

The following is an overview of the software for the Version 22 CMS-HCC risk-adjustment model. The software includes a SAS program - **V221602P** that calls several SAS Macros to create HCC score variables using coefficients from the following regression models:

- Community - Non-dual aged
- Community - Non-dual disabled
- Community - Full Benefit dual aged
- Community - Full Benefit dual disabled
- Community - Partial Benefit dual aged
- Community - Partial Benefit dual disabled
- Institutional
- New enrollee
- C-SNP new enrollee.

The set of C-SNP new enrollee coefficients is applicable to enrollees in Chronic Disease Special Needs Plans (SNP) only. These coefficients account for the fact that all new enrollees in these plans have one of the medical conditions required for C-SNP enrollment.

## Software description

The software consists of a main program V221602P that supplies user parameters to the main SAS Macro program V221602M. This macro program reads in two input files and assigns HCCs for each person. First, the program crosswalks diagnoses to Condition Categories (CCs) using SAS formats which were previously stored in the FORMAT library. Then the program creates Hierarchical Condition Categories (HCCs) by imposing hierarchies on the CCs. For persons without claims, zeros are assigned to all HCCs. After HCCs are created the program computes predicted scores from 9 regression models.

The main macro V221602M uses 6 external SAS Macro programs:

- %AGESEXV2 - create age/sex, originally disabled, disabled variables
- %V22I9ED1 - perform edits to ICD9 codes
- %V22I0ED1 - perform edits to ICD10 codes
- %V22H79L1 - assign labels to HCCs
- %V22H79H1 - set HCC=0 according to hierarchies
- %SCOREVAR - calculate a score variable

The main program, main macro and 6 external macros have a .txt extension to make the files easier to view. Please rename them to have .sas extension before running the software.

Steps performed by the software:

```
step1: include external macros
step2: define internal macro variables
step3: merge person and diagnosis files outputting one
       record per person for each input person-level
       record
       step3.1: declaration section
       step3.2: bring in regression coefficients
       step3.3: merge person and diagnosis files
       step3.4: for the first record for a person set
       CCs to 0 and create person's age
       step3.5: if there are any diagnoses for a person
               then do the following:
               - perform diagnosis edits using macro V22I9ED1 if
               ICD9 or V22I0ED1 if ICD10
               - create CCs using format provided in format
               library
               - create additional CCs using additional formats
               provided in format library
       step3.6: for the last record for a person do the
       following:
       - create demographic variables needed for score
       calculation (macro AGESEXV2)
       - create HCCs using hierarchies (macro V22H79H1)
       - create HCCs interaction variables
       - create HCCs and disabled interaction variables
       - set HCCs and interaction vars to zero if there
       are no diagnoses for a person
       - create scores for 6 community models
       - create score for institutional model
       - create score for new enrollee model
       - create score for C-SNP new enrollee model
step4: data checks and proc contents
```

## **PART 1.** Files supplied by the software.

The following SAS programs and files are included in this software:

- **V221602P** - main program that has all the parameters supplied by a user (see below for parameter and variable list). It calls main macro V221602M
- **V221602M** - main macro that creates HCC and SCORE variables by calling other external macros
- **AGESEXV2** - create age/sex, originally disabled, disabled variables
- **V22I9ED1** - performs edits to ICD9 code. Medicare Code Editor (MCE) is source of edits
- **V22I0ED1** - performs edits to ICD10 code. Medicare Code Editor (MCE) is source of edits
- **V22H79L1** - assigns labels to HCCs
- **V22H79H1** - sets HCC=0 according to hierarchies
- **SCOREVAR** - calculates a score variable
- **F221690P.ICD9.TXT** - a txt version of the format that has a cross-walk from ICD9 codes to V22 CC categories (use for reference only). This format contains ICD9 codes valid in FY2015.
- **F221690P.ICD10.TXT** - a txt version of the format that has a cross-walk from ICD10 codes to V22 CC categories (use for reference only). This format contains ICD10 codes valid in FY2016.
- **F221690P** - format library containing all the formats for the software.
- **C221405P** - relative coefficients for 9 regression models developed using CY2013/2014 data and CMS denominator \$9,185.29 (CMS 3/6/2016).

Format library and coefficients file are SAS transport files, which may be used on any platform running SAS, after uploading and converting using PROC CIMPORT. Users should use the following code to convert them.

Code for converting coefficients transport file to SAS file:

```
filename inc "C:\user defined location of the transport
file\C221405P";
libname incoef "C:\user defined location of the sas
coefficients file";
proc cimport data=incoef.hcccoefn infile=inc;
run;
```

Code for converting formats transport file to SAS file:

```
filename inf "C:\user defined location of the transport
file\F221690P";
```

```
libname library "C:\user defined location of the sas
formats file";
proc cimport library=library infile=inf;
run;
```

If you are operating in an MVS - z/OS environment, the transport files should be uploaded using the following parameters:

```
RECFM(F or FB) LRECL(80) BLKSIZE(8000)
```

## **PART 2.** Files supplied by a user.

Two SAS input files needed for the software must be presorted in ascending order by the person ID variable

- 1) **PERSON** file--a person-level file of demographic and enrollment information
- 2) **DIAG** file--a diagnosis-level input file of diagnoses

Data requirements for the SAS input files. The variable names listed are required by the programs as written:

### 1) **PERSON** file

- **HICNO** (or other person identification variable. It must be set in the macro variable IDVAR)  
-character or numeric type and unique to an individual
- **SEX**  
-one character, 1=male; 2=female
- **DOB**  
- SAS date format, date of birth
- **LTIMCAID**  
-numeric, =1 if number of months in Medicaid in *payment* year >0;  
=0 otherwise
- **NEMCAID**  
-numeric, =1 if a new Medicare enrollee and number of months in Medicaid in *payment* year >0;  
=0 otherwise

- **OREC**  
-one character, original reason for entitlement with the following values:  
0 - OLD AGE (OASI)  
1 - DISABILITY (DIB)  
2 - ESRD  
3 - BOTH DIB AND ESRD

2) **DIAG** file--a diagnosis file with at least one record per person-specific unique diagnosis.

- **HICNO** (or other person identification variable that must be the same as in PERSON file)  
- person identifier of character or numeric type and unique to an individual
- **DIAG**  
-Diagnosis code, 7 character field, no periods, left justified. The user may include all diagnoses or limit the codes to those used by the model. Codes should be to the greatest level of available specificity. Diagnoses should be included **only** from providers and physician specialties allowable for risk adjustment reporting (as specified in CMS notices).
- **DIAG\_TYPE**  
- Diagnosis code version, 1 character field, with the following values:  
9 for ICD9  
0 for ICD10

### **Part 3.** Parameters supplied by a user:

*NOTE: All user-supplied parameters should be reentered by the user. The default settings are examples only, and should not be used.*

The user must supply the following:

- **INP** - SAS input person dataset name
- **IND** - SAS input diagnosis dataset name
- **OUTDATA** - SAS output dataset name
- **IDVAR** - name of person identifier variable (HICNO for Medicare data)
- **KEEPVAR** - variables kept in the output dataset. There is a list of KEEP variables in the program, but the user can alter the list.

- **SEDITS** - a switch that controls whether to perform MCE edits on ICD9 and ICD10  
1-YES, 0-NO
- **DATE\_ASOF** - reference date to calculate age. Set to February 1 of the payment year for consistency with CMS.

#### **Part 4.** Variables output by the software.

The software outputs a person level file. Any variables that the user wants to keep in it should be specified in the main program **V221602P** in **KEEPVAR** parameter of macro **V221602M** call. The following variables can be specified:

- 1) Any person level variables from the original person level file
- 2) Demographic variables created by the software:  
**AGEF ORIGDS DISABL**  
**F0\_34 F35\_44 F45\_54 F55\_59 F60\_64 F65\_69**  
**F70\_74 F75\_79 F80\_84 F85\_89 F90\_94 F95\_GT**  
**M0\_34 M35\_44 M45\_54 M55\_59 M60\_64 M65\_69**  
**M70\_74 M75\_79 M80\_84 M85\_89 M90\_94 M95\_GT**  
**NEF0\_34 NEF35\_44 NEF45\_54 NEF55\_59 NEF60\_64**  
**NEF65 NEF66 NEF67 NEF68 NEF69**  
**NEF70\_74 NEF75\_79 NEF80\_84 NEF85\_89 NEF90\_94**  
**NEF95\_GT**  
**NEM0\_34 NEM35\_44 NEM45\_54 NEM55\_59 NEM60\_64**  
**NEM65 NEM66 NEM67 NEM68 NEM69**  
**NEM70\_74 NEM75\_79 NEM80\_84 NEM85\_89 NEM90\_94**  
**NEM95\_GT**
- 3) **HCCs** defined in the main program **V221602P** by the macro variable **&HCCV22\_list79**
- 4) **CC's** (condition categories assigned before hierarchies are applied) defined in the main program **V221602P** by the macro variable **&CCV22\_list79**
- 5) Score variables:
  - **SCORE\_COMMUNITY\_NA** - community model Non Dual Aged
  - **SCORE\_COMMUNITY\_ND** - community model Non Dual Disabled
  - **SCORE\_COMMUNITY\_FBA** - community model Full Benefit Dual Aged
  - **SCORE\_COMMUNITY\_FBD** - community model Full Benefit Dual Disabled

- **SCORE\_COMMUNITY\_PBA** - community model Partial Benefit Dual Aged
- **SCORE\_COMMUNITY\_PBD** - community model Partial Benefit Dual Disabled
- **SCORE\_INSTITUTIONAL** - institutional model
- **SCORE\_NEW\_ENROLLEE** - new enrollee model
- **SCORE\_SNP\_NEW\_ENROLLEE** - new enrollee model for Chronic Disease SNP plans only

The user should determine which of the nine scores is appropriate for the beneficiary depending upon the status of that beneficiary.