PharmaSUG 2020 - Paper AP-278

SAS® Formats: Same Name, Different Definitions FORMAT-ters of Inconvenience

Jacqueline (Jackie) Fitzpatrick, Senior Clinical Programmer at SCHARP/Fred Hutch

ABSTRACT

While working with multiple data sets, especially if they were created by different programmers, do you check the formats that are applied to the variables. This paper shows why it is best to proactively examine format catalogs while working with multiple data libraries, how to check the uniformity in their definitions, and what is necessary to fix discrepancies. Hopefully, you will benefit by adding this tool to your current collection.

INTRODUCTION

In SAS® version 9.4, the maximum length of the numeric format name is 32 bytes; 31 bytes for character format names. It can contain letters of the Latin alphabet, numerals, or underscores, but it must start and finish with a letter of the Latin alphabet (A–Z, a–z) or an underscore. With these rules, you can have over 8.33×10^{49} different numeric format names and over 2.25×10^{48} different character format names. It's amazing how one can have so many choices to name a format, but one format defined differently wreaks havoc with your analysis.

BACKGROUND

Our organization receives behavioral questionnaire data from outside sources as SPSS sav files. Typically, they use question number as their variable names. Now, this seems like a great idea, but question number one in section B at Baseline may not be the same for the following study visits. For example, the question could have 1 as “Yes”, 2 as “No”, but the following visits use 1 as “Once”, 2 as “Twice”, and 3 as “Three or More”. Both surveys will create a format named B1A after being imported into SAS; the definition of the format depends on the last imported survey.

WHAT CAN I DO?

I can tell those creating the structure, “Hey! Don’t do that! Use different format names.” They can say, “Sure!”, which gives me time to ride unicorns through cotton candy clouds. Now, let’s be real: most likely, they will say, “No.”

I can scour the surveys as PDF/ TXT / RTF / DOC files with a highlighter in your hand. This method is overwhelming. It would be hard to find what needs to be fixed.

Programmatically, I can rename all the formats by incrementing by letter (A-Z, AA-ZZ, etc.). While it is a fun logic puzzle, the format names are not intuitive to the field names (“Format YYZ is for which field?”). Also, some formats CAN be used across the different surveys; I don’t want to ruin what already works.

I want something efficient with little effort.

WHAT I DO

I use a macro to compare the format catalogs and export the results in a Microsoft Excel workbook.

DEFINE AND %DO

I start my macro facility by defining my macro variables and using a macro %DO loop. I convert each SPSS file into a SAS data set with the IMPORT procedure; this will automatically create a format catalog in the WORK library. Using the FORMAT procedure with the CNTLOUT option, the catalog becomes a

SAS data set, keeping only the essential variables: FMTNAME, START, END, and LABEL. For this data set, I change the name and label for the LABEL variable to match the survey name to avoid confusion and overwriting variables when merging. I also create a subset of the format catalog data set with one record per format name (FMTNAME) and an indicator variable set to one. My program starts as:

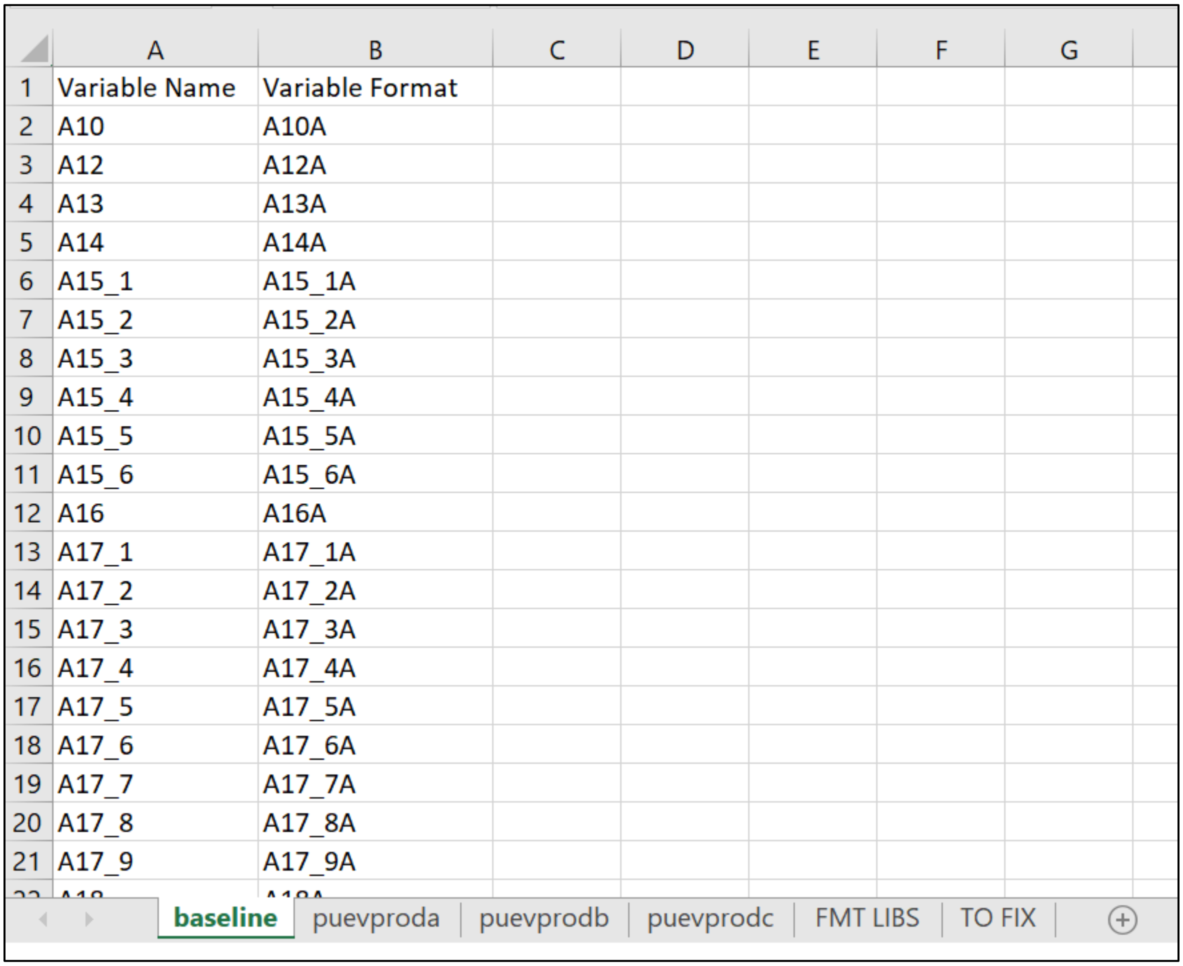
```
%let z=%STR(baseline puevproda puevprodb puevprodc);
%let nsav = %sysfunc(countw(&z));

filename xout "sav_fmts.xlsx" encoding="utf-8";

%macro checkfmt;
    %do x= 1 %to &nsav.;
        %let savn= %scan(&z,&x);

        proc import datafile = "&savn..sav"
            out=dat&x. dbms=sav replace;
        run;

        proc contents data=dat&x. noprint out=pc&x.;
        run;

        proc format library=work cntlout=fmtf&x.
            (keep=fmtname start end label rename=(label=&savn.));
        run;

        data fmt&x.;
               attrib &savn. label="&savn.";
            set fmtf&x.;
        run;

        proc sort;
            by fmtname start end;
        run;

        proc catalog cat=WORK.formats kill;
        run;

        proc sql noprint;
            create table inf&x. as
                select distinct fmtname, 1 as nf&x.
                from fmtf&x.
                order by fmtname;

            create table pcf&x. as
                select distinct a.name, a.format
                from pc&x. as a join inf&x. as b
                on a.format=b.fmtname;
        quit;

        PROC EXPORT DATA=pcf&x. OUTFILE=xout dbms=xlsx label replace;
            sheet="&savn.";
        RUN;
        %end;
```

WHERE IS THE $

The surveys I work with use numeric formats, but I would modify the code above if character formats were included. When creating a data set from PROC FORMAT, the dollar sign in the character format name is removed. If I was working with a numeric format that had the same name as a character format without its dollar sign, the above code would treat them as if they were the same format. For example, formats YESNO and $YESNO will appear in the output data set as YESNO with possible values of 0, 1, Y, and N. Therefore, to turn off unnecessary alerts, I would include the variable TYPE in the output data set and use it as a key with FMTNAME.

HELPFUL CONTENTS

I added the CONTENTS procedure for each survey data set and use the EXPORT procedure to list variable names and formats in a Microsoft Excel worksheet for reference. If any discrepancies exist, I can easily find the variable I need to update. Figure 1 displays the output.



	A	B
1	Variable Name	Variable Format
2	A10	A10A
3	A12	A12A
4	A13	A13A
5	A14	A14A
6	A15_1	A15_1A
7	A15_2	A15_2A
8	A15_3	A15_3A
9	A15_4	A15_4A
10	A15_5	A15_5A
11	A15_6	A15_6A
12	A16	A16A
13	A17_1	A17_1A
14	A17_2	A17_2A
15	A17_3	A17_3A
16	A17_4	A17_4A
17	A17_5	A17_5A
18	A17_6	A17_6A
19	A17_7	A17_7A
20	A17_8	A17_8A
21	A17_9	A17_9A
22	A18	A18A

Figure 1. Output of data set contents

CRITICAL

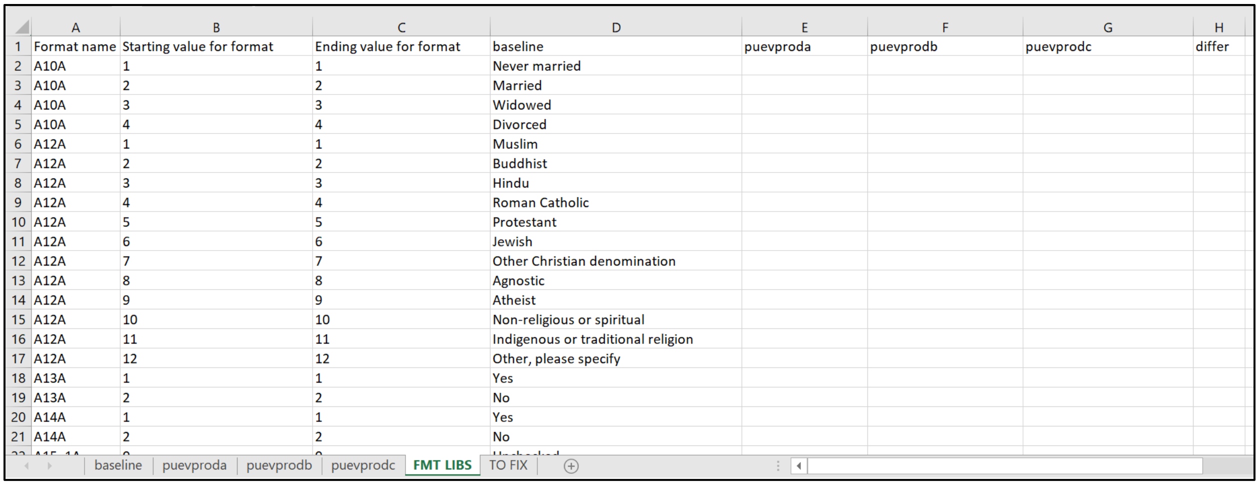
Before each PROC IMPORT step that reads the SPSS file, it is important to delete the format catalog in the WORK library using the CATALOG procedure. We must not cross-contaminate our format catalogs.

COMPARE CATALOGS

After reading in each survey file, I merge my data sets together. First, I merge the format catalog data sets by FMTNAME START END in a DATA Step MERGE. Afterwards, I merge the format indicator data sets with this file by FMTNAME in a DATA Step MERGE. In the same DATA Step, I create arrays and DO loops to find which formats exist, compare the values across the surveys, and add an indicator for records with differences. Afterwards, I use PROC EXPORT the file into a Microsoft Excel worksheet. The code following the %DO looks like this:

```
data fmtlbl;
    merge fmt1-fmt&nsav.;
    by fmtname start end;
run;

data fmtlibs (drop=q r svn1-svn&nsav. nf1-nf&nsav.);
    merge fmtlbl inf1-inf&nsav.;
    by fmtname;

    array s $ svn1-svn&nsav.;
    array n   nf1-nf&nsav.;

    differ=" ";
    do q=1 to &nsav.;
        if n{q}=1 then do r=1 to &nsav.;
            if n{r}=1 and s{r} ne s{q} then differ="X";
            end;
        end;
run;

PROC EXPORT DATA=fmtlibs OUTFILE=xout dbms=xlsx label replace;
    sheet="FMT LIBS";
RUN;
```

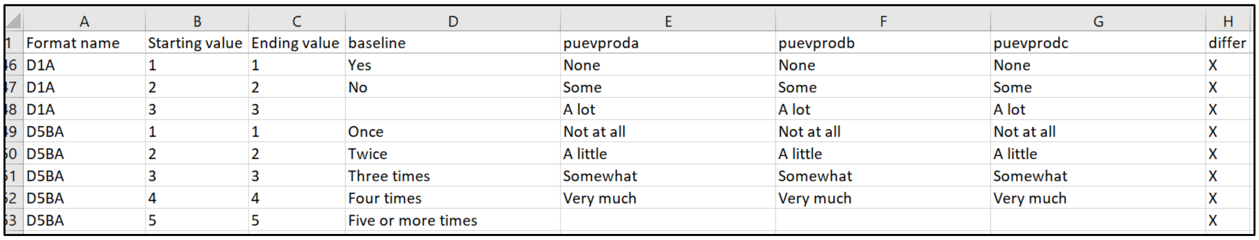
Figure 2 shows the output file.



	A	B	C	D	E	F	G	H
1	Format name	Starting value for format	Ending value for format	baseline	puevproda	puevprodb	puevprodc	differ
2	A10A	1	1	Never married				
3	A10A	2	2	Married				
4	A10A	3	3	Widowed				
5	A10A	4	4	Divorced				
6	A12A	1	1	Muslim				
7	A12A	2	2	Buddhist				
8	A12A	3	3	Hindu				
9	A12A	4	4	Roman Catholic				
10	A12A	5	5	Protestant				
11	A12A	6	6	Jewish				
12	A12A	7	7	Other Christian denomination				
13	A12A	8	8	Agnostic				
14	A12A	9	9	Atheist				
15	A12A	10	10	Non-religious or spiritual				
16	A12A	11	11	Indigenous or traditional religion				
17	A12A	12	12	Other, please specify				
18	A13A	1	1	Yes				
19	A13A	2	2	No				
20	A14A	1	1	Yes				
21	A14A	2	2	No				

Figure 2. Output of Format Catalog Data Set

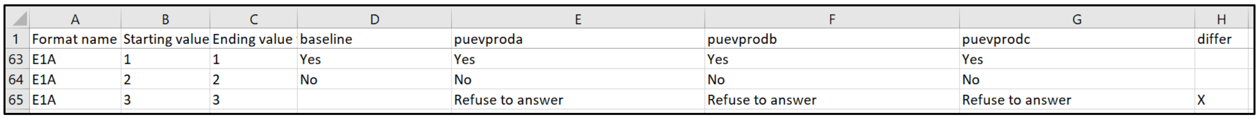
In the beginning, we see the baseline survey has formats that do not exist in the remaining surveys. Our discrepancy indicator is <null>. Eventually, we do find discrepancies between the baseline survey and surveys from the following visits. Figure 3 exhibits a discrepancy.



	A	B	C	D	E	F	G	H
1	Format name	Starting value	Ending value	baseline	puevproda	puevprodb	puevprodc	differ
46	D1A	1	1	Yes	None	None	None	X
47	D1A	2	2	No	Some	Some	Some	X
48	D1A	3	3		A lot	A lot	A lot	X
49	D5BA	1	1	Once	Not at all	Not at all	Not at all	X
50	D5BA	2	2	Twice	A little	A little	A little	X
51	D5BA	3	3	Three times	Somewhat	Somewhat	Somewhat	X
52	D5BA	4	4	Four times	Very much	Very much	Very much	X
53	D5BA	5	5	Five or more times				X

Figure 3. Discrepancies between Surveys

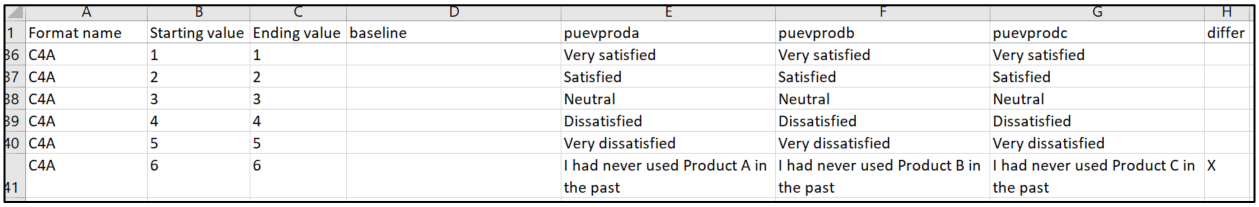
Sometimes, the surveys for the following visits have an extra category compared to the baseline survey. This may or may not be considered a discrepancy. Here, it is declared a discrepancy because the format indicator data set looks at format (FMTNAME) as a whole and not by each category of the format. While merging the format catalog data sets, I could use the IN= option for each dataset in the MERGE statement. Then, this would not show up as a discrepancy. Use your discretion and ask others on how to handle this. Figure 4 provides this example.



	A	B	C	D	E	F	G	H
1	Format name	Starting value	Ending value	baseline	puevproda	puevprodb	puevprodc	differ
63	E1A	1	1	Yes	Yes	Yes	Yes	
64	E1A	2	2	No	No	No	No	
65	E1A	3	3		Refuse to answer	Refuse to answer	Refuse to answer	X

Figure 4. Possible Discrepancy between Catalogs

We even have differences between the surveys at follow-up visits. Again, check with those who also work with this data. As separated data sets, they may want to keep the current values as is which means change the format names. Still, if appending the data sets together, they may need a more generic format. Using Figure 5 as an example, I would create another format where the value for 6 would be "I had never used the study product in the past".



	A	B	C	D	E	F	G	H
1	Format name	Starting value	Ending value	baseline	puevproda	puevprodb	puevprodc	differ
36	C4A	1	1		Very satisfied	Very satisfied	Very satisfied	
37	C4A	2	2		Satisfied	Satisfied	Satisfied	
38	C4A	3	3		Neutral	Neutral	Neutral	
39	C4A	4	4		Dissatisfied	Dissatisfied	Dissatisfied	
40	C4A	5	5		Very dissatisfied	Very dissatisfied	Very dissatisfied	
41	C4A	6	6		I had never used Product A in the past	I had never used Product B in the past	I had never used Product C in the past	X

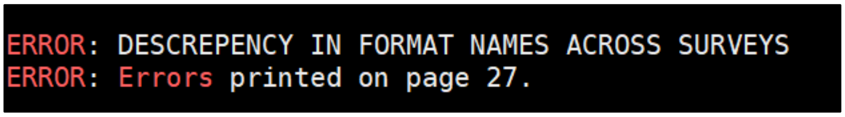
Figure 5. Minor Discrepancy between Surveys

SET THE ALARM

Honestly, I only want to open this Microsoft Excel workbook when necessary and easily find why. Therefore, I end my macro by looking for variations. If any exists, I export only the formats marked as discrepant to an additional Microsoft Excel worksheet and I generate an error message in the log. I add the following code:

```
    proc sql noprint;
            select count(*) into :anyf2fix from fmtlibs where differ="X";
        quit;

        %if &anyf2fix>0 %then %do;
            proc sql noprint;
                select distinct strip(fmtname) into :fmt2fix separated by
'" "'
                from fmtlibs where differ="X";

                create table fmt2fix as
                    select * from fmtlibs
                    where fmtname in ("&fmt2fix");
            quit;

            PROC EXPORT DATA=fmt2fix OUTFILE=xout dbms=xlsx label replace;
                sheet="TO FIX";
            RUN;
            %put %str(E)%str(RROR: DESCREPENCY IN FORMAT NAMES ACROSS
SURVEYS);
            %end;
    %mend checkfmt;
```

The Microsoft Excel worksheet “TO FIX” is like Figure 2, but only the formats that need my attention. The error message shown in Figure 6 tells me to investigate these issues. If this error message is not generated, then everything is fine as it is; nothing to see here.



```
ERROR: DESCREPENCY IN FORMAT NAMES ACROSS SURVEYS
ERROR: Errors printed on page 27.
```

Figure 6. Error Message

FIX IT

Since there are discrepancies, I now look through the formats and decide how to fix this issue. To check my work on how to fix this, I create a copy of the very same program that found the discrepancies and add more steps. After the PROC IMPORT of each file, I use PROC CATALOG to rename formats, PROC

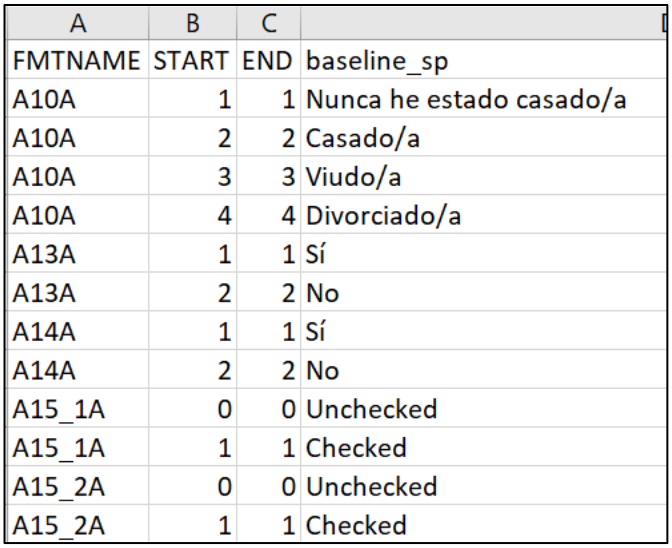
FORMAT to add any necessary formats, and the DATASET procedure to modify variables of each data set. The code looks like:

```
proc import datafile = "&savn..sav"
    out=dat&x. dbms=sav replace;
run;

/*** FIXING BASELINE FORMAT CATALOG ***/
%if &savn=baseline %then %do;
    proc catalog catalog=work.Formats;
        change c8a = c8ax (et=format);
        change d1a = d1ax (et=format);
        change D5BA = D5BAx (et=format);
        change d8a = d8ax (et=format);
        change d9a = d9ax (et=format);
    run;quit;

    proc datasets library=work;
        modify dat&x.;
            format c8 c8ax.;
            format d1 d1ax.;
            format d5b d5bax.;
            format d8 d8ax.;
            format d9 d9ax.;
    run; quit;
    %end;

proc contents data=dat&x. noprint out=pc&x.;
run;
```

I apply similar code to the remaining surveys. Once the inconsistencies disappear, I add these procedures to my program in production.

SAME LOGIC, DIFFERENT APPLICATION

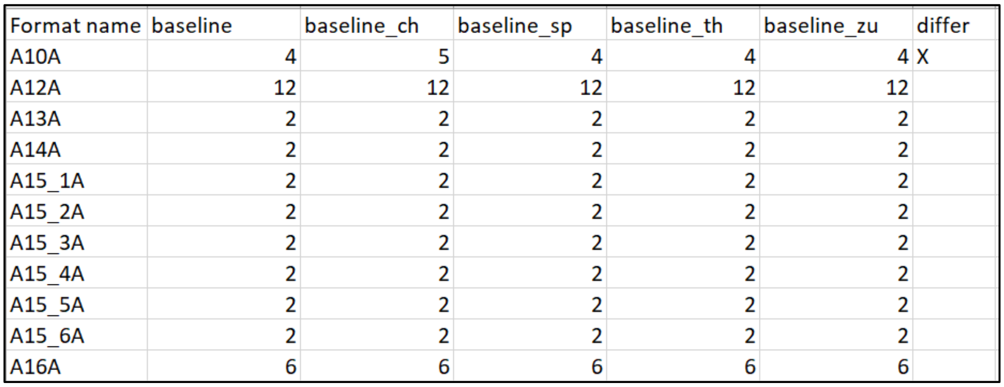
For this study, our surveys were written in English, Spanish, Thai, Chichewa, and Zulu. Therefore, most formats were written in their respective languages. See Figure 7 as an example. For analysis, we need to translate the responses to English. In theory, one would think we would apply the formats from the English surveys to the foreign language surveys. That seems too easy; how do we check this?



A	B	C	
FMTNAME	START	END	baseline_sp
A10A	1	1	Nunca he estado casado/a
A10A	2	2	Casado/a
A10A	3	3	Viudo/a
A10A	4	4	Divorciado/a
A13A	1	1	Sí
A13A	2	2	No
A14A	1	1	Sí
A14A	2	2	No
A15_1A	0	0	Unchecked
A15_1A	1	1	Checked
A15_2A	0	0	Unchecked
A15_2A	1	1	Checked

Figure 7. Baseline Survey in Spanish

Again, I read in each survey and create a SAS data set. Instead of comparing format labels across surveys, I calculate the number of labels for each format in each language, compare them, and set an indicator where the numbers are different. Figure 8 displays the output.



Format name	baseline	baseline_ch	baseline_sp	baseline_th	baseline_zu	differ
A10A	4	5	4	4	4	X
A12A	12	12	12	12	12	
A13A	2	2	2	2	2	
A14A	2	2	2	2	2	
A15_1A	2	2	2	2	2	
A15_2A	2	2	2	2	2	
A15_3A	2	2	2	2	2	
A15_4A	2	2	2	2	2	
A15_5A	2	2	2	2	2	
A15_6A	2	2	2	2	2	
A16A	6	6	6	6	6	

Figure 8. Output between Languages

For cultural reasons, some surveys will need extra questions, or some questions will need an extra category. The real alarm is when one language has less categories than the English version. The person setting up the survey may have left out a response option and it needs to be corrected.

OTHER WAYS THIS CAN BE USED

Your company or organization may not work with SPSS data. However, this code is still helpful when working across different data libraries and format catalogs while combining them into a single library and catalog. One group may define YESNO format as 1 for “Yes” and 0 for “No” while someone uses 2 for “No”. One thing to keep in mind while using this code: do **NOT** delete permanent format catalogs. Leave them alone!

CONCLUSION

This paper is not to invoke paranoia of past projects. It’s to remind you to look at the format catalogs while working across multiple data libraries. By using the techniques describe in this paper, you can quickly find and fix any formatting issues before they become an inconvenience.

ACKNOWLEDGMENTS

I want to thank Kobie O’Brian, Drew Edwards, and Paul Stutzman for reviewing my paper and providing feedback. I also want to thank David Kerr for checking my math. Cheers!

CONTACT INFORMATION

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

Jackie Fitzpatrick
Fred Hutch
jackie@fredhutch.org