

Calculating the Hospital Readmission Interval

Craig Dickstein
Tamarack Professional Services, LLC

Boston Area SAS Users Group
September 2012

The Business Case

- Re-admission rates are a hot topic in the industry.
 - performance measures
 - MCOs
 - hospitals
 - Medicare reimbursement
 - healthcare reform
- Why is it happening?
 - how do you define a re-admission?

2

The Technical Challenge

- Data architecture needs to support the requisite processing.
- How do you look ahead for a subsequent event?
 - SAS does *not* have a *look-ahead* function.
- Do you have the data elements necessary for decision making?

3

Caveats

- This material only looks at the re-admission interval calculation.
 - the data architecture challenge
- Other business/data challenges to consider:
 - is there a clear definition of 're-admission'?
 - where did the re-admission occur? does it matter?
 - what was the medical event that caused the new admission? does it matter?
 - do co-morbid conditions matter?

4

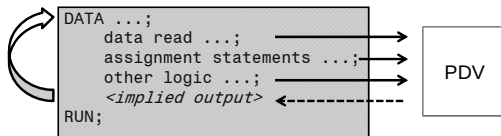
Solutions

- The LAG function
 - needs a reverse sort
 - can be problematic with conditional logic
- Multiple SET statements with the POINT= option
 - 'point' to the N+1 observation
 - not as intuitive
- Use hiter and hash objects to “walk” through the data looking for the next matching key.
 - not for the faint-of-heart
- Co-locate the elements of two records in the program data vector (PDV) simultaneously

5

The Program Data Vector (PDV)

- Logical area in memory where SAS builds a dataset observation, one at a time
 - defined at compilation time
- Data values are assigned to the PDV
 - at execution time
- At end of each data step iteration, PDV values written to a SAS dataset as a single observation



6

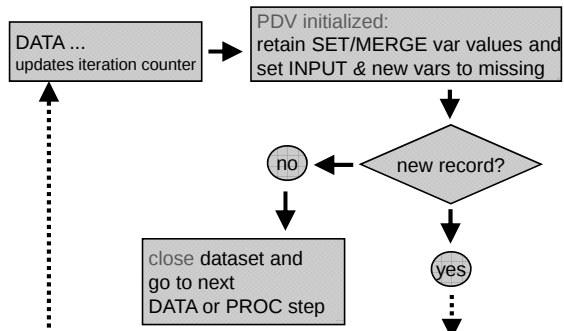
The Program Data Vector (PDV)

- Contains all variables defined by the user
 - input variables
 - computed variables
- Contains SAS temporary variables
 - `_N_` – count of times the DATA step begins to iterate
 - `_ERROR_` – data error?
 - `FIRST.var` – first occurrence of the BY group?
 - `LAST.var` – last occurrence of the BY group?
 - `IN=var` – contribution from a dataset?
 - `END=var` – end of the input stream?

true or false (1,0)

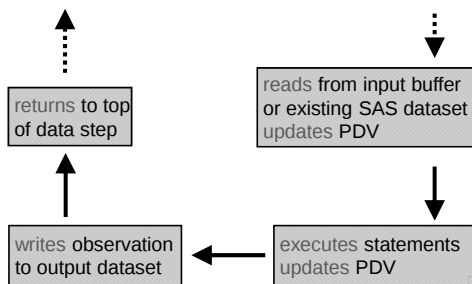
7

The Execution Phase



8

The Execution Phase (cont.)



9

The PDV – an Example

```

DATA professional;
  SET fred.prof_clms_2007
    (IN=alpha KEEP=mbr_id dos  OBS=3)
    fred.prof_clms_2008
    (IN=beta  KEEP=mbr_id dos  OBS=3)
  ;
  BY mbr_id dos ;
  IF alpha = 1 THEN year = "2007";
  IF beta  = 1 THEN year = "2008";

```

what does SET / BY accomplish for us?

10

The Two Source Datasets

Dataset is fred.prof_clms_2007		
Obs	Mbr_ID	DOS
1	0000871	10/19/2007
2	0001827	07/07/2007
3	0001827	11/27/2007

Dataset is fred.prof_clms_2008		
Obs	Mbr_ID	DOS
1	0000871	05/10/2008
2	0000871	05/17/2008
3	0000871	05/17/2008

11

Visual Representation of the PDV

alpha	beta	mbr_id	dos	FIRST. mbr_id	LAST. mbr_id	year	_ERROR_	_N_	
0	0			1	1		0	1	top of data step bottom of data step
1	0	0000871	10/19/2007	1	0	2007	0	1	
1	0	0000871	10/19/2007	1	0		0	2	
0	1	0000871	5/10/2008	0	0	2008	0	2	
0	1	0000871	5/10/2008	0	0		0	3	
0	1	0000871	5/17/2008	0	0	2008	0	3	
0	1	0000871	5/17/2008	0	0		0	4	
0	1	0000871	5/17/2008	0	1	2008	0	4	
0	1	0000871	5/17/2008	0	1		0	5	
1	0	0001827	7/7/2007	1	0	2007	0	5	
1	0	0001827	7/7/2007	1	0		0	6	
1	0	0001827	11/27/2007	0	1	2007	0	6	
1	0	0001827	11/27/2007	0	1		0	7	

12

Rules of Retention

Operation	PDV Element Set to Missing
SET and MERGE - Existing variables - New variables	NO YES
ERROR, _N_, IN=var	NO
SUM Statement Variable	NO
INPUT/INFILE	YES
RETAIN Statement Variables	NO

13

Hospital re-admission

Goal: Have the requisite elements of re-admit calculations in the PDV.

PDV

Mbr_ID	Admit_DT	Dschg_DT	Next_Admit_DT	D2A_Interval
				<i>Calculation of interest</i>

14



Defining Data Structures and Generating re-admission Calculations

This demonstration illustrates the following:

- Defining inherent data structures to support project design
- Generating re-admission calculations

15

Presenter Contact Information

Craig Dickstein

Tamarack Professional Services, LLC
P.O. Box 128, Caratunk, ME 04925-0128
FAX: by request – Voice: (207) 672-4989

tamarack.llc@myfairpoint.net

16

```

/* ***** */
|      SET UP JOB PARAMETERS      |
/* ***** */

%let rootdir = c:\documents and settings\craig dickstein\my documents;
%let datadir = &rootdir\SAS Institute\BHCDW\Data;
%let codedir = &rootdir\SAS Institute\BHCDW Institute\Source Code;

%let work      = work;

*** A BUSINESS EXAMPLE ***** hedis measure *****
If the discharge is followed by a readmission or direct transfer to an acute or nonacute facility
within the 30-day follow-up period, count only the readmission discharge or the discharge from the
facility to which the member was transferred.

Exclude both the initial discharge and the readmission/direct transfer discharge if the
readmission/direct transfer discharge occurs after December 1 of the measurement year.
*****;

data &work..IP_Claims;
  infile cards firstobs=2;
  input  mbr_id $ @7 Admit_DT mmddyy8. @16 Dschg_DT mmddyy8. keep $ rec_nbr $;
  format Admit_DT Dschg_DT mmddyy8.;
  cards;
123456789012345678901234567890
11111 1/1/08 1/6/08 N 1
11111 1/8/08 1/12/08 Y 2
11111 3/1/08 3/6/08 Y 3
11111 5/1/08 5/6/08 N 4
11111 5/8/08 5/20/08 N 5
11111 5/21/08 5/26/08 Y 6
11111 11/21/08 11/26/08 N 7
11111 11/28/08 12/26/08 Y 8
22222 3/3/08 3/9/08 Y 1
22222 5/8/08 5/16/08 N 2
22222 5/18/08 5/26/08 N 3
22222 5/29/08 6/6/08 Y 4
22222 8/8/08 9/2/08 Y 5
run;

proc sort data=&work..IP_Claims ;
  by mbr_id Admit_DT Dschg_DT;
run;

*options mergenoby=warn;

data &work..ip_dischgs_01;
  put 'BEFORE:' / _all_;
  merge &work..IP_Claims(drop=keep rec_nbr)
        &work..IP_Claims(firstobs=2
                          keep=mbr_id Admit_DT
                          rename=(mbr_id=next_mbr_id
                                Admit_DT=next_Admit_DT)
        );

  *****
  Notice the MERGE without a BY statement. Generally not a good thing, but we are merging the
  dataset on itself - offset by one record.
  This technique allows us to look ahead one record to effect "contiguous record" logic checks.
  *****;

  if next_Admit_DT > (Dschg_DT + 30) then wanted = 1;
  if mbr_id ne next_mbr_id
    then do;
    wanted = 1;
    next_Admit_DT = .;
  end;
  interval = next_Admit_DT - Dschg_DT ;
  * if wanted;
  * if wanted * (dschg_dt <= "01dec2008"d);
  put 'AFTER:' / _all_;
run;

```

```
*****
Documentation of MERGENOBY system option
*****;
```

```
proc options option=mergenoby value define;
run;
```

```
*****;
*** AN ALTERNATE SOLUTION *****;
*****;

*** ----- ***;
*** GENERATE ADMISSION DATA ***;
*** ----- ***;

run;
data &work..gen_claims(drop=i);
  attrib pat_id format=z2. /* define physical position on the dataset */
        admit_dt format=date9.
        dschg_dt format=date9.
  ;
  do pat_id = 1 to 9; /* generate nine patients */
    do i = 1 to 300; /* generate 300 claims per patient
                     uniformly distributed in CY 2010 */

      admit_dt = int(("31dec2010"d - "01jan2010"d)*ranuni(6) + "01jan2010"d);
      dschg_dt = int(("31dec2010"d - "01jan2010"d)*ranuni(8) + "01jan2010"d);

      if admit_dt < dschg_dt then output; /* reasonable data following RANUNI
                                          might there still be issues */
    end;
  end;
run;

proc sort data=&work..gen_claims; /* chronological arrangement */
  by pat_id dschg_dt;
run;

data &work..ip_dischgs_02; /* further refine to good test data */
  attrib pat_id format=z2. admit_dt format=date9.
        dschg_dt format=date9. prior_dschg_dt format=date9.
  ;
  prior_dschg_dt = dschg_dt; /* key design element - prior to SET */
                             /* no longer relevant */
  set &work..gen_claims;
  by pat_id dschg_dt ;
  if first.pat_id
    then prior_dschg_dt = .;
  if prior_dschg_dt < admit_dt /* reasonable data following RANUNI */
    then output;
run;

*** ----- ***;
*** CALCULATE RE-ADMISSIONS ***;
*** ----- ***;

data &work..readmits;
  attrib pat_id format=z2. admit_dt format=date9.
        dschg_dt format=date9. prior_dschg_dt format=date9.
  ;
  prior_dschg_dt = dschg_dt; /* key design element - prior to SET */

  set &work..ip_dischgs_02;
  by pat_id dschg_dt ;

  los = max(dschg_dt - admit_dt,1);
  days2readm = admit_dt - prior_dschg_dt;

  output readmits; /* put obs to target dataset */

  if last.pat_id
    then dschg_dt = . ; /* reset PDV element for next iteration */

  _error_ = 1; /* force dump of PDV for inspection in log */
run;
```