**Paper 3704-2019**

# Procedure Code Explosion: Applying SAS® Macro Language, HASH Tables, and PROC FCMP to Manage Large Volumes of Procedure Codes in Administrative Data Research

Matthew Keller, MS; Dustin Stwalley, MS; Katelin B. Nickel, MPH
Washington University in St. Louis, School of Medicine

## ABSTRACT

Managing a large volume of diagnosis and procedure codes is a common task in health services research that uses administrative data. One such collection of procedure codes is provided by the Centers for Disease Control and Prevention's National Healthcare Safety Network (NHSN), which tracks 39 unique categories of surgical procedures for the purposes of healthcare-associated infection surveillance.[1] To identify these surgical categories in a dataset, a simple programming method would be to write 39 If-Else If statements referencing a single variable containing procedure codes. The issue with this approach is that there are over 1,100 Current Procedural Terminology (CPT) codes in the most recent NHSN release. Our approach to reducing the written text in a program is to use a macro to write If statements for the 39 variables using a user-defined function. The function searches the inputted CPT code in a hash table holding the NHSN-specific CPT codes. The major benefits of this approach is reducing data entry errors that could be introduced when manually entering procedure codes into the program, as well as making a more concise program. Additionally, the process is scalable and has been applied to many similar challenges in our research.

## INTRODUCTION

Health services research takes advantage of large administrative databases, including hospital discharge and insurance claims databases, to answer a broad range of health-related research questions. Researchers in this field deploy a wide range of programming, analytic and data management techniques to organize and quantify their data. Diagnosis and procedure codes are extracted from these databases to identify cohorts of interest and define variables needed for analysis. Managing these codes can be cumbersome as the programmer needs to both keep track of coding systems (Current Procedural Terminology [CPT-4] or International Classification of Diseases [ICD-9-CM, ICD-10-PCS] for procedures; ICD-9-CM, ICD-10-CM for diagnoses) depending on the database and timeframe of interest, and handle a large number of individual codes within a coding system to define conditions and procedures of interest. This is an even more unwieldy but important task with the proliferation of codes since the introduction of ICD-10.

One large collection of procedure codes we use in our research is provided by the Centers for Disease Control and Prevention's National Healthcare Safety Network (NHSN), which tracks 39 unique categories of surgical procedures for the purposes of healthcare-associated infection surveillance. Our programming approach for identifying the NHSN surgical procedure categories was to minimize the manual handling of procedure codes. The datasets in the SET statement are outpatient and inpatient medical insurance claims. The variable PROC1 contains CPT codes; NHSN tracks surgical procedures using ICD-10-PCS (formerly ICD-9-CM) procedure codes as well, but for the purposes of this paper we will present our CPT code data management. The NHSN surgical category abbreviations and CPT codes for the surgical procedures are downloaded from the CDC's website as a csv file and imported into a SAS dataset as two variables. This dataset is NHSN_CPT_2018, and loaded into a hash table for quick and easy lookup of the CPT codes and corresponding surgical category abbreviation.

The process starts with using macro language that allows for repetitive programming code to be written with minimal effort.[2] Next, the FCMP procedure is utilized to store a user-defined function that will accept the user's abbreviation and the value in PROC1 and compare these as a composite key to the two variables in the NHSN_CPT_2018 hash table. In the example macro code below, a simple list of the NHSN abbreviations is iterated using a Do While loop to produce 39 variables. The %LET statements and %SCAN function automatically keep the programming code running without manually entering an iterator value which allows for easier maintenance:

```
%macro create_nhsn_categories();
```

```
%let nhsnlist = aaa amp appy avsd bili brst htp vhys csec thyr
cbgb cbgc ktp cea chol fusn fx hpro vshn card
xlap ltp prst sple thor sb pvby rec pace hyst
her gast cran colo ovry neph kpro neck lam;

data claims_nhsn_categories;
format prefix $4.;
set outpatient_claims inpatient_claims;

%let j = 1;
%let nhsnvar = %scan(&nhsnlist, &j);
%do %while(&nhsnvar ne);

      prefix = %upcase("&nhsnvar");
      nhsn_&nhsnvar = is_nhsncpt(prefix, proc1);

      %let j = %eval(&j+1);
      %let nhsnvar = %scan(&nhsnlist, &j);

%end;
run;

%mend create_nhsn_categories;
```

Each iteration of the macro list creates one variable corresponding to the supplied NHSN abbreviation. The function, called IS_NHSNCPT, will accept the individual prefixes from the macro list and the CPT code found in PROC1. If the composite key created from these two values in the claims dataset matches the composite key created from the two variables in the NHSN_CPT_2018 dataset, then the function returns 1 to the corresponding NHSN surgical category variable, otherwise it returns 0. It is important to point out additional details. First, the order of the function inputs must match the order of the variables in the NHSN_CPT_2018 dataset. Second, they must match exactly including case sensitive and format. To control for this the %UPCASE function is used, as well as an explicit format for the abbreviation. The CPT codes and abbreviations found in the NHSN_CPT_2018 dataset were imported into SAS from the NHSN website and the function was built to match the structure of the dataset. See the supplied references below for a more detailed instruction on using hash tables[3] and the FCMP procedure.[4, 5] The function referenced in the macro above is written here:

```
proc fcmp outlib=work.functions.isin;
function is_nhsncpt(procedure_abbrv $, procedure_code $);
format procedure_abbrv $4.;

declare hash nhsncpt(dataset: 'u.nhsn_cpt_2018');
rc = nhsncpt.definekey('procedure_abbrv', 'procedure_code');
rc = nhsncpt.definedata('procedure_abbrv', 'procedure_code');
rc = nhsncpt.definedone();
rc = nhsncpt.find();

if rc = 0 then return(1);
else return(0);
endsub;

run;
```

```
options cmplib=(work.functions.isin);
```

## CONCLUSION

The combination of saving procedure codes in a lookup table and referencing them through hashing in the FCMP procedure has proven to be a valuable technique for us and is preferable over repetitive If-Else If statements for many reasons. First, reducing tedious manual data entry and the possibility of mistyping codes is always welcome. With our approach, programming code maintenance is also simplified; when CDC releases its annual list of procedure codes, the files simply need to be imported as SAS datasets rather than be reviewed manually for insertions and deletions in order to modify If statements. By storing the procedure codes "behind the scenes" in separate datasets rather than in the programming code itself, it is easier to follow the programming logic and prevents inadvertent manipulation of the procedure codes.

Macro language makes the process more flexible as it allows for adding inputs such as changing the database that contains the raw claims. Macro language also makes the process expandable by adding error-handling capabilities, which is important since the programming code is supplied to a team of programmers, and additional procedure coding systems. We are currently scaling up the NHSN procedure identification by expanding the Do While loop to incorporate ICD-9-CM and ICD-10-PCS codes which adds more than 9,000 additional codes. In future, the process will be expanded further to include non-NHSN procedures.

## REFERENCES


1. Centers for Disease Control and Prevention. 2017. "National Healthcare Safety Network (NHSN)." Accessed October 12, 2018. https://www.cdc.gov/nhsn/index.html.
2. Cohen, J.J. 2012. "A Tutorial on the SAS Macro Language." *Proceedings of the SAS Global Forum.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings12/249-2012.pdf.
3. Dorfman, P. 2016. "Fundamentals of The SAS® Hash Object." Available at https://analytics.ncsu.edu/sesug/2016/HOW-195_Final_PDF.pdf.
4. Carpenter, A. L. 2013. "Using PROC FCMP to the Fullest: Getting Started and Doing More." *Proceedings of the SAS Global Forum.* Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings13/139-2013.pdf.
5. McNeill, B., Henrick, A., Whitcher, M., Mays, A. 2018. "FCMP: A Powerful SAS® Procedure You Should Be Using." *Proceedings of the SAS Global Forum.* Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2125-2018.pdf.


## ACKNOWLEDGMENTS


We would like to thank Margaret A. Olsen PhD, MPH for her comments on the project as well as her continued support at Washington University in St. Louis, School of Medicine. The Center for Administrative Data Research is supported in part by the Washington University Institute of Clinical and Translational Sciences Grant UL1 TR002345 from the National Center for Advancing Translational Sciences (NCATS) of the National Institute of Health (NIH), Grant Number R24 HS19455 through the Agency for Healthcare Research and Quality (AHRQ).


## CONTACT INFORMATION

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


Matthew Keller
kellermr@wustl.edu