SAS® GLOBAL FORUM 2021

1063-2021

How Sick Is My Cohort Of Patients? A General Approach to Identify Chronic Conditions

Patricia Ferido, Leonard D. Schaeffer Center for Health Policy & Economics; Laura Gascue, Leonard D. Schaeffer Center for Health Policy & Economics; Patricia St. Clair, Leonard D. Schaeffer Center for Health Policy & Economics

ABSTRACT

Analysis of medical claims data frequently requires identifying a disease cohort or controlling for chronic conditions. The Chronic Conditions Warehouse (ccwdata.org) applies validated algorithms for discerning multiple chronic conditions in Medicare claims data, and CMS provides the results of their application with Medicare claims data if requested. The algorithms use a combination of diagnosis and procedure codes and selection rules, which vary among conditions. For example, to eliminate rule-out diagnoses, the algorithm for diabetes requires one set of diagnosis codes observed on any inpatient or skilled nursing facility claim, or two claims at least a day apart on a physician claim. This presentation discusses a SAS® macro that applies CCW-like rules to any data set either from insurance claims, or electronic health records (EHR) containing a full picture of diagnoses and procedures from patient medical visits. The macro package includes the CCW-validated algorithms (the default option), but also has the flexibility for the user to apply the algorithm to a different set of diagnoses and procedures. The user can either implement variations of the CCW definitions or identify entirely new conditions, so long as they can be implemented using diagnosis or procedure codes, claim types and CCW-like rules. The macro package is described, including how it's applied to medical claims from a private insurer.

INTRODUCTION

The detailed information found in claims data and electronic medical records offers the promise of insights into health and health care through secondary analysis in observational studies. These types of studies often select a disease cohort for analysis, and nearly always need to control for comorbid conditions. However, analysts do not always apply the same set of rules for identifying health conditions in claims data, making it difficult to effectively compare results across studies. The Chronic Conditions Data Warehouse (ccwdata.org) provides standard validated methods for defining a variety of health conditions in Medicare claims. Good research practice suggests that incorporation of standard definitions of health condition measures leads to more comparable results, comparing apples to apples, so to speak.

The CCW definitions have limitations. They represent a broad but not complete set of conditions, that is, not all health conditions of interest are included. Though validated, they do not always perform perfectly. Studies have examined the effectiveness of CCW algorithms in identifying conditions, with varying results depending on the particular disease [Gorina 2011, St.Clair 2017]. However, the definitions tend to be broad and provide specific algorithms, with cited validation studies. Before applying these definitions one should verify that they perform reasonably for one's study. This paper does not address the validation of conditions, which is beyond its scope.

Medicare claims data from beneficiaries under the Medicare Fee-For-Service (FFS) plans requested from CMS may include the chronic conditions indicators. These provide mid-year and end-year flags indicating whether a condition has been observed for a particular individual. CCW bases its algorithms on diagnosis and procedure codes in claims, the settings in which they were reported, and sometimes their frequency. The measures also include the earliest date of indication. Applying the CCW algorithms to other datasets (e.g. private claims data) requires an analyst to develop the flags themselves.

We present a package that facilitates the application of CCW algorithms to any claims data, or other source of comprehensive health information like EHR, structured, or restructured. The package combines Microsoft Excel spreadsheets to provide specifications to define conditions and SAS macros to apply those definitions to claims. The package includes the CCW definitions, but also allows the flexibility to define new sets of codes and CCW-like rules, either for additional conditions or to tailor a CCW definition for an analysis. The resulting dataset includes monthly condition flags, as opposed to the two flags (mid-year and end-year) that are included in the CMS CCW data files.

DESCRIPTION

PACKAGE COMPONENTS

The package includes the following components:

- CCW definition files in SAS and csv format
- Main macro function to run the algorithm with user provided options (%idcond)
- Auxiliary program to read the excel file for customized versions
- Auxiliary macro and program to process enrollment information and conditions codes
- Program template to run the %idcond macro (input_program.sas)

The diagram below broadly follows the steps of package.

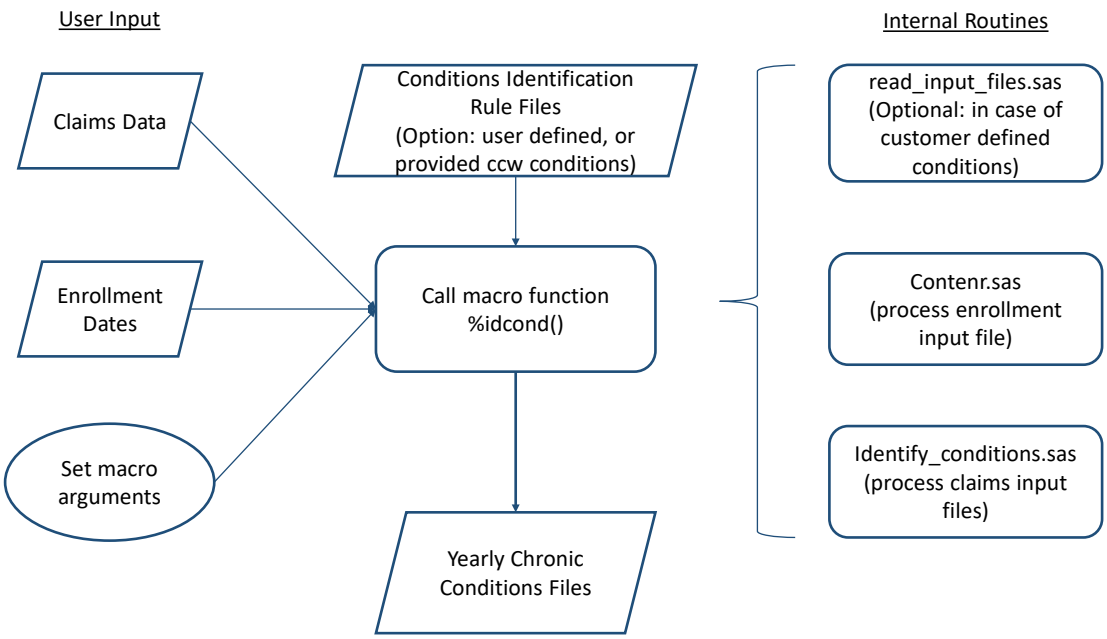

Figure 1. Diagram of Package Structure

INPUT FILES

The %idcond() macro in the center of the above diagram, by default will run the CCW algorithms as defined. In order to run the health conditions identification macro, you need to pass two SAS datasets: a dataset with all patients' diagnoses and procedures, with the date they were assigned, and a dataset with a unique entry per patient specifying the enrollment period for each patient. A comprehensive set of diagnoses is expected for the reference period.

Alternatively, the algorithm can be customized by adding new conditions or adjusting the rules of the pre-defined conditions. Any adjustment to the algorithm has to be entered in the csv template files provided. In this case, instead of the SAS datasets with the CCW definitions, the macro will read the set of csv files defining the new rules. The template csv files contain the default CCW conditions.

Claims Input

The claims input requires a unique patient identifier, the date of each claim, a list of ICD-9 or ICD-10 codes by code order, and claim type or location of claim. Following the organization of Medicare data, the claim type will identify whether the diagnosis was given in an inpatient, outpatient, skilled nursing facility, home health agency, or carrier setting. Private claims data may not be organized into these claim type categories, in which case custom claim types can be used or source data can be coded to fit the Medicare categories. The order in which diagnoses are given is also important, as some algorithms require a valid diagnosis code to be the primary or secondary code on a claim. The table below summarizes the necessary structure for the claims input file, with asterisked variables noting requirements.

Variable Type	Standardized Variable Names	Format
Unique patient identifier*	No standard name required	No standard format
ICD-9 Diagnosis Codes	Icd9dx1-icd9dx[max]*	Character*
ICD-10 Diagnosis Codes	Icd10dx1-icd10dx[max]*	Character*
ICD-9 Procedure Code	Icd9prcdr1-icd9prcdr[max]*	Character*
ICD-10 Procedure Code	Icd10prcdr1-icd10prcdr[max]*	Character*
HCPCS Procedure Codes	Hcpcs1-hcpcs[max]*	Character*
Claim date*	Claim_dt*	Date*
Claim type*	Claim_type*	Character*

Table 1. Claims Input Requirements

Enrollment Input

Each condition is associated with a reference period in which the algorithm will look for a valid claim. Enrollment data for each beneficiary is required to ensure that data is available during the entire reference period. The package takes as valid input two formats of enrollment data – a yearly file with one record per beneficiary-month flagging for enrollment, and a period format with one record per beneficiary-period. Below are tables describing the required elements of each format.

Annual Shape: Enrollment input files for each year with one record per beneficiary-month. Named as such - libref.[prefix][yyyy]

Required Variables	Standardized Variable Names	Format
Unique patient identifier*	No standard, but should be same as input claims data sets	No standard format, but should be same as input claims data sets
First day of month* (one for each month)	Date*	Date format*
Enrollment variable*	No standard	Binary, 1=enrolled, 0=not enrolled*

Table 2. Enrollment Input Requirements - Annual Shape

Period Shape: Enrollment input files with one record per beneficiary and period of enrollment. It can have multiple records per beneficiary (i.e. if there is a gap in enrollment, the file can have a record for the first period and a record for the second period after the gap)		
Required Variables	**Standardized Variable Names**	**Format**
Unique patient identifier*	No standard, but should be same as input claims data sets	No standard format, but should be same as input claims data sets
Start of enrollment period*	Begdt*	Date format*
End of enrollment period*	Enddt*	Date format*

Table 3. Enrollment Input Requirements - Period Shape

Custom Input – CSV Files

As mentioned, three csv tables are provided for customizing the algorithm. One table is dedicated to listing all the valid ICD-9 and ICD-10 diagnosis codes for each condition, and whether or not the code needs to be the primary or secondary diagnosis on the claim. Another table is dedicated to listing all the codes that mark exclusion of a claim if found. Either of these tables can be customized by the addition or removal of any codes. The third table holds all the algorithm rules for each condition, including reference period, number of claims to find, and where to find them. In addition, in order for two claims to contribute to an algorithm, the claims must occur at a default minimum of one day apart. A rule for maximum days apart can also be specified, in which two claims must occur within the specified maximum days to qualify. For example, a default maximum of 7 days would require that 2 claims occur within a week of each other. Any of these algorithm rules can be adjusted depending on the research question. Below is an example from the default table showing the algorithm rules for diabetes:

Condition	Condition Long	Claim Types 1	Number of Claims 1	Claim Types 2	Number of Claims 2	Minimum Days Apart	Maximum Days Apart	Reference Period (Months)
DIABETES	Diabetes	IP,SNF,HHA	1	OP,CAR	2	1		24

Table 4. Diabetes Example from CSV for Default Algorithms –

To identify a diabetes condition in a patient using the CCW definition it is necessary to observe either one claim of type IP (inpatient), SNF (skill nursing facility), or HHA (home health agency), or to observe two claims at least 1 day apart of type OP (outpatient) or CAR (carrier), in the two years previous to the measuring point.

OUTPUT FILES

The %idcond() macro creates a set of annual files with one entry per individual, monthly flags for each condition, and the first date when a condition is met. The flags created by the macro follow the same classification used by the CCW flags. Each individual in the sample is assigned a flag for each condition that indicates both the presence or absence of diagnosis or procedure codes as well as whether the beneficiary was enrolled for the entire reference period. Their possible values are:

- 0 = Condition criteria not met, enrollment gaps during the reference period
- 1 = Condition criteria met, enrollment gaps during the reference period
- 2 = Condition criteria not met, enrolled during the entire reference period
- 3 = Diagnosis/procedure criteria met, enrolled during the entire reference period

The enrollment status through the reference period distinguishes between cases where the claims record is complete (beneficiary enrolled throughout) or incomplete (beneficiary not enrolled throughout). The analyst must decide which values are appropriate for the purpose at hand. For example, if estimating prevalence, one would only include those with complete claims coverage during the reference period, otherwise prevalence will be underestimated.

VALIDATION OF THE PACKAGE

The table below shows the resulting match rate of the algorithm for a 5% sample of 2016 Medicare data compared to the corresponding official CMS CCW data set. The end-year variable matches for around 96% of beneficiaries across all conditions. One difference between these algorithms and the CMS algorithms is that the CMS end-year flag reflects the value for the beneficiary either at the end of the year or the last month they are alive, if they died in that year. That is, in the CCW algorithm, a beneficiary who died in August with qualifying codes for hypothyroidism and qualifying enrollment will be flagged as having hypothyroidism in the end-year flag. In our algorithm, these beneficiaries would not be flagged as enrolled in the months after their death, but may have a qualifying hypothyroidism claim if it falls within the reference period. When we carry the value from the month of death to the end-year flag, as CMS does, the match rate improves to nearly 100%. The last column of the table below shows those results.

Condition	End-Year Flag Match	End-Year Flag Match - CMS Death Assumption
Acute Hypothyroidism	96.3%	99.6%
Alzheimer's Disease	95.9%	99.0%
Alzheimer's Disease and Related Dementias	95.7%	98.8%
Atrial Fibrillation	96.4%	99.6%
Acute Myocardial Infarction	96.4%	99.8%
Asthma	89.0%	91.5%
Breast Cancer	96.4%	99.8%
Cataract	96.3%	99.6%
Chronic Kidney Disease	96.3%	99.5%
Chronic Obstructive Pulmonary Disease	96.3%	99.6%
Cancer Colorectal	96.4%	99.8%
Depression	96.3%	99.5%
Diabetes	96.4%	99.6%

Endometrial Cancer	96.4%	99.9%
Glaucoma	94.3%	97.6%
Heart Failure	96.4%	99.6%
Hip/Pelvic Fracture	96.2%	99.6%
Hyperlipidemia	96.2%	99.2%
Hypertension	96.2%	99.2%
Ischemic Heart Disease	96.4%	99.6%
Lung Cancer	96.4%	99.8%
Osteoporosis	96.3%	99.7%
Prostate Cancer	96.4%	99.8%
Rheumatoid Arthritis/Osteoarthritis	96.4%	99.5%
Stroke/Transient Ischemic Attack	96.3%	99.6%
Anemia	96.3%	99.4%
Hyperplasia	96.3%	99.7%

Table 5. Validation Results

CASE STUDIES

CASE STUDY #1: POPULATION COMPARISONS

This package can be used to compare different populations using a standardized approach, and to test the algorithm used to identify conditions. Here we show an application comparing Medicare Advantage (MA) and Medicare FFS populations, and comparing the MA population over time.

The macro presented here was tested on a large database of MA claims from 2016 to 2018 using the default options of the macro (i.e. using CCW conditions definition). Before running the macro we adjusted the enrollment and claims files so they have the format expected by the macro, as described above. We used the facility/non-facility indicator and the type of service variable to classify the claims into the same categories used in the CCW algorithm (Inpatient, Outpatient, SNF, HHA, and Carrier).

Patient's identification of conditions is generated by calling the macro function %idcond. The macro can be called from the provided file input_program.sas, or from any other program where the macro is included (use %include "idcond.sas"). We used the provided input program, and edited it to indicate all needed parameters as shown in the sample code below:

```
libname clmin "&projhm./prepareclms";

%let clmsfls=clmin.clms2012q1 clmin.clms2012q2 clmin.clms2012q3
clmin.clms2012q4 clmin.clms2013q1 clmin.clms2013q2 clmin.clms2013q3
clmin.clms2013q4 clmin.clms2014q1 clmin.clms2014q2 clmin.clms2014q3
clmin.clms2014q4 clmin.clms2015q1 clmin.clms2015q2 clmin.clms2015q3
clmin.clms2015q4 clmin.clms2016q1 clmin.clms2016q2 clmin.clms2016q3
clmin.clms2016q4 clmin.clms2017q1 clmin.clms2017q2 clmin.clms2017q3
clmin.clms2017q4 clmin.clms2018q1 clmin.clms2018q2 clmin.clms2018q3
clmin.clms2018q4 clmin.clms2019q1 clmin.clms2019q2 clmin.clms2019q3
clmin.clms2019q4;

****** Wrapper macro;
%include "idcond.sas";

****** Macro Function;
%idcond(projhome=&projhm,
        id=patid,
        minyear=2012,
```

```
maxyear=2019,
claims_data=&clmsfls,
create_enr=Y,
create_enr_shape=P,
create_enr_filein=clmin.ma_dates,
enr_prefix=enr.contenrol_,
enr_var=enr,
claims_out_prefix=output.ccw_,
custom_algorithm=N,
custom_suffix=,
custom_cond=);
```

The claims were prepared in quarterly files, so all files needed were listed in a macro variable called “clmsfls”. Patients enrollment dates are provided in the file “clmin.ma_dates” with a period shape (table with unique beneficiary id, date when coverage starts and date when it ends). Other user provided parameters are “projhm” containing the path to where the package is installed, “patid” referring to the name of the variable with a patient unique identifier, and the minimum and maximum years to process (2012 and 2019).

The custom_algorithm option is set to “N” so the default CCW conditions are going to be created. The option create_enr is set to “Y” requesting the macro to create yearly files counting the number of months a beneficiary has been continuously enrolled before and after any point in time on a monthly basis. This file is needed to generate the chronic conditions flags, and the macro derives it from the enrollment information.

The final set of parameters are the ones defining the output files. All output files are generated by year, with the year indicated in the file name. The user needs to define the name prefix and include a library name if the files are to be saved permanently. In this example the prefix “enr.contenrl_” is used for the set of enrollment files, and the prefix “output.ccw_” is used for the set of condition flags. In both cases the files are saved permanently, but in different libraries, “enr” and “output”. Note that the structure of folders with all the input files has to be kept as it comes in the package, but the output file can be placed anywhere.

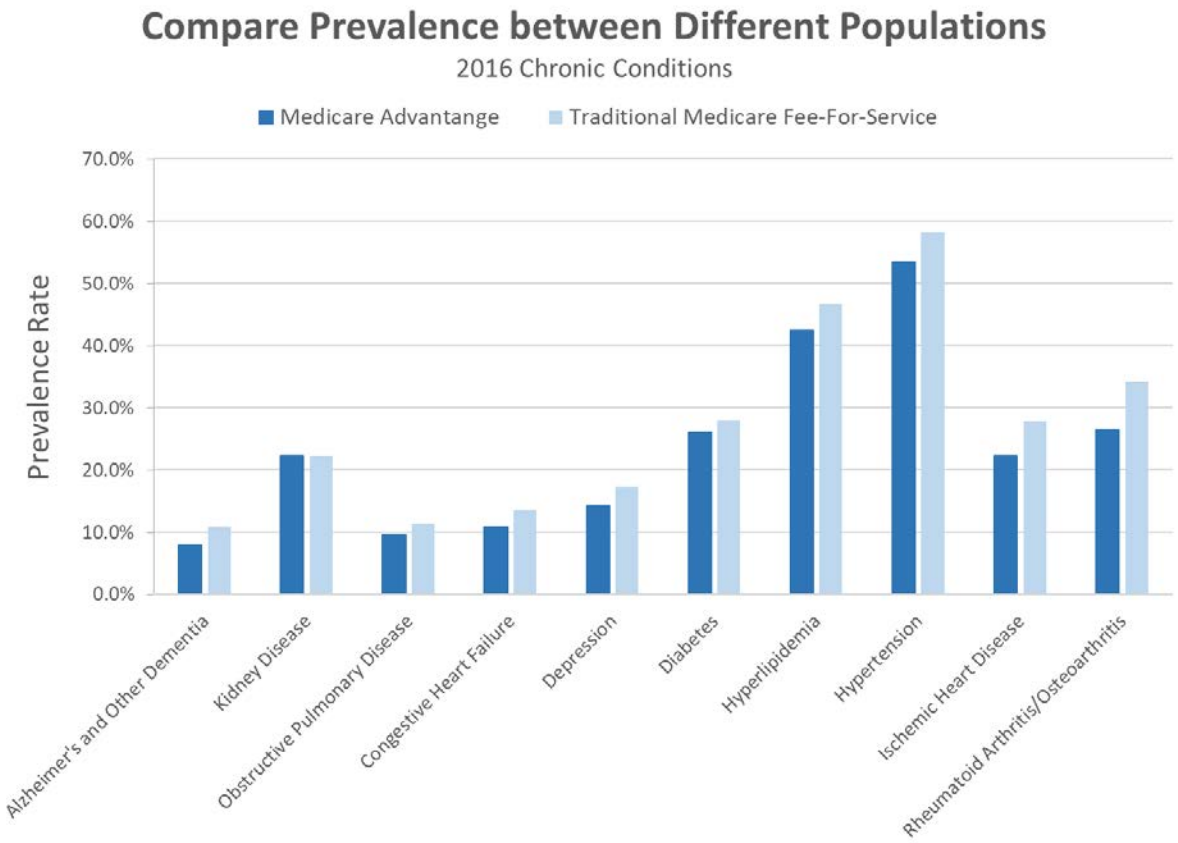

Figure 2. Case Study - Comparison of Prevalence Rates between Different Populations

After the macro creates the chronic conditions flags, we can derive prevalence rates by computing the percent of beneficiaries tagged with a flag value of 3 among all beneficiaries with enrollment during the entire reference period (flag values 2 and 3). We can use the validation run of the macro on Medicare FFS 2016 to compare prevalence rates between these two populations. Figure 4 shows a comparison of FFS vs MA prevalence rates for a selection of the default CCW standard conditions. The comparison shows that the MA Medicare population has lower rates across the board than the FFS Medicare population.

Using the same run on the MA population we can compare prevalence rates over time. One particularly interesting thing to look at is the change in October 2015 in the coding system used to record patients' diagnoses, from the ICD-9 to ICD-10 system. The change was significant and introduced more detail in the diagnosis codes. The CCW definitions were adapted to incorporate the new ICD-10 coding system, but this was a process that in some cases took multiple adjustments over time. The %idcond macro can be run using the custom options to compare the effect of different definitions on prevalence trend lines.

Figure 3 shows a striking effect of the change in the coding systems and a poorly defined crosswalk between ICD-9 and ICD-10 used to define chronic asthma. The figure shows prevalence rates for chronic asthma between 2012 and 2018 using the CCW definitions with the 2017 revision. The rate increase from a little below 4% to over 11% in asthma prevalence is clearly an effect of the coding system change and the subsequent change in the algorithm to identify asthma and not a change in the population over time. The Chronic Conditions Warehouse revised the definition again in 2021 to exclude ICD-10 J44.0, J44.1, and J44.9 from the 2017 list to adjust for this problem.

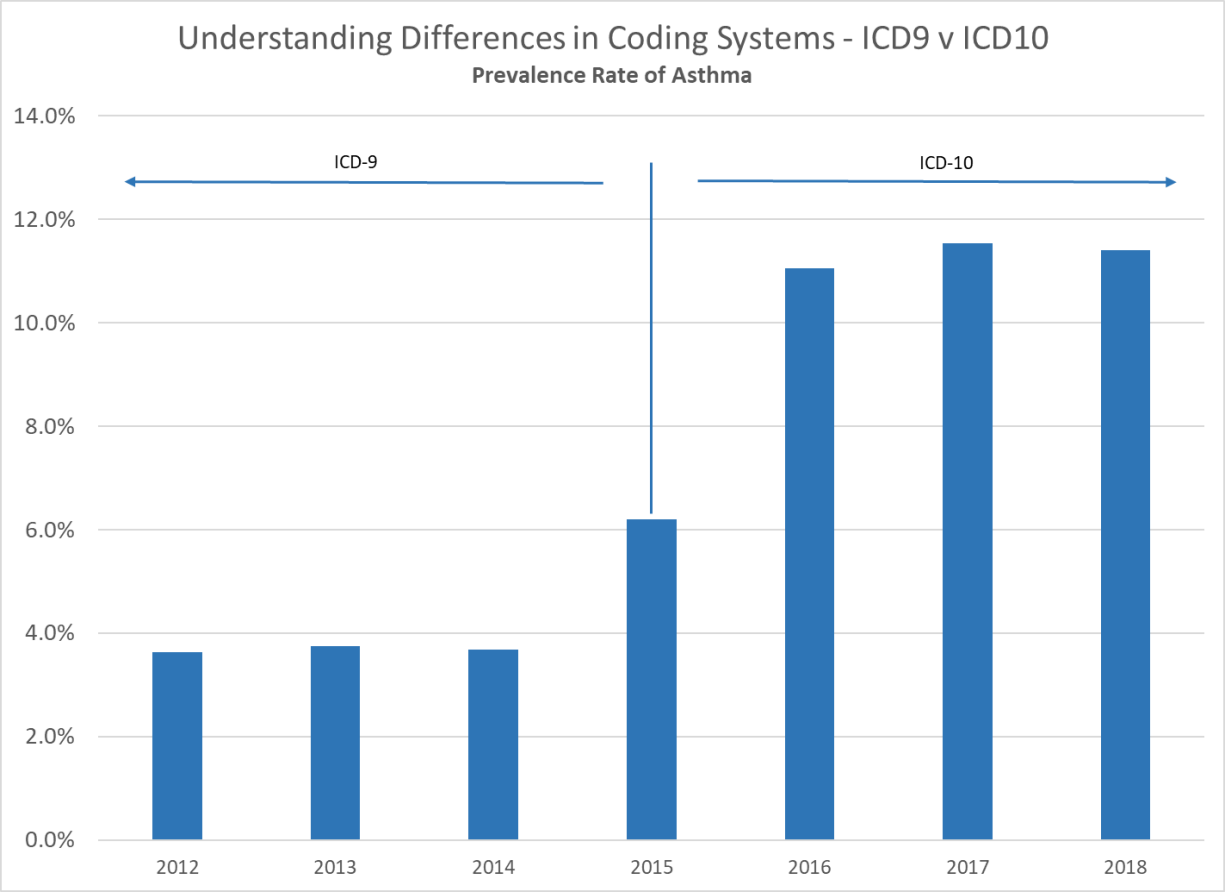

Figure 3. Case Study - Comparison Across Coding Systems

CASE STUDY #2: VARIATIONS ON THE CCW DIABETES ALGORITHM

Based on a comparison of diabetes prevalence as estimated from self-reports in surveys and the application of the CCW algorithm to claims, it appeared possible that the latter may over-identify prevalence of diabetes (St.Clair 2017). As described earlier, the CCW diabetes algorithm requires observation of related diagnosis codes on one institutional claim (inpatient, SNF, or home health), or two outpatient or carrier claims, at least one day apart, with a two-year reference period. One possible contribution to an over-estimate of diabetes prevalence using claims could be diagnoses associated with rule-out tests, perhaps done at annual physical exams. To explore this possibility, modifications were made to the CCW algorithm, which requires at least a day between outpatient or carrier claims diagnosis. The modifications add the requirement of a maximum number of days of less than a year (90, 180, and 365 days), e.g., two physician diabetes claims must be between 1 to 90 days apart. In addition, longer reference periods were applied (3, 4, and 5 years as well as the 2-year standard reference) to see if the prevalence stabilized as the look-back period grew longer.

Figure 4 shows estimates of diabetes prevalence for the Medicare Advantage enrollees 65 and older applying four algorithms to a private claims database. The results show that imposing a maximum of 90, 180, or 365 days between diabetes diagnoses observed in claims does reduce the estimated prevalence when compared to the CCW algorithm. Note that years 2015 and 2016 are excluded as they are transition years for ICD diagnosis coding (ICD9 to ICD10). The differences between the standard CCW algorithm results and the modifications appear to be slightly larger in 2014 when ICD9 codes are used, but relatively similar in years 2017 to 2019.

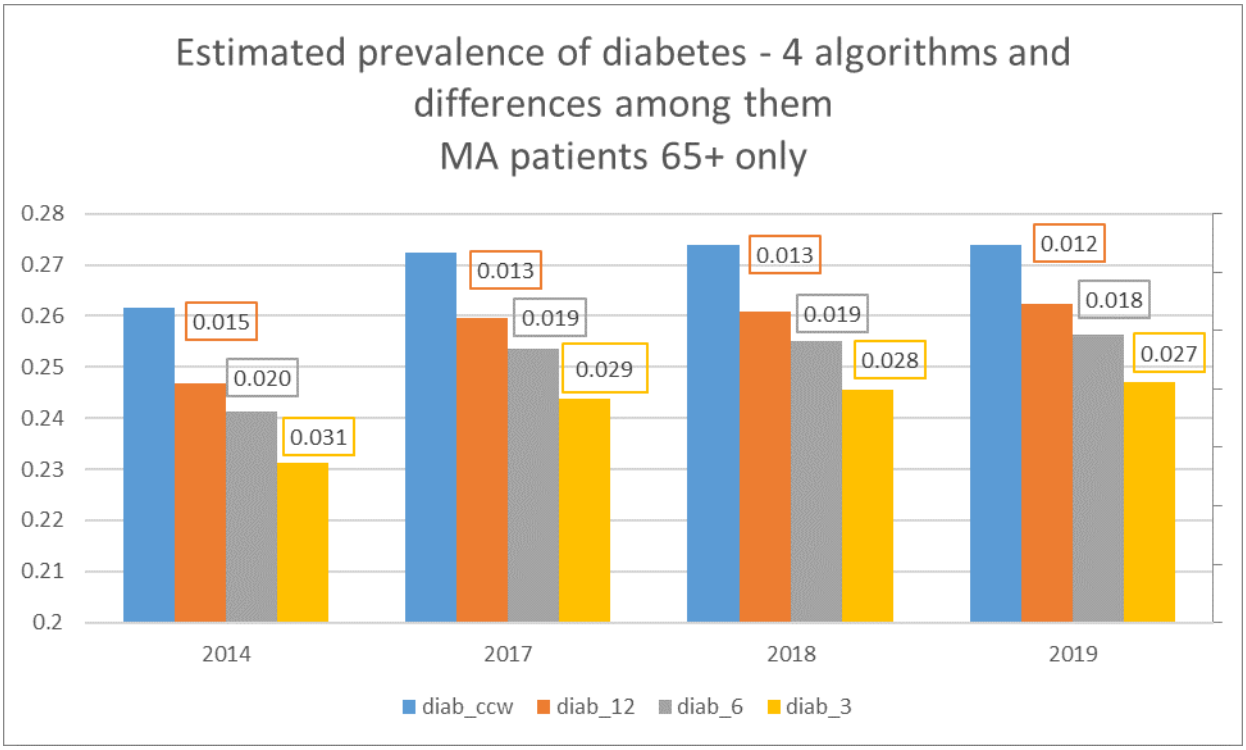

Figure 4. Sample includes Medicare Advantage patients 65 and older. Bars show estimated prevalence from four algorithms: diab_ccw is as defined by CCW; diab_12 imposes a maximum of 365 days between diabetes claims; diab_6 imposes a maximum of 180 days between diabetes claims; diab_3 imposes a maximum 90 days between diabetes claims. Text boxes show the difference between the diab_ccw and each of the alternative methods.

Figure 5 shows the difference in 2019 prevalence estimates among the four algorithms as the reference period is increased up to five years. The sample includes only patients enrolled continuously for five years, regardless of the reference period. The trendlines show the effects of lengthening the reference period; the coefficient of *x* estimates the slope for prevalence increase as the reference period increases. The effects are largest for the standard CCW definition, followed by that for the 90-day maximum between claims. The 180-day and 365-day maximum algorithms increase at about the same rate, lower than for the CCW method and slightly lower than that using the 90-day maximum. This suggests that rates from imposing a maximum time between diagnosis dates are less affected by increasing the look-back time, implying they may be more stable.

Note that this is an exploratory exercise, not a validation of these modifications. Clearly much more work would need to be done to validate when and if this type of alternative to the CCW algorithm would be useful, but there is some indication that it is worth investigating further. Application of the package made it possible to explore these variations simply and quickly.

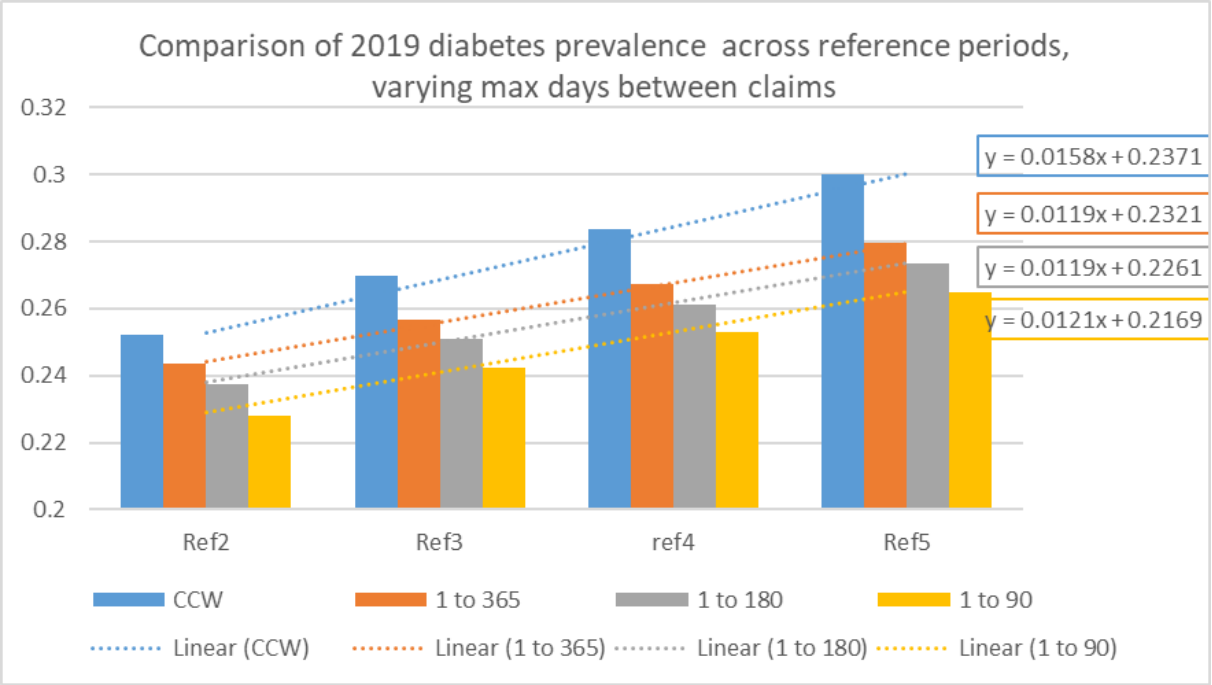

Figure 5. Sample includes Medicare Advantage patients 70 or older, all continuously enrolled for 5 years. Ref2, Ref3, Ref4, and Ref5 represent algorithms applying a 2, 3, 4, and 5 year reference period, respectively. The CCW series requires at least 1 day between outpatient and carrier diabetes claims; the other series require 1 to a maximum of 90, 180, or 365 days between claims. Trendlines have R-sq values of over .9. X coefficient is the slope of the trendline.

CONCLUSION

The package presented here provides a means to conveniently apply CCW or CCW-like algorithms to identify health conditions to any claims data or other type of data containing comprehensive records of patient diagnosis. It builds on the CCW definition structure, which uses a combination of reference period, diagnosis and procedure code sets, service types, and enrollment to specify the algorithm identifying a condition. The package facilitates application of these algorithms, producing monthly condition flags indicating whether the condition criteria were met combined with enrollment status during the reference period. It provides flexible but structured condition algorithm definitions, and SAS code to apply them consistently.

The package has multiple potential uses, including application of CCW condition algorithms to claims data from non-CMS sources such as private insurers, and updating or modifying CCW algorithms such as adding or changing the list of diagnosis codes or the reference period associated with a specific condition. New conditions of interest defined using CCW-like rules and codes can be implemented easily by providing the parameters through a csv spreadsheet, with the caveat that the user exercise caution and ensure a validated definition is specified before integrating results into analysis.

In summary we encourage the use of validated algorithms for detecting health conditions in claims data and present this package with current CCW definitions to facilitate this practice. Using standardized methods to implement algorithms should produce more replicable results across analyses. If widely adapted, users of the package who develop additional validated health condition definitions can share them with others. The authors encourage use of this package and welcome feedback for its improvement.

The package can be downloaded from GitHub (V0 used here).

REFERENCES

CCW Chronic Condition Algorithms. https://www2.ccwdata.org/documents/10280/19139421/ccw-chronic-condition-algorithms.pdf

CCW Chronic Condition Algorithms Change History. https://www2.ccwdata.org/documents/10280/19139421/chronic-condition-change-history.pdf

CCW Chronic Condition Algorithms Reference List. https://www2.ccwdata.org/documents/10280/19139421/ccw-chronic-condition-algorithms-reference-list.pdf

CCW Technical Guidance: Calculating Medicare Population Statistics. January 2018, Version 1.2. https://www2.ccwdata.org/documents/10280/19002248/ccw-technical-guidance-calculating-medicare-population-statistics.pdf

Gorina, Y. and Kramarow, E.A. (2011), Identifying Chronic Conditions in Medicare Claims Data: Evaluating the Chronic Condition Data Warehouse Algorithm. Health Services Research, 46: 1610-1627. https://doi.org/10.1111/j.1475-6773.2011.01277.x

Optum's de-identifed Clinformatics® Data Mart Database (2007-2019)

St. Clair, Patricia ScB; Gaudette, Étienne PhD; Zhao, Henu PhD; Tysinger, Bryan MSc; Seyedin, Roxanna MPh; Goldman, Dana P. PhD Using Self-reports or Claims to Assess Disease Prevalence, Medical Care: August 2017 - Volume 55 - Issue 8 - p 782-788. https://doi.org/10.1097/MLR.0000000000000753

ACKNOWLEDGMENTS

The authors would like to thank Katrina Kaiser, Khristina Lung and Henu Zhao for helping in the testing and debugging of this algorithm and the Data Core@USC Schaeffer Center for enlightening discussions on the topic. We also thank Deborah Testa for coding an early version of some of the package's programs.

CONTACT INFORMATION

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

Patricia Ferido
pferido@healthpolicy.usc.edu