Paper number: PH03

Data Quality Review for Missing Values and Outliers

Ying Guo, i3, Indianapolis, IN
Bradford J. Danner, i3, Lincoln, NE

ABSTRACT

Before performing any analysis on a dataset, it is often useful to perform a review of the dataset in order to detect missing values and outliers. The macro provided in this paper is a SAS® macro program designed to check the data in a time-efficient and user-friendly way, which can generate the following content: First, the macro can generate an Excel report to determine whether the variables have missing values or outliers, and report the percentage of missing values. Second, the macro can display all records which contain missing values and/or outliers, grouped by variables in an Excel file. Each tab will represent the exception records for one variable. This macro can automatically check all variables regardless of the dataset structure and variable names. Also, an extended version of this macro can perform an automatic check of a whole SAS data library at one time.

THE PROBLEM

Missing values and outliers create difficulties for most analyses. Without knowing the structure and details of missing data and outliers, problems may arise when analysis is performed, such as undetected outliers which may bias an analysis if these outliers are due to previously undetected errors. Therefore, when receiving datasets with dozens (even hundreds) of variables and thousands of observations, knowing the summary of missing values and outliers can be helpful to identify any upstream problems that may have crept into the data.

Currently, a common method for this is to use PROC MEANS, PROC UNIVARIATE, and PROC FREQ for each variable to detect the missing values and outliers. If it is necessary, datasets are created to indicate which records have missing values and/or outliers. Performing checks for each variable, going through output for each procedure, and generating datasets for each variable which have missing values and/or outliers is quite time consuming, especially with large numbers of huge datasets. Individual variable names need to be typed in for each dataset to find missing values and/or outliers and output them to exception datasets. Additionally such programs are not reusable due to different structures and variable names for different studies. This can be very time-consuming.

THE SOLUTION

In order to expedite review of the datasets and detect missing values and outliers in a more time-efficient way, a SAS macro was developed. This macro will automatically process check each variable in a dataset for missing values and outliers. When missing values and/or outliers are found, the macro will automatically generate an exception dataset which will contain the records with missing values and/or outliers. The exception dataset will be output to an Excel file. Each tab will represent an exception data for one variable. There is no need to type in variable names for each dataset; all that is required is to specify the name and the location of the input dataset and the output Excel file. In this macro, an outlier was defined as a value outside the range for mean ± three standard deviations (SD). In order to eliminate the effect of outliers to the mean, a trimmed mean and SD was used, which cuts off certain percentage of both minimum and maximum data before calculating the mean and SD. The reason is that when there are not enough observations in a dataset, the outliers can significantly affect the mean and SD, thus keeping the outliers from being detected. Automatic checks can be performed on a whole data library without specifying each dataset in the library by name with the extended version of the macro. The results for each dataset are generated individually.

THE OUTPUT OF THE MACRO

First, the macro produces a summary output. The report will indicate if the variables have missing value or outliers. The percentage of missing value is also reported.

Example 1: The dataset named CCC have five variables, name1, name2, number1, number2, number3, while name1 and name2 are character variables, and number1 number2 and number3 are numerical variables.

dataset name	variable name	type	check	check label
TEST	name1	2	n	Non missing value found
TEST	name2	2	m	missing value found 20 (20.202020202%)
TEST	number1	1	o	outlier found
TEST	number2	1	m	Missing value found 7 (7.07%)
TEST	number3	1	m	Missing value found 7 (7.07%)
TEST	number3	1	o	outlier found

Secondly, the macro can generate all records which contain missing values (**Example 2**) and outliers (**Example 3**), grouped by variable in the Excel file. In the Excel file, each variable will create a new tab. For character variables, only missing values are reported. For numerical variables, both missing values and outliers are reported.

Example 2: Missing values shown for name2, number2, and number3 variables. **Example 2a** represents all records with the variable name2 missing, **Example 2b** represents all records with the variable number2 missing, and **Example 2c** represents all records with the variable number3 missing.

Example 2a

name1	name2	number1	number2	number3
ccc		91	73	
ccc		92		62
ccc		1000	75	63
bbb		60	10	34
bbb		2	96	84
eee		87	60	57
eee		88	61	58
eee		39	38	20
bbb		84		30
bbb		590	62	40
bbb		28	39	9
ddd		29		10
ddd		30	41	11
ddd		32	58	13
ddd		0.001	59	0.002
ddd		37	25	18
ddd		73	84	
bbb		61	11	35
bbb		62	12	36
bbb		1	95	83

Example 2b

name1	name2	number1	number2	number3
bbb	eee	80		26
ccc		92		62
ddd	ccc	53		65
bbb		84		30
ddd		29		10
ggg	bbb	21		72
ggg	bbb	99		99

Example 2c

name1	name2	number1	number2	number3
bbb	ddd	14	65	
ccc		91	73	
bbb	ccc	45	7	
bbb	ddd	86	15	
bbb	ddd	11	63	
ddd		73	84	
ggg	bbb	71	52	

Example 3: Outliers are displayed as follows: In Example 3a, the observations which have outliers in variable number1 are shown. In Example 3b, the observations which have outliers in variable number3 are shown.

Example 3a

name1	name2	number1	number2	number3
ccc		1000	75	63
bbb		590	62	40

Example 3b

name1	name2	number1	number2	number3
fff	ddd	55	28	3500
ddd	ggg	76	87	1000

Example 4: In the Excel file, it also gives a summary.

MEMNAME	NAME	TYPE	LABEL	check	checkl
TEST	name1	2	name1	n	Non missing value found
TEST	name2	2	name2	m	missing value found 20 (20.20%)
TEST	number1	1	number1	o	outlier found
TEST	number2	1	number2	m	Missing value found 7 (7.07%)
TEST	number3	1	number3	m	Missing value found 7 (7.07%)
TEST	number3	1	number3	o	outlier found

SUMMARY

In summary, this macro is a very useful tool to check data efficiently. It can check all data values automatically, regardless of different structures and names for datasets; therefore, checks can be performed in a time-efficient manner. The macro can find potential data issues detected in input datasets, and report these issues properly to the data management team, enabling them to keep the data clean and correct. Finally, the macro can check multiple datasets from the same library. This special feature is especially helpful for people who work on data integration or anything dealing with multiple datasets at the same time.

CONCLUSION

This macro is simple and practical. It will be a very useful tool for people who want to review and understand data quickly and correctly.

ACKNOWLEDGMENTS

We would like to thank my collaborator Lixiang Liu in InVentiv Clinical solutions who validated this macro. His comments are really appreciated.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please contact the primary author at:

Ying Guo
I3
Indianapolis, IN 46285
Email: evelyn.guo@i3statprobe.com

APPENDIX

```
/******User guide******
xlsfile: Input the path and excel file name which the missing and outlier is output to.
indata:  Input the data name which you want to check.
outfile: Input the path and file name for rtf report.
trim:    Input the percentage you want to trim when calculate mean and SD.
************************************************************/;


/********** Check for one dataset**********/
ods listing close;
%macro onedata (xlsfile=, indata=, outfile=, trim=);

/*Create content table*/
  proc contents data=&indata
    out=tmp(keep=memname name type label);
  run;

/*Check frequency for Character variable, mean for numerical variable*/

  /*Get the number of character and numerical variable*/
  proc freq data=tmp;
    table type;
    ods output  Freq.Table1.OneWayFreqs=tmp1(keep=type frequency);
  run;

  data _null_;
    set tmp1;
    call symput('type'||'_'||trim(left(type)),frequency);
  run;
%put &type_1 &type_2;

  /*Check frequency for Character variable*/
    data tmpc;
      set tmp;
      if type=2;
      output;
    run;

    data _null_;
      set tmpc;
      call symput('char'||'_'||trim(left(_n_)),trim(left(name)));
    run;

  %let i=1;
  %macro loopcha;
    %do i=1 %to &type_2;

      proc freq data=&indata;
        table &&char_&i/missing;
        ods output onewayfreqs=&&char_&i(keep=&&char_&i frequency percent);
      run;

      data &&char_&i;
        set &&char_&i;
           if &&char_&i=' ' and frequency ne 0 then
          do;
              name="&&char_&i";
              check='m';
              checkl="missing value found"||" "||trim(left(frequency))||"
("||trim(left(percent))||"%)";
```

```
        end;
    run;

    proc sort data=&&char_&i nodupkey;
      by name check;
    run;

    data tmpc;
      merge tmpc(in=a) &&char_&i(keep=name check checkl);
      by name;
      if a;
         if check = ' ' then do;
         check='n'; checkl="Non missing value found";
         end;
    run;

  %end;
%mend loopcha;
%loopcha

    /*output missing values for character variable*/
libname lib "&xlsfile";
data tmpcm;
  set tmpc;
  where check='m';
  call symput('varc'||'_'||trim(left(_n_)),trim(left(name)));
run;

proc sql noprint;
  select count(*) into: n from tmpcm;
quit;

%macro missingvalue;
  %do i=1 %to &n;
   data lib.&&varc_&i;
     set &indata;
     where &&varc_&i = ' ';
   run;
  %end;
%mend missingvalue;
%missingvalue;
Libname lib clear; run;

  /*Check outliner and missing value for numerical variable*/
  data tmpn;
    set tmp;
    if type=1;
    output;
  run;

  data _null_;
    set tmpn;
    call symput('num'||'_'||trim(left(_n_)),trim(left(name)));
  run;

 %let i=1;
 %macro loopnum;
   %do i=1 %to &type_1;

     proc means data=&indata n min max nmiss;
       var &&num_&i;
       ods output means.summary=temp;
     run;
```

```
      proc univariate data=&indata trimmed=&trim;
        var &&num_&i;
        ods output TrimmedMeans=&&num_&i;
      run;

      data &&num_&i;
        merge &&num_&i temp;
      run;

      data &&num_&i;
        set &&num_&i;
        per=&&num_&i.._nmiss/(&&num_&i.._n+&&num_&i.._nmiss);
        if &&num_&i.._nmiss ne 0 then
        do;
          name="&&num_&i";
             check='m';
             checkl="Missing value found"||" "||trim(left(&&num_&i.._nmiss))||"
("||trim(left(round(per, 0.0001)*100))||"%)";
             output;
        end;

           if . < &&num_&i.._min < (mean - 3*stdmean*sqrt(&&num_&i.._n-1)) or
            &&num_&i.._max > (mean + 3*stdmean*sqrt(&&num_&i.._n-1)) then
        do;
                name="&&num_&i";
                check='o';
                checkl="outlier found";
                output;
        end;
           else output;
      run;

      proc sort data=&&num_&i nodupkey;
        by name checkl;
      run;

      data tmpn;
        merge tmpn(in=a) &&num_&i(keep=name check checkl);
        by name;
        if a;
           if check = ' ' then do;
          check='n'; checkl="Non missing or outlier found";
          end;
      run;

   %end;
   %mend loopnum;
   %loopnum

         /*output missing values for numerical variable*/
  libname lib "&xlsfile";
    data tmpnm;
    set tmpn;
    where check='m';
    call symput('varnm'||'_'||trim(left(_n_)),trim(left(name)));
  run;

  proc sql noprint;
    select count(*) into: n from tmpnm;
  quit;

  %macro missingvalue;
    %do i=1 %to &n;
     data lib.m_&&varnm_&i;
```

```
     set &indata;
     where &&varnm_&i = .;
   run;
  %end;
%mend missingvalue;
%missingvalue;
Libname lib clear; run;

         /*output outlier values for numerical variable*/
libname lib "&xlsfile";
data tmpno;
  set tmpn;
  where check='o';
  call symput('varno'||'_'||trim(left(_n_)),trim(left(name)));
run;

proc sql noprint;
  select count(*) into: n from tmpno;
quit;

data &indata;
  set &indata;
  id=1;
run;

%macro outlier;
  %do i=1 %to &n;

  ods listing close;
  proc univariate data=&indata trimmed=&trim;
    var &&varno_&i;
    ods output TrimmedMeans=&&varno_&i (keep=mean stdmean);
  run;

  proc means data=&indata n ;
     var &&varno_&i;
     ods output means.summary=temp(rename=(&&varno_&i._n=n));
  run;
  ods listing;

  data &&varno_&i;
    merge &&varno_&i temp;
  run;

  data &&varno_&i;
    set &&varno_&i;
    id=1;
  run;

  data &indata;
    merge &indata &&varno_&i;
    by id;
  run;

 data lib.o_&&varno_&i(drop=mean stdmean n id);
   set &indata;
   if . < &&varno_&i < (mean - 3*stdmean*sqrt(n-1)) or
          &&varno_&i > (mean + 3*stdmean*sqrt(n-1)) then
   output;
 run;

 data &indata;
   set &indata (drop=mean stdmean n);
 run;
```

```
  %end;
  %mend outlier;
  %outlier


ods listing;
data final;
  merge tmp tmpc tmpn;
  by memname name;
run;

data lib.summary;
  set final;
run;
Libname lib clear; run;

filename outfile "&outfile";

ods rtf file=outfile;

proc report data=final nowindows headline headskip;
   column memname name type check checkl;
   define memname  / width=15 left "dataset name" flow id;
   define name     / width=10 left "variable name";
   define type     / width=6  left "type" ;
   define check    / width=6  left "check";
   define checkl   / width=30 left "check label";

   compute after ;
   ;
   endcomp;
run;
ods rtf close;
%mend onedata;

/*Example macro call*/;
/*%onedata(xlsfile=H:\lixliu\UAT\test1.xls, indata=aaa, outfile='H:\lixliu\UAT\test2.rtf',
trim=0.2)*/

*****************Check for whole library*************;
/*Here is an example, libname was define on my local computer.*/;
libname gwcv 'Y:\BYETTA._G\GWCV\GLS';

ods output members=gwcvmem;
proc datasets mt=data library=gwcv;
run;
quit;

proc copy out=work in=gwcv;
run;

proc sql noprint;
   select count(*) into:num from gwcvmem;
quit;
%put &num;

data _null_;
   set gwcvmem;
   call symput('name'||'_'||trim(left(_n_)),trim(left(name)));
run;

%let j=1;
%macro loop;
   %do j=1 %to &num;
```

```
   %onedata(indata=&&name_&j, outfile=&&name_&j..final.rtf trim=0.2)
   %end;
%mend loop;
%loop
```