

The SAS Data Set Missing Values Analyzer

Michael A. Raithel, Westat, Inc.

ABSTRACT

Occasionally, SAS programmers receive very wide SAS data sets with hundreds or thousands of variables. The data could have been extracted from Blaise, downloaded from a SQL Server table, obtained from an Oracle table, or sent by a client. Not all of the variables in the large SAS data set may have data stored in them. This can happen for any number of reasons, such as partial surveys, test tables, database tables with variables in them for future use, and upstream processing issues.

It is useful to know which variables actually have values stored in them. Analysts can concentrate on analyzing those variables and skip the others. Completely empty variables may indicate a problem with a file. Missing values might signify an issue with the collection instrument's skip patterns. Perhaps the chronically missing variables can be dropped from the SAS data set or from the originating database tables. Additionally, it is sometimes helpful to know which variables hold values for the individual cases within the SAS data set.

The SAS Data Set Missing Value Analyzer identifies all the variables that have one or more non-missing value in them for a given SAS data set. The program creates four types of Excel report files:

- Count of the Variables That Have Non-Missing Values For Each Distinct ID Variable
- Characteristics of All Variables With At Least One Non-Missing Value
- Characteristics of Variables With ALL Missing Values
- *Optionally*, a separate report for each distinct ID Variable titled: Characteristics of The Variables with One or More Non-Missing Values

This paper reviews the SAS Data Set Missing Value Analyzer macro program, and the report files it creates. Readers can begin using the macro directly after reading this paper.

INTRODUCTION

A SAS *missing value* is a value that indicates that no data value is stored for the variable in the current observation. SAS has several types of missing values:

- Numeric – Represented by a single decimal point: .
- Character – Represented by “”
- Special Missing – Numeric variables only that are represented by .a through .z and ._

Note that projects sometimes use negative numbers to signify missing data. For instance, -7, -8, -9, and so on. However, these are not true SAS missing values; and because of the breadth of possibilities, such missing values are not covered in this paper.

Figure 1 provides an example of a SAS data set that has missing values. The values highlighted in yellow indicate missing values. Notice that some variables (TOTINC, TOTINC_AT, VALUE, and WAGES) have low amounts of missingness, some variables (RETIR, and SEMPI) have high amounts of missingness, and some variables (NOTES, EMAIL, TOPCODE, and BOTTOMCODE) are completely missing for every observation.

DealerID	REGNO	REPAIR	RETR	ROOMS	SEMPI	SEX	SHELCO	SSGRAD	SUBSIDY	TENUR	TOTINC	TOTINC_AT	VALUE	VISMIN	WAGES	WKSWRK	WRKACT	YRIMM	notes	email	topcode	bottomcode
DEL00001	0	1		8		2	3904	11	9	1	44000	40000	1170000	8	43000	6	11	2007				
DEL00002																						
DEL00003	0	1		5		1	1400	8	0	2	48000	38000		13	47000	6	11	-1				
DEL00004	0	1		8		2	1000	8	9	1	37000	32000	956142	13	37000	6	11	-1				
DEL00005	0	3		8		2	3904	4	9	1	14000	15000	1922085	13	14000	3	6	-1				
DEL00006	0	2		7		2	700	6	9	1	54000	44000	260000	13	54000	6	11	-1				
DEL00007	0	1		8		2	1700	4	0	2	14000	14000		13	14000	6	12	-1				
DEL00008	0	1		6		1	800	99	9	1			550000	2		9	99	-1				
DEL00009	0	1		6		2	800	6	9	1	652888	400891	510000	13	562553	6	11	-1				
DEL00010	0	1		10		1	1000	4	9	1	14000	14000	650000	1	7000	4	8	2014				
DEL00011	0	2		88		2	1200	6	8	8	79000	54000	8888888	13	80000	6	11	-1				
DEL00012	0	2		3		1	600	4	9	1	29000	25000	210000	13	24000	6	11	-1				
DEL00013	0	1		10		2	600	8	9	1	52000	45000	1710000	88	53000	5	9	7				
DEL00014	1	2		6		1	1600	8	0	2	82000	82000		13	64000	6	11	-1				
DEL00015	0	1		5		2	2100	99	9	1			460000	13		9	99	-1				
DEL00016	0	1		3		1	300	1	1	2	15000	14000		7		9	1	1993				
DEL00017	0	3		88		1	1500	6	8	2	31000	32000		88	13000	6	12	1996				
DEL00018	0	1		8		2	1200	99	9	1			250000	7		9	99	-1				
DEL00019	0	2		9	2000	1	1400	8	9	1	4000	4000	700000	13	1000	5	10	-1				
DEL00020	0	2		4		2	900	8	0	2	65000	54000		13	65000	6	11	-1				
DEL00021	0	3		4		1	1500	6	0	2	15000	16000		3	14000	3	6	1995				
DEL00022	0	2		8		1	1200	6	9	1	21000	21000	160000	13	9000	5	10	-1				
DEL00023	0	1		7		1	1600	8	9	1	26000	26000	470000	13	24000	6	12	-1				
DEL00024	0	1		8		1	1100	6	9	1	11000	11000	220000	13	10000	6	12	-1				
DEL00025	0	1		6		1	400	99	0	2				13		9	99	-1				
DEL00026	0	2		5		2	400	6	9	1	42000	36000	270000	13		6	11	-1				
DEL00027	0	1	25700	8	-1	2	2300	6	9	1	140000	100000	440000	13	140000	6	11	-1				
DEL00028	0	1		8		1	1300	99	9	1			330000	13		9	99	-1				
DEL00029	0	1		11		2	2722	4	9	1	20000	19000	420000	13	19000	5	9	-1				
DEL00030	0	1		4		2	1400	1	9	1	1000	1000	220000	1		9	1	2002				
DEL00031	0	1		6		1	300	99	0	2				13		9	99	-1				
DEL00032	0	2		7		1	600	99	9	1			100000	13		9	99	-1				
DEL00033	0	3		9		2	2000	3	9	1	15000	16000	280000	13	14000	5	9	-1				
DEL00034	0	2		8		2	1200	4	9	1	36000	36000	300000	13	15000	3	6	7				

Figure 1. Graphical view of a SAS data set with missing values.

The SAS Data Set Missing Value Analyzer (SDSMVA) offers an automated method of checking for missingness within a SAS data set by identifying all variables that have one or more non-missing values in them. The SDSMVA creates four types of Excel report files:

- Count of the variables that have non-missing values for each Distinct ID variable
- Characteristics of all variables with at least one non-missing value
- Characteristics of variables with all missing values
- **OPTIONALLY:** A separate report for each distinct ID variable titled: Characteristics of the Variables with One or More Non-Missing Values

Output 1 is an example of a report showing the number of variables that have data in them for each unique ID variable in a data set. Only two observations have non-missing data for all 147 variables. And one observation—with DealerID “DEL000002” has missing values for all but the DealerID variable.

	A	B
		Number of Variables with Non-missing Values
1	DealerID	
2	DEL00001	147
3	DEL00002	1
4	DEL00003	145
5	DEL00004	144
6	DEL00005	145
7	DEL00006	144
8	DEL00007	145
9	DEL00008	138
10	DEL00009	146
11	DEL00010	147
12	DEL00011	146
13	DEL00012	146
14	DEL00013	144
15	DEL00014	146

Output 1. Count of the variables that have non-missing values for each Distinct ID variable

Output 2 is an example of a report showing the characteristics of all variables that have at least one non-missing value in them.

	Variable Name	Numeric or Character	Length	Format	Label
1	ABOID	N	8	BEST12.	ABOID
2	AGEGRP	N	8	BEST12.	AGEGRP
3	AGEIMM	N	8	BEST12.	AGEIMM
4	ATTSCH	N	8	BEST12.	ATTSCH
5	BEDRM	N	8	BEST12.	BEDRM
6	BFNMEMB	N	8	BEST12.	BFNMEMB
7	CAPGN	N	8	BEST12.	CAPGN
8	CFINC	N	8	BEST12.	CFINC
9	CFINC_AT	N	8	BEST12.	CFINC_AT
10	CFSIZE	N	8	BEST12.	CFSIZE
11	CFSTAT	N	8	BEST12.	CFSTAT
12	CHDBN	N	8	BEST12.	CHDBN
13	CHLDC	N	8	BEST12.	CHLDC
14	CIP2011	N	8	BEST12.	CIP2011
15	CIP2011_STEM_SUM	N	8	BEST12.	CIP2011_STEM_SUM
16	CITIZEN	N	8	BEST12.	CITIZEN
17	CITOTH	N	8	BEST12.	CITOTH
18	CMA	N	8	BEST12.	CMA
19	CONDO	N	8	BEST12.	CONDO
20	COW	N	8	BEST12.	COW
21					

Output 2. Characteristics of all variables with at least one non-missing value

Output 3 is an example of a report of all variables that contain nothing but missing values for the entire SAS data set.

Variable Name	Variable Type	Length	Format	Label
bottomcode	num	8		Bottom Coded Case
email	char	32		Subject's Email
notes	char	10		Interviewer Notes
OASGI	num	8	BEST12.	
OTINC	num	8	BEST12.	
topcode	num	8		Marked for Topcoding

Output 3. Characteristics of variables with all missing values

Output 4 is an example of one of the reports that was generated for each of the hundreds of distinct DealerID's in the SAS data set. That is; one report was generated for each observation in the SAS data set. It specifies the characteristics of each variable that does not contain missing variables for that particular observation. Note that, since there is a potential for hundreds or even thousands of these reports to be generated, this is a purely optional report.

	Variable Name	Numeric or Character	Length	Format	Label
1	ABOID	N	8	BEST12.	ABOID
2	AGEGRP	N	8	BEST12.	AGEGRP
3	AGEGRP	N	8	BEST12.	AGEGRP
4	AGEIMM	N	8	BEST12.	AGEIMM
5	ATTSCH	N	8	BEST12.	ATTSCH
6	BEDRM	N	8	BEST12.	BEDRM
7	BFNMEMB	N	8	BEST12.	BFNMEMB
8	CFINC	N	8	BEST12.	CFINC
9	CFINC_AT	N	8	BEST12.	CFINC_AT
10	CFSIZE	N	8	BEST12.	CFSIZE
11	CFSTAT	N	8	BEST12.	CFSTAT
12	CIP2011	N	8	BEST12.	CIP2011
13	CIP2011_STEM_SUM	N	8	BEST12.	CIP2011_STEM_SUM
14	CITIZEN	N	8	BEST12.	CITIZEN
15	CITOTH	N	8	BEST12.	CITOTH
16	CMA	N	8	BEST12.	CMA
17	CONDO	N	8	BEST12.	CONDO
18	COW	N	8	BEST12.	COW

Output 4. Report for the observation where DealerID equals “DEL00001” showing the characteristics all variables with non-missing values

The MissingValueAnalyzer macro has five parameters:

- **DIRNAME** - The directory of the target SAS data set
- **DSNAME** - The name of the target SAS data set
- **IDVARIABLE** - An ID variable that uniquely identifies an observation
- **REPTDIR** - The directory where the output Excel report files will be written
- **INDIVIDUALREPORTS** - A value of **Y** will produce a separate Excel report file for each distinct ID Variable

Display 1 is an example of specifying the parameters for the MissingValueAnalyzer macro.

```
%INCLUDE "\\CarsProject\ProgramLibrary\SAS Data Set Missing Values Analyzer.sas";

%MissingValuesAnalyzer(DIRNAME=\\CarsProject\DataLibrary,
                        DSNAME=CollectedData,
                        IDVARIABLE=DealerID,
                        REPTDIR=\\CarsProject\ReportsDirectory,
                        INDIVIDUALREPORTS=Y);
```

Display 1. Specifying the parameters for the MissingValueAnalyzer macro.

The next five sections provide the annotated code found in the MissingValueAnalyzer macro. Note that the entire program can be found in Appendix A.

PART 0 – OPTIONS AND FILE SPECIFICATIONS

The program opens with the macro statement listing the parameters that will be passed to the macro. Several ODS statements perform various housekeeping tasks. A libref of **InputLib** is specified for the directory holding the target SAS data set. The full path name of the directory is input via the &DIRNAME macro variable. See **Display 2**.

```

%MACRO MisssingValuesAnalyzer(DIRNAME=, DSNAME=, IDVARIABLE=, REPTDIR=, INDIVIDUALREPORTS=);

/* Options and file specifications */
options nofmterr symbolgen mprint mlogic;

ods html close;
ods listing close;
ods results off;
ods noproctitle;

libname InputLib "&DIRNAME" access=readonly;

```

Display 2.

PART 1 – CREATING A REPORT OF THE NUMBER OF NON-MISSING-VALUE VARIABLES PER UNIQUE ID VARIABLE

This section of the program starts with a DATA step that is going to determine the name, length, label, format, and type (numeric or character) of each variable in the target SAS data set. The ID variable is passed to the KEEP option of the DATA statement via the &IDVARIABLE macro variable. The name of the target data set is passed to the SET statement via the &DSNAME macro variable. Next, two arrays are created: one containing all of the numeric variables, the other containing all of the character variables. A LABEL and LENGTH statement complete the first part of this DATA step. See **Display 3**.

```

/* Read target SAS data set and only keep variables */
/* that DO NOT have missing values in them. */
data filledvars(keep= &IDVARIABLE varname numeric_or_character varlength varlabel varformat);
set InputLib.&DSNAME;

array nums{*} _numeric_;
array chars{*} _character_;

label varname          = "Variable Name"
      varlength        = "Length"
      varlabel         = "Label"
      varformat        = "Format"
      numeric_or_character = "Numeric or Character"
      ;

length varname          $32.
      varlength         8.
      varlabel          $256.
      varformat         $49.
      numeric_or_character $1.
      ;

```

Display 3.

Next, two DO loops are used to determine the name, length, label, and format of each variable. The first DO loop works on the numeric variables, using the numeric array previously created. It uses a group of variable functions (VNAME, VLENGTH, VLABEL, and VFORMAT) to identify each numeric variable's metadata. The second DO loop does the same thing for the character variables.

Then the data are sorted by the ID variable. See **Display 4** and **Display 5**.

```

/* Inspect numeric variables for non-missing values and keep the */
/* names, lengths, labels, and formats of non-missing variables */
do I = 1 to dim(nums);

    if not missing(nums{i}) then do;

        varname    = vname(nums{i});
        varlength  = vlength(nums{i});
        varlabel   = vlabel(nums{i});
        varformat  = vformat(nums{i});

        numeric_or_character = "N";

        output;

    end;
end;

```

Display 4.

```

/* Inspect character variables for non-missing values and keep the */
/* names, lengths, labels, and formats of non-missing variables */
do I = 1 to dim(chars);

    if not missing(chars{i}) then do;

        varname = vname(chars{i});
        varlength = vlength(chars{i});
        varlabel = vlabel(chars{i});
        varformat = vformat(chars{i});

        numeric_or_character = "C";

        output;

    end;
end;

run;

/* Sort the data set by &IDVARIABLE varname to get data in &IDVARIABLE order */
proc sort data=filledvars;
    by &IDVARIABLE varname;
run;

```

Display 5.

The program continues by running a frequency against the new, sorted data set (FILLEDVARS) and writing the frequency output to data set FREQOUT. Then ODS is used to create the first report spreadsheet, which is a *count of all variables having non-missing values in the observation* for each unique ID variable. This is the report that was illustrated in **Output 1**. See **Display 6**.

```

/* Calculate the frequency of non-missings for each individual &IDVARIABLE */
proc freq data=filledvars noprint;
  tables &IDVARIABLE / out=freqout norow nocol nopercent nocum ;
  title1 "Number of Variables That Have Non-Missing Values";
  title2 "For Each Distinct: &IDVARIABLE";
run;

/* Create the spreadsheet report of the number of non-missing-value variables by &IDVARIABLE */

ODS EXCEL file="%REPTDIR%\&DSNAME Count of Variables That Have Non-Missing Values For Each Distinct &IDVARIABLE.xlsx";
ODS EXCEL options(sheet_name="Obs With Nonmissing Values" flow="header,data");

proc print noobs data=freqout label;
  var &IDVARIABLE count;
  title1 "Count of Variable Variables That Have Non-Missing Values";
  title2 "For Each Distinct: &IDVARIABLE";
  label count = "Number of Variables with Non-missing Values";
run;

ODS EXCEL CLOSE;

```

Display 6.

PART 2 – CREATING A REPORT WITH A LIST OF VARIABLES THAT HAD ONE OR MORE NON-MISSING VALUES IN THEM

A simple PROC SORT is used to sort the FILLEDVARS data set into ascending alphabetical order by the VARNAME variable. Then, the report of *all variables with one or more non-missing values in them* is created. This is the report that was illustrated in **Output 2**. See **Display 7**.

```

/* Sort the data into variable name order */
proc sort nodupkey data=filledvars (keep=varname vartype varlength varformat varlabel) out=uniquevars;
  by varname;
run;

/* Create the Excel report file in the specified report directory */
ODS EXCEL file="%REPTDIR%\&DSNAME - Characteristics of All Variables With At Least One Non-Missing Value.xlsx";

ODS EXCEL options(sheet_name="Vars With Nonmissing Values" flow="header,data");

proc print noobs data = uniquevars label;
  var varname vartype varlength varformat varlabel;
  title1 "Characteristics of All Variables With At Least One Non-Missing Value";
run;

ods EXCEL close;

```

Display 7.

PART 3 – CREATING A REPORT OF CHARACTERISTICS OF VARIABLES WITH ALL MISSING VALUES

This section of the program is the busiest with a SORT, PROC DATASETS, another SORT, a MERGE, and a DATA _NULL_ before running a PROC PRINT to create a report of the characteristics of all variables that have at least one non-missing value in them in the target SAS data set.

The sort obtains all variable attributes from the target SAS data set from sashelp.vcolumn, sorts them into variable name order, and outputs them to a temporary SAS data set named ALLVARS. The DATASETS

procedure is then used to rename variables in ALLVARS and to assign labels to them. ALLVARS will be used in a future merge in this section. See **Display 8**.

```
/* Sort all of the target data set variable metadata into variable name order */
proc sort data=sashelp.vcolumn(where=(libname="INPUTLIB" and memname=upcase("&DSNAME")))
    out=allvars(keep=name type length label format);
    by name;
run;

/* Rename and label selected variables for the upcoming report */
proc datasets library=work noprint;
    modify allvars;
    rename name      = varname
           length    = varlength
           label      = varlabel
           format     = varformat
           type       = vartype
    ;

    label varname    = "Variable Name"
          varlength   = "Length"
          varlabel     = "Label"
          varformat    = "Format"
          vartype      = "Variable Type"
    ;

run;
quit;
```

Display 8.

Part 3 continues by sorting FILLEDVARS by VARNAME for the upcoming match-merge. Then, FILLEDVARS and ALLVARS are match-merged together into new SAS data set ALLMISSING. Only observations that are in ALLVARS but not in FILLEDVARS are kept in the new data set. So, ALLMISSING will contain only variables that did not have a single value in them—the variables that had missing values in them throughout the entire SAS data set. See **Display 9**.

```
/* Sort previously-created filledvars into variable name order for the match-merge with allvars */
proc sort data=filledvars;
    by varname;
run;

/* Match merge with filledvars to find only variables with _ALL_ missing values */
data allmissing;
    merge filledvars(in=a)
          | allvars(in=b)
    ;
    by varname;

    if b and not a;

run;
```

Display 9

The last section of Part 3 first determines if there are any variables in the ALLMISSING data set. If so, then the macro variable &NUMMISSING will be set to the value 1 or higher. If not, then it will be zero.

So, if the value of NUMMISSING is not equal to zero, then ALLMISSING is sorted by VARNAME and the report is created. This is the report that was illustrated in **Output 3**. See **Display 10**.

```
/* Determine if there are any observations in ALLMISSING */
data _null_;
set sashelp.vtable(where=(libname = "WORK" and memname = "ALLMISSING" and memtype = "DATA"));

call symput("NUMMISSING", nobs);

run;

/* Only create this report if there are observations in ALLMISSING */
%IF &NUMMISSING NE 0 %THEN %DO;

    /* Sort the allmissing variables by varname for printing */
    proc sort data=allmissing sortseq=linguistic(numeric_collation=on);
        by varname;
    run;

    /* Create the Excel report file in the specified report directory */
    ods excel file="%REPTDIR%\&DSNAME - Characteristics of Variables With ALL MISSING VALUES.xlsx";

    ods excel options(sheet_name="Vars With ALL MISSING Values" flow="header,data");

    proc print noobs data = allmissing label;
    var varname vartype varlength varformat varlabel;
    title1 "Characteristics of Variables With All Missing Values";
    run;

    ods excel close;

%END;
```

Display 10.

PART 4 – CREATING A REPORT OF THE NUMBER OF NON-MISSING-VALUE VARIABLES FOR EACH INDIVIDUAL, UNIQUE ID VARIABLE

This final section of the program is only run if the optional &INDIVIDUALREPORTS macro variable was set to “Y” by the person running the macro. If &INDIVIDUALREPORTS = Y, then a separate report file is created for each observation in the SAS data set. The beginning of each report file’s name is the value of the ID variable (&IDVARIABLE) so that the reports have distinct names and can be correlated with the ID that uniquely identifies them.

If invoked (&INDIVIDUALREPORTS = Y), this section begins by determining the number of observations in the data set and storing it in macro variable TOTOB. The next PROC SQL stores the value of each ID variable (identified by macro parameter &IDVARIABLE) into a macro array of variables CASEID1, CASEID2, CASEID3, and so on. See **Display 11**.

```

%IF %UPCASE(&INDIVIDUALREPORTS) = Y %THEN %DO;

*****;
* Determine number of observations in filledvars data set *;
*****;
proc sql noprint;
create table &IDVARIABLE as
    select distinct &IDVARIABLE
        from filledvars;

%let TOTOBS = &SQLOBS;

quit;

*****;
* Load &IDVARIABLE values into distinct macro variables *;
*****;
proc sql noprint;
select distinct &IDVARIABLE into : caseidl - :caseid&TOTOBS
    from filledvars
        ;
quit;

```

Display 11.

Now, a macro DO loop is executed to create each individual report using a DATA step and a PROC PRINT. The macro DO loop executes from 1 to the value of &TOTOBS, which is the number of the last observation in the FILLEDVARS data set. The DATA step subsets FILLEDVARS to a single observation, based on the value stored in the CASEIDx macro variable. The ODS statements use the current value of CASEIDx to create a unique Excel file for the report. The Print procedure prints the report for the particular observation in the original SAS data set. This is the report that was illustrated in **Output 4**. See **Display 12**.

The %MEND statement signals the end of the %MissingValuesAnalyzer macro program.

```

*****;
* Loop to create the individual SAS data sets and Excel spreadsheets *;
*****;
%Do i = 1 %to &totobs %by 1;

    /* Subset the data set for each individual value of IDVARIABLE */
    data Individual;
    set filledvars(where=(IDVARIABLE = "&caseid&i"));
    run;

    /* Create the Excel report file in the specified report directory */
    ODS EXCEL file="&REPTDIR\&caseid&i.. Variables with One or More Non-Missing Values.xlsx";

    ODS EXCEL options(sheet_name="&caseid&i.. Non-missing Vars" flow="header,data");

    proc print noobs data = Individual label;
    var varname vartype varlength varformat varlabel;
    titel "Variables with One or More Non-Missing Values for IDVARIABLE: &caseid&i ";

    run;

    ods EXCEL close;

%END;

%END;

%MEND MissingValuesAnalyzer;

```

Display 12.

OPERATIONALIZING THE SAS DATA SET MISSING VALUES ANALYZER

You can operationalize the *SAS Data Set Missing Values Analyzer.sas* program by copying it from Appendix A into a new **.sas** program file on your computer with the same name. Once the program has been copied, move it to one of your autocall macro libraries. This will ensure that when you invoke the program with %MissingValuesAnalyzer(*and specifying the proper parameters*), it will automatically be invoked.

An alternative is to copy the driver program, *Execute SAS Data Set Missing Values Analyzer.sas*, from Appendix B into a new **.sas** program file on your computer. Modify the %INCLUDE statement in that file to reflect where you stored the *SAS Data Set Missing Values Analyzer.sas* program. Then, fill in the parameters and run the program. The code below illustrates the %INCLUDE and the parameters that must be specified:

```

%INCLUDE "\\CarsProject\ProgramLibrary\SAS Data Set Missing Values Analyzer.sas";

%MissingValuesAnalyzer(DIRNAME=\\CarsProject\DataLibrary,
                      DSNAME=CollectedData,
                      IDVARIABLE=DealerID,
                      REPTDIR=\\CarsProject\ReportsDirectory,
                      INDIVIDUALREPORTS=Y);

```

Display 13.

CONCLUSION

In very large SAS data sets, it is useful to know which variables actually hold values. Analysts can concentrate on those variables and skip the others. Missing values may indicate a problem with the file, with a data collection instrument's skip patterns, or with upstream processing. You may be able to drop variables with all missing values altogether. Additionally, it can be helpful to know which variables do have values for individual cases.

The SAS Data Set Missing Value Analyzer identifies all the variables that have one or more non-missing value in them for a given SAS data set. The program creates four types of Excel report files:

- Count of the Variables That Have Non-Missing Values For Each Distinct ID Variable
- Characteristics of All Variables With At Least One Non-Missing Value
- Characteristics of Variables With ALL Missing Values
- *Optionally*, a separate report for each distinct ID Variable titled: Characteristics of The Variables with One or More Non-Missing Values

So, copy the code in Appendix A and B and start analyzing your own organization's large SAS data sets for missing values.

REFERENCES

Raithel, Michael A. 2017. **Did You Know That? Essential Hacks for Clever SAS Programmers: Over 100 Essential Hacks to Make Your Programs Leaner, Cleaner, and More Competitive**. Bethesda, Maryland: Michael A. Raithel

Available: http://www.amazon.com/Michael-A.-Raithel/e/B001K8GG90/ref=ntt_dp_epwbk_0

PROC DATASETS; The Swiss Army Knife of SAS® Procedures, Michael A. Raithel
[3440-2016.pdf \(sas.com\)](http://www.sas.com/3440-2016.pdf)

SAS 9.4 and SAS Viya 3.5 Programming Documentation
[SAS Help Center: Missing Variable Values](http://www.sas.com/help/missing-variable-values)

SAS® 9.4 Macro Language: Reference, Fifth Edition
[SAS Help Center: SAS 9.4 Macro Language: Reference, Fifth Edition](http://www.sas.com/help/sas-9.4-macro-language-reference-fifth-edition)

ACKNOWLEDGMENTS

The author like to thank Westat management for supporting his participation in SESUG 2024.

RECOMMENDED READING

Raithel, Michael A. 2017. **Did You Know That? Essential Hacks for Clever SAS Programmers: Over 100 Essential Hacks to Make Your Programs Leaner, Cleaner, and More Competitive**. Bethesda, Maryland: Michael A. Raithel

Available: http://www.amazon.com/Michael-A.-Raithel/e/B001K8GG90/ref=ntt_dp_epwbk_0

CONTACT INFORMATION

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

Michael A. Raithel
Westat, Inc.
MichaelRaithel@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.

APPENDIX A. THE SAS DATA SET MISSING VALUES ANALYZER.SAS PROGRAM

```

/*****
/*  Program: SAS Data Set Missing Values Analyzer.sas
/*
/*
/*  Author: Michael A. Raithel
/*
/*
/*  Created: 04/05/2023
/*
/*
/*  Purpose: This program examines a SAS data set and identifies all
/* variables that have one or more non-missing values in them. It is
/* designed to allow a relatively quick way for analysts to inspect very
/* "wide" SAS data sets and skip over possibly thousands of variables with
/* missing values to zero in on only those that actually contain data that
/* can be analyzed. The program creates four types of Excel spreadsheet
/* reports--see "Outputs" for details.
/*
/*
/*  Parameters: Users must specify the following parameters:
/*
/*
/*          DIRNAME      - The directory of the target SAS data set
/*          DSNAME        - The name of the target SAS data set
/*          IDVARIABLE    - An ID variable that uniquely identifies an
/*                        observation - REQUIRED!
/*          REPTDIR       - The directory where the output Excel report
/*                        files will be written
/*          INDIVIDUALREPORTS - A value of "Y" (no quotes, please) will
/*                        produce a separate Excel report file for
/*                        each distinct IDVARIABLE.
/*
/*
/*  Outputs: This program creates the following Excel report files in the
/* &REPTDIR directory:
/*
/*          1."&DSNAME Count of Variables That Have Non-Missing Values
/*            For Each Distinct IDVARIABLE..xlsx" - A report with one row
/*            for each unique IDVARIABLE with the number of variables with
/*            non-missing values in that particular observation.
/*
/*          2."&DSNAME - Characteristics of All Variables With At Least
/*            One Non-Missing Value.xlsx" A data set level report of all
/*            variables that have at least one non-missing value in them.
/*            it includes whether the variable is character or numeric,
/*            length, format, and label.
/*            label.
/*
*****/

```

```

/*          3. "&DSNAME - Characteristics of Variables With ALL MISSING  */
/*          VALUES" - This Excel report file is generated for variables*/
/*          that have missing values for EVERY OBSERVATION in the      */
/*          target SAS data set. If all of the variables have at least */
/*          one non-missing value, the report file is not generated.    */
/*                                                                    */
/*          4. "&caseid&i.. Variables with One or More Non-Missing    */
/*          Values.xlsx" - An individual Excel report file is generated*/
/*          for EACH unique IDVARIABLE listing the variables with non- */
/*          missing values in that specific observation. It includes    */
/*          whether the variable is character or numeric, and the      */
/*          variable's length, format, and label.                      */
/*                                                                    */
/*          NOTE: MACRO PARAMETER INDIVIDUALREPORTS=Y MUST BE SPECIFIED*/
/*          FOR THE INDIVIDUAL REPORTS TO BE PRODUCED! IT IS OKAY TO   */
/*          LEAVE IT BLANK OR SPECIFY: N IF YOU DO NOT WANT THE        */
/*          INDIVIDUAL-LEVEL REPORTS.                                  */
/*                                                                    */
/*****

%MACRO MisssingValuesAnalyzer(DIRNAME=, DSNAME=, IDVARIABLE=, REPTDIR=,
                           INDIVIDUALREPORTS=);

/* Options and file specifications */
options nofmterr symbolgen mprint mlogic;

ods html close;
ods listing close;
ods results off;
ods noproctitle;

libname InputLib "&DIRNAME" access=readonly;

/*****
/*          PART 1
/*****
/* Create a report of the number of non-missing-value variables per unique */
/* &IDVARIABLE
/*
/*****

/* Read target SAS data set and only keep variables */
/* that DO NOT have missing values in them.
data filledvars(keep= &IDVARIABLE varname vartype varlength varlabel
                  varformat);
set   InputLib.&DSNAME;

array nums{*} _numeric_;
array chars{*} _character_;

label varname    = "Variable Name"
      varlength  = "Length"
      varlabel   = "Label"
      varformat  = "Format"
      vartype    = "Numeric or Character"
      ;

```

```

length varname    $32.
      varlength    8.
      varlabel     $256.
      varformat    $49.
      vartype      $4.
      ;

/* Inspect numeric variables for non-missing values and keep the */
/* names, lengths, labels, and formats of non-missing variables */
do I = 1 to dim(nums);

      if not missing(nums{i}) then do;

            varname    = vname(nums{i});
            varlength  = vlength(nums{i});
            varlabel   = vlabel(nums{i});
            varformat  = vformat(nums{i});

            vartype = "num";

            output;

      end;
end;

/* Inspect character variables for non-missing values and keep the */
/* names, lengths, labels, and formats of non-missing variables */
do I = 1 to dim(chars);

      if not missing(chars{i}) then do;

            varname = vname(chars{i});
            varlength = vlength(chars{i});
            varlabel = vlabel(chars{i});
            varformat = vformat(chars{i});

            vartype = "char";

            output;

      end;
end;

run;

/*Sort the data set by &IDVARIABLE varname to get data in &IDVARIABLE order*/
proc sort data=filledvars;
      by &IDVARIABLE varname;
run;

/* Calculate the frequency of non-missings for each individual &IDVARIABLE */
proc freq data=filledvars noprint;
      tables &IDVARIABLE / out=freqout norow nocol nopercnt nocum ;
      title1 "Number of Variables That Have Non-Missing Values";
      title2 "For Each Distinct: &IDVARIABLE";

```

```

run;

/* Create the spreadsheet report of the number of non-missing-value */
/* variables by &IDVARIABLE */

ODS EXCEL file="%REPTDIR%\&DSNAME Count of Variables That Have Non-Missing
              Values For Each Distinct &IDVARIABLE..xlsx";

ODS EXCEL options(sheet_name="Obs With Nonmissing Values"
                  flow="header,data");

proc print noobs data=freqout label;
    var &IDVARIABLE count;
title1 "Count of Variable Variables That Have Non-Missing Values";
title2 "For Each Distinct: &IDVARIABLE";
label count = "Number of Variables with Non-missing Values";
run;

ODS EXCEL CLOSE;

/*****
/*
/*
PART 2
/*
/*
/* Create a spreadsheet with a list of variables that have one or more
/* values across ALL of the observations in the specified SAS data set.
/*
/*
*****/

/* Sort the data into variable name order */
proc sort nodupkey data=filledvars(keep=varname vartype varlength varformat
                                varlabel) out=uniquevars;

    by varname;
run;

/* Create the Excel report file in the specified report directory */
ODS EXCEL file="%REPTDIR%\&DSNAME - Characteristics of All Variables With At
              Least One Non-Missing Value.xlsx";

ODS EXCEL options(sheet_name="Vars With Nonmissing Values"
                  flow="header,data");

proc print noobs data = uniquevars label;
    var varname vartype varlength varformat varlabel;
title1 "Characteristics of All Variables With At Least One Non-Missing
Value";
run;

ods EXCEL close;

/*****
/*
/*
PART 3
/*
/*
/* Create a spreadsheet with a list of variables that have ONLY MISSING
/* VALUES across ALL of the observations in the specified SAS data set.
*****/

```

```

/*                                                                 */
/*****
/* Sort all of the target data set variable metadata into variable name */
/* order                                                                 */
proc sort data=sashelp.vcolumn(where=(libname="INPUTLIB" and
                                memname=upcase("&DSNAME")))
    out=allvars(keep=name type length label format);
    by name;
run;

/* Rename and label selected variables for the upcoming report */
proc datasets library=work noprint;
    modify allvars;
    rename name      = varname
           length    = varlength
           label      = varlabel
           format     = varformat
           type       = vartype
           ;

    label varname     = "Variable Name"
          varlength    = "Length"
          varlabel     = "Label"
          varformat    = "Format"
          vartype      = "Variable Type"
          ;

run;
quit;

/* Sort previously-created filledvars into variable name order for the match-
merge with allvars */
proc sort data=filledvars;
    by varname;
run;

/* Match merge with filledvars to find only variables with _ALL_ missing
values */
data allmissing;
merge filledvars(in=a)
      allvars(in=b)
      ;
by varname;

if b and not a;

run;

/* Determine if there are any observations in ALLMISSING */
data _null_;
set sashelp.vtable(where=(libname = "WORK" and memname = "ALLMISSING" and
                          memtype = "DATA"));

call symput("NUMMISSING", nobs);

run;

```

```

/* Only create this report if there are observations in ALLMISSING */
%IF &NUMMISSING NE 0 %THEN %DO;

    /* Sort the allmissing variables by varname for printing */
    proc sort data=allmissing sortseq=linguistic(numeric_collation=on);
        by varname;
    run;

    /* Create the Excel report file in the specified report directory */
    ODS EXCEL file="&REPTDIR\&DSNAME - Characteristics of Variables With
        ALL MISSING VALUES.xlsx";

    ODS EXCEL options(sheet_name="Vars With ALL MISSING Values"
        flow="header,data");

    proc print noobs data = allmissing label;
        var varname vartype varlength varformat varlabel;
    title "Characteristics of Variables With All Missing Values";
    run;

    ods EXCEL close;

%END;

/*****
/*
PART 4
*****/
/*
/* Create a report of the number of non-missing-value variables for each */
/* individual, unique value of the &IDVARIABLE variable. This results in */
/* as many spreadsheets as there are &IDVARIABLEs. Each spreadsheet name */
/* will begin with the value of its &IDVARIABLE. NOTE: This section is only */
/* executed if the &INDIVIDUALREPORTS parameter was coded: Y */
*****/

%IF %UPCASE(&INDIVIDUALREPORTS) = Y %THEN %DO;

*****;
* Determine number of observations in filledvars data set *;
*****;
proc sql noprint;
create table &IDVARIABLE as
    select distinct &IDVARIABLE
        from filledvars;

%let TOTOB = &SQLOB;

quit;

*****;
* Load &IDVARIABLE values into distinct macro variables *;
*****;
proc sql noprint;
select distinct &IDVARIABLE into : caseid1 - :caseid&TOTOB
    from filledvars;

```

```

quit;

*****;
* Loop to create the individual SAS data sets and Excel spreadsheets *;
*****;
%Do i = 1 %to &totobs %by 1;

    /* Subset the data set for each individual value of IDVARIABLE */
    data Individual;
    set filledvars(where=(&IDVARIABLE = "&&caseid&i"));
    run;

    /* Create the Excel report file in the specified report directory */
    ODS EXCEL file="&REPTDIR\&&caseid&i.. Variables with One or More Non-
        Missing Values.xlsx";

    ODS EXCEL options(sheet_name="&&caseid&i.. Non-missing Vars"
        flow="header,data");

    proc print noobs data = Individual label;
        var varname vartype varlength varformat varlabel;
    title1 "Variables with One or More Non-Missing Values for &IDVARIABLE:
        &&caseid&i ";

    run;

    ods EXCEL close;

%END;

%END;

%MEND MissingValuesAnalyzer;

```

APPENDIX B. THE EXECUTE SAS DATA SET MISSING VALUES ANALYZER.SAS DRIVER PROGRAM

```

/*****
/*  Program: Execute SAS Data Set Missing Values Analyzer.sas          */
/*                                                                    */
/*  Author: Michael A. Raithel                                         */
/*                                                                    */
/*  Created: 04/05/2023                                                */
/*                                                                    */
/*  Purpose: This is the driver program for the SAS Data Set Missing */
/*  Analyzer macro program.                                           */
/*                                                                    */
/*  That SAS macro program examines a SAS data set and identifies all */
/*  variables that have one or more non-missing values in them. It is */
/*  designed to allow a relatively quick way for analysts to inspect very */
/*  "wide" SAS data sets and skip over possibly thousands of variables with */
/*  missing values to zero in on only those that actually contain data that */
/*  can be analyzed. The program creates four types of Excel spreadsheet */
/*  reports--see "Outputs" for details.                               */
/*                                                                    */
/*

```

```

/* Parameters: Users must specify the following parameters: */
/* */
/*      DIRNAME      - The directory of the target SAS data set */
/*      DSNAME       - The name of the target SAS data set */
/*      IDVARIABLE    - An ID variable that uniquely identifies an */
/*                      observation - REQUIRED! */
/*      REPTDIR       - The directory where the output Excel report */
/*                      files will be written */
/*      INDIVIDUALREPORTS - A value of "Y" (no quotes, please) will */
/*                      produce a separate Excel report file for */
/*                      each distinct IDVARIABLE. */
/* */
/* Outputs: This program creates the following Excel report files in the */
/* &REPTDIR directory: */
/* */
/*      1."&DSNAME Count of Variables That Have Non-Missing Values */
/*        For Each Distinct IDVARIABLE..xlsx" - A report with one row */
/*        for each unique IDVARIABLE with the number of variables with */
/*        non-missing values in that particular observation. */
/* */
/*      2."&DSNAME - Characteristics of All Variables With At Least */
/*        One Non-Missing Value.xlsx" A data set level report of all */
/*        variables that have at least one non-missing value in them. */
/*        it includes whether the variable is character or numeric, */
/*        length, format, and label. */
/* */
/*      3. "&DSNAME - Characteristics of Variables With ALL MISSING */
/*        VALUES" - This Excel report file is generated for variables */
/*        that have missing values for EVERY OBSERVATION in the */
/*        target SAS data set. If all of the variables have at least */
/*        one non-missing value, the report file is not generated. */
/* */
/*      4. "&caseid&i.. Variables with One or More Non-Missing */
/*        Values.xlsx" - An individual Excel report file is generted */
/*        for EACH unique IDVARIABLE listing the variables with non- */
/*        missing values in that specific observation. It includes */
/*        whether the variable is character or numeric, and the */
/*        variable's length, format, and label. */
/* */
/*      NOTE: MACRO PARAMETER INDIVIDUALREPORTS=Y MUST BE SPECIFIED */
/*      FOR THE INDIVIDUAL REPORTS TO BE PRODUCED! IT IS OKAY TO */
/*      LEAVE IT BLANK OR SPECIFY: N IF YOU DO NOT WANT THE */
/*      INDIVIDUAL-LEVEL REPORTS. */
/* */
/*****

```

```

%INCLUDE "<YOUR PROJECT DRIVE>\SAS Data Set Missing Values Analyzer.sas";

```

```

%MissingValuesAnalyzer(DIRNAME=,
                      DSNAME=,
                      IDVARIABLE=,
                      REPTDIR=,
                      INDIVIDUALREPORTS=);

```