**Chronic Conditions Package Checklist**  
By Patricia Ferido

The purpose of the Chronic Conditions Package is to analyze claims data and flag the presence of chronic conditions based on user-defined criteria. The final outputs will be yearly beneficiary level files with monthly condition and enrollment flags.

Condition flags will have one of the four following values:

|  |  |
| --- | --- |
| .I | Insufficient data for analyzing criteria |
| 0 | No insurance coverage or qualifying claims |
| 1 | Qualifying claims, but no coverage |
| 2 | Qualifying coverage, but no claims |
| 3 | Qualifying coverage and claims |

The current set up will identify the 27 conditions currently part of the Center for Medicare/Medicaid Service’s Chronic Condition Warehouse as of the August 2017 revision. Researchers can customize the input files to look for any conditions of interest. Below is a shortened checklist of the steps needed to run the package correctly. For more detailed instructions, look at the User Guide.

* + 1. Copy the “Chronic\_Conditions\_Package” folder (and all its contents) into the location where you intend to work.
  + 2. If you want to do any of the following, navigate to the “csv\_input” folder and continue on to steps 3-5. Otherwise, skip to step 6.
* Add/remove a condition other than the 27 standard CCW conditions
* Add/remove diagnosis or procedure codes that contribute to a condition
* Change the reference period
* Change the minimum and maximum days between qualifying claims
* Change the claim types to fit your data set

**Customizing Condition Algorithms**

* + 3. Customize Diagnosis/Procedure Codes– Open the “CC\_codes” csv. Any of the four variables in the list can be customized as described below.
  + DxCode – all the diagnosis/procedure codes that will identify the condition, codes can be cleared, new codes can be added
  + Condition – shorthand condition name, must be less than 10 characters
  + Codetype – specifies kind of code used to identify the condition, can only take the following values: ICD9DX, ICD10DX, ICD9PRCDR, ICD10PRCDR, HCPCS
  + DxCodeLocation – specifies which diagnosis/procedure codes will be considered, can only take the following values:
    - ANY – any of the codes
    - DX1 DX2 – only the primary or secondary codes
    - DX1 – only the primary code
* 4. Customize reference period, claim types, minimum and maximum days apart – Open the “CC\_Desc” csv. Any of the variables can be customized as described below.
  + - Condition – has to be same shorthand name of the condition as used in “CC\_Codes”
    - Condition\_long – full name of the condition
    - Claim\_type1 & Num\_dx1 – These variables go hand in hand to specify the type of claims and number of this type of claim that qualify for the condition. If you want to look within multiple claim types without discriminating between type, then list them all in the same variable separated by a comma. For example, if you want to find 2 outpatient and carrier claims and don’t care about seeing a specific combination, then put both separated by a comma. See the following:

|  |  |
| --- | --- |
| Claim\_type1 | Num\_dx1 |
| OP,CAR | 2 |

* Can list as many claim\_type and num\_dx variables as you’d like
* You can list any claim\_type
* Min\_days\_apart – Minimum number of days between claims before counted as a new claim
* Max­\_days\_apart – Maximum number of days allowed between claims – useful for excluding claims with long periods of time between each other
* Ref\_months – look-back reference period where qualifying claims will be found
* 5. Customize exclusion codes– If you want to exclude claims based on finding certain codes, then customize “CC\_Exclude” similar to “CC\_Codes”. Otherwise, skip to step 6.

**Preparing Input Claims Data**

Multiple claims files can be input into the package. Steps 7-11 and 14 can be skipped if not relevant to your data.

* 6. If necessary, limit your claims data to people, codes, or dates of interest. To replicate the CCW, you will need to pull codes from inpatient, hospital outpatient, skilled nursing facility, carrier, and home health agency.
* 7. Identify any ICD-9 diagnosis codes and rename or recreate the variables with the prefix **ICD9dx** followed by the number of diagnosis priority. Primary diagnoses will be renamed as **ICD9dx1**. Secondary diagnoses will be renamed as **ICD9dx2**, etc.
* 8. Identify any ICD-10 diagnosis codes and rename or recreate the variables with the prefix **ICD10dx** followed by the number of diagnosis priority. Primary diagnoses will be renamed as **ICD10dx1**. Secondary diagnoses will be renamed as **ICD10dx2**, etc.
* 9. Identify any ICD-9 procedure codes and rename or recreate the variables with the prefix **ICD9prcdr** followed by the number of procedure priority.
* 10. Identify any ICD-10 procedure codes and rename or recreate the variables with the prefix **ICD10prcdr** followed by the number of procedure priority.
* 11. Identify any HCPCS codes and rename or recreate the variables with the prefix **HCPCS** followed by the number of HCPCS priority.
* 12. Identify the most reliable date for the claim in your data. In Medicare, the *thru*\_*dt* is used. Rename this variable to **claim\_dt**.
* 13. Create a variable named **claim\_type** that identifies the type of claim and matches the claim types found in the “CC\_Desc” csv or the Medicare CCW shorthand if using standard.

|  |  |
| --- | --- |
| **Claim Type** | **Standard CCW Shorthand** |
| Inpatient | IP |
| Outpatient | OP |
| Home Health Agency | HHA |
| Skilled Nursing Facility | SNF |
| Carrier | CAR |

* 14. If using Medicare and wish to replicate the CCW algorithm as much as possible then drop claims with BETOS codes D1A, D1B, D1C, D1D, D1E, D1F, D1G (which is DME), or O1A (which is ambulance services). Limit to carrier claim types 71 and 72.
* 15. Sort the data by unique beneficiary ID and claim\_dt.
* 16. Check that you have the following variables, standardized variable names and formats. For faster processing, drop any extra variables.

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

**Preparing Enrollment Data**

Enrollment data must be yearly, beneficiary level files.

* 17. Check for the existence of the continuous enrollment data which are already available on NBER, Optum and VRDC. If they exist, note their location, contents and prefix of data set names and skip to step 19. If they don’t, choose between either 18A or 18P depending on the structure of the raw enrollment data.
* 18A. Create annual enrollment data sets with the following structure:

|  |  |  |
| --- | --- | --- |
| **Annual Shape:** Enrollment input files for each year with one record per beneficiary-month. Named as such - libref.[prefix][yyyy] | | |
| **Required Variables** | **Standardized Variable Names** | **Format** |
| Unique patient identifier\* | No standard, but should be same as input claims data sets | No standard format, but should be same as input claims data sets |
| First day of month\* (one for each month) | Date\* | Date format |
| Enrollment variable\* | No standard | Binary, 1=enrolled, 0=not enrolled |

* 18P. Create a period enrollment dataset with the following structure:

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

**Running the Package**

* 19. Navigate to the “programs” folder. Open and fill out the variables in the “input\_program” SAS file.
* 25. Run “input\_program.sas”. Check the log for errors.

Your final output will be yearly beneficiary-level files with monthly condition and enrollment flags. The values for the monthly flags are below:

|  |  |
| --- | --- |
| .I | Insufficient data for analyzing criteria |
| 0 | No insurance coverage or qualifying claims |
| 1 | Qualifying claims, but no coverage |
| 2 | Qualifying coverage, but no claims |
| 3 | Qualifying coverage and claims |