no right or wrong way to count using SAS. The best approach is the approach that you feel confident using, knowing that the answers are going to be correct and your reports will be accurate. Once you have the confidence to start counting you can use these steps to add missing categories, combine totals from several datasets into one report, and to add highlighting to emphasis important numbers on your reports.

ACKNOWLEDGMENTS

Many thanks to Mike Rhoads, Michael Raithel, Rick Mitchell, Marie Alexander, and Sarah Woodruff

CONTACT INFORMATION

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

Sharon Avrunin-Becker
Westat
1650 Research Blvd. – WB432
Rockville, MD 20850
Work Phone: (301) 738-8384
Email: SharonAvruninBecker@westat.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.