

CMISS® the SAS® Function You May Have Been “MISSING”

Mira Shapiro, Analytic Designers LLC, Bethesda, MD

ABSTRACT

Those of us who have been using SAS for more than a few years often rely on our tried- and-true techniques for standard operations like assessing missing values. Even though the old techniques still work, we often miss some of the “new” functionality added to SAS that would make our lives much easier. In effort to ascertain how many people skipped questions on a survey and, what percentage of people answered each question, I did a search of past conference papers and came across a function that was introduced in SAS 9.2-- CMISS. By using a combination of CMISS and Proc Transpose, a full missing assessment can be done in a concise program. This paper will demonstrate how CMISS makes assessing survey completeness an easy task.

INTRODUCTION

The first step in exploring a new data set includes a careful assessment of each variable's fill rate and missing values. In the past, MISSING was the only function specifically available for evaluating missing values in the Data Step. In SAS 9.2, the CMISS and NMISS functions were introduced simplifying the programming required to assess character and numeric missing values, respectively. Now, the CMISS function operates on both numeric and character variables simplifying the Data Step code required to perform a missing value assessment to using just one function.

As an experienced SAS programmer, I was unaware of CMISS but happily discovered this function in SAS 9.4. This discussion is focused on the CMISS function and how it can be used to quickly and easily create a missing value report for results of a survey by evaluating each variable's missing values and further, by evaluating how many questions respondents left unanswered. To assess whether a survey is too long, the ratio of unanswered to answered questions is often used and this discussion shows a quick and straightforward approach for this process.

The data used in this discussion was generated for this purpose. However, these analytic techniques may be applied to real world survey results.

DATA and METHODS

SAS® OnDemand for Academics, accessible without cost on the web, was used to run the SAS code and to create all of the results. The interface to SAS 9.4 is SAS Studio 3.4.

The dataset used throughout this paper was generated by a Haskell (GHC) program for the purposes of this demonstration. (The code to generate the dataset is available upon request.) The data includes the responses to a series of questions for music lovers along with their age. For clarity, the variables were named so that the question asked can be inferred from the variable names. In practice, the variable names would be simpler and label statements would be used for the purpose of description in the output. Table 1 lists the generated variables and their initial Type, Length, and Format.

Table 1 Synthetic Survey Data Imported from CSV File

Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
2	Age	Num	8	BEST12.	BEST32.
8	DidYouPlayInTheSchoolBand	Char	5	\$5.	\$5.
3	DoYouGoToConcerts	Char	5	\$5.	\$5.
9	DoYouKnowWhatAHouseConcertIs	Char	5	\$5.	\$5.
6	DoYouPlayAnInstrument	Char	5	\$5.	\$5.
1	FavoriteGenre	Char	11	\$11.	\$11.
10	HaveYouEverBeenToAHouseConcert	Char	5	\$5.	\$5.
5	HowManyTimePerMonthDoYouSeeAConc	Char	10	\$10.	\$10.
7	WhatInstrumentDoYouPlay	Char	8	\$8.	\$8.
4	WhyDontYouGoToConcerts	Char	15	\$15.	\$15.

CMISS: Numeric and Character Missing Assessment with One Function

There are many ways to work with missing values in SAS, among them is to use Data Step programming, procedure options and Proc SQL statements. This discussion focuses specifically on how to use the Data Step and employ the CMISS function effectively. Table 2 summarizes the appropriate use and output for each of the missing functions, including CMISS, and can be used as a quick reference when embarking on a missing value assessment

Table 2 Comparison of Missing Value Functions

Function	Numeric	Character	Results
MISSING	YES	YES	Numeric: missing(.) value returns 1 Character: blank value returns 1 single parameter only
NMISS	YES	NO	Numeric Variable *one variable: missing (.) value returns 1, valid returns 0 *multiple: adds 1 for each missing; returns sum Character Variable * all values return 1, NOTES indicate invalid numeric data and data converted to numeric single or multiple parameters numeric only (converts all arguments to numeric)
CMISS	YES / MIXED	YES / MIXED	Numeric Variable *one variable: missing (.) value returns 1, valid returns 0 *multiple: adds 1 for each missing; returns sum Character Variable *one variable: blank value returns 1, valid returns 0 *multiple: adds 1 for each missing; returns sum Mixed Character and Numeric *returns the sum of the blank character variables and the missing (.) numeric variables single or multiple parameters numeric, character or mixed

Table 2: Summary of missing value function usage and results.

SAS Program Using the CMISS: Function

This short SAS program provides all that is needed to assess the missing characteristics for both responders and survey questions. Note that the data, contained in a CSV file, was imported prior to this step using Proc Import code that was generated by SAS Studio and created as a temporary dataset named “work.import”.

The best way to describe the way this program works is to describe the statements and their purpose line by line.

Line 5: Use the set statement to read the temporary dataset into the SAS dataset, “survey_results”

Line 7: The retain statement is used here to initialize the var_missing variable to 0 so that the results of the CMISS function do not include the result assignment variable as missing.

Line 8: The format statement is used to format age (and any other numeric variables) to make sure that the empty fields are coded as “.”, the SAS standard definition for missing.

Line 9: This is where most of the important work is done:

The CMISS function is given the list of all of the variables to assess. In this case “of _all_” was used to include all variables that are defined in the program data vector (PDV). Using this approach required initialization of the assignment variable “var_missing”, otherwise the count of missing variables for a survey responder would be increased by one since the assignment variable would be counted as missing. It is worth noting here that the variables could be named individually and separated by commas or a Proc SQL step

could be used to create a macro variable that includes the names of all of the variables and be passed to CMISS. This approach used was chosen for its brevity and simplicity.

Line 10: The percent missing is calculated for each respondent. Note the numerator of 10 is the number of questions in the survey. To write a more general version of this program, the programmer would want to make this a variable or macro variable.

Line 13-14: This Proc Print will produce a report for all variables which allows checking of missing values and the results of the calculations. For display and reporting purposes SAS provides a multitude of ways to tailor results. One recommended way would be to chart or graph the % missing.

Line 17: This is where the fun begins. Proc Transpose is used to create the dataset “survey_results_t” where the original rows (observations) become variables and the original columns (variables) become observations. By doing this we can repeat almost the same code used in the first Data Step to assess missing by variable. The missing value approach could be implemented as a macro and then called with a few simple parameters including the data step name.

Line 18: The important feature of the “Var” statement to take note of here is the use of “_”. This allows for choosing a range of variables of interest in the report. In order to use this feature, it is important to understand the order in which the variables are stored internally in SAS.

Line 21-26: In this Data Step, the newly transformed dataset is evaluated in the same way as the original but, here we are looking at the data by question so we have a good understanding of the missing patterns for each question. Note that the denominator in the equation is 200 this time. That denotes the number of responders to the survey. Again, generalizing this program could be easily achieved by using either a variable or macro variable for this quantity.

Lines 29-33: The Proc Print used here displays the results of the question missing patterns.

```

1 * CSV File read in with SAS Studio generated Proc Import code;
2 * Use CMISS in the Data Step to evaluate missing values for responders;
3
4 data survey_results;
5     set work.import;
6     * initialize var_missing to get an accurate missing count from CMISS;
7     retain var_missing(0);
8     format age f2.0;
9     var_missing=cmiss(of _all_);
10    percent_missing=(var_missing/10)*100.00;
11 run;
12 * Quick way to check results;
13 proc print data=survey_results;
14     title "Survey Results: Missing & Percent Missing for Responders";
15 run;
16 * Transpose data to get missing value counts per question;
17 proc transpose data=survey_results out=survey_results_t prefix=person;
18     var FavoriteGenre--percent_missing;
19 run;
20 * Use CMISS in the Data Step to evaluate missing values for questions;
21 data survey_results_more;
22     set survey_results_t;
23     * initialize per_var_missing to get an accurate missing count from CMISS;
24     retain per_var_missing(0);
25     per_var_missing=cmiss(of _all_);
26     Percent_missing_per_var=(per_var_missing/200)*100.00;
27 run;
28
29 proc print data=survey_results_more;
30     title "Survey Results: Missing & Percent Missing for Questions";
31     var _name_ per_var_missing percent_missing_per_var;
32     label _name_="Question" per_var_missing="Count of Missing"
33           percent_missing_per_var="Percent Missing";
34 run;

```

The results of running the program are displayed below. There are many ways to display and further refine the output. The purpose of the discussion was to demonstrate the power of the CMISS function and to illustrate the power of the Data Step to simply explore data and provide insights for follow-on processing and reporting.

Results Part 1: Missing and Percent Missing for Responders

Survey Results: Missing & Percent Missing for Responders

Obs	FavoriteGenre	Age	DoYouGoToConcerts	WhyDontYouGoToConcerts	HowManyTimePerMonthDoYouSeeAConc
1	Jazz	29	True		Two
2		20	True		One
3	Other	38			
4	BlueGrass	42	True		Two
5	Pop	20	False	IDontLikeCrowds	
6	Folk	38	True		Two

DoYouPlayAnInstrument	WhatInstrumentDoYouPlay	DidYouPlayInTheSchoolBand	DoYouKnowWhatAHouseConcertIs
True	Violin	False	False
False			True
False			False
True	Trombone	True	True
True	Trombone	False	True
True	Violin	False	True

HaveYouEverBeenToAHouseConcert	var_missing	percent_missing
	2	20
False	4	40
	6	60
True	1	10
False	1	10
True	1	10
False	3	30

Result Part 2: Missing and Percent Missing for Survey Questions

Survey Results: Missing & Percent Missing for Questions

Obs	_NAME_	per_var_missing	Percent_missing_per_var
1	FavoriteGenre	16	8.0
2	Age	0	0.0
3	DoYouGoToConcerts	11	5.5
4	WhyDontYouGoToConcerts	149	74.5
5	HowManyTimePerMonthDoYouSeeAConc	62	31.0
6	DoYouPlayAnInstrument	6	3.0
7	WhatInstrumentDoYouPlay	61	30.5
8	DidYouPlayInTheSchoolBand	73	36.5
9	DoYouKnowWhatAHouseConcertIs	10	5.0
10	HaveYouEverBeenToAHouseConcert	57	28.5
11	var_missing	0	0.0
12	percent_missing	0	0.0

CONCLUSIONS

Seasoned SAS programmers are not always aware of new functions that are introduced in SAS. The CMISS function introduced in SAS 9.2, provides an elegant path to assessing missing values. CMISS in particular is a very useful tool in assessing survey missing patterns in a very straightforward and parsimonious way. This approach can be used for longer and more complex surveys and can be implemented in a more general way via a macro. It can also be embellished to account for skip patterns in a survey and other more specific survey requirements.

CMISS is one of many enhancements that have been made to SAS throughout the years. By researching a particular topic, through the SAS website and numerous SAS community resources, even seasoned SAS programmers can discover a new approach that might not have previously known about or considered.

SOURCES OF ADDITIONAL INFORMATION

There is a wealth of information available on SAS missing value functions and all aspects of SAS. Some useful resources are listed below.

Table 2: Resources

URL	Description
http://www.lexjansen.com/	A website that provides access and a search engine for SAS conference papers from SAS Global Forum, Regional conferences and specialized SAS conferences such as PharmaSUG
http://www.sascommunity.org/wiki/Main_Page	A SAS community resource that serves as a portal to many sources of SAS information on the web.
https://support.sas.com/	The SAS Customer Support website that contains a wealth of information including: troubleshooting, documentation and training.

CONTACT INFORMATION

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

Name: Mira Shapiro
E-mail: mira.shapiro at gmail.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.