ABSTRACT

The study of missing data patterns may yield insight regarding the possibility of quality problems in a database. New in SAS® Version 9, and experimental in SAS Version 8, PROC MI allows analysts to use multiple imputation to handle missing values. One feature of PROC MI is the display of the patterns of missingness in a data set. To capitalize on this for data quality purposes, PROC MI was mimicked in the development of a SAS macro, MISS_PAT, which summarizes missing data patterns and tailors the output to ease interpretation. Multiple imputation via PROC MI is briefly summarized. Features of MISS_PAT that are improvements upon PROC MI are described. To illustrate the usefulness of the study of missing data patterns, results are presented from the application of MISS_PAT to a highway safety database.

Key Words: Missing data; Data quality; Multiple imputation; PROC MI; SAS macro; Motor vehicle traffic safety data

INTRODUCTION

The study of the patterns of missingness in a set of observations can be helpful in the quality assurance of data as well as in their statistical analysis. It is also the case that the study of these patterns is useful for multiple imputation of missing data. Thus PROC MI, a new SAS/STAT® procedure experimental in Version 8 and fully implemented in Version 9, provides summaries of the patterns of missing values in a SAS dataset. In attempting to exploit these summaries for data quality and analytical purposes, it was advantageous to develop a SAS macro, MISS_PAT, to overcome some disadvantages in PROC MI’s display of missing patterns.

MULTIPLE IMPUTATION (MI)

Missing data is a nuisance to the applied statistician. Either ignoring incomplete observations or replacing missing values with plausible values leads to greater uncertainty in the results of statistical analysis. Replacing missing values can also create the appearance of greater certainty than really exists. Multiple imputation (MI) can alleviate the latter to some extent.
In multiple imputation, a missing value is replaced by several plausible values. That is, from one dataset with missing values, several versions of the data are created each with a different set of imputed values to replace the missing values. This is done using PROC MI in SAS. Then each of these data sets is analyzed using the appropriate analysis. To draw appropriately qualified statistical conclusions, the multiple analyses need to be combined in the appropriate fashion. In SAS, this is done using PROC MIANALYZE.

MISSING PATTERNS

Suppose the data set of interest has P variables. Groups of observations can be created as a function of exactly which variables are known and which variables are missing. This is done by thinking in terms of an array of length P whose values are either 0 or 1. If the j-th variable in an observation is known, the j-th value of the array is 0; if missing, then 1.

Consider the very simple case of observations consisting of two variables, Y1 and Y2. There are four possible groups. Rather than indicating missingness by 1 and nonmissingness by 0, we do so with a ‘.’ or an ‘X’, respectively. The groups are shown in the following table.

Table 1. Missing Patterns for the Case of Two Variables

<table>
<thead>
<tr>
<th>Group</th>
<th>Y1</th>
<th>Y2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>.</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

Each of the four groups is actually a missing pattern. For example, going back to the 0/1 notation, the pattern for Group 3 is (1,0).

In PROC MI output, the pattern for Group 3 is indicated by a ‘.’ and an ‘X’. In addition to the columns for the variables, there are also columns for actual and relative frequency of the groups in the output. For example, suppose there were ten observations broken out by the four groups as in Table 2 below. Note the hypothetical variable names.

Table 2. Missing Patterns for Two Hypothetical Variables, with Frequency Information

<table>
<thead>
<tr>
<th>Group</th>
<th>Variab01</th>
<th>Second_Var</th>
<th>Freq</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>.</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>20</td>
</tr>
</tbody>
</table>
Note that the table above shows the frequencies for all possible missing patterns, i.e., groups. In the PROC MI output, the only groups that are shown are those with non-zero frequency. Also, the displayed groups are numbered consecutively. Thus, the actual SAS output would look like the following.

Table 3. PROC MI Output of Missing Patterns for Two Hypothetical Variables

<table>
<thead>
<tr>
<th>Group</th>
<th>Variab01</th>
<th>Second Var</th>
<th>Freq</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>.</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>.</td>
<td>2</td>
<td>20</td>
</tr>
</tbody>
</table>

The suppression of groups with zero frequency can lead to a big reduction in the number of groups printed. Furthermore with P variables, there are $2^P$ possible groups. Since there are only N observations, if P is large, $2^P$ is often much larger than N.

DISADVANTAGES WITH THE PROC MI OUTPUT OF MISSING PATTERNS

The basic approach used by PROC MI in the display of missing patterns is quite useful. However, there are two sets of difficulties in trying to use PROC MI to display the patterns. The first set consists essentially of formatting problems. The second set consists of problems of obtaining information in certain situations.

First, the formatting problems. In addition to the output shown above in Table 3, group means for each variable are given. As these will not usually be of interest for data quality purposes, and because the means require a lot of space to display when P is large, a way to suppress the group means was needed. This could be solved using SAS’s Output Delivery System (ODS) to put all the output related to the missing patterns into a data set. Then only the relevant subset of variables would be printed to eliminate the means.

Another problem is in the display of the missing patterns. Note that the column headers used to display the missing patterns are the variable names. With the advent of long variable names in SAS, this leads to wide headers in many instances. If a variable name were, say, 30 characters long, there would be a great deal of white space in the column because only one character is used to indicate the missingness status (i.e., ‘X’ for ‘not missing, and ‘.’ for ‘missing’). When P is fairly large, there can easily be a great deal of white space. This makes it difficult to decipher the multivariate nature of each missing pattern. A solution to this problem is to create aliases for each variable, using the name V1 as an alias for the first variable, V2 for the second variable, etc.

A third problem is the order in which the patterns are listed. In PROC MI the missing pattern, treated as a string of zeros and ones, can be viewed as a binary representation of
a number. The missing patterns are listed in the order of the magnitude of these numbers. For quality assurance purposes, we will usually be interested in the few patterns having the largest frequencies. If there are a substantial number of missing patterns for a given data set, it can be tedious to find the few with large frequencies and to try to decipher any relationship between those few patterns. A solution to this would be to take the ODS created SAS data set containing the missing pattern information, and to sort on FREQ prior to printing.

Next, there are the problems of obtaining information from PROC MI. If there are no missing data, there is just one pattern, that with an ‘X’ in every column. But in that situation, PROC MI suppresses the listing of the missing pattern in the output, and puts a warning message in the log. Thus, ODS is unable to save this missing pattern for outputting in the desired fashion.

Another problem occurs when BY processing is used with PROC MI. In most cases, different BY groups will have different sets of missing patterns. A BY group’s missing patterns will be numbered by PROC MI beginning with 1 and ending with the total number of missing patterns in the group. Thus if a certain missing pattern occurs in two different BY groups, PROC MI may assign it one group (i.e., pattern) number in the first BY group but a different group (pattern) number in the second BY group. This makes it difficult to compare how a missing pattern’s percentages vary as a function of BY group.

A final problem is that PROC MI excludes character variables in determining the missing patterns. For data quality purposes, it is desirable that all variables are used in constructing the missing patterns.

MISS_PAT, THE MACRO

Because of the desirability of avoiding the above problems when outputting the missing patterns for a data set, it was decided to write a macro to make a variety of enhancements. In the first attempt to do so, ODS was used to extract the desired information from PROC MI, and that information was manipulated to accomplish the desired formatting enhancements. However, the problems of obtaining information from PROC MI as described at the end of the last section were then discovered. So it was decided to write a new macro, named MISS_PAT, so as to output the missing patterns with all of the desired features. Base SAS programming including PROC SQL was used in doing so.

The main features of MISS_PAT are as follows.

- Aliases are substituted in the column headers in the display of the missing patterns.
- Two lists are generated displaying the alias name, the variable name, the variable label, and the variable type. One list is sorted by alias name, and the second list is sorted by variable name.
• All variables, including those of character type, are included in the missing patterns.
• The patterns are sorted by FREQ.
• If BY group processing is used, a first pass is made ignoring the BY groups so as to construct a complete list of patterns (groups) with non-zero frequency. Then in the BY group processing, the group number displayed is that from the complete list so as to ease comparisons between BY groups.
• Whether or not the complete list is printed in BY group processing is an option.
• To use the BY processing feature, sorting the desired BY variables is not needed.

The example in Tables 2 and 3 above does not involve BY group processing. The actual SAS output would appear as follows in the figure below.

**Figure 1. Macro MISS_PAT Output for Two Hypothetical Variables**

Variable Name Aliases for WORK.TWOVARS
Sorted by VARIABLE

<table>
<thead>
<tr>
<th>Obs</th>
<th>ALIAS</th>
<th>VARIABLE</th>
<th>LABEL</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V2</td>
<td>Second_Var</td>
<td>VARIABLE NUMBER 2</td>
<td>CHARACTER</td>
</tr>
<tr>
<td>2</td>
<td>V1</td>
<td>Variab01</td>
<td>VARIABLE NUMBER 1</td>
<td>NUMERIC</td>
</tr>
</tbody>
</table>

Variable Name Aliases for WORK.TWOVARS
Sorted by ALIAS

<table>
<thead>
<tr>
<th>Obs</th>
<th>ALIAS</th>
<th>VARIABLE</th>
<th>LABEL</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V1</td>
<td>Variab01</td>
<td>VARIABLE NUMBER 1</td>
<td>NUMERIC</td>
</tr>
<tr>
<td>2</td>
<td>V2</td>
<td>Second_Var</td>
<td>VARIABLE NUMBER 2</td>
<td>CHARACTER</td>
</tr>
</tbody>
</table>

Missing Data Patterns in WORK.TWOVARS Sorted By Descending PERCENT
(Variable Names Replaced By Aliases)

<table>
<thead>
<tr>
<th>Obs</th>
<th>V1</th>
<th>V2</th>
<th>Group</th>
<th>Freq</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>.</td>
<td>2</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>.</td>
<td>.</td>
<td>3</td>
<td>2</td>
<td>20</td>
</tr>
</tbody>
</table>

**USING MISS_PAT**

MISS_PAT and four associated utility macros are stored on the NESUG03 program CD in the CODE subdirectory under the name ST009.ZIP. The CD table of contents gives
the exact location of the ZIP file. On a PC, unzip the code. Then bring up SAS and run that code to make the macro available for use.

The macro has three keyword parameters: DS, BY, and COLLAPSE. DS is mandatory, and the other two are optional.

**DS** is a character string specifying the library name and data set (i.e., `libname.dataset`) for which you want to obtain the missing patterns.

**BY** is a character string giving a list of BY variables delimited by blanks. A separate set of missing patterns will be determined for each unique BY group in the data. Presorting on the BY variables is not required.

**COLLAPSE**, which defaults to NO, can be set to YES if in addition to a list of missing patterns for each BY group, you wish to see the missing patterns for all the data in the data set specified by DS (excluding the BY variables).

It is recommended that the MPRINT option be set prior to calling the macro. That way, the code generated by the macro is put into the SAS log.

The simplest call to MISS_PAT only specifies the DS parameter. As an example, the output in Figure 1 can be replicated using a data set named TWOVARS, which can be built by running the program TWOVARS.SAS given in ST009.ZIP. After running TWOVARS.SAS, the missing pattern analysis is produced using the following code.

```sas
OPTIONS MPRINT;
%MISS_PAT(DS=WORK.TWOVARS)
OPTIONS NOMPRINT;
```

Please report any bugs encountered using MISS_PAT to me. My contact information is given below.

**A REAL-LIFE EXAMPLE**

MISS_PAT was applied to NHTSA’s FARS (Fatality Analysis Reporting System) database, consisting of data obtained annually in a census of fatal motor vehicle crashes occurring in the United States. The data are hierarchical, consisting of crashes, vehicles within crashes, and persons within vehicles (or within crashes, as pedestrians). Prior to doing the work reported in this paper, my understanding was that the usual convention in storing FARS variables is to code unknown values as a numeric code, usually a ‘9’, and to use a special numeric code like ‘8’ in situations where the variable is not applicable. For example, Restraint Usage is not applicable to pedestrians.

Thus, I thought that except in special situations, SAS missing values like ‘.’ would generally not used in FARS. An example of a special situation is a driverless vehicle.
LCOMPL (compliance with license restrictions by the driver) is coded as a ‘.’ in that instance, which is quite unusual.

With the above understanding, I proceeded to apply MISS_PAT to the three FARS datasets for the year 2001. My intent was to get a comprehensive picture of the situations in which SAS missing values are used in FARS. I anticipated seeing few missing value patterns in each of the three datasets.

The crash dataset contained 49 variables. There was only one missing pattern, the special case in which no variable is missing. Thus, all was well in that dataset.

However, in the vehicle dataset, with 92 variables, there were 84 missing patterns. Of these, the first four comprised 56.8 percent, 30.8 percent, 5.0 percent, and 4.4 percent of the observations, respectively. The fifth pattern had only 0.5 percent of the observations. The first pattern consisted of cases in which the 58th and 60th variables were missing. These variables are MCYCL_DS (CC displacement of a motorcycle) and WGTCD_TR (Truck Weight Code). Instead of using a special value like ‘0’ to indicate ‘not applicable’, a SAS missing value is used for that purpose for these data elements. Similar explanations apply to patterns two and three. Pattern four (4.4 percent of observations) is the special case in which no variable is missing.

The person dataset, with 75 variables, had 18 missing patterns. Five patterns seemed to be the most prevalent, with the fourth highest being the special case of nonmissingness at 4.5 percent. Similar explanations to those for the vehicle dataset seem to apply.

Spot checks using MISS_PAT on earlier years back to 1995 gave similar results. Thus, the use of the SAS missing value code to indicate ‘not applicable’ occurs more frequently than I had previously realized.

CONCLUSIONS

Study of the patterns of missing data is worthwhile for data quality assurance, and for the analyst to understand what he may be facing in analyzing large sets to data. The macro MISS_PAT given with this paper should prove to be a useful tool to many data collection organizations and to many data users.

ACKNOWLEDGEMENTS

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 registered trademarks or trademarks of their respective companies.
CONTACT INFORMATION

Carl E. Pierchala
National Highway Traffic Safety Administration
NCSA, Room 6125, NPO-123
400 Seventh Street, SW
Washington, DC  20590
Work Phone:  (202) 366-6736
FAX:  (202) 366-3189
Email:  cpierchala@nhtsa.dot.gov