ABSTRACT

The Charlson score is one of the most widely used measures of the burden of disease and the prognostic impact of that disease burden. It is often used in epidemiologic studies, and as method of risk adjustment in policy analyses. The Charlson score considers which of the 17 diseases of interest a person has, assigns a point score to each based on its mortality risk, and sums them up to generate a score of disease burden. The final risk scores range between 0 and 41, with higher values associated with higher mortality risk over 10 years. Each disease in the index has an assigned weight of 1, 2, 3 or 6, with AIDS and Metastatic Solid Tumor having the highest weight of 6. Computing Charlson score from large claims datasets, including Medicare claims data, is possible due to richness of data in those claims. For each patient we have bills (claims) submitted to Medicare, which cover bills for services delivered in a wide variety of settings. Each claim includes an ICD-9 code, the diagnosis for which the patient was treated, and these ICD-9 codes can be translated into the diseases that make up the Charlson score. We developed a SAS macro that calculates the Charlson comorbidity score from Medicare claims. The macro determines if it is possible to calculate the index for each subject (i.e., if the subject had complete Medicare claims for at least one year), converts the ICD9 codes to comorbidities and combines the diagnoses recorded in each claim from outpatient, inpatient and carrier files. The macro outputs for each subject the presence of the individual comorbidities and the overall Charlson score.

INTRODUCTION

Comorbidity is defined as presence and severity of medical comorbid conditions in a patient. The medical conditions can exist independently from each other, or they can be related to each other. Comorbidity can be used to describe the overall health of a patient and should be taken into account when deciding on treatment options. Comorbidity levels might alter the mortality risk or risk of other health outcomes in aging population and should therefore be accounted for in statistical models for those outcomes. Not adjusted for the comorbidity status might lead to overestimates of effect that the predictor of interest has on the outcome. The most efficient way to adjust for comorbidity is a single comorbidity score. This is especially useful way of determining overall health status in research using administrative data because we can translate a large amount of claims data into one single value.

We developed a SAS macro program that calculates the Charlson Comorbidity Index (CCI) from Centers for Medicare & Medicaid Services (CMS) claims data. Charlson score was originally developed by Charlson et al in 1987, and there have since been many modified version of CCI published. The version that we use in this macro considers 17 conditions listed in Table 1 and can range from 0 to 41. Each condition is identified by ICD 9 codes in one of the diagnoses fields in claims submitted by medical providers. Claims data are not an exact representation of a patient’s health status, but they are a very accurate representation of services that were provided to the patient and the reason (diagnosis) those services were provided. Since providers’ payment for services also partially depends on patient’s health status, providers have an incentive to report all medical conditions the patient has. CCI is often calculated by using the diagnoses in the year prior to the index date. That does not mean that the patient had to receive the initial diagnosis in the previous year, but that the patient had to be tested, treated or required any kind of medical intervention for that condition in the last year.

In this paper we briefly describe the CMS claims data files that we use in the macro and we describe the five main steps needed for calculating the Charlson Comorbidity Index.
Macro computing Charlson Comorbidity Index from CMS Claims Data, continued

Table 1: Conditions included in Charlson Comorbidity Index and the points associated with each condition.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Points in CCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Myocardial Infarction</td>
<td>1</td>
</tr>
<tr>
<td>CHF</td>
<td>1</td>
</tr>
<tr>
<td>Peripheral Vascular Disease</td>
<td>1</td>
</tr>
<tr>
<td>Cerebrovascular Disease</td>
<td>1</td>
</tr>
<tr>
<td>COPD</td>
<td>1</td>
</tr>
<tr>
<td>Dementia</td>
<td>1</td>
</tr>
<tr>
<td>Paralysis</td>
<td>1</td>
</tr>
<tr>
<td>Diabetes</td>
<td>1</td>
</tr>
<tr>
<td>Diabetes With Sequelae</td>
<td>2</td>
</tr>
<tr>
<td>Chronic Renal Failure</td>
<td>2</td>
</tr>
<tr>
<td>Various Cirrhodites</td>
<td>1</td>
</tr>
<tr>
<td>Moderate-Severe Liver Disease</td>
<td>3</td>
</tr>
<tr>
<td>Ulcers</td>
<td>1</td>
</tr>
<tr>
<td>Rheumatitis</td>
<td>1</td>
</tr>
<tr>
<td>AIDS</td>
<td>6</td>
</tr>
<tr>
<td>Any Malignancy</td>
<td>2</td>
</tr>
<tr>
<td>Metastatic Solid Tumor</td>
<td>6</td>
</tr>
</tbody>
</table>

**MEDICARE DATA DESCRIPTION**

CMS organizes the Medicare claims data in multiple data files, depending on the billing party. The four data files that we use for calculating Charlson comorbidity index are described by Research Data Assistance Center (RedDAC) as follows:

- **Denominator file** – includes demographic and enrollment information about each patient. This is the file we will use to determine whether a patient is enrolled in Fee-For-Service (FFS) program or Health Management Organization (HMO).
- **Inpatient file** – includes claims data submitted by inpatient hospital providers.
- **Outpatient file** – includes claims data submitted by institutional outpatient providers.
- **Carrier file** – includes claims submitted by non-institutional providers. (For example, physicians, physician assistant, nurse practitioners, ambulance providers, etc.)

Each of the last three files contains information about dates of service and diagnosis (ICD-9 codes) for each claim.

**STEPS FOR CALCULATING CHARLSON INDEX**

Below is the short description of each step needed to calculate the Charlson Comorbidity Index.

**STEP 1:** Exclude patients that had not been enrolled for one full year in Fee-For-Service program

Claims data only include information on services provided while a patient was in Fee-For-Service program. If a patient enrolls in HMO, Medicare pays HMO a flat annual rate for the patient but does not pay providers for any services performed. HMOs are not required to report their services provided (claims), and therefore it is impossible to know what procedures were done on the patients and what conditions the patient was diagnosed with from CMS data. If we attempt to calculate CCI for the patients that were on HMO for one or more months in the last year, we will underestimate their disease burden. Therefore, we will exclude the HMO patients from the CCI calculations.
Macro computing Charlson Comorbidity Index from CMS Claims Data, continued

We start with identifying patients that are on FFS and that have been on FFS for at least one year before the index date. We create a master denominator file that contains the HMO status variable for each year of our study. In CMS data the HMO status is stored in variable HMOIND12, which is a string variable with 12 characters. Each character indicates whether the patient was on HMO that month. FFS program is identified by values 0 or 4. If we want to calculate CCI for any date between 2000 and 2008, our master file should include variables HMOIND12_99 to HMOIND12_08. The following piece of code excludes the individuals that were on HMO for one or more months in the year prior to the index date.

```
if admission_year = 2000 and admission_month ne 12 then
    if (substr(HMOIND12_00, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_00, admission_month+1, 1) not in ("0" "4")) then delete;
else if (substr(HMOIND12_00, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_01, admission_month, 1) not in ("0" "4")) then delete;
    if (substr(HMOIND12_01, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_01, admission_month+1, 1) not in ("0" "4")) then delete;
    if (substr(HMOIND12_01, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_02, admission_month, 1) not in ("0" "4")) then delete;
else if (substr(HMOIND12_02, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_02, admission_month+1, 1) not in ("0" "4")) then delete;
    if (substr(HMOIND12_02, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_03, admission_month, 1) not in ("0" "4")) then delete;
else if (substr(HMOIND12_03, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_03, admission_month+1, 1) not in ("0" "4")) then delete;
    if (substr(HMOIND12_03, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_04, admission_month, 1) not in ("0" "4")) then delete;
else if (substr(HMOIND12_04, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_04, admission_month+1, 1) not in ("0" "4")) then delete;
else if (substr(HMOIND12_04, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_05, admission_month, 1) not in ("0" "4")) then delete;
    if (substr(HMOIND12_06, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_05, admission_month, 1) not in ("0" "4")) then delete;
else if (substr(HMOIND12_06, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_06, admission_month+1, 1) not in ("0" "4")) then delete;
else if (substr(HMOIND12_06, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_07, admission_month, 1) not in ("0" "4")) then delete;
else if (substr(HMOIND12_07, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_07, admission_month+1, 1) not in ("0" "4")) then delete;
else if (substr(HMOIND12_08, admission_month, 1) not in ("0" "4") or
        substr(HMOIND12_08, admission_month+1, 1) not in ("0" "4")) then delete;
```

Output 1: Exclude patients on HMO

**STEP 2:** Identify the presence of each condition in each one of the three claims datasets

Each claim contains the ICD9 codes for the conditions that were diagnosed at the time. We use those codes to determine if the patient has each one of 17 conditions included in CCI. Providers usually report all conditions identified or affecting the patient at the time of treatment, and not only the condition that was the primary reason for the medical encounter. This leads to more accurate and complete description of a patient’s health status. The code below shows how we identify the conditions in the Inpatient file. The same logic is applied in the Outpatient and Carrier files.
Macro computing Charlson Comorbidity Index from CMS Claims Data, continued

There are 11 fields with relevant information in the Inpatient file: one Admitting Diagnosis (AD_DGNS) and 10 additional diagnoses (DGNSCD01-DGNSCD10).

$$\text{array CONDITIONS(17) ANYMI CHF VASCULI CVD PULMON1 DEMENTIA PARALYS DIABET1 DIABET3 RENAL1 LIVER1 LIVER2 ANYULCER RHEUM AIDS MALIGNANCY METASTATIC;}
$$
$$\text{do i=1 to 17; CONDITIONS(i)=0; end;}
$$
$$\text{array dxcode AD_DGNS DGNSCD01-DGNSCD10; do over dxcode;}
$$
$$\text{dx_3=strchr(dxcode,1,3); dx_4=strchr(dxcode,1,4);}
$$
$$\text{//********** MYOCARDIAL INFARCTION WEIGHT=1 **********/}
$$
$$\text{if dx_3=410' then CVD=1;}
$$
$$\text{else if dx_3=412' then CVD=1;}
$$
$$\text{//********** PERIPHERAL VASCULAR DISEASE WEIGHT=1***/}
$$
$$\text{if dx_3=441' dx_4 in (4439,7854,'V434', 'V434') then VASCULI=1;}
$$
$$\text{//********** CEREBROVASCULAR DISEASE WEIGHT=1/****/}
$$
$$\text{if '430'<=dx_3<='437' then CHF=1;}
$$
$$\text{//********** COPD WEIGHT=1**************/}
$$
$$\text{if '490'<=dx_3<='496' '500'<dx_3<='5064' then PULMON1=1;}
$$
$$\text{//********** DEMENTIA WEIGHT=1**************/}
$$
$$\text{if dx_3=290' then DEMENTIA=1;}
$$
$$\text{//********** PARALYSIS WEIGHT=1 **************/}
$$
$$\text{if dx_3=3441' then PARALYS=1;}
$$
$$\text{//********** DIABETES WEIGHT=1**************/}
$$
$$\text{if dxcode='250 '|dx_4='2507'|'2500'<dx_4<='2503' then DIABET1=1;}
$$
$$\text{//********** DIABETES WITH SEQUELAE WEIGHT=2********/}
$$
$$\text{if '2504'<dx_4<='2506' then DIABET3=1;}
$$
$$\text{//********** MODERATE-SEVERE LIVER DISEASE WEIGHT=3*/}
$$
$$\text{if dx_3 in ('5312', '5314', '5714', '5716') then LIVER1=1;}
$$
$$\text{//********** VARIOUS CIRRHODITES WEIGHT=1********/}
$$
$$\text{if dxcode in ('5324', '5327') then LIVER2=1;}
$$
$$\text{//********** CHRONIC RENAL FAILURE WEIGHT=2*********/}
$$
$$\text{if dx_3 in ('582', '585', '586', '588')|('5830'<dx_4<='5837') then RENAL1=1;}
$$
$$\text{//********** MODERATE-SEVERE LIVER DISEASE WEIGHT=3*/}
$$
$$\text{if dxcode in ('5300', '5320')|('5330'<dx_4<='5337') then RENAL1=1;}
$$
$$\text{//********** MODERATE-SEVERE LIVER DISEASE WEIGHT=3*/}
$$
$$\text{if dx_4 in ('531 ', '5319', '532 ', '5329', '53 ', '5339', '534 ', '5349') then do; ULCER1=1; ULCER1=1;}
$$
$$\text{//********** VARIOUS CIRRHODITES WEIGHT=1********/}
$$
$$\text{if dxcode in ('5314', '5324')|('5334'<dx_4<='5344') then do; ULCER2=1; ULCER1=1;}
$$
$$\text{//********** AIDS WEIGHT=6**********/}
$$
$$\text{if dxcode in ('71481', '7125', '7100 ', '7101 ', '7104 ')|('7140'<dx_4<='7142' then RHEUM=1;}
$$
$$\text{//********** ANY MALIGNANCY WEIGHT=2**********/}
$$
$$\text{if (140'<dxcode='1729')|('174'<dxcode='1958')|('200'<dxcode='2089') then MALIGNANCY=1;}
$$
$$\text{//********** METASTATIC SOLID TUMOR WEIGHT=6***/}
$$
$$\text{if 196'<dxcode='1991' then METASTATIC=1; end;}
$$

**Output 2: Determine presence of each condition**

**STEP 3: Determine if the condition was diagnosed within one year before the index date**

We are only interested in conditions that were diagnosed within one year before the index date. If a patient was diagnosed with a condition, but that diagnosis occurred more than one year ago, it will not contribute to CCI. The code below shows how we determined if a diagnosis was recorded within one year of index date in the Inpatient file. The same logic is applied in the Outpatient and Carrier files.

The date of hospitalization in the Inpatient file is found in the field ADMSN_DATE.

```sql
proc sql;
create table cci_inpatient as
select a1.ID, a1.admission_date,
  case when x1.ANYMI =1 then 1 else 0 end AS ANYMI,
  case when x2.CHF = 1 then 1 else 0 end AS CHF,
  case when x3.VASCULI = 1 then 1 else 0 end AS VASCULI,
  case when x4.CVD = 1 then 1 else 0 end AS CVD,
  case when x5.PULMON1 = 1 then 1 else 0 end AS PULMON1,
  case when x6.DEMENTIA = 1 then 1 else 0 end AS DEMENTIA,
  case when x7.PARALYS = 1 then 1 else 0 end AS PARALYS,
  case when x8.DIABET1 = 1 then 1 else 0 end AS DIABET1,
  case when x9.DIABET3 = 1 then 1 else 0 end AS DIABET3,
  case when x10.RENAL1 = 1 then 1 else 0 end AS RENAL1,
  case when x11.LIVER1 = 1 then 1 else 0 end AS LIVER1,
  case when x12.LIVER2 = 1 then 1 else 0 end AS LIVER2,
  case when x13.ANYULCER = 1 then 1 else 0 end AS ANYULCER,
  case when x14.RHEUM = 1 then 1 else 0 end AS RHEUM,
  /*...*/
  case when x17.METASTATIC = 1 then 1 else 0 end AS METASTATIC
  from cci_record as a1
  inner join sub_cms_claims as a2
    on a1.admission_date = a2.ADMN_DATE
  where a2.CLAIMS_DATE <='19601001';
```
CASE WHEN x15.AIDS = 1 THEN 1 ELSE 0 END AS AIDS,
CASE WHEN x16.MALIGNANCY = 1 THEN 1 ELSE 0 END AS MALIGNANCY,
CASE WHEN x17.METASTATIC = 1 THEN 1 ELSE 0 END AS METASTATIC
FROM original_dataset a1
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.ANYMI = 1 THEN 1 ELSE 0 END AS ANYMI
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.ANYMI = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x1
ON a1.ID = x1.ID AND a1.admission_date = x1.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.CHF = 1 THEN 1 ELSE 0 END AS CHF
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.CHF = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x2
ON a1.ID = x2.ID AND a1.admission_date = x2.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.VASCUL1 = 1 THEN 1 ELSE 0 END AS VASCUL1
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.VASCUL1 = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x3
ON a1.ID = x3.ID AND a1.admission_date = x3.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.CVD = 1 THEN 1 ELSE 0 END AS CVD
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.CVD = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x4
ON a1.ID = x4.ID AND a1.admission_date = x4.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.PULMON1 = 1 THEN 1 ELSE 0 END AS PULMON1
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.PULMON1 = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x5
ON a1.ID = x5.ID AND a1.admission_date = x5.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.DEMENTIA = 1 THEN 1 ELSE 0 END AS DEMENTIA
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.DEMENTIA = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x6
ON a1.ID = x6.ID AND a1.admission_date = x6.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.PARALYS = 1 THEN 1 ELSE 0 END AS PARALYS
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.PARALYS = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x7
ON a1.ID = x7.ID AND a1.admission_date = x7.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.DIABET1 = 1 THEN 1 ELSE 0 END AS DIABET1
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.DIABET1 = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x8
ON a1.ID = x8.ID AND a1.admission_date = x8.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.DIABET3 = 1 THEN 1 ELSE 0 END AS DIABET3
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.DIABET3 = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x9
ON a1.ID = x9.ID AND a1.admission_date = x9.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.RENAL1 = 1 THEN 1 ELSE 0 END AS RENAL1
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.RENAL1 = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x10
ON a1.ID = x10.ID AND a1.admission_date = x10.admission_date
LEFT OUTER JOIN
(SELECT DISTINCT a.ID, a.admission_date, CASE WHEN i.LIVER1 = 1 THEN 1 ELSE 0 END AS LIVER1
FROM original_dataset a
JOIN inpatient_charlson i ON a.ID = i.ID AND i.LIVER1 = 1
AND (a.admission_date - i.ADMSN_DATE) >= 0 AND (a.admission_date - i.ADMSN_DATE) <= 365) x11
ON a1.ID = x11.ID AND a1.admission_date = x11.admission_date
Macro computing Charlson Comorbidity Index from CMS Claims Data, continued

Output 4: Check for conditions in each claims dataset

```sql
LEFT OUTER JOIN
   (select distinct a.ID, a.admission_date, case when i.LIVER2 = 1 then 1 else 0 end as LIVER2
   from original_dataset a
JOIN inpatient_charlson i on a.ID =i.ID and i.LIVER2 =1
   and (a.admission_date-i.ADMSN_DATE)>=0 and (a.admission_date-i.ADMSN_DATE)<365) x12
ON a1.ID =x12.ID and a1.admission_date = x12.admission_date

LEFT OUTER JOIN
   (select distinct a.ID, a.admission_date, case when i.ANYULCER = 1 then 1 else 0 end as ANYULCER
   from original_dataset a
JOIN inpatient_charlson i on a.ID =i.ID and i.ANYULCER =1
   and (a.admission_date-i.ADMSN_DATE)>=0 and (a.admission_date-i.ADMSN_DATE)<365) x13
ON a1.ID =x13.ID and a1.admission_date = x13.admission_date

LEFT OUTER JOIN
   (select distinct a.ID, a.admission_date, case when i.METASTATIC = 1 then 1 else 0 end as METASTATIC
   from original_dataset a
JOIN inpatient_charlson i on a.ID =i.ID and i.METASTATIC =1
   and (a.admission_date-i.ADMSN_DATE)>=0 and (a.admission_date-i.ADMSN_DATE)<365) x17
ON a1.ID =x17.ID and a1.admission_date = x17.admission_date;
```

Output 3: Determine if the condition was diagnosed in the last year

**STEP 4:** Combine the three dataset and see if each condition appears in at least one claims dataset

Each patient can have records in one or more of the claims datasets, and not necessarily in all three. It is important to check for each condition in each of the three claims datasets. After identifying conditions that were recorded within one year of the index date in each dataset, we performed another step that combined the information from all Inpatient, Outpatient and Carrier datasets.

```sql
SELECT DISTINCT ic.ID, ic.admission_date,
case when ic.ANYMI = 1 or cc.ANYMI = 1 or oc.ANYMI = 1 then 1 else 0 end AS ANYMI,
case when ic.CHF = 1 or cc.CHF = 1 or oc.CHF =1 then 1 else 0 end as CHF,
case when ic.VASCUL1 =1 or oc.VASCUL1 = 1 or cc.VASCUL1 =1 then 1 else 0 end AS VASCUL1,
case when ic.CVD =1 or oc.CVD = 1 or cc.CVD =1 then 1 else 0 end as CVD,
case when ic.PULMON1 = 1 or cc.PULMON1 = 1 or oc.PULMON1 =1 then 1 else 0 end AS PULMON1,
case when ic.DEMENTIA = 1 or cc.DEMENTIA = 1 or oc.DEMENTIA =1 then 1 else 0 end AS DEMENTIA,
case when ic.PARALYS =1 or oc.PARALYS =1 or cc.PARALYS =1 then 1 else 0 end as PARALYS,
case when ic.DIABET1 = 1 or cc.DIABET1 = 1 or oc.DIABET1 =1 then 1 else 0 end AS DIABET1,
case when ic.DIABET3 = 1 or cc.DIABET3 = 1 or oc.DIABET3 =1 then 1 else 0 end AS DIABET3,
case when ic.RENAL1 = 1 or cc.RENAL1 = 1 or oc.RENAL1 =1 then 1 else 0 end AS RENAL1,
case when ic.LIVER1 = 1 or cc.LIVER1 = 1 or oc.LIVER1 =1 then 1 else 0 end AS LIVER1,
case when ic.LIVER2 = 1 or cc.LIVER2 = 1 or oc.LIVER2 =1 then 1 else 0 end AS LIVER2,
case when ic.ANYULCER = 1 or oc.ANYULCER = 1 or cc.ANYULCER =1 then 1 else 0 end AS ANYULCER,
case when ic.RHEUM = 1 or cc.RHEUM = 1 or oc.RHEUM =1 then 1 else 0 end AS RHEUM,
case when ic.AIDS = 1 or cc.AIDS = 1 or oc.AIDS =1 then 1 else 0 end AS AIDS,
case when ic.MALIGNANCY = 1 or cc.MALIGNANCY = 1 or oc.MALIGNANCY =1 then 1 else 0 end AS MALIGNANCY,
case when ic.METASTATIC = 1 or cc.METASTATIC = 1 or oc.METASTATIC =1 then 1 else 0 end AS METASTATIC
FROM cci_inpatient ic
INNER JOIN cci_outpatient oc ON ic.ID = oc.ID AND ic.admission_date = oc.admission_date
INNER JOIN cci_carrier cc ON ic.ID = cc.ID AND ic.admission_date = cc.admission_date;
```
Macro computing Charlson Comorbidity Index from CMS Claims Data, continued

**STEP 5: Calculate the Charlson Comorbidity Index**

Finally, we use the information we have from each of the three claims datasets about each condition to calculate the Charlson Comorbidity Index.

```sas
Charlson=(ANYMI * 1) +
(CHF * 1) +
(VASCULI * 1) +
(CVD * 1) +
(PULMONI * 1) +
(DEMENTIA * 1) +
(FARALYS * 1) +
(DIABET1 * 1) +
(DIABET3 * 2) +
(RENALL * 2) +
(LIVER1 * 1) +
(LIVER2 * 3) +
(ANYULCER * 1) +
(RHEUM * 1) +
(AIDS * 6) +
(MALIGNANCY * 2) +
(METASTATIC * 6);
r
```

**Output 5: Calculate CCI**

**CONCLUSION**

Comorbidity indices are efficient ways to evaluate patients’ overall health status and can be adjusted in statistical models. The Charlson score is one of the most widely used measures of the burden of disease and the prognostic impact of that disease burden. Claims data include a lot of information that can be used to calculate the Charlson score, but a few steps are required to properly use the data. SAS code presented here calculates the CCI from the original claims data. Full macro can be obtained by contacting the authors.

**REFERENCES**


**CONTACT INFORMATION**

Your comments and questions are valued and encouraged. Contact the author at:

Name: Irena Stijacic Cenzer  
Enterprise: UCSF  
Address: 4150 Clement St.  
City, State ZIP: San Francisco, CA 94121  
Work Phone: (415) 221 4810  
E-mail: irena.stijacic@ucsf.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.