# Paper 3707-2019
# The Missing Numerical Data Plot

**Robin High, University of Nebraska Medical Center, Omaha, NE**

## ABSTRACT

Missing numerical data commonly exist when working with SAS® data sets. The extent of non-observed data and their relationships to other numerical and categorical data are often not immediately evident by inspection of the data file or with summary statistics of the individual variables. When the number of observations is not too large (e.g., a maximum of a few hundred) a missing data plot can be constructed by assigning rows of the plot to the subjects and columns to the numerical variables. The shade of color of a thin horizontal line indicates the relative magnitude of observed data (e.g., an increasing shade of a blue or gray) or if the value is missing (e.g., a bright red). The distribution of colors helps to visually evaluate the extent and patterns of missing data and provides insight into whether they are missing at random (MAR), a feature necessary to apply data imputation techniques with the SAS procedures MI and MIANALYZE. Each column of the graph is annotated with the number and percent of missing values within the observations.

## INTRODUCTION

This paper describes how to produce a graph and a corresponding table that visualizes the extent and patterns of observed and non-observed (missing) numerical data. The graph resembles the plot produced with the vis_miss() function in R (Tierney, 2019) or the heatmap plot produced with the IML procedure (Wicklin, 2016). Although there is no upper limit on the size of the data set, the practicality of making the graph works best with moderate size data sets (e.g., from 50 to 500 observations and up to 15-20 numerical variables). The numerical variables are broadly defined as continuous; integers or ordinal data are also appropriate with one important restriction defined below. The graphical approach consists of a series of DATA steps and SAS procedures that produce a data set with the ranks of the observed data and are plotted in columns for each variable using short, horizontal lines separated by a small white-space with the SGPLOT procedure. Missing data are plotted with a thin line having a distinctly different color or line type. The data processing steps can be entered into a SAS macro in which the dimensions of the graph should be modified in accordance with the number of observations and selected numerical variables. The primary purpose of the plot is to visually observe the extent of missing numerical data combined with a table of missing data patterns.

## METHOD

The data set to make the missing data graph with the SGPLOT procedure is constructed with a series of DATA steps along with the MI, RANK, FREQ, and FORMAT procedures. The steps in the SAS code include the following tasks:

- Assign the total number of observations to a macro variable (the upper bound on the vertical axis)
- Enter the variable names of numerical data to be evaluated into a macro variable; a variable of interest with no or the fewest missing values should be placed first. This list of names with its specified order is accessed several times in the process. The variable names should be relatively short, easily identifiable acronyms, no longer than 10-15 characters.
- The variable names are recoded as successive integers beginning with 1 in the order placed on the macro variable. The actual names are then assigned to a format to be

printed below the horizontal axis of the graph. Deviations of 0.48 above and below these integers are the plotted values on the horizontal axis; e.g., the variable name coded as 1 will range from 0.52 to 1.48, the variable coded as 2 will range from 1.52 to 2.48, etc.

- Determine the number and percent of missing numerical data with the MI procedure and convert them into labels for each variable to be printed at the top of the graph. Enter the labels into an annotation data set.
- Rank the values of each numerical variable into a pre-specified number of groups (e.g., from 5 to 8) with PROC RANK. The graph requires the number of distinct values for each numerical variable to be greater than or equal than the number of groups (e.g., a variable containing a 5-point Likert scale will not work with 6 or more groups). PROC RANK assigns ranks beginning with 0; in the output file, the ranks are incremented by 1 which allows them to be associated with increasing shades from light to dark of a specified color (e.g., gray or blue). This step could also be accomplished by grouping the data within boundaries set by user-defined quantiles (from PROC UNIVARIATE) or into specific ranges entered into a format to provide an ordinal ranking with unequal group sizes (both situations are not illustrated here)
- Missing values produce ranks which are missing. Assign the code for the ranks of missing data as 1 value larger than the number of color groups used to rank the numerical data
- Sort the data set by the value of the first numerical variable which appears on the macro list. A sequentially ordered variable is then added to the data set in a DATA step, beginning with 1 and increasing by 1 up to the number of observations in the data set (to plot on the vertical axis)
- The SGPLOT procedure accesses this data set of ranks to produce the missing numerical data plot. Choose a color and select from five to eight shades increasing from light to dark from colorbrewer2.org (for the HEX representation of colors from the colorbrewer codes, replace the leading # with CX, where CX specifies it is a color code).
- A color with distinctively different contrast and brightness is selected to represent missing values (e.g., a bright red provides a stark visual contrast with shades of gray or blue).

## PRODUCING THE GRAPH WITH SGPLOT

The graph is made in SGPLOT with two required statements. A STYLEATTRS statement assigns the color codes and patterns of the individual lines. These color codes line up with the increasing ranks of the observations within each variable produced with PROC RANK. For example, the SGPLOT statement to enter eight sequential data color codes (increasing shades of blue from light to dark); the ninth color code which appears at the end is a bright red for the non-observed data:

```
STYLEATTRS DATACONTRASTCOLORS=(CXDEEBF7 CXC6DBEF CX9ECAE1 CX6BAED6
                               CX4292C6 CX2171B5 CX08519C CX08306B CXE31A1C)
              DATALINEPATTERS=(solid    solid    solid    solid
                               solid    solid    solid    solid mediumdash);
```

The datacontrastcolors option begins with the color code for the lightest shade and the penultimate color code is the darkest; both of these codes are be associated with the numerical values of the specified number of ranks in increasing order. The 8 line types for the observed data are specified as solid; the line type for non-observed data with rank coded as 9 has medium spaced dashes. With this STYLEATTRS statement the HIGHLOW

statement accesses the observation number for the vertical axis and the lower and upper deviations from the integer assigned to each variable name for the horizontal axis:

```
HIGHLOW y=obsvno low=xvL high=xvU / group=rank type=line
                                   lineattrs=(thickness=1) ;
```

A method to produce this plot for a black/white printer is to define the color codes on the grayscale:

```
STYLEATTRS DATACONTRASTCOLORS=(Grayf0 Grayd9 Graybd Gray96
                               Gray73 Gray52 Gray25 Gray00 CXE31A1C);
```

In general, a specific color can be chosen with the first two letters of a hex value for eight shades of gray with bright red for missing. For example, the gray-scale color codes are of the form `GRAY**` where the `**` indicate the shade of the gray given as a hexadecimal value between FF (white) and 00 (black). Selecting a gray-scale with different line types will allow one to differentiate the missing values (medium dash) from observed data (solid).

The SCATTER statement also produces the plot where the coordinates for the horizontal axis x=xv is the integer code for the variable name and y=obsvno is the observation number in the sorted data set:

```
SCATTER x=xv y=obsvno / group=rank GROUPORDER=ascending markerattrs=(size=0)
                        xerrorlower=xvL xerrorupper=xvU NOERRORCAPS
                        errorbarattrs=(pattern=solid thickness=1);
```

The SCATTER statement does not access the different line type patterns defined by the STYLEATTRS statement for error bars; both observed and missing data will have solid lines.

The plotted lines for the observed data are solid with a small amount of white space on both ends, primarily to give the appearance of a vertical bar for each variable. The density of the lines can be adjusted by modifying the height of the graph. The variable names are printed on the horizontal axis (xv ranges from 1 to the number of variables and xvL and xvU are the endpoints defined by deviations of .48 above and below xv); the actual variable names are printed with a format produced with the data processing to produce the graph. The ranks define the shades of color of the plotted values, specified by the location on the y axis by the observation number (y=obsvno where obsvno=1 to number of observations) of the variable assigned to the first column, with its values sorted from low to high. The variable placed in the first column should have little or no missing data. The option NOERRORCAPS in the SCATTER statement suppresses the error caps at the ends of the horizontal lines.

The option group=rank assigns the rank of a value within each variable which ranges from 1 to the number of defined shades of color. The STYLEATTRS statement assumes the value of the group=rank specification does not skip an integer, which is true here with ranks produced with PROC RANK. Another method of producing this plot not illustrated here is to define color and line attributes from an external attribute data set. The remaining statements in the SGPLOT procedure are conventional choices for the horizontal and vertical axes and if desired, to revise or add a title statement relevant to the data source.

To be assured the plot is produced with the defined combinations of color and line type attributes, the attribute priority option should be set to none:

```
ODS GRAPHICS ON / attrpriority=none;
```

## EXAMPLE

The missing numerical data plot is most practical for moderate size data sets. The maximum number of variables and observations is limited only by the graph dimensions and its resolution. It may work best with data sets having up to 500 hundred observations and 15-20 numerical variables, depending on graph height and width and also considering the length of variable names. For illustration, the process of constructing the missing numerical data plot uses the fitness data from Example 77.6, FCS for Continuous Variables, from the PROC MI documentation (SAS/STAT® 14.3) ranking the values into 6 groups (see the supplementary file with SAS code). The dataset contains three numerical variables and thirty-one observations. The ranks of the values for each variable are depicted with the shade of color for each line: the lightest shades are the smallest values, the darkest shades have the largest. A column with no missing data, the observation number, was added to display the position of each observation within the dataset. In other applications, a variable with little or no missing data should be entered in the first column, sorted from smallest to largest, which provides visual comparisons with the other columns (try the baseball data example from the supplementary SAS code). At the top of the graph for each variable (column), the number of missing values and percent missing of the total number of observations are printed.

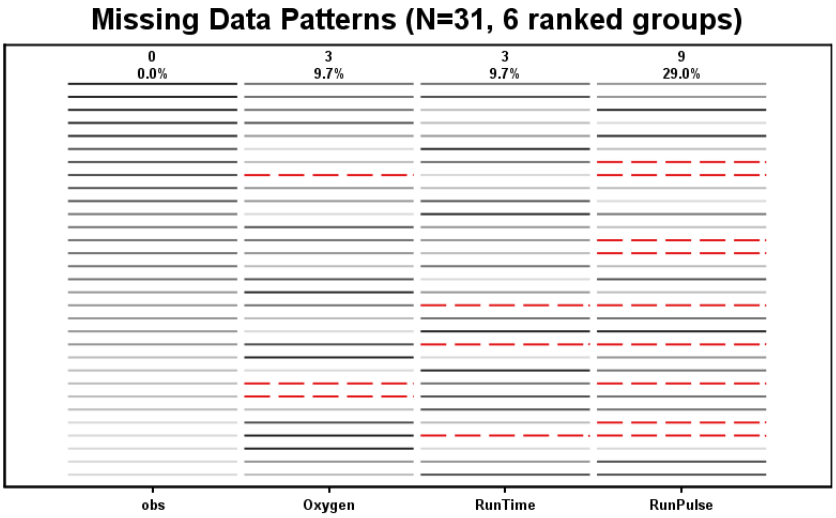


**Figure 1. Missing Data Plot**

One additional feature of the plot is the visual comparisons which can be made across columns; the contrast of light and dark colors across the rows display insight into inherent positive or negative relationships with the variable defined to the first column.

Other relationships can be illustrated by switching the order of the variable names entered on the macro variable. If the data set has complete data for all variables, relationships among the variables can still be examined; just add a variable with randomly generated numbers at the far right side with one value set to missing.

The graph can also be interpreted with the aid of a table of summary statistics:

```
PROC TABULATE DATA=fitness1 NOseps;
VAR Oxygen RunTime RunPulse;
TABLE Oxygen RunTime RunPulse,
      (n nmiss)*f=5.0 (min q1 median q3 max)*f=6.1 / rts=12;
RUN;
```

| | N | NMiss | Min | Q1 | Median | Q3 | Max |
|---|---|---|---|---|---|---|---|
| Oxygen | 28 | 3 | 37.4 | 44.8 | 46.7 | 50.1 | 60.1 |
| RunTime | 28 | 3 | 8.6 | 9.8 | 10.5 | 11.4 | 14.0 |
| RunPulse | 22 | 9 | 148.0 | 168.0 | 172.0 | 178.0 | 186.0 |

An extension of the plot can also be produced defined by two to four levels of a categorical variable with unequal numbers of observations within each level. The data processing requirements and plotting process are more complex and require the use of an external attribute data set to define the colors and types of the individual lines within the levels of the categorical variable (the STYLEATTRS statement usually does not work in this situation).

## MISSING DATA PATTERNS

Missing data patterns refer to the configuration of observed and non-observed values in a data set; they simply describe the location of missing values in the data without inferring the mechanisms that generate missing data. There are up to six types of patterns that can be observed (Enders, pp. 4-6). The plot displays general and monotone patterns of missing data (the latter depends on the order the variables are defined on the horizontal axis). However, the number and types of patterns for non-observed values across the selected variables are usually not easily discernable from the plot itself. The IML procedure has the CMISS function which will produce a matrix of 0/1 values to show patterns of missing data for each row (Wicklin, 2016). PROC MI produces a table of missing data patterns along with the number of observations and percent of the total having each pattern (variables with observed values are identified with an X). The MI procedure assumes that at least one numerical variable has at least one missing value; otherwise, no output is produced.

```
PROC MI data=fitness1 nimpute=0 simple;
VAR obs Oxygen RunTime RunPulse;
RUN;
```

For the example data set, the missing data patterns produced with the MI procedure is:

| Group | obs | Oxygen | RunTime | RunPulse | Freq | Percent |
|---|---|---|---|---|---|---|
| 1 | X | X | X | X | 21 | 67.74 |
| 2 | X | X | X | . | 4 | 12.90 |
| 3 | X | X | . | . | 3 | 9.68 |
| 4 | X | . | X | X | 1 | 3.23 |
| 5 | X | . | X | . | 2 | 6.45 |

The data processing steps to make the missing data plot also produces missing patterns which can be displayed with PROC TABULATE:

```
Missing Data Patterns
1=obs 2=oxygen 3=runtime 4=runpulse
```

| m=Missing | | N | Percent |
|---|---|---|---|
| group | 1234 | | |
| 1 | xxxx | 21 | 67.7 |
| 2 | xxx. | 4 | 12.9 |
| 3 | xx.. | 3 | 9.7 |
| 4 | x.x. | 2 | 6.5 |
| 5 | x.xx | 1 | 3.2 |
| Total | | 31 | 100.0 |

A DATA step assigns a character value with a format for each observation that writes the pattern of existing and missing data. The macro variable `&nrank` contains the number of ranks:

```
PROC FORMAT;
VALUE mssp 1-&nrank. ='x' %EVAL(&nrank. + 1) ='.';
RUN;
```

A DO loop in a DATA step writes the pattern of observed/missing data for each observation defined with an x for observed or . for missing with each of the four variables, the ranks `(_i1-_i4)` produced with PROC RANK are stored in an ARRAY referenced as `gp{i}`:

```
DATA _1b; SET _1b;
DROP i ;
LENGTH pttrn $4;
ARRAY gp{4} _i1 - _i4 ;
DO i = 1 to 4;
  gp{i} = gp{i} + 1; * ranks need to start at 1;
  IF gp{i} = . then gp{i} = &nrank. + 1; * value to display missing data;
END;
DO i = 1 to 4 ;
  pttrn= cats(pttrn,put(gp{i},mssp.));  * missing data pattern;
END;
RUN;
```

The same missing data patterns as MI are produced (which may appear in a different order) with numerical summaries (counts and percents) are printed with PROC TABULATE in a more compact layout; the sequential numbers at the top of the second column refer to the order of the variable names printed in the title above the table.

The complete code for producing this graph and the table of missing data patterns is provided in the supplementary SAS file.

## SUMMARY

Visualizing the extent and presence of observed and non-observed data distributed throughout a data set is an important preliminary step in data analysis. The visual patterns of missing values compliments the table of summary statistics. Fortunately, from a practical view, if the missing data patterns can be distinguished as arbitrary, maximum likelihood estimation and multiple imputation techniques work well (Enders, p. 5).

## REFERENCES

Berglund, Patricia and Heeringa, Steven (2014) Multiple Imputation of Missing Data Using SAS®. Cary, NC: SAS Institute Inc.

Enders, Craig K. (2010) Applied Missing Data Analysis. The Guilford Press: New York.

Tierney, Nicholas. (2019) "Gallery of Missing Data Visualisations" accessed Feb. 25, 2019, https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html

The MI Procedure, Example 77.6 FCS Methods for Continuous Variables, accessed March 5, 2019, https://documentation.sas.com/?docsetId=statug&docsetTarget=statug_mi_examples06.htm&docsetVersion=14.3&locale=en

Wicklin, Rick, (2016) "Examine patterns of missing data in SAS" "DO Loop" article accessed Feb. 26, 2019, https://blogs.sas.com/content/iml/2016/04/18/patterns-of-missing-data-in-sas.html.

Wicklin, Rick, (2016) "Visualize missing data in SAS" "DO Loop" article accessed Feb. 26, 2019, https://blogs.sas.com/content/iml/2016/04/20/visualize-missing-data-sas.html.

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

Robin High
Department of Biostatistics
College of Public Health
University of Nebraska Medical Center
984375 Nebraska Medical Center
Omaha, NE 68198-4375
email: rhigh@unmc.edu

Using the sashelp.baseball data set, the following missing numerical data plot was produced: