SAS® GLOBAL FORUM 2021

Paper 1027-2021

The SAS® Data Set Characterization Utility

Michael A. Raithel, Westat, Inc.

ABSTRACT

Most SAS programmers reach for two tools when they first receive a new SAS data set: PROC CONTENTS and PROC MEANS. They use PROC CONTENTS to review the data set's metadata; the physical attributes of the variables such as name, label, type, and length. They use PROC MEANS to determine the basic arithmetic characteristics of the numerical variables, such as the minimum, maximum, and mean values. Doing this involves running two different SAS procedures, combing through two separate SAS-generated reports, and correlating the information about specific variables between the disparate reports.

The SAS Data Set Characterization Utility generates a single report file that contains the best of both the CONTENTS and the MEANS procedures. It produces an Excel file with a single row of consolidated metrics for each variable found in the SAS data set. The variable's metrics include its key metadata attributes and—for numeric variables—its basic statistical properties. Additionally, the report contains the number of missing values for character and for numeric variables. Consequently, SAS programmers can utilize this utility to determine both the composition and the characteristics of a new SAS data set from a single amalgamated report.

This paper presents the details of the SAS Data Characterization Utility and provides information on how SAS programmers can begin using it right away.

INTRODUCTION

PROC CONTENTS provides metadata level variable information; PROC MEANS provides numerical variable value information. You run each one separately to get a basic understanding of a SAS data set. Wouldn't it be nice to have the key information from both procedures in one report?

The new **SAS Data Set Characterization Utility.sas** macro program allows you to get very detailed information about every variable in a particular SAS data set. This utility program produces an Excel report file with a row for each variable in the SAS data set. The Excel report file has the following columns in it:

- Variable Name
- Variable Label
- Variable Type
- Variable Length
- Variable Number
- Number of Non-Missing Values
- Number of Missing Values
- Variable Format
- For Numeric variables:
    - Min

    - Max
    - Mean
    - Standard Deviation

…for each variable in the specified SAS data set.

Appendix A provides an example of the Excel report that the utility program produces.

The **SAS Data Set Characterization Utility.sas** macro program requires three parameters:

- **DIRNAME** - The directory holding the SAS data set that is to be characterized
- **DSNAME** - The name of the target SAS data set that is to be described
- **REPTDIR** - The directory where the Excel report file will be written.

The Macro program is composed of five Parts. The following sections of this paper describe what the macro's SAS code does in each part. Refer to **Appendix B** to see the actual code of the SAS Data Set Characterization Macro.

PART 1 - USE PROC CONTENTS TO GET BASIC VARIABLE INFORMATION

This section begins by running PROC CONTENTS against the target SAS data set. The CONTENTS procedure output is stored in the **Contents_Vars** SAS data set so that we can capture the variable information important to us:

- Name
- Label
- Type
- Length
- Format
- Varnum

Next, we create variable Vtype in a DATA step. Vtype will have the value of "Num" or "Char", depending upon whether Type was a 1 or a 2, respectively. We do this to make the variable type obvious to both non-SAS programmers and SAS programmers alike. That is; they do not have to remember what a Type 1 or 2 is; they can easily discern that a variable is numeric or character.

The last thing done in Part 1 is to sort the **Contents_Vars** SAS data set by Name. This is done for the future match merging that will be done in Part 4.

PART 2 - ACCESS VTABLE TO DETERMINE THE NUMBER OF CHARACTER AND NUMERIC VARIABLES

This part executes a DATA _NULL_ step to access the SASHELP.VTABLE observation for the target data set. Once it gets that observation, it stores the number of character variables in the SAS data set in macro variable CHARCOUNT; and stores the number of numeric variables in the SAS data set in macro variable NUMCOUNT. Those two macro variables are going to be very important in the subsequent parts of this program.

PART 3 – CREATE SEPARATE DATA SETS CONTAINING ATTRIBUTES OF THE NUMERIC AND CHARACTER VARIABLES

This *workhorse* section of the program creates a distinct data set for numeric variables and one for character variables; if they exist in the target SAS data set. Each data set contains the properties of greatest interest for the variables. Part 3 first addresses numeric variables, and then addresses character variables.

If the NUMCOUNT macro variable does not equal zero, then this section processes the numeric variables. It starts by running an open-ended PROC MEANS against the target SAS data set and storing the output in a SAS data set named **MeansOutput**. Next, we run PROC TRANSPOSE against MeansOutput to create **MeansTranspose**—a linearized version of the original data set output from the MEANS procedure.

Next, the **MeansTranspose** data set is input to a DATA step, which ultimately creates our final numeric-oriented SAS data set: **NumericVars**. The DATA step processes our transposed MEANS data and creates the following variables from it:

- N
- Min
- Max
- Mean
- STD
- NMiss

That last variable, NMiss is the number of missing values for a particular numeric variable.

When the **NumericVars** SAS data set has been constructed, the next step is to use the SORT procedure. The numeric variable data are sorted by Name for future merging. That concludes the actions taken on numeric variables in Part 3.

If the CHARCOUNT macro variable does not equal zero, then the second section of Part 3 processes the character variables. The only thing that this section computes is the number of missing values for each character variable. This is done by creating an array of the character variables and a temporary array that will hold the number of missing values for each character variable.

For each observation, we loop through the values of each character variable and, if it is missing, add a 1 to the missing value counter for that variable. At the end of the data step, when all observations have been processed, we determine the number of missing values (Nmiss) for each variable. The **CharacterVars** SAS data set ends up with three variables:

- Name
- N
- NMiss

The final action of Part 3 is to sort the **CharacterVars** SAS data set by Name for future merging.

PART 4 - MERGE PREVIOUSLY CREATED DATA SETS TOGETHER WITH CONTENTS_VARS

Part 4 merges the **Contents_Vars** SAS data set created in Part 1 with the numeric and character SAS data sets we created in Part 3. It is macrotized into three main sections that first test whether the target SAS data set has numeric and/or character variables before executing. As with Part 3, this is done by means of inspecting the NUMCOUNT and the CHARCOUNT macro variables that were created in Part 2. The final output is the **FinalMetaData** SAS data set.

Starting off; if there are numeric variables, then we merge data sets **NumericVars** with **Contents_Vars** by name. This gets us the first cut of the **FinalMetaData** SAS data set containing all of the computed and characteristics information for each numeric variable.

At this point, if there are character variables in the target SAS data set, we merge **CharacterVars** into **FinalMetaData**. In doing so, we compute the number of non-missing values (N) for each character variable. This DATA step nets us a **FinalMetaData** SAS data set that contains the relevant metadata and summary statistics for all variables in the target SAS data set. So, we can now move on to Part 5.

In some cases, the target data set may not have any numeric variables. When that happens, we drop to the last DATA step in Part 4 which handles only character variables. This data step merges **CharacterVars** with **Contents_Vars**. It computes the number of non-missing values (N) for each character variable. All of this is written to the final SAS data set: **FinalMetaData**. Now, we can move on to Part 5.

PART 5 – CREATE THE REPORT FILE

This part capitalizes on our good work from previous sections to create a report file from the **FinalMetaData** SAS data set using the Output Delivery System. This is achieved by using the PRINT procedure; being sure to specify the NOOBS option. The report file is an Excel data set that is created in the directory specified in the REPTDIR macro. The name of the Excel file is of the form:

&DSNAME Data Set Characterization Report &SYSDATE..xlsx

...where:

- DSNAME – is the actual name of the target SAS data set (without the .sas7bdat extension)
- SYSDATE – is the current system date

If there are numeric variables in the SAS data set, then a format of 10.2 is stated for the min, max, mean, and std output variables; and a label is created for the STD variable.

OPERATIONALIZING THE SAS DATA SET CHARACTERIZATION UTILITY MACRO

Here are some simple steps you can consider implementing to operationalize the SAS Data Set Characterization Utility SAS program in your own environment.

1. Copy the macro from Appendix B into a SAS program on your computer. You can save it in your autocall macro library or save it in a directory of your choosing.

2. If you do not choose to put the program in one of your organization's macro libraries, take a look at Appendix C, which contains the Execute SAS Data Set Characterization Utility SAS program. This is a driver program for the SAS Data Set Characterization Utility SAS program. Simply copy the SAS code from Appendix C to a SAS program in your environment. Then, update the %INCLUDE statement in the driver program to specify the full path to where you placed the utility macro.

3. Once you have "downloaded" the macro program and the driver program, simply specify the three parameters in the macro call:

    - **DIRNAME** - The directory holding the SAS data set that is to be characterized
    - **DSNAME** - The name of the SAS data set that to be described
    - **REPTDIR** - The directory where the report will be written.

…and execute the macro to create your consolidated SAS data set report.

CONCLUSION

This paper introduced the *SAS Data Set Characterization Utility* program; which can be used to provide summary statistics and attributes of each variable in a SAS data set. The program requires for you to specify the directory holding the SAS data set, the SAS data set, and the directory where you want the report to be written. The utility program creates an Excel output file with up to twelve columns of information for each variable.

The *SAS Data Set Characterization Utility* program is ideal for creating a digest of important information about each variable in a SAS data set. Consider how this program can be of help when you need to investigate the characteristics of a new SAS data set in your own programming environment.

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

Raithel, M.A. (2017). PROC DATASETS; The Swiss Army Knife of SAS Procedures. *Proceedings of the SAS Global Forum 2017 Conference*.
Available: http://support.sas.com/resources/papers/proceedings17/0963-2017.pdf

SAS Institute Inc. 2015. *Base SAS® 9.4 Procedures Guide, Fifth Edition*. Cary, NC: SAS Institute Inc.
Available: http://support.sas.com/documentation/cdl/en/proc/68954/PDF/default/proc.pdf

SAS Institute Inc. 2015. *SAS® 9.4 Language Reference: Concepts, Fifth Edition*. Cary, NC: SAS Institute Inc.
Available: http://support.sas.com/documentation/cdl/en/lrcon/68089/PDF/default/lrcon.pdf

ACKNOWLEDGMENTS

The author would like to thank Bob McConnaughey for giving him the idea for this program and for helping him to develop it. The author would also like to thank Westat management for supporting his participation in SAS Global Forum 2021.

RECOMMENDED READING

- *Base SAS® Procedures Guide*
- *SAS® For Dummies®*

CONTACT INFORMATION

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

Michael A. Raithel
michaelraithel@westat.com

APPENDIX A – THE SAS DATA SET CHARACTERIZATION UTILITY REPORT FILE

Variable Name	Variable Label	Variable Type	Variable Length	Variable Number	Non-Missing Values	Missing Values	Min	Max	Mean	Standard Deviation	Variable Format
Cylinders		Num	8	9	426	2	3.00	12.00	5.81	1.56	
DriveTrain		Char	5	5	428	0	.	.	.	.	
EngineSize	Engine Size (L)	Num	8	8	411	17	1.30	8.30	3.20	1.11	
Horsepower		Num	8	10	428	0	73.00	500.00	215.89	71.84	
Invoice		Num	8	7	428	0	9875.00	173560.00	30014.70	17642.12	DOLLAR
Length	Length (IN)	Num	8	15	428	0	143.00	238.00	186.36	14.36	
MPG_City	MPG (City)	Num	8	11	411	17	10.00	60.00	20.03	5.13	
MPG_Highway	MPG (Highway)	Num	8	12	428	0	12.00	66.00	26.84	5.74	
MSRP		Num	8	6	428	0	10280.00	192465.00	32774.86	19431.72	DOLLAR
Make		Char	13	1	428	0	.	.	.	.	
Model		Char	40	2	403	25	.	.	.	.	
Origin		Char	6	4	403	25	.	.	.	.	
Type		Char	8	3	428	0	.	.	.	.	
Weight	Weight (LBS)	Num	8	13	428	0	1850.00	7190.00	3577.95	758.98	
Wheelbase	Wheelbase (IN)	Num	8	14	411	17	89.00	144.00	108.08	8.35	

APPENDIX B – THE SAS DATA SET CHARACTERIZATION UTILITY.SAS

```
/*****************************************************************************************************/
/*  Program: SAS Data Set Characterization Utility.sas                                               */
/*                                                                                                   */
/*  Author: Michael A. Raithel based on a program and GREAT IDEA by: Bob McConnaughey                */
/*                                                                                                   */
/*  Created: 5/1/2020                                                                                */
/*                                                                                                   */
/*  Purpose: This SAS program creates a report of detailed variable-level information for a specific  */
/*           SAS data set, including counts of non-missing and missing values; and min, max, mean,   */
/*           and std for numeric variables.                                                          */
/*                                                                                                   */
/*  Parameters: Users must specify the following three parameters:                                    */
/*                                                                                                   */
/*             DIRNAME    - The directory holding the SAS data set that is to be characterized.      */
/*             DSNAME     - The name of the SAS data set to be characterized.                       */
/*             REPTDIR    - The directory where the report will be written.                         */
/*                                                                                                   */
/*  Outputs: This program creates a Excel spreadsheet with the same name as the SAS data set with    */
/*           the words "Data Set Characterization Report <DATE REPORT WAS CREATED>" appended to it.  */
/*                                                                                                   */
/*  Change Log:                                                                                      */
/*                                                                                                   */
/*         12/15/20 - MAR - Consolidated missing value computation for character variables in Part 3. */
/*                                                                                                   */
/*****************************************************************************************************/

%MACRO CHARFILE(DIRNAME=, DSNAME=, REPTDIR=);

options symbolgen mprint mlogic source2 orientation=landscape nodate nonumber;

libname sasdata "&DIRNAME" access=readonly;

*************************************************************************;
* PART 1 - Use PROC CONTENTS to get basic variable information          *;
*************************************************************************;

/* PROC CONTENTS to get variable attributes */
proc contents data=sasdata.&DSNAME noprint
           out=Contents_Vars(keep=name label type length format varnum);
run;

/* Create Vtype variable */
data Contents_Vars;
set  Contents_Vars;

drop type;

if type = 1 then Vtype = "Num ";
       else if type = 2 then Vtype = "Char";

run;

/* Sort for future match-merge */
proc sort data=Contents_Vars;
       by Name;
run;


*****************************************************************************;
* PART 2 - Access VTABLE to determine the number of character and numeric variables.*;
*          Store them in macro variables which will then be used to determine which *;
*          subsequent sections of the program are executed.                   *;
*****************************************************************************;
data _null_;
set  sashelp.vtable(where=(libname = "SASDATA" and memname = upcase("&DSNAME") and memtype =
"DATA"));

call symput("CHARCOUNT", num_character);

call symput("NUMCOUNT",num_numeric);

run;
```

```
*******************************************************************************;
* PART 3 - If &NUMCOUNT NE 0 then the data set has numeric variables that are     *;
*          processed in this part.                                                *;
*          If &CHARCOUNT NE 0 then the data set has character variables that are  *;
*          processed in this part.                                                *;
*******************************************************************************;


        /*****************************************************/
        /* ONLY DO THIS SECTION IF THERE ARE NUMERIC VARIABLES*/
        /*****************************************************/

%IF &NUMCOUNT NE 0 %THEN %DO;

        /* Execute PROC MEANS against data set of interest to get stats of interest */
        proc means data = sasdata.&DSNAME maxdec = 2 noprint;
        output out=MeansOutput;
        run;

        /* Transpose to linearize the data and get variables in the desired order */
        proc transpose data=MeansOutput out=MeansTranspose;
        run;

        /* Create variables we want and drop superfluous observations */
        Data NumericVars(keep= Name NonMissing NumMissingValues Min Max Mean STD);
        retain Name NonMissing NumMissingValues Min Max Mean STD;
        set  MeansTranspose;

        length Name $32.;

        retain TotalObs;

        if _NAME_ = "_TYPE_" then delete;

        if _NAME_ = "_FREQ_ " then do;
                TotalOBs = COL1;
                delete;
        end;

        Name       = _NAME_;
        NonMissing = COL1;
        Min        = COL2;
        Max        = COL3;
        Mean       = COL4;
        STD        = COL5;

        NumMissingValues = TotalObs - COL1;

        run;

        /* Sort for future match-merge */
        proc sort data=NumericVars;
                by Name;
        run;

%END;


        /*******************************************************/
        /* ONLY DO THIS SECTION IF THERE ARE CHARACTER VARIABLES*/
        /*******************************************************/

%IF &CHARCOUNT NE 0 %THEN %DO;

        /* Determine missing values for character variables */
        data CharacterVars;
        set sasdata.&DSNAME end = eof;

        length Name $32.;

        keep Name NonMissing NumMissingValues;
```

```
        array charvars[*] _character_;
        array missing_count[&CHARCOUNT] _temporary_;

        do i = 1 to &CHARCOUNT;
                if missing(charvars[i]) then missing_count[i] + 1;
        end;

        if eof then do;
                do J = 1 to &CHARCOUNT;
                  Name = vname(charvars[J]);
                  NumMissingValues = missing_count[J];
                        if NumMissingValues = . then NumMissingValues = 0;
                        NonMissing = _N_ - NumMissingValues;
                  output;
              end;
        end;

        run;

        /* Sort for future match-merge */
        proc sort data=CharacterVars;
                by Name;
        run;

%END;

*******************************************************************************;
* PART 4 - Merge previously created data sets together with Contents_Vars.     *;
*                                                                              *;
*          The first &If/&DO/&END block addresses numeric variables if they exist.*;
*          Once the numeric variables are merged, it checks whether character  *;
*          variables exist. If so, they are merged too.                        *;
*                                                                              *;
*          The second &If/&DO/&END block is ONLY executed if NO numeric variables *;
*          exist. It processes the character variables in the SAS data set.    *;
*******************************************************************************;

        /*****************************************************************************/
        /* This section is executed if there are NUMERIC variables in the data set  */
        /*****************************************************************************/

%IF &NUMCOUNT NE 0 %THEN %DO;  /* BEGIN block for when there are numeric variables */

        /* Merge numeric variable data with metadata for each variable */
        data FinalMetaData;
        retain Name Label Vtype Length Varnum NonMissing NumMissingValues Min Max Mean STD;
        merge NumericVars Contents_Vars;
                by Name;
        run;

        %IF &CHARCOUNT NE 0 %THEN %DO; /* BEGIN block for when there are BOTH character and
numeric variables */

                /* Merge in character data to get character variable missing value counts */
                data FinalMetaData;
                merge FinalMetaData CharacterVars;
                        by Name;
                run;

        %END; /* END block for when there are BOTH character and numeric variables */

%END;    /* END block for when there are numeric variables */


        /*****************************************************************************/
        /* This section is executed if there are ONLY CHARACTER variables in the data set */
        /*****************************************************************************/

%ELSE %DO;

        /* Merge character data with metadata to get character variable missing value counts */
        data FinalMetaData;
        retain Name Label Vtype Length Varnum NonMissing NumMissingValues;
```

```
        merge Contents_Vars CharacterVars;
                by Name;
        run;

%END;

*************************************************************************;
* PART 5 - Create the report file                                       *;
*************************************************************************;

ODS EXCEL file="&REPTDIR\&DSNAME Data Set Characterization Report &SYSDATE..xlsx";

ods EXCEL options(sheet_name="SAS Data Set Characteristics");

proc print noobs data=FinalMetaData label;
label   vtype = "Variable Type"
                NonMissing     = "Non-Missing Values"
                NumMissingValues = "Missing Values"
                ;

%IF &NUMCOUNT NE 0 %THEN %DO; /* Not executed if there are no NUMERIC variables*/

        label STD   = "Standard Deviation";

        format min max mean std 10.2 ;

%END;

run;

ODS Excel close;


%MEND CHARFILE;
```

APPENDIX C – EXECUTE THE SAS DATA SET CHARACTERIZATION UTILITY.SAS

```
/*************************************************************************************************/
/*  Program: Execute SAS Data Set Characterization Utility.sas                                   */
/*                                                                                               */
/*  Author: Michael A. Raithel based on a program and GREAT IDEA by: Bob McConnaughey            */
/*                                                                                               */
/*  Created: 5/1/2020                                                                            */
/*                                                                                               */
/*  Purpose: This is the driver program for the "Execute SAS Data Set Characterization Utility" SAS   */
/*           program that creates a report of detailed variable-level information for a specific SAS  */
/*           data set, including counts of non-missing and missing values; and min, max, mean, and    */
/*           std for numeric variables.                                                          */
/*                                                                                               */
/*  Parameters: Users must specify the following three parameters:                               */
/*                                                                                               */
/*             DIRNAME    - The directory holding the SAS data set that is to be characterized.      */
/*             DSNAME     - The name of the SAS data set to be characterized.                    */
/*             REPTDIR    - The directory where the report will be written.                      */
/*                                                                                               */
/*  Outputs: This program creates a Excel spreadsheet with the same name as the SAS data set with     */
/*           the words "Data Set Characterization Report <DATE REPORT WAS CREATED>" appended to it.   */
/*                                                                                               */
/*  Change Log:                                                                                  */
/*                                                                                               */
/*************************************************************************************************/

options symbolgen mprint mlogic source2 nodate nonumber;

%INCLUDE "<Put your directory address here>\SAS Data Set Characterization Utility.SAS";

%CHARFILE(DIRNAME=,
          DSNAME=,
          REPTDIR=);
```