Paper 3485-2015

Health Services Research Utilizing Electronic Health Record Data: A Grad Student 'How-To' Paper

Ashley W. Collinsworth, ScD, MPH, Baylor Scott & White Health and Tulane University School of Public Health and Tropical Medicine; Elisa L. Priest, DrPH, Baylor Scott & White Health

ABSTRACT

Graduate students encounter many challenges when conducting health services research utilizing "real world" data obtained from electronic health records (EHRs). These challenges include cleaning and sorting data, summarizing and identifying present on admission diagnosis codes, identifying appropriate metrics for risk-adjustment, and determining the effectiveness and cost effectiveness of treatments. This paper provides graduate students with basic tools for the conduct of health services research with EHR data. We will examine SAS tools and step-by-step approaches used in an analysis of the effectiveness and cost-effectiveness of the ABCDE (Awakening and Breathing Coordination, Delirium monitoring/management, and Early exercise/mobility) bundle in improving outcomes for intensive care unit (ICU) patients. These tools include: 1) ARRAYS, 2) Table Look-up, 3) LAG functions, 4) PROC TABULATE, 5) recycled predictions, and 6) bootstrapping. We will discuss challenges and lessons learned in working with data obtained from the EHR. This content is appropriate for beginning SAS users.

INTRODUCTION

Electronic Health Records (EHRs) are being adopted rapidly by health care delivery organizations in the United States and have the potential to streamline and improve the delivery of health care. In addition, the increased use of EHRs is expanding opportunities for the conduct of health services research, allowing researchers to utilize data collected through routine care instead of resource-intensive methods such as clinical trials. Although EHRs can be a valuable data source, there are often many challenges associated with using these data to conduct meaningful research. These challenges include cleaning and sorting data, summarizing and identifying present on admission diagnosis codes, identifying appropriate metrics for risk-adjustment, and determining the effectiveness and cost effectiveness of treatments.

This paper provides graduate students with basic tools for the conduct of health services research with EHR data. We will examine SAS tools and step-by-step approaches used in an analysis of the effectiveness and cost-effectiveness of the ABCDE (Awakening and Breathing Coordination, Delirium monitoring/management, and Early exercise/mobility) bundle in improving outcomes for intensive care unit (ICU) patients. These tools include: 1) ARRAYS, 2) Table Look-up, 3) LAG functions, 4) PROC TABULATE, 5) recycled predictions, and 6) bootstrapping. We will discuss challenges and lessons learned in working with data obtained from the EHR. This content is appropriate for beginning SAS users.

DATASETS

The study dataset contained patient demographics, clinical outcomes, and cost data for patients treated in 12 Baylor Scott & White Health (BSWH) ICUs and was created by merging data from the EHR (AllScripts) and BSWH administrative and financial (Trendstar) datasets.

TOPICS

1. ARRAYS
2. LAG functions
3. Table Look-up
4. PROC TABULATE
5. Recycled Predictions

6. Bootstrapping

1. ARRAYS

We needed to determine which patients in our study dataset had delirium or dementia that was present on admission (POA) based on ICD-9 codes in order to control for baseline mental status in our risk-adjusted models. Up to 25 ICD-9 diagnoses codes are collected for each patient and were indicated by variables DX1-DX25 in the study dataset. A separate variable POA1-POA25 indicated whether or not each diagnosis was POA. Creating one variable to indicate if the patient had POA dementia or delirium required us to determine if there was a diagnosis code for delirium/dementia in any 1 of the 25 DX variables and if the corresponding POA variable was "Y" indicating a POA diagnosis. Using an ARRAY allowed us to perform this function across all 25 DX and corresponding POA variables with a minimal amount of coding.

Step 1: Identify ICD-9 codes that indicate diagnosis of delirium or dementia.

Step 2: Create an ARRAY.

```
DATA work.ibid_array;
Set work.ibid_drop;

*** Set array to process the 25 DX codes and the 25 POA codes ***;
ARRAY DXNEW (25) $ DX1 - DX25;
ARRAY POANEW (25) $ POA1 - POA25;

*** Do loop processing for variables 1-25 ***;
    Do i = 1 to 25;

*** Determine if any of the diagnosis codes for delirium and dementia are
present and if corresponding POA variable is "Y" indicating POA.  Create a
variable Delirium coded 1 to indicate POA delirium/dementia and 0 to indicate
the absence of POA delirium/dementia ***;

      IF DXNEW(i)IN
("290.0","290.10","290.11","290.12","290.13","290.20","290.21","290.3","290.4
0","290.41","290.42","290.43", "291.0", "291.2", "292.81", "292.82", "293.0",
"293.1","294.10","294.11","294.20","294.21","331.19","331.82")

 AND POANEW(i)="Y" THEN Delirium = 1;
      IF Dementia = . then Delirium = 0;
END;
run;
```

2. LAG FUNCTIONS

We needed to identify which patients in our dataset had multiple admissions. In our dataset, each admission for each patient was indicated by a separate row of data. Unique patients could be identified by medical record number (MRN), and unique admissions could be identified by admission date. We had to use a LAG Function to create the variable AdmitNum to count the number of records per MRN. Since we only wanted to include each patient's first visit in our analysis, we then deleted records with AdmitNum > 1.

Step 1: Sort dataset by medical record number and admission date.

Step 2: Use a LAG Function to create a variable (AdmitNum) that indicates visit number.

```
proc sort data = work.ibid_clean4;
by MRN AdmitDtm;
run;

data work.ibid_repeat;
set work.ibid_clean4;
by MRN admitDtm;
retain AdmitNum;
if first.MRN AND first.admitDTM then AdmitNum=0;
AdmitNum=AdmitNum+1;
run;
```

Used LAG Function to add AdmitNum as a variable indicating number of records per MRN

Study Dataset

	MRN	AdmitDtm	Name	Delirium	Expired	AdmitNum
1	001	2/3/14	Smith	0	0	1
2	002	6/20/14	Johnson	1	0	1
3	002	8/8/14	Johnson	1	1	2
4	003	3/4/14	Green	0	0	1
5	003	4/16/14	Green	1	0	2
6	003	12/1/14	Green	0	0	3

Figure 1. Using LAG Function to Count Admissions

3. TABLE LOOK-UP

For the cost-effectiveness analysis, we calculated potential patient life years saved by estimating the number of life years lost for each person who died in the study. The study dataset contained birthdates and inpatient mortality data. Life expectancy was projected based on the age and sex of the patient using the Social Security Administration's actuarial life tables for 2010[1], discounted based on the five-year survival for patients discharged from ICUs compared to the general population.[2] The number of life years saved was calculated as the difference in projected life expectancy and the age of the patient at the time of death.

Step 1: Export SSA Period Life Table (Figure 2) into Excel.

Step 2: Restructure data in Excel into 3 columns (Figure 3): Gender, Age, and Life Expectancy.

Step 3: Import Excel file as SAS dataset.

Step 4: Perform Table Look-up in SAS to add Life Expectancy variable (LifeEx) to study dataset.

Step 5: Calculate total and average life years lost for study population.

Period Life Table, 2010						
	Male			Female		
Exact age	Death probability	Number of lives	Life expectancy	Death probability	Number of lives	Life expectancy
70	0.024139	73,355	14.07	0.016233	82,740	16.33
71	0.026364	71,584	13.40	0.017882	81,397	15.59

Life expectancy = average remaining number of years expected prior to death for a person at that exact age

Figure 2. Social Security Administration Period Life Table, 2010

Excel and SAS Life Expectancy Dataset Created from Period Life Table

	Male	pat_age	LifeEx
1	1	70	14.07
2	1	71	13.40
3	0	70	16.33
4	0	71	15.59

Add LifeEx Variable

Study Dataset

	Male	pat_age	Intervention	Delirium	Expired
1	1	70	1	0	0
2	1	71	0	1	1
3	0	70	0	1	0
4	0	71	1	0	0

Figure 3. Create SAS Life Expectancy Dataset to Merge with Study Dataset

```
*** Code for Table Look-up ***;
*** Sort study dataset by gender and age ***;
proc sort data = work.ibid;
by male pat_age;
run;

*** Sort life expectancy dataset by gender and age ***;
proc sort data = ibid.LifeEx ;
by male pat_age;
run;

*** Merge datasets ***;
data work.ibid_Merge;
merge work.ibid (in=a) ibid.LifeEx (in=b) ;
if a;
by male pat_age;
run;

*** Code to Calculate Life Years Lost ***;
data work.ibid_Final1;
set work.ibid_Merge;
if expired = "1" then YearsLost = LifeEx;
else YearsLost = "." ;
```

```
*** Discount life expectancy based on ICU survival rates ***;
DiscountYears = YearsLost * .667;
run;

*** Calculate average life years lost ***;
proc means data = work.ibid_Final1;
var DiscountYears;
run;
```

4. PROC TABULATE

We wanted to examine ABCDE bundle adherence rates (total_abcde_rate) for ICUs that received a basic intervention (intervention = 0) versus an enhanced intervention (intervention = 1) over a 12 month period. Using **PROC TABULATE**, we were able to calculate the bundle adherence rate for each group as well as both groups combined for each month.

```
*** Display Bundle Adherence BY Intervention Group and Month ***;
proc tabulate data = work.ibid_FINAL1;
class intervention month;
VAR total_abcde_rate;
table month ALL, MEAN*total_abcde_rate *(intervention ALL) ;
run;
```

	Mean total_abcde_RATE		
	Intervention		All
	0	1	
month			
1	0.12	0.26	0.25
2	0.14	0.30	0.26
3	0.17	0.35	0.30
4	0.16	0.30	0.27
5	0.16	0.35	0.30
6	0.19	0.34	0.31
7	0.15	0.38	0.29
8	0.19	0.37	0.33
9	0.40	0.46	0.45
10	0.38	0.45	0.43
11	0.40	0.48	0.47
12	0.37	0.46	0.43

Figure 4. Output Generated by PROC TABULATE

5. RECYCLED PREDICTIONS

For the cost-effectiveness analysis we wanted to estimate the effect of high versus low bundle adherence on inpatient mortality and inpatient costs. We used recycled predictions to account for and balance

baseline differences between patients who had high and low levels of bundle adherence. Specifically, mortality and costs were predicted from the modeled equations based on two scenarios: 1) every patient had high bundle adherence (>60%) (ABCDE_compliance = 1) and 2) every patient had low bundle adherence (<60%) (ABCDE_compliance = 0). The difference between these two predictions constituted the predicted mean differences in in-hospital mortality and cost. Using recycled predictions allowed us to avoid biases in the cost estimates that occur during retransformation when using multiplicative models such as OLS regression.

Step 1: Create 2 new datasets (Grp1, Grp2) from study data, setting the independent variable of interest (ABCDE_compliance) to "0" for all patients in one dataset and to "1" for all patients in the second dataset and creating the variable "Group" that = "Low_Adherence" in Grp1 and "High Adherence" in Grp2.

Step 2: Set dependent variables of interest (mortality "expired" and cost "adj_direct_cost") to missing "."

Step 3: Create new dataset (recycled) including original dataset and two new datasets.

Step 4: Run regression model for cost using combined datasets and output new dataset containing predicted values of y (yhat) for each observation where group = Low_Adherence or High_Adherence.

Step 5: Output the mean yhat for each group to new dataset.

Step 6: Transpose dataset to facilitate bootstrapping (Figure 5).

Step 7: Repeat Steps 4-6 with regression model for mortality.

```
*** CREATE DATASETS FOR RECYCLED PREDICTIONS ***;
data work.Grp1;
set study_data;
ABCDE_compliance = 0;
adj_direct_cost = . ;
expired = .;
Group = 'Low_Adherence';
run;

data work.Grp2;
set study_data;
ABCDE_compliance = 1;
adj_direct_cost = . ;
expired = .;
Group = 'High_Adherence';
run;
data recycled ;
      length Group $20.;
      set study_data Grp1 Grp2;
      run;


***COST REGRESSION MODEL***;
ods listing close;
proc genmod data = recycled desc;
model adj_direct_cost = ABCDE_compliance quintile / dist = gamma link = log;
output out = direct_predcost(where = (Group ne'')) p=yhat;
```

```
run;

***Output Mean yhat for High and Low Adherence Groups***;
ods listing;
proc means data = direct_predcost noprint;
      class group;
      var yhat;
      output out = direct_sampstat(where = (group ne '')) n = mean=/autoname;

*** TRANSPOSE DATASET ***;
proc transpose data = direct_sampstat out = transpose_sampstat;
      var yhat_mean;
      id group;
run;
```

	NAME	_LABEL_	High_Adherence	Low_Adherence
1	Yhat_Mean	Estimated Probability	0.0756514063	0.1938411404

Figure 5. Output for Mortality Model

	NAME	_LABEL_	High_Adherence	Low_Adherence
1	Yhat_Mean	Estimated Probability	33042.140263	30635.945975

Figure 6. Output for Cost Model

6. BOOTSTRAPPING

Generating Confidence Intervals for Statistics Obtained through Recycled Predictions:

We generated 1,000 bootstraps of our recycled predictions to estimate (1) the mean differences in in-hospital mortality and costs and (2) the standard errors of these statistics. This approach permitted the estimation of incremental cost-effectiveness ratios (ICERs) and surrounding confidence intervals.

Step 1: Create Macro to incorporate with Recycled Predictions code (previous example) that generates 10 sets of 100 samples from the study dataset.

Step 2: Add "by replicate" to PROC MEANS to calculate statistics by each replicate (1-100) group.

Step 3: Append statistics for each replicate to 1 dataset (Figure 7).

Step 4: Use PROC TTEST to obtain the mean and standard error for mortality and cost for the 100 replicates. Fieller's method can then be used to generate confidence intervals around the incremental cost effectiveness ratio (ICER).

```
*** BOOTSTRAP MACRO ***
loop= number of times entire macro loop runs; rep= number of replicates of
the big dataset;

total number of simulations= loop x rep ;
```

```
*** Run this code first to activate the macro ***;
%macro simulation
(loop,rep,seed_1,seed_2,seed_3,seed_4,seed_5,seed_6,seed_7,seed_8,seed_9,seed
_10);
%do i=1 %to &loop.;


*** Use PROC SURVEYSELECT to Generate Samples ***;
*** We used a replicate function to first create a dataset of 1000
observations (consisting of 100 samples per macro loop). We then ran the
analysis for each replication (1-10). Creating the dataset first and then
running the analyses is much more efficient than creating and analyzing 1000
separate datasets.***;

proc surveyselect data=work.ibid_final1 out=work.ibid_simulation noprint
outhits
            seed=&&seed_&i.
            method=urs
            samprate=1
            rep=&rep.;
            strata unit;
      run;
%end;
 %mend simulation;

*** Call the Macro and generate seed numbers so that you pull the same
samples each time you run the code if you want to replicate the analysis ***;

%simulation (loop=10,rep=100,seed_1=12323, seed_2=32480, seed_4=97898,
seed_5=98432, seed_6=79978 seed_7=433903, seed_8=423982, seed_9=838302,
seed_10=38373);


*** Insert “by replicate” into Recycled Predictions code to Output Mean yhat
for High and Low Adherence Groups ***;
ods listing;
proc means data = direct_predcost noprint;
      by replicate;
      class group;
      var yhat;
      output out = direct_sampstat(where = (group ne '')) n = mean=/autoname;
run;

*** Append statistics from all replicates into 1 dataset ***;
data work.direct_simulation_&i.;
set work.direct_tr_sampstat;
simcount=&i.;
run;

proc datasets library=work force noprint;
      append base=work.direct_simulation data=work.direct_simulation_&i.;
      run;
      quit;
```

Replicate	_NAME_	_LABEL_	High_Adherence	Low_Adherence
1	Yhat_Mean	Estimated Probability	0.0852606852	0.2165944651
2	Yhat_Mean	Estimated Probability	0.0651169996	0.2246684623
.				
100	Yhat_Mean	Estimated Probability	0.0760712966	0.2103272653

Figure 7. Combined Dataset

```
*** Obtain mean and standard error for 100 replicates ***;
proc ttest data = ibid.AppendedDataset;
var High_Adherence Low_Adherence;
run;
```

CONCLUSION

EHRs contain a wealth of data that can be used for meaningful health services research and allow for data to be collected through the provision of routine care instead of costly and time consuming clinical trials. However, researchers may need to employ a number of additional SAS tools to create datasets and perform meaningful analyses using these data.

REFERENCES

1. U.S. Social Security Administration. Period Life Table, 2010, from http://www.ssa.gov/oact/index.html
2. Wright, J. C., Plenderleith, L., & Ridley, S. A. (2003). Long-term survival following intensive care: subgroup analysis and comparison with the general population. *Anaesthesia, 58*(7), 637-642.

ACKNOWLEDGEMENTS

This project was supported by a grant (R18HS021459) from the Agency for Healthcare Research and Quality (AHRQ).

CONTACT INFORMATION

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

Ashley W. Collinsworth
Ashley.Collinsworth@baylorhealth.edu