A Novel Approach to Calculating Medicare Hospital 30-Day Readmissions for the SAS® Novice

Karen E. Wallace, Centene® Corporation, Orlando, Florida

ABSTRACT

Hospital Medicare readmission rate has become a key indicator for measuring the quality of healthcare in the US, currently adopted by major healthcare stakeholders including the Centers for Medicare and Medicaid Services (CMS), the Agency for Healthcare Research and Quality (AHRQ), and the National Committee for Quality Assurance (NCQA) (Fan and Sarfarazi, 2014).

Although many papers have been written about how to calculate readmissions, this paper includes updated coding to include ICD-10 (International Classification of Diseases) as well as offering a novel and comprehensive approach using the options of the SAS DATA Step as well as PROC SQL for: 1) de-identifying patient data, 2) calculating sequential admissions, and 3) subsetting criteria required to report for CMS 30-day readmissions. Additionally, it demonstrates: 1) using Output Delivery System (ODS) to create a labeled and de-identified data set, 2) macro variables to examine data quality, and 3) summary statistics useful for further reporting and analysis.

INTRODUCTION

Excessive Medicare readmissions to inpatient prospective payment system (IPPS) hospitals are costly, affecting not only a patient's personal quality of life but moreover, accrue penalties that are levied by CMS. When the Affordable Care Act added section 1886(q) to the Social Security Act, it established the Hospital Readmissions Reduction Program (HRRP), requiring CMS to reduce payments to IPPS facilities with readmissions that became effective for discharges that began on October 1, 2012 (Fan and Sarfarazi, 2014).

For fiscal year 2015 CMS added an additional two conditions—chronic obstructive pulmonary disease and total hip and total knee replacements—and the penalty rose to 3%. The majority of hospitals faced fines with the number subject to penalties in fiscal year 2016 rising to 2,665. Only 799 out of more than 3,400 hospitals subject to the HRRP performed well enough on CMS' 30-day readmission program to face no penalty. Thirty-eight hospitals will be subject to the maximum 3% reduction (Rice, 2015).

As of October 1, 2015, healthcare facilities were required to implement the World Health Organization's authored ICD-10 codes, a major transition from ICD-9 codes. This adds another layer of complexity for the following reasons:

☐ There are nearly 19 times as many procedure codes in ICD-10-PCS (Procedure Coding System) than in ICD-9-CM (Clinical Modification) volume 3
☐ There are nearly 5 times as many diagnosis codes in ICD-10-CM than in ICD-9-CM
☐ ICD-10 has alphanumeric codes instead of numeric ones; hence, there is no definitive 1:1 crosswalk between the two versions (the General Equivalency Mappings, or GEMs, created by CMS to address this issue have been disputed)

The order of some chapters have changed, some titles have been renamed, and conditions have been grouped differently (CMS and CDC, 2015)

Varying measurement methodologies pose challenges due to differences in defining the initial index admission, identifying what constitutes a readmission, and calculating the readmission rate (HCUP, 2012, Jencks, 2009 and Yale, 2013). For the purposes of this paper, a de-identified data set examines 30-day readmissions using the initial three ICD-9 conditions plus the additional ICD-10 codes that came into effect in October 2015 to identify the conditions acute myocardial infarction (AMI), congestive heart failure (CHF) and pneumonia (PNE).

Criteria for index admissions include counting inpatient admissions for patients that meet the following:

 Are discharged alive
 Are Medicare recipients at time of admission
 Discharge date is not the same as the index admission date
 Patients are not discharged to another acute care hospital (ACH)
 Patients were not discharged against medical advice

Taking these many factors into account, SAS software offers various approaches to cleanse, analyze and subset large and complex datasets.

I) IMPORTING RAW DATA

BACKGROUND INFORMATION
The raw data used is inpatient data collected at two points in time, June 2015 and January 2016. The reason for the differing time periods was to evaluate both ICD-9 and ICD-10 codes. The file is an extract to an Excel spreadsheet, sorted by medical record number ("MRN"), index date ("Index_Date") and discharge date ("DC_Date"). The sample is approximately 2,500 records. The other key variables and respective data types include the following:

MRN 8
DUMMY 8
SeqNo 8
Index_date 8
DC_Date 8
First_DC_Disp $ 35
Payer $ 47
First_Dx $ 8
Dx_Desc $ 151

"MRN" is the number that allows a patient to be uniquely tracked across the hospital system. "Dummy" is a binary variable that was created in Excel using the IF function to see if a patient had a readmission based upon looking at a previous row of data. Additionally, "SeqNo" was manually added in Excel to identify in what order, if applicable, a patient was readmitted. For the purpose of this exercise, "Dummy" and "SeqNo" were created as added

validation checks to ensure the SAS coding correctly assigns readmissions (Fan and Sarfarazi, 2014).

Other variables include "First_DC_Disp", the location where the patient was discharged. "Payer" indicates what insurance carrier the patient has at admission. "First_Dx" is the primary initial ICD-9 or ICD-10 code the patient had (there may be secondary and further complicating comorbid diagnoses; however, these are not analyzed). Lastly, "Dx_Desc" is the free text that describes the diagnosis in detail.

II) FILTERING BY CRITERIA AND DATA VALIDATION/ VISUALIZATION

DE-IDENTIFY THE MRN

After reading the file into SAS, the data is sorted and de-identified by creating a new variable, "PTID" (subsequently, "MRN" is dropped). Using the RETAIN statement and "FIRST. Logical" variable (Cody, 2012), the "PTID" auto-increments based upon each new "MRN" identified.

```
PROC SORT DATA=MIDAS_TEST_TO_DEIDENTIFY_V3 OUT=PT_SORTED;
BY MRN Index_date DC_Date;
RUN;
DATA PT_CLEAN;
SET PT_SORTED;
BY MRN Index_date DC_Date;
RETAIN PTID 0;
IF FIRST.MRN THEN PTID = PTID + 1;
DROP MRN;
RUN;
```

Output 1: Results from the first DATA step

Checking that data is correct

DUMMY	SeqNo	Index_date	DC_Date	First_DC_Disp	Payer	First_Dx	Dx_Desc	PTID
0	1	01JUN2015	01JUN2015	O/P HOME/SELF CARE (ROUTINE)	MEDICARE	599	URIN TRACT INFECTION NOS	1
0	1	02JUN2015	02JUN2015	O/P HOME/SELF CARE (ROUTINE)	MEDICARE	627.2	SYMPT FEM CLIMACT STATE	2
0	1	01JUN2015	01JUN2015	O/P HOME/SELF CARE (ROUTINE)	MEDICARE	847	SPRAIN OF NECK	3

The results are illustrated in Output 1 (only the first three observations are displayed). As there are no readmissions in this sample, "PtID" auto-increments and "SeqNo" shows one count for each of the visits.

ESTABLISH SEQUENCE NUMBERS AND CATEGORIZE BY ICD CODE

After creating the "PTID" variable, the sequence of readmissions are established, again using the RETAIN statement and "FIRST. Logical" variable. Using the SUBSTR function (Wilson and Scerbo, 1999), the ICD codes are evaluated based upon the first three alpha-numeric characters. If any of the diagnoses fall into one of the three diagnosis buckets of AMI, CHF or PNE, then variable "Dx1" is populated accordingly. The data is formatted and labeled and the variable "SeqNo" is dropped.

```
DATA DATA_TEST;
SET PT_CLEAN;
BY PTID Index_date DC_Date;
RETAIN N;
IF FIRST.PTID THEN N = 0;
N = N + 1;
IF NOT MISSING(First_Dx) THEN
IF SUBSTR(First_Dx,1,3) = '410'
OR (SUBSTR(First_Dx,1,3) = 'I21')
OR (SUBSTR(First_Dx,1,3) = 'I22')
OR (SUBSTR(First_Dx,1,3) = 'I23') THEN Dx1 = 'AMI';
ELSE IF (SUBSTR(First_Dx,1,3) = '402')
OR (SUBSTR(First_Dx,1,3) = '404')
OR (SUBSTR(First_Dx,1,3) = '428')
OR (SUBSTR(First_Dx,1,3) = 'I50') THEN Dx1 = 'CHF';
ELSE IF (SUBSTR(First_Dx,1,3) = '480')
OR (SUBSTR(First_Dx,1,3) = '481')
OR (SUBSTR(First_Dx,1,3) = '482')
OR (SUBSTR(First_Dx,1,3) = '483')
OR (SUBSTR(First_Dx,1,3) = '485')
OR (SUBSTR(First_Dx,1,3) = '486')
OR (SUBSTR(First_Dx,1,3) = '487')
OR (SUBSTR(First_Dx,1,3) = '488')
OR (SUBSTR(First_Dx,1,3) = 'J18') THEN Dx1 = 'PNE';
FORMAT Index_Date DC_Date YYMMDD10.;
LABEL Dx1 = "ICD9 Grouping"
PTID = "Patient ID"
N = "SeqNo"
Index_date = "Index Date"
DC_Date = "Discharge Date"
First_Dx = "ICD9/10 Primary Diagnosis"
First_DC_Disp = "Discharge Disposition"
Dx_Desc = "ICD9/10 Diagnosis Description";
DROP DUMMY SeqNo;
RUN;
```

Output 2: Results from the second DATA step

Checking that data is correct

Index Date	Discharge Date	Discharge Disposition	Payer	ICD9/10 Primary Diagnosis	ICD9/10 Diagnosis Description	Patient ID	SeqNo	ICD9 Grouping
2015-06-01	2015-06-01	O/P HOME/SELF CARE (ROUTINE)	MEDICARE	599	URIN TRACT INFECTION NOS	1	1	
2015-06-02	2015-06-02	O/P HOME/SELF CARE (ROUTINE)	MEDICARE	627.2	SYMPT FEM CLIMACT STATE	2	1	
2015-06-01	2015-06-01	O/P HOME/SELF CARE (ROUTINE)	MEDICARE	847	SPRAIN OF NECK	3	1	

The results are illustrated in Output 2 (only the first three observations are displayed).
Labels clearly identify the relevant variables for analysis.

DATA ELEMENTS ANALYZED: COUNT OF READMISSIONS BY DIAGNOSIS

A PROC FREQ statement is used to examine the sequential count of readmissions by diagnosis code.

```
PROC FREQ DATA=DATA_TEST;
TABLES N * Dx1 / NOROW NOCOL MISSING;
RUN;
```

Output 3: Results from PROC FREQ statement

The FREQ Procedure

Table of N by Dx1					
		Dx1(ICD9 Grouping)			
		AMI	CHF	PNE	Total
N(SeqNo)					
1 Frequency		2567	52	96	131	2846
Percent		87.94	1.78	3.29	4.49	97.50
2 Frequency		29	13	10	15	67
Percent		0.99	0.45	0.34	0.51	2.30
3 Frequency		2	0	2	1	5
Percent		0.07	0.00	0.07	0.03	0.17
4 Frequency		1	0	0	0	1
Percent		0.03	0.00	0.00	0.00	0.03
Total Frequency		2599	65	108	147	2919
Percent		89.04	2.23	3.70	5.04	100.00

The results are illustrated in Output 3. Based upon the "N" variable, 73 visits qualified as a readmission (less than 3%) regarding the ICD codes.

EXPORT DE-IDENTIFIED DATA SET TO EXCEL: ODS VISUALIZATION

A de-identified data set is created in Excel for possible distribution purposes.

```
ODS LISTING CLOSE;
ODS HTML BODY='/folders/myfolders/Capstone_test.xls';
DATA _NULL_;
SET DATA_TEST;
FILE PRINT ODS;
PUT _ODS_;
RUN;
ODS HTML CLOSE;
ODS LISTING;
```

Output 4: Results displaying ODS visualization

	A	B	C	D	E	F	G	H
1	Index Date	Discharge Date	Discharge Disposition	Payer	ICD9 Primary	ICD9 Diagnosis Description	PTID	N
2					Diagnosis			
3	6/1/2015	6/1/2015	O/P HOME/SELF CARE (ROUTINE)	MEDICARE	599	URIN TRACT INFECTION NOS	1	1
4	6/2/2015	6/2/2015	O/P HOME/SELF CARE (ROUTINE)	MEDICARE	627.2	SYMPT FEM CLIMACT STATE	2	1
5	6/1/2015	6/1/2015	O/P HOME/SELF CARE (ROUTINE)	MEDICARE	847	SPRAIN OF NECK	3	1

The results are illustrated in Output 4 (only the first three observations are displayed). Having the data in this format obviates any concern of violating patient confidentiality in accordance with Health Insurance Portability and Accountability Act of 1996 (HIPAA).

FURTHER DATA CLEANSING TO IDENTIFY READMITS

Within this DATA step, several variables are created or dropped based upon restriction criteria. The LAG function is used to identify the value from the previous discharge date (if applicable). The "GAP" variable calculates days elapsing from the index date to the most recent readmission. The "Flg" variable is a flag created for readmissions occurring within 30 days. A cumulative readmissions "log" is created (Fan and Sarfarazi, 2014).
Under the following circumstances, the record is dropped:

☐ "GAP" is null or zero
☐ "Payer" is null or not equal to Medicare
☐ Diagnosis is not within the three categories
☐ If the discharge disposition includes an ACH transfer, discharge against medical advice, or the patient expired

Length of stay (LOS), a key performance indicator to evaluate, is calculated. Finally, formats and labels are applied and other unneeded variables are dropped.

```
DATA DATA_TEST2;
SET DATA_TEST;
BY PTID Index_date DC_Date;
REF_DATE = LAG(DC_Date);
GAP =(Index_Date - REF_DATE);
IF FIRST.PTID THEN DO;
Ref_Date =.;
Gap =.;
Flg =.;
Readmissions =.;
END;
IF 0 <= Gap <=30 then Flg=1;
IF (Gap = '.' OR Gap le 0) THEN DELETE;
Readmissions + Flg;
IF ((Payer ne 'MEDICARE') OR (Payer ='')) THEN DELETE;
IF ((Payer ne 'MEDICARE') OR (Payer ='')) THEN DELETE;
IF Dx1 NOT IN('AMI','CHF','PNE') THEN DELETE;
IF NOT MISSING(First_DC_Disp) THEN
IF First_DC_Disp eq 'I/P ACUTE HOSPITAL TRANSFER' OR 'O/P ACUTE HOSPITAL
TRANSFER'
OR First_DC_Disp eq 'DISCHARGED AGAINST MED ADVICE'
OR First_DC_Disp eq 'I/P EXPIRED'
THEN DELETE;
```

LOS = (DC_Date - Index_date) + 1
ICD9_DESC = UPCASE(Dx_Desc);
FORMAT INDEX_DATE REF_DATE YYMMDD10.;
LABEL Dx1 = 'ICD9/10 Grouping'
Gap = 'Gap to Readmission (d)'
LOS = 'Length of Stay (d)'
Ref_Date = 'Index Reference Date';
DROP Flg Dx_Desc;

RUN;

Output 5: Results showing cleansed data with calculations

Patient ID	SeqNo	Readmissions	ICD9/10 Primary Diagnosis	ICD9/10 Grouping	Gap to Readmission (d)	Length of Stay (d)	Index Date	Index Reference Date	Discharge Date	Discharge Disposition
92	2	1	428.33	CHF	1	8	2015-06-03	2015-06-02	2015-06-10	O/P HOME/SELF CARE (ROUTINE)
155	2	1	402.01	CHF	1	2	2015-06-02	2015-06-01	2015-06-03	O/P HOME/SELF CARE (ROUTINE)
155	3	2	402.1	CHF	2	5	2015-06-05	2015-06-03	2015-06-09	O/P HOME/SELF CARE (ROUTINE)

The results are illustrated in Output 5 (only the first three observations are displayed). Labels clearly identify the relevant variables for analysis.

MACRO VARIABLES EVALUATE DATA QUALITY

Using PROC SQL, three macro variables, "MAXRED", "PTID", and "DX1" are evaluated to determine which patient had the most readmissions, the count and the diagnosis (Slaughter and Delwiche, 2011). The option SYMBOLGEN helps with clearly interpreting the output.

```
OPTIONS SYMBOLGEN;
PROC SORT DATA=DATA_TEST2 OUT=ORDERS;
BY DESCENDING Readmissions;
PROC SQL NOPRINT;
SELECT MAX(Readmissions) as MaxReadmits
, PTID
, DX1
INTO :MAXRED
, :PTID
, :DX1
FROM ORDERS;
```

Output 6: Results from the macro variables

%PUT The max readmission count is &MAXRED by patient &PTID with diagnosis &DX1
SYMBOLGEN: Macro variable MAXRED resolves to 2
SYMBOLGEN: Macro variable PTID resolves to 155
SYMBOLGEN: Macro variable DX1 resolves to CHF

The results are illustrated in Output 6. The macro output for each variable is prefaced by SYMBOLGEN and displayed in the SAS log

GENERATING SUMMARY STATISTICS

With PROC MEANS, summary statistics including the minimum, maximum, median, and first and third quartiles are generated which examine "GAP" and "LOS" variables by diagnosis.

```
PROC MEANS DATA=DATA_TEST2 NOPRINT MAXDEC=1;
CLASS Dx1;
VAR Gap LOS;
OUTPUT OUT = dx_stats
MIN = min_gap min_los
Q1 = q1_gap q1_los
MEDIAN = median_gap median_los
Q3 = q3_gap q3_los
MAX = max_gap max_los;
RUN;
```

Output 7: Results of descriptive statistics for each diagnosis

	Dx1	_TYPE_	_FREQ_	min_gap	min_los	q1_gap	q1_los	median_gap	median_los	q3_gap	q3_los	max_gap
1		0	14	1	1	1	2	3.5	5.5	4	6	1
2	AMI	1	6	1	1	1	2	4	4	5	6	1
3	CHF	1	6	1	2	1	4	1.5	5	3	6	
4	PNE	1	2	4	6	4	6	6.5	6.5	9	7	

The results are illustrated in Output 7. For each diagnosis, the descriptive statistics are displayed.

III) DATA CLEANSING USING PROC SQL

AN ALTERNATIVE TO THE DATA STEP

PROC SQL is used in conjunction with a self-join to create a table that houses data almost identical to that produced in the earlier DATA steps (Williams, 2012). A limitation to using PROC SQL with either Enterprise Guide or University Edition is that the variable "Ref_Date" cannot be generated. In BASE SAS, the LEAD function is akin the LAG function, the latter not recognized in PROC SQL.

```
OPTIONS MSGLEVEL=I;
PROC SQL;
CREATE TABLE PATIENTS_SQL AS
SELECT
a.PTID LENGTH = 4 LABEL = "Patient ID"
,a.N LENGTH = 3 LABEL = "SeqNo"
,a.DC_Date LENGTH = 8 LABEL = "Discharge Date"
,b.Index_Date LENGTH = 8 LABEL = "Index Date"
,(b.Index_Date - a.DC_Date) as Gap LENGTH = 3 LABEL = "Gap to Readmission (d)"
,(b.DC_Date - b.Index_Date) + 1 as LOS LENGTH = 3

LABEL = "Length of Stay (d)"
,a.Payer
,1 as Flg
,b.First_Dx LENGTH = 6 LABEL = "ICD9/10 Primary Diagnosis"
```

```
,b.Dx1 LENGTH = 6 LABEL = "ICD9 Grouping"
,a.First_DC_Disp LENGTH = 155 LABEL = "Discharge Disposition"
FROM DATA_TEST as a
INNER JOIN DATA_TEST as b
ON a.PTID = b.PTID
WHERE (b.N = a.N + 1) AND
(b.Index_Date - a.DC_Date) BETWEEN 1 AND 30
AND a.Payer = 'MEDICARE'
AND (a.First_DC_Disp NOT IN('I/P ACUTE HOSPITAL TRANSFER'
,'O/P ACUTE HOSPITAL TRANSFER'
,'DISCHARGED AGAINST MED ADVICE'
,'I/P EXPIRED'))
AND b.DX1 ne ''
ORDER BY a.PTID, a.N;
QUIT;
```

Output 8: Results from the PROC SQL statement

TESTING SQL TABLE CREATION

Patient ID	SeqNo	Discharge Date	Index Date	Gap to Readmission (d)	Length of Stay (d)	Payer	Flg	ICD9/10 Primary Diagnosis	ICD9 Grouping	Discharge Disposition
92	1	2015-06-02	2015-06-03	1	8	MEDICARE	1	428.33	CHF	O/P HOME/SELF CARE (ROUTINE)
155	1	2015-06-01	2015-06-02	1	2	MEDICARE	1	402.01	CHF	O/P HOME/SELF CARE (ROUTINE)
155	2	2015-06-03	2015-06-05	2	5	MEDICARE	1	402.1	CHF	O/P HOME/SELF CARE (ROUTINE)

The results are illustrated in Output 8. Labels clearly identify the relevant variables for analysis.

CONCLUSION

One of the great attributes of SAS is that there are often several alternatives to achieve the desired result. In this paper, the SAS DATA step was compared and contrasted with PROC SQL, each displaying varying programming language to output similar datasets.

In a step-wise manner, a novice SAS programmer may proceed through de-identifying patient data, calculating sequential admissions, and subsetting criteria required for CMS 30-day readmissions. Also shown is an example using ODS to create a labeled and de-identified data set, macro variables to examine data quality, and summary statistics for further reporting.

The caveat is that this example is neater than most healthcare data scenarios. Oftentimes, data is not as "clean" and there are often disparate sources from which to pull and reconcile. The statistics shown are merely descriptive and the sample size is relatively small. With a more robust dataset, such as one having cost and observational data (e.g. height/weight, blood pressure, heart rate, etc.), greater analysis may be done to discern trends, associations, and patterns (Shen and Lu, 2014) and whether these are statistically significant.

Despite these challenges, this "use case" is readily reproducible and lends itself to incorporate many factors. Using additional data may build a predictive model to determine which patients are most "at risk" so proactive clinical interventions may be performed to decrease readmission rate in the Medicaid population.

REFERENCES

☐ Centers for Disease Control. "Classification of Diseases, Functioning, and Disability". 2015. Available at http://www.cdc.gov/nchs/icd/icd10cm_pcs_background.htm.

☐ Centers for Medicare and Medicaid. "Road to 10". 2015. Available at http://www.roadto10.org/icd-10-basics/.

☐ Cody R. "Longitudinal Data Techniques: Looking Across Observations". SUGI 27. Orlando, Florida. Available at http://www.ats.ucla.edu/stat/sas/library/nesug00/ad1002.pdf.

☐ Jencks S, Williams MV and Coleman EA (2009). "Rehospitalizations among Patients in the Medicare Fee-for-Service Program". N ENGL J Med (360): 1418-1428.

☐ Osman, M *et al*. "Using SAS® to Create Episodes-of-Hospitalization for Health Services Research". SAS® Global Forum 2015. Dallas, Texas. Available at http://www.sas.com/content/dam/SAS/en_ca/User%20Group%20Presentations/Saskatoon-User-Group/Osman-EoHMethod-March2015.pdf.

☐ Rice S. "Most hospitals face 30-day readmissions penalty in fiscal 2016". Available at http://www.modernhealthcare.com/article/20150803/NEWS/150809981.

☐ Shen L and Lu J. "Healthcare Data Manipulation and Analytics Using SAS®". PharmaSUG 2014. San Diego, California. Available at http://www.pharmasug.org/proceedings/2014/PO/PharmaSUG-2014-PO17.pdf.

☐ Slaughter S and Delwiche L. "SAS® Macro Programming for Beginners". SAS Global Forum Paper 258-2011. Las Vegas, Nevada. Available at http://support.sas.com/resources/papers/proceedings11/258-2011.pdf.

☐ U.S. Agency for Healthcare Research and Quality. "HCUP Methods Series: Methodological Issues when Studying Readmissions and Revisits using Hospital Administrative Data (Report # 2012-04)". Available at https://www.hcup-us.ahrq.gov/reports/methods/2015_02.pdf.

☐ Weifeng F and Sarfarazi M. "SAS® Solutions to Identifying Hospital Readmissions". SAS® Global Forum Proceedings 2014. Washington, D.C. Available at http://support.sas.com/resources/papers/proceedings14/1622-2014.pdf.

☐ Williams, Christianna S. "Queries, Joins, and WHERE Clauses, Oh My!! Demistifying PROC SQL". SAS® Global Forum 2012. Orlando, Florida. Available at http://support.sas.com/resources/papers/proceedings12/149-2012.pdf.

☐ Wilson A and Scerbo M. "Dealing with Health Care Data Using the SAS® System". Proceedings of the Twenty-fourth Annual SAS® Users Group International Conference 1999. Cary, North Carolina. Available at http://www2.sas.com/proceedings/sugi24/Proceed.pdf.

☐ Yale New Haven Health Services Corporation/Centers for Outcomes Research & Evaluation (YNHHSC/CORE) 2012. "Measures Maintenance Technical Report: Acute Myocardial Infarction, Heart Failure, and Pneumonia 30-Day Risk-Standardized Readmission Measure", April 16, 2012. Available at http://www.henlearner.org/wp-content/uploads/2012/03/Hospital-Readmissions_Metric-Maintenance-Technical-Report.pdf

ACKNOWLEDGMENTS

RECOMMENDED READING

☐ Base SAS Procedures Guide

☐ SAS For Dummies

CONTACT INFORMATION

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

Name: Karen E. Wallace
Enterprise: Centene® Corporation
Address: 8427 Southpark Circle, Suite 400
City, State ZIP: Orlando, Florida 32819
Work Phone: 855-422-2742 ext. 809-3211
E-mail: Karen.wallace@EnvolveHealth.com