Using Arrays to Calculate Medication Utilization
Scott Leslie, MedImpact Healthcare Systems, Inc., San Diego, CA

ABSTRACT
Assessing duration of medication therapy involves managing a data set with multiple prescriptions per patient. This paper offers an innovative approach to calculating medication utilization as the proportion of days supplied over a specified time period. In this example, a few data steps use simple arrays to create multiple indicator variables which are then used to calculate medication utilization. Variations of this code can calculate gaps and overlaps in therapy as well as calculate concomitant medication utilization.

INTRODUCTION
Many health outcomes related to pharmacy utilization involve length of therapy measurements. The purpose of this paper is to offer code that calculates a patient’s medication utilization as the proportion of days medication is supplied over a time period. This code is a helpful start for building code to calculate additional outcome measures such as compliance, adherence, and persistence.

EXAMPLE 1: PROPORTION OF DAYS MEDICATION SUPPLIED
This example uses a pharmacy claims data set that has multiple observations per patient. The steps below calculate the number of days a single drug is supplied over a 180-day study period, with the date of first claim as the first day of study period. A cut of the data set shows 3 claims for a patient.

<table>
<thead>
<tr>
<th>Obs</th>
<th>member_id</th>
<th>fill_dt</th>
<th>drug</th>
<th>days_supply</th>
</tr>
</thead>
<tbody>
<tr>
<td>946</td>
<td>603</td>
<td>02/17/2005</td>
<td>a</td>
<td>30</td>
</tr>
<tr>
<td>947</td>
<td>603</td>
<td>06/13/2005</td>
<td>a</td>
<td>30</td>
</tr>
<tr>
<td>948</td>
<td>603</td>
<td>08/11/2005</td>
<td>a</td>
<td>30</td>
</tr>
</tbody>
</table>

Figure 1 illustrates the above data showing 3 claims and the 180-day study period. Date of first fill is the start of study period and end of study period is 180 days post first fill.

STEP 1
The first step is to transpose the data to a single observation per patient data set. This is done twice for the purposes of detailing the fill dates and corresponding days supply for each fill. It is essential to sort the data set by patient and fill date. Start and end dates for each subject are also calculated.

```sql
proc sort data=claims;
by member_id fill_dt;
run;
```
The result of the above code creates a patient level data set, showing the medication fill pattern and days supply for each fill. Note that missing values are given to those variables where the variable being transposed has no value in the input data set. That is, this patient has three claims; therefore the values for fill_dt3 and fill_dt4 are missing.

```
Obs  member_id  fill_dt1  fill_dt2  fill_dt3  fill_dt4  fill_dt5  

Obs  days_supply1  days_supply2  days_supply3  days_supply4  days_supply5  
265   30             30              30             .              .

Obs  start_dt  end_dt  
265  02/17/2005  08/15/2005
```

**STEP 2**

Next, a data step uses arrays and DO loops to find the days the patient was supplied the medication and calculates the proportion of days the medication was supplied in the review period. The first array, daydummy, creates a dummy variable for each day in the review period. The next two arrays, groups the fill_dt and days_supply variables setting up the DO loops. In this data set, the maximum number of fills incurred by a patient was 11 so there are 11 elements for these two arrays. One can set the number of elements to a value beyond the reasonable amount of fills or get the maximum number of fills in the data set from a proc contents procedure. The first do loop sets each dummy variable, daydummy, to 0. The second do loop uses an IF statement to flag the days of the review period that the patient was supplied the medication. Next, the variable dayscovered sums the daydummy variables. This sum is used as the numerator in calculating, p_dayscovered, the proportion of days medication was supplied in the 180 day study period, which is one of many measures for assessing compliance.

```
data pdc;
  set both;
  array daydummy(180) day1-day180;
  array filldates(*) fill_dt1 - fill_dt11;
  array days_supply(*) days_supply1-days_supply11;
  do ii=1 to 180; daydummy(ii)=0;end;
  do ii=1 to 180;
    do i = 1 to dim(filldates) while (filldates(i) ne .);
      if filldates(i)<= start_dt + ii -1 <= filldates(i)+days_supply(i)-1
        then daydummy(ii)=1;
    end;
  end;
  drop i ii;
  dayscovered=sum(of day1 - day180);label dayscovered='Total Days Covered';
  p_dayscovered=dayscovered/180;label p_dayscovered='Proportion of Days Covered';
  run;

proc print data=pdc;where member_id=603;run;
```
The result is a data set that has 180 dummy variables, one for each day of the time period, which indicates medication supplied. Only a few of the dummy variables are displayed below. In this example the patient’s last fill_date is on day 176 (see Figure 1 above) with a majority of the days supply for this claim extending beyond the study period. The claim is truncated and only 5 of days of this claim are included in the days covered count.

<table>
<thead>
<tr>
<th>Obs</th>
<th>member_id</th>
<th>day1</th>
<th>day2</th>
<th>day3</th>
<th>day4</th>
<th>day5</th>
<th><strong>day6-day29</strong></th>
<th>day30</th>
<th>day31</th>
<th>day32</th>
<th>day33</th>
</tr>
</thead>
<tbody>
<tr>
<td>265</td>
<td>603</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th><strong>day34-day115</strong></th>
<th>day116</th>
<th>day117</th>
<th>day118</th>
<th>day119</th>
<th><strong>day120-day145</strong></th>
<th>day146</th>
</tr>
</thead>
<tbody>
<tr>
<td>265</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th><strong>day147-day174</strong></th>
<th>day175</th>
<th>day176</th>
<th>day177</th>
<th>day178</th>
<th>day179</th>
<th>day180</th>
</tr>
</thead>
<tbody>
<tr>
<td>265</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>dayscovered</th>
<th>p_dayscovered</th>
</tr>
</thead>
<tbody>
<tr>
<td>265</td>
<td>65</td>
<td>0.36111</td>
</tr>
</tbody>
</table>

**EXAMPLE 2: CREDITING OVERLAPPING DAYS SUPPLY**

In this example, a patient refills their medication before exhausting the previous fill. Figure 2 shows this scenario, where the fourth claim, filled on 7/30/05, occurs before the end of supply of the third claim (8/5/05). Proportion of days supplied is calculated in the same manner and credits the subject with finishing the previous fill. This code is similar to the previous example with one extra step that identifies the overlapping days supply and shifts the fill date forward to the day after the end of supply of the previous fill.

**Figure 2: Medication Coverage with Overlapping Days Supply**

**ADDITIONAL STEP**

Using the same steps as in Example 1, adding this additional DO loop adjusts fill dates by shifting them forward. This starts with the second fill:

```r
do u=2 to 11 while (filldates(u) ne .);
    if filldates(u)<filldates(u-1)+days_supply(u-1)
        then filldates(u)=filldates(u-1)+days_supply(u-1);
end;
```
Shifting the fourth fill date credits the patient with 7 more days supply, increasing the days covered from 113 to 120 and increasing proportion of days supplied from to 62.8% to 66.6%. Larger differences would be seen in cases where a patient has multiple claims and multiple overlaps. These steps can be used to calculate length of therapy of multiple medications. To assess compliance to therapy of two medications, two sets of day dummy variables would be created and the DO loops would then flag the days that both medications were supplied.

CONCLUSION
This paper offers examples of calculating duration of therapy, as proportion of days supplied over a time period. Using dummy variables to specify treatment for each day of a review period, one can identify discontinuation of therapy and measure other outcomes such as compliance, adherence, and persistence.

REFERENCES


ACKNOWLEDGMENTS
The author would like to thank Patrick Thiebaud for explaining the concept of medication coverage and for his programming advice.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author at:
Scott Leslie
MedImpact Healthcare Systems, Inc.
10680 Treena Street
San Diego, CA  92131
Work Phone: 858-790-6685
Fax: 858-689-1799
Email: scott.leslie@medimpact.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.