

Using SAS® to proactively monitor drug alerts: Implementing the URAC “Drug-Drug Interactions” measure for a specialty pharmacy

Karen E. Wallace, Centene® Corporation, Bedford, New Hampshire

ABSTRACT

To satisfy annual URAC accreditation requirements, one of the mandatory reported “Safe Care” domain measures is “Drug-Drug Interactions” (DM2012-13). This measure relies upon pharmacy and enrollment data to assess the percentage of patients who: 1) received a prescription for a target medication during the measurement period, and 2) were dispensed a concurrent prescription for a precipitant medication. The resultant data are stratified by line of business (URAC, 2016). Using SAS® Base programming, in conjunction with technical and clinical review, the preliminary methodology: 1) imports and joins several files using the data elements of patient identifier as well as medication and insurance plan, 2) cleanses the mixed-case, punctuated MediSpan-generated drug alert text, 3) categorizes the text into discrete flags based upon predetermined criteria, 4) sums the number of flags per observation, and finally, 5) segments data by insurance type. The reporting showcases de-identified, summarized descriptive statistics that may be used to better monitor drug alerts and, more importantly, intervene where there is perceived harm in the patient population. This paper is intended for the SAS® novice who may be challenged on ways to perform on-the-fly textual analysis with a mindset towards implementing standardized, automated reporting after receiving requisite oversight and sign-off.

INTRODUCTION

Drug Utilization Review (DUR) is the authorized, structured, on-going review of prescribing, dispensing, and use of medication. Using predetermined criteria, DUR is a review that results in therapy modification when such criteria are unmet. To ensure appropriate medication decision-making and to promote positive health outcomes, a comprehensive review of the patients’ prescription and non-prescription (e.g. over-the-counter, OTC) data before, during, and after dispensing is required. An essential component of a pharmacy’s continuous quality improvement program, DUR provides corrective action, prescriber feedback, and further evaluation where necessary (AMCP, 2009).



Specialty Pharmacy

Mandatory Measures (5)

Note: Mandatory measures are those measures that are a requirement of accreditation and must be reported to URAC on an annual basis.

Measure #	Measure Name	Measure Steward	URAC Domain	Measure Description	Numerator	Denominator	Data Source
DM2012-13	Drug-Drug Interactions	Pharmacy Quality Alliance (PQA)	Safe Care	This measure assesses the percentage of patients who received a prescription for a target medication during the measurement period and who were dispensed a concurrent prescription for a precipitant medication. Stratify by Commercial, Medicaid, and Medicare (i.e., report each product line separately).	The number of patients in the denominator who were dispensed a concurrent precipitant medication during the measurement period.	Patients who received a target medication.	Enrollment Data; Pharmacy Data

Figure 1: The URAC specifications for DM2012-13

In terms of governmental oversight, NCQA, CMS and several other government agencies mandate DUR to ensure appropriate drug therapy. Specifically, the Omnibus Budget Reconciliation Act of 1990 (OBRA 90) requires that pharmacists conduct medication reviews whenever an outpatient prescription is dispensed to a Medicaid recipient. Since the passing of OBRA 90, many states have enacted their own regulations requiring pharmacists to conduct medication reviews for all outpatients (AMCP, 2009).

Similarly, the Medication Therapy Management (MTM) mandate introduced in the Medicare Prescription Drug Improvement and Modernization Act (MMA) of 2003, requires all prescription drug plan sponsors who provide a drug benefit to offer medication reviews and appropriate interventions. With legislation beginning in 2006, plans must offer MTM programs to target at-risk beneficiaries to optimize therapeutic outcomes by improving medication usage and reducing risk of adverse events. Further, the CMS Medicare Part D 2010 Call Letter requires prescription drug plan sponsors to conduct DUR for enrollees (AMCP, 2009).

The *three* distinct stages of DUR include:

1. Prospective: Evaluate the patient's planned therapy *before* the medication dispense
2. Concurrent: Monitor outcomes *during* the patient's treatment
3. Retrospective: Review therapy *after* the patient has received medication (AMCP, 2009)

Ideally, issues are detected at the prospective stage to ward off potentially deleterious outcomes before they occur. This may occur during the prior authorization and/or medication reconciliation review when drug-drug interactions (DDIs) may be realized for the patient prior to the dispensing of a new drug. Defined as when two or more different drugs interact and alter their intended effects, DDI may be easier than some other measures to implement in a DUR program. Occurring in complex pharmacodynamic and pharmacokinetic ways, as well as being dependent upon a myriad of factors, the full scope of DDIs is beyond the scope of this paper. Suffice it to say, the advent and adoption of clinical decision support systems that contain pre-programmed drug alerts for providers has revolutionized the ability to manage the DUR process more efficiently and effectively.

Implementing these systems, however, is not without risk, requiring robust resources and planning. Training personnel thoroughly on the software is as critical as teaching and reinforcing active surveillance of the alerts as they pertain to the safety of the patients. Additionally, the infrastructure must be setup properly with parameters adequately monitored and titrated to optimal sensitivity and threshold regarding alerts. Without this operational guidance, clinicians may encounter increased noise with a heavy volume of warnings and the undesirable consequence of alert fatigue. Under these circumstances, an important DDI may inadvertently go undetected.

Thus in establishing the "Safe Care" domain URAC reporting, DDI was the first major challenge tackled. For simplification purposes, DDI is bucketed into one category while the five other initial alerts to track, which include:

- Overutilization,
- Potential Abuse,
- Safety,

- Underutilization, and
- Other (mostly system-related, non-clinical alerts)

Are categorized into a non-DDI bucket. This ensures that while all six alerts are reported, there is better management of classification, delegation, and follow-through by various personnel across the organization.

BACKGROUND

I) DESCRIPTION OF FILES AND DATA ELEMENTS

Based upon preliminary analysis, four files are imported into SAS® (listed in quotations, names have been modified for simplicity; the original name of the file as used in the code is encased in parentheses):

1. “Patient Medical History” (MED_HX) *: provided by ScriptMed™, this file includes the patient’s self-reported history of medications, both prescribed and over-the-counter (OTC), that have been taken while an eligible member. Where known, start and end dates of the medication therapy are populated.
2. “Insurance Plan Crosswalk” (INSURANCE): identifies which pay plans correspond to which line of business.
3. “Drug Utilization Report (DUR)” (DDI): provided by ScriptMed™, the DUR includes the patient identifier, medication name, and MediSpan text alerts among other pharmacy attributes such as override date, pharmacist and reason.
4. “Shipped Drug Report” (CREEHAN_2017): provided by ScriptMed™, includes the patient identifier, medication name, shipped date and facility.

* This file was used for validation purposes only. When beginning the process, desired was verifying the accuracy of the MediSpan alerts tying to medications contained in the patient’s medical history. Therefore, a separate preliminary analysis was performed where for each patient, all the medications having a null end date (thus, assumed that the patient was still taking) were transposed and compared to the target medication, the medication most recently shipped from the specialty pharmacy directly related to the patient’s disease state (see Appendix for code).

PATIENT_ID	RX	Max_ShipDate	DRUGNAME	drugname1	drugname2	drugname3	drugname4	drugname5
		4-Jan-17	XTANDI 40MG CAPSULE	ASPIRIN LOW DOSE	GLUCOVANCE	LUPRON DEPOT	SIMVASTATIN	ZOLPIDEM TARTRATE

Figure 2: An example of a validation row taken from the data exported to the Excel spreadsheet

Note the shipped medication (column “DRUGNAME”), Xtandi, had a MediSpan drug alert text (Figure 2) regarding Simvastatin (column “drugname4”). Upon examining the patient’s medical history, Simvastatin was confirmed to be a medication (column “drugname4”) being concurrently taken (along with Aspirin, Glucovance, Lupron Depot, and Zolpidem Tartrate, columns “drugname1 – drugname3” and “drugname5”, respectfully).

METHODOLOGY

I) RETURN THE MAXIMUM SHIP DATE AND REFILL # FOR MEDICATION

The table “max_shipdate” is created using the “CREEHAN_2017” file. The maximum ship date and refill number are selected.

```
proc sql;
  create table SASUSER.max_shipdate as
  select cs.PATIENT_ID
        , cs.RX
        , cs.NDC
        , cs.DRUGNAME
        , cs.REFCAT
        , cs.PHARM_CODE
        , cs.PP
        , max(RF)           as Max_RF
        , max(Ship_Date)   as Max_ShipDate
  from SASUSER.CREEHAN_2017 cs
  group by cs.PATIENT_ID
        , cs.RX
        , cs.NDC
        , cs.DRUGNAME
        , cs.REFCAT
        , cs.PHARM_CODE
        , cs.PP;

quit;
```

II) JOIN DATA UPON COMMON ELEMENTS

The table “creehan_urac_ddi” is created using a DISTINCT statement and by performing an inner join upon the table “max_shipdate” and data sets “urac” and “INSURANCE” on the patient identifier, prescription number and pay plan identifier.

```
proc sql;
  create table SASUSER.creehan_urac_ddi as
  select distinct cs.PATIENT_ID
        , cs.RX
        , cs.Max_RF
        , cs.PP
        , cs.DRUGNAME
        , cs.REFCAT
        , cs.Max_ShipDate
        , cs.PHARM_CODE
        , us.Description as Alert_Desc
        , is.Plan_Type
  from SASUSER.max_shipdate cs
        , SASUSER.urac us
        , SASUSER.INSURANCE is
  where us.PATIENT_ID = cs.PATIENT_ID
        and us.RX      = cs.RX
        and cs.PP      = is.Unique_Plan_ID;

quit;
```

III) CLEANSING & CATEGORIZING THE DRUG ALERTS

As the MediSpan drug alerts are often punctuated with special characters as well as having several concatenated alerts contained within each observation, depending on how

extensive the patient's medical history is (Figure 1), the full alert needs to be "deconstructed" in a series of steps.

```
***** Drug Interactions ****Onset Code : Delayed Severity: Major Documentation : SuspectedPlasma concentrations and
pharmacologic effects of Simvastatin Oral Tablet 5 MG may be reduced by Xtandi Oral Capsule 40 MG.-----
----- Not Screened: DoseCheck: Data Not Available (index 0) :No dosing information available for
this drug.Xtandi Oral Capsule 40 MG XTANDI 40MG CAPSULE;00469012599 -----
-----
```

Figure 3: An example of a typical MediSpan drug alert text before cleansing

Using the functions UPCASE and COMPRESS, the drug alert text is converted to all uppercase letters and special character punctuation is stripped out of the string, respectively, for easier analysis.

Based upon the modified drug alert text, the six aforementioned flags are populated by counting the text contained in the criteria. A non-DDI flag ("NonDDI_flag") contains the sum of the five alerts, separate from the "DrugInt_flag" reserved for DDI only alerts. Finally, the pay plans are segmented into lines of business including "Commercial", "Medicaid", "Medicare", "PBM" and "Other".

```
data SASUSER.creehan_urac_ddi_clean (drop = pp);
set SASUSER.creehan_urac_ddi;
length desc_clean2 $1024 PType $20 NonDDI_flag 3.
      PotAbuse_flag Overtutil_flag Underutil_flag DrugInt_flag
      SafetyChk_flag Other_flag 3;
      desc_clean2      =      upcase(compress(alert_desc, "*!.","P"));
      DrugInt_flag     =      count(desc_clean2, "DRUG INTERACTIONS");
      PotAbuse_flag    =      count(desc_clean2, "POTENTIAL ABUSE");
      Overtutil_flag   =      count(desc_clean2, "HIGH DAILY DOSE ALERT")
+ count(desc_clean2, "HIGH DURATION DOSE ALERT");
      Underutil_flag   =      count(desc_clean2, "LOW DAILY DOSE ALERT")
+ count(desc_clean2, "LOW DURATION DOSE ALERT");
      SafetyChk_flag   =      count(desc_clean2, "PRIOR ADVERSE REACTIONS
ALERTS") + count(desc_clean2, "DUPLICATE THERAPY");
      Other_flag       =      count(desc_clean2, "NOT SCREENED")
+ count(desc_clean2, "PRECAUTION");
      NonDDI_flag      =      PotAbuse_flag + Overtutil_flag + Underutil_flag
+ SafetyChk_flag + Other_flag;
      if Plan_Type in('MEDICARE PART B', 'MEDICARE PART D')
      then PType       = 'MEDICARE';
      else if Plan_Type = 'PBM'
      then PType       = 'COMMERCIAL';
      else if Plan_Type = 'MEDICAID'
      then PType       = 'MEDICAID';
      else PType       = 'OTHER';
drop plan_type alert_desc pharm_code PotAbuse_flag Overtutil_flag
Underutil_flag SafetyChk_flag Other_flag;
run;
```

Some preliminary analysis is performed in SAS; however, the final file is exported into Excel where creating pivot tables provides easier manipulation and presentation options for stakeholders.

CONCLUSION

Although only scratching the surface towards implementation of full scale, on-going URAC accreditation reporting, this analysis lays the groundwork for key discussions on potential resource allocation and operational modifications. Regarding the former, further analysis may drill into the drug text alerts to understand in greater detail the “top” target, concurrent, and precipitant medications causing the highest volume of warnings. Results may allow technical staff to adjust the sensitivity of the MediSpan alerts so that clinical staff are not overwhelmed by DDIs deemed innocuous (currently, all DDIs are interpreted by the system as “severe”). This may allow greater productivity, lower queue loads of DDIs, and allow keener focus towards those warnings that may indeed cause harm and require intervention.

REFERENCES

1. Academy of Managed Care Pharmacy® (AMCP). “Drug Utilization Review”. 2009. Available at <http://www.amcp.org/WorkArea/DownloadAsset.aspx?id=9296>
2. Cody, Ronald. “An Introduction to SAS® Character Functions (Including Some New SAS® 9 Functions). SUGI Proceedings Paper 233-30. Available at <http://www2.sas.com/proceedings/sugi30/233-30.pdf>
3. Murphy, William C. “Squeezing Information out of Data”. SUGI Proceedings Paper 028-31. Rochester, NY. Available at <http://www2.sas.com/proceedings/sugi31/028-31.pdf>
4. ScriptMed™ Specialized Pharmacy Software. “MediSpan Solution and MediSpan Bridge Training Document”. 2015.
5. URAC. “Specialty Pharmacy – URAC”. 2016. Available at https://www.urac.org/wp-content/uploads/Spec_Rx_Measures-at-a-Glance_FINAL-20161114.pdf
6. Zirbel, Douglas. “Learn the Basics of Proc Transpose”. SAS Global Forum Paper 060-2009. Available at <http://support.sas.com/resources/papers/proceedings09/060-2009.pdf>

ACKNOWLEDGMENTS

The technical and clinical staff affiliated with the Clinical Quality Committee at Envolve Pharmacy Solutions (EPS) provided guidance and oversight of this analysis and reporting. Personal thanks to Elvin Slaughter, Director, Reporting and Business Analytics (EPS) and Bruno Berszoner, Manager, Data Analytics and Reporting, Celticare Health, for support in preparing this documentation.

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: 2 Executive Park Drive
City, State Zip: Bedford, New Hampshire 03110
Work Phone: 603-263-7130
E-mail: Karen.E.Wallace@centene.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

APPENDIX

```
proc sort data = SASUSER.MED_HX out = SASUSER.MED_HX_SORTED;
  by patient_id drug_short_name;
run;

proc transpose data = SASUSER.MED_HX_SORTED
  out = ddi_drugs (drop = _NAME_ _LABEL_)
  prefix = drugname_;
  by patient_id;
  var drug_short_name;
  where end_date = '';
run;
```