# Handling missing values in Analysis

Before we analyze the data, which includes missing values, we should make sure that all the missing values have been coded as SAS missing values. There are many ways to code missing data in SAS. The mostly used is

Missing numeric data: a single period (.);
Missing character data: a single period (.) or a blank space.

Please refer to our document of "Handling missing values in your data".

## TESTING FOR MISSING VALUES & GETTING THE NUMBER OF MISSING VARIABLE

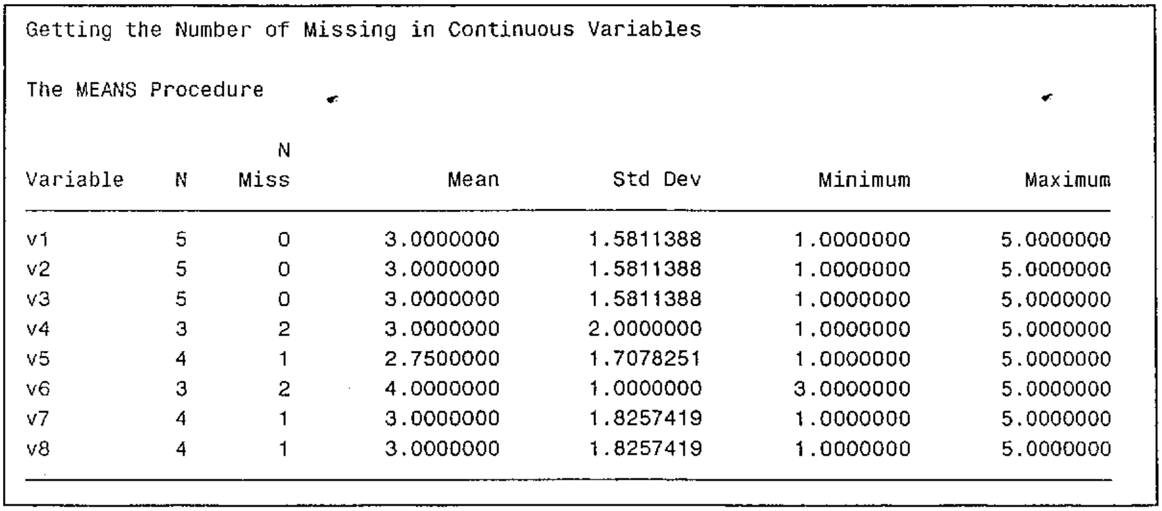
1. Continuous Variable

We can use the option of NMISS in PROC MEANS to get the number of missing values.

```
data raw;
    input v1-v9 v10 $;
cards;
1 1 1 1 1 . 1 1 1 a
2 2 2 . 2 . 2 2 0 b
3 3 3 3 3 3 . . . b
4 4 4 . . 4 4 4 0 a
5 5 5 5 5 5 5 5 . .
;
proc means data=raw n nmiss mean std min max;
      var v1-v8;
run;
```

"$" indicates that v10 is a character variable.

The SAS output of the proceeding code is as following,



```
Getting the Number of Missing in Continuous Variables

The MEANS Procedure

                     N
Variable     N    Miss          Mean       Std Dev       Minimum       Maximum
------------------------------------------------------------------------------
v1           5       0     3.0000000     1.5811388     1.0000000     5.0000000
v2           5       0     3.0000000     1.5811388     1.0000000     5.0000000
v3           5       0     3.0000000     1.5811388     1.0000000     5.0000000
v4           3       2     3.0000000     2.0000000     1.0000000     5.0000000
v5           4       1     2.7500000     1.7078251     1.0000000     5.0000000
v6           3       2     4.0000000     1.0000000     3.0000000     5.0000000
v7           4       1     3.0000000     1.8257419     1.0000000     5.0000000
v8           4       1     3.0000000     1.8257419     1.0000000     5.0000000
------------------------------------------------------------------------------
```

n

gives the total number of observations, including missing values.

nmiss
gives the number of missing values.

mean
gives the mean value of the variable. Missing values are not included in computation.

std
gives the standard deviation of the variable. Missing values are not included in computation.

min, max
gives the minimum and maximum of the variable.

These options are helpful in checking for missing values, because impossible numbers that lie outside the relevant range, such as -99 or -9, often represent missing data. If there is an odd value in the minimum or maximum field, it is a flag that there is a missing value in that variable. We should convert it to a SAS missing value before any analysis.

2. Categorical Variable (or character data)

PROC FREQ gives frequency table for categorical variables, which, by default, does not include missing value as a level of category.

```
proc freq data=raw;
      title "Getting the Number of Missing in Categorical Variables 1";
      table v10 v9*v10;
run;
```

The SAS output of the proceeding code is in Table 1. The number of missing is indicated by "Frequency Missing=". It is fine for one-way tables. But for two-way tables, we have no idea where the 2 missing values are from. Are they missing in v9, or v10 or both?

The option of MISSING in TABLE statement helps to locate the missing values.

```
proc freq data=raw;
      title "Getting the Number of Missing in Categorical Variables 2";
      table v9*v10 /missing nocol norow nopercent;
run;
```

missing
treat missing values as a category

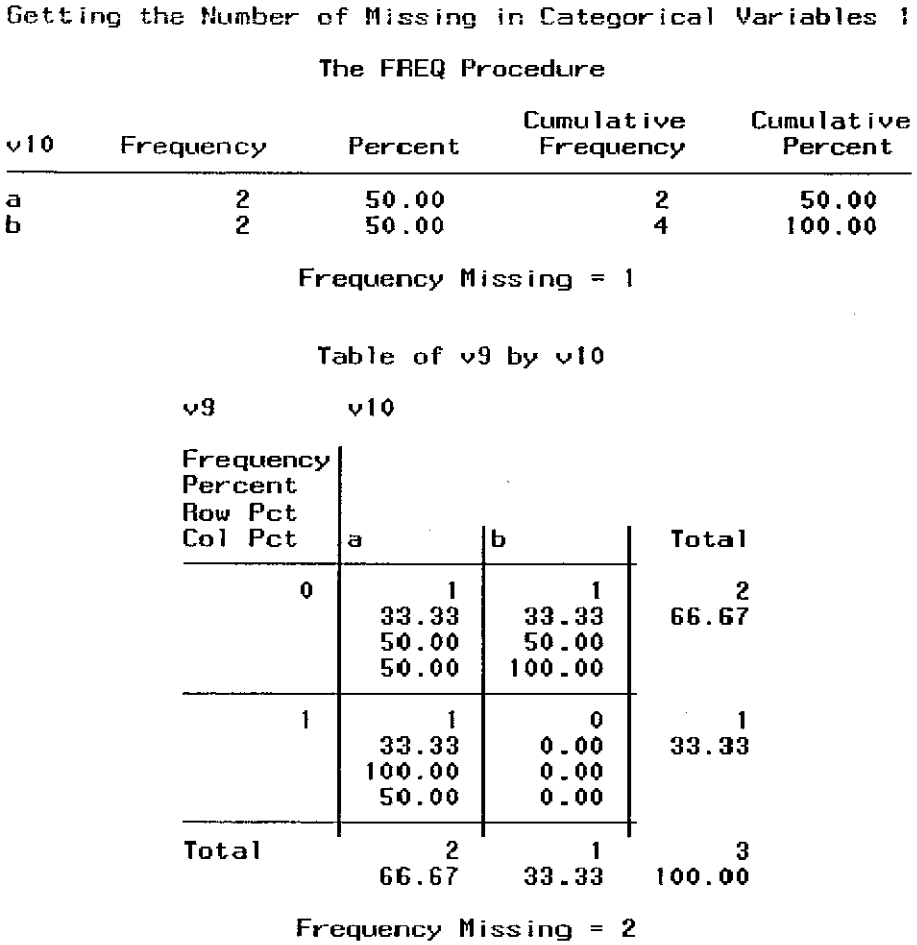
nocol, norow, nopercent
tell SAS not to show the row percentage, column percentage and overall percentage.

The SAS output is shown in Table 2. Now, we are treating missing values as a category. We know that there are two missing values in v9, and one of them is also missing in v10.

Note:

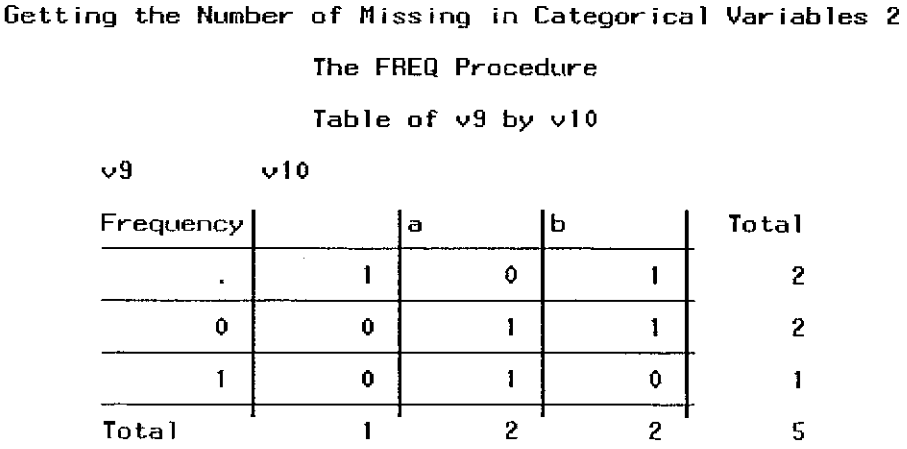
In doing chi-squared test, without "missing" option, the procedure excludes all missing values in the test. But with the "missing" option, it treats missing value as an additional category.

Table 1.



```
Getting the Number of Missing in Categorical Variables 1

                       The FREQ Procedure

                                        Cumulative    Cumulative
v10     Frequency      Percent        Frequency       Percent
--------------------------------------------------------------------
a              2         50.00                2         50.00
b              2         50.00                4        100.00

                      Frequency Missing = 1


                       Table of v9 by v10

              v9          v10

              Frequency|
              Percent  |
              Row Pct  |
              Col Pct  |a       |b       |  Total
              ---------+--------+--------+
                     0 |      1 |      1 |      2
                       |  33.33 |  33.33 |  66.67
                       |  50.00 |  50.00 |
                       |  50.00 | 100.00 |
              ---------+--------+--------+
                     1 |      1 |      0 |      1
                       |  33.33 |   0.00 |  33.33
                       | 100.00 |   0.00 |
                       |  50.00 |   0.00 |
              ---------+--------+--------+
              Total           2        1        3
                          66.67    33.33   100.00

                      Frequency Missing = 2
```

Table 2.



```
Getting the Number of Missing in Categorical Variables 2

                       The FREQ Procedure

                       Table of v9 by v10

          v9          v10

          Frequency|        |a       |b       |  Total
          ---------+--------+--------+--------+
                 . |      1 |      0 |      1 |      2
          ---------+--------+--------+--------+
                 0 |      0 |      1 |      1 |      2
          ---------+--------+--------+--------+
                 1 |      0 |      1 |      0 |      1
          ---------+--------+--------+--------+
          Total           1        2        2        5
```

## SUMMING VARIABLES WITH MISSING DATA

1. Direct adding

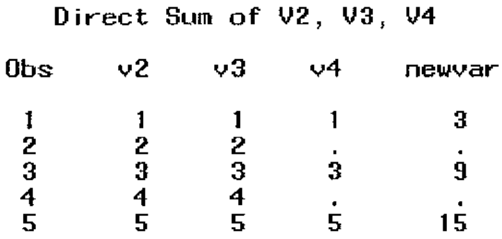
```
newvar=v2+v3+v4;
```

With this method, if any of v2, v3 and v4 is missing, the new variable, newvar, would be missing.

The same rule applies to the operation of +, -, ×, ÷.

```
data sum;
   set raw;
   newvar=v2+v3+v4;
proc print data=sum;
   title "Direct Sum of V2, V3, V4";
   var v2-v4 newvar;
run;
```

SAS output:



```
            Direct Sum of V2, V3, V4

    Obs      v2      v3      v4      newvar

     1        1       1       1         3
     2        2       2       .         .
     3        3       3       3         9
     4        4       4       .         .
     5        5       5       5        15
```

2. SUM function

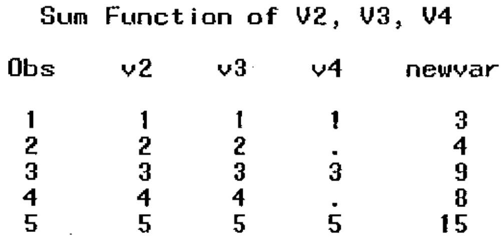
The advantage of this method is that the syntax is much less laborious to type, especially for large numbers of variables.

```
newvar=sum(of v2-v4);
newvar=sum(of v2 v3 v4);
```

Unfortunately, with this method any variable to be summed which has a missing value is treated as zero by SAS.

```
data sum;
   set raw;
   newvar=sum(of v2-v4);
proc print data=sum;
   title "Sum Function of V2, V3, V4";
   var v2-v4 newvar;
run;
```

SAS output:



```
              Sum Function of V2, V3, V4

         Obs      v2      v3      v4     newvar

          1        1       1       1        3
          2        2       2       .        4
          3        3       3       3        9
          4        4       4       .        8
          5        5       5       5       15
```

If you have both a large number of variables to sum and missing data, what can you do? One solution (provided by Karl Wuensch over the Internet) is to use the **NMISS (OF** function in conjunction with the **SUM (OF** function.

```
if nmiss(of v2-v4) > 0 then newvar = . ;
else newvar = sum(of v2-v4);
```

nmiss(of v2-v4)

calculates the number of missing values across the variables v2 through v4.

If SAS finds any missing data, it sets the value of newvar to be missing. Otherwise, the value of newvar is set to be the sum of the Oldvar1 through Oldvar3 values which have non-missing cases.

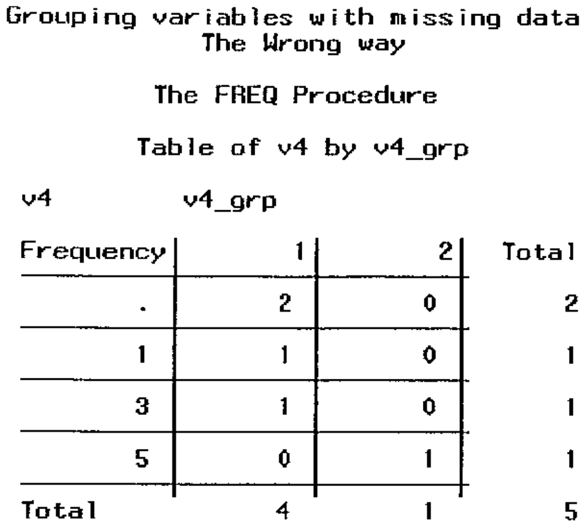
## Grouping Variables with Missing Data

SAS treats any missing value as SMALLER than any non-missing value, i.e. missing is smaller than 0, is smaller than -999, is smaller than any number. So, when we categorize variables, we shall first take care of those missing values.

We want to categorize variable v4 into two levels so that 1 to 3 are level 1, 4 to 5 are level 2.
The following code is WRONG. It groups the missing value into level 1.

```
data grpv4;
      set raw;
      if v4<=3 then v4_grp=1;
      else v4_grp=2;
proc freq data=grpv4;
      title "Grouping variables with missing data";
      title2 "The Wrong way";
      table v4 * v4_grp / missing nocol norow nopercent;
run;
```

SAS output:

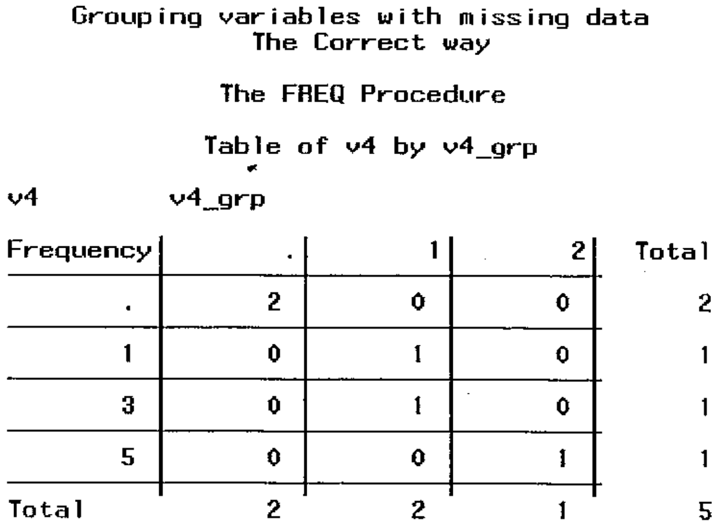


```
                Grouping variables with missing data
                          The Wrong way

                       The FREQ Procedure

                      Table of v4 by v4_grp

v4            v4_grp

Frequency|       1|       2|  Total
---------+--------+--------+
       . |      2 |      0 |      2
---------+--------+--------+
       1 |      1 |      0 |      1
---------+--------+--------+
       3 |      1 |      0 |      1
---------+--------+--------+
       5 |      0 |      1 |      1
---------+--------+--------+
Total           4        1        5
```

The CORRECT way to do it is

```
data grpv4;
      set raw;
      if v4=. then v4_grp=.;
      else if v4<=3 then v4_grp=1;
      else v4_grp=2;
proc freq data=grpv4;
      title "Grouping variables with missing data";
      title2 "The Correct way";
      table v4 * v4_grp / missing nocol norow nopercent;
run;
```

SAS output:



```
                Grouping variables with missing data
                          The Correct way

                       The FREQ Procedure

                      Table of v4 by v4_grp

v4            v4_grp

Frequency|       .|       1|       2|  Total
---------+--------+--------+--------+
       . |      2 |      0 |      0 |      2
---------+--------+--------+--------+
       1 |      0 |      1 |      0 |      1
---------+--------+--------+--------+
       3 |      0 |      1 |      0 |      1
---------+--------+--------+--------+
       5 |      0 |      0 |      1 |      1
---------+--------+--------+--------+
Total           2        2        1        5
```