ICD_CLASS SAS® Software User's Guide

Version FY 2016

Prepared for:

U.S. Centers for Disease Control and Prevention

Prepared by:

Center for Healthcare Policy and Research,
University of California, Davis
2103 Stockton Blvd.,
Sacramento, CA 95817

National Association of Health Data Organizations
124 South 400 East, Suite 220
Salt Lake City, Utah 84111

Acknowledgment

The development of this software has been supported by Cooperative Agreement 1U38OT000216 from the Centers for Disease Control and Prevention to the Task Force for Global Health, Public Health Informatics Institute, through a subcontract with the National Association of Health Data Organizations (NAHDO).

This macro is distributed with helpful intent, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

Overview

Transactions are electronic exchanges involving the transfer of information between two parties for specific purposes. For example, a health care provider will send a claim to a health plan to request payment for medical services. The Health Insurance Portability & Accountability Act of 1996 (HIPAA) named certain types of organizations as covered entities, including health plans, health care clearinghouses, and certain health care providers. In the HIPAA regulations, the Secretary of Health and Human Services (HHS) named certain standard transactions for Electronic Data Interchange of health care data: claims and encounter information, payment and remittance advice, eligibility, enrollment and disenrollment, referrals and authorizations, coordination of benefits and premium payment. Under HIPAA, if a covered entity conducts one of the adopted transactions electronically, they must use the adopted standard and adhere to the associated content and format requirements. Under HIPAA, HHS also adopted specific code sets for diagnoses and procedures to be used in all transactions. Effective October 1, 2015, all covered entities are required to implement the International Classification of Diseases, 10th Revision, Clinical Modification (ICD-10-CM) for diagnosis coding and the International Classification of Diseases, 10th Revision, Procedural Coding System (ICD-10-PCS) for inpatient hospital procedure coding.[1] These new codes replace the current International Classification, 9th Revision, Clinical Modification, Volumes 1 and 2 and the International Classification, 9th Revision, Clinical Modification, Volume 3 for diagnosis and procedure codes respectively.

Although the HIPAA code set requirement affects all covered entities that perform electronic transactions in the United States, it does not necessarily affect all health care data sets. For example, the following entities are not "covered entities" and are therefore not subject to the HIPAA code set requirement:

1. Employers and employer-based health care facilities, such as occupational health clinics supported entirely by an employer;
2. Workers' compensation plans;
3. Correctional health services;
4. Liability insurance carriers, including those offering home/property, disability, automobile, and business coverage.

Similarly, the HIPAA code set requirement does not apply to paper transactions or transactions that only involve health care providers and patients (e.g., concierge practice).[2]

Recognizing the burden of requiring providers to provide data to different recipients using two different code sets, health data agencies and public health agencies implemented ICD-10-CM and ICD-10-PCS on October 1, 2015. However, there was no statutory or regulatory requirement for them to do so under HIPAA, except to the extent that these agencies functioned separately as covered entities (e.g., by administering covered programs such as state Medicaid programs). In fact, the Centers for Medicare & Medicaid Services (CMS) signed off on a "crosswalk" technique to translate ICD-10-CM/PCS codes into ICD-9-CM codes and keep using the older codes as a workaround for Medicaid fee-for-service programs in California, Louisiana,

[1] http://www.cms.gov/Regulations-and-Guidance/HIPAA-Administrative-Simplification/TransactionCodeSetsStands/TransactionsandCodeSetsRegulations.html ; accessed April 2, 2015.

[2] http://library.ahima.org/xpedio/groups/public/documents/ahima/bok1_048535.hcsp?dDocName=bok1_048535; accessed April 2, 2015.

Maryland and Montana. The claims processing systems in those four states are unable to perform payment calculations using ICD-10-CM/PCS codes, even though they are collecting ICD-9-CM codes from covered entities.[3]

Finally, for some health care services that spanned the October 1, 2015 transition date, providers were required to split each claim into two: one claim representing services provided prior to October 1, 2015 using ICD-9 –CM codes and one claim for services on or after October 1, 2015 using ICD-10-CM/PCS codes. Claims that are not compliant with these policies are being returned to the provider for correction before payment.[4]

As a result of these policies, some surveillance data systems may capture some ICD-9-CM diagnosis or procedure codes from claims or other data submitted to payers, private registries, or state public health agencies or health data agencies after October 1, 2015. If the same data set includes both ICD-9-CM coded diagnoses and ICD-10-CM coded diagnoses, then problems could arise in the course of subsequent analyses. CDC requested a tool that could be used to ascertain whether a specific claim or other record had been coded in ICD-9-CM or ICD-10-CM. This tool was subsequently adapted to differentiate ICD-9-CM and ICD-10-PCS procedure codes, and to flag any invalid or incomplete diagnosis or procedure codes on a record.

The ICD_CLASS software is a set of two SAS macros, %ICD_CLASS and %ICD_PCS_CLASS, that separate ICD-9-CM and ICD-10-CM coded diagnoses, or ICD-9-CM and ICD-10-PCS coded procedures, when they are included in a single SAS data set. The purpose is to assist the healthcare and health research communities in assessing the quality of ICD-coded data, during and after the code set transition, and properly preparing the data for subsequent analyses. The software incorporates ICD-9-CM and ICD-10-CM/PCS files that can be downloaded from the Center for Medicare & Medicaid Services website:

1) http://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html
2) http://www.cms.gov/Medicare/Coding/ICD10/2016-ICD-10-CM-and-GEMs.html
3) https://www.cms.gov/Medicare/Coding/ICD10/2016-ICD-10-PCS-and-GEMs.html

[3] http://www.modernhealthcare.com/article/20150904/NEWS/150909938 ; accessed December 20, 2015.
[4] http://www.cms.gov/Outreach-and-Education/Medicare-Learning-Network-MLN/MLNMattersArticles/Downloads/SE1408.pdf ; accessed April 2, 2015.

Table of Contents

1.0 Before You Begin

1.1 Prerequisites

1. Please read this file carefully and try the examples provided in the end of this document before running your own analyses.
2. This macro requires SAS ®Version 9.1 or later to run since DATA step hash objects.

1.2 Precautions:

Do not use your original data for the analyses.

1.3 Contact for reporting bugs/comments:

Please e-mail with a clear description of the problem to eapoltav@ucdavis.edu.

1.4 Assumptions:

When a diagnosis code can be identified in both ICD-9-CM and ICD-10-CM, the tool follows *ICD-10-CM Official Guidelines for Coding and Reporting, FY 2011*, and *ICD-10-CM Official Guidelines for Coding and Reporting, FY 2016*.[5] For example, when the principal diagnosis in a record is an E or V code, the following rules are applied:

- In ICD-9-CM coded records, the Supplementary Classification of Factors Influencing Health Status and Contact with Health Services codes (V01.0-V91.99) may be used as either a first (principal diagnosis in patient setting) or secondary code.

- In ICD-9-CM coded records, an E code can never be a principal (first listed) diagnosis.

- In ICD-10-CM coded records, the external causes of morbidity codes (V00-Y99) should never be sequenced as the first-listed or principal diagnosis.

In a case when a record contains ICD-9-CM and ICD-10-PCS procedure codes, the tool marks a record as not classified.

2.0 %ICD_CLASS: Getting Started

2.1 Setup

1. Create a directory for the files downloaded/extracted in the following steps 2 and 3.
2. Download the package zip file to the created directory.
3. Extract the content of the zip file to the created directory.
4. To compile the SAS macro, copy the following lines in your SAS program: *%include"<path for the created directory>\ICD_CLASS.sas";*

[5] http://www.cdc.gov/nchs/data/icd/icd9cm_guidelines_2011.pdf and http://www.cdc.gov/nchs/data/icd/10cmguidelines_2016_Final.pdf ; accessed December 22, 2015.

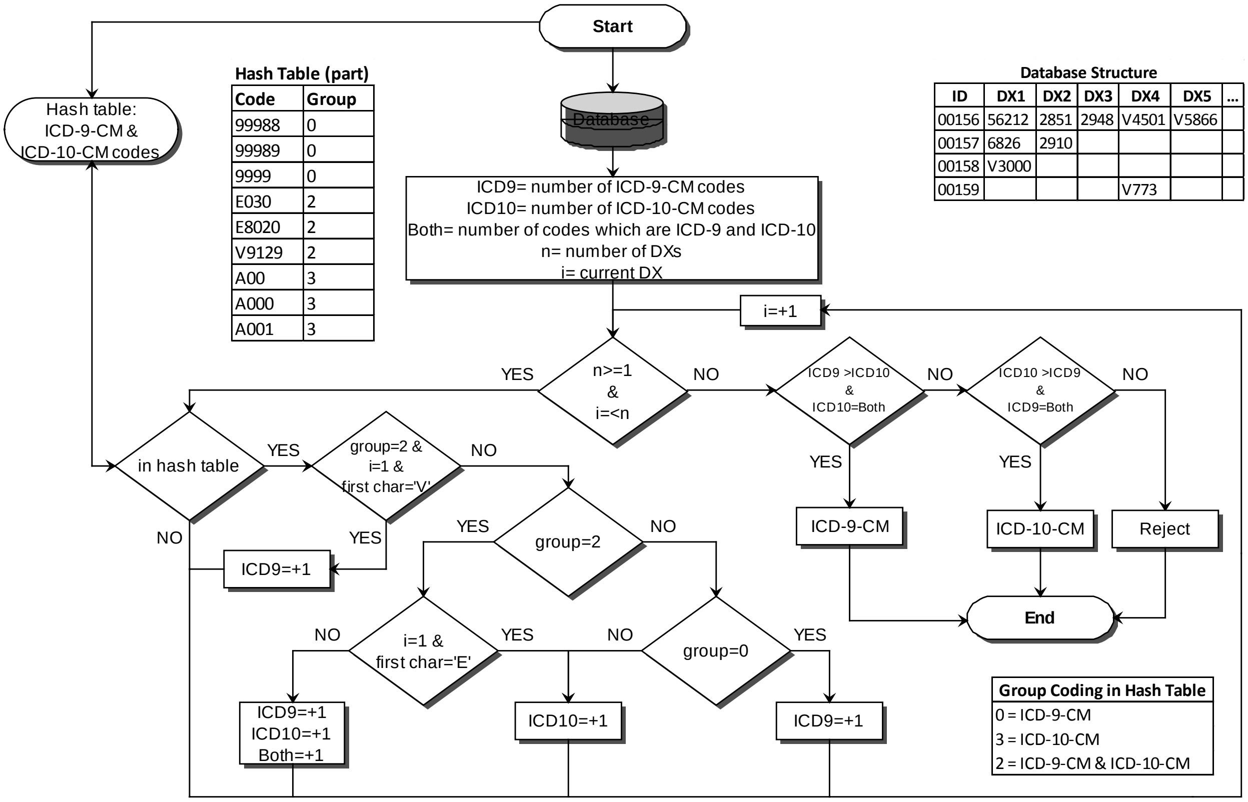
5. Replace <*path for the created directory*> with the actual pathname. Backslash “\” should be used inside the path.

2.2 Description

1. The macro, %ICD_CLASS, classifies each discharge record as ICD-9-CM or ICD-10-CM or Unclassified (Rejected).
2. The macro adds four new columns at the end of the data set:
   variable *error_n,* number of invalid codes;
   variable *code_n*, number of valid codes;
   variable *ICD_9*;
   variable *ICD_10*.
3. Variables *ICD_9* and *ICD_10* can have only two values: 0 and 1. If *ICD_9* = 1, then the record was classified as ICD-9-CM; if *ICD_10* = 1, then the record was classified as ICD-10-CM; if *ICD_9* = 0 and *ICD_10* = 0, then the record was not classified. In other words, a record must have at least one valid ICD-9-CM or ICD-10-CM diagnosis code to be classified.
4. The macro requires one permanent SAS data set containing the ICD-9-CM and ICD-10-CM codes. Users can use the provided codes.sas7bdat data set – they must copy it to Work directory, or users can create their own codes.sas7bdat data set using the instructions provided in **Appendix A.**
5. The macro requires one permanent SAS data set with coded data from a health care encounter. In this data set, provided by a user, each row represents a single encounter record, and columns have ICD-9-CM or ICD-10-CM coded diagnosis. There is no limit on the number of diagnoses in the data set.

2.3 Algorithm

The algorithm sequentially proceeds row by row, comparing each code to the codes (variable Code) in the hash table. The hash table is the codes. sas7bdat data set loaded into the computer memory.



Hash Table (part)

Code	Group
99988	0
99989	0
9999	0
E030	2
E8020	2
V9129	2
A00	3
A000	3
A001	3

Database Structure

ID	DX1	DX2	DX3	DX4	DX5	...
00156	56212	2851	2948	V4501	V5866	
00157	6826	2910				
00158	V3000					
00159				V773		

Group Coding in Hash Table
0 = ICD-9-CM
3 = ICD-10-CM
2 = ICD-9-CM & ICD-10-CM

2.4 Classification rules

The macro counts the number of valid ICD-9-CM, valid ICD-10-CM, and invalid codes that could not be found in either ICD-9-CM or ICD-10-CM code sets. Then, it compares those counts to classify a record. A record will be flagged as Unclassified (Rejected):

1. If it has no valid codes or
2. If the number of valid ICD-9-CM codes equals the number of valid ICD-10-CM codes or
3. If the number of codes found in the both the ICD-9-CM and ICD-10-CM code sets is different from the number of codes classified as either ICD-9-CM or ICD-9-CM (i.e., if the record cannot be resolved as either an ICD-9-CM coded record or an ICD-10-CM coded record).

2.5 Macro parameters

All parameters should be specified without any quotation marks. The values of the parameters are not case sensitive. They are divided in two categories:

2.5.1 Required parameters:

The following parameters must be provided; otherwise the macro will not run properly.

dsn: to specify the name of the input SAS data set. It must be a permanent SAS data set stored in Work or any other defined directory.

ordervars: to specify the name(s) of character variable(s) containing diagnosis codes. It can be only one variable (for example Dx1), or a list of variables with no comma (for example Dx1 Dx2 Dx7), or a range of variables with double minus, --, as a separator (for example Dx1--Dx22). **Note**: the first variable in a list or in a range is assumed to be the principal diagnosis variable.

2.5.2 Optional parameters:

The following parameter can be left unspecified, and the default analysis will be performed.

distinct: to specify if additional analysis of the number of valid distinct codes used should be done, a table with unique invalid codes, inv_codes.sas7bdat in Work library, should be created, and all invalid codes should be replaced with 'inv' value. The values of this parameter must be 0 or 1. By default, i.e. if this parameter is not specified, no additional analysis is performed. **WARNING**: if you are working with a large dataset, choosing *distinct*=1 may cause an error.

2.6 Data set requirements

1. An analyzed data set **must** be located in Work library; all variables with diagnosis codes **must** have $char format; decimal points **must** be removed from diagnosis codes.
2. codes.sas7bdat data set **must** be located in Work library.

2.6.1 Example

An example of discharge data, mix.sas7bdat, is included in the package zip file. Users are advised to run this example as an exercise by following the steps below (more details in **Appendix C**):

1. Follow the steps in the Setup section above,
2. Copy the following SAS codes:
   %ICD_CLASS(dsn=<*path for the created directory*>.mix,
   Dx1 Dx2 Dx3 Dx4 Dx5 Dx6 Dx7 Dx8,
   distinct=1);
3. Replace <*path for the created directory*> with its actual path. Backslash "\" should be used inside the path.
4. Run the SAS code. Without error message, the output should match exactly with the output provided in **Example 2** of **Appendix C**.

3.0 %ICD_PCS_CLASS: Getting Started

3.1 Setup

1. Create a directory for the files downloaded/extracted in the following steps 2 and 3.
2. Download the package zip file to the created directory.
3. Extract the content of the zip file to the created directory.
4. To compile the SAS macro, copy the following lines in your SAS program:

%include"<path for the created directory>\ICD_PCS_CLASS.sas";

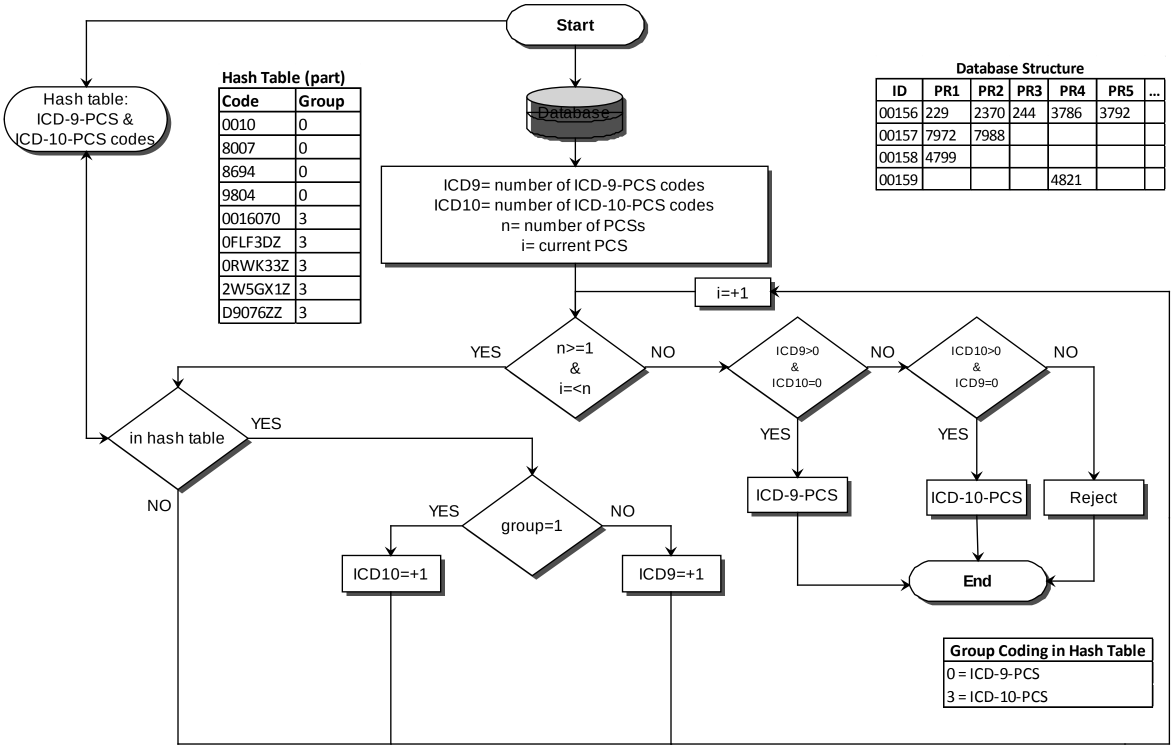
5. Replace <*path for the created directory*> with the actual pathname. Backslash "\" should be used inside the path.

3.2 Description

1. The macro, %ICD_PCS_CLASS, classify each discharge record as ICD-9-CM or ICD-10-PCS or Unclassified (Rejected).
2. The macro adds four new columns at the end of the data set:
   variable *error_n,* number of invalid codes;
   variable *code_n*, number of valid codes;
   variable *ICD_9*;
   variable *ICD_10*.
3. Variables *ICD_9* and *ICD_10* can have only two values: 0 and 1. If *ICD_9* = 1, then the record is classified as ICD-9-CM; if *ICD_10* = 1, then the record is classified as ICD-10-PCS; if *ICD_9* = 0 and *ICD_10* = 0, then the record is not classified. In other words, a record must have at least one valid ICD-9-CM or ICD-10-PCS procedure code to be classified.
4. The macro requires one permanent SAS data set containing the ICD-9-CM and ICD-10-PCS codes. Users can use the provided codes_pcs.sas7bdat data set – they must copy it to Work directory, or users can create their own codes_pcs.sas7bdat data set using the instructions provided in **Appendix B.**
5. The macro requires one permanent SAS data set with coded data from a health care encounter. In this data set, provided by a user, each row represents a single encounter record, and columns have ICD-9-CM or ICD-10-PCS coded procedures. There is no limit on the number of procedures in the data set.

3.3 Algorithm

The algorithm sequentially proceeds row by row, comparing each code to the codes (variable Code) in the hash table. The hash table is the codes_pcs. sas7bdat data set loaded into the computer memory.



Hash Table (part)

Code	Group
0010	0
8007	0
8694	0
9804	0
0016070	3
0FLF3DZ	3
0RWK33Z	3
2W5GX1Z	3
D9076ZZ	3

Database Structure

ID	PR1	PR2	PR3	PR4	PR5	...
00156	229	2370	244	3786	3792	
00157	7972	7988				
00158	4799					
00159				4821		

Group Coding in Hash Table
0 = ICD-9-PCS
3 = ICD-10-PCS

3.4 Classification rules

The macro counts the number of valid ICD-9-CM, valid ICD-10-PCS, and invalid codes that could not be found in either ICD-9-CM or ICD-10-PCS code sets. Then, it compares those counts to classify a record. A record will be flagged as Unclassified (Rejected)

1. If it has no valid procedure codes or
2. If it has a valid ICD-9-CM code(s) and a valid ICD-10-PCS code(s).

3.5 Macro parameters

All parameters should be specified without any quotation marks. The values of the parameters are not case sensitive. They are divided in two categories:

3.5.1 Required parameters:

The following parameters must be provided; otherwise the macro will not run properly.

dsn: to specify the name of the input SAS data set. It must be a permanent SAS data set stored in Work or any other defined directory.

ordervars: to specify the name(s) of character variable(s) containing procedure codes. It can be only one variable (for example Pr1), or a list of variables with no comma (for example Pr1 Pr2 Pr7), or a range of variables with double minus, --, as a separator (for example Pr1--Pr22).

3.5.2 Optional parameters:

The following parameter can be left unspecified, and the default analysis will be performed.

distinct: to specify if additional analysis of the number of valid distinct codes used should be done, a table with unique invalid codes, inv_pcs_codes.sas7bdat in Work library, should be created, and all invalid codes should be replaced with 'inv' value. The values of this parameter must be 0 or 1. By default, i.e. if this parameter is not specified, no additional analysis is performed. **WARNING**: if you are working with a large dataset, choosing *distinct*=1 may cause an error.

3.6 Data set requirements

1. An analyzed data set **must** be located in Work library; all variables with procedure codes **must** have $char format; decimal points **must** be removed from procedure codes.
2. Codes_pcs.sas7bdat data set **must** be located in Work library.

3.6.1 Example

An example of discharge data, mix.sas7bdat, is included in the package zip file. Users are advised to run this example as an exercise by following the steps below (more details in **Appendix D)**:

1. Follow the steps in the Setup section above,
2. Copy the following SAS codes:
   %ICD_PCS_CLASS(dsn=<*path for the created directory*>.mix,
   Pr1 Pr2 Pr3 Pr4 Pr5 Pr6 Pr7 Pr8,
   distinct=1);
3. Replace <*path for the created directory*> with its actual path. Backslash "\" should be used inside the path.
4. Run the SAS code. Without error message, the output should match exactly with the output provided in **Example 2** of **Appendix D**.

4.0 Appendix A

4.1. ICD-9-CM codes

To create a combined 2012-2014 version of ICD-9-CM codes, we used the following source files: Version 30, Version 31, and Version 32 of ICD-9-CM Diagnosis Codes (http://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html). We downloaded three zip files and extracted their content to a newly created folder called ICD9. Each zip file contains multiple files, but we only used one text file from each extracted zip file: CMS30_DESC_SHORT_DX.txt, CMS31_DESC_SHORT_DX.txt, and CMS32_DESC_SHORT_DX.txt. All these files have only two columns: ICD-9-CM code and its short description; we need the first column from each txt file.

```
data txt;
infile datalines;
length txt $31; /*user should change it based on the length of the path in
the datalines statement*/
   input txt $;
   infile dummy filevar=txt end=done;
   do while(not done);
         input @1 d9 $char5.;
         output;
   end;
datalines;
C:\ICD9\CMS30_DESC_SHORT_DX.txt
C:\ICD9\CMS31_DESC_SHORT_DX.txt
C:\ICD9\CMS32_DESC_SHORT_DX.txt
;
proc sql noprint;
   create table icd_9_list as select distinct(D9) from txt;
quit;
```

4.2 ICD-10-CM codes

To create a list of ICD-10-CM codes, we used the 2016 ICD-10-CM code file (http://www.cms.gov/Medicare/Coding/ICD10/2016-ICD-10-CM-and-GEMs.html). We downloaded and extracted this zip file into newly created folder called ICD10. The zip file has multiple files, but we only used a text file called icd10cm_codes_2016. This text file has two columns, but we need the first one with CD-10-CM codes.

```
data icd_10_list;
   infile "c:\icd10\icd10cm_codes_2016.txt";
   input @001  d10   $char7.;
run;
```

4.3 How to create a data set called codes.sas7bdat

```
proc sql;
   create table CODE as
   select D9 as Code from icd_9_list
   union
   select D10 as Code from icd_10_list;
quit;
```

```
data CODES;
length D9 $7 D10 $7;
   if _n_=1 then
         do;
               declare hash nine(dataset: "icd_9_list");
               nine.definekey('D9');
               nine.definedone();
               declare hash ten(dataset: "icd_10_list");
               ten.definekey('D10');
               ten.definedone();
         end;
drop rc_9 rc_10 D9 D10;
set CODE;
   attrib gr length=$1;
               rc_9=nine.check(key: Code);
               rc_10=ten.check(key: Code);
               if rc_9 = 0 & rc_10 = 0 then gr='2';
               else if rc_9 = 0 & rc_10 ne 0 then gr='0';
               else if rc_10 = 0 & rc_9 ne 0 then gr='3';
run;
```

Without error messages, a file identical to the file provided in this package, codes.sas7bdat, should be created.

5.0 Appendix B

5.1. ICD-9-CM codes

To create a combined 2012-2014 version of ICD-9-CM codes, we used the following source files: Version 30, Version 31, and Version 32 of ICD-9-CM Procedure Codes (http://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes.html). We downloaded three zip files and extracted their content to a newly created folder called ICD9. Each zip file contains multiple files, but we only used one text file from each extracted zip file: CMS30_DESC_SHORT_SG.txt, CMS31_DESC_SHORT_SG.txt, and CMS32_DESC_SHORT_SG.txt. All these files have only two columns: ICD-9-CM code and its short description; we need the first column from each txt file.

```
data txt;
infile datalines;
length txt $31; /*user should change it based on the length of the path in
the datalines statement*/
   input txt $;
   infile dummy filevar=txt end=done;
   do while(not done);
         input @1 d9 $char4.;
         output;
   end;
datalines;
C:\ICD9\CMS30_DESC_SHORT_SG.txt
C:\ICD9\CMS31_DESC_SHORT_SG.txt
C:\ICD9\CMS32_DESC_SHORT_SG.txt
;
proc sql noprint;
```

```
   create table icd_9_list as select distinct(D9) from txt;
quit;
```

5.2 ICD-10-PCS codes

To create a list of ICD-10-PCS codes, we used 2016 ICD-10-PCS code file (https://www.cms.gov/Medicare/Coding/ICD10/2016-ICD-10-PCS-and-GEMs.html). We downloaded and extracted this zip file into newly created folder called ICD10. The zip file has multiple files, but we only used a text file called icd10pcs_codes_2016. This text file has multiple columns, but we need the first one with ICD-10-PCS codes.

```
data icd_10_list;
   infile "c:\icd10\icd10pcs_codes_2016.txt";
   input @001  d10   $char7.;
run;
```

5.3 How to create a data set called codes_pcs.sas7bdat

```
proc sql;
   create table CODES_PCS as
   select D9 as Code, '0' as gr from icd_9_list
   union
   select D10 as Code, '3' as gr from icd_10_list;
quit;
```

Without error messages, a file identical to the file provided in this package, codes_pcs.sas7bdat, should be created

6.0 Appendix C

6.1 Setup

1. Create a directory for the files downloaded/extracted in the following steps 2 and 3.
2. Download the package zip file to the created directory.
3. Extract the content of the zip file to the created directory.
4. To compile the SAS macro, copy the following lines in your SAS program:
   %include”<path for the created directory>\ICD_CLASS.sas”;
5. Replace <*path for the created directory*> with the actual pathname. Backslash “\” should be used inside the path.
6. Copy the following SAS codes:

   data codes;

   set <*path for the created directory*>.codes;

   run;

   data mix;

   set <*path for the created directory*>.mix;

   run;
7. Replace <*path for the created directory*> with its actual path. Backslash “\” should be used inside the path.

Now, you are ready to run the examples.

6.2 Example 1.

```
%ICD_class(dsn=mix,
        ordervars = Dx1 Dx2 Dx3 Dx4 Dx5 Dx6 Dx7 Dx8,
        distinct=0);
```

```
                        RESULTS:

      8 diagnoses were used for classification
      17 records were read from MIX dataset:

      5 of them were ICD-9-CM.
    The mean number of codes per record was 3.60 (SD=2.51).

      6 of them were ICD-10-CM.
    The mean number of codes per record was 2.83 (SD=2.64).

      6 of them were not classified.

      Some classified records had invalid codes:
       4 records had at least one invalid code.
       1 records had at least two invalid codes.
```

6.3 Example 2.

```
%ICD_class(dsn=mix,
         ordervars = Dx1--Dx8,
         distinct=1);
```

```
                        RESULTS:

      8 diagnoses were used for classification
      17 records were read from MIX dataset:

      5 of them were ICD-9-CM (18 distinct valid codes).
    The mean number of codes per record was 3.60 (SD=2.51).

      6 of them were ICD-10-CM (16 distinct valid codes).
    The mean number of codes per record was 2.83 (SD=2.64).

      6 of them were not classified.

      Some classified records had invalid codes:
       4 records had at least one invalid code.
       1 records had at least two invalid codes.
```

7.0 Appendix D

7.1 Setup

1. Create a directory for the files downloaded/extracted in the following steps 2 and 3.
2. Download the package zip file to the created directory.
3. Extract the content of the zip file to the created directory.
4. To compile the SAS macro, copy the following lines in your SAS program:

%include"<path for the created directory>\ICD_PCS_CLASS.sas";

5. Replace <*path for the created directory*> with the actual pathname. Backslash "\" should be used inside the path.
6. Copy the following SAS codes:

data codes;
        set <*path for the created directory*>.codes;
run;
data mix;
        set <*path for the created directory*>.mix;
run;

7. Replace <*path for the created directory*> with its actual path. Backslash "\" should be used inside the path.

Now, you are ready to run the examples.

7.2 Example 1.

```
 %ICD_PCS_class(dsn=mix,
         ordervars = Pr1 Pr2 Pr3 Pr4 Pr5 Pr6 Pr7 Pr8,
         distinct=0);
```

```
                              RESULTS:

         8 procedures were used for classification
         17 records were read from MIX dataset:

         5 of them were ICD-9-PCS .
      The mean number of codes per record was 3.80(SD=1.48).

         10 of them were ICD-10-PCS .

The mean number of procedure codes per record was 4.70(SD=2.06).
         2 of them were not classified.

         Some classified records had invalid procedure codes:
          2 records had at least one invalid code.
```

7.3 Example 2.

```
%ICD_PCS_class(dsn=mix,
             ordervars = Pr1--Pr8,
             distinct=1);
```

```
                          RESULTS:

         8 procedures were used for classification
         17 records were read from MIX dataset:

         5 of them were ICD-9-PCS (13 distinct valid codes).
      The mean number of codes per record was 3.80(SD=1.48).

         10 of them were ICD-10-PCS (27 distinct valid codes).

The mean number of procedure codes per record was 4.70(SD=2.06).
         2 of them were not classified.

         Some classified records had invalid procedure codes:
          2 records had at least one invalid code.
```