Paper XXX-2017

A SAS® Format Catalog for ICD-9/ICD-10 Diagnoses and Procedures: Data Research Example and Custom Reporting Example

Robert Richard Springborn,
Office of Statewide Health Planning and Development,
Information Services Division, Healthcare Analytics Branch, Clinical Data Unit
Sacramento, California

ABSTRACT

The Office of Statewide Health Planning and Development (OSHPD) provides patient level administrative data containing ICD-9 and ICD-10 diagnosis and procedure codes to users worldwide including: researchers; health care providers; health care insurance providers; and media.

A SAS format catalog, which adds 161,360 ICD-9 and ICD-10 diagnosis and procedure code descriptions to administrative data, would greatly benefit all users and encourage new uses of administrative data by reducing the need for large ICD-9 and ICD-10 look-up manuals. This paper provides SAS program code for the SAS format catalog and both examples below.

A research example shows the benefit to epidemiological studies of adding diagnosis and procedure code descriptions to a cohort constructed from administrative data. Hospital and patient identifiers along with diagnosis and procedure codes and text descriptions for both ICD-9 and ICD-10 were sent to MS Excel to: verify diagnoses and procedures selected for each patient record; consider changes to diagnoses and procedures selected; and conduct preliminary graphical, tabular, and statistical analyses using MS Excel tools. Macros were used to construct the cohort thus enabling construction of an unlimited number of cohorts.

A custom reporting example using ODS Report Writing Interface is given where diagnosis and procedure code descriptions were incorporated into a custom data discrepancy report. This report compares clinical and administrative data to verify risk factors used in a risk-adjusted operative mortality model which provides hospital and surgeon performance ratings for coronary artery bypass graft surgery (CABG) in California. In today's sophisticated business environment preparing easy to read clearly designed reports to convey your message in a clear and concise manner can enhance your organization's credibility and reputation. Using ODS Report Writing Interface with the SAS format catalog enables all users of administrative data to meet and exceed their own custom data reporting needs.

INTRODUCTION

MOTIVATION FOR THIS PAPER

Worldwide, Office of Statewide Health Planning and Development (OSHPD) provides administrative data containing ICD-9 and ICD-10 diagnosis and procedure codes to different people, including their SAS programmers, for different purposes. Administrative data is used for many purposes including; researchers for epidemiological studies; health care providers who abstract and bill for diagnoses and procedures; health care insurance providers who assess risk of people with certain disease (diagnoses), procedures, and outcomes; and the media who are interested in outbreak of a disease (diagnosis) or prevalence of disease or hospital procedure in California.

Another use of administrative data features a custom built data discrepancy report. Each year OSHPD releases the *California Report on Coronary Artery Bypass Graft Surgery* which presents findings from analyses of data collected from California-licensed hospitals where surgeons performed adult CABG surgery. This report features risk-adjusted operative mortality used to evaluate hospital and surgeon performance. A custom data discrepancy report compares clinical and administrative data to verify operative mortality in the above report.

Worldwide researchers, health care providers, health insurance providers, administrative billing agents, and their SAS programmers, have been busy learning the new ICD-10 codes which replaced the ICD-9 codes on October 1, 2015. Centers for Disease Control and Prevention (CDC) explains "The U.S. has been using ICD-9-CM since 1979, and it is not sufficiently robust to serve the health care needs of the future. The content is no longer clinically accurate and has limited data about patients' medical conditions and hospital inpatient procedures, the number of available codes is limited, and the coding structure is too restrictive."[1]

---

1 *International Classification of Diseases, (ICD-10-CM/PCS) Transition – Background, Why Change?* (2015) Centers for Disease Control and Prevention. National Center for Health Statistics. Retrieved October 26, 2016, from http://www.cdc.gov/nchs/icd/icd10cm_pcs_background.htm

American Health Information Management Association (AHIMA) reports that "ICD-10 benefits patients through improving: Patient outcomes and patient safety with better data for analysis and research; Ability to manage chronic disease by better capturing patient populations; More accurate reflections of patients' clinical complexity and severity of illness; Ability to better identify high-risk patients; Ability to manage population health; Ability to assess effectiveness and safety of new medical technology; Increased patient engagement"[2].

With diagnosis codes increasing from 15,964 (ICD-9) to 64,945 (ICD-10); and procedure codes increasing from 4,650 (ICD-9) to 75,801 (ICD-10) a SAS format catalog tool which adds text description of diagnoses and procedures for both ICD-9 and ICD-10 codes to administrative data thus reducing dependence on ICD Manuals to look-up code descriptions would be greatly beneficial for all users. This paper describes this tool.

This paper focuses on two examples. First a research example where administrative data is used to identify a cohort of hospital patients and text descriptions are used to refine the cohort and guide epidemiological analyses. Second an example featuring the data discrepancy report built using Report Writing Interface (RWI) or object oriented programming using the DATA _NULL_ to compare clinical data with administrative data to improve data quality of risk-adjusted CABG surgery operative mortality used evaluate hospital and surgeon performance.

Let's begin by building the SAS Format catalog used by both examples, and then examine the details of each example.

METHODS

CREATE A SAS FORMAT CATALOG FOR TWO EXAMPLES

This paper begins by describing a SAS Format Catalog which is used in both examples. Let's look at the SAS program code for the SAS format catalog and then return to each example.

Let's look at the details of how I created the SAS Format Catalog.

Part 1. Create four MS Excel files containing ICD-9 and ICD-10 diagnoses and procedures (Display 1).

At my office, the ICD-9 and ICD-10 diagnosis and procedure codes are stored in two SAS data set views which are updated in October of every year. Since each of four SAS formats will be created by importing an MS Excel file the first step is to create four MS Excel files containing 15,964 ICD-9 diagnosis codes and text definitions; 4,650 ICD-9 procedure codes and text definitions; 64,945 ICD-10 diagnosis codes and text definitions; and 75,801 ICD-10 procedure codes and text definitions.

Let's look at the program in Display 1 which accomplishes this objective.

The SAS data sets containing the diagnosis and procedure codes and their text are stored in SAS library icdcodes (Line 5).

First we read the diagnosis data set called "icdcodes.vw_diagnosis_dim (Lines 9-11). Also it is important to generate a list of variables in the SAS data set using PROC CONTENTS (Lines 12-13). It was discovered that these diagnosis codes do not have a label for "icd-9 code" or "icd-10 code" but instead have start and end dates given in SAS date time values. Recall the ICD-10 procedure and diagnosis codes started October 1, 2015. So we need to tabulate the start dates and end dates using PROC FREQ so we know how to select them in our program (Lines 14-16). Now we can select ICD-9 diagnosis codes with an end date less than or equal to "30SEP2015:00:00:00" and ICD-10 diagnosis codes with a start date equal to or greater than "01OCT2015:00:00:00" to create two new data sets diagnosis_icd9 and diagnosis_icd10 (Lines 17-24). Notice the use of "30SEP2015:00:00:00"DT which converts the display of SAS date time to a number that we can perform mathematical operations on.

Second we repeat the above and read SAS dataset icdcodes.vw_procedure_dim containing ICD-9 and ICD-10 procedure codes (Lines 29-32); list the variables (Lines 33-34); tabulate the start and end dates (Lines 35-37); and select the ICD-9 procedures and ICD-10 procedures to create two new SAS data sets named procedures_icd9 and procedures_icd10 (Lines 38-45).

Finally we use the SAS Enterprise Guide Filter and Sort utility to convert the above four SAS data sets to four MS Excel files for use in Parts 2 and 3 below, namely

1. DIAGNOSIS_ICD9_18APR2017.xls;
2. PROCEDURE_ICD9_18APR2017.xls;
3. DIAGNOSIS_ICD10_18APR2017.xlsx;
4. PROCEDURE_ICD10_18APR2017.xlsx.

---

2 *Choose Your Path to ICD-10, ICD-10 Leveraging an Asset Infographic.* (2016) American Health Information Management Association. Retrieved October 26, 2016, from http://www.ahima.org/icd10

Part 2. Create SAS Format for ICD-9 diagnoses and procedures (Display 2).

First we designate the location of the format catalog using SAS library forcat6 (Lines 23-27).

Next we import 15,964 ICD-9 diagnosis codes and descriptions from MS Excel file DIAGNOSIS_ICD9_18APR2017 (Lines 34-39). Options "out=" is used to create SAS data set diagnosis_icd9; "range=" is used to specify the range of the data in the MS Excel spreadsheet; "getnames=" is used to read variable names in the header row of each column.

Now we create three items (Lines 41-50); **start** which is a list of diagnosis codes; **label** which is a concatonation of the diagnosis code and the diagnosis text; and the format named "$dx_icd9f". The length of the variable label is then tabulated to confirm that the label has been created for all diagnosis codes (Lines 51-54).

Duplicate values of start and label are removed in Lines 56-64 using a special statement named "First.var." Next we print values of start, label and format name for all ICD-9 diagnosis codes (Lines 65-67). The format is then created in lines 68-70 from SAS data set dx_icd9 and stored in SAS library forcat6.

Simarily we create a format for ICD-9 Procedure Codes (Lines 77-112).

First we import 4,650 ICD-9 procedure codes and descriptions from MS Excel file PROCEDURE_ICD9_18APR2017 (Lines 77-82). Options "out=" is used to create SAS data set procedure_icd9; "range=" is used to specify the range of the data in the MS Excel spreadsheet; "getnames=" is used to read variable names in the header row of each column.

Now we create three items (Lines 84-92); **start** which is a list of procedure codes; **label** which is a concatonation of the procedure code and the procedure text; and the format named "$px_icd9f". The length of the variable label is then tabulated to confirm that the label has been created for all procedure codes (Lines 93-96).

Duplicate values of start and label are removed in Lines 98-106 using a special statement named "First.var." Next we print values of start, label and format name for all ICD-9 procedure codes (Lines 107-109). The format is then created in lines 110-112 from SAS data set px_icd9 and stored in SAS library forcat6.

Part 3. Create SAS Format for ICD-10 diagnoses and procedures (Display 3).

In Part 1 (above) we created two MS Excel files containing 64,945 ICD-10 diagnosis codes and 75,801 ICD-10 procedure codes with file extensions "xlsx". Here we use the SAS Enterprise Guide utility to read both files and create two SAS data sets containing ICD-10 diagnosis WORK.DIAGNOSIS_ICD10_18APR2017 and ICD-10 procedure WORK.PROCEDURE_ICD10_18APR2017 information. As a note the PROC IMPORT used to create ICD-9 diagnosis and procedure code SAS data sets required a file with extension "xls."

The SAS program code in Display 3 shows how we created the ICD-10 diagnosis and procedure formats.

As before we designate the location of the SAS format catalog using SAS library forcat6 and create a SAS library icd10dat for the two SAS data sets created above (Lines 28-32).

Next we read the SAS data set WORK.DIAGNOSIS_ICD10_18APR2017 containing the 64,945 ICD-10 diagnosis codes and text (created above) (lines 46-51).

Again we create three items (Lines 53-63); **start** which is a list of diagnosis codes; **label** which is a concatonation of the start variable and diagnosis text; and the format named "$dx_icd10f". Notice how forgiving SAS is to understand that I omitted the "RUN" statement. The length of the variable label is then tabulated to confirm that the label has been created for all diagnosis codes (Lines 64-67).

A new method is used to remove duplicate values of start and label in lines 78-80 using PROC SORT with options "out=" which is a data set containing duplicate records, if any are found; "uniqueout=" which is a data set with all unique observations; and "nouniquekey" which is required to make this all work. Next we print values of start, label and format name for a sample of 2,000 ICD-10 diagnosis codes (Lines 82-85). The format is then created in lines 86-88 from SAS data set dx_icd10new and stored in SAS library forcat6.

Simarily we create a format for ICD-10 Procedure Codes (Lines 90-143).

We begin by reading SAS data set WORK.PROCEDURE_ICD10_18APR2017 containing 75,801 ICD-10 procedure codes and text (created above) (lines 102-107).

Again we create three items (Lines 109-118); **start** which is a list of procedure codes; **label** which is a concatonation of the start variable and procedure text; and the format named "$px_icd10f". Again, notice how forgiving SAS is to understand that I omitted the "RUN" statement. The length of the variable label is then tabulated to confirm that the label has been created for all diagnosis codes (Lines 119-122).

A new method is used to remove duplicate values of start and label in lines 133-135 using PROC SORT with options "out=" which is a data set containing duplicate records, if any are found; "uniqueout=" which is a data set with all unique observations; and "nouniquekey" which is required to make this all work. Next we print values of start, label and format name for a sample of 2,000 ICD-10 procedure codes (Lines 137-140). The format is then created in lines 141-143 from SAS data set px_icd10new and stored in SAS library forcat6.

A useful bookeeping item. Note the SAS program code to import an MS Excel file (lines 39-44; and 95-100); and remove duplicates (lines 69-77; and 124-132); and is archived for future use. This is a wise practise if the new method does not work or later you want to return to the earlier method.

Now let's use the SAS format catalog in our first example of administrative data research.

RESULTS

DATA RESEARCH EXAMPLE

Let's look at details of the first example using administrative data for epidemiological study. Let's suppose a researcher wants to know the diagnoses and procedures for a cohort of hospital patients who had coronary artery bypass graft (CABG) surgery in March, June, September, and December of 2015. This data year was selected because it includes both ICD-9 and ICD-10 diagnosis and procedure codes. This researcher wants the results to include not only the diagnosis and procedure codes but description of those codes in an MS Excel file to study results of interest either visually using MS Excel tools or programmatically using SAS for two reasons. First to understand the secondary diagnoses and procedures of patients who had CABG surgery. Second to thereby refine the cohort by adding or removing procedures or diagnoses included in the study. Finally the researcher would like an automated solution that is flexible enough to change the cohort study, for example, to patients with liver disease diagnosis and again review the results in MS Excel.

The SAS format catalog contains a universal reference list of all procedures and diagnoses codes with their descriptions for both ICD-9 and ICD-10. This format catalog is the foundation for both examples (Display 1-3).

For the adminstrative data research example we need the foundation and SAS program code contained in four parts described below and listed in Appendix 1-2.

Part 1. Create Two ICD-10 Macro Lists that Define CABG Surgery (Appendix 1).

In lines 10-33 PROC IMPORT is used to read 296 procedure codes from an MS Excel file and create SAS data set pxcabg_icd10. These procedure codes define CABG surgery and quotation markes are placed around each code for visibility. Then PROC SQL reads this data set and creates one very large macro list named &cabgicd10 containing our 296 procedure codes.

The process is repeated in lines 35-58 where PROC IMPORT is used to read 4,645 procedure codes from an MS Excel file and create SAS data set pxcabgexcl_icd10. These procedure codes define CABG surgery exclusions and quotation markes are placed around each code for visibility. Then PROC SQL reads this data set and creates one very large macro list named &cabgexclicd10 containing our 4,645 procedure codes.

Both macro lists will be used in Part 2 which constructs the cohort study population. Isolated CABG surgery means that no other major procedure, such as valve repair or carotid endarterectomy was performed at the same time as the bypass surgery. Isolated CABG with another major procedure (exclusion) is called Non-isolated CABG. A Non-isolated CABG surgery has an higher risk of mortality. In order to compare results, providers are given more credit for these surgeries in a risk-adjusted model that calculates performance scores published in an annual CABG surgery report.

Lines 60-138 are used to compile several macros used in Part 2.

Part 2. Construct the Cohort Study Population (Appendix 2 lines 1-405)

The SAS format catalog is located in SAS library forcat6 (lines 9-12). Note the statement "options fmtsearch=(forcat6). The default location for a format catalog is the default SAS library work. Several SAS formats are created in lines 17-132.

Lines 139 to 374 contain the SAS program code to select isolated and non-isolated CABG surgery patients. Let's look at only a few highlights of how the cohort was created.

First we examine lines 139-204 which creates SAS data set beate_partone which is a subset of data containing CABG surgery cases. Lines 168-169 show how discharge date (date patient left hospital) is used to divide the patient data into a group with ICD-9 codes and a second group using ICD-10 codes. Notice lines 180-183 where macro %effdoS looks for 31 ICD-9 codes that identify a CABG procedure. Compare this to line 194 where the same macro %effdoS uses the macro list created in Part 1 "&cabgicd10" to look for 296 ICD-10 codes that identify CABG procedures. It would not be practical to hard code all 296 codes in the SAS progam. In addition the macro list created in Part 1 can be updated when changes are needed with minimal effort.

Next SAS reads the data set beate_partone and identifies Isolated and Non-isolated CABG surgery cases (lines 205-374). For the ICD-9 patient group (line 228) we use macro %effdo with ICD-9 procedure code 36.1 to identify the date of the CABG procedure (line 232) followed by the use of macro %effdodtd to search for 50 CABG surgery exclusions that will identify non-isolated CABG surgery (lines 242-310).

For the ICD-10 patient group (line 333), we use macro %effdo with a list of 296 ICD-10 CABG procedures contained in macro list &cabgicd10 to identify the date of the CABG surgery (line 336). Next one call to macro %effdodtd using

macro list &cabgexclicd10 is used to search for 4,645 CABG surgery exclusions (line 347). Again note the use of macro lists &cabgicd10 and &cabgexclicd10 to store a large list of ICD-10 procedure codes and how easy it is to update the MS Excel containing those codes to create new lists.

Now we select the patient data in lines 375 to 384 for use in Part 3.

Also note throughout the SAS program several uses of PROC FREQ to tabulate variables that confirm the accuracy of the SAS program to extract and identify specific information. This is a very good practise.

Part 3. Construct Diagnosis and Procedures Code Descriptions for Both ICD-9 and ICD-10 (Appendix 2 lines 407-701).

The administrative data contains ICD-9 and ICD-10 diagnosis and procedure codes where the decimal point has been removed. The SAS format catalog uses the decimal point, so our task is to reconstruct diagnosis and procedure codes with the decimal so the format catalog will function correctly.

We begin with setting the length of several variables (lines 409-415). In administrative data ICD-9 diagnosis has no decimal point but have an implied decimal after the third number or character from the left. Therefore lines 424-449 reconstruct the diagnosis code using; the first three numbers of the diagnosis, a period; and the remaining numbers of the diagnosis value. Here the original administrative diagnosis code is given in variables diag_p for principal diagnosis and odiag1 to odiag 24 for 24 other diagnosis codes. The new diagnosis value with the decimal is given as prdx for principal diagnosis, and dx1-dx24 for 24 other diagnosis codes. Next we apply the ICD-9 diagnosis format $dx_icd9f to the diagnosis code and create new text variable containing the diagnosis code and description as prdx_txt for principal diagnosis and dx1_txt to dx24_txt for the other diagnoses (lines 458-483). The concatenate call function “catx” is great for this purpose.

We repeat this process for ICD-9 procedure codes. In administrative data ICD-9 procedure has no decimal point but have an implied decimal after the second number or character from the left. Therefore lines 488-509 reconstruct the procedure code using; the first two numbers of the procedure, a period; and the remaining numbers of the procedure value. Here the original administrative procedure code is given in variables proc_p for principal procedure and oproc1 to oproc20 for 20 other procedure codes. The new procedure value with the decimal is given as prpx for principal procedure, and px1-px20 for 20 other procedure codes. Next we apply the ICD-9 procedure format $px_icd9f to the procedure code and create new text variable containing the procedure code and description as prpx_txt for principal procedure and px1_txt to px20_txt for the other procedures (lines 517-538). The concatenate call function “catx” is great for this purpose.

In administrative data ICD-10 diagnosis has no decimal point but also has an implied decimal after the third number or character from the left. Therefore lines 555-580 reconstruct the diagnosis code using; the first three numbers of the diagnosis, a period; and the remaining numbers of the diagnosis code. Here the original administrative diagnosis code is given in variables diag_p for principal diagnosis and odiag1 to odiag 24 for 24 other diagnosis codes. The new diagnosis value with the decimal is given as prdx for principal diagnosis, and dx1-dx24 for 24 other diagnosis codes. Next we apply the ICD-10 diagnosis format $dx_icd10f to the diagnosis code and create new text variable containing the diagnosis code and description as prdx_txt for principal diagnosis and dx1_txt to dx24_txt for the other diagnoses (lines 589-614). The concatenate call function “catx” is great for this purpose.

Finally, in administrative data ICD-10 procedure codes do not have a decimal point. Therefore we do not have to use the “catx” call function to reconstruct the procedure codes. Here original administrative procedure codes given as proc_p as principal procedure, and oproc1 to oproc20 for 20 other procedures is assigned to the new variables of prpx as principal procedure, and px1 to px20 for 20 other procedures (lines 616-636). Again we apply the ICD-10 procedure format “$px_icd10f” to the procedure codes and create new text variables containing the procedure code and description as prpx_txt for principal procedure, and px1_txt to px20_txt for 20 other procedures (lines 644-665).

Also note lines 540-549 and lines 667-674 where we use a date format “mmddyy10” to create the procedure dates for ICD-9 and ICD-10 respectively.

Last but not least, we add hospital names to our SAS data set (lines 680-697).

Part 4. Send Results to MS Excel (Appendix 2; lines 699-895)

In this last step the tagset ExcelXP is used to by PROC REPORT to read the final SAS data set CabgData_Hosp and send the results to MS Excel (lines 699-895). The location and name of the MS Excel file is defined in lines 706, and 710. Several options in lines 711-718 are used to improve the appearance of the spreadsheet. Namely “absolute_column_width=” to specific column width; “autofit_height=” enables the height of the rows to adjust to the data; “autofilter='Yes'” provides a filter and sort utility at the top of each column; “frozen_headers='1'” fixes the header row as stationary while the scroll bars apply only to data rows; “sheet_name=” provides spreadsheet name; and “embedded_titles='Yes'” includes titles in the worksheet body.

Style statements in lines 722-726 change the font size and appearance for several parts of the report. The column statement lists variable names for each column of the report (lines 728-733).

Variables in the report are defined using statements of “define,” “display,” and “style”. For example, in lines 739-740 we *define* a column of the report using variable hospitalfull which is the license number and name of the hospital. The variable is *displayed* in the report with column heading “Hospital Name.” The style attributes of the column include bold font, left justified, and font_size of 1. In addition the background color for the column is “very light moderate blue” with a foreground (font color) or “purple” (lines 737-738).

Additional patient identifier variables include discharge month and year; type of cabg surgery (Isolated or Non-isolated); birthdate; admission date; date of CABG surgery; discharge date (Lines 735-750).

Next for both ICD-9 and ICD-10 we have primary diagnosis and 24 other diagnoses (lines 752-801); and primary procedure and 20 other procedures for both ICD-9 and ICD-10 (lines 803-886).

Notice the use of alternating color schemes to make the columns visually appealing. Notice the use of the same variables; diagnoses prdx_txt, dx1_txt to dx24_txt, prpx_txt, and procedures px1_txt to px20_txt for both ICD-9 and ICD10 codes. Recall that the ICD-9 version is used if discharge of the patient occurred up to and including September 30, 2015 and the ICD-10 version is used if discharge of the patient occurred beginning with October 1, 2015.

Appendix 3 shows small section of the resulting MS Excel file for Administrative Data Research. Note the first six rows labeled “09-2015” shows the first two ICD-9 diagnoses, and the first two ICD-9 procedures with procedure date. The second set of six rows labeled “12-2015” shows the first two ICD-10 diagnoses, and the first two ICD-10 procedures with procedure date. Notice the code and text description is provided for both diagnoses and procedures for both ICD-9 and ICD-10. The full MS Excel file with all 7 patient identifiers, 25 diagnoses, and 21 procedures is available upon request.

DATA RESEARCH EXAMPLE CONCLUSION

A cohort population was constructed from administrative data consisting of CABG surgery patients in March, June, September, and December for epidemiological and clinical research. A SAS format catalog was used to add ICD-9 and ICD-10 diagnosis and procedure codes and descriptions to the cohort. We used 2015 data year because it contained both ICD-9 and ICD-10 codes however we could use any year such as 2014 with all ICD-9 codes or 2016 with all ICD-10 codes and the program code would function correctly. Hospital and patient identifiers along with diagnosis and procedure codes and text descriptions for both ICD-9 and ICD-10 were sent to MS Excel for three reasons. First to verify diagnoses and procedures selected for each patient record, and consider revising the cohort. Second for preliminary graphical, and tabular analyses using MS Excel tools. Third to consider importing results into SAS Enterprise Guide for statistical analyses. Finally this automated process can be quickly revised to create an unlimited number of cohort populations using diagnoses codes or procedure codes or both.

Now let’s use the SAS format catalog in our second example of a custom built data discrepancy report.

A CUSTOM REPORTING EXAMPLE

The second example shows how the SAS format catalog was incorporated into a custom built data discrepancy report.

The driving force for this example was a need to create hospital level data discrepancy reports which compare California CABG Outcome Reporting Program (CCORP) clinical data, to OSHPD’s hospital administrative data, to verify risk factors used in a risk-adjusted operative mortality model (Springborn 2013). This model provides hospital and surgeon performance scores for coronary artery bypass graft surgery (CABG) in California. Many changes have occurred in this data discrepancy report. This paper focuses the SAS format catalog which was updated to include diagnoses and procedure code text descriptions for both ICD-9 and ICD-10.

This example also used ODS Report Writing Interface. In today’s sophisticated business environment preparing easy to read clearly designed reports to convey your message in a clear and concise manner can enhance your organization’s credibility and reputation. The review below emphasizes the unlimited power of this technology to create a custom report using this SAS format catalog for your own reporting needs. ODS Report Writing Interface and ODS LAYOUT are two new production features offered in the Output Delivery System (ODS) in SAS® 9.4. Several papers describe ODS Report Writing Interface and ODS LAYOUT including; O’Connor (2013), Lund (2013), Huff (2014), Kummer (2014), Huntley (2015), Lund (2015), Zender (2016).

O’Connor (2013) describes the advantages of ODS Report Writing Interface. “The ODS Report Writing Interface was designed exclusively to address the custom report writing requirements that go far beyond the traditional tabular output produced by procedures. The custom report writing capabilities are also more commonly referred to as DATA _NULL_ report writing which has long been an integral part of the SAS reporting solution. The ODS Report Writing Interface is intended to fully embrace ODS features such as proportional fonts, traffic lighting, colors, images, and Unicode characters, while at the same time providing pixel perfect placement capabilities. This interface has methods to allow you to produce tables, text, insert images, and control the placement of everything in the report. This interface is not only fully integrated with all the capabilities of the ODS System, but also takes advantage of the rich programming features that the DATA step offers, such as conditional logic, formatting capabilities, by-group processing, arrays, and a wealth of other features. The ODS Report Writing Interface is an object-oriented language

that provides you with flexibility and control to address even the most rigid reporting requirements with ease" (See O'Connor 2013, p.1).

Kummer (2014) states "There are two types of layouts available within ODS LAYOUT: the absolute layout and gridded layout. "But in general, starting a layout (no matter the type) with the ODS LAYOUT statement creates what we call a layout container. This container defines the space where the different kinds of output objects can be arranged in. The ODS LAYOUT statement controls the type, size, and placement of the container, but also the behavior within it. Depending on the type of layout, this container can have a fixed size and position, or it can be sized and positioned dynamically. Individual region containers are specified inside this layout container. These region containers are controlled with the ODS REGION statement. Again, depending on the type of layout, these region containers can have a fixed size and position or they can be sized and positioned dynamically. The actual output objects are displayed inside the individual region containers. A single region can display one-to-many output objects" (See Kummer 2014 p.1).

Kummer (2014) states "The biggest strength of the ODS Report Writing Interface is the possibility to display any variable that is available within the DATA step anywhere inside the defined space of the report. Besides presenting the data in tabular form, the ODS Report Writing Interface can also use a gridded or absolute layout. The concept is exactly the same as for ODS LAYOUT. Instead of specifying ODS LAYOUT and ODS REGION statements, the respective methods of the Report Writing Interface are used. The biggest difference between the two is the type of output that is displayed within the regions of the layout. The ODS Report Writing Interface is confined to the information available within the DATA step. It can display only individual, computed, or aggregated values from variables available in the Program Data Vector (PDV)" (See Kummer 2014, p.10). In addition, images can be displayed, and titles, footnotes, and page breaks can be controlled within the DATA step.

The bonus is that Kummer (2014) provides "ready to use" program code for several examples. "If and how you use ODS LAYOUT or the ODS Report Writing Interface depends on your reporting needs. Both provide new ways to arrange different types of output within a report with ease. Before ODS LAYOUT and the ODS Report Writing Interface were available, reports like the ones presented in Kummer (2014) were just not possible. Perhaps the use of macros, data manipulation, and reporting procedures would get you close but the code necessary is large, complicated, and hard to maintain" (Kummer 2014, p.13).

Springborn (2012; 2013) describes how the administrative data is used in a data discrepancy report. He states "The California Coronary Artery Bypass Graft (CABG) Outcome Reporting Program (CCORP) is the largest public reporting program on CABG surgery outcomes in the United States. Each year OSHPD releases the *California Report on Coronary Artery Bypass Graft Surgery* which presents findings from analyses of data collected from California-licensed hospitals where surgeons performed adult isolated CABG surgery. This report features risk-adjusted operative mortality used to evaluate hospital and surgeon performance. Surgeon and Hospital level reports for *The California Report on Coronary Artery Bypass Graft Surgery, California CABG Outcomes Reporting Program* can be found at (http://www.oshpd.ca.gov/HID/)."

Springborn (2013) gives more details of the data discrepancy report. He states "CCORP reviews data submitted by each hospital for completeness and errors. The data discrepancy report compares the CCORP clinical data to OSHPD's hospital administrative data source, the Patient Discharge Data (PDD). Hospitals are asked to review and account for discrepancies between the two data sources via patient medical chart review to verify that coding for the following eleven data variables (risk factors) are consistent. Data variables include: 1) CABG Surgeries Reported; 2) Isolated CABG Reporting; 3) CABG+Valve Reporting; 4) Resuscitation Prior to CABG Procedure; 5) CABG surgery deaths; 6) Post-Operative Stroke; 7) Dialysis Requirement; 8) Prolonged Ventilation; 9) Reoperation for Bleed; 10) Deep Sternal Wound Infection; and 11) Graft Occlusion."

Springborn (2013) explains the importance of risk adjusting patient mortality outcomes. He states "It is important to verify that both data sources reported the same risk factor value because these risk factors will determine hospital and surgeon performance published in the public report. To make fair comparisons of care delivered by different healthcare providers, it is necessary to adjust for differences in severity of illness (case mix) of patients across providers. CCORP "levels the playing field" by considering the pre-operative condition of each patient. Providers that handle more complex cases receive a larger risk-adjustment weight in the risk model, and providers that handle less complex cases receive a smaller weight. Thus, hospitals and surgeons treating sicker patients are not at a disadvantage when their performance is compared with other surgeons and hospitals." Springborn (2013) provides the SAS program code used to build the data discrepancy report.

Now let's look at a data discrepancy report example.

A sample data discrepancy report is provided from a sample hospital from two different reporting periods so we can illustrate the use of the SAS format catalog for both ICD-9 and ICD-10. Patient identifiers have been masked. (Appendices 4-7). This report includes eleven sections corresponding to the eleven risk factors described in the Help Sheet on page 1 of the report (Appendix 4). Each section compares a particular risk factor between the California CABG Outcome Reporting Program (CCORP) clinical data, to OSHPD's hospital administrative data. The gold standard for the report is the administrative data. If the risk factor is absent from CCORP data but present in administrative data we say the risk factor is under-reported. If the converse is true then the risk factor is over-

reported. This hospital level data discrepancy report identifies specific actions hospitals should take for both under-reporting and over-reporting for each risk factor.

Let's review details of the first section for CABG-Reporting (Appendix 5). For this hospital the report identified 3 under-reported patient records "Missing CABG Cases" in which clinical data did not report a CABG surgery but administrative data did report CABG surgery. This report also identified 21 over-reported patient records "NON-CABG cases" in which clinical data reported CABG surgery but the administrative data did not. Let's look at an example of each using ICD-9 and ICD-10 codes.

Appendix 6 gives an ICD-10 under-reporting example from the first half of 2016 (January to June) where the patient identifiers of social security number, sex, birth date, surgery date, and discharge date have been masked. This patient record shows 12 ICD-10 diagnoses, and 7 ICD-10 procedures both complete with code value and description. So the administrative data reported a CABG surgery as ICD-10 procedure code 021109W performed on March 29, 2016 but the clinical data did not report CABG surgery for this patient. So the action in yellow is to ask the hospital staff, "Should this case be added to CCORP?." Hospital staff then review their records and see if the patient record should be added to CCORP.

Appendix 7 gives an ICD-9 example from the first half of 2015 where patient identifiers of medical record number, sex, birth date, surgery date, and discharge date have been masked. This patient record shows 15 ICD-9 diagnoses, and 6 ICD-9 procedures both complete with code value and description. Again, the gold standard for the report is the administrative data. So the administrative data did not report a CABG surgery because ICD-9 procedure code 36.1 is missing. However the clinical data did report CABG surgery for this patient. So the action in yellow is to ask the hospital staff, "Should this case be removed from CCORP?." Hospital staff then review their records and see if the patient record should be removed from CCORP.

This report continues to identify under-reporting and over-reporting for each of the remaining 10 risk factors. Every six months this report is created for each of 125 CCORP hospitals.

A CUSTOM REPORTING EXAMPLE CONCLUSION

The SAS format catalog was incorporated into a custom built data discrepancy report. This hospital level data discrepancy report compares clinical and administrative data to verify risk factors used in a risk-adjusted operative mortality model which provides hospital and surgeon performance ratings for coronary artery bypass graft surgery (CABG) in California.

This data discrepancy report has been a great success. Managers were very happy with this report because they could customize how the information was presented to hospitals. Several changes have been made to this report in response to changes in the clinical and administrative data; changes to ICD-10 procedures and diagnoses; and adding new risk factors. Using ODS Report Writing Interface and ODS Layout have greatly facilitated making these changes in a timely manner.

This tool has benefited California health care providers because hospitals and surgeons have praised CCORP data discrepancy reports for their usefulness and clarify of presentation to show discrepancies and actions recommended to hospitals.

In today's sophisticated business environment preparing easy to read clearly designed reports to convey your message in a clear and concise manner can enhance your organization's credibility and reputation. This example of building a custom report using ODS Report Writing Interface and ODS Layout enables all users of administrative data to create their own custom report for their own data reporting needs.

Display 1. Create Four SAS Data Sets for ICD-9 ICD-10 Diagnoses and Procedures.

```
1  ********************************************************************************************;
2  *** IMPORT DATA FILES FOR ICD9 AND ICD10 AND CREATE SEPARATE WORKSHEETS FOR PROCEDURES
3  AND DIAGNOSES ;
4
5  LIBNAME icdcodes meta library= "DW_INPATIENT" repname= "Foundation" ;
6
7  ************************************************************;
8  title1 "QUICK VIEW OF DIAGNOSES";
9  data diagnoses;
10 set icdcodes.vw_diagnosis_dim;
11 run;
12 proc contents data=diagnoses fmtlen varnum ;
13 run;
14 proc freq data=diagnoses ;
15 tables start_date end_date /norow nocol ;
16 run;
17 data diagnosis_icd9 diagnosis_icd10;
18 length start_datecmp end_datecmp $9. ;
19 set diagnoses;
20 start_datecmp= compress(put(start_date,datetime20.));
21 end_datecmp= compress(put(end_date,datetime20.));
22 if end_date le "30SEP2015:00:00:00"DT then output diagnosis_icd9 ;
23 if start_date ge "01OCT2015:00:00:00"DT then output diagnosis_icd10;
24 run;
25 proc sort data=diagnosis_icd9 ;
26 by diagnosis_code;
27 run;
28 ************************************************************;
29 title1 "QUICK VIEW OF PROCEDURES";
30 data procedures;
31 set icdcodes.vw_procedure_dim;
32 run;
33 proc contents data=procedures fmtlen varnum ;
34 run;
35 proc freq data=procedures ;
36 tables start_date end_date /norow nocol ;
37 run;
38 data procedures_icd9 procedures_icd10;
39 length start_datecmp end_datecmp $9. ;
40 set procedures;
41 start_datecmp= compress(put(start_date,datetime20.));
42 end_datecmp= compress(put(end_date,datetime20.));
43 if end_date le "30SEP2015:00:00:00"DT then output procedures_icd9 ;
44 if start_date ge "01OCT2015:00:00:00"DT then output procedures_icd10;
45 run;
46 proc sort data=procedures_icd9;
47 by procedure_code;
48 run;
49 quit;
```

Display 2. Create SAS Format for ICD-9 Diagnosis and Procedure Codes.

```
1   ********************************************************************************************;
2   ** BUILD A SAS FORMAT CATALOG CONSISTING OF ICD-9 DIAGNOSIS AND PROCEDURE CODES            ;
3   ********************************************************************************************;
4
5   proc format ;
6   value lenfmt
7   0-1= "Missing"
8   2-10= "2-10"
9   11-20= "11-20"
10  21-30= "21-30"
11  31-40= "31-40"
12  41-50= "41-50"
13  51-60= "51-60"
14  61-70= "61-70"
15  71-80= "71-80"
16  81-90= "81-90"
17  91-100= "91-100"
18  101-110= "101-110"
19  111-120= "111-120"
20  121-high= "121-plus";
21  run;
22
23  ** Set location for catalogue  18April2017;
24  %Let TargetZ=\\SBPWSAS1\Development\HOC\CCORP_Projects\SpecialProjects\
25  Springborn_Pgm_Updates\SASFormatCatalog_07April2017\;
26
27  LIBNAME forcat6 "&TargetZ.format_catalog6\";
28
29  ********************************************************************************************;
30  ** STEP A :: UPDATE ICD9 DIAGNOSIS CODES FOR SAS FORMAT CATALOG 6                           ;
31  ** PDD DIAGNOSIS CODES FROM DW_INPATIENT.VW_DIAGNOSIS_DIM 18APRIL2017                      ;
32  ********************************************************************************************;
33
34  ************************************************************************;
35  ** Read in data for format DIAGNOSIS ICD9 CODES;
36  title1 "VERIFY DIAGNOSIS FORMAT";
37  proc import datafile="&TargetZ.UpdatedDxPxFiles\DIAGNOSIS_ICD9_18APR2017.xls"
38    out=diagnosis_icd9 dbms=xls replace; range="Sheet1$D1:I15965"; getnames=yes;
39    run;
40
41  ** Add SAS Format "$dx_icd9f." to format catalogue ;
42  DATA dx_icd9 (keep= start label fmtname lenlabel);
43    LENGTH start $ 6 label $ 125;
44    set diagnosis_icd9
45  (keep= diagnosis_code diagnosis_full_desc
46  rename= (diagnosis_code=start diagnosis_full_desc=labeltext ));
47
48      label = TRIM(LEFT(start))||": "||strip(lowcase(labeltext));
49  lenlabel= length(label);
50      fmtname = "$dx_icd9f";
51  proc freq data=dx_icd9;
52    tables lenlabel /norow nocol nopercent;
53  format lenlabel lenfmt. ;
54  run;
55
56  ** Remove Duplicate Entries;
57  proc sort data=dx_icd9 ;
58  by start;
59  run;
60  data dx_icd9;
61    set dx_icd9;
62  by start;
```

```
63   if FIRST.start;
64    run;
65   proc print noobs data=dx_icd9 ;
66   var start label fmtname;
67    run;
68   proc format library=forcat6 CNTLIN= dx_icd9;
69    run;
70   title1; title2;
71
72   ********************************************************************************************;
73   ** STEP B :: UPDATE ICD9 PROCEDURE CODES FOR SAS FORMAT CATALOG 6                          ;
74   ** PDD PROCEDURE CODES DW_INPATIENT.VW_PROCEDURE_DIM  18APRIL2017                         ;
75   ********************************************************************************************;
76
77   ***************************************************************************;
78   ** Read in data for format PROCEDURE ICD9 CODES;
79   title1 "VERIFY PROCEDURE FORMAT";
80   proc import datafile="&TargetZ.UpdatedDxPxFiles\PROCEDURE_ICD9_18APR2017.xls"
81    out=procedure_icd9 dbms=xls replace; range="Sheet1$D1:I4651"; getnames=yes;
82    run;
83
84   ** Add SAS Format "$px_icd9f." to format catalogue ;
85   DATA px_icd9 (keep= start label fmtname lenlabel);
86    LENGTH start $ 6 label $ 125;
87    set procedure_icd9
88   (keep= procedure_code procedure_full_desc
89   rename= (procedure_code= start procedure_full_desc= labeltext ));
90      label = TRIM(LEFT(start))||": "||strip(lowcase(labeltext));
91   lenlabel= length(label);
92      fmtname = "$px  icd9f";
93   proc freq data=px_icd9;
94    tables lenlabel /norow nocol nopercent;
95   format lenlabel lenfmt. ;
96   run;
97
98   ** Remove Duplicate Entries;
99   proc sort data=px_icd9 ;
100  by start;
101  run;
102  data px_icd9;
103   set px_icd9;
104  by start;
105  if FIRST.start;
106   run;
107  proc print noobs data=px_icd9 ;
108  var start label fmtname;
109   run;
110  proc format library=forcat6 CNTLIN= px_icd9;
111   run;
112  title1; title2;
113
114  QUIT;
```

Display 3. Create SAS Format for ICD-10 Diagnosis and Procedure Codes.

```
1   ********************************************************************************************;
2   ** BUILD A SAS FORMAT CATALOG CONSISTING OF ICD-10 DIAGNOSIS AND PROCEDURE CODES           ;
3   ********************************************************************************************;
4
5   proc format ;
6   value lenfmt
7   0-1= "Missing"
8   2-10= "2-10"
9   11-20= "11-20"
10  21-30= "21-30"
11  31-40= "31-40"
12  41-50= "41-50"
13  51-60= "51-60"
14  61-70= "61-70"
15  71-80= "71-80"
16  81-90= "81-90"
17  91-100= "91-100"
18  101-110= "101-110"
19  111-120= "111-120"
20  121-150= "121-150"
21  151-175= "151-175"
22  176-200= "176-200"
23  201-225= "201-225"
24  226-250= "226-250"
25  251-high= "251-plus";
26  run;
27
28  ** Set location for catalogue and icd10 data 07April2017;
29  %Let TargetZ=\\SBPWSAS1\Development\HOC\CCORP_Projects\SpecialProjects\
30  Springborn_Pgm_Updates\SASFormatCatalog_07April2017\;
31  LIBNAME forcat6 "&TargetZ.format_catalog6\";
32  Libname icd10dat "&TargetZ.datasets\";
33
34  ********************************************************************************************;
35  ** STEP A :: UPDATE ICD10 DIAGNOSIS CODES FOR SAS FORMAT CATALOG 6                          ;
36  ** PDD DIAGNOSIS CODES USING DW_INPATIENT.VW_DIAGNOSIS_DIM 18APR2017                       ;
37  ********************************************************************************************;
38
39  ***************************************************************************;
40  ** Read in data for format DIAGNOSIS ICD10 CODES;
41  /*title1 "VERIFY DIAGNOSIS FORMAT";
42  proc import datafile="&TargetZ.UpdatedDxPxFiles\STAGE_DIAGNOSIS_CODE_STGICD9.xls"
43    out=diagnosis_icd9 dbms=xls replace; range="Sheet1$A1:W15965"; getnames=yes;
44    run; */
45
46  ** Keep only variables for format;
47  title1 "ICD 10 DIAGNOSIS STATISTICS";
48  data icd10dat.diagnosis_icd10;
49  set WORK.DIAGNOSIS_ICD10_18APR2017
50  (keep= DIAGNOSIS_CODE DIAGNOSIS_FULL_DESC);
51  run;
52
53  ** Add SAS Format "$dx_icd10f." to format catalogue ;
54  DATA dx_icd10 (keep= start label fmtname lenstart lenlabel);
55    LENGTH start $ 8 label $ 240;
56    set icd10dat.diagnosis_icd10
57  (keep= diagnosis_code diagnosis_full_desc
58  rename= (diagnosis_code=start diagnosis_full_desc=labeltext ));
59
60      label = TRIM(LEFT(start))||": "||strip(lowcase(labeltext));
61  lenstart= length(start);
62  lenlabel= length(label);
63      fmtname = "$dx_icd10f";
```

```
proc freq data=dx_icd10;
  tables lenstart lenlabel /norow nocol nopercent;
format lenlabel lenfmt. ;
run;

** Remove Duplicate Entries, New method 27July2015;
/*proc sort data=dx_icd10 ;
by start;
run;
data dx_icd10;
  set dx_icd10;
by start;
if FIRST.start;
  run;  */
proc sort data=dx_icd10 out=dup_dxicd10 uniqueout=dx_icd10new nouniquekey ;
by start label;
run;

proc print noobs data=dx_icd10new(obs=2000) ;
title2 "PRINT OUT FIRST 2000 OF ICD 10 DIAGNOSIS FORMATS";
var start label fmtname;
  run;
proc format library=forcat6 CNTLIN= dx_icd10new;
  run;
title1; title2;

*************************************************************************************************************************;
** STEP B :: UPDATE ICD9 PROCEDURE CODES FOR SAS FORMAT CATALOG 6                                                       ;
** PDD PROCEDURE CODES USING DW_INPATIENT.VW_PROCEDURE_DIM  18APR2017                                                   ;
*************************************************************************************************************************;

*************************************************************************;
** Read in data for format PROCEDURE ICD10 CODES;
/*title1 "VERIFY PROCEDURE FORMAT";
proc import datafile="&TargetZ.UpdatedDxPxFiles\STAGE_PROCEDURE_CODE_STGICD9.xls"
  out=procedure_icd9 dbms=xls replace; range="Sheet1$A1:S4649"; getnames=yes;
  run; */

*** Keep Only Variables to Format;
title1 "ICD 10 PROCEDURE STATISTICS";
data icd10dat.procedure_icd10;
set WORK.PROCEDURE_ICD10_18APR2017
(keep= PROCEDURE_CODE PROCEDURE_FULL_DESC);
run;

** Add SAS Format "$px_icd10f." to format catalogue ;
DATA px_icd10 (keep= start label fmtname lenstart lenlabel);
  LENGTH start $ 7 label $ 175;
  set icd10dat.procedure_icd10
(keep= procedure_code procedure_full_desc
rename= (procedure_code= start procedure_full_desc= labeltext ));
    label = TRIM(LEFT(start))||": "||strip(lowcase(labeltext));
lenstart= length(start);
lenlabel= length(label);
    fmtname = "$px  icd10f";
proc freq   data=px_icd10;
  tables lenstart lenlabel /norow nocol nopercent;
format lenlabel lenfmt. ;
run;

** Remove Duplicate Entries, New method 27July2015;
/*proc sort data=px_icd10 ;
by start;
run;
data px_icd10;
  set px_icd10;
```

```
130     by start;
131     if FIRST.start;
132       run; */
133     proc sort data=px_icd10 out=dup_pxicd10 uniqueout=px_icd10new nouniquekey ;
134     by start label;
135     run;
136
137     proc print noobs data=px_icd10new(obs=2000);
138     title2 "PRINT OUT FIRST 2000 OF ICD 10 PROCEDURE FORMATS";
139     var start label fmtname;
140       run;
141     proc format library=forcat6 CNTLIN= px_icd10new;
142       run;
143     title1; title2;
144
145     QUIT;
```

CONCLUSION

A SAS format catalog which adds *161,360* ICD-9 and ICD-10 diagnosis and procedure code descriptions to administrative data would greatly benefit all users and encourage new uses of administrative data. SAS program code was provided for the catalog and both examples.

A research example was given where the SAS format catalog adds diagnosis and procedure descriptions to a cohort population constructed from administrative data diagnoses and procedures. Results were sent to MS Excel to: verify diagnoses and procedures selected for each patient record and consider revising the cohort; and conduct preliminary graphical, tabular, and statistical analyses using MS Excel tools. An unlimited number of cohort populations can be created.

In today's sophisticated business environment preparing easy to read clearly designed reports to convey your message in a clear and concise manner can enhance your organization's credibility and reputation. A custom reporting example where the SAS format catalog was incorporated into a custom data discrepancy report which compares clinical and administrative data to verify risk factors used in a risk-adjusted operative mortality model which provides hospital and surgeon performance scores for coronary artery bypass graft surgery (CABG) in California was given. An unlimited number of custom reports for administrative data users reporting needs is possible.

TRADEMARK CITATION

ACKNOWLEDGEMENTS

The author would like to thank OSHPD staff for motivating me to write and present this paper. Also to Daniel O'Connor and Daniel Kummer at SAS Institute for sharing several papers on ODS Report Writing Interface and ODS Layout. In addition I could not have completed this project without my mentor Art Carpenter who gave me several code examples.

REFERENCES

Huntley, S. 2015. "An Insider's Guide to ODS LAYOUT Using SAS 9.4" *Proceedings of the SAS Global Forum 2015 Conference. Paper 1836-2015. Cary,N.C.:SAS Institute Inc.* Available at https://support.sas.com/resources/papers/proceedings15/SAS1836-2015.pdf

Huff, G. 2014. "Simple ODS Tipes to Get RWI (Really Wonderful Information) from your RWI (Report Writing Interface)" *Proceedings of the SAS Global Forum 2014 Conference. Paper 1689-2014. Cary,N.C.:SAS Institute Inc.* Available at http://support.sas.com/resources/papers/proceedings14/1689-2014.pdf

Kummer, D. 2014. "Toe to Toe: Comparing ODS LAYOUT and the ODS Report Writing Interface" *Proceedings of the SAS Global Forum 2014 Conference.* Paper SAS330-2014*. Cary,N.C.:SAS Institute Inc.* Available at https://support.sas.com/resources/papers/proceedings14/SAS330-2014.pdf

Lund, P. 2013. "Have it Your Way: Creating Reports with the Data Step Report Writing Interface" *Proceedings of the SAS Global Forum 2013 Conference. Paper 040-2013*. *Cary,N.C.:SAS Institute Inc.* Available at http://support.sas.com/resources/papers/proceedings13/040-2013.pdf

Lund, P. 2015. "Something Old, Something New… Flexible Reporting with DATA Step-based Tools" *Proceedings of the SAS Global Forum 2015 Conference. Paper 3496-2015. Cary,N.C.:SAS Institute Inc.* Available at https://support.sas.com/resources/papers/proceedings15/3496-2015.pdf

O'Connor, D. 2013. "Take Home the ODS Crown Jewels: Master the New Production Features of ODS LAYOUT and Report Writing Interface Techniques" *Proceedings of the SAS Global Forum 2013 Conference. Paper 015-2013. Cary,N.C.:SAS Institute Inc. Available at https://support.sas.com/resources/papers/proceedings13/015-2013.pdf*

Springborn, R. 2012. "Set Yourself Free –Use ODS Report Writing Technology in SAS Enterprise Guide Instead of Dynamic Data Exchange in PC SAS" *Proceedings of the Western Users of SAS Software 2012 Conference. Paper 035-2012. Cary,N.C.:SAS Institute Inc.* Available at http://www.lexjansen.com/wuss/2012/35.pdf .

Springborn, R. 2013. "Set Yourself Free –Use ODS Report Writing Technology in SAS Enterprise Guide Instead of Dynamic Data Exchange in PC SAS Part II SAS Code Revealed" *Proceedings of the SAS Global Forum 2013 Conference. Paper 016-2013. Cary,N.C.:SAS Institute Inc.* Available at http://support.sas.com/resources/papers/proceedings13/016-2013.pdf

Zender, C.L., 2016. “That’s All Right: More Complex Reports” *Proceedings of the SAS Global Forum 2016 Conference. Paper 5762-2016. Cary,N.C.:SAS Institute Inc.* Available at http://support.sas.com/resources/papers/proceedings16/SAS5762-2016.pdf

CONTACT INFORMATION

Robert Richard Springborn, Ph.D.
Office of Statewide Health Planning and Development
Information Services Division, Healthcare Analytics Branch
Clinical Data Unit
2020 West El Camino Avenue
Sacramento, CA 95833
Office (916) 326- 3874

ROBERT.SPRINGBORN@OSHPD.CA.GOV

Appendix I Create Two ICD-10 Macro Lists that Define CABG Surgery

```
1  ********************************************************************************************************************;
2  *** CREATE MACRO LISTS OF CODES AND COMPILE MACROS                                                                  ;
3  *** UPDATED FROM DDR 2016 JANUARY TO JUNE STEP 3 MACRO LIST PROGRAM                                                ;
4  ********************************************************************************************************************;
5  options mcompilenote= all;
6
7  %Let TargetV= folder location;
8  %Let TargetVV= folder location;
9
10 ********************************************************************************;
11 ** MACRO #1 : READ ICD10 PROCEDURE CODES FOR CABG *********************************;
12 title1 "IMPORT ICD10 CABG CODES";
13 proc import datafile="&TargetV.CABGandExclusion2017Updates.xls"
14   out=pxcabg_icd10 dbms=xls replace;  range="CABG only updates$E1:E297"; getnames=yes;
15   run;
16
17 *** USING SQL TO BUILD MACRO LIST  ******************************************;
18 data proc_codesone (drop= quote);
19 length quote $1. px_icd10 $7. px_cabg $11.;
20 set pxcabg_icd10;
21
22 ** Add quote marks;
23 quote="'";
24 call catx(" ",px_cabg,quote,px_icd10,quote);
25 px_cabg= compress(px_cabg);
26 run;
27
28 proc sql noprint;
29 select px_cabg
30    into :cabgicd10 separated by ' '
31       from proc_codesone;
32 %let codecnt=&sqlobs;
33 %put &=cabgicd10;
34
35 ********************************************************************************;
36 ** MACRO #2 : READ ICD10 EXCLUSION CODES FOR CABG          ;
37 title1 "IMPORT ICD10 CABG EXCLUSION CODES";
38 proc import datafile="&TargetVV.CABGExclusionsandCABGValveExclusionsRevised2017.xls"
39   out=pxcabgexcl_icd10 dbms=xls replace;  range="CABG Exclusion$M1:M4646"; getnames=yes;
40   run;
41
42 *** USING SQL TO BUILD MACRO LIST  ******************************************;
43 data proc_codestwo (drop= quote);
44 length quote $1. px_icd10excl $7. px_cabgexcl $11.;
45 set pxcabgexcl_icd10;
46
47 ** Add quote marks;
48 quote="'";
49 call catx(" ",px_cabgexcl,quote,px_icd10excl,quote);
50 px_cabgexcl= compress(px_cabgexcl);
51 run;
52
53 proc sql noprint;
54 select px_cabgexcl
55    into :cabgexclicd10 separated by ' '
56       from proc_codestwo;
57 %let codecntexcl=&sqlobs;
58 %put &=cabgexclicd10;
59
60 ********************************************************************************;
61 ** COMPILE SEVERAL MACROS                                                       ;
62 ********************************************************************************;
63
```

```
64  /* Extract all open heart procedures and MDC 05 from PDD (Beattes Program 4-23-2008) */
65  %MACRO effdoS(cond,incond,maxloop,len,name,des);
66      DO i=1 TO &maxloop;
67        IF SUBSTR(&incond.{i},1,&len) IN (&cond) THEN DO;
68              &name = 1;
69              i = &maxloop;
70              END;
71          ELSE IF SUBSTR(&incond.{i},1,&len) = '   ' THEN DO;
72              i = &maxloop;
73              END;
74        END;
75      LABEL &name="&des";
76  %MEND effdoS;
77
78   /* Identify 361 bypass anastomosis and exclusionary procedures */
79  %MACRO effdo(cond,incond,maxloop,len,name,des);
80      &name = 0;
81      DO i=1 TO &maxloop;
82        IF SUBSTR(&incond.{i},1,&len) IN (&cond) THEN DO;
83              &name = 1;
84              cabgdate = prcdt{i};
85              i = &maxloop;
86              END;
87          ELSE IF SUBSTR(&incond.{i},1,&len) = '   ' THEN DO;
88              i = &maxloop;
89              END;
90        END;
91      LABEL &name="&des";
92  %MEND effdo;
93
94  %MACRO effdodtd(cond,incond,maxloop,len,name,des,prvsday=N);
95      &name = 0;
96      DO i=1 TO &maxloop;
97  %IF &prvsday EQ N %THEN %DO;
98        IF SUBSTR(&incond.{i},1,&len) IN (&cond) AND prcdt{i} EQ cabgdate THEN DO;
99  %END;
100 %ELSE %IF &prvsday EQ ANY %THEN %DO;
101       IF SUBSTR(&incond.{i},1,&len) IN (&cond) AND .< prcdt{i} < cabgdate THEN DO;
102 %END;
103 %ELSE %IF &prvsday EQ TWO %THEN %DO;
104       IF SUBSTR(&incond.{i},1,&len) IN (&cond) AND (prcdt{i} EQ cabgdate OR prcdt{i} EQ (cabgdate -1)
105          OR prcdt{i} EQ (cabgdate-2)) THEN DO;
106 %END;
107 %ELSE %IF &prvsday EQ ONLY %THEN %DO;
108       IF SUBSTR(&incond.{i},1,&len) IN (&cond) AND (/*(prcdt{i} EQ cabgdate) OR */
109         (prcdt{i} EQ (cabgdate-1))) THEN DO;
110 %END;
111 %ELSE %DO;
112       IF SUBSTR(&incond.{i},1,&len) IN (&cond) AND (/*(prcdt{i} EQ cabgdate) OR */
113         (prcdt{i} <= (cabgdate-1))) THEN DO;
114 %END;
115             &name = 1;
116             i = &maxloop;
117             END;
118         ELSE IF SUBSTR(&incond.{i},1,&len) = '   ' THEN DO;
119             i = &maxloop; END;
120       END;
121     LABEL &name="&des";
122 %MEND effdodtd;
123
124 %MACRO effdoatadmt(cond,incond,maxloop,len,name,des);
125     &name = 0;
126     DO i=1 TO &maxloop;
127       IF SUBSTR(&incond.{i},1,&len) IN (&cond) AND cp{i} EQ 'Y' THEN DO;
128             &name = 1;
129             i = &maxloop;
```

```
130                 END;
131               ELSE IF SUBSTR(&incond.{i},1,&len) = '    ' THEN DO;
132                 i = &maxloop; END;
133             END;
134         LABEL &name="&des";
135     %MEND effdoatadmt;
136     ************************************************************************************;
137     title1; title2; title3;
138     QUIT;
```

Appendix 2 Construct the Cohort Study Population and Send Results to MS Excel

```
1   ********************************************************************************************************************;
2   *** PROGRAM FROM ROBERT TALK ICD9 and ICD10 TO VERIFY NEW FORMAT CATALOG 6        ;
3   *** FOR 2015 PDD ***************************************************************************************************;
4   *** PROVIDE TEXT FOR DIAGNOSIS, PROCEDURE AND PROCEDURE DATE ********************;
5   *** GO TO LINE 408-411 TO INSERT SELECTION CRITERIA ****************************;
6   ************************************************************************;
7   options errors=5 pageno=1 ;
8
9   ** Set location for Format Catalogue Data File and Created 27July2015;
10  %Let TargetZ=\\folder location\;
11  LIBNAME forcat6 "&TargetZ.format_catalog6\";
12  options fmtsearch=(forcat6);
13
14  *%let yr= 12;
15  %let period_label= 2015 FULL YEAR;
16
17  *** CREATE GENERAL FORMATS ;
18  proc format;
19    value $sex1fmt
20      '1'='(a)Male'
21    '2'='(b)Female'
22    '3'='(c)Other'
23      '4'='(d)Unknown';
24    value $sexfchr(notsorted)
25      '1'='Male'
26      '2'='Female';
27    value  sexfnum(notsorted)
28      1='Male'
29      2='Female';
30    value fmtbmi
31      0-<18.51= '(a)0-<18.5'
32    18.51-<40.0= '(b)18.5-<40'
33    40.0-high= '(c)40plus';
34    value fmt1bmi
35      0-<18.51='(a)le 18.5'
36    18.51-high='(b)gt 18.5';
37    value fmt2bmi
38      0-<40.0='(a) lt 40.0'
39    40.0-high='(b) ge 40.0';
40    value age1fmt
41      0-<18='(a)under 18'
42    18-<20='(b)18-19'
43    20-<30='(c)20-29'
44    30-<40='(d)30-39'
45    40-<50='(e)40-49'
46    50-<60='(f)50-59'
47    60-<70='(g)60-69'
48    70-<80='(h)70-79'
49    80-<90='(i)80-89'
50    90-<100='(l)90-99'
51    100-125='(k)100-125'
52  9999= "(l) Missing";
53    value volqrp
54      0-<26='(a)0-25'
55  26-<51='(b)26-50'
56  51-<76='(c)51-75'
57    76-<101='(d)75-100'
58    101-<126='(e)101-125'
59    126-<151='(f)126-150'
60    151-<176='(g)151-175'
61    176-<201='(h)176-200'
62    201-<226='(i)201-225'
63    226-<251='(j)226-250'
```

```
64        251-<276='(k)251-275'
65        276-<301='(l)276-300'
66        301-<326='(m)301-325'
67        326-<351='(n)326-350'
68        351-<376='(o)351-375'
69        376-<401='(p)376-400'
70        401-high='(q)401plus';
71      value $mdcfmt
72      '00'= 'UNGROUPABLE'
73      '01'= 'NERVOUS SYSTEM, DISEASES & DISORDERS'
74      '02'='EYE, DISEASES & DISORDERS'
75      '03'='EAR, NOSE, MOUTH, & THROAT, DISEASES & DISORDERS'
76      '04'='RESPIRATORY SYSTEM, DISEASES & DISORDERS'
77      '05'='CIRCULATORY SYSTEM, DISEASES & DISORDERS'
78      '06'='DIGESTIVE SYSTEM, DISEASES & DISORDERS'
79      '07'='HEPATOBILIARY SYSTEM & PANCREAS, DISEASES & DISORDERS'
80      '08'='MUSCULOSKELETAL SYSTEM & CONNECTIVE TISSUE, DISEASES & DISORDERS'
81      '09'='SKIN, SUBCUTANEOUS TISSUE & BREAST, DISEASES & DISORDERS'
82      '10'='ENDOCRINE, NUTRITIONAL, AND METABOLIC, DISEASES & DISORDERS'
83      '11'='KIDNEY AND URINARY TRACT, DISEASES & DISORDERS'
84      '12'='MALE REPRODUCTIVE SYSTEM, DISEASES & DISORDERS'
85      '13'='FEMALE REPRODUCTIVE SYSTEM, DISEASES & DISORDERS'
86      '14'='PREGNANCY, CHILDBIRTH, & THE PUERPERIUM'
87      '15'='NEWBORNS AND NEONATE CONDITIONS BEGAN IN PERINATAL PERIOD'
88      '16'='BLOOD, BLOOD FORMING ORGANS,IMMUNOLOGICAL, DISEASES & DISORDERS'
89      '17'='MYELOPROLIFERATIVE DISEASES & POORLY DIFFERENTIATED NEOPLASMS'
90      '18'='INFECTIOUS & PARASITIC DISEASES'
91      '19'='MENTAL DISEASES & DISORDERS'
92      '20'='ALCOHOLDRUG USE AND ALCOHOLDRUG INDUCED ORGANIC MENTAL DISEASES'
93      '21'='INJURIES, POISONINGS, AND TOXIC EFFECTS OF DRUGS'
94      '22'='BURNS'
95      '23'='FACTORS ON HEALTH STATUS & OTHER CONTACTS WITH HEALTH SERVICES'
96      '24'='MULTIPLE SIGNFICANT TRAUMA'
97      '25'='HUMAN IMMUNODEFICIENCY VIRUS INFECTIONS';
98      value exclfmt
99      0="Isolated CABG (icd361=1,No Exclusions)"
100     1="Non-Isolated CABG (icd361=1,Exclusions)";
101     value isofmt
102     0="Non-Isolated CABG"
103     1="Isolated CABG";
104     value num10fmt
105     0-<99999= "Number Present"
106     99999= "Number Absent ";
107     value $ssnfmt
108     '0','000000001'= "Invalid SSN"
109     OTHER= "Valid SSN";
110     value $textfmt
111     '1'-'9999999999', 'A'-'ZZZZZZZZZZ', 'a'-'zzzzzzzzzz' = "Text Present"
112     OTHER= "Text Absent" ;
113       value yearfmt
114     1900-1925='a. 1900-1925'
115     1926-1950='b. 1926-1950'
116     1951-1975='c. 1951-1975'
117     1976-2000='d. 1976-2000'
118        2001-2012='e. 2001-2012'
119     9999= "f. Missing";
120     value byrfmt
121     1900-1925='a. 1900-1925'
122     1926-1950='b. 1926-1950'
123     1951-1975='c. 1951-1975'
124     1976-1990='d. 1976-1990'
125        1991-1994='e. 1991-1994'
126     1995-2012='f Juvenile'
127     9999= "g. Missing";
128       value yesnofnum (notsorted)
129       1='Yes'
```

```
      0='No'
other= "Missing";
 Run;

*************************************************************************************;
*** IDENTIFY ISOLATED AND NON-ISOLATED CABG AND
*** PRINT SELECTED VARIABLES FROM SAMPLE DATA ;
*************************************************************************************;

*************************************************************************************;
*** PART ONE ::  IDENTIFY HEART RELATED PROCEDURES  **************;

data beate_partone;
  Length HID $6. groupmonth $15. ;
  hid= oshpd_id;

  ** Create arrays to search for PDD elements;
  Array dg {25} $ diag_p odiag1-odiag24;        ** Diagnosis :: Principal and Other Diagnosis FORMAT $5. ;
  Array cp {25}  $ poa_p opoa1-opoa24;          ** Diagnosis Present at Admission :: Principal and Other FORMAT $1;
  Array prc   {21} $ proc_p oproc1-oproc20;     ** Procedure :: Principal and Other FORMAT $4 ;
  Array prcdt {21} proc_pdt procdt1-procdt20;  ** Procedure Date :: Principle and Other FORMAT MMDDYY10. ;
  LENGTH include1 include2 3;

** Springborn 08March2016 Read Sample Data file;
set pddfiles.pdd2015 ;

bthdate_yr= year(bthdate);
admtdate_yr= year(admtdate);
dschdate_yr= year(dschdate);
len_dg= length(diag_p);
len_prc= length(proc_p);
dis_monyr= put(dschdate,mmyyd7.);

  ** SELECT ONLY 18 YEARS OR OLDER AT ADMISSION ***************************************;
** Early Release of 2013 Cycle 1 data did not contain agyradm;
*if year(dschdate)= 2013 and month(dschdate) le 6 then agyradm= (admtdate- bthdate)/364.25;
  If AgYrAdm ge 18 ;

if dschdate le "30SEP2015"D then groupmonth= "ICD9Months";
if dschdate ge "01OCT2015"D then groupmonth= "ICD10Months";

  *** Springborn 08Mar2016 Second block for ICD10 codes;
IF GROUPMONTH= "ICD9Months" THEN DO;

  ** (Beattes Program 4-23-2008) ;
     FORMAT a: b: c: d: e: h: l: m: p: r: s: w: z: ;
     IF mdc EQ '05' THEN OUTPUT;    * Major disease category (MDC) of Circulatory System;
ELSE DO;
include1=0;
 *ARRAY px{21} prproc proc1-proc20;
  %effdoS('361' '374',prc,21,3,include1);
  %effdoS('3510' '3511' '3512' '3514' '3520' '3521' '3522' '3523' '3524' '3527' '3528' '3531' '3532'
     '3533' '3539' '3551' '3553' '3561' '3562' '3571' '3593' '3603' '3610' '3691' '3699' '3732' '3765'
      '3766' '3961',prc,21,4,include1);
        IF include1 THEN OUTPUT;
END;
   END;

*** Springborn 08Mar2016 Second block for ICD10 codes;
ELSE IF GROUPMONTH= "ICD10Months" THEN DO;
FORMAT a: b: c: d: e: h: l: m: p: r: s: w: z: ;
IF mdc EQ '05' THEN OUTPUT;    * Major disease category (MDC) of Circulatory System;
ELSE DO;
include2=0;
 %effdoS(&cabgicd10,prc,21,7,include2,Bypass anastomosis for heart revascularization ICD10);
   IF include2 THEN OUTPUT;
```

```
196   END;
197   END;
198    Run;
199   proc freq data= beate_partone;
200   title1 "ASSESS LENGTH OF DX AND PX";
201   tables dis_monyr*(len_dg len_prc) /norow nocol nopercent;
202   run;
203   title1;
204
205   ********************************************************************************;
206   *** PART TWO :: SELECT ISOLATED CABG ******************************************;
207   data beate_parttwo ;
208
209   ** Create arrays to search for PDD elements;
210    Array dg {25} $ diag_p odiag1-odiag24;     ** Diagnosis :: Principal and Other Diagnosis FORMAT $5. ;
211    Array cp {25}  $ poa_p opoa1-opoa24;      ** Diagnosis Present at Admission :: Principal and Other FORMAT $1;
212    Array prc   {21} $ proc_p oproc1-oproc20;  ** Procedure :: Principal and Other FORMAT $4 ;
213    Array prcdt {21} proc_pdt procdt1-procdt20;  ** Procedure Date :: Principle and Other FORMAT MMDDYY10. ;
214
215    Length I 3. ;
216    LENGTH icd361 3 cabgdate 4 ;
217    LENGTH icd351  icd352  icd353  icd354 icd355 icd356 icd357 icd358 icd359  icd361 icd3731 icd3732
218      icd3735 icd3767 icd375  icd3810 icd3811 icd3812 icd3814 icd3815 icd381x icd3834 icd3841
219      icd3842 icd3844 icd3845 icd3865 icd3885 icd3921 icd3922 icd3923 icd3924 icd3925 icd3926
220      icd3928 icd3951 icd3952 icd3953 icd3954 icd3955 icd3957 icd3958 icd3959
221      icdV433 icd3229 icd324  icd84x  icd8522 icd8523 icd854  3;
222    LENGTH icd335 icd336 3;
223   Length iso_cabg exclusions cabg_excl 3. ;
224
225    Set beate_partone ;
226
227   *** Springborn 08Mar2016 First block for ICD9 codes ;
228   IF groupmonth= "ICD9Months" THEN DO;
229
230    /* Find all records with CABG procedure 36.1x */
231    /* Determine wait days from admit to CABG */
232    %effdo('361',prc,21,3,icd361,Bypass anastomosis for heart revascularization);
233
234    ** Only CABG surgery cases during 2008;
235    cabgdate_yr= year(cabgdate);   * calculated from cabg date in PDD file;
236    admtdate_yr= year(admtdate);   * Admission date from PDD file;
237    dschdate_yr= year(dschdate);   * Discharge date from PDD file;
238
239    if dschdate ne . then datayear= dschdate_yr ;
240    If cabgdate_yr= datayear or admtdate_yr= datayear or dschdate_yr= datayear ;
241
242        %effdodtd('351',prc,21,3,icd351,Open heart valvuloplasty without replacement);
243      %effdodtd('352',prc,21,3,icd352,Replacement of heart valve);
244      %effdodtd('353',prc,21,3,icd353,Operations on structures adjacent to heart valves);
245      %effdodtd('354',prc,21,3,icd354,Production of septal defect in heart);
246      %effdodtd('355',prc,21,3,icd355,Repair of atrial and ventricular septa with prosthesis);
247      %effdodtd('356',prc,21,3,icd356,Repair of atrial and ventricular septa with tissue graft);
248      %effdodtd('357',prc,21,3,icd357,Other and unspecified repair of atrial and ventricular septa);
249      %effdodtd('358',prc,21,3,icd358,Total repair of certain congenital cardiac anomalies);
250      %effdodtd('359',prc,21,3,icd359,Other operations on valves and septa of heart);
251
252      ** Remove Exclusion px 37.31 per Holly 22Mar2013 ;
253      ** Update this step to include px 37.31 and dx 423.2 next run;
254      icd3731= 0;
255      *%effdodtd('3731',prc,21,4,icd3731,%STR(Percardiectomy with DX423.2 constructive pericarditis));
256
257      %effdodtd('3732',prc,21,4,icd3732,Excision of aneurysm of heart);
258      %effdodtd('3735',prc,21,4,icd3735,Partial ventriculectomy);
259      %effdodtd('375',prc,21,3,icd375,Heart transplantation);
260      %effdodtd('3767',prc,21,4,icd3767,Implantation of cardiomyostimulation system);
261      %effdodtd('3810',prc,21,4,icd3810,Endarterectomy NOS);
```

```
262       %effdodtd('3811',prc,21,4,icd3811,Intracranial endarterectomy);
263       %effdodtd('3812',prc,21,4,icd3812,%STR(Head and neck endarterectomy));
264       %effdodtd('3814',prc,21,4,icd3814,Endarterectomy of aorta);
265       %effdodtd('3815',prc,21,4,icd3815,Thoraic endarterectomy);
266       /* lower limb endarterectomy taken out on 2007-09-15: does not make a CABG non-isolated */
267     /* Remove invalid code 3817 08Jan2016 Springborn; */
268       %effdodtd('3816',prc,21,4,icd381x,%STR(Endarterectomy));
269       %effdodtd('3834',prc,21,4,icd3834,%STR(Resection of vessel with replacement: Aniectomy, excision of
270     aneurism, blood vessel with anastomosis));
271       %effdodtd('3841',prc,21,4,icd3841,%STR(Resection of vessel with replacement: Angiectomy, excision of
272     aneurism with replacement, 1=intracranial vessel));
273       %effdodtd('3842',prc,21,4,icd3842,%STR(Resection of vessel with replacement: Angiectomy, excision of
274     aneurism with replacement, 2=other vessels));
275       %effdodtd('3844',prc,21,4,icd3844,%STR(Resection of vessel with replacement: aorta, abdominal));
276       %effdodtd('3845',prc,21,4,icd3845,Resection of thorcic vessel with replacement);
277       %effdodtd('3865',prc,21,4,icd3865,%STR(Other excision of vessels));
278       %effdodtd('3885',prc,21,4,icd3885,%STR(Other surgical occlusion of vessels));
279       %effdodtd('3921',prc,21,4,icd3921,%STR(Caval-pulmonary artery anastomosis));
280       %effdodtd('3922',prc,21,4,icd3922,%STR(Aorta-subclavian-carotid bypass));
281       %effdodtd('3923',prc,21,4,icd3923,%STR(Other intrathoracic vascular shunt or bypass));
282       %effdodtd('3924',prc,21,4,icd3924,%STR(Aorta-renal bypass));
283       %effdodtd('3925',prc,21,4,icd3925,%STR(Aorta-illac-femoral bypass));
284       %effdodtd('3926',prc,21,4,icd3926,%STR(Other intra-abdominal vascular shunt of bypass));
285       %effdodtd('3928',prc,21,4,icd3928,%STR(Extracranial-intracranial (EJ-IC) vascular bypass));
286       %effdodtd('3951',prc,21,4,icd3951,%STR(Clipping of aneurysm));
287       %effdodtd('3952',prc,21,4,icd3952,%STR(Other repair of aneurysm));
288       %effdodtd('3953',prc,21,4,icd3953,%STR(Repair of arteriovenous fistula));
289       %effdodtd('3954',prc,21,4,icd3954,%STR(Re-entry operation aorta: fenestration of dissecting aneurysm of
290     thoracic aorta));
291       %effdodtd('3955',prc,21,4,icd3955,%STR(Reimplantation of aberrant renal vessel));
292       %effdodtd('3957',prc,21,4,icd3957,%STR(Repair of blood vessel with synthetic patch graft));
293       %effdodtd('3958',prc,21,4,icd3958,%STR(Repair of blood vessel with unspecified patch graft));
294       %effdodtd('3959',prc,21,4,icd3959,%STR(Other repair of vessel: aorticopulmonary windows operation,
295     arterioplasty NOS, construction of venous valves, plications of vein, reimplantation of artery));
296
297        icdV433 = 0; /* updated on 02/07/2007 */
298       ***Remove Exclusion dx v43.3 per Joe 12Mar2013 ;
299       *%effdoS('V433',dg,25,4,icdV433,%STR(Organ or tissue replaced by other means: Heart valve));
300
301       %effdodtd('3229',prc,21,4,icd3229,%STR(Other local excision of lesion or tissue of lung));
302       %effdodtd('324', prc,21,3,icd324, %STR(Lobectomy with segmental resection of adjacent lobes of lung));
303       %effdodtd('840' '841',prc,21,3,icd84x,%STR(Amputation));
304       %effdodtd('8522',prc,21,4,icd8522,%STR(Resection of quadrant of breast));
305       %effdodtd('8523',prc,21,4,icd8523,%STR(Subtotal mastectomy          ));
306       %effdodtd('854' ,prc,21,3,icd854 ,%STR(Mastectomy));
307
308       /* Needed for DDR */
309       %effdodtd('335',prc,21,3,icd335,Lung transplantation);
310       %effdodtd('336',prc,21,3,icd336,Combined Heart and Lung transplantation);
311
312       exclusions = (icd351   + icd352   + icd353   + icd354   + icd355   + icd356   + icd357   +
313                  icd358   + icd359   + icd3731  + icd3732  + icd3735  + icd375   + icd3767  +
314                  icd3810  + icd3811  + icd3812  + icd3814  + icd3815  + icd381x  + icd3834  +
315                  icd3841  + icd3842  + icd3844  + icd3845  + icd3865  + icd3885  + icd3921  +
316                  icd3922  + icd3923  + icd3924  + icd3925  + icd3926  + icd3928  + icd3951  +
317                  icd3952  + icd3953  + icd3954  + icd3955  + icd3957  + icd3958  + icd3959  +
318                  icdV433  + icd3229  + icd324   + icd84x   + icd854   + icd8522  + icd8523  +
319                  icd335   + icd336 )>0;
320
321       iso_cabg = ^exclusions AND icd361;
322
323       /* Cardiogenic Shock */
324       %effdoatadmt('78551',dg,25,5,PDDcardshock,%STR(Cardiogenic Shock - PDD));
325
326       /* Salvage acuity */
327       %effdodtd('9960' '9963' ,prc,21,4,PDDacuity,%STR(Acuity Salvage - PDD),prvsday=TWO);
```

```
328
329         DROP i;
330     END;
331
332     *** Springborn 08Mar2016 Second Block for ICD10 Codes;
333     Else if groupmonth= "ICD10Months" THEN DO;
334
335       *** Find all records with CABG procedure 36.1x ;
336       %effdo(&cabgicd10,prc,21,7,icd361,Bypass anastomosis for heart revascularization ICD10);
337
338     ** Only CABG surgery cases ;
339       cabgdate_yr= year(cabgdate);   * calculated from cabg date in PDD file;
340       admtdate_yr= year(admtdate);   * Admission date from PDD file;
341       dschdate_yr= year(dschdate);   * Discharge date from PDD file;
342
343       if dschdate ne . then datayear= dschdate_yr ;
344       If cabgdate_yr= datayear or admtdate_yr= datayear or dschdate_yr= datayear ;
345
346     *** Count Exclusions for Isolated CABG    ;
347       %effdodtd(&cabgexclicd10,prc,21,7,cabg_excl,CABG Exclusions ICD10);
348
349         exclusions = (cabg_excl)>0;
350
351         iso_cabg = ^exclusions AND icd361;
352     END;
353
354      if cabgdate= . then cabgdate= 99999;
355      if cabgdatePDD= . then cabgdatePDD= 99999;
356     *** Springborn 08Mar2016 Set Beate variables to missing;
357     *cabgdatePDD= .;
358     *isocabgPDD= .;
359
360     label include1= "Part One Include1 (ICD-9)"
361     include2= "Part One Include2(ICD-10)"
362     cabgdate= "Part Two cabgdate"
363     cabgdate_yr= "Part Two cabgdate yr"
364     iso_cabg= "Part Two iso_cabg"
365     icd361= "Part Two CABG "
366     exclusions= "Part Two exclusions"
367       cabgdatePDD= "Beate cabg date"
368     icdv433= "Exclusion DX V43.3"
369     icd3731= "Exclusion PX 37.31";
370
371       Run;
372     proc contents data=beate_parttwo fmtlen;
373     run;
374
375     *** SELECT SAMPLE OF ONLY CABG RECORDS ********************************;
376     data cabgdata;
377     set beate  parttwo;
378     if icd361= 1;
379     if dis_monyr in('03-2015','06-2015','09-2015','12-2015');
380     birth= put(bthdate,mmddyy10.);
381     admit= put(admtdate,mmddyy10.);
382     discharge= put(dschdate,mmddyy10.);
383     surgery= put(cabgdate,mmddyy10.);
384     run;
385
386     *** VERIFY SELECTION ISOCABG AND NON-ISOCABG FOR ICD9 AND ICD10 TIME PERIODS  ;
387     title1 "COUNT CABG AND ISOCABG FOR 2015 PDD DATA";
388     /*proc freq data=cabgdata (where= (groupmonth= "ICD9Months"));
389     tables mdc iso_cabg icd361*exclusions cabgdate_yr admtdate_yr dschdate_yr
390     /norow nocol nopercent;
391     format mdc $mdcfmt. cabgdate_yr admtdate_yr yearfmt. iso_cabg isofmt. ;
392     run;
393     title2 "OCTOBER THROUGH DECEMBER ICD10 CODES";
```

```
proc freq data=cabgdata (where= (groupmonth= "ICD10Months"));
tables mdc iso_cabg icd361*exclusions cabgdate_yr admtdate_yr dschdate_yr
/norow nocol nopercent;
format mdc $mdcfmt. cabgdate_yr admtdate_yr yearfmt. iso_cabg isofmt. ;
run; */
title2 "JANUARY THROUGH DECEMBER SUMMARY";
proc freq data=cabgdata ;
tables mdc iso_cabg icd361*exclusions cabgdate_yr admtdate_yr dschdate_yr dis_monyr*(icd361 iso_cabg)
/norow nocol nopercent;
format mdc $mdcfmt. cabgdate_yr admtdate_yr yearfmt. iso_cabg isofmt. ;
run;
title1; title2; title3;

*** ADD TEXT TO DIAGNOSIS AND PROCEDURES TO ICD10 FULL VARIABLES;
data CabgDataAddText;
      length  point $1.  prdx dx1-dx24 $10.
            prdx_txt dx1_txt dx2_txt dx3_txt dx4_txt dx5_txt dx6_txt dx7_txt dx8_txt dx9_txt dx10_txt dx11_txt
            dx12_txt dx13_txt  dx14_txt dx15_txt dx16_txt dx17_txt dx18_txt dx19_txt dx20_txt dx21_txt dx22_txt
            dx23_txt dx24_txt $175.  prpx px1-px20 $8.
            prpx_txt px1_txt px2_txt px3_txt px4_txt px5_txt px6_txt px7_txt px8_txt px9_txt px10_txt px11_txt
            px12_txt px13_txt  px14_txt px15_txt px16_txt px17_txt px18_txt px19_txt px20_txt $175.
            prpxdt pxdt1-pxdt20 $12.;

set cabgdata;
hid= oshpd_id;

*** RECONSTRUCT TEXT FOR ICD9 CODES ONLY **********;
*** Springborn First block for ICD9 codes ;
IF groupmonth= "ICD9Months" THEN DO;

point= ".";
call catx(" ", prdx, substr(diag_p,1,3), point, substr(diag_p,4,2) );
call catx(" ", dx1, substr(odiag1,1,3), point, substr(odiag1,4,2) );
call catx(" ", dx2, substr(odiag2,1,3), point, substr(odiag2,4,2) );
call catx(" ", dx3, substr(odiag3,1,3), point, substr(odiag3,4,2) );
call catx(" ", dx4, substr(odiag4,1,3), point, substr(odiag4,4,2) );
call catx(" ", dx5, substr(odiag5,1,3), point, substr(odiag5,4,2) );
call catx(" ", dx6, substr(odiag6,1,3), point, substr(odiag6,4,2) );
call catx(" ", dx7, substr(odiag7,1,3), point, substr(odiag7,4,2) );
call catx(" ", dx8, substr(odiag8,1,3), point, substr(odiag8,4,2) );
call catx(" ", dx9, substr(odiag9,1,3), point, substr(odiag9,4,2) );
call catx(" ", dx10, substr(odiag10,1,3), point, substr(odiag10,4,2) );
call catx(" ", dx11, substr(odiag11,1,3), point, substr(odiag11,4,2) );
call catx(" ", dx12, substr(odiag12,1,3), point, substr(odiag12,4,2) );
call catx(" ", dx13, substr(odiag13,1,3), point, substr(odiag13,4,2) );
call catx(" ", dx14, substr(odiag14,1,3), point, substr(odiag14,4,2) );
call catx(" ", dx15, substr(odiag15,1,3), point, substr(odiag15,4,2) );
call catx(" ", dx16, substr(odiag16,1,3), point, substr(odiag16,4,2) );
call catx(" ", dx17, substr(odiag17,1,3), point, substr(odiag17,4,2) );
call catx(" ", dx18, substr(odiag18,1,3), point, substr(odiag18,4,2) );
call catx(" ", dx19, substr(odiag19,1,3), point, substr(odiag19,4,2) );
call catx(" ", dx20, substr(odiag20,1,3), point, substr(odiag20,4,2) );
call catx(" ", dx21, substr(odiag21,1,3), point, substr(odiag21,4,2) );
call catx(" ", dx22, substr(odiag22,1,3), point, substr(odiag22,4,2) );
call catx(" ", dx23, substr(odiag23,1,3), point, substr(odiag23,4,2) );
call catx(" ", dx24, substr(odiag24,1,3), point, substr(odiag24,4,2) );

prdx= compress(prdx); dx1= compress(dx1); dx2= compress(dx2); dx3= compress(dx3); dx4= compress(dx4);
dx5= compress(dx5); dx6= compress(dx6); dx7= compress(dx7); dx8= compress(dx8); dx9= compress(dx9);
dx10= compress(dx10); dx11= compress(dx11);  dx12= compress(dx12); dx13= compress(dx13);
dx14= compress(dx14); dx15= compress(dx15); dx16= compress(dx16); dx17= compress(dx17);
dx18= compress(dx18); dx19= compress(dx19); dx20= compress(dx20); dx21= compress(dx21);
dx22= compress(dx22); dx23= compress(dx23);  dx24= compress(dx24);

** Apply format;
call catx(" ", prdx_txt, put(prdx,$dx_icd9f.) );
```

```
460     call catx(" ", dx1_txt, put(dx1,$dx_icd9f.) );
461     call catx(" ", dx2_txt, put(dx2,$dx_icd9f.) );
462     call catx(" ", dx3_txt, put(dx3,$dx_icd9f.) );
463     call catx(" ", dx4_txt, put(dx4,$dx_icd9f.) );
464     call catx(" ", dx5_txt, put(dx5,$dx_icd9f.) );
465     call catx(" ", dx6_txt, put(dx6,$dx_icd9f.) );
466     call catx(" ", dx7_txt, put(dx7,$dx_icd9f.) );
467     call catx(" ", dx8_txt, put(dx8,$dx_icd9f.) );
468     call catx(" ", dx9_txt, put(dx9,$dx_icd9f.) );
469     call catx(" ", dx10_txt, put(dx10,$dx_icd9f.) );
470     call catx(" ", dx11_txt, put(dx11,$dx_icd9f.) );
471     call catx(" ", dx12_txt, put(dx12,$dx_icd9f.) );
472     call catx(" ", dx13_txt, put(dx13,$dx_icd9f.) );
473     call catx(" ", dx14_txt, put(dx14,$dx_icd9f.) );
474     call catx(" ", dx15_txt, put(dx15,$dx_icd9f.) );
475     call catx(" ", dx16_txt, put(dx16,$dx_icd9f.) );
476     call catx(" ", dx17_txt, put(dx17,$dx_icd9f.) );
477     call catx(" ", dx18_txt, put(dx18,$dx_icd9f.) );
478     call catx(" ", dx19_txt, put(dx19,$dx_icd9f.) );
479     call catx(" ", dx20_txt, put(dx20,$dx_icd9f.) );
480     call catx(" ", dx21_txt, put(dx21,$dx_icd9f.) );
481     call catx(" ", dx22_txt, put(dx22,$dx_icd9f.) );
482     call catx(" ", dx23_txt, put(dx23,$dx_icd9f.) );
483     call catx(" ", dx24_txt, put(dx24,$dx_icd9f.) );
484
485     ** APPLY PROCEDURE FORMAT ************************************************************;
486     ** NOTE TO UPDATE VARIABLE NAMES EACH YEAR;
487
488     point= ".";
489     call catx(" ", prpx, substr(proc_p,1,2), point, substr(proc_p,3,2) );
490     call catx(" ", px1, substr(oproc1,1,2), point, substr(oproc1,3,2) );
491     call catx(" ", px2, substr(oproc2,1,2), point, substr(oproc2,3,2) );
492     call catx(" ", px3, substr(oproc3,1,2), point, substr(oproc3,3,2) );
493     call catx(" ", px4, substr(oproc4,1,2), point, substr(oproc4,3,2) );
494     call catx(" ", px5, substr(oproc5,1,2), point, substr(oproc5,3,2) );
495     call catx(" ", px6, substr(oproc6,1,2), point, substr(oproc6,3,2) );
496     call catx(" ", px7, substr(oproc7,1,2), point, substr(oproc7,3,2) );
497     call catx(" ", px8, substr(oproc8,1,2), point, substr(oproc8,3,2) );
498     call catx(" ", px9, substr(oproc9,1,2), point, substr(oproc9,3,2) );
499     call catx(" ", px10, substr(oproc10,1,2), point, substr(oproc10,3,2) );
500     call catx(" ", px11, substr(oproc11,1,2), point, substr(oproc11,3,2) );
501     call catx(" ", px12, substr(oproc12,1,2), point, substr(oproc12,3,2) );
502     call catx(" ", px13, substr(oproc13,1,2), point, substr(oproc13,3,2) );
503     call catx(" ", px14, substr(oproc14,1,2), point, substr(oproc14,3,2) );
504     call catx(" ", px15, substr(oproc15,1,2), point, substr(oproc15,3,2) );
505     call catx(" ", px16, substr(oproc16,1,2), point, substr(oproc16,3,2) );
506     call catx(" ", px17, substr(oproc17,1,2), point, substr(oproc17,3,2) );
507     call catx(" ", px18, substr(oproc18,1,2), point, substr(oproc18,3,2) );
508     call catx(" ", px19, substr(oproc19,1,2), point, substr(oproc19,3,2) );
509     call catx(" ", px20, substr(oproc20,1,2), point, substr(oproc20,3,2) );
510
511     prpx= compress(prpx); px1= compress(px1); px2= compress(px2); px3= compress(px3); px4= compress(px4);
512     px5= compress(px5); px6= compress(px6); px7= compress(px7); px8= compress(px8); px9= compress(px9);
513     px10= compress(px10); px11= compress(px11);  px12= compress(px12); px13= compress(px13);
514     px14= compress(px14); px15= compress(px15); px16= compress(px16); px17= compress(px17);
515     px18= compress(px18); px19= compress(px19); px20= compress(px20);
516
517     ** Apply format;
518     call catx(" ", prpx_txt, put(prpx,$px_icd9f.) );
519     call catx(" ", px1_txt, put(px1,$px_icd9f.) );
520     call catx(" ", px2_txt, put(px2,$px_icd9f.) );
521     call catx(" ", px3_txt, put(px3,$px_icd9f.) );
522     call catx(" ", px4_txt, put(px4,$px_icd9f.) );
523     call catx(" ", px5_txt, put(px5,$px_icd9f.) );
524     call catx(" ", px6_txt, put(px6,$px_icd9f.) );
525     call catx(" ", px7_txt, put(px7,$px_icd9f.) );
```

```
526     call catx(" ", px8_txt, put(px8,$px_icd9f.) );
527     call catx(" ", px9_txt, put(px9,$px_icd9f.) );
528     call catx(" ", px10_txt, put(px10,$px_icd9f.) );
529     call catx(" ", px11_txt, put(px11,$px_icd9f.) );
530     call catx(" ", px12_txt, put(px12,$px_icd9f.) );
531     call catx(" ", px13_txt, put(px13,$px_icd9f.) );
532     call catx(" ", px14_txt, put(px14,$px_icd9f.) );
533     call catx(" ", px15_txt, put(px15,$px_icd9f.) );
534     call catx(" ", px16_txt, put(px16,$px_icd9f.) );
535     call catx(" ", px17_txt, put(px17,$px_icd9f.) );
536     call catx(" ", px18_txt, put(px18,$px_icd9f.) );
537     call catx(" ", px19_txt, put(px19,$px_icd9f.) );
538     call catx(" ", px20_txt, put(px20,$px_icd9f.) );
539
540     ** Generate Procedure Dates;
541     ** NOTE TO UPDATE VARIABLE NAMES EACH YEAR;
542     prpxdt= put(proc_pdt,mmddyy10.); pxdt1= put(procdt1,mmddyy10.); pxdt2= put(procdt2,mmddyy10.);
543     pxdt3= put(procdt3,mmddyy10.); pxdt4= put(procdt4,mmddyy10.);  pxdt5= put(procdt5,mmddyy10.);
544     pxdt6= put(procdt6,mmddyy10.); pxdt7= put(procdt7,mmddyy10.); pxdt8= put(procdt8,mmddyy10.);
545     pxdt9= put(procdt9,mmddyy10.); pxdt10= put(procdt10,mmddyy10.); pxdt11= put(procdt11,mmddyy10.);
546     pxdt12= put(procdt12,mmddyy10.); pxdt13= put(procdt13,mmddyy10.); pxdt14= put(procdt14,mmddyy10.);
547     pxdt15= put(procdt15,mmddyy10.); pxdt16= put(procdt16,mmddyy10.); pxdt17= put(procdt17,mmddyy10.);
548     pxdt18= put(procdt18,mmddyy10.); pxdt19= put(procdt19,mmddyy10.); pxdt20= put(procdt20,mmddyy10.);
549     END;
550
551     *** REC0NSTRUCT TEXT FOR ICD10 CODES ONLY **********;
552     Else If groupmonth= "ICD10Months" THEN DO;
553
554     *** Recreate DIAGNOSIS Code ICD10;
555     point=".";
556     call catx(" ", prdx, substr(diag_p,1,3), point, substr(diag_p,4,4) );
557     call catx(" ", dx1, substr(odiag1,1,3), point, substr(odiag1,4,4) );
558     call catx(" ", dx2, substr(odiag2,1,3), point, substr(odiag2,4,4) );
559     call catx(" ", dx3, substr(odiag3,1,3), point, substr(odiag3,4,4) );
560     call catx(" ", dx4, substr(odiag4,1,3), point, substr(odiag4,4,4) );
561     call catx(" ", dx5, substr(odiag5,1,3), point, substr(odiag5,4,4) );
562     call catx(" ", dx6, substr(odiag6,1,3), point, substr(odiag6,4,4) );
563     call catx(" ", dx7, substr(odiag7,1,3), point, substr(odiag7,4,4) );
564     call catx(" ", dx8, substr(odiag8,1,3), point, substr(odiag8,4,4) );
565     call catx(" ", dx9, substr(odiag9,1,3), point, substr(odiag9,4,4) );
566     call catx(" ", dx10, substr(odiag10,1,3), point, substr(odiag10,4,4) );
567     call catx(" ", dx11, substr(odiag11,1,3), point, substr(odiag11,4,4) );
568     call catx(" ", dx12, substr(odiag12,1,3), point, substr(odiag12,4,4) );
569     call catx(" ", dx13, substr(odiag13,1,3), point, substr(odiag13,4,4) );
570     call catx(" ", dx14, substr(odiag14,1,3), point, substr(odiag14,4,4) );
571     call catx(" ", dx15, substr(odiag15,1,3), point, substr(odiag15,4,4) );
572     call catx(" ", dx16, substr(odiag16,1,3), point, substr(odiag16,4,4) );
573     call catx(" ", dx17, substr(odiag17,1,3), point, substr(odiag17,4,4) );
574     call catx(" ", dx18, substr(odiag18,1,3), point, substr(odiag18,4,4) );
575     call catx(" ", dx19, substr(odiag19,1,3), point, substr(odiag19,4,4) );
576     call catx(" ", dx20, substr(odiag20,1,3), point, substr(odiag20,4,4) );
577     call catx(" ", dx21, substr(odiag21,1,3), point, substr(odiag21,4,4) );
578     call catx(" ", dx22, substr(odiag22,1,3), point, substr(odiag22,4,4) );
579     call catx(" ", dx23, substr(odiag23,1,3), point, substr(odiag23,4,4) );
580     call catx(" ", dx24, substr(odiag24,1,3), point, substr(odiag24,4,4) );
581
582     prdx= compress(prdx); dx1= compress(dx1); dx2= compress(dx2); dx3= compress(dx3); dx4= compress(dx4);
583     dx5= compress(dx5); dx6= compress(dx6); dx7= compress(dx7); dx8= compress(dx8); dx9= compress(dx9);
584     dx10= compress(dx10); dx11= compress(dx11);  dx12= compress(dx12); dx13= compress(dx13);
585     dx14= compress(dx14); dx15= compress(dx15); dx16= compress(dx16); dx17= compress(dx17);
586     dx18= compress(dx18); dx19= compress(dx19); dx20= compress(dx20); dx21= compress(dx21);
587     dx22= compress(dx22); dx23= compress(dx23);  dx24= compress(dx24);
588
589     ** Apply Format To Diagnosis;
590     call catx(" ", prdx_txt, put(prdx,$dx_icd10f.) );
591     call catx(" ", dx1_txt, put(dx1,$dx_icd10f.) );
```

```
592     call catx(" ", dx2_txt, put(dx2,$dx_icd10f.) );
593     call catx(" ", dx3_txt, put(dx3,$dx_icd10f.) );
594     call catx(" ", dx4_txt, put(dx4,$dx_icd10f.) );
595     call catx(" ", dx5_txt, put(dx5,$dx_icd10f.) );
596     call catx(" ", dx6_txt, put(dx6,$dx_icd10f.) );
597     call catx(" ", dx7_txt, put(dx7,$dx_icd10f.) );
598     call catx(" ", dx8_txt, put(dx8,$dx_icd10f.) );
599     call catx(" ", dx9_txt, put(dx9,$dx_icd10f.) );
600     call catx(" ", dx10_txt, put(dx10,$dx_icd10f.) );
601     call catx(" ", dx11_txt, put(dx11,$dx_icd10f.) );
602     call catx(" ", dx12_txt, put(dx12,$dx_icd10f.) );
603     call catx(" ", dx13_txt, put(dx13,$dx_icd10f.) );
604     call catx(" ", dx14_txt, put(dx14,$dx_icd10f.) );
605     call catx(" ", dx15_txt, put(dx15,$dx_icd10f.) );
606     call catx(" ", dx16_txt, put(dx16,$dx_icd10f.) );
607     call catx(" ", dx17_txt, put(dx17,$dx_icd10f.) );
608     call catx(" ", dx18_txt, put(dx18,$dx_icd10f.) );
609     call catx(" ", dx19_txt, put(dx19,$dx_icd10f.) );
610     call catx(" ", dx20_txt, put(dx20,$dx_icd10f.) );
611     call catx(" ", dx21_txt, put(dx21,$dx_icd10f.) );
612     call catx(" ", dx22_txt, put(dx22,$dx_icd10f.) );
613     call catx(" ", dx23_txt, put(dx23,$dx_icd10f.) );
614     call catx(" ", dx24_txt, put(dx24,$dx_icd10f.) );
615
616     prpx= proc_p;
617     px1= oproc1;
618     px2= oproc2;
619     px3= oproc3;
620     px4= oproc4;
621     px5= oproc5;
622     px6= oproc6;
623     px7= oproc7;
624     px8= oproc8;
625     px9= oproc9;
626     px10= oproc10;
627     px11= oproc11;
628     px12= oproc12;
629     px13= oproc13;
630     px14= oproc14;
631     px15= oproc15;
632     px16= oproc16;
633     px17= oproc17;
634     px18= oproc18;
635     px19= oproc19;
636     px20= oproc20;
637
638     prpx= compress(prpx); px1= compress(px1); px2= compress(px2); px3= compress(px3); px4= compress(px4);
639     px5= compress(px5); px6= compress(px6); px7= compress(px7); px8= compress(px8); px9= compress(px9);
640     px10= compress(px10); px11= compress(px11); px12= compress(px12); px13= compress(px13);
641     px14= compress(px14); px15= compress(px15); px16= compress(px16); px17= compress(px17);
642     px18= compress(px18); px19= compress(px19); px20= compress(px20);
643
644     ** Apply Format To Procedure;
645     call catx(" ", prpx_txt, put(prpx,$px_icd10f.) );
646     call catx(" ", px1_txt, put(px1,$px_icd10f.) );
647     call catx(" ", px2_txt, put(px2,$px_icd10f.) );
648     call catx(" ", px3_txt, put(px3,$px_icd10f.) );
649     call catx(" ", px4_txt, put(px4,$px_icd10f.) );
650     call catx(" ", px5_txt, put(px5,$px_icd10f.) );
651     call catx(" ", px6_txt, put(px6,$px_icd10f.) );
652     call catx(" ", px7_txt, put(px7,$px_icd10f.) );
653     call catx(" ", px8_txt, put(px8,$px_icd10f.) );
654     call catx(" ", px9_txt, put(px9,$px_icd10f.) );
655     call catx(" ", px10_txt, put(px10,$px_icd10f.) );
656     call catx(" ", px11_txt, put(px11,$px_icd10f.) );
657     call catx(" ", px12_txt, put(px12,$px_icd10f.) );
```

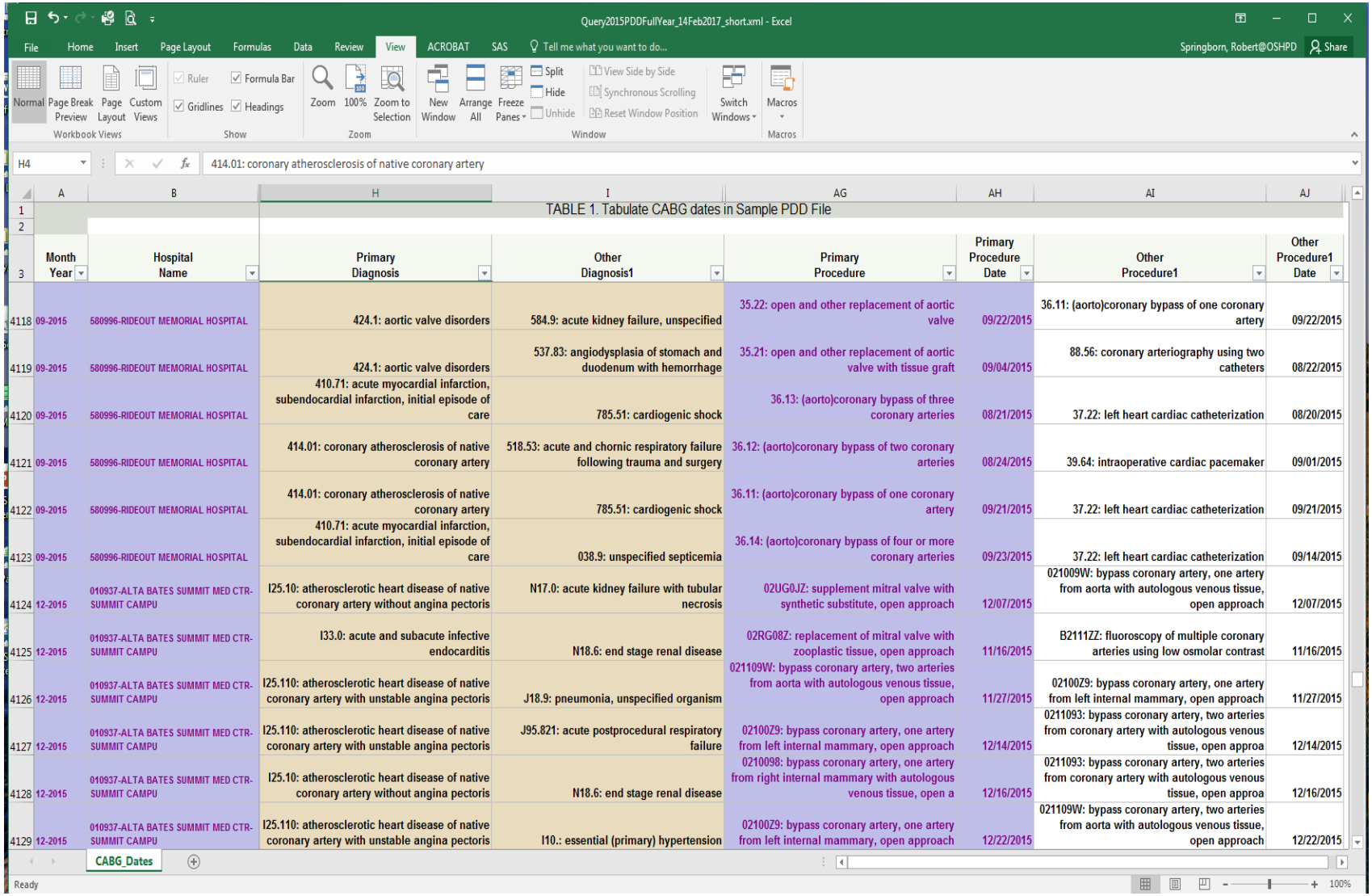
```
658   call catx(" ", px13_txt, put(px13,$px_icd10f.) );
659   call catx(" ", px14_txt, put(px14,$px_icd10f.) );
660   call catx(" ", px15_txt, put(px15,$px_icd10f.) );
661   call catx(" ", px16_txt, put(px16,$px_icd10f.) );
662   call catx(" ", px17_txt, put(px17,$px_icd10f.) );
663   call catx(" ", px18_txt, put(px18,$px_icd10f.) );
664   call catx(" ", px19_txt, put(px19,$px_icd10f.) );
665   call catx(" ", px20_txt, put(px20,$px_icd10f.) );
666
667   ** Generate Procedure Dates;
668   prpxdt= put(proc_pdt,mmddyy10.); pxdt1= put(procdt1,mmddyy10.); pxdt2= put(procdt2,mmddyy10.);
669   pxdt3= put(procdt3,mmddyy10.); pxdt4= put(procdt4,mmddyy10.); pxdt5= put(procdt5,mmddyy10.);
670   pxdt6= put(procdt6,mmddyy10.); pxdt7= put(procdt7,mmddyy10.); pxdt8= put(procdt8,mmddyy10.);
671   pxdt9= put(procdt9,mmddyy10.); pxdt10= put(procdt10,mmddyy10.); pxdt11= put(procdt11,mmddyy10.);
672   pxdt12= put(procdt12,mmddyy10.); pxdt13= put(procdt13,mmddyy10.); pxdt14= put(procdt14,mmddyy10.);
673   pxdt15= put(procdt15,mmddyy10.); pxdt16= put(procdt16,mmddyy10.); pxdt17= put(procdt17,mmddyy10.);
674   pxdt18= put(procdt18,mmddyy10.); pxdt19= put(procdt19,mmddyy10.); pxdt20= put(procdt20,mmddyy10.);
675
676   END;
677   run;
678   title1; title2; title3;
679
680   ** Add hospital name to combined string of hospital id value;
681   ** Springborn ---updated hospital list June 2015;
682   proc sort data=CabgDataAddText;
683   by hid;
684   run;
685   data CabgData_Hosp ;
686     length hospitalfull $45. surgeondetails $85 ;
687     merge CabgDataAddText  (in=x) hsplst.hosplistJune2015;
688     by hid;
689      if x;
690       surglname= Propcase(surglname);
691   surgmi= Propcase(surgmi);
692   surgfname= Propcase(surgfname);
693   call catx(",",surgeondetails,surglname,surgfname,surgmi,surglicnum);
694
695     hospitalfull=hid||'-'||hname;
696     attrib hospitalfull label="Hospital ID and Name";
697    run;
698
699   proc sort data=CabgData_Hosp ;
700   by dis_monyr hospitalfull iso_cabg ssn admtdate;
701   run;
702
703   ** OUTPUT DATA VARIABLES TO MSEXCEL TO LOOK FOR MULTIPLE CABGDATES;
704   ods _all_ close;
705
706   %Let TargetW=\\folder location\;
707
708   ** Use ExcelXP and add doc='help' for list of additional options;
709   ods tagsets.excelxp
710   file="&TargetW.Program\Ver1_07April2017\Query2015PDDFullYear_14Feb2017.xml"
711   style=statistical
712         options(absolute_column_width= '10,25,15,10,10,10,10,30,30,30,30,30,30,30,30,30,30 ,30,30,30,30,
713         30,30,30,30,30,30,30,30,30,30,30,30,10,30,10,30,10,30,10,30,10,30,10,30,10,30,10,30,10,30,10,30,10,3
714         0,10,30, 10,30,10,30,10,30,10,30,10,30,10,30,10,30,10'
715   autofit_height='Yes' autofilter= 'Yes' frozen_headers= '1'
716   sheet_name='CABG_Dates' embedded_titles='yes' suppress_bylines='yes');
717
718   title1 height=3 font=arial 'TABLE 1. Tabulate CABG dates in Sample PDD File';
719
720   proc report data=CabgData_Hosp nowd center headline headskip split='/'
721
722     style(report)=[cellspacing=1 borderwidth=1 ]
723     style(header)=[font_size=2 just=center font_weight=bold]
```

```
724       style(column)=[font_face=arial font_size=1 just=right]
725       style(lines)=[font_style=italic font_size=1 ]
726       style(summary)=[font_face=helvetica font_weight=bold font_size=2 just=right];
727
728     column dis_monyr hospitalfull iso_cabg bthdate admtdate dschdate cabgdate
729     prdx_txt dx1_txt dx2_txt dx3_txt dx4_txt dx5_txt dx6_txt dx7_txt dx8_txt dx9_txt dx10_txt dx11_txt dx12_txt
730     dx13_txt dx14_txt dx15_txt dx16_txt dx17_txt dx18_txt dx19_txt dx20_txt dx21_txt dx22_txt dx23_txt dx24_txt
731     prpx_txt prpxdt px1_txt pxdt1 px2_txt pxdt2 px3_txt pxdt3 px4_txt pxdt4 px5_txt pxdt5 px6_txt pxdt6 px7_txt pxdt7
732     px8_txt pxdt8 px9_txt pxdt9 px10_txt pxdt10 px11_txt pxdt11 px12_txt pxdt12 px13_txt pxdt13 px14_txt pxdt14
733     px15_txt pxdt15 px16_txt pxdt16 px17_txt pxdt17 px18_txt pxdt18 px19_txt pxdt19 px20_txt pxdt20;
734
735     define dis_monyr      / display "Month/Year"
736        style(column)=[font_weight=bold just=left font_size=1 background= very light moderate blue foreground= purple ];
737     define hospitalfull   / display "Hospital/Name"
738        style(column)=[font_weight=bold just=left font_size=1 background= very light moderate blue foreground= purple ];
739     define Iso_cabg   / display "Type of CABG" format= isofmt.
740        style(column)=[font_weight=bold just=right font_size=2 background= light yellow];
741     *define ssn   / display "SSN/Number"
742        style(column)=[font_weight=bold just=right font_size=2 background= light yellow];
743     define bthdate     / display  "Birth/Date"  format=mmddyy10.
744        style(column)=[font_weight=bold just=right font_size=2 background= light yellow];
745     define admtdate    / display "Admit/Date"     format=mmddyy10.
746        style(column)=[font_weight=bold just=right font_size=2 background= light yellow];
747     define cabgdate    / display  "CABG/Date"    format=mmddyy10.
748        style(column)=[font_weight=bold just=right font_size=2 background= light yellow];
749     define dschdate    / display  "Discharge/Date"   format=mmddyy10.
750        style(column)=[font_weight=bold just=right font_size=2 background= light yellow];
751
752     define prdx_txt    / display "Primary/Diagnosis"
753        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
754     define dx1_txt    / display  "Other/Diagnosis1"
755        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
756     define dx2_txt    / display  "Other/Diagnosis2"
757        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
758     define dx3_txt    / display  "Other/Diagnosis3"
759        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
760     define dx4_txt    / display  "Other/Diagnosis4"
761        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
762     define dx5_txt    / display  "Other/Diagnosis5"
763        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
764     define dx6_txt    / display  "Other/Diagnosis6"
765        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
766     define dx7_txt    / display  "Other/Diagnosis7"
767        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
768     define dx8_txt    / display  "Other/Diagnosis8"
769        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
770     define dx9_txt    / display  "Other/Diagnosis9"
771        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange];
772     define dx10_txt    / display  "Other/Diagnosis10"
773        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
774     define dx11_txt    / display  "Other/Diagnosis11"
775        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
776     define dx12_txt    / display  "Other/Diagnosis12"
777        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
778     define dx13_txt    / display  "Other/Diagnosis13"
779        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
780     define dx14_txt    / display  "Other/Diagnosis14"
781        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
782     define dx15_txt    / display  "Other/Diagnosis15"
783        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
784     define dx16_txt    / display  "Other/Diagnosis16"
785        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
786     define dx17_txt    / display  "Other/Diagnosis17"
787        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
788     define dx18_txt    / display  "Other/Diagnosis18"
789        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
```

```
790     define dx19_txt    / display  "Other/Diagnosis19"
791        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
792     define dx20_txt    / display  "Other/Diagnosis20"
793        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
794     define dx21_txt    / display  "Other/Diagnosis21"
795        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
796     define dx22_txt    / display  "Other/Diagnosis22"
797        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
798     define dx23_txt    / display  "Other/Diagnosis23"
799        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
800     define dx24_txt    / display  "Other/Diagnosis24"
801        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate orange ];
802
803     define prpx_txt    / display "Primary/Procedure"
804        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
805     define prpxdt    / display "Primary/Procedure/Date"
806        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
807     define px1  txt    / display  "Other/Procedure1"
808        style(column)=[font_weight=bold just=right font_size=2 ];
809     define pxdt1    / display  "Other/Procedure1/Date"
810        style(column)=[font_weight=bold just=right font_size=2 ];
811     define px2_txt    / display  "Other/Procedure2"
812        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
813     define pxdt2    / display  "Other/Procedure2/Date"
814        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
815     define px3_txt    / display  "Other/Procedure3"
816        style(column)=[font_weight=bold just=right font_size=2 ];
817     define pxdt3    / display  "Other/Procedure3/Date"
818        style(column)=[font_weight=bold just=right font_size=2 ];
819     define px4  txt    / display  "Other/Procedure4"
820        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
821     define pxdt4    / display  "Other/Procedure4/Date"
822        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
823     define px5_txt    / display  "Other/Procedure5"
824        style(column)=[font_weight=bold just=right font_size=2 ];
825     define pxdt5    / display  "Other/Procedure5/Date"
826        style(column)=[font_weight=bold just=right font_size=2 ];
827     define px6  txt    / display  "Other/Procedure6"
828        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
829     define pxdt6    / display  "Other/Procedure6/Date"
830        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
831     define px7_txt    / display  "Other/Procedure7"
832        style(column)=[font_weight=bold just=right font_size=2 ];
833     define pxdt7    / display  "Other/Procedure7/Date"
834        style(column)=[font_weight=bold just=right font_size=2 ];
835     define px8_txt    / display  "Other/Procedure8"
836        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
837     define pxdt8    / display  "Other/Procedure8/Date"
838        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
839     define px9  txt    / display  "Other/Procedure9"
840        style(column)=[font_weight=bold just=right font_size=2 ];
841     define pxdt9    / display  "Other/Procedure9/Date"
842        style(column)=[font_weight=bold just=right font_size=2 ];
843     define px10_txt    / display  "Other/Procedure10"
844        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
845     define pxdt10    / display  "Other/Procedure10/Date"
846        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
847     define px11  txt    / display  "Other/Procedure11"
848        style(column)=[font_weight=bold just=right font_size=2 ];
849     define pxdt11    / display  "Other/Procedure11/Date"
850        style(column)=[font_weight=bold just=right font_size=2 ];
851     define px12_txt    / display  "Other/Procedure12"
852        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
853     define pxdt12    / display  "Other/Procedure12/Date"
854        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
855     define px13_txt    / display  "Other/Procedure13"
```

```
856        style(column)=[font_weight=bold just=right font_size=2 ];
857     define pxdt13    / display  "Other/Procedure13/Date"
858        style(column)=[font_weight=bold just=right font_size=2 ];
859     define px14_txt    / display  "Other/Procedure14"
860        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
861     define pxdt14    / display  "Other/Procedure14/Date"
862        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
863     define px15  txt    / display  "Other/Procedure15"
864        style(column)=[font_weight=bold just=right font_size=2 ];
865     define pxdt15    / display  "Other/Procedure15/Date"
866        style(column)=[font_weight=bold just=right font_size=2 ];
867     define px16_txt    / display  "Other/Procedure16"
868        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
869     define pxdt16    / display  "Other/Procedure16/Date"
870        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
871     define px17_txt    / display  "Other/Procedure17"
872        style(column)=[font_weight=bold just=right font_size=2 ];
873     define pxdt17    / display  "Other/Procedure17/Date"
874        style(column)=[font_weight=bold just=right font_size=2 ];
875     define px18_txt    / display  "Other/Procedure18"
876        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
877     define pxdt18    / display  "Other/Procedure18/Date"
878        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
879     define px19_txt    / display  "Other/Procedure19"
880        style(column)=[font_weight=bold just=right font_size=2 ];
881     define pxdt19    / display  "Other/Procedure19/Date"
882        style(column)=[font_weight=bold just=right font_size=2 ];
883     define px20  txt    / display  "Other/Procedure20"
884        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
885     define pxdt20    / display  "Other/Procedure20/Date"
886        style(column)=[font_weight=bold just=right font_size=2 background= very light moderate blue foreground= purple ];
887
888      run;
889
890     ods tagsets.excelxp close;
891     ods listing;
892
893     title1; title2; title3; footnote1; footnote2; footnote3;
894
895     QUIT;
```

Appendix 3. MS Excel Output for Administrative Data Research Example



Query2015PDDFullYear_14Feb2017_short.xml - Excel

H4: 414.01: coronary atherosclerosis of native coronary artery

TABLE 1. Tabulate CABG dates in Sample PDD File

	Month Year	Hospital Name	Primary Diagnosis	Other Diagnosis1	Primary Procedure	Primary Procedure Date	Other Procedure1	Other Procedure1 Date
4118	09-2015	580996-RIDEOUT MEMORIAL HOSPITAL	424.1: aortic valve disorders	584.9: acute kidney failure, unspecified	35.22: open and other replacement of aortic valve	09/22/2015	36.11: (aorto)coronary bypass of one coronary artery	09/22/2015
4119	09-2015	580996-RIDEOUT MEMORIAL HOSPITAL	424.1: aortic valve disorders	537.83: angiodysplasia of stomach and duodenum with hemorrhage	35.21: open and other replacement of aortic valve with tissue graft	09/04/2015	88.56: coronary arteriography using two catheters	08/22/2015
4120	09-2015	580996-RIDEOUT MEMORIAL HOSPITAL	410.71: acute myocardial infarction, subendocardial infarction, initial episode of care	785.51: cardiogenic shock	36.13: (aorto)coronary bypass of three coronary arteries	08/21/2015	37.22: left heart cardiac catheterization	08/20/2015
4121	09-2015	580996-RIDEOUT MEMORIAL HOSPITAL	414.01: coronary atherosclerosis of native coronary artery	518.53: acute and chornic respiratory failure following trauma and surgery	36.12: (aorto)coronary bypass of two coronary arteries	08/24/2015	39.64: intraoperative cardiac pacemaker	09/01/2015
4122	09-2015	580996-RIDEOUT MEMORIAL HOSPITAL	414.01: coronary atherosclerosis of native coronary artery	785.51: cardiogenic shock	36.11: (aorto)coronary bypass of one coronary artery	09/21/2015	37.22: left heart cardiac catheterization	09/21/2015
4123	09-2015	580996-RIDEOUT MEMORIAL HOSPITAL	410.71: acute myocardial infarction, subendocardial infarction, initial episode of care	038.9: unspecified septicemia	36.14: (aorto)coronary bypass of four or more coronary arteries	09/23/2015	37.22: left heart cardiac catheterization	09/14/2015
4124	12-2015	010937-ALTA BATES SUMMIT MED CTR-SUMMIT CAMPU	I25.10: atherosclerotic heart disease of native coronary artery without angina pectoris	N17.0: acute kidney failure with tubular necrosis	02UG0JZ: supplement mitral valve with synthetic substitute, open approach	12/07/2015	021009W: bypass coronary artery, one artery from aorta with autologous venous tissue, open approach	12/07/2015
4125	12-2015	010937-ALTA BATES SUMMIT MED CTR-SUMMIT CAMPU	I33.0: acute and subacute infective endocarditis	N18.6: end stage renal disease	02RG08Z: replacement of mitral valve with zooplastic tissue, open approach	11/16/2015	B2111ZZ: fluoroscopy of multiple coronary arteries using low osmolar contrast	11/16/2015
4126	12-2015	010937-ALTA BATES SUMMIT MED CTR-SUMMIT CAMPU	I25.110: atherosclerotic heart disease of native coronary artery with unstable angina pectoris	J18.9: pneumonia, unspecified organism	021109W: bypass coronary artery, two arteries from aorta with autologous venous tissue, open approach	11/27/2015	02100Z9: bypass coronary artery, one artery from left internal mammary, open approach	11/27/2015
4127	12-2015	010937-ALTA BATES SUMMIT MED CTR-SUMMIT CAMPU	I25.110: atherosclerotic heart disease of native coronary artery with unstable angina pectoris	J95.821: acute postprocedural respiratory failure	02100Z9: bypass coronary artery, one artery from left internal mammary, open approach	12/14/2015	0211093: bypass coronary artery, two arteries from coronary artery with autologous venous tissue, open approa	12/14/2015
4128	12-2015	010937-ALTA BATES SUMMIT MED CTR-SUMMIT CAMPU	I25.10: atherosclerotic heart disease of native coronary artery without angina pectoris	N18.6: end stage renal disease	0210098: bypass coronary artery, one artery from right internal mammary with autologous venous tissue, open a	12/16/2015	0211093: bypass coronary artery, two arteries from coronary artery with autologous venous tissue, open approa	12/16/2015
4129	12-2015	010937-ALTA BATES SUMMIT MED CTR-SUMMIT CAMPU	I25.110: atherosclerotic heart disease of native coronary artery with unstable angina pectoris	I10.: essential (primary) hypertension	02100Z9: bypass coronary artery, one artery from left internal mammary, open approach	12/22/2015	021109W: bypass coronary artery, two arteries from aorta with autologous venous tissue, open approach	12/22/2015

Appendix 4. Sample Data Discrepancy Report Showing Help Sheet

2016 FIRST HALF CCORP DATA DISCREPANCY REPORT
(190796) RONALD_REAGAN_UCLA_MEDICAL_CENTER

GENERAL INFORMATION	INFORMATION ON DATA ELEMENT DEFINITIONS IS AVAILABLE AT http://www.oshpd.ca.gov/hid/corc/ManualsGuides.html
CABG-REPORTING	Verify ALL CABG surgeries are reported in your CCORP database. Non-CABGs should be deleted from CCORP database. Missing CABGs should be added to CCORP database.
ISOLATED CABG REPORTING	Verify Type of CABG cases are accurately coded in your CCORP database.
CABG+VALVE REPORTING	Verify CABG+Valve cases are accurately coded in your CCORP database. Revise coding in CCORP database if necessary.
RESUSCITATION	Review Resuscitation cases in your CCORP database. If Resuscitation cannot be verified, revise coding in CCORP database.
DISCHARGE STATUS	Verify deaths at the time of discharge reported in your CCORP database.
NEURO-STROKE PERMANENT	Review Post-Operative Stroke cases NOT reported in your CCORP database. Revise coding in CCORP database if necessary.
RENAL-DIALYSIS REQUIREMENT	Review Dialysis Requirement cases NOT reported in your CCORP database. Revise coding in CCORP database if necessary.
PULM-VENTILATION PROLONGED	Review Prolonged Ventilation cases NOT reported in your CCORP database. Revise coding in CCORP database if necessary.
REOPERATION FOR BLEED	Review Reoperation for Bleed cases NOT reported in your CCORP database. Revise coding in CCORP database if necessary.
DEEP STERNAL INFECTION	Review Deep Sternal Infection cases NOT reported in your CCORP database. Revise coding in CCORP database if necessary.
REINTERVENTION GRAFT OCCLUSION	Review Graft Occlusion cases NOT reported in your CCORP database. Revise coding in CCORP database if necessary.
Contact for questions:	Email as ccorp@oshpd.ca.gov Phone number as 916-326-3865

Page 1 of 87
(190796) RONALD_REAGAN_UCLA_MEDICAL_CENTER

Appendix 5. Sample Data Discrepancy Report Showing Summary Table

2016 FIRST HALF CCORP DATA DISCREPANCY REPORT
CABG-REPORTING

VERIFY ALL CABG SURGERIES HAVE BEEN REPORTED IN YOUR CCORP DATABASE.	
Missing CABG cases.	3
NON-CABG cases.	21

Appendix 6. Sample Data Discrepancy Report Showing Patient Record With ICD-10 Codes

2016 FIRST HALF CCORP DATA DISCREPANCY REPORT
CABG-REPORTING

CASE # 1 SHOULD THIS CASE BE ADDED TO CCORP?			
Patient MRN		**Surgery Date**	02/02/1995
Patient SSN	123456789	**Discharge Date**	03/03/2000
Patient Sex	Male	**Death Date**	.
Birth Date	01/01/1990	**Surgeon**	

DETAILED INFORMATION FOR CASE # 1		
DIAGNOSIS	**PROCEDURE(PX)**	**PX DATE**
I25.10: atherosclerotic heart disease of native coronary artery without angina pectoris	021109W: bypass coronary artery, two sites to aorta with autologous venous tissue, open approach	03/29/2016
I21.4: non-st elevation (nstemi) myocardial infarction	02RG07Z: replacement of mitral valve with autologous tissue substitute, open approach	03/29/2016
J95.822: acute and chronic postprocedural respiratory failure	0PC00ZZ: extirpation of matter from sternum, open approach	03/29/2016
I34.2: nonrheumatic mitral (valve) stenosis	5A1221Z: performance of cardiac output, continuous	03/29/2016
E78.5: hyperlipidemia, unspecified	3E0F3SD: introduction of nitric oxide gas into respiratory tract, percutaneous approach	03/30/2016
I48.91: unspecified atrial fibrillation	5A1945Z: respiratory ventilation, 24-96 consecutive hours	03/29/2016
Z79.01: long term (current) use of anticoagulants	0BH17EZ: insertion of endotracheal airway into trachea, via natural or artificial opening	03/29/2016
Z86.73: personal history of transient ischemic attack (tia), and cerebral infarction without residual deficits		.
E11.65: type 2 diabetes mellitus with hyperglycemia		.
Z87.891: personal history of nicotine dependence		.
Z95.2: presence of prosthetic heart valve		.
Z78.1: physical restraint status		.

Page 3 of 87
(190796) RONALD_REAGAN_UCLA_MEDICAL_CENTER

Appendix 7. Sample Data Discrepancy Report Showing Patient Record With ICD-9 Codes

2015 FIRST HALF CCORP DATA DISCREPANCY REPORT
CABG-REPORTING

CASE # 11 SHOULD THIS CASE BE REMOVED FROM CCORP?			
Patient MRN	1234567	**Surgery Date**	02/02/1995
Patient SSN		**Discharge Date**	03/03/2000
Patient Sex	Male	**Death Date**	.
Birth Date	01/01/1990	**Surgeon**	A63040 M . Kwon

DETAILED INFORMATION FOR CASE # 11		
DIAGNOSIS	**PROCEDURE(PX)**	**PX DATE**
441.3: abdominal aneurysm, ruptured	38.45: resection of other thoracic vessels with replacement	06/16/2015
518.81: acute respiratory failure	38.44: resection of abdominal aorta with replacement	06/16/2015
785.51: cardiogenic shock	39.61: extracorporeal circulation auxiliary to open heart surgery	06/16/2015
286.3: congenital deficiency of other clotting factors	96.72: continuous invasive mechanical ventilation for 96 consecutive hours or more	06/16/2015
285.1: acute posthemorrhagic anemia	00.12: administration of inhaled nitric oxide	06/16/2015
723.0: spinal stenosis in cervical region	96.04: insertion of endotracheal tube	06/16/2015
185.: malignant neoplasm of prostate	.	.
441.01: dissecting aortic aneurysm (any part), thoracic	.	.
362.50: macular degeneration (senile) of retina, unspecified	.	.
241.0: nontoxic uninodular goiter	.	.
V12.71: personal history of peptic ulcer disease	.	.
724.02: spinal stenosis, lumbar region, without neurogenic claudication	.	.
564.00: unspecified constipation	.	.
401.9: unspecified essential hypertension	.	.
414.01: coronary atherosclerosis of native coronary artery	.	.