Coding ICD-9 and Charlson Index Scores in R for SAS Users

Vanessa L. Cox, MS[1,2], and Kimberly A. Wildes, DrPH, MA, LPC, NCC[3]

[1]General Internal Medicine and Ambulatory Treatment, The University of Texas MD Anderson Cancer Center, Houston, TX. vlcox@mdanderson.org

[2]PhD Candidate, Department of Statistics, Texas A&M University

[3]Private Practice, Houston, TX. info@kimberlycounseling.com

Introduction:

ICD-9 codes are used in large hospital and medical record databases to capture patient comorbidity information and cause of death. By weighting certain comorbidities, the Charlson Index calculates a comorbidity score that assigns a level of health to an individual. Accounting for or controlling for patient comorbidity in statistical analyses, for example in survival analysis, is crucial and, in fact, generally expected, as demonstrated by the widely reported use of comorbidity scores in the scientific literature,. Coding algorithms for the Charlson Index exist for SAS and SPSS; however, such algorithms do not exist for analyses with R.

The most difficult part of developing a coding algorithm function in R for the Charlson Index is data manipulation. It is common to use scripting languages to prepare datasets for analysis with R. The goal of this project was to develop an R scoring package that was "all inclusive" and could be run with minimal data preparation in order to compute the Charlson Index.

ICD-9

Wikipedia defines International Statistical Classification of Diseases and Related Health Problems (ICD) as "codes to classify diseases and a wide variety of signs, symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or disease. Under this system, every health condition can be assigned to a unique category and given a code, up to six characters long. Such categories can include a set of similar diseases." The ICD coding is currently in version 10, with version 11 expected to be released in 2015. However, version 9 is commonly used in data sets that are being analyzed currently. Version 10 is being used to code

new information. The ICD codes are published by the World Health Organization and are nearly ubiquitously used in hospital billing and patient records.

For example, consider the following classifications for elbow and hypertension. Each classification level branches off into more specific categories:

Example 1: Dislocated elbow - coded as 832 based on the following classification:
Injury and Poisoning->Dislocation->elbow
Example 2: Hypertension – coded as 401.x as follows:
Diseases of circulatory system->hypertensive diseases->essential hypertension->Hypertension
Hypertension coding would continue into Hypertension (benign), Hypertension (malignant), or Hypertension (unspecified) with codes 401.0, 401.1, or 401.9, respectively.

Charlson

The Charlson Index was designed to predict one-year mortality for a patient by examining 22 medical conditions. Each condition is assigned a weight of 1, 2, 3, or 6 depending on the level of its impact on mortality. The weighted conditions are then summed, with a higher score corresponding to a higher risk of mortality.

It is possible to have multiple comorbidities per patient because comorbidity data are often collected across time and across multiple visits. It is usually expected that none of these comorbidities medically resolve. Therefore, we would like to capture all of the comorbidities that each patient has reported at least once. The Charlson Index assigns a value to a patient's

health by assessing a set of comorbidities, some with higher ranks than others. This index has been validated in numerous studies as to the accuracy of patient health denoted by the Charlson Index rating. The ICD-9 code is a way to standardize how comorbidities are recorded. With so much electronic information available, a standardized system for collecting disease information becomes extremely helpful when analyzing data.

The Charlson R package created in this project accomplishes two things. First, it recodes Statistical Classification of Diseases and Related Health Problems version 9 (ICD-9) entries into categories useful for computing the Charlson Index. Second, it computes the Charlson Index for entries in a dataset. It contains one function named *Charlson*, which takes two vector arguments.

Methods

A function named Charlson was developed in R (Complete code is in Appendix 1). After writing the function, an R package was built. The package includes the code, a description of the code, input variables, output from the function, sample data set, and examples. This package requires the *Reshape* and *Plyr* packages to run properly. The melt and cast functions from *Reshape* are used in *Charlson*, while the *Reshape* package itself requires *Plyr*. To create an R package, the basic steps performed were:

1. Open a clean R session.
2. Load the data sets.
3. Run the functions.
4. Run the function package.skeleton. This created the basic structure of the files and directories needed for a package.

5. Update and modify all of the files created.
6. From the batch prompt, type "R CMD check *package-name"*. Correct any errors or warnings.
7. If there were not any errors or warnings, from the batch prompt, type "R CMD build *package-name"*.

A summary of the code for the *Charlson* function is listed below.

- Input the data (column 1 = Patient ID, column 2 = ICD-9 codes).
- Initialize a vector for the disease codes.
- Create vectors of disease codes (based on ICD-9 codes) for each of the 23 disease categories: Acute Myocardial Infarction, Congestive Heart Failure, Peripheral Vascular Disease, Cerebrovascular Disease, Dementia, Chronic Pulmonary Disease, Rheumatologic Disease, Peptic Ulcer Disease, Mild Liver Disease, Diabetes without complications, Diabetes with chronic complications, Hemiplegia or Paraplegia, Renal Disease, Cancer, Moderate or Severe Liver Disease, Metastatic Carcinoma, and AIDS/HIV. Each of these vectors could contain one or more disease codes, depending on the number of ICD-9 codes that correspond to that illness/condition.
- Check row by row for each disease category, assign corresponding disease classification to ICD-9 code, denoted "ch1" to "ch17".
- Add rows to the end with one of each of the comorbidities to ensure even categories that no patients have a variable created when transposed. The ID for this row is marked "xxxREMOVExxx" to ensure it is not accidentally included in the final data set.
- Use the *melt* function in the *Reshape* package to summarize the data by patients.

- Use the cast function in the Reshape package to transpose the data by Patient ID.
- Replace any disease category count over 1 with a 1 to indicate presence. Note, the number of times a disease is recorded is not important in scoring the Charlson Index.
- Assign appropriate weights per disease category.
- Compute a comorbidity count for each patient.
- Compute the Charlson Index for each patient.
- Return the final dataset.

The beginning of the program classifies each ICD-9 code to the corresponding disease category for the Charlson. The data are then transposed by patient so that across each patient row we can see which of the disease categories are present. Then a count of comorbidities and the Charlson Index is calculated.

Portion of sample input data:

YearDeath	Sequence	ICD9
83	5157	2
86	18061	2
90	17986	2
86	17817	2
77	19306	2
84	9408	1
76	15659	1
79	16887	1
80	3946	410
80	3946	428
80	3946	3
80	3946	7100
80	3946	3
80	3946	3
80	3946	3
80	3946	3
80	3946	5830
80	3946	3
80	3946	3
80	3946	199
80	3946	3
80	3946	2505
80	3946	3
80	3946	3
80	3946	2504
84	107	410
84	107	428

Corresponding Sample Output Data:

Sequence	ch1	ch10	ch11	ch12	ch13	ch14	ch15	ch16	ch17	ch2	ch3	ch4	ch5	ch6	ch7	ch8	ch9	NA	comorbidity.n	charleson
15659	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
16887	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
17817	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
17986	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
18061	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
19306	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3946	1	0	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	10	6	12
5157	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
9408	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Results

Creation of a program to code the Charlson Index resulted in the following R code. The code ran without any errors or warnings. The dataset used to validate the function was the National Health and Nutrition Examination Survey (NHANES) mortality file from 1992. According to their website, “The updated NHANES II Linked Mortality File provides mortality follow-up data through December 31, 2006 for the 9,252 NHANES II participants who were 30-75 years of age and completed a medical examination during the survey period (1976-1980).” These data are not for public use, therefore, a sample data file that had a similar layout to the mortality file was created.

R code:

```
# Function input is a vector of patient ID and a vector of corresponding ICD-9 codes
charlson <- function(Sequence,icd9) {
d <- cbind.data.frame(Sequence,icd9)
nobs<-nrow(d)
```

```
# initialize code variable
d$code<-rep(NA,nobs)
# Create a vector of disease codes for each of the 23 categories (based on ICD-9 codes):
# Acute Myocardial Infarction, Congestive Heart Failure, Peripheral Vascular Disease,
Cerebrovascular Disease, Dementia, Chronic Pulmonary Disease, Rheumatologic Disease, Peptic
Ulcer Disease, Mild Liver Disease, Diabetes without complications, Diabetes with chronic
complications, Hemiplegia or Paraplegia, Renal Disease, Cancer, Moderate or Severe Liver
Disease, Metastatic Carcinoma, AIDS/HIV
# disease classes based on ICD-9 coding
# create vectors of disease codes
                mi  <- c(410,412)
                chf <- c(428)
                pvd <- c(4439,7854)
                cvd <- c(430:438)
                dem <- c(290)
                cpd <- c(490:496,500,501)
                rhm <- c(7100,7101,7104,7140:7142)
                pep <- c(531:534)
                mld <- c(5712,5714:5716)
                dnc <- c(2500:2503,2507)
                dwc <- c(2504:2506)
                ple <- c(342)
                ren <- c(5830:5832,5834,5836,5837)
```

```
        can <- c(172)
        liv <- c(5722:5724,5728)
        met <- c(196:199)
        hiv <- c(42:44)
# check for each disease category, assign corresponding disease classification to ICD-9 code
        for (i in 1:nobs) {
         for (j in 1:length(mi)) {
         if (d[i,2] == mi[j]) {
          d$code[i] <- "ch1"}
         }
         for (j in 1:length(chf)) {
         if (d[i,2] == chf[j]) {
          d$code[i] <- "ch2"}
         }
         for (j in 1:length(pvd)) {
         if (d[i,2] == pvd[j]) {
          d$code[i] <- "ch3"}
         }
        … do this for each of the 17 disease categories
         for (j in 1:length(hiv)) {
         if (d[i,2] == hiv[j]) {
          d$code[i] <- "ch17"}
         }
```

```
}
# Select case number and disease class
#Need to add this to the end of dvars so all codes will be there
ch <- c("ch1",  "ch2",  “ch3", "ch4", "ch5", "ch6", "ch7", "ch8",
"ch9", "ch10", "ch11", "ch12", "ch13", "ch14", "ch15", "ch16",
"ch17")
k <- data.frame(cbind(x="xxxREMOVEXXX", y=ch, z=ch))
# renaming vars in k so i can do a row merge
names(k) <- names(d)
d_vars <- rbind(d,k)
# use melt and cast functions from the reshape package to transpose the data from rows to
columns
mydata <- melt(d_vars, id=c("Sequence","code"))
mydata$value <-mydata$code
mydata2<-cast(mydata, Sequence~code)
# Then replace >1 with 1, to code as present/absent
mydata2$ch1  <-ifelse(mydata2$ch1> 0, 1,0)
mydata2$ch2  <-ifelse(mydata2$ch2> 0, 1,0)
mydata2$ch3  <-ifelse(mydata2$ch3> 0, 1,0)
mydata2$ch4  <-ifelse(mydata2$ch4> 0, 1,0)
mydata2$ch5  <-ifelse(mydata2$ch5> 0, 1,0)
mydata2$ch6  <-ifelse(mydata2$ch6> 0, 1,0)
mydata2$ch7  <-ifelse(mydata2$ch7> 0, 1,0)
```

```
                mydata2$ch8  <-ifelse(mydata2$ch8> 0, 1,0)
                mydata2$ch9  <-ifelse(mydata2$ch9> 0, 1,0)
                mydata2$ch10<-ifelse(mydata2$ch10> 0, 1,0)
                mydata2$ch11<-ifelse(mydata2$ch11> 0, 1,0)
                mydata2$ch12<-ifelse(mydata2$ch12> 0, 1,0)
                mydata2$ch13<-ifelse(mydata2$ch13> 0, 1,0)
                mydata2$ch14<-ifelse(mydata2$ch14> 0, 1,0)
                mydata2$ch15<-ifelse(mydata2$ch15> 0, 1,0)
                mydata2$ch16<-ifelse(mydata2$ch16> 0, 1,0)
                mydata2$ch17<-ifelse(mydata2$ch17> 0, 1,0)
#get weights - the last 6 comorbidities have increased weights assigned to them
# get a count of comorbidities and the Charlson Index score
                for (i in 1:nrow(mydata2)) {
                mydata2$comorbidity.n[i]<-sum(mydata2$ch1[i], mydata2$ch2[i],
                mydata2$ch3[i], mydata2$ch4[i], mydata2$ch5[i],
                mydata2$ch6[i], mydata2$ch7[i], mydata2$ch8[i],
                mydata2$ch9[i], mydata2$ch10[i],
                mydata2$ch11[i], mydata2$ch12[i], mydata2$ch13[i],
                mydata2$ch14[i], mydata2$ch15[i],   mydata2$ch16[i],
                mydata2$ch17[i])

                mydata2$charlson[i]<-sum(mydata2$ch1[i], mydata2$ch2[i],
                mydata2$ch3[i], mydata2$ch4[i], mydata2$ch5[i],
```

```
mydata2$ch6[i], mydata2$ch7[i], mydata2$ch8[i],
mydata2$ch9[i], mydata2$ch10[i],
mydata2$ch11[i], 2*mydata2$ch12[i], 2*mydata2$ch13[i],
2*mydata2$ch14[i], 3*mydata2$ch15[i], 6*mydata2$ch16[i],
6*mydata2$ch17[i])
 }
# dropping the extra row, the last row with a 1 for each category so
all would be counted
mydata3 <- mydata2[-nrow(mydata2),]
# Output of the function is the modified dataset with the summary
scores
return(mydata3)
}
#  Sample call of function
#  charlson(full.data$Sequence,full.data$icd9)
```

Conclusion

In this project, an R package was created to accomplish two things: 1) recode ICD-9 entries into categories useful for computing the Charlson Index, and 2) compute the Charlson index for entries in a dataset. The R package was implemented without error or warnings and resulted in successful computation of the Charlson Index.

This R package will be useful to any programmer with data containing ICD-9 disease codes. In addition, it would be easy to incorporate more ICD-9 codes into the package to develop other coding algorithms based on these codes for other disease indices. Coding algorithms based on ICD-9 currently exist to stratify severity of Crohn's Disease, hospital mortality in pediatric patients, and many other diseases. Future applications would include adding the classifications for ICD-10 codes to give the user the choice of choosing ICD-9 or ICD-10 codes.

References:

Charlson ME, Pompei P, Ales KL, MacKenzie CR (1987). A new method of classifying prognostic comorbidity in longitudinal studies: development and validation. J Chron Dis, 40(5): 373-383.

Comorbidity [Internet]. Los Angeles (CA): Wikipedia, The Free Encyclopedia; [updated 2010 Mar 27; cited 2010 May 5]. Available from: http://en.wikipedia.org/wiki/Comorbidity

Greenfield S, Apolone G, McNeil BJ, Cleary PD: The importance of co-existent disease in the occurrence of postoperative complications and one-year recovery in patients undergoing total hip replacement. Comorbidity and outcomes after hip replacement. *Med Care* 1993, 31:141-154.

de Groot V, Beckerman H, Lankhorst GJ, Bouter LM: How to measure comorbidity. a critical review of available methods. *J Clin Epidemiol* 2003, 56:221-229.

ICD-9 [Internet]. Los Angeles (CA): Wikipedia, The Free Encyclopedia; [updated 2010 Mar 27; cited 2010 May 5]. Available from: http://en.wikipedia.org/wiki/ICD9

Kaplan MH, Feinstein AR: The importance of classifying initial co-morbidity in evaluation the outcome of diabetes mellitus. *J Chronic Dis* 1974, 27:387-404.

Miller MD, Paradis CF, Houck PR, Mazumdar S, Stack JA, Rifai AH, Mulsant B, Reynolds CF III: Rating chronic medical illness burden in geropsychiatric practice and research: application of the Cumulative Illness Rating Scale. *Psychiatry Res* 1992, 41:237-248.

Murray SB, Bates DW, Ngo L, Ufberg JW, Shapiro NI. Charlson Index is associated with one-year mortality in emergency department patients with suspected infection. Acad Emerg Med. 2006 May;13(5):530-6. Epub 2006 Mar 21.

Quan H, Sundararajan V, Halfon P, et al. Coding algorithms for defining Comorbidities in ICD-9-CM and ICD-10 administrative data. Med Care. 2005 Nov; 43(11): 1130-9.

Von KM, Wagner EH, Saunders K: A chronic disease score from automated pharmacy data. *J Clin Epidemiol* 1992, 45:197-203.

Appendix 1 – Example input data

YearDeath	Sequence	ICD9
83	5157	2
86	18061	2
90	17986	2
86	17817	2
77	19306	2
84	9408	1
76	15659	1
79	16887	1
80	3946	410
80	3946	428
80	3946	3
80	3946	7100
80	3946	3
80	3946	3
80	3946	3
80	3946	3
80	3946	5830
80	3946	3
80	3946	3
80	3946	199
80	3946	3
80	3946	2505
80	3946	3
80	3946	3
80	3946	2504
84	107	410
84	107	428
84	107	4439
84	107	438
84	107	290
84	107	490
84	107	7100
84	107	531
84	107	5712
84	107	2500
84	107	342
84	107	5830
84	107	172
84	107	5722
81	2229	196
81	2229	42
81	2229	847
81	2229	673
81	2229	1112
81	2229	428
81	2229	4439
81	2229	438