PharmaSUG 2018 - Paper QT-18

Quickly create your own format with hundreds of values and other tips

Sergey Sian, IQVIA, Cambridge, MA

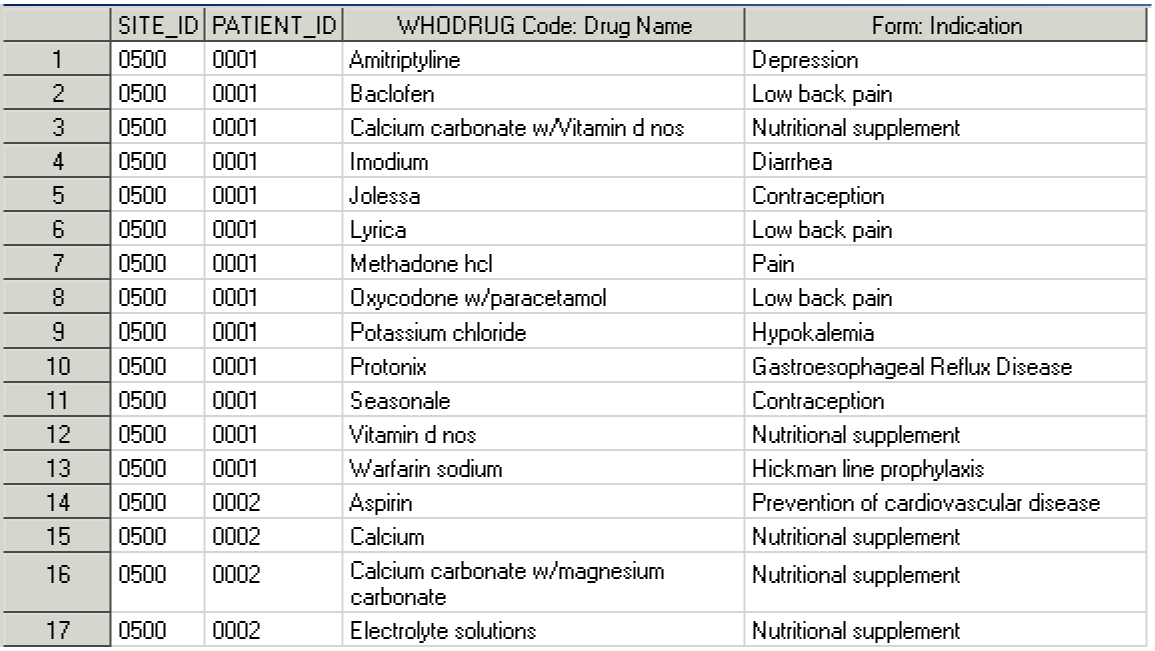
ABSTRACT

Writing SAS® code which automatically defines format(s) with hundreds of values, and using defined format for report is not new. However, this is sometimes a necessary and certainly useful tool to use.

INTRODUCTION

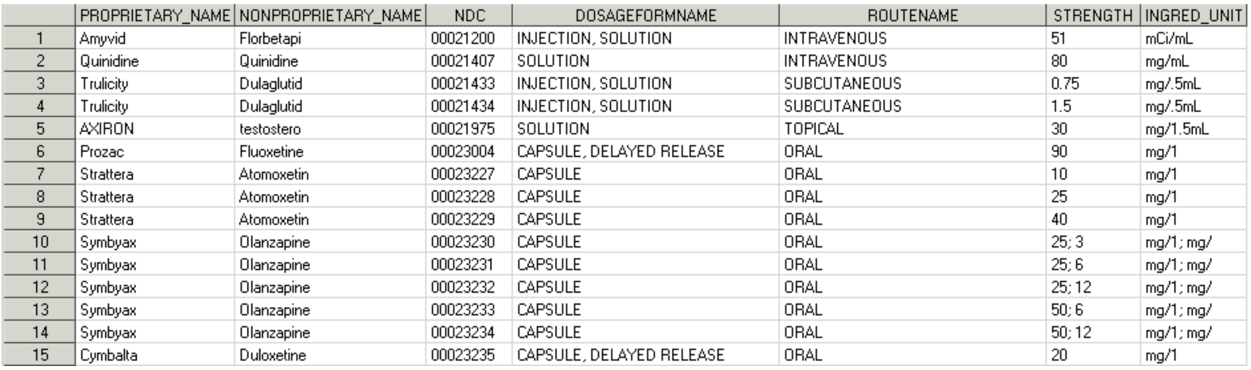
Part I: **create your own format with hundreds of values**.

It's pretty common in late phase research and clinical trials to report concomitant medications used by patients. Usually you get this information from WhoDrug (dictionary of medicinal product information) SAS® dataset, which might look like:



	SITE_ID	PATIENT_ID	WHODRUG Code: Drug Name	Form: Indication
1	0500	0001	Amitriptyline	Depression
2	0500	0001	Baclofen	Low back pain
3	0500	0001	Calcium carbonate w/Vitamin d nos	Nutritional supplement
4	0500	0001	Imodium	Diarrhea
5	0500	0001	Jolessa	Contraception
6	0500	0001	Lyrica	Low back pain
7	0500	0001	Methadone hcl	Pain
8	0500	0001	Oxycodone w/paracetamol	Low back pain
9	0500	0001	Potassium chloride	Hypokalemia
10	0500	0001	Protonix	Gastroesophageal Reflux Disease
11	0500	0001	Seasonale	Contraception
12	0500	0001	Vitamin d nos	Nutritional supplement
13	0500	0001	Warfarin sodium	Hickman line prophylaxis
14	0500	0002	Aspirin	Prevention of cardiovascular disease
15	0500	0002	Calcium	Nutritional supplement
16	0500	0002	Calcium carbonate w/magnesium carbonate	Nutritional supplement
17	0500	0002	Electrolyte solutions	Nutritional supplement

There is also an Rx lookup SAS® dataset (www.fda.gov/Drugs/InformationOnDrugs/, www.drugs.com, www.rxlist.com, etc.) with drugs' names, NDC (National Drug Code), form of usage, strength and other information.



	PROPRIETARY_NAME	NONPROPRIETARY_NAME	NDC	DOSAGEFORMNAME	ROUTENAME	STRENGTH	INGRED_UNIT
1	Amyvid	Florbetapi	00021200	INJECTION, SOLUTION	INTRAVENOUS	51	mCi/mL
2	Quinidine	Quinidine	00021407	SOLUTION	INTRAVENOUS	80	mg/mL
3	Trulicity	Dulaglutid	00021433	INJECTION, SOLUTION	SUBCUTANEOUS	0.75	mg/.5mL
4	Trulicity	Dulaglutid	00021434	INJECTION, SOLUTION	SUBCUTANEOUS	1.5	mg/.5mL
5	AXIRON	testostero	00021975	SOLUTION	TOPICAL	30	mg/1.5mL
6	Prozac	Fluoxetine	00023004	CAPSULE, DELAYED RELEASE	ORAL	90	mg/1
7	Strattera	Atomoxetin	00023227	CAPSULE	ORAL	10	mg/1
8	Strattera	Atomoxetin	00023228	CAPSULE	ORAL	25	mg/1
9	Strattera	Atomoxetin	00023229	CAPSULE	ORAL	40	mg/1
10	Symbyax	Olanzapine	00023230	CAPSULE	ORAL	25; 3	mg/1; mg/
11	Symbyax	Olanzapine	00023231	CAPSULE	ORAL	25; 6	mg/1; mg/
12	Symbyax	Olanzapine	00023232	CAPSULE	ORAL	25; 12	mg/1; mg/
13	Symbyax	Olanzapine	00023233	CAPSULE	ORAL	50; 6	mg/1; mg/
14	Symbyax	Olanzapine	00023234	CAPSULE	ORAL	50; 12	mg/1; mg/
15	Cymbalta	Duloxetine	00023235	CAPSULE, DELAYED RELEASE	ORAL	20	mg/1

Usually NDC is a 10-digit code, in this example it’s 8-digits for simplicity. Please, notice that drug Symbyax (Olanzapine) in lines 10 through 14 has different NDCs depending on its strength. Say, you want to know what are the most common deceases (indications) treated by certain drug (on NDC level). You need to define your own format like:

```
Proc format ;
  Value $drugtrt
   “01214774” = “Analgesic”
   “01210657” = “Analgesic”
   “01439654” = “Antidepressive”
   “00040058” = “Anxiety”
   “00543532” = “Anxiety”
Etc.
;
quit ;
run ;
```

The $drugtrt. format is supposed to have hundreds of values, so you likely will want to automate it.

Step 1: sort out and merge WhoDrug and Rx_lookup SAS® datasets by DRUG_NAME. Save results in pharmsug.rxlookup dataset.

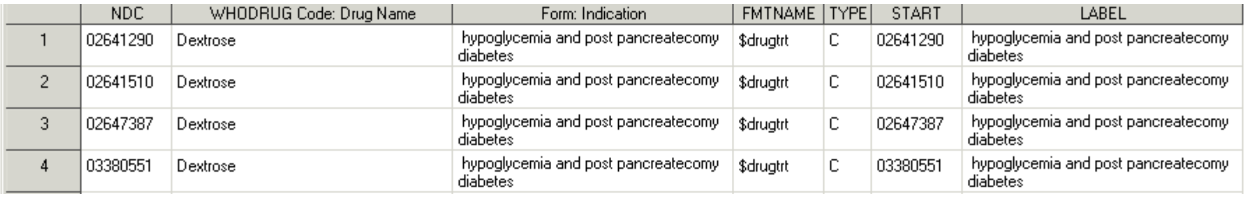
Step 2: create a “drug_class_treat_1” SAS® dataset as:

```
data drug_class_treat_1 ;
     set pharmsug.rxlookup ;
     retain fmtname "$drugtrt" type "C";     /* Type "C" - for Character */
     start = ndc ;
     label = FORM__INDICATION ;
     output;
run;
```

Step 3: Delete possible duplicate values in “ndc” variable:

```
proc sort data = drug_class_treat_1 out = drug_class_treat(where = (NDC ne ' ')) nodupkey ;
  by ndc ;
run ;
```

After step 3 “drug_class_treat” SAS® dataset will look like:



	NDC	WHODRUG Code: Drug Name	Form: Indication	FMTNAME	TYPE	START	LABEL
1	02641290	Dextrose	hypoglycemia and post pancreatecomy diabetes	$drugtrt	C	02641290	hypoglycemia and post pancreatecomy diabetes
2	02641510	Dextrose	hypoglycemia and post pancreatecomy diabetes	$drugtrt	C	02641510	hypoglycemia and post pancreatecomy diabetes
3	02647387	Dextrose	hypoglycemia and post pancreatecomy diabetes	$drugtrt	C	02647387	hypoglycemia and post pancreatecomy diabetes
4	03380551	Dextrose	hypoglycemia and post pancreatecomy diabetes	$drugtrt	C	03380551	hypoglycemia and post pancreatecomy diabetes

Step 4: create the $drugtrt. format which can be saved (in a format catalog in which you want to store it) permanently in defined directory or temporarily in “work” directory as in below code:

```
proc format library = work
    cntlin = drug_class_treat ;
run;
```

Please note that “cntlin” option in proc format allows to create a format from a SAS® dataset that contains value information (called a control data set). After this procedure you’ll get in LOG window:

```
2203  proc format library = work
2204        cntlin = drug_class_treat ;
NOTE: Format $DRUGTRT has been output.
2205  run;
NOTE: PROCEDURE FORMAT used (Total process time):
    real time           0.01 seconds
    cpu time            0.01 seconds
NOTE: There were 5931 observations read from the data set WORK.DRUG_CLASS_TREAT.
```

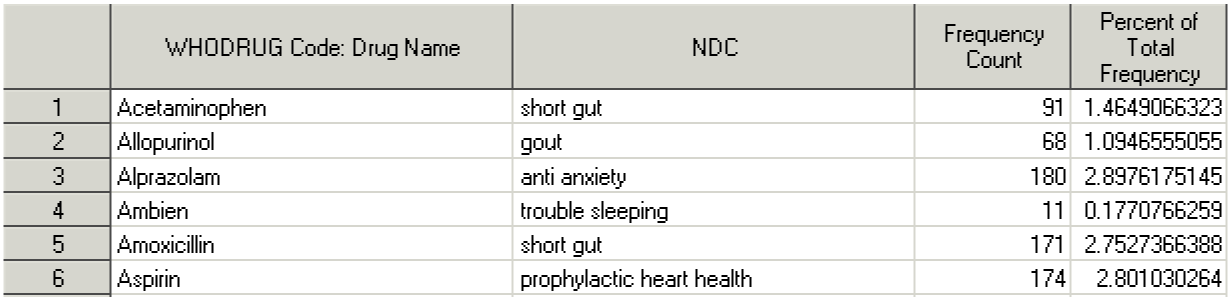
To look at all values defined in “$DRUGTRT.” format run:

```
title 'Custom $ DRUGTRT. format' ;
proc format library = work  ;
    select $drugtrt ;
run ;
```

Finally, to accomplish our ultimate goal of finding out what the most common deceases (indications) are treated by certain drug (on NDC level):

```
proc freq data = pharmsug.rxlookup noprint ;
tables WHODRUG_CODE__DRUG_NAME*ndc / missing list out =
WHODRUG_CODE_DRUG_NAME_ndc ;
format ndc $drugtrt. ;
title "Distribution on 'WHODRUG_CODE__DRUG_NAME*ndc' " ;
title2 "The 'pharmsug.rxlookup' dataset" ;
run ;
```

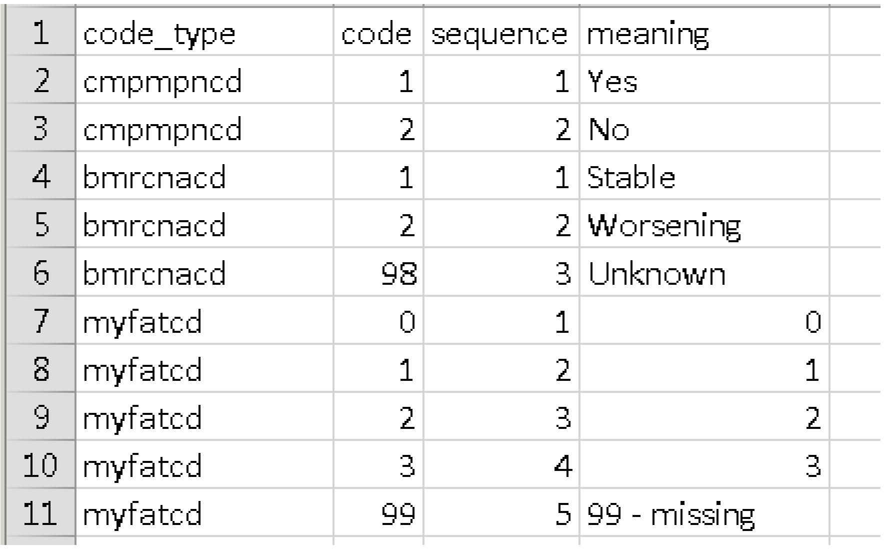
The results are:



	WHODRUG Code: Drug Name	NDC	Frequency Count	Percent of Total Frequency
1	Acetaminophen	short gut	91	1.4649066323
2	Allopurinol	gout	68	1.0946555055
3	Alprazolam	anti anxiety	180	2.8976175145
4	Ambien	trouble sleeping	11	0.1770766259
5	Amoxicillin	short gut	171	2.7527366388
6	Aspirin	prophylactic heart health	174	2.801030264

Part 2. **Define hundreds of formats from “code” extract from EDC (Electronic Data Capture) system**.

EDC system keeps and allows to extract various files (baseline, follow up, medical history, etc.) for further analysis. One of the extracted file is “code” which has the list of possible values for every field in all files. It looks like:

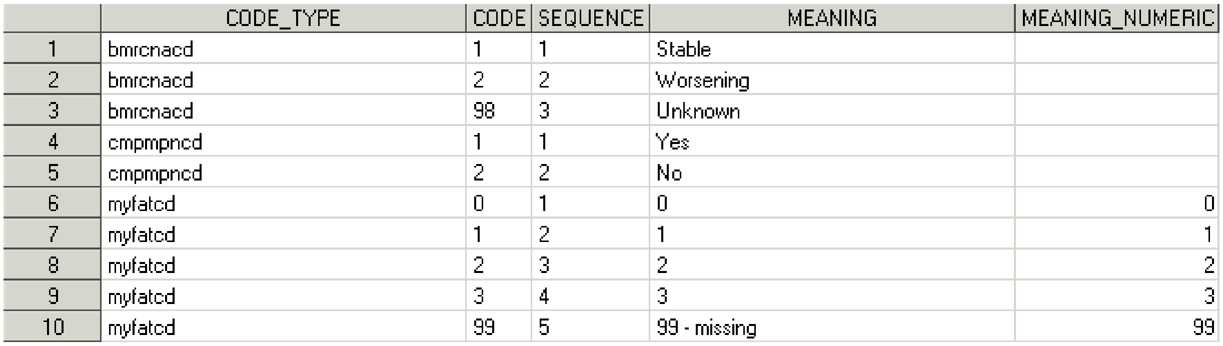


1	code_type	code	sequence	meaning
2	cmpmpncd	1	1	Yes
3	cmpmpncd	2	2	No
4	bmrcnacd	1	1	Stable
5	bmrcnacd	2	2	Worsening
6	bmrcnacd	98	3	Unknown
7	myfatcd	0	1	0
8	myfatcd	1	2	1
9	myfatcd	2	3	2
10	myfatcd	3	4	3
11	myfatcd	99	5	99 - missing

Step1: Read in CSV 'codes' file and sort out created SAS® dataset.

```
data csv_file1 ;
   length code_type my_code_type $50 code sequence $2 meaning my_meaning $135 meaning_numeric
8 meaning_1 $50 ;

   infile  "&pass_to_the_folder.\&csv_file..csv" missover dsd lrecl= 500 ;

   input my_code_type $ code $ sequence $ my_meaning $;
   meaning = compress(my_meaning,"") ;
   my_length = length(my_code_type) ;

   /* If "code_type" ends with "number" replace it  */
   /* with "_" since format can't end with "number" */
   if substr(my_code_type,my_length,1) in ('0','1','2','3','4','5','6','7','8','9')
      then do ;
        code_type = tranwrd(my_code_type,'0','_') ;
                        code_type = tranwrd(my_code_type,'1','_') ;
                        code_type = tranwrd(my_code_type,'2','_') ;
                        code_type = tranwrd(my_code_type,'3','_') ;
                        code_type = tranwrd(my_code_type,'4','_') ;
                        code_type = tranwrd(my_code_type,'5','_') ;
                        code_type = tranwrd(my_code_type,'6','_') ;
                        code_type = tranwrd(my_code_type,'7','_') ;
                        code_type = tranwrd(my_code_type,'8','_') ;
                        code_type = tranwrd(my_code_type,'9','_') ;
   end ;
   else code_type = my_code_type ;

   meaning_1 = compress(meaning," ") ;
   if substr(meaning_1,1,2) in ('99') then meaning_numeric = 99 ;
   else if substr(meaning_1,1,1) in ('0','1','2','3','4','5','6','7','8','9') then
        meaning_numeric = put(meaning_1, 8.) ;
   else meaning_numeric = . ;

   drop my_meaning my_code_type my_length meaning_1 ;
   if  _n_ ge 2 then output; run;
Proc sort data = csv_file1 out = codes nodupkey ;
     by code_type sequence;
run;
```

The above SAS® data step “identifies” if variable can/will have a “numeric” or “character” format. The work.codes SAS® dataset looks like:



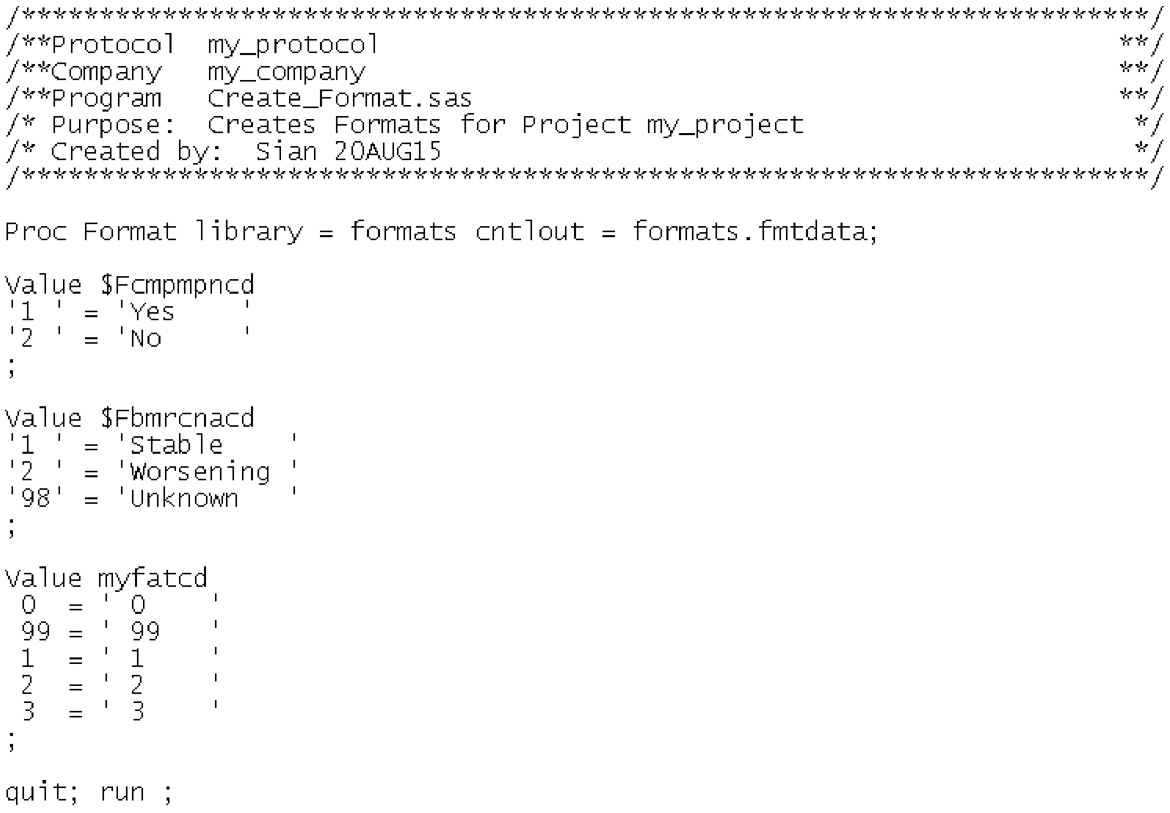
	CODE_TYPE	CODE	SEQUENCE	MEANING	MEANING_NUMERIC
1	bmrcnacd	1	1	Stable	
2	bmrcnacd	2	2	Worsening	
3	bmrcnacd	98	3	Unknown	
4	cmpmpncd	1	1	Yes	
5	cmpmpncd	2	2	No	
6	myfatcd	0	1	0	0
7	myfatcd	1	2	1	1
8	myfatcd	2	3	2	2
9	myfatcd	3	4	3	3
10	myfatcd	99	5	99 - missing	99

Step 2: Create the makeformat.sas program, which can be run to create a FORMAT catalog in “formats” folder.

```
Data _null_;
     set codes end=last ;
     by code_type;

file "&path_to_the_folder.\makeformat.sas" ;

if _n_=1 then do;
             put '/***************************************************************************/';
             put "/**Protocol &name_of_the_protocol.                              **/";
             put "/**Company  &name_of_the_company.                               **/";
             put '/**Program  Create_Format.sas                                   **/';
             put "/* Purpose: Creates Formats for Project &name_of_the_project.         */";
```

```
            put "/* Created by: &name-of_the_programmer. &sysdate.                    */";
            put '/*******************************************************************/';
    put ;

            put 'Proc Format library = formats cntlout = formats.fmtdata; ';
            put ;
end;

if first.code_type then do;
    if substr(compress(meaning," "),1,2) in ('99') then do ;
               put 'Value ' @7 code_type  ;
         put  @2 code @4 " = '"  @9 meaning_numeric @14 "'";
        end ;
    else if length(compress(meaning," ")) <= 2 AND substr(compress(meaning," "),1,1)
         in ('0','1','2','3','4','5','6','7','8','9') then do ;
                   put 'Value ' @7 code_type  ;
             put  @2 code @4 " = '"  @9 meaning_numeric @14 "'";
        end ;
            else do ;
                 put 'Value $F' @9 code_type  ;
           put  "'"@2 code @4 "' = '"  @9 meaning @145 "'";
        end ;
end;
else do ;
     if substr(compress(meaning," "),1,2) in ('99') then put @2 code @4 " = '"
        @9 meaning_numeric @14 "'";
     else if length(compress(meaning," ")) <= 2 AND substr(compress(meaning," "),1,1)
        in ('0','1','2','3','4','5','6','7','8','9') then put @2 code @4 " = '"
        @9 meaning_numeric @14 "'";
     else put  "'"@2 code @4 "' = '"  @9 meaning @145 "'";
end ;

if  last.code_type then do;
    put ';' ;
    put;
end;
if last then put 'quit; run ; ' ;
run;
```

The automatically created “makeformat.sas“ program looks like:



```
/************************************************************************/
/**Protocol  my_protocol                                               **/
/**Company   my_company                                                **/
/**Program   Create_Format.sas                                         **/
/* Purpose:  Creates Formats for Project my_project                     */
/* Created by:  Sian 20AUG15                                            */
/************************************************************************/

Proc Format library = formats cntlout = formats.fmtdata;

Value $Fcmpmpncd
'1 ' = 'Yes    '
'2 ' = 'No     '
;

Value $Fbmrcnacd
'1 ' = 'Stable    '
'2 ' = 'Worsening '
'98' = 'Unknown   '
;

Value myfatcd
 0  = ' 0    '
 99 = ' 99   '
 1  = ' 1    '
 2  = ' 2    '
 3  = ' 3    '
;

quit; run ;
```

CONCLUSION

The above code shows how to automatically define format(s) with hundreds of values either using the WhoDrug (dictionary of medicinal product information) and Rx lookup SAS® datasets, or from a “code” CSV file extracted from EDC (Electronic Data Capture) system.

The described approach will take care of 2 or 2000 distinct formats without error prone typing and time consuming “copy-and-paste” procedure.

ACKNOWLEDGMENTS

A special thank you to David Franklin, Sr. Manager statistical programming, who has provided valuable suggestions and inputs to my paper. I’m also grateful to Greg Tebbenkamp, Director of statistical programming, whose support and encouragement has helped me to write this paper. SAS® code for part 2 was implemented together with my colleague Evguenia Jilinskaia – a Sr. statistical programmer at IQVIA.

CONTACT INFORMATION

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

Name: Sergey Sian
Enterprise: Quintiles
Address: 201 Broadway
City, State ZIP: Cambridge, MA 02139, USA
Work Phone: 1-617-715-6854

Fax: 1-617-621-1620
E-mail: sergey.sian@quintilesims.com
Web: www.iqvia.com