Paper 3719-2019

Does the Cat Have Your Tongue? Using the VNAME and CATX Functions to Get SAS® to Speak to You

Gayle Springer, Johns Hopkins Bloomberg School of Public Health

ABSTRACT

In our longitudinal health study, when certain health events occur, we restrict access to biological specimens from that time period. The reason or reasons for restricted access are maintained in a data table. Our challenge is to provide the reasons for restriction at the individual and population level for reporting purposes. When biological specimens are requested for use with a project, we must review the ID/visits requested against the list of restricted specimens. Then, the restricted ID/visits, along with the reasons, are provided to an oversight committee to determine whether the restricted specimens can be used for the project. We developed an easy method for creating the reason list in words by creating variable names that are succinct and descriptive. Using an ARRAY statement and the CATX and VNAME functions, we quickly compile the reason list to give to our oversight committee to aid in their decision. We also show how to make minor changes to this method to create a similar list when variables are not descriptively named.

INTRODUCTION

In the perfect world, obtaining information from your database is easy to do. The REPORT, FREQ, and TABULATE procedures may summarize the data. When the information needed is the actual variable name, SAS® has the VNAME function to capture that information.

When a group of related variables that do not contain mutually exclusive information has information one needs, such as a list of diseases or prescription medications, an ARRAY statement, the CATX function, and the VNAME function may be used together to create lists of information.

WHAT IS VNAME?

SAS has several functions that will return information about a variable. The VNAME function is used in the DATA step and will return the variable name. The default length of the returned value is 200 characters. Use the VNAME function when the variable name is the data we are interested in obtaining instead of the value of the variable. Below is the SAS syntax for VNAME, where variable is the name of the variable to be returned:

```
VNAME(variable)
```

A SIMPLE VNAME EXAMPLE

To assign the name of a variable to another variable, the following syntax is used in a DATA step. By using a conditional statement, the variable name is captured only when the condition is met. The LENGTH statement assigns the length of the variable:

```
LENGTH reason $8.;
IF cancer = 1 THEN reason=VNAME(cancer);
ELSE IF stroke = 1 THEN reason=VNAME(stroke);
```

Table 1 shows output from the above SAS statements.

Obs	CANCER	STROKE	REASON
1	0	1	STROKE
2	1	0	CANCER
3	1	0	CANCER
4	0	1	STROKE

Table 1. VNAME Output

THE CATX FUNCTION

The CATX function is used in the DATA step and will concatenate information together while applying a user-defined delimiter. The syntax for CATX is:

```
CATX (delimiter, item-1 <, … item-n>)
```

where delimiter is a character (string) to be inserted between items; item is any variable or expression to be concatenated together. Output from CATX is character. The default length of the returned CATX value ranges from 200 to 65,534 characters. Always use a LENGTH statement when using CATX.

The following code shows an example of building a comma-delimited list using CATX:

```
LENGTH fullname $25.;
fullname = CATX(",", lname, fname);
```

USING AN ARRAY TO RETURN A LIST OF VARIABLE NAMES

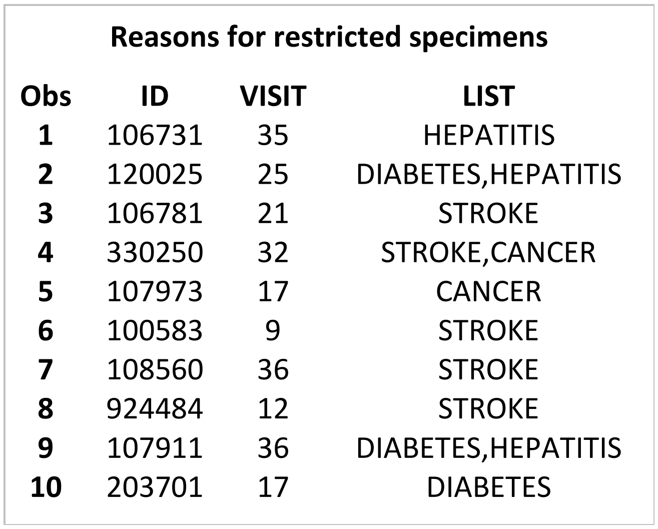
For our project, we have several binary variables that provide reasons for restricted use of a participant’s specimens. All variables flagged as yes (where the binary variable = 1) need to be added to the reason list for restriction.

We can use an ARRAY to list all the pertinent variables and conditional statements to concatenate the information together.

In the SAS code below, the LENGTH statement assigns a length of 50 characters to the variable named LIST. The ARRAY is named REASON while the variables in this statement are descriptively named numeric variables. The DO loop will run through all items listed in the ARRAY REASON. The conditional IF statement checks if the array item is equal to one; if so, the CATX function is used to append, with a comma, the name of the variable onto variable LIST:

```
DATA reasons;
  SET restricted;
  LENGTH list $50.;
  ARRAY reason (*) hivsc stroke diabetes hepatitis cancer death;
  DO I = 1 TO DIM(reason);
    IF REASON(I) = 1 THEN
      list = CATX(",", list, VNAME(reason(I)));
  END;
RUN;
PROC PRINT DATA=reasons; VAR id visit list;
  TITLE "Reasons for restricted specimens";
RUN;
```

The output from the PRINT procedure is below. As specified in our call to function CATX, variable names are separated by a comma.



Reasons for restricted specimens

Obs	ID	VISIT	LIST
1	106731	35	HEPATITIS
2	120025	25	DIABETES,HEPATITIS
3	106781	21	STROKE
4	330250	32	STROKE,CANCER
5	107973	17	CANCER
6	100583	9	STROKE
7	108560	36	STROKE
8	924484	12	STROKE
9	107911	36	DIABETES,HEPATITIS
10	203701	17	DIABETES

Table 2: PROC PRINT Output from CATX and VNAME

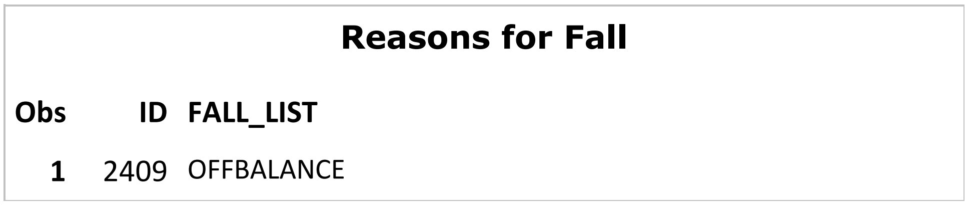
MY VARIABLES ARE NOT NAMED PROPERLY

Sometimes variable names are not as informative as in the example listed above. If this is the case, similar information may be gathered using two ARRAY statements, with the second ARRAY containing the descriptive names for the data. In the following example dealing with a bone density and fall risk study, the variables were named for the form (HB) and question number (05a-f) instead of the reasons for falling.

In the SAS code below, the first ARRAY, HB05, lists the non-descriptive binary variable names. In the second ARRAY, RSN, a parallel list with descriptive variable names corresponds to each of the variables in ARRAY HB05. The conditional IF statement checks if the ARRAY HB05 item is equal to one; if so, the CATX function is used to append, with a comma, the name of the ARRAY RSN variable onto variable FALL_LIST.

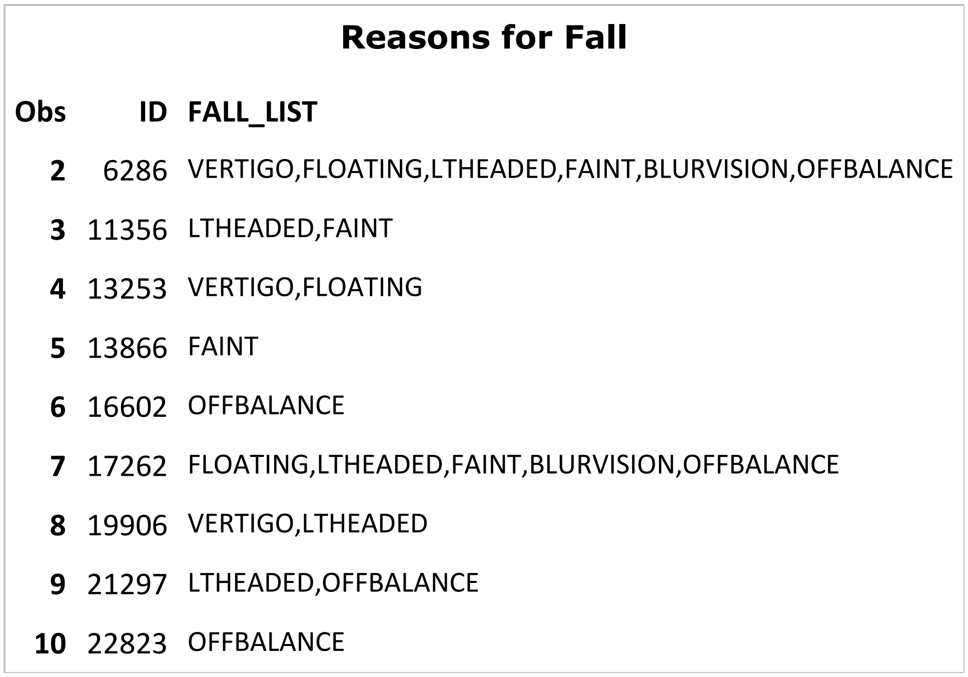
```
LENGTH fall_list $60.;
ARRAY hb05 (*) hb05a hb05b hb05c hb05d hb05e hb05f;
ARRAY rsn (*) vertigo floating ltheaded faint blurvision offbalance;
DO i = 1 TO DIM(hb05);
  IF hb05(i) = 1 THEN
      fall_list = CATX(",", fall_list, VNAME(rsn(i)) );
END;

PROC PRINT; VAR id fall_list ; TITLE "Reasons for Fall"; RUN; TITLE;
```

The printed output is below.



Reasons for Fall

Obs	ID	FALL_LIST
1	2409	OFFBALANCE



```
                         Reasons for Fall

Obs      ID  FALL_LIST
  2    6286  VERTIGO,FLOATING,LTHEADED,FAINT,BLURVISION,OFFBALANCE
  3   11356  LTHEADED,FAINT
  4   13253  VERTIGO,FLOATING
  5   13866  FAINT
  6   16602  OFFBALANCE
  7   17262  FLOATING,LTHEADED,FAINT,BLURVISION,OFFBALANCE
  8   19906  VERTIGO,LTHEADED
  9   21297  LTHEADED,OFFBALANCE
 10   22823  OFFBALANCE
```

Table 3: PROC PRINT Output Using Two Array Statements

ANOTHER ALTERNATIVE: VLABEL

Another alternative when working with non-descriptive variable names is to use the variable label for information. The VLABEL function is used in the DATA step and returns the variable label text. The default length of the returned value is 200 characters. If the variable does not have a label, the variable name is returned. The syntax for VLABEL is:

```
VLABEL(variable)
```

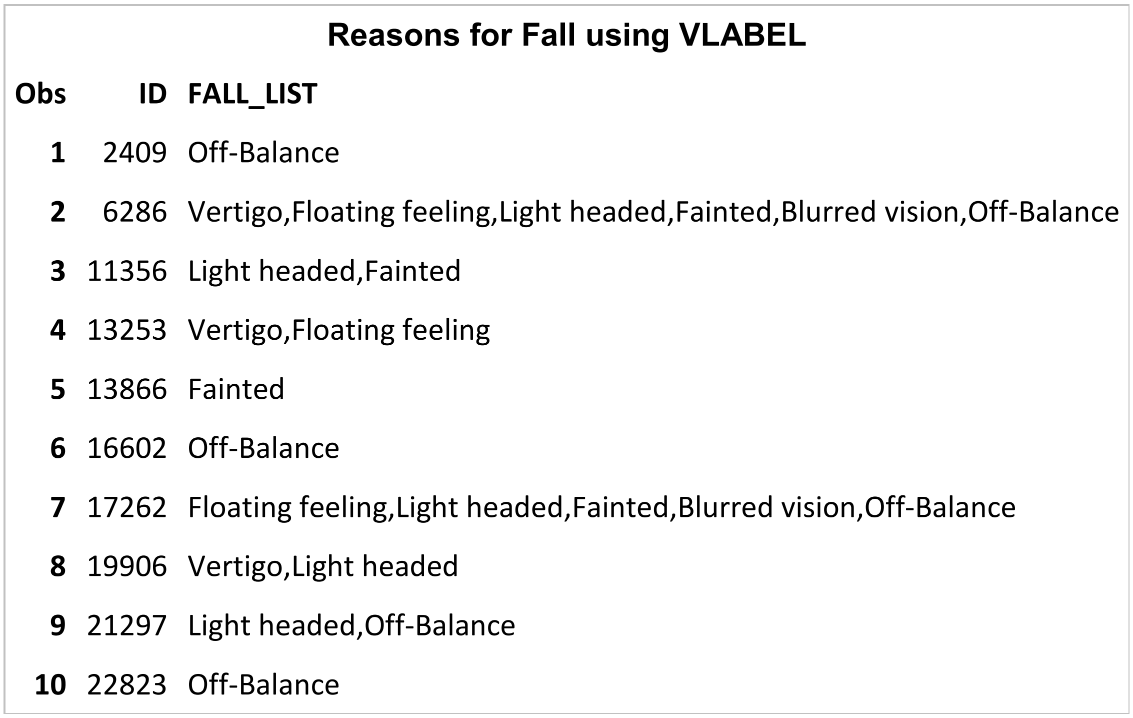
Using our above example with nonsensical variable names, SAS labels are assigned to each variable. An advantage to using a label is that it allows you to have a longer description. The disadvantage is the same—a verbose label may provide too much information for a descriptive list.

The following SAS code assigns a label to each variable and the ARRAY statement extracts that information for our list. After the LABEL is assigned to each variable, the SAS syntax is similar for this program. The number of characters needed for the LENGTH statement is increased to 75. The CATX function appends, with a comma, the variable label onto variable FALL_LIST:

```
LABEL hb05a= "Vertigo"
    hb05b= "Floating feeling"
    hb05c= "Light headed"
    hn05d= "Fainted"
    hb05e= "Blurred vision"
    hb05f= "Off-Balance";
LENGTH fall_list $75.;
ARRAY hb05 (*) hb05a hb05b hb05c hb05d hb05e hb05f;
  DO i = 1 TO DIM(hb05);
    IF hb05(i) = 1 THEN
       fall_list = CATX(",", fall_list, VLABEL(hb05(i)));
  END;
```

```
PROC PRINT; VAR id fall_list ; TITLE "Reasons for Fall using VLABEL"; RUN;
TITLE;
```

The output from the PRINT procedure is below. As specified in our call to function CATX, variable labels are separated by a comma:



```
                    Reasons for Fall using VLABEL

Obs     ID  FALL_LIST

  1   2409  Off-Balance

  2   6286  Vertigo,Floating feeling,Light headed,Fainted,Blurred vision,Off-Balance

  3  11356  Light headed,Fainted

  4  13253  Vertigo,Floating feeling

  5  13866  Fainted

  6  16602  Off-Balance

  7  17262  Floating feeling,Light headed,Fainted,Blurred vision,Off-Balance

  8  19906  Vertigo,Light headed

  9  21297  Light headed,Off-Balance

 10  22823  Off-Balance
```

Table 4: Printed Output Using CATX and VLABEL

CONCLUSION

The CATX and VNAME functions can be used to combine lists of variable names. Adding conditional statements allows for the selective insertion of variable names. Even if the variable names are not descriptive, by using two ARRAY statements or the VLABEL function, it is easy to have SAS speak to you.

ACKNOWLEDGMENTS

Data in this manuscript were collected by the Women's Interagency HIV Study (WIHS). The contents of this publication are solely the responsibility of the authors and do not represent the official views of the National Institutes of Health (NIH). WIHS (Principal Investigators): UAB-MS WIHS (Mirjam-Colette Kempf and Deborah Konkle-Parker), U01-AI-103401; Atlanta WIHS (Ighovwerha Ofotokun and Gina Wingood), U01-AI-103408; Bronx WIHS (Kathryn Anastos and Anjali Sharma), U01-AI-035004; Brooklyn WIHS (Howard Minkoff and Deborah Gustafson), U01-AI-031834; Chicago WIHS (Mardge Cohen and Audrey French), U01-AI-034993; Metropolitan Washington WIHS (Seble Kassaye), U01-AI-034994; Miami WIHS (Margaret Fischl and Lisa Metsch), U01-AI-103397; UNC WIHS (Adaora Adimora), U01-AI-103390; Connie Wofsy Women's HIV Study, Northern California (Ruth Greenblatt, Bradley Aouizerat, and Phyllis Tien), U01-AI-034989; WIHS Data Management and Analysis Center (Stephen Gange and Elizabeth Golub), U01-AI-042590; Southern California WIHS (Joel Milam), U01-HD-032632 (WIHS I – WIHS IV). The WIHS is funded primarily by the National

Institute of Allergy and Infectious Diseases (NIAID), with additional co-funding from the Eunice Kennedy Shriver National Institute of Child Health and Human Development (NICHD), the National Cancer Institute (NCI), the National Institute on Drug Abuse (NIDA), and the National Institute on Mental Health (NIMH).

CONTACT INFORMATION

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

Gayle Springer
Johns Hopkins University Bloomberg School of Public Health
+1 410-223-1657
gspringer@jhu.edu