SESUG Paper 268-2019

How to Keep Multiple Formats in One Variable after Transpose

Mindy Wang, Independent Consultant

ABSTRACT

In many industries and research fields, proc transpose are used very often. When many variables with their individual formats are transposed into one variable, we lose the formats. We can do a series of if then statements to put the formats back in. However, when the variables involved are too many, the above method can be very tedious. This paper illustrates how to extract formats from dictionary.columns or sashelp.vcolumn, and then use PUTN function to assign the formats at run time and make the task much easier. In addition, it is much easier to apply the same method to other projects without a lot of hard coding in the SAS program. Efficiency is largely increased with this method.

INTRODUCTION

Problems arise when transposing the different observations with different formats into one variable. It is impossible to have different formats for different observations for the same variable. Therefore the original formats do not show up at all after transpose.

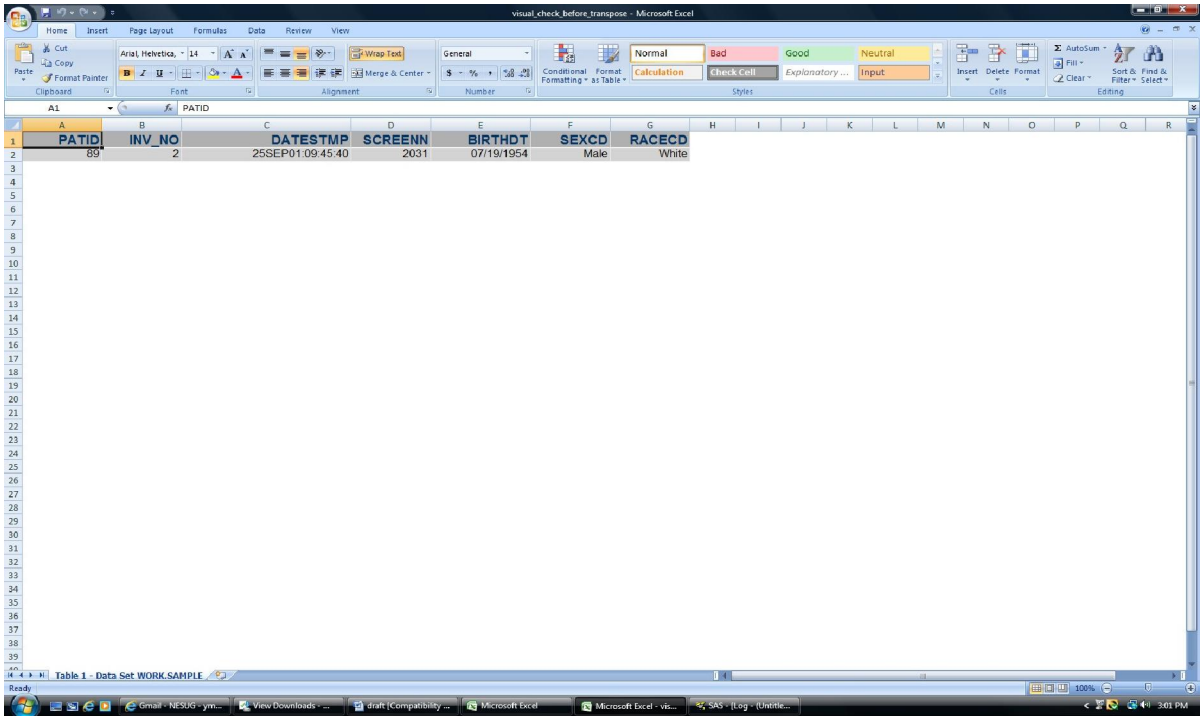
PROBLEM

I have the following formats before transpose:

```
PROC FROMAT;
   VALUE sexcode
1 = 'Male  '
2 = 'Female'
     ;
   VALUE racecode
1 = 'White'
2 = 'Black'
3 = 'Hispanic'
4 = 'Asian'
5 = 'Other'
6 = 'Multi-racial'
     ;
……

RUN;
```

```
DATA demo2;
SET desk.demo;
FORMAT datestmp
datetime17.
       birthdt mmddyy10.

       sexcd sexcode.
       racecd racecode.
       ……
;
RUN;
```

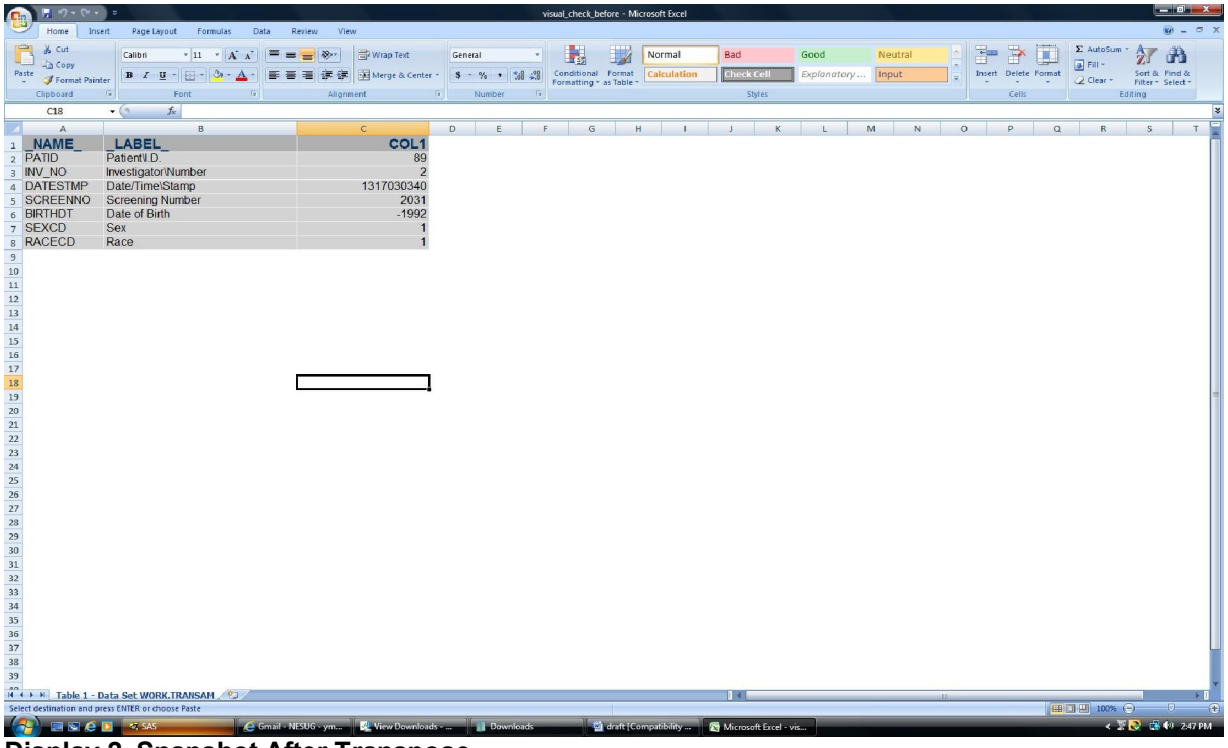
Following is the before-transpose snapshot. Please note that the sexcd and racecd variables display text formats.



PATID	INV_NO	DATESTMP	SCREENN	BIRTHDT	SEXCD	RACECD
89	2	25SEP01:09:45:40	2031	07/19/1954	Male	White

Display 1. Snapshot Before Transpose

Display 2 is the after-transpose snapshot. Please note the sexcd and racecd now are numeric, instead of text.



NAME	_LABEL_	COL1
PATID	Patient\I.D.	89
INV_NO	Investigator\Number	2
DATESTMP	Date/Time\Stamp	1317030340
SCREENNO	Screening Number	2031
BIRTHDT	Date of Birth	-1992
SEXCD	Sex	1
RACECD	Race	1

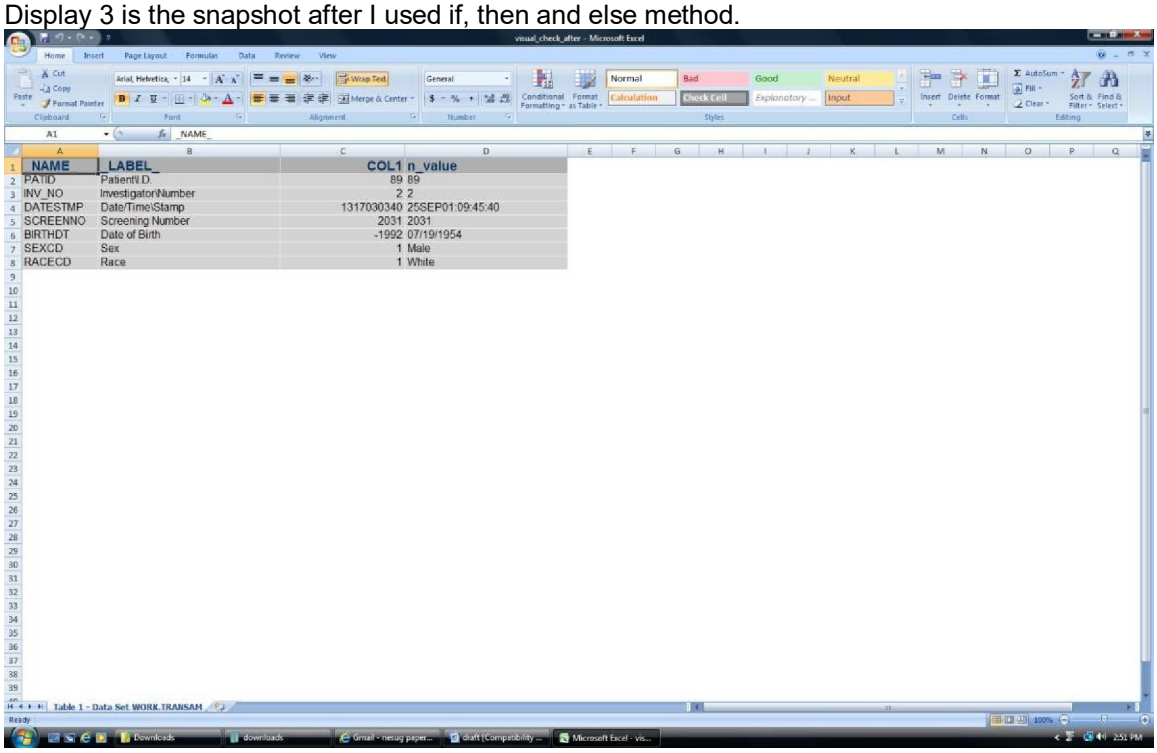
Display 2. Snapshot After Transpose

IF THEN AND ELSE METHOD

There is some work needed to fix the problem. One way of doing it is to have a series of if then statements to accommodate value labels and hard code the value labels into a new variable. We can do it individually to each variable with format. Following is the code that I used with this method:

```
IF UPCASE(COMPRESS(_name_))="DATESTMP" THEN DO;
 n_value=PUT(Col1, datetime17.);

END;


ELSE IF UPCASE(COMPRESS(_name_))="BIRTHDT" THEN DO;
 n_value=PUT(Col1,mmddyy10. );

END;


ELSE IF UPCASE(COMPRESS(_name_))="SEXCD" THEN DO;
 IF Col1=1 THEN n_value= 'Male   ' ;
      ELSE IF Col1=2 THEN n_value= 'Female' ;

END;


ELSE IF UPCASE(COMPRESS(_name_))="RACECD" THEN DO;
   IF Col1=1 THEN n_value= 'White       ' ;
   ELSE IF Col1=2 THEN n_value= 'Black'    ;
   ELSE IF Col1=3 THEN n_value= 'Hispanic'   ;
   ELSE IF Col1=4 THEN n_value= 'Asian'    ;
   ELSE IF Col1=5 THEN n_value= 'Other' ;
   ELSE IF Col1=6 THEN n_value= 'Multi-racial'   ;

END;


ELSE n_value=Col1;
........
```

Display 3 is the snapshot after I used if, then and else method.



	NAME	LABEL_	COL1	n_value
2	PATID	PatientID	89	89
3	INV_NO	Investigator\Number	2	2
4	DATESTMP	Date/Time\Stamp	1317030340	25SEP01:09:45:40
5	SCREENNO	Screening Number	2031	2031
6	BIRTHDT	Date of Birth	-1992	07/19/1954
7	SEXCD	Sex	1	Male
8	RACECD	Race	1	White

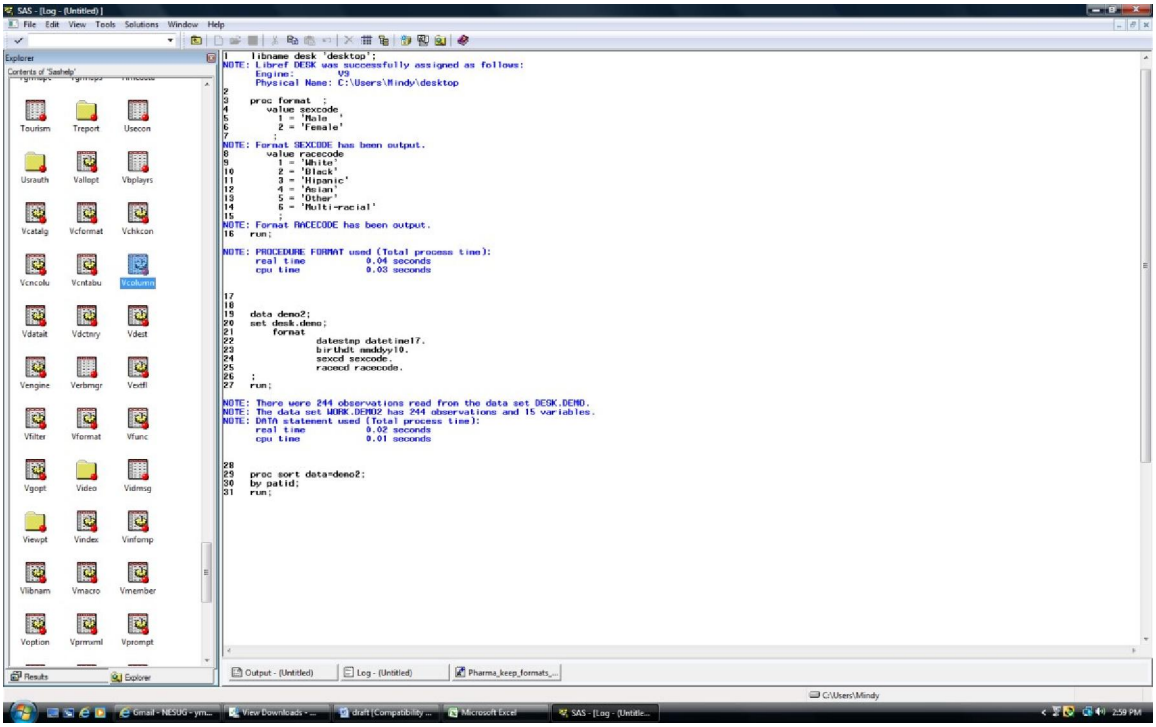
Display 3. Snapshot Using If Then Statements

While the method is working well with only a few variables with formats to be transposed, it becomes very tedious when the observations are in the hundreds or thousands.

EXTRACTING FORMATS FROM DICTIONARY.COLUMNS OR SASHELP.VCOLUMN AND APPLYNG PUTN FUNCTION

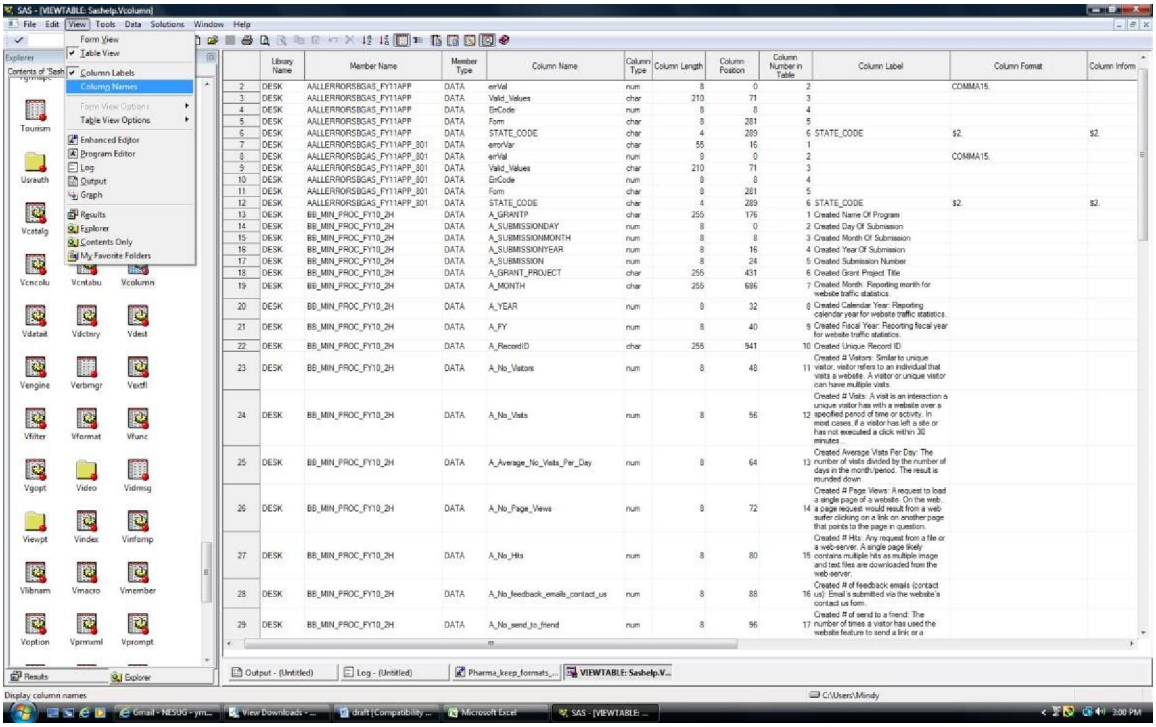
An easier way of doing it is to pull out the format information from dictionary. columns using sql, and then apply putn function to assign the individual format for each observation at run time. If you are not a sql person, you can do the same by pulling out format information from sashelp.vcolumn. Following is the demonstration using dictionary.columns or sashelp.vcolumn then applying the formats using putn function. This method is much easier to adapt to other situations without a lot of hard coding.

Display 4 is where we can find the sashelp.vcolumn file.



Display 4. Where to Find Sashelp.Vcolumn File

Display 5 shows you what sashelp.vcolumn file looks like.



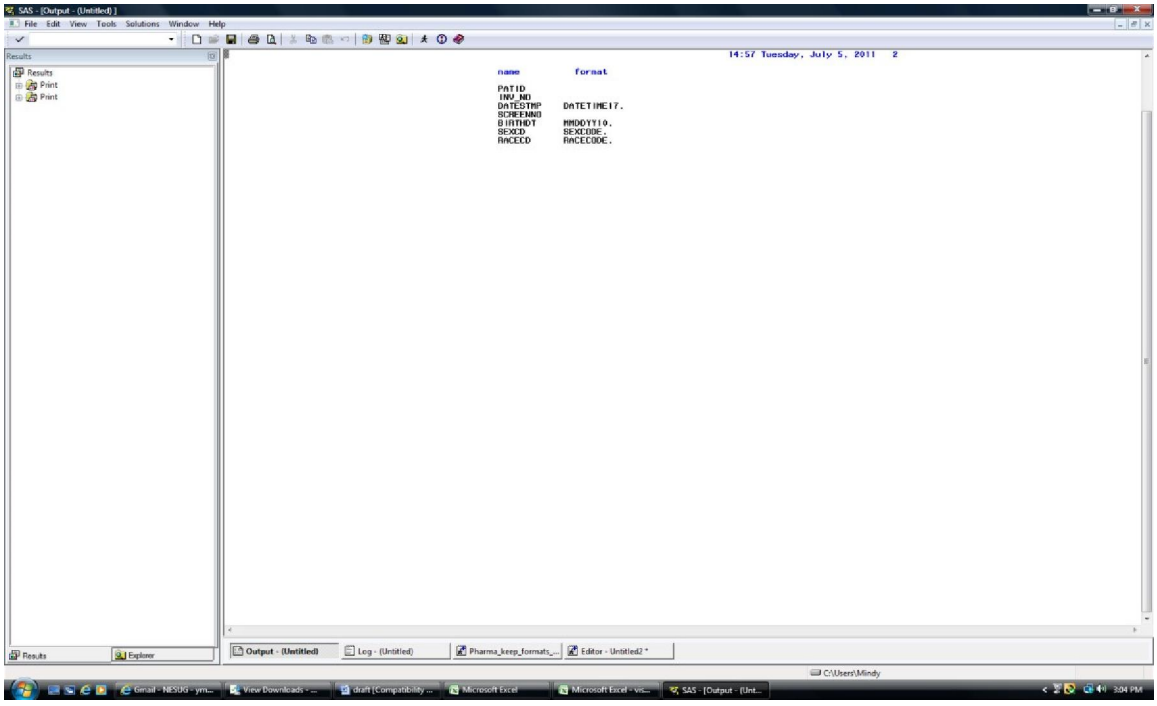
	Library Name	Member Name	Member Type	Column Name	Column Type	Column Length	Column Position	Column Number in Table	Column Label	Column Format	Column Informat
2	DESK	AALLERRORSBGAS_FY11APP	DATA	errVal	num	8	0	2		COMMA15.	
3	DESK	AALLERRORSBGAS_FY11APP	DATA	Valid_Values	char	210	71	3			
4	DESK	AALLERRORSBGAS_FY11APP	DATA	ErrCode	num	8	8	4			
5	DESK	AALLERRORSBGAS_FY11APP	DATA	Form	char	8	281	5			
6	DESK	AALLERRORSBGAS_FY11APP	DATA	STATE_CODE	char	4	289	6	STATE_CODE	$2.	$2.
7	DESK	AALLERRORSBGAS_FY11APP_801	DATA	errorVar	char	55	16	1			
8	DESK	AALLERRORSBGAS_FY11APP_801	DATA	errVal	num	8	0	2		COMMA15.	
9	DESK	AALLERRORSBGAS_FY11APP_801	DATA	Valid_Values	char	210	71	3			
10	DESK	AALLERRORSBGAS_FY11APP_801	DATA	ErrCode	num	8	8	4			
11	DESK	AALLERRORSBGAS_FY11APP_801	DATA	Form	char	8	281	5			
12	DESK	AALLERRORSBGAS_FY11APP_801	DATA	STATE_CODE	char	4	289	6	STATE_CODE	$2.	$2.
13	DESK	BB_MIN_PROC_FY10_2H	DATA	A_GRANTP	char	255	176	1	Created Name Of Program		
14	DESK	BB_MIN_PROC_FY10_2H	DATA	A_SUBMISSIONDAY	num	8	0	2	Created Day Of Submission		
15	DESK	BB_MIN_PROC_FY10_2H	DATA	A_SUBMISSIONMONTH	num	8	8	3	Created Month Of Submission		
16	DESK	BB_MIN_PROC_FY10_2H	DATA	A_SUBMISSIONYEAR	num	8	16	4	Created Year Of Submission		
17	DESK	BB_MIN_PROC_FY10_2H	DATA	A_SUBMISSION	num	8	24	5	Created Submission Number		
18	DESK	BB_MIN_PROC_FY10_2H	DATA	A_GRANT_PROJECT	char	255	431	6	Created Grant Project Title		
19	DESK	BB_MIN_PROC_FY10_2H	DATA	A_MONTH	char	255	686	7	Created Month: Reporting month for website traffic statistics.		
20	DESK	BB_MIN_PROC_FY10_2H	DATA	A_YEAR	num	8	32	8	Created Calendar Year: Reporting calendar year for website traffic statistics.		
21	DESK	BB_MIN_PROC_FY10_2H	DATA	A_FY	num	8	40	9	Created Fiscal Year: Reporting fiscal year for website traffic statistics.		
22	DESK	BB_MIN_PROC_FY10_2H	DATA	A_RecordID	char	255	941	10	Created Unique Record ID		
23	DESK	BB_MIN_PROC_FY10_2H	DATA	A_No_Visitors	num	8	48	11	Created # Visitors: Similar to unique visitor, visitor refers to an individual that visits a website. A visitor or unique visitor can have multiple visits.		
24	DESK	BB_MIN_PROC_FY10_2H	DATA	A_No_Visits	num	8	56	12	Created # Visits: A visit is an interaction a unique visitor has with a website over a specified period of time or activity. In most cases, if a visitor has left a site or has not executed a click within 30 minutes...		
25	DESK	BB_MIN_PROC_FY10_2H	DATA	A_Average_No_Visits_Per_Day	num	8	64	13	Created Average Visits Per Day: The number of visits divided by the number of days in the month/period. The result is rounded down.		
26	DESK	BB_MIN_PROC_FY10_2H	DATA	A_No_Page_Views	num	8	72	14	Created # Page Views: A request to load a single page of a website. On the web, a page request would result from a web surfer clicking on a link on another page that points to the page in question.		
27	DESK	BB_MIN_PROC_FY10_2H	DATA	A_No_Hits	num	8	80	15	Created # Hits: Any request from a file or a web-server. A single page likely contains multiple hits as multiple image and text files are downloaded from the web-server.		
28	DESK	BB_MIN_PROC_FY10_2H	DATA	A_No_feedback_emails_contact_us	num	8	88	16	Created # of feedback emails (contact us): Email's submitted via the website's contact us form.		
29	DESK	BB_MIN_PROC_FY10_2H	DATA	A_No_send_to_friend	num	8	96	17	Created # of send to a friend: The number of times a visitor has used the website feature to send a link or a		

Display 5. What Sashelp.Vcolumn File Looks Like

You can use the following code to print the variable names and formats from sashelp.vcolumn:

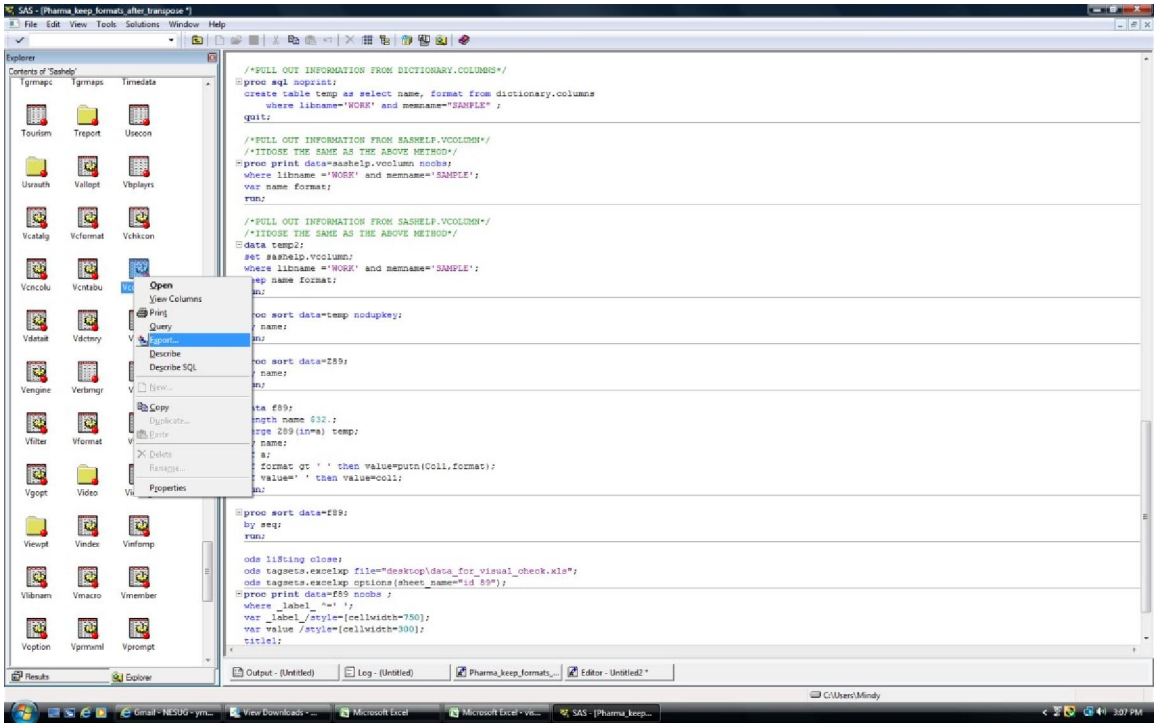
```
PROC PRINT DATA=sashelp.vcolumn NOOBS;
WHERE libname ='WORK' and memname='SAMPLE';
VAR name format;
RUN;
```

Display 6 is the snapshot of the output window after running the above code.



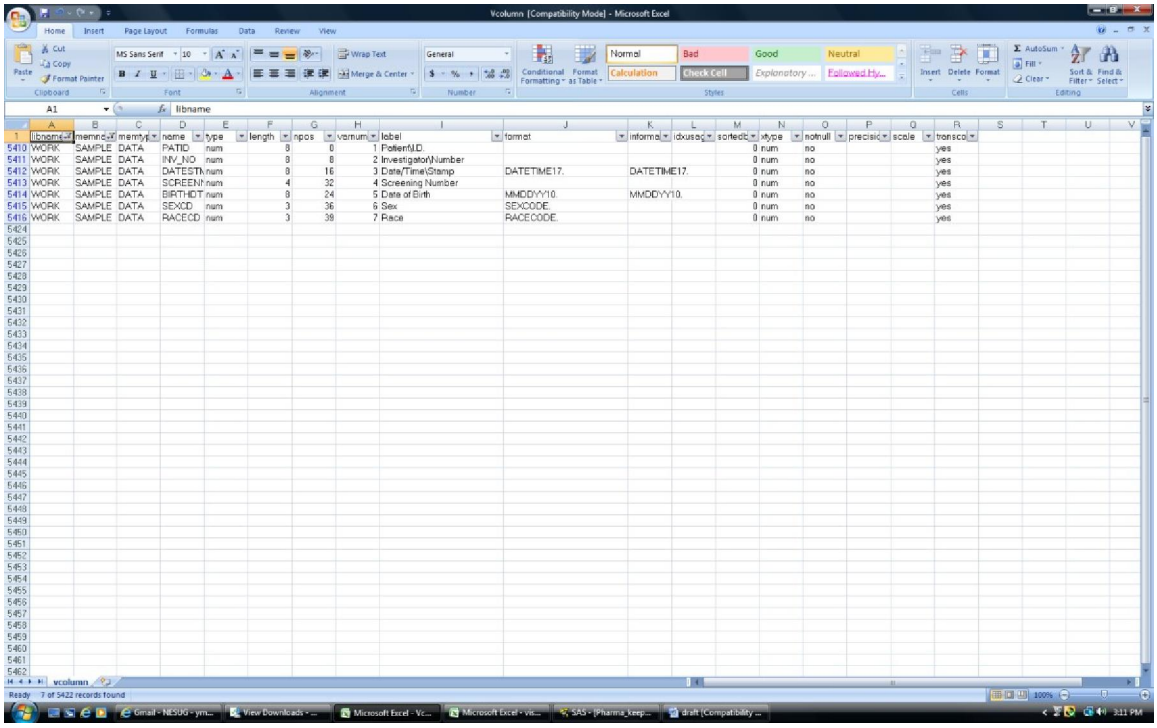
Display 6. To See All the Formats in Sashelp.Vcolumn File Using the Above Code

You can also export the sashelp.vcolumn file to excel if you are more familiar with Excel.



Display7. How To View Sashelp.Vcolumn File in Excel

Display 8 is the snapshot of the Excel file with all other data from Sashelp.Vcolumn except the sample file (that we are interested) filtered out.



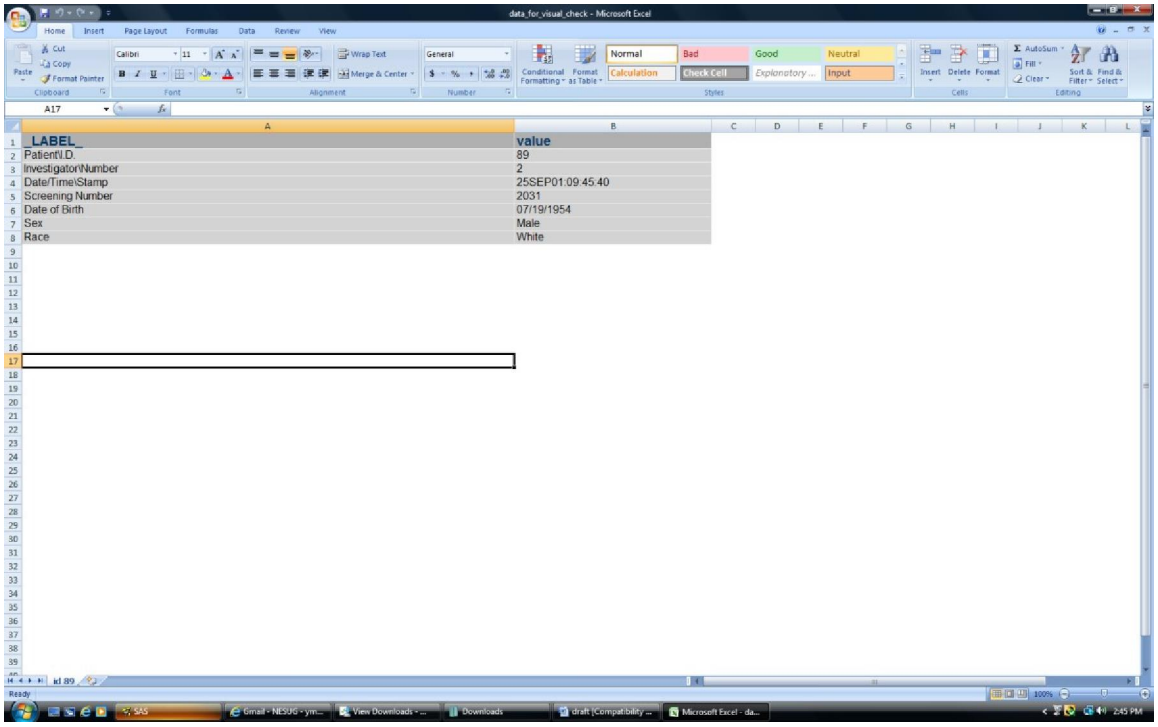
Display 8. Sashelp.Vcolumn File in Excel Formats

Following is the code using proc sql and then print the output file to Excel using ods tagsets.excelxp:

```
PROC TRANSPOSE DATA=sample NAME=name OUT=t89 ;

RUN;


DATA Z89;
SET t89;
seq+1;
RUN;


/*PULL OUT INFORMATION FROM DICTIONARY.COLUMNS*/

PROC SQL NOPRINT;
CREATE TABLE temp AS SELECT name, format FROM dictionary.columns
 WHERE libname='WORK' and memname="SAMPLE" ;

QUIT;


/*PULL OUT INFORMATION FROM SASHELP.VCOLUMN*/
/*IT DOSE THE SAME AS THE ABOVE METHOD*/
/*USE THIS ONE OR THE ABOVE SQL METHOD*/
DATA temp2;
SET sashelp.vcolumn;
WHERE libname ='WORK' and memname='SAMPLE';
KEEP name format;

RUN;


PROC SORT DATA=temp NODUPKEY;
BY name;
RUN;

PROC SORT DATA=Z89;
BY name;

RUN;


DATA f89;
LENGTH name $32.;
MERGE Z89(in=a) temp;
BY name;
IF a;
IF format gt ' ' THEN value=PUTN(Col1,format);
IF value=' ' THEN value=col1;
RUN;

PROC SORT DATA=f89;
  BY seq;
RUN;
```

```
ODS LISTING CLOSE;

ODS tagsets.excelxp FILE="desktop\data_for_visual_check.xls";
ODS tagsets.excelxp OPTIONS(SHEET_NAME="id 89");

PROC PRINT DATA=f89 NOOBS;

  WHERE _label_ ^=' ';
VAR _label_/STYLE=[CELLWIDTH=750];
VAR value /STYLE=[CELLWIDTH=300];
TITLE1;

RUN;


ODS tagsets.excelxp CLOSE;

ODS LISTING;
```

Following is the snapshot of the file after applying formats using PUTN function.



	LABEL	value
2	Patient\I.D.	89
3	Investigator\Number	2
4	Date/Time\Stamp	25SEP01:09:45:40
5	Screening Number	2031
6	Date of Birth	07/19/1954
7	Sex	Male
8	Race	White

Display 9. Snapshot of The File After Applying Formats Using PUTN Function

CONCLUSION

The latter method certainly makes your life a lot easier. Not only it has minimized typing when the variables involved are too many. It is also more adaptable to other projects. It is always beneficial to write the programs that are easier to adapt to new situations, even though at first it takes more time to develop. In the long run, it really saves time.

ACKNOWLEDGMENTS

Many thanks to SAS® Technical Support for providing valuable suggestions in the developing of this paper.

CONTACT INFORMATION

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

Mindy Wang
14412 Stonebridge View Dr.
North Potomac, MD 20878
Phone: (240) 760-9988
Email: ymindywang@yahoo.com