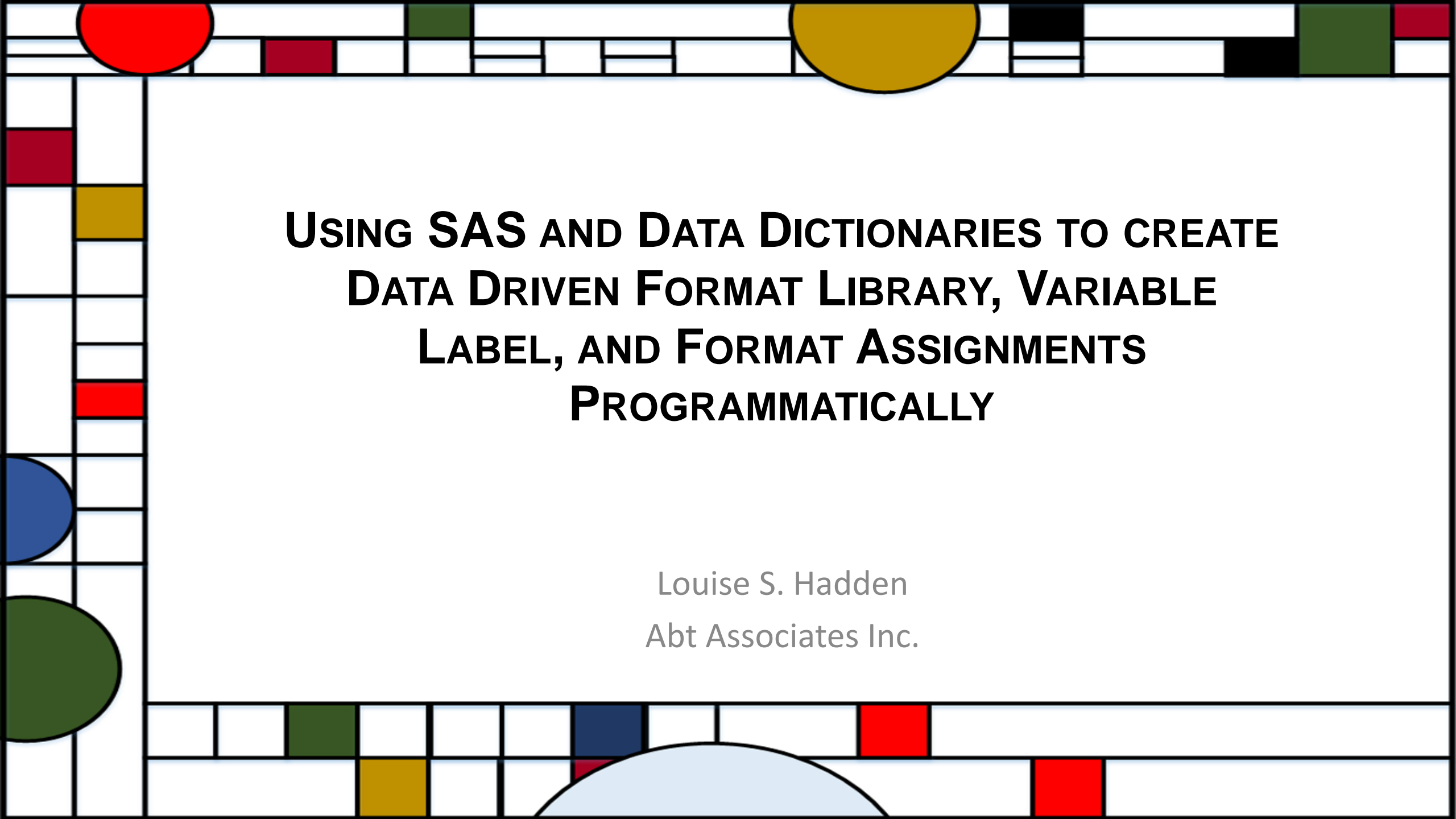




USING SAS AND DATA DICTIONARIES TO CREATE DATA DRIVEN FORMAT LIBRARY, VARIABLE LABEL, AND FORMAT ASSIGNMENTS PROGRAMMATICALLY

Louise Hadden has been using, and loving, SAS since the days of punch cards and computers the size of a not so tiny house. She spends most of her time in support of health policy analytics at Abt Associates Inc., and loves a good SAS reporting challenge. She is also the girl with the SAS tattoo!

A decorative border surrounds the central text. It consists of a grid of squares and rectangles in various colors (red, yellow, green, blue, black, white) and several large semi-circles in red, yellow, blue, and green.

USING SAS AND DATA DICTIONARIES TO CREATE DATA DRIVEN FORMAT LIBRARY, VARIABLE LABEL, AND FORMAT ASSIGNMENTS PROGRAMMATICALLY

Louise S. Hadden
Abt Associates Inc.

INTRODUCTION

- Large scale, multi site study involving COVID testing and other health metrics
- Data dictionaries are used as both documentation and specs for data abstraction from Electronic Medical Records, data derivation and data extraction from REDCAP
- # signs in variable names indicate an unknown number of iterations of a variable

INTRODUCTION

- Data files for the study require variable labels
- Variable labels need to be file specific
- A format library is required for delivery of data
- Format assignment statements are required, and need to be specific to each deliverable file
- Files are received on a monthly basis from sites and data is cumulative

SOLUTIONS

- Ingest data dictionaries to collect metadata allowing identification of groups for processing
- Drive processing with information derived from data dictionaries
- SAS tools used:
 - SAS functions
 - SAS macro facility
 - Ability to write and read text files within a process
 - PROC FORMAT CNTLIN and CNTLOUT

A decorative border surrounds the central text. It features a grid of colored squares in red, maroon, olive green, gold, black, and blue. Large semi-circles in red, gold, blue, and light blue are also integrated into the border design.

DATA DICTIONARY READ IN

LABELS

DATA DICTIONARY

- Data dictionaries in use for the study have multiple tabs and thousands of variables
- Individual variables can be present in one or more of three separate files at two different levels for: pregnant women, post-partum women, and infants
- Individual variables have up to two # signs (iteration flags), appear in a single row in the data dictionary, and may occur 60+ times

DATA DICTIONARY

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW SAS Louise Hadden

J10 :

	A	B	C	D
1	Visit-level file (variables pulled during the data pull that covers the time window in which the visit took place)			
2	4. Virus Testing (Neonate)			
3	These variables describe the characteristics of virus testing received by the neonate at the delivery hospitalization, including dates of tests and test results. For details on iterating variables ending in _NEO#_VTST#, see the Key (Guidance for variables with multiple iterations)			
4	Variable Name	Variable Description	Variable Values	Notes
5			0 = SARS-CoV-2 negative	<Multiple Testing AND Multiple Fetuses> NEO# will iterate with each fetus/newborn iterate with each test performed on that specific fetus/newborn.
6			1 = SARS-CoV-2 positive	
7	COVID_IgM_NEO#_VTST#	First test for IgM for SARS-CoV-2 antibody (during the visit/admission)	2 = No SERUM testing	
8			888 = Missing	
9			999 = Unknown	

Key Revisions Identifiers Event Description Virus Testing **Virus Testing_Neo** ARFI and other Dx Codes Medications_Flu Medic ...

READY 100%

DATA DICTIONARY – READ IN GOALS

- Import information from data dictionaries for different purposes
 - Variable labels
 - Expected variables
 - Value labels (formats)
 - Label, format, retain, attrib statements
 - SAS procedure statements
 - Drive macro processing

DATA DICTIONARY – READ IN CHALLENGES

- Multiple tabs
- Read in specified ranges
- Clean up of special characters (tabs, carriage returns)
- Disaggregation of some fields
- Feedback loop for errors found
- Requirements for two levels
 - Variable information
 - Value Label information

DATA DICTIONARY READ IN

```
*****;  
*** Import Personal Data Dictionary one tab at a time ***;  
*****;  
  
%macro imptabs(tabn=1, tabnm=identifiers, intab=Identifiers,  
startrow=10, endcol=H);  
  
proc import dbms=xlsx out = temp datafile = " \file.xlsx"  
replace;  
                  RANGE="&intab.$A&startrow.:&endcol.999";  
                  getnames=YES;  
  
run;
```

DATA DICTIONARY READ IN

```
data labels&tabn.;
```

```
length label labelstr $ 300 variable_type $ 8;
```

```
set &tabnm (keep=variable_: pw_preg pw_pp inf  
  where=(variable_name ne '  
  and variable_description ne '));
```

```
variable_length=length(variable_name);
```

```
length_flag=(variable_length+7 GT 32);
```

```
label variable_length="Length of Variable"
```

```
length_flag="Variable Length + 7 exceeds 32";
```

DATA DICTIONARY READ IN

```
/* find out the # of iterations in a variable name */  
  
iteration_flag=(indexc(variable_name,'#') gt 0);  
iteration_count=countc(variable_name,'#');  
label iteration_flag="Binary: Variable iterations"  
iteration_count="# of iteration points within  
variable name";
```

DATA DICTIONARY READ IN

```
data labels&tabn.;  
  length label labelstr $ 300 variable_type $ 8;  
  . . .  
  label=catx(":", "&tabnm.", variable_description);  
  labelstr=cats(variable_name, "=", label, "'");
```

ITERATION

- It is a simple task to replace a single iterator in a variable name. It is more complicated when you don't know how many there are or where they are.
- Once we locate the iterator(s), we use the SUBSTR function to replace the iterator. Note that the SUBSTR function is sensitive to the size of the replacement value.
- You can use functions to discover the number of iterations needed, including COUNTC, REVERSE, or ANYNUM.

ITERATION: SIMPLE LOOP

```
%macro do_list1(maxiter=1,suffix=neo);  
%do i=1 %to &maxiter;  
data iter&suffix.1_&i (drop=loc);  
    length variable $ 50 labelstr $ 300;  
    set formats0 (where=(count(variable,"#")=1  
and index(variable,"IDENTIFIER#")>0));  
    *get the first indexed # location;  
    loc=index(variable,"#");  
    substr(variable,loc,1)="&i";  
    labelstr=catt(labelstr," #&i");  
  
run;  
%END;  
%MEND DO_LIST1;
```


ITERATION: COMPLEX LOOP

```
%macro do_list2(maxiter=20,suffix=vtst);  
%if &maxiter le 9 %then %do i=1 %to &maxiter;  
data iter&suffix.1_&i (drop=loc);  
    length variable $ 50 labelstr $ 300;  
    set formats0 (where=(count(variable,"#")=1  
and index(variable,"VTST#")>0));  
    *get the first indexed # location;  
    loc=index(variable,"#");  
    substr(variable,loc,1)="&i";  
    labelstr=catt(labelstr," #&i");  
run;
```

ITERATION: COMPLEX LOOP

```
%if &maxiter gt 9 %then %do;  
%do i=1 %to 9;  
data iter&suffix.1_&i (drop=loc);  
    length variable $ 50 labelstr $ 300;  
    set formats0 (where=(count(variable,"#")=1  
and index(variable,"VTST#")>0));  
    *get the first indexed # location;  
    loc=index(variable,"#");  
    substr(variable,loc,1)="&i";  
    labelstr=catt(labelstr," #&i");  
  
run;  
%END;
```

ITERATION: COMPLEX LOOP

```
%do i=10 %to &maxiter;  
data iter&suffix.1_&i (drop=loc);  
    length variable $ 50 labelstr $ 300;  
    set formats0 (where=(count(variable,"#")=1  
and index(variable,"VTST#")>0));  
loc=index(variable,"#");  
    substr(variable,loc,2)="&i";  
    labelstr=catt(labelstr," #&i");  
  
run;  
%END;  
%END;  
%MEND DO_LIST2;
```

APPLYING THE TECHNIQUE TO CREATE LABELS

```
filename label1 ".\&short._Labels.txt";
%do_list1(maxiter=3,suffix=id);
%do_list2(maxiter=4,suffix=vtst); . . .
data expand_labels;
    set iterid: itervtst: . . . _ ;
run;
data labels;
    length variable $ 32;
    set labels0 (where=(index(variable,"#")=0))
        expand_labels
    (where=(index(variable,"#")=0));
run;
```

LABEL CREATION

```
data tolabel;
  retain VARIABLE_CATEGORY ...;
  file label1 lrecl=400;
  set matchtest (keep=VARIABLE_CATEGORY ...);
  by NUM;
  STATEMENT=compbl(cats(variable, '=' ,
  labelstr, '"));
  if inlabels=1 and inpos=1
  then put statement;
run;
```

LABEL CREATION

```
filename label1 ".\&short._Labels.txt";
filename retain1 ".\&short._retain.txt";
run;

data &outfi. (label="Labeled &short");
  retain
    %include retain1;
  ;
  set &infi.;
  label
    %include label1;
  ;
run;
```

LABEL CREATION

B_P_IN_7v2_Labels.txt - Notepad

File Edit Format View Help

```
IN_DATA_EXTRCT_DT="Mo 7v2: Date of data extraction"  
INF_IDENTIFIER="Mo 7v2: Infant identifier #1"  
IN_DOB="Mo 7v2: Date of birth"  
IN_HR_AC_ENDOCRINE_PL="Mo 7v2: Acute endocrine complications"  
IN_HR_AC_GASTRO_PL="Mo 7v2: Acute gastrointestinal complications"  
IN_HR_AC_HEART_FAIL_PL="Mo 7v2: Acute heart failure diagnosis"  
IN_HR_AC_HEART_PL="Mo 7v2: Acute heart disease diagnosis"  
IN_HR_AC_HEMATO_PL="Mo 7v2: Acute hematological complications"  
IN_HR_AC_LIVER_FAIL_PL="Mo 7v2: Acute liver failure diagnosis"  
IN_HR_AC_NEURO_PL="Mo 7v2: Only Acute Neurologic disease diagnosis"
```

DATA DICTIONARY READ IN FOR FORMATS

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW SAS Louise Hadden

J10 : X ✓ fx

1 Visit-level file (variables pulled during the data pull that covers the time window in which the visit took place)

2 4. Virus Testing (Neonate)

3 These variables describe the characteristics of virus testing received by the neonate at the delivery hospitalization, including dates of tests and test results. For details on iterating variables ending in _NEO#_VTST#, see the Key (Guidance for variables with multiple iterations)

Variable Name	Variable Description	Variable Values	Notes
COVID_IgM_NEO#_VTST#	First test for IgM for SARS-CoV-2 antibody (during the visit/admission)	0 = SARS-CoV-2 negative 1 = SARS-CoV-2 positive 2 = No SERUM testing 888 = Missing 999 = Unknown	<Multiple Testing AND Multiple Fetuses> NEO# will iterate with each fetus/newborn iterate with each test performed on that specific fetus/newborn.

Key Revisions Identifiers Event Description Virus Testing **Virus Testing_Neo** ARFI and other Dx Codes Medications_Flu Medic ...

READY 100%

DATA DICTIONARY READ IN FOR FORMATS

```
*****;  
*** Import Personal Data Dictionary one tab at a time ***;  
*****;  
  
%macro imptabs(tabn=1, tabnm=identifiers, intab=Identifiers,  
startrow=10, endcol=H);  
  
proc import dbms=xlsx out = temp datafile = " \file.xlsx"  
replace;  
           RANGE="&intab.$A&startrow.:&endcol.999";  
           getnames=YES;  
  
run;
```

DATA DICTIONARY READ IN FOR FORMATS

```
data labels&tabn.;
  length variable_name $ 32 variable_values_edited varlabel $ 300 start $ 8
        variable_type $ 8 ;
  set &tabnm (keep=variable_ : pw_preg pw_pp inf
             where=(variable_values ne ' ' or variable_name ne ' '));

/* replace special characters such as tabs with blank and remove extraneous
blanks */
variable_values_edited=translate(variable_values,' ','09'x);
variable_values_edited=translate(variable_values_edited,' ','0A'x);
variable_values_edited=translate(variable_values_edited,' ','0D'x);
variable_values_edited=compbl(variable_values_edited);

/* create start and label variables for a start on building formats */
if variable_type not in('ID','DATE') then do;
  start=scan(variable_values_edited,1,"=");
  varlabel=scan(variable_values_edited,2,"=");
end;
```

DATA DICTIONARY READ IN FOR FORMATS

```
retain _variable_type;  
    if not missing(variable_type) then  
_variable_type=variable_type;  
    else variable_type=_variable_type;  
drop _variable_type;  
  
formatstr=variable_values_edited;  
  
    if variable_name ne '' then  
fmtname=cats(variable_name, '_');
```

DATA DICTIONARY READ IN

DD_QC_Personal_Format_Asmt_edited.xlsx - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW SAS

Calibri 11 A A

General

Clipboard Font Alignment Number Styles Cells Editing

K135

	A	B	C	H	I	J	K	L	M	N	O	P
1	variable_name	format_re	fmtname	varlabel	start	end	hlo	sexcl	eeexcl	ite	i	Notes
119	FLUVX_SEASON	Y	FLUVX_SEASON_	Yes (received influenza vaccine)	1	1		N	N			0 0
120	FLUVX_SEASON	Y	FLUVX_SEASON_	No (unvaccinated)	0	0		N	N			0 0
121	FLUVX_SEASON	Y	FLUVX_SEASON_	Missing	888	888		N	N			0 0
122	FLUVX_SEASON	Y	FLUVX_SEASON_	Unknown	999	999		N	N			0 0
123	FLUVX_PR_SEASON	Y	FLUVX_PR_SEASON_	Yes (received influenza vaccine)	1	1		N	N			0 0
124	FLUVX_PR_SEASON	Y	FLUVX_PR_SEASON_	No (unvaccinated)	0	0		N	N			0 0
125	FLUVX_PR_SEASON	Y	FLUVX_PR_SEASON_	Missing	888	888		N	N			0 0
126	FLUVX_PR_SEASON	Y	FLUVX_PR_SEASON_	Unknown	999	999		N	N			0 0
127	FLUVX_SEASON_DT	Y	FLUVX_SEASON_DT_	[mmddyy10.]	-21914		99999 OF	Y	N			0 0
128	FLUVX_SEASON_DT	Y	FLUVX_SEASON_DT_	Unknown or could not be determined	-21914	-21914		N	N			0 0
129	FLUVX_PR_SEASON_DT	Y	FLUVX_PR_SEASON_DT_	[mmddyy10.]	-21914		99999 OF	Y	N			0 0
130	FLUVX_PR_SEASON_DT	Y	FLUVX_PR_SEASON_DT_	Unknown or could not be determined	-21914	-21914		N	N			0 0
131	COVIDVX1	Y	COVIDVX1_	Yes (received first COVID-19 vaccine)	1	1		N	N			0 0
132	COVIDVX1	Y	COVIDVX1_	No (unvaccinated)	0	0		N	N			0 0
133	COVIDVX1	Y	COVIDVX1_	Missing	888	888		N	N			0 0
134	COVIDVX1	Y	COVIDVX1_	Unknown	999	999		N	N			0 0
135	COVIDVX2	Y	COVIDVX2_	Yes (received second COVID-19 vaccine)	1	1		N	N			0 0
136	COVIDVX2	Y	COVIDVX2_	No (unvaccinated)	0	0		N	N			0 0
137	COVIDVX2	Y	COVIDVX2_	Missing	888	888		N	N			0 0

DD_PERSONAL_FORMAT_ASMT

READY 100%

BUILDING A FORMAT LIBRARY

- Variable Name – variable name
- Format Name – name of the format
- Varlabel – value label
- Start – start of a range or value
- End – end of a range
- Hlo – specialized formats – high, low, other
- Sexcl – exclude the start of a range
- Eexcl – exclude the end of a range

BUILDING A FORMAT LIBRARY

```
data temp;
    d1='01jan1900'd; d2='01jan1960'd; d3=today(); d4='01jan1940'd;
run;
proc print data=temp;
run;
proc print data=temp;
format d1 d2 d3 d4 mmddyy10.;
run;
proc format fmtlib;
    value foo '01jan1900'd='Invalid'
              '01jan1940'd='Still in'
              '01jan1960'd='SAS zero'
              other=[mmddyy10.];

run;
proc print data=temp;
format d1 d2 d3 d4 foo.;
run;
proc format cntlout=foo2;
run;
proc print data=foo2;
run;
```

BUILDING A FORMAT LIBRARY

Obs	d1	d2	d3	d4
1	-21914	0	22475	-7305

Obs	d1	d2	d3	d4
1	01/01/1900	01/01/1960	07/14/2021	01/01/1940

Obs	d1	d2	d3	d4
1	Invalid	SAS zero	07/14/2021	Still in

BUILDING A FORMAT LIBRARY

```
-----  
|          FORMAT NAME: FOO          LENGTH:   10  NUMBER OF VALUES:   4  |  
|    MIN LENGTH:   1  MAX LENGTH:  40  DEFAULT LENGTH:  10  FUZZ: STD  |  
|-----  
| START           | END           | LABEL (VER. V7|V8   14JUL2021:13:08:59) |  
|-----+-----+-----  
|          -21914|          -21914| Invalid |  
|          -7305|          -7305| Still in |  
|              0|              0| SAS zero |  
| **OTHER**    | **OTHER**    | [MMDDYY10.] |  
|-----
```


BUILDING A FORMAT LIBRARY

01.1_Person_CNTLIN.lst - Notepad

File Edit Format View Help

```

                                D L
                                D A A
                                D I T N
                                E G A G
                                C 3 T U
                                S S Y A
                                P P E E
                                F E M F E T E E
                                U F U I D Y X X H
                                Z I L L I P C C L E E P G
                                X T L T E L L O P P E E

1 FOO          -21914          -21914 Invalid  1 40 10 10 1E-12  0  0 N N N
2 FOO          -7305          -7305  Still in  1 40 10 10 1E-12  0  0 N N N
3 FOO          0              0 SAS zero  1 40 10 10 1E-12  0  0 N N N
4 FOO  **OTHER**          **OTHER**  MMDDYY10. 1 40 10 10 1E-12  0  0 N N N OF

```

Ln 44, Col 1

BUILDING A FORMAT LIBRARY

```
01.1_Person_CNTLIN.sas - Notepad
File Edit Format View Help
*****
*** Create CNTLIN file
*****

data personformats_&procmo (keep=variable_name fmtname start end
  label type hlo sexcl eexcl iterated);
  length fmtname $32 type $1 start $14 label $300;
  set person_cntlin (where=(format_req ne 'N') rename=(varlabel=label));
  if variable_type in('CHAR','ID') then type='c';
  else type='n';
  iterated=(index(variable_name,'#')>0);

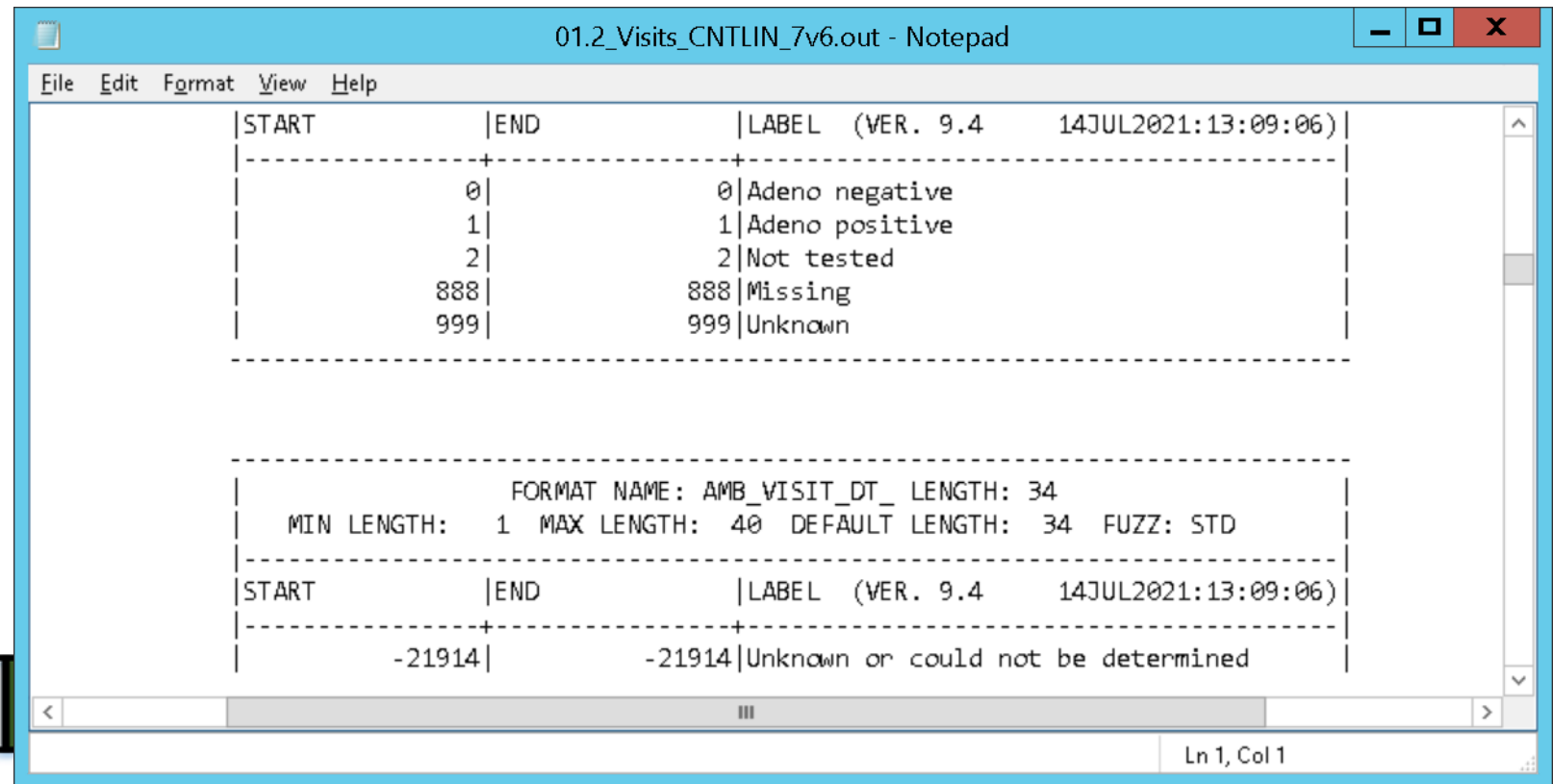
  label fmtname = 'Name of Format'
        start = 'Start of Range for Format'
        end = 'End of Range for Format'
        sexcl = 'Starting value excluded from Range'
        eexcl = 'Ending value excluded from Range'
        label = 'Label for Format'
        type = 'Type of Format'|
        hlo = 'High-Low-Other flag'
        iterated = 'Iterated variable' ;

run;
```

Ln 151, Col 27

BUILDING A FORMAT LIBRARY

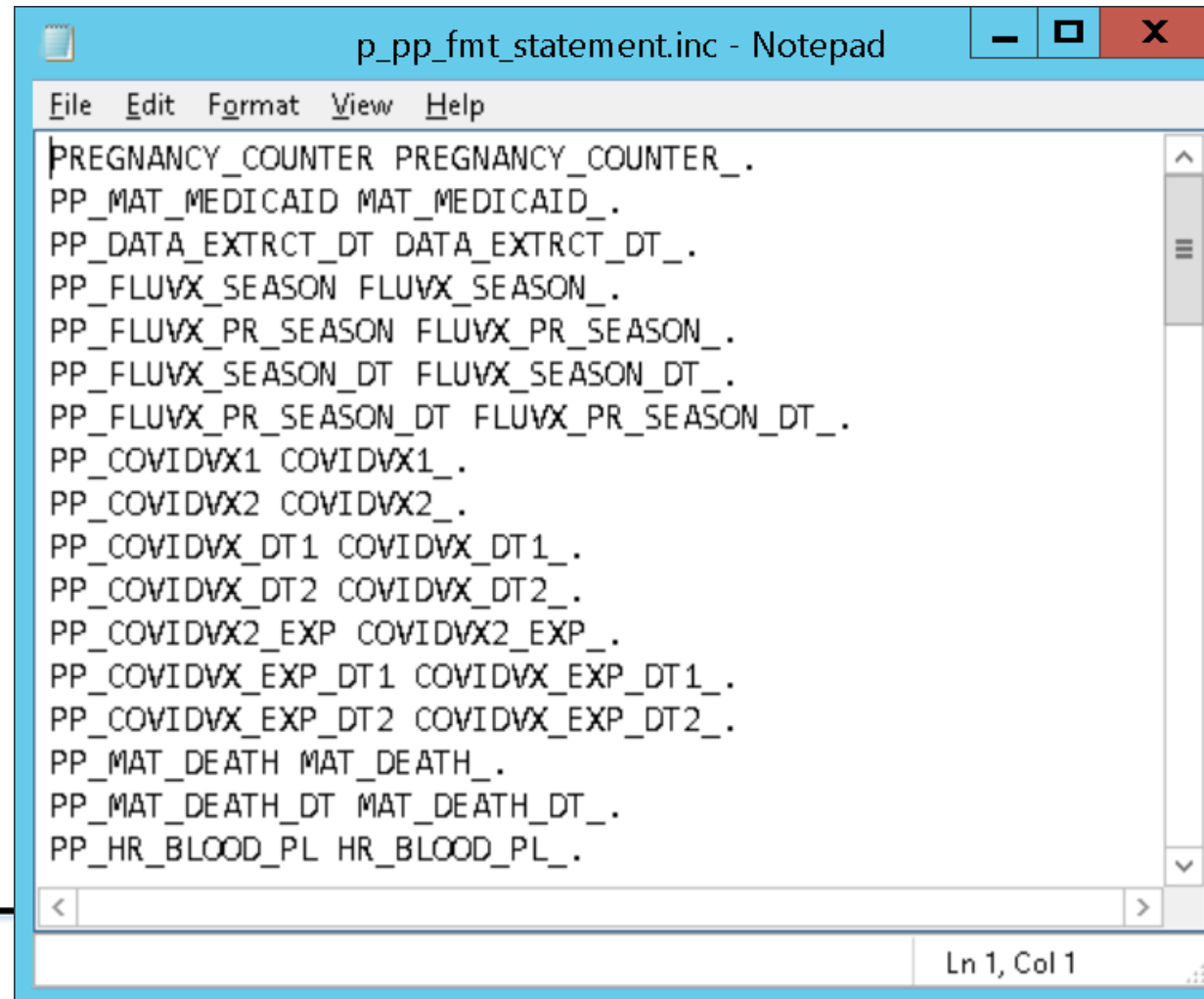
```
proc format library=library.personformats  
cntlin=personformats_&procmo fmtlib ;  
run;
```



DATA DICTIONARY READ IN

```
data fmtstm;  
  length fmtstm $ 80 fmtdot $ 33 variable_name $ 32;  
  file "&outfi._fmt_statement.inc" lrecl=80;  
  set temp (where=(indata ne 0));  
  fmtdot=cats(fmtname, '.');  
  fmtstm=catx(' ', variable_name, fmtdot);  
  put fmtstm;  
run;
```

DATA DICTIONARY READ IN



```
p_pp_fmt_statement.inc - Notepad
File Edit Format View Help
PREGNANCY_COUNTER PREGNANCY_COUNTER_.
PP_MAT_MEDICAID MAT_MEDICAID_.
PP_DATA_EXTRCT_DT DATA_EXTRCT_DT_.
PP_FLUVX_SEASON FLUVX_SEASON_.
PP_FLUVX_PR_SEASON FLUVX_PR_SEASON_.
PP_FLUVX_SEASON_DT FLUVX_SEASON_DT_.
PP_FLUVX_PR_SEASON_DT FLUVX_PR_SEASON_DT_.
PP_COVIDVX1 COVIDVX1_.
PP_COVIDVX2 COVIDVX2_.
PP_COVIDVX_DT1 COVIDVX_DT1_.
PP_COVIDVX_DT2 COVIDVX_DT2_.
PP_COVIDVX2_EXP COVIDVX2_EXP_.
PP_COVIDVX_EXP_DT1 COVIDVX_EXP_DT1_.
PP_COVIDVX_EXP_DT2 COVIDVX_EXP_DT2_.
PP_MAT_DEATH MAT_DEATH_.
PP_MAT_DEATH_DT MAT_DEATH_DT_.
PP_HR_BLOOD_PL HR_BLOOD_PL_.
Ln 1, Col 1
```

A decorative border surrounds the text area, featuring a grid of colored squares and semi-circles in red, yellow, green, blue, and black.

USEFUL BY-PRODUCTS

The same process of iteration and concatenation based on metadata elements is used to create macro calls to create a codebook, a range report and a “missingness” report.

We hope you’ll have some fun iterations with functions with your metadata as well!

USEFUL BY-PRODUCTS

Variable Name	Variable Description	# of Variable values	Missing Value Levels	Missing Value Levels
PR_ASSISTED_REP	Mo 7v2: Was the pregnancy a result of Assisted Reproduction?	3	1	2
PR_DATA_EXTRCT_DT	Mo 7v2: Date of data extraction	1	0	1
PR_FLUVX_SEASON	Mo 7v2: Current season influenza vaccination (August 1st 2020 to May 31st, 2021)	4	0	4
PR_FLUVX_PR_SEASON	Mo 7v2: Prior season influenza vaccination (August 1st 2019 to May 31st, 2020)	4	0	4
PR_FLUVX_SEASON_DT	Mo 7v2: Current season influenza vaccination date (August 1st 2020 to May 31st, 2021)	132	0	132
PR_FLUVX_PR_SEASON_DT	Mo 7v2: Prior season influenza vaccination date (August 1st 2019 to May 31st, 2020)	232	0	232
PR_COMDVX1	Mo 7v2: First COVID-19 vaccination (if vaccine available)?	1	1	0
PR_COMDVX2	Mo 7v2: Second COVID-19 vaccination (if vaccine available)?	3	1	2

USEFUL BY-PRODUCTS

RangeChecks_kpnw_IN_person_7v6.xlsx - Excel

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW SAS Louise Hadden

Clipboard Font Alignment Number Styles Cells Editing

F16 : 0, 1, 999

	A	B	C	D	E	F
1	VARNUM	ANALVAR	LABEL	TYPE	LEN	VALUelist
2	1	STUDY_ID	Mo 7v6: Participant ID	Char	8	Suppressed
3	2	SITE	Mo 7v6: Sub-site or region	Char	4	KPNW
4	3	INF_IDENTIFIER	Mo 7v6: Infant identifier #1	Char	8	Suppressed
5	4	PREGNANCY_COUNTER	Mo 7v6: Counter indicating which pregnancy this is during the study window for each eligible pregnant woman.	Num	8	1, 2
6	5	IN_DOB	Mo 7v6: Date of birth	Num	8	03/01/2020 - 02/28/2021
7	6	IN_SEX	Mo 7v6: Infant sex	Num	8	1, 2, 999
8	7	IN_DATA_EXTRCT_DT	Mo 7v6: Date of data extraction	Num	8	04/21/2021
9	8	IN_LVSYS_DT	Mo 7v6: Approximate date of person leaving the health system	Num	8	03/31/2020 - 01/31/2021
10	9	IN_INF_DEATH	Mo 7v6: Infant death	Num	8	0, 1
11	10	IN_INF_DEATH_DT	Mo 7v6: Date of infant death	Num	8	., 05/16/2020 - 08/22/2020
12	11	IN_HR_BLOOD_PL	Mo 7v6: Blood disorder diagnosis	Num	8	0, 1, 999
13	12	IN_HR_BLOOD_EXCLD_ANEM	Mo 7v6: Blood disorder diagnosis OTHER THAN ANEMIA	Num	8	0, 1, 999
14	13	IN_HR_DIABT_PL	Mo 7v6: Diabetes diagnosis	Num	8	0, 999
15	14	IN_HR_AC_HEART_FAIL_PL	Mo 7v6: Acute heart failure diagnosis	Num	8	0, 999
16	15	IN_HR_HEART_FAIL_PL	Mo 7v6: (Acute or chronic) heart failure diagnosis	Num	8	0, 1, 999
17	16	IN_HR_CAPD_APPST_PL	Mo 7v6: Cardiac arrest diagnosis	Num	8	0, 999

PRINTALL

READY 100%



CONTACT INFORMATION

Louise S. Hadden

Louise_hadden@abtassoc.com

www.linkedin.com/in/louisehadden

<https://communities.sas.com/t5/user/viewprofilepage/user-id/12504>

Code samples provided on request.

A decorative border surrounds the text area, featuring a grid of squares in various colors (red, green, yellow, blue, black) and several large, semi-circular shapes in red, yellow, blue, and green.

ACKNOWLEDGEMENTS

This type of complex programming and processing is a team sport. I could not have created and implemented these techniques on my own.

A very heartfelt thank you to team members Mary Juergens, Jenna Spirt, Nickolas Ferguson, Michael Duckworth and Peiyi Zhang.