

Moving from data to information: SAS programming to reduce statistical and programming errors

AnnMaria De Mars, University of Southern California, Los Angeles, CA

ABSTRACT

Although improvements in statistical software capabilities have made it easier to obtain output for complex statistical procedures, producing error-free output has not become easier. Without proper programming, numbers are just data, susceptible to errors in analysis and interpretation. SAS can be used in many ways to transform data into information. In the process, programmers and statisticians tend to make different types of errors. Four ways are discussed for producing error-free information.

1. The sheer volume of numbers can be reduced to a manageable amount via the use of SAS procedures such as PROC FREQ, PROC SUMMARY and PROC TRANSPOSE, producing output usable to quickly identify and eliminate variables and categories with no variance or excessive percentages of missing data.
2. Arrays, DIM function and Do- loops can be used to reduce programming errors while performing repetitive tasks and correct data errors.
3. Selection of appropriate statistical procedures based on knowledge of sample, weights and stratification will prevent errors in estimation of statistics.
4. Descriptive procedures, Graph-N-Go, Enterprise Guide and the Analyst Application are all useful for gaining an understanding of the data structure, distribution and accuracy.

Additional features are described for further expanding the statistician/ programmer's tools for reducing large quantities of data to useful conclusions.

INTRODUCTION

Over the past twenty-five years improvements in statistical software capabilities have made it easier to obtain output for complex statistical procedures. Unfortunately, this does not mean that obtaining error-free output has become easier. Complicating the problem further, statisticians and SAS programmers bring different areas of expertise to data analysis and often encounter different types of problems or errors in statistical analysis using SAS.

The first part of this paper is aimed at everyone who uses large-scale databases and needs to reduce an enormous amount of sheer numbers to a comprehensible set.

The remainder of the paper is divided into two parts, based on the recognition that people knowledgeable about SAS (but not statistics) tend to make one kind of error, errors in interpretation. People who are knowledgeable about statistics but not SAS programmers tend to make another kind of error, the kind that shows up in a SAS log with the word ERROR in capital letters.

Errors in interpretation can often be avoided by using proc summary, proc univariate graph-n-go and other SAS application to gain an understanding of one's data. Using SAS survey procedures to correctly applying sample weights will reduce another major source of potential error.

For statisticians who are not programmers, simple programming 'tricks' such as the use of %include statements can greatly improve readability of the code. Other common tools in the programmer's arsenal, including a format library, arrays, DIM function, character functions, macros and global macro variables can reduce the frequency of those annoying ERROR messages.

DROWNING IN NUMBERS: How to get your head above water

A distinction needs to be made between numbers and information. Many codebooks for large databases provide so many numbers that it is, paradoxically, difficult to obtain information. For example, it is rather difficult for anyone to mentally process 400 pages of frequency distributions. It's a good idea to begin by getting an understanding of the data structure and distribution, but there has to be a simpler, less time-consuming way to get a handle on large-scale databases.

One reason for drowning in numbers is inefficient programming techniques.

For example, is it really necessary to have a print out of every single subject ID number for 13,000 subjects? The ostensible rationale is that it will allow me to check if there are any duplicate subject numbers. Wolf (2005) gives a useful tip for checking categories with low percentages.

```
Proc freq data = mydata ;  
tables subject / out = check (where = ( count > 1 ) ;
```

Turning Data into Information

These two statements will save reading through 13,000 plus subject IDs looking for a count of 2 or higher.

A second problem is simply having too many variables. Beginning with a large dataset with hundreds of variables, the challenge is to reduce this to a manageable set that relate to the question to be answered through any analyses. To move from data to information, decisions must be two-dimensional, made on both statistical and conceptual bases. Although users are usually fully aware that a study will not require an entire dataset, the task of sorting through hundreds of pages of descriptive statistics, and reading reams of documentation seems daunting. Reducing a dataset from several hundred variables will always require some element of time spent in deliberation and variable selection. This bottleneck can be reduced with some strategic programming.

As shown below, a summary procedure can be used to create a dataset of selected statistics. The example used is the Early Years of Marriage dataset, which has 595 variables. The summary dataset is then transposed to create a new dataset which has 595 records and six variables, `_name_` (the name of the former variable), `label`, `mean`, `min`, `n` and `std`.

```
proc summary data = in.eym mean min n std ;
  output out = eym_stats ;
  var caseid -- v1338 ;
proc transpose data = eym_stats out = eym_trans ;
  id _STAT_ ;
```

A new dataset is then created which only includes variables that meet criteria for further examination on statistical grounds. Specific criteria will vary based on requirements of a particular study. In the example below, variables were selected if more than 5% of the data were missing, that had negative values out of range, i.e., less than zero, or which all 746 respondents had given the same response (`std = 0`).

```
data in.eymchk1 ;
  set eym_trans ;
  pctmiss = 1 - (n/746) ;
  if min < 0 then neg_min = 1 ;
  else neg_min = 0 ;
  if std = 0 then constant = 1 ;
  else constant = 0 ;
  if (pctmiss > .05 or neg_min = 1 or constant = 1) then output ;
```

<code>_LABEL_</code>	<code>N</code>	<code>MIN</code>	<code>MAX</code>	<code>MEAN</code>	<code>STD</code>	<code>pctmiss</code>	<code>neg_min</code>	<code>constant</code>
SEX OF HU PERSON #14	0	1.00000	1	0
AGE OF HU PERSON #14	0	1.00000	1	0
HW LNG DECDE MARRY:WEEKS	8	2	71	30.00	24.41	0.98928	0	0
HW LNG DECDE MARRY:MNTHS	364	1	54	12.99	7.81	0.51206	0	0
REASONS MARRIED:2ND MENT	642	11	90	26.58	16.34	0.13941	0	0
REASONS MARRIED:3RD MENT	299	11	90	29.29	19.59	0.59920	0	0

Doing the summary, transpose and then outputting the questionable variables to a dataset allows for quickly eliminating those of the 500+ variables that fail on statistical grounds. Using this method reduced the initial set of variables from 595 to 369 with less than two hours of actual human inspection time required. Many of the variables with more than 20% of the data missing were examined and quickly discarded as irrelevant to the question of interest. If there happened to be 14 people in the household, were the age and relationship to the respondent of the 14th person relevant to the study question? Variables that had no variance were eliminated on purely statistical grounds. Since it has no variance it is statistically an impossibility to use it to explain variance in any other variable.

Why not just automatically delete variables?

If 80% or more of the respondents did not answer a question, it needs to be asked how representative the remaining 15-20% are. Why not just automatically delete all of those variables, or substitute zeroes for the missing data? In some cases, there may be a legitimate reason for the number of missing observations. For example, the Early Years of Marriage dataset includes questions on, e.g., year of birth of the oldest child, a question that would be blank for people without children. Replacing a missing value with zero is clearly incorrect, and, while the data are legitimately missing and not due to bias, these may still be useful, e.g. in categorizing respondents as without children, children

Turning Data into Information

under age three, etc. While using the Proc Freq, Proc Summary, Proc Transpose and data steps above can significantly reduce the requirement for time spent in deciding on individual variables, it can never be eliminated entirely.

PREVENTING STATISTICAL ERRORS: I. USING CORRECT DATA

Reducing the number of variables cuts down on the “noise” in the dataset, allowing for selection of the correct variables. Returning to the output from the summary and transpose steps above, it may be possible to identify variables with data out of range, e.g., with values of less than zero for number of children. Often the researcher wants to keep these variables in the dataset but correct the errors.

As with many surveys, the Early Years of Marriage coding is “too helpful”. Unusable data, rather than being coded as missing, is broken down into multiple categories, including “refused to say”, “not applicable” and so on and numbered these -1, -2. So... you can perform the above calculations perfectly and end up with results that show the average American has -1.3 children. Results showing negative children per family are obviously incorrect. The real danger arises when 5- 10% of the data has those negative numbers. The results obtained *look* pretty much correct but is actually wrong.

The following code uses a few simple tricks to recode all negative values for all numeric variables.

```
Data mylib.eym_fix;
Set mylib.eym ;
Array rec {*} _numeric_ ;
Do I = 1 to dim(rec) ;
    if rec{i} < 0 then rec{i} = . ;
End ;
```

This little snippet illustrates several extremely useful SAS features.

1. Rather than going through the dataset and counting how many variables are numeric, the * for the array dimension instructs SAS to set the dimension of the array to equal however many variables there are in the array.
2. The `_numeric_` keyword instructs SAS to include all numeric variables in the array.
3. The DIM function returns the dimension of the array.
4. Finally, the Do- loop performs the statements between the Do and End statements for all variables in the array. In this example, the only statement is an If statement which sets values to missing, if the value is less than zero.

PREVENTING STATISTICAL ERRORS: II. SELECTING THE CORRECT PROCEDURE

At this point, readers who are programmers are feeling smug because they were already well aware of the uses of ARRAY statement, DO-loops, and the DIM function. Unfortunately, correct data can still lead to incorrect results without the appropriate application of weights and accommodation of stratification variables.

The following examples use another large dataset, the American Time Use Survey, which sampled over 13,000 Americans who completed questionnaires on how they had spent their time in the preceding twenty-four hours. These data were used to answer two simple questions:

- “Do men or women have more time alone in an average day?”
- “Does the amount of time one has alone differ based on having children in the home?”

At first glance, these questions could be answered very quickly using either a proc sort and proc means, or by use of proc tabulate.

```
Proc tabulate data = in.atus ;
Class child sex ;
Var timealone ;
Tables child*sex, timealone*mean*f= 12.1 ;
```

In short order, this method will provide the completely wrong answer. Chapter Seven of the American Time Use Survey documentation (Bureau of Labor Statistics, 2007) includes the following statement:

“Users need to apply weights when computing estimates with the ATUS data because simple tabulations of unweighted ATUS data produce misleading results.”

A random sample of 13,000 people, where everyone had an equal chance of being included can be generalized to

Turning Data into Information

the whole population. However, this survey is not a random sample but rather, a stratified sample selected to have a large enough number from certain groups to make generalizations to the population.

Think of it this way --- if a researcher samples 10,000 people with 25% from each of four ethnic groups, Caucasian, African-American, Latino and Asian-American, this sample can be used to generate pretty good estimates for each group on that, but it would be inaccurate to add the whole sample together and say that the country is 25% Asian-American.

To get an accurate population sample, multiply the weights by the number of minutes or hours a person report doing a particular activity. Then, divide by the sum of the weights.

The equation provided by ATUS is:

$$T_i = \frac{\sum \text{fwgt } T_{ij}}{\sum \text{fwgt}}$$

In other words

The average amount of time the population spends in activity j T_j is equal to

The sum of the weight for each individual multiplied by the individual responses of how much time they spend on activity j

Divided by the sum of the weights.

Fortunately, ATUS includes a weighting variable, TUFINLWGT , that can be used in the statement added to the TABULATE procedure

Weight tufinlwt ;

In this case, simply using the correct weights provides accurate results. The next example uses a dataset that is a subset of the same survey. In this case, the data were selected stratified by education and sex. The weight variable is called SAMPLINGWEIGHT. Either tabulate or means procedure could be used and both would give the correct means, however, the standard error of the mean would be incorrect, in some cases dramatically so.

In this example, a minimum of 40 subjects were selected from each strata in order to have an adequate sample size. The result is that, of the males with the lowest level of education, 40 of 41 (98%) were included in the sample, giving a very small error in estimating that population. For males with a high school diploma, again 40 were sampled, but out of 1,457 (3%), thus a much greater error can be expected in estimating the mean for that group.

If a stratified sample has been used, the correct procedure to calculate standard errors is the surveymeans procedure. Frequency counts for the number in each strata are included in a SAS dataset as shown below

	Edited: sex	educ	Frequency Count
1	1	0	41
2	1	1	119
3	1	2	150
4	1	3	208

```
proc surveymeans data=in.atu40 total = strata_count ;
  weight samplingweight ;
  strata sex educ ;
  var hrsworked numchildren ;
```

The above code will give correct means and standard errors for all strata.

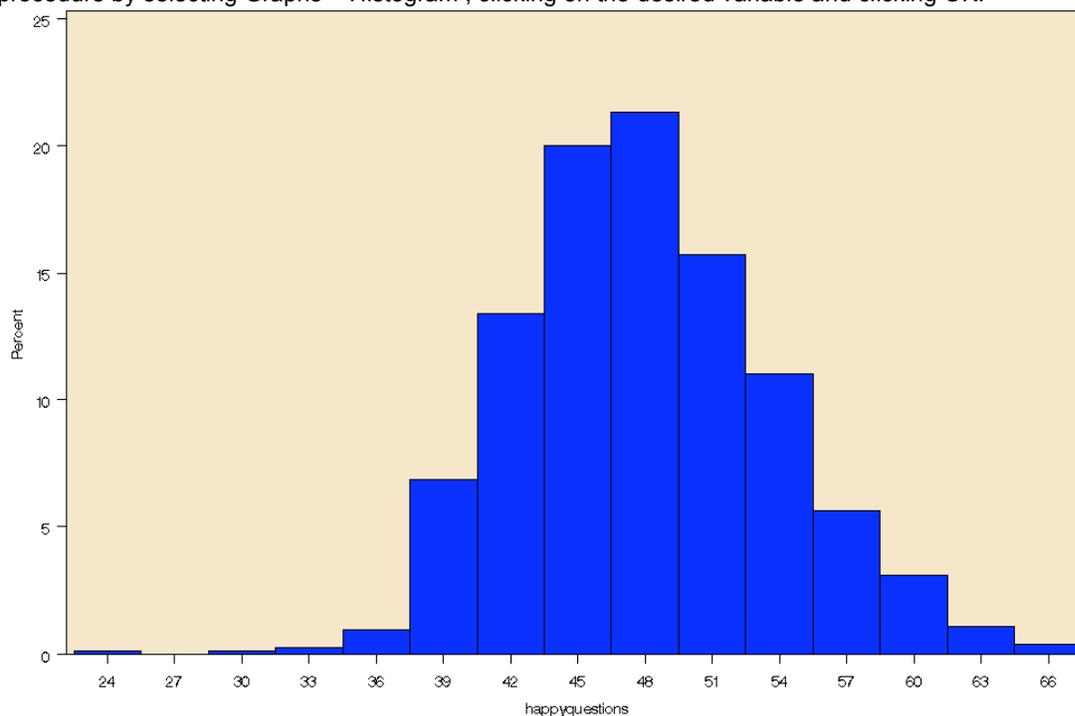
IT'S SCIENCE BUT IT DOESN'T HAVE TO BE ROCKET SCIENCE

Before a multiple regression, repeated measures multivariate analysis of variance or any other inferential statistical analysis, it is crucial to have an understanding of your data. Is it skewed or normally distributed? Parametric statistics

Turning Data into Information

make certain assumptions about the data, e.g., that the dependent variable is normally distributed. In relatively small sample sizes, outliers can produce misleading results. Avoid these pitfalls by spending some time with some of the under-used SAS applications for getting to know your data. These include Enterprise Guide, Graph-N-Go and the Analyst application.

Most statisticians are not motivated to become SAS programmers, but rather, to conduct accurate statistical analyses. Much simpler to use than SAS/Graph, the purpose of these sample charts produced with the SAS applications is not to provide publication quality output, but rather, to yield quick views of the data distribution. For example, the graph below, showing the distribution of a scale on marital happiness, is produced in the Analyst procedure by selecting Graphs > Histogram, clicking on the desired variable and clicking OK.



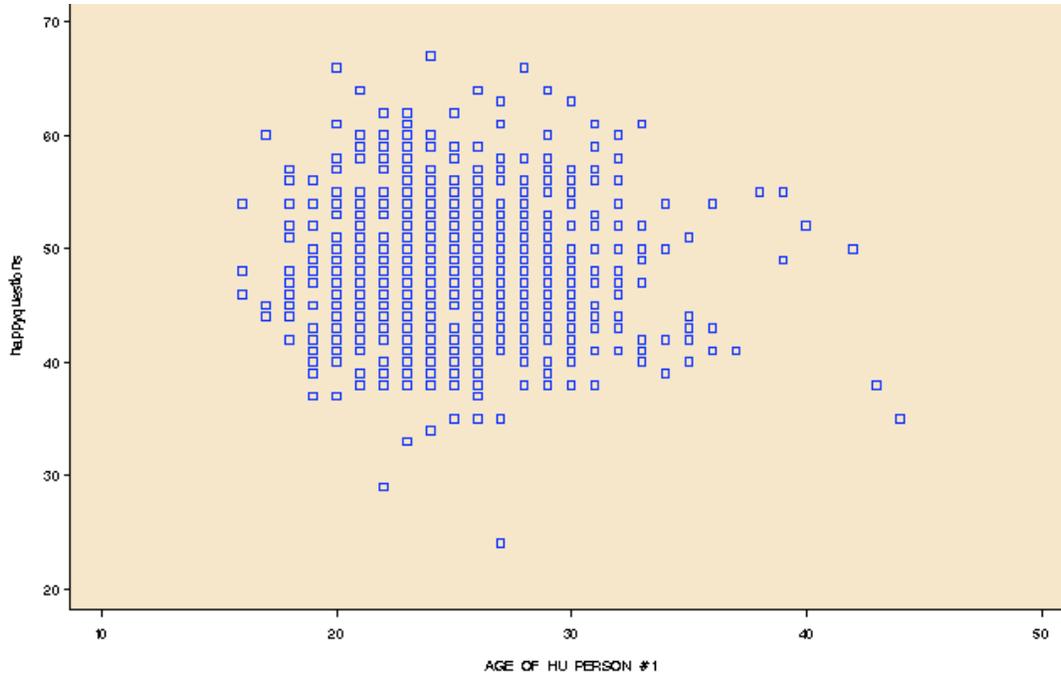
By clicking on “code”, the following code, produced by the Analyst application, can be seen, copied to the Program Editor and saved, if desired.

```
title; footnote;
goptions ftext=SWISS ctext=BLACK htext=1 cells;
symbol v=SQUARE c=BLUE h=1 cells;
proc univariate data=Work.Example noprint;
  var HAPPYQUESTIONS;
  histogram / caxes=BLACK cframe=CXF7E1C2 waxis= 1
             cbarline=BLACK cfill=BLUE pfill=SOLID
             vscale=percent hminor=0 vminor=0
             name='HIST';
symbol;
goptions ftext= ctext= htext=;
```

From the graph above, it is clear that the distribution approximates normality. While a statistician may wish to augment these graphs with measures of skewness and kurtosis, for purposes of discussion, visual displays of data may be preferable.

A second example of an Analyst graph, a scatter plot of questions on marital happiness by age, is shown below. Why not simply produce a correlation matrix? Both correlation coefficients and scatter plots would be recommended for understanding variables central to the analysis. While correlation provides a measure of linear relationship, scatter plots allow simultaneous assessment of curvilinear relationships, outliers and restriction of range.

Turning Data into Information



CONCLUSION

This paper has described a few methods for using SAS to increase the usefulness of data by reducing “noise” via elimination of extraneous variables, automated correction and identification of data entry errors, selection of appropriate statistical procedures and use of the Analyst procedure for visualizing data. In each of these four areas, SAS offers a vast number of additional resources. For example, to reduce variables, a combination of correlation procedures, internal consistency analysis and factor analysis using the CORR and FACTOR procedures can be used to combine multiple variables into a single scale.

Macros and %Include statements can be used to perform repetitive tasks across programs in the same way that Array statements and DO-loops are used within programs. Enterprise Guide, Graph-N-Go and other SAS applications provide a wide variety of point-and-click options for data exploration. Of course, SAS is well-known for the breadth and depth of statistical procedures, from frequency distributions to logistic regression for

While statisticians and programmers each tend to have areas of respective strengths, weaknesses and preferred tools, each group is strongly encouraged to explore and use the tools available in SAS, both simple and complex. Fewer errors would happen if we all would move from using what we comfortably know – Proc Means or using 47 If-Then statements – to learning what we don’t.

REFERENCES

Bureau of Labor Statistics (2007). American Time Use Survey user’s guide: Understand ATUS 2003-2006. Washington, D.C. U.S. Census Bureau.

Wolf, C. (2005). Using Proc Freq for Manageable Data Summarization Paper presented at Pittsburg SAS Users Group.
<http://www.dataceutics.com/papers/psug2005-CC27.pdf>

CONTACT INFORMATION

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

University Park Campus
Annmaria DeMars MC-2812
University of Southern California
Information Technology Services – CAL Building
Los Angeles, CA 90089
ademars@usc.edu
(213) 740-2840

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.