CCW Review comments

Changes introduced on December 2020 by Patricia (12/8/20)

* Incorporated the continuous enrollment macro so that it is no longer a separate piece. There are now macro variables to indicate whether you want the package to create continuous enrollment files or if they already exist. If the package is to create the files, it takes as input the shape of the input files and the input file names.
  + Edit documentation in User Guide (Step 4) and the Preparation Checklist (Steps 17-18) to account for this change. . Laura, it would be great if you could review the parts of the documentation that I changed in reference to the continuous enrollment program to make sure that I’m integrating it smoothly.
* Remove the merge between the condition flags and the enrollment files so that it now keeps everyone in either – full outer merge
* To the guides, I added more background information on the chronic conditions and clarified whether or not to include procedure codes

Comments

1. Program identify\_conditions.sas
   * Format for the “allcodes” variable, making it $12000
     + Suggestion: make the variable holding all codes into a macro, and eliminate the length restriction. Or give an error if the string becomes too long.
   * Added line to keep variables in the file &condname.1:

keep &id. first\_&condname.dx %do yr=&minyear. %to &maxyear.; &condname.\_&yr.: %end;;

* + Change the name of a macro holding part of a file name (enr\_prefix instead of enr\_pref). I don’t see the macro being declared.

1. Program readin\_input\_files\_temp.sas
   * The last portion of the program is dropped: “\*\*\*\*\* Check Input Claims Data Sets;” and below.
   * That last portion is added to the program idcond.sas
2. Program idcond.sas
   * What are the if/else conditions on lines 79 and 80 doing?
   * I don’t think the check on variable names for id, clmdt and clmtype is working. The retained variables won’t be zero ever, just missing.
3. Program input\_program.sas
   * Variable “enr\_prefix” inside the %idcond() statement
     + It’s misspelled
     + What is its definition? It’s confusing – this is the prefix for an output file with month enrollment count. If a library is added, the file will be saved permanently.
4. Suggestions
   * Make the variable holding all codes into a macro, and eliminate the length restriction. Or give an error if the string becomes too long. May be use formats. No urgency to implement now.
   * State how you run the package at the beginning on both documents. May be move the auxiliary programs one level down, to make it clear what is the file to use and edit.
   * Eliminate the enr\_prefix parameter in the function, and save the continuous enrollment files permanently (no option given). Also ensure the name is consistent.
   * Make internal libnames with a unique name (something that probably won’t be already present, like the current name, “proj”)
   * If the %idcond is provided like in the current form, with a list of all the parameters, then default values should be added.
   * Evaluate resources needed to run. Current run on Optum data with 27 conditions is using work space up to at least 424 GB
   * Drop the print out from the lst (proc contents and proc freq)

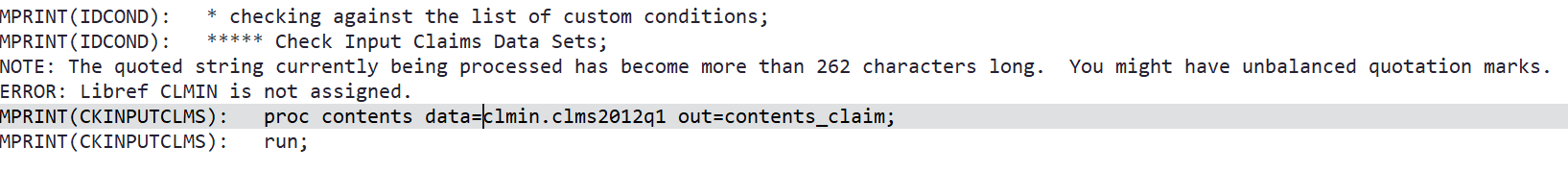
**Changes:**

Program input\_program.sas:

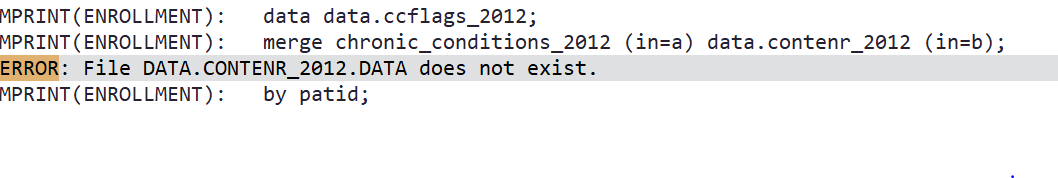
1. Drop the set up libname.
2. Idcond.sas: add a condition when checking for enrolment file shape so if the enroll (line 44)
3. Eliminate the request (allowance) of a libname before the output file prefix. Now force the permanent save, and create the libname internally (no user input).
4. Idcond.sas: drop warning on checking that prefix to the output files have a library. Now the saving is forced (lines71-76).
5. Idcond.sas: change the name for the library “proj” to “\_projxyz”. Change the definition of the library name (lines 79-80), and then its reference (3 out of 6 times).
6. Identify\_conditions.sas: change the path to the continuous enrollment files to read from the data subfolder (with libname \_projxyz) (line 996).
7. Identify\_conditions.sas: change the output file name. If prefix left blank the program still runs and it doesn’t overwrite the continuous enrollment files.
8. Idcond.sas: create a library name pointing to the csv\_input project subfolder. Place the SAS version of the default files in that folder.
9. Readin\_input\_files.sas: change the reference to the library “proj” to the library “\_projxyz”.
10. Idcond.sas: change the call to the continuous enrollment macro. Now the output file will be saved permanently in the project data library (“\_projxyz”). Change the description of the macro argument (“enr\_prefix”) accordingly.
11. Identify\_conditions.sas: change the output path to save the files permanently under the project “data” folder that comes with the package. Change the definition of the argument in Idcond.sas accordingly.
12. Identify\_conditions.sas: change the last step – the merge when saving the yearly files, to keep just individuals in enrolled at some point during the year.

Errors encountered when running:

1. Need to assign a library for the claims to be read. It seems obvious but we may want to have it on top of the documentation and have a statement in place to create the library.



1. Error in the path to the continuous enrollment files. See suggestion #3



**PF – Changes Made 3/25/2021**

* Removed all prints to the lst and some %put statements that lengthen the log
* Changed code macro creation to use proc sql loops instead of data step with set variable length
* Changed proj libname to ‘\_ccproj’
* Removed libname check and instead just use double slash //
* Changed accumulating append of all condition claims in ‘tagcond’ macro to a set of a of all the condition data sets
* Added some code to correct for user error if user requests more years out of the macro than provided in the data eg. they request years 2014-2017 but they only provide 2016 data, macro will only output 2016
  + When an error like this occurs, a warning is issued and the program adjusts to only output the years that overlap between request and provision. This equates to outputting the maximum of the min year of data requested or provided, and the minimum of the max year of data requested or provided.
* Fixing codetype length error