

SAS® Programming Methods to Identify Episodes of Hospital Readmissions

Jennifer Cano, VA Center for Clinical Management Research

Cainnear K. Hogan, VA Center for Clinical Management Research

Taylor N. Whittington, VA Center for Clinical Management Research

Julien Weinstein, VA Center for Clinical Management Research;

Department of Internal Medicine, University of Michigan Medical School

Scott J. Denstaedt, Department of Internal Medicine, University of Michigan Medical School

Sarah M. Seelye, VA Center for Clinical Management Research

Hallie C. Prescott, VA Center for Clinical Management Research;

Department of Internal Medicine, University of Michigan Medical School

ABSTRACT

Cause-specific hospital readmissions (e.g., readmission for infection, readmission for cardiovascular disease) are an important outcome in many clinical studies. However, identification of cause-specific readmissions requires complex coding and consideration of longitudinal hospital encounter data. Using SAS® programming tools to facilitate data manipulation, we describe a concise method for identifying cause-specific readmissions that can be adapted to identify readmissions for different causes and over different time horizons.

INTRODUCTION

Hospital readmissions are an important metric in health care delivery quality and contribute to increased healthcare costs. Assessing the specific causes of readmissions may provide valuable insight into how to prevent and reduce readmissions. Cause-specific hospital readmissions, such as readmission for infection or readmission for cardiovascular disease are an important outcome in many clinical studies.

Readmissions are frequently studied in the Veterans Affairs (VA) health care system. A key source of electronic health record (EHR) data in the VA is the Corporate Data Warehouse (CDW), a national relational database. EHR data often need to be cleaned and restructured before they can be analyzed. Defining cause-specific readmissions can be challenging due to the longitudinal structure of the data and particularly when multiple readmissions need to be accounted for.

The following SAS® programming tools allow for easier data manipulation of hospitalization data:

- Numeric and character functions
- PROC SQL
- Macros
- DO Loops
- FIRST. variable
- Arrays

We describe a VA hospitalization data set and explain how to identify hospitalizations that were followed by all-cause and cause-specific 90-day readmissions by using these SAS programming tools. Although this paper focuses on clinical data, the same methods can be applied to longitudinal data in other fields.

DATA SET STRUCTURE

Using CDW data, the Veterans Affairs Patient Database (VAPD) identifies acute hospitalizations within the VA (Wang et al., 2019). The VAPD is structured at a patient-facility-day level, meaning every row represents an individual calendar date that a patient spent in a particular hospital during their hospitalization (Vincent, 2018; Wang et al., 2019). Patients may have multiple hospitalizations (i.e., readmissions) and numerous rows per hospitalization depending on their length of stay, as demonstrated by patient 002 in Figure 1 (not real patient data). Each patient (PatientID), facility (FacilityID), and hospitalization (UniqueHospID) are assigned their own unique identifier.

Obs	PatientID	FacilityID	Date	UniqueHospID	HospitalDay	AdmitDate	DischargeDate
1	001	222	06/11/2018	4567	1	06/11/2018	06/13/2018
2	001	222	06/12/2018	4567	2	06/11/2018	06/13/2018
3	001	222	06/13/2018	4567	3	06/11/2018	06/13/2018
4	002	123	01/07/2020	1234	1	01/07/2020	01/08/2020
5	002	123	01/08/2020	1234	2	01/07/2020	01/08/2020
6	002	444	02/20/2020	5678	1	02/20/2020	02/21/2020
7	002	444	02/21/2020	5678	2	02/20/2020	02/21/2020
8	002	321	03/01/2020	6543	1	03/01/2020	03/02/2020
9	002	321	03/02/2020	6543	2	03/01/2020	03/02/2020
10	003	333	10/26/2019	2468	1	10/26/2019	10/27/2019
11	003	333	10/27/2019	2468	2	10/26/2019	10/27/2019
12	003	333	01/30/2020	3897	1	03/30/2020	03/30/2020

Figure 1. VAPD Data Set Structure Using Hypothetical Data

All-cause readmissions are defined as any readmission that followed an index hospitalization, regardless of the cause. Cause-specific readmissions refer to readmissions for or with a particular health condition or diagnosis, such as infection-related readmission. In this paper, we are interested in identifying hospitalizations that were followed by 90-day all-cause and cause-specific readmissions.

ALL-CAUSE READMISSIONS

The first objective of the paper is to identify hospitalizations that were followed by an all-cause readmission within 90 days of discharge. Since the data are originally at a daily level, they must first be subset by unique hospitalization so that there is only one row per hospitalization using the NODUPKEY option in the SORT procedure:

```
proc sort data= vapd out = vapd_onerow_ac nodupkey;
by PatientID descending AdmitDate descending DischargeDate;
run;
```

In order to retrieve the future admission dates (LeadAdmitDate) for each hospitalization, the LAG function is used. The first lagged variable (LeadAdmitDate) per patient will be missing due to the LAG function queue being empty. Therefore, the data are sorted by descending AdmitDate and DischargeDate. In this way, the final hospitalization per patient is listed first because there would not be any subsequent hospitalizations to consider. Since we are interested in 90-day readmissions, the time between discharge dates and future admission dates are calculated (TimetoReadm). The FIRST.PatientID variable is used to prevent calculating LeadAdmitDate and TimetoReadm for different patients by resetting the values to missing every time there is a new PatientID per row, which can be seen in Output 1. The variable FollBy_AC_Readm90 is created to indicate whether a hospitalization had a 90-day all-cause readmission.

```
data vapt_onerow_ac;
set vapt_onerow_ac;
by PatientID descending AdmitDate descending DischargeDate;
LeadAdmitDate=lag(AdmitDate);
format LeadAdmitDate mmddyy10.;
TimetoReadm=abs(DischargeDate-LeadAdmitDate);
if first.PatientID then do;
LeadAdmitDate=.;
TimetoReadm=.;
end;
if 0 <= TimetoReadm <=90 then FollBy_AC_Readm90=1;
else FollBy_AC_Readm90=0;
run;
```

Obs	PatientID	Date	UniqueHospID	HospitalDay	AdmitDate	DischargeDate	LeadAdmitDate	TimetoReadm
1	001	06/11/2018	4567	1	06/11/2018	06/13/2018	.	.
2	002	03/01/2020	6543	1	03/01/2020	03/02/2020	.	.
3	002	02/20/2020	5678	1	02/20/2020	02/21/2020	03/01/2020	9
4	002	01/07/2020	1234	1	01/07/2020	01/08/2020	02/20/2020	43
5	003	01/30/2020	3897	1	03/30/2020	03/30/2020	.	.
6	003	10/26/2019	2468	1	10/26/2019	10/27/2019	03/30/2020	155

Output 1. Hypothetical Data Set After Using LAG Function and FIRST. Variable

For easier viewing, the data are sorted back into ascending AdmitDate and DischargeDate, as shown in Output 2:

```
proc sort data= vapt_onerow_ac;
by PatientID AdmitDate DischargeDate HospitalDay;
run;
```

PatientID	Date	UniqueHospID	HospitalDay	AdmitDate	DischargeDate	LeadAdmitDate	TimetoReadm	FollBy_AC_Readm90
001	06/11/2018	4567	1	06/11/2018	06/13/2018	.	.	0
002	01/07/2020	1234	1	01/07/2020	01/08/2020	02/20/2020	43	1
002	02/20/2020	5678	1	02/20/2020	02/21/2020	03/01/2020	9	1
002	03/01/2020	6543	1	03/01/2020	03/02/2020	.	.	0
003	10/26/2019	2468	1	10/26/2019	10/27/2019	03/30/2020	155	0
003	01/30/2020	3897	1	03/30/2020	03/30/2020	.	.	0

Output 2. Sorted Hypothetical Data Set After Using PROC SORT

CAUSE-SPECIFIC READMISSIONS

Cause-specific readmissions refer to readmissions for or with a particular condition or diagnosis. Our second objective is to identify hospitalizations that were followed by a cause-specific readmission within 90 days of discharge. We were interested in knowing if a patient was readmitted for acute hypoxic respiratory failure. Specifically, we assessed whether patients received 4 or more liters per minute (4+LPM) of supplemental oxygen at time of readmission, operationalized as between 24 hours prior to and 48 hours after emergency department [ED] arrival during the 90-day readmission. In terms of coding, this is more challenging to identify because now all subsequent readmissions that occurred within 90 days need to be considered when confirming whether they received 4+LPM of supplemental oxygen, and not merely the first admission occurring in the 90-day window. The following macro demonstrates how to achieve this by using SAS programming tools.

Obs	PatientID	FacilityID	Date	UniqueHospID	HospitalDay	AdmitDate	DischargeDate	O2_GE4LPM_Hosp
1	001	222	06/11/2018	4567	1	06/11/2018	06/13/2018	1
2	001	222	06/12/2018	4567	2	06/11/2018	06/13/2018	1
3	001	222	06/13/2018	4567	3	06/11/2018	06/13/2018	1
4	002	123	01/07/2020	1234	1	01/07/2020	01/08/2020	0
5	002	123	01/08/2020	1234	2	01/07/2020	01/08/2020	0
6	002	444	02/20/2020	5678	1	02/20/2020	02/21/2020	0
7	002	444	02/21/2020	5678	2	02/20/2020	02/21/2020	0
8	002	321	03/01/2020	6543	1	03/01/2020	03/02/2020	1
9	002	321	03/02/2020	6543	2	03/01/2020	03/02/2020	1
10	003	333	10/26/2019	2468	1	10/26/2019	10/27/2019	1
11	003	333	10/27/2019	2468	2	10/26/2019	10/27/2019	1
12	003	333	01/30/2020	3897	1	03/30/2020	03/30/2020	1

Output 3. Hypothetical Data Set Before Applying Macro

Since the data are in long format, as displayed in Output 3, we must transpose them per patient so that we can compare future hospitalization data on a row-by-row basis for each hospitalization. We must first subset the data by unique hospitalizations and sort them by descending AdmitDate and DischargeDate, as we did when identifying all-cause readmissions:

```
%macro lag;

proc sort data=vapd out=vapd_onerow_cs nodupkey;
  by PatientID descending AdmitDate descending DischargeDate;
run;
```

We will need to transpose the values of each subsequent admission date and supplemental oxygen indicator variable (O2_GE4LPM_Hosp) for every hospitalization per patient. The number of hospitalizations per patient varies, therefore the maximum number of hospitalizations must be counted to know how many new transposed variables to create. To do this, we used PROC SQL to count the maximum number of hospitalizations across all patients and store it into a macro variable named MaxHosps, by using the INTO clause. We subtract this value by 1 since the final hospitalization for each patient will not have any future data that will need to be transposed:

```
proc sql noprint;
```

```

select (max(total)-1) into :MaxHosps trimmed
from (select count(UniqueHospID) as total from vapt_onerow_cs
group by PatientID);
quit;

```

The data are transposed by creating a series of arrays and by using the LAG function for the future admission dates (LeadAdmitDate n) and supplemental oxygen indicators (O2Lead n) for each readmission per patient, as shown in Output 4. To avoid erroneously including records from a different patient in the LAG queue, the IFN function and FIRST.PatientID variable are used (Keintz, 2016).

```

data vapt_onerow_cs;
set vapt_onerow_cs ;
by PatientID;
  array LeadAdmitDate [&MaxHosps] LeadAdmitDate1-LeadAdmitDate&MaxHosps;
  array O2Lead [&MaxHosps] O2Lead1-O2Lead&MaxHosps;
  array O2Readm [&MaxHosps] O2Readm1-O2Readm&MaxHosps;

LeadAdmitDate1 = ifn(not(first.PatientID),lag(AdmitDate),.);
O2Lead1 = ifn(not(first.PatientID),lag(O2_GE4LPM_Hosp),.);
format LeadAdmitDate1 mmddyy10.;

```

```

%do i = 2 %to &MaxHosps;
  LeadAdmitDate&i = ifn(lag&i(PatientID)=PatientID,lag&i(AdmitDate),.);
  O2Lead&i = ifn(lag&i(PatientID)=PatientID,lag&i(O2_GE4LPM_Hosp),.);
  format LeadAdmitDate&i mmddyy10.;

%end;

```

The variables O2Readm n specify whether 4+LPM of supplemental oxygen were administered during the n^{th} 90-day readmission. To ultimately identify the hospitalizations that were followed by a cause-specific 90-day readmission, we used the O2Readm n variables to conditionally create the indicator variable FollBy_CS_Readm90.

```

%do i = 1 %to &MaxHosps;
  if (0 <= (LeadAdmitDate&i - DischargeDate) <= 90) and (O2Lead&i = 1)
  then O2Readm&i = 1;
  else O2Readm&i = 0;

%end;

if sum(of O2Readm1-O2Readm&MaxHosps) => 1 then FollBy_CS_Readm90= 1;
else FollBy_CS_Readm90= 0;

```

PatientID	UniqueHospID	AdmitDate	DischargeDate	O2_GE4LPM_Hosp	LeadAdmitDate1	LeadAdmitDate2	O2Lead1	O2Lead2	O2Readm1	O2Readm2	FollBy_CS_Readm90
001	4567	06/11/2018	06/13/2018	1	-	-	-	-	0	0	0
002	6543	03/01/2020	03/02/2020	1	-	-	-	-	0	0	0
002	5678	02/20/2020	02/21/2020	0	03/01/2020	-	1	-	1	0	1
002	1234	01/07/2020	01/08/2020	0	02/20/2020	03/01/2020	0	1	0	1	1
003	3897	03/30/2020	03/30/2020	1	-	-	-	-	0	0	0
003	2468	10/26/2019	10/27/2019	1	03/30/2020	-	1	-	0	0	0

Output 4. Hypothetical Data After Transposing Data

It is useful to obtain the earliest cause-specific readmission date (FollBy_CS_Readm_Date) per hospitalization in case you will need a time variable (TimetoReadm_CS) to use in a time-to-event or similar analysis, as shown in the code below:

```

%do i = &MaxHosps %to 1 %by -1;

```

```

if O2Readm&i = 1 then FollBy_CS_Readm_Date= LeadAdmitDate&i;
else if sum(of O2Readm1-O2Readm&MaxHosps)=0 then FollBy_CS_Readm_Date=
.;
format FollBy_CS_Readm_Date mmddyy10.;
%end;

TimetoReadm_CS = FollBy_CS_Readm_Date-DischargeDate;

run;

```

The data are sorted back into ascending AdmitDate and DischargeDate per patient for a clearer demonstration of the final data set. In Output 5, you can see patient 002 had two hospitalizations (UniqueHospID 1234 and 5678) that had a readmission (UniqueHospID 6543) with 4+LPM of supplemental oxygen administered.

```

proc sort data=vapd_onerow_cs;
by PatientID AdmitDate DischargeDate;
run;

%mend;
%lag

```

PatientID	UniqueHospID	AdmitDate	DischargeDate	O2_GE4LPM_Hosp	LeadAdmitDate1	LeadAdmitDate2	O2Lead1	O2Lead2	O2Readm1	O2Readm2	FollBy_CS_Readm90	FollBy_CS_Readm_Date
001	4567	06/11/2018	06/13/2018	1	.	.	.	.	0	0	0	.
002	1234	01/07/2020	01/08/2020	0	02/20/2020	03/01/2020	0	1	0	1	1	03/01/2020
002	5678	02/20/2020	02/21/2020	0	03/01/2020	.	1	.	1	0	1	03/01/2020
002	6543	03/01/2020	03/02/2020	1	.	.	.	.	0	0	0	.
003	2468	10/26/2019	10/27/2019	1	03/30/2020	.	1	.	0	0	0	.
003	3897	03/30/2020	03/30/2020	1	.	.	.	.	0	0	0	.

Output 5. Final Hypothetical Data Set After Applying Macro

CONCLUSION

Hospital readmission data are extensively studied in clinical research and are often used as a health care quality metric. Analyzing cause-specific readmissions can aid in identifying causes of readmissions to guide strategies to improve patient care and health outcomes. Longitudinal hospitalization data can be challenging to manipulate, especially when managing big data. In this paper, we introduce a number of SAS programming tools that efficiently identify hospitalizations that were followed by all-cause and cause-specific readmissions. While there may be other methods of achieving the same results, this is a concise and easy program that can be applied to other data sets.

REFERENCES

- Denstaedt SJ, Cano J, Wang XQ, Donnelly JP, Seelye S, Prescott HC. 2023 Feb 28. "Blood count derangements after sepsis and association with post-hospital outcomes". *Front Immunol*. 14. Available from: <https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2023.1133351/full>
- Keintz, Mark. 2016 "Leads and Lags: Static and Dynamic Queues in the SAS® DATA STEP." *MWSUG 2016 Proceedings*, Cincinnati, Ohio. Available at <https://support.sas.com/resources/papers/proceedings16/11221-2016.pdf>.
- Vincent BM, Wiitala WL, Burns JA, Iwashyna TJ, Prescott HC. 2018 Sep. "Using Veterans Affairs Corporate Data Warehouse to identify 30-day hospital readmissions". *Health Serv Outcomes Res Methodol*. 18(3):143–54.

Wang XQ, Vincent BM, Wiitala WL, Luginbill KA, Viglianti EM, Prescott HC, Iwashyna TJ. Veterans Affairs patient database (VAPD 2014-2017): building nationwide granular data for clinical discovery. BMC Med Res Methodol. 2019 May 8;19(1):94. doi: 10.1186/s12874-019-0740-x. PMID: 31068135; PMCID: PMC6505066.

CONTACT INFORMATION

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

Jennifer Cano
VA Center for Clinical Management Research
734-222-8957
jennifer.cano@va.gov

APPENDIX: HYPOTHETICAL DATA AND FULL MACRO

```
/* Create hypothetical data set */

data vapd;
input PatientID $ FacilityID $ Date :mmddyy10. UniqueHospID HospitalDay
      AdmitDate :mmddyy10. DischargeDate :mmddyy10. O2_GE4LPM_Hosp;
format Date AdmitDate DischargeDate mmddyy10.;
datalines;
001 222 06/11/2018 4567 1 06/11/2018 06/13/2018 1
001 222 06/12/2018 4567 2 06/11/2018 06/13/2018 1
001 222 06/13/2018 4567 3 06/11/2018 06/13/2018 1
002 123 01/07/2020 1234 1 01/07/2020 01/08/2020 0
002 123 01/08/2020 1234 2 01/07/2020 01/08/2020 0
002 444 02/20/2020 5678 1 02/20/2020 02/21/2020 0
002 444 02/21/2020 5678 2 02/20/2020 02/21/2020 0
002 321 03/01/2020 6543 1 03/01/2020 03/02/2020 1
002 321 03/02/2020 6543 2 03/01/2020 03/02/2020 1
003 333 10/26/2019 2468 1 10/26/2019 10/27/2019 1
003 333 10/27/2019 2468 2 10/26/2019 10/27/2019 1
003 333 01/30/2020 3897 1 03/30/2020 03/30/2020 1
;
run;

/* All-cause Readmissions */

/*get the latest hospitalizations listed first and subset data set to only
have 1 row per hospitalization */
proc sort data= vapd out = vapd_onerow_ac nodupkey;
by PatientID descending AdmitDate descending DischargeDate;
run;

/*create lag variables and create all-cause 90-day readmission indicator*/
data vapd_onerow_ac;
set vapd_onerow_ac;
by PatientID descending AdmitDate descending DischargeDate;
LeadAdmitDate=lag(AdmitDate);
format LeadAdmitDate mmddyy10.;
TimetoReadm=abs(DischargeDate-LeadAdmitDate);
if first.PatientID then do;
LeadAdmitDate=.;
TimetoReadm=.;
end;
if 0 <= TimetoReadm <=90 then FollBy_AC_Readm90=1; *you can change
number of days to readmission here;
else FollBy_AC_Readm90=0;
run;

proc sort data= vapd_onerow_ac;
by PatientID AdmitDate DischargeDate HospitalDay;
run;
```

```

/* Cause-specific Readmissions */

options macrogen symbolgen mlogic merror;

%macro lag;

/*subset data set to only have 1 row per hospitalization*/
proc sort data=vapd out=vapd_onerow_cs nodupkey;
by PatientID descending AdmitDate descending DischargeDate;
run;

/*count the max total hospitalizations per patient*/
proc sql noprint;
select (max(total)-1) into :MaxHosps trimmed
from (select count(UniqueHospID) as total from vapd_onerow_cs
group by PatientID);
quit;

/*create lag variables*/
data vapd_onerow_cs;
set vapd_onerow_cs ;
by PatientID;
    array LeadAdmitDate [&MaxHosps] LeadAdmitDate1-LeadAdmitDate&MaxHosps ;
    array O2Lead [&MaxHosps] O2Lead1-O2Lead&MaxHosps;
    array O2Readm [&MaxHosps] O2Readm1-O2Readm&MaxHosps;

LeadAdmitDate1 = ifn(not(first.PatientID),lag(AdmitDate),.);
O2Lead1 = ifn(not(first.PatientID),lag(O2_GE4LPM_Hosp),.);
format LeadAdmitDate1 mmddyy10.;

%do i = 2 %to &MaxHosps;
    LeadAdmitDate&i = ifn(lag&i(PatientID)=PatientID,lag&i(AdmitDate),.);
    O2Lead&i = ifn(lag&i(PatientID)=PatientID,lag&i(O2_GE4LPM_hosp),.);
    format LeadAdmitDate&i mmddyy10.;

%end;

%do i = 1 %to &MaxHosps;
    if (0 <= (LeadAdmitDate&i - DischargeDate) <= 90) and (O2Lead&i = 1)
then O2Readm&i = 1; *you can change number of days to readmission here;
    else O2Readm&i = 0;

%end;

/*create cause-specific readmission indicator variable*/
if sum(of O2Readm1-O2Readm&MaxHosps) => 1 then FollBy_CS_Readm90= 1;
    else FollBy_CS_Readm90= 0;

/*get earliest date of readmission with supplemental oxygen and create time
to earliest readmission variable*/
%do i = &MaxHosps %to 1 %by -1;
if O2Readm&i = 1 then FollBy_CS_Readm_Date= LeadAdmitDate&i;
else if sum(of O2Readm1-O2Readm&MaxHosps) = 0 then FollBy_CS_Readm_Date= .;
format FollBy_CS_Readm_Date mmddyy10.;
%end;

TimetoReadm_CS = FollBy_CS_Readm_Date-DischargeDate;

```

```
run;

proc sort data=vapd_onerow_cs;
by PatientID AdmitDate DischargeDate;
run;

%mend;
%lag

proc print data = vapd_onerow_cs;
run;
```