GUI Method of Mapping SDTM from Raw Data

Elisa Natale
Martina Garlet

CROS NT, Verona, Italy
Introduction

Purpose

Improving and speeding up the mapping of SDTM from raw data, as the standard CDASH are not frequently used

How to reach the purpose

By using Excel as GUI support of SAS programming

Why Excel

More user-friendly interface, easier and faster to compile (especially for a junior programmer) and gives less opportunity for programming errors
Overall Process

1. Complete the mapping information for every SDTM domain into an Excel tool

2. Writing and run a SAS program for each domain:
   2.1 Recall of macro %createSDTM(domain=);
   2.2 Eventually fix the temporary dataset “union”;
   2.3 Recall of the final macro %createDATASET.
The Excel tool

Section A: Domain reference
Name of the domain (ex: QS, QS AE, all (key word))

Section B: Variables to create information
Name of the SDTM variables (ex: STUDYID, DOMAIN...)
Their format (Char or Num)

Section C: Input information
Name of the dataset (ex: libname.dsname or dsname)
Selection (ex: where CRIT='Y';)

Section D: Mapping codes
List of variables to transpose (When variable is ‘TRANSPOSE VAR’) 
Otherwise: Algorithms, Variables, String or just ‘same’ (key word)
### The Excel tool

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Variable</th>
<th>Type</th>
<th>Formula/Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>STUDYID</td>
<td>Char</td>
<td>&quot;&amp;Protocol_Number.&quot;</td>
</tr>
<tr>
<td>QS AE CM</td>
<td>USUBJID</td>
<td>Char</td>
<td>USUBJID=compress(&quot;&amp;Protocol_Number.&quot;</td>
</tr>
<tr>
<td>VS DM</td>
<td>USUBJID</td>
<td>Char</td>
<td>USUBJID=compress(&quot;&amp;Protocol_Number.&quot;</td>
</tr>
<tr>
<td>QS</td>
<td>data</td>
<td></td>
<td>data.visit1 where prteevst=1; data.visit5 where prteevst=2;</td>
</tr>
<tr>
<td></td>
<td>Subgroup</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QS</td>
<td>TRANSPOSE_VAR</td>
<td>Num</td>
<td>v1prteevP,v1prtese5, v1prteevU v5prteevP v5prtese5 prtee11--prtee2B4 prtee11--prtee2B4</td>
</tr>
<tr>
<td>QS</td>
<td>DOMAINE</td>
<td>Char</td>
<td>&quot;QS&quot; same same same</td>
</tr>
<tr>
<td>QS</td>
<td>QSTESTCD</td>
<td>Char</td>
<td>qstestcd=put(<em>name</em>,$qstcd.); same qstest=put(QSTESTCD,$qst.); same</td>
</tr>
<tr>
<td>QS</td>
<td>QSTEST</td>
<td>Char</td>
<td>qstest=put(QSTESTCD,$qst.); same</td>
</tr>
<tr>
<td>QS</td>
<td>QSCAT</td>
<td>Char</td>
<td>&quot;PRTEE QUESTIONNAIRE&quot; same same</td>
</tr>
<tr>
<td>QS</td>
<td>QSSCAT</td>
<td>Char</td>
<td>&quot;PRTEE SCORE&quot; same same</td>
</tr>
<tr>
<td>QS</td>
<td>QSORRES</td>
<td>Char</td>
<td>qsortor=strip(col1); same</td>
</tr>
<tr>
<td>QS</td>
<td>QSSTRESC</td>
<td>Char</td>
<td>qsortor=strip(col1); same</td>
</tr>
<tr>
<td>QS</td>
<td>QSSLFL</td>
<td>Char</td>
<td>&quot;1&quot; same same same</td>
</tr>
<tr>
<td>QS</td>
<td>VISITNUM</td>
<td>Num</td>
<td>&quot;VISIT&quot; same same</td>
</tr>
<tr>
<td>QS</td>
<td>VISIT</td>
<td>Char</td>
<td>&quot;VISIT 1&quot; same same</td>
</tr>
<tr>
<td>QS</td>
<td>QSDTC</td>
<td>Char</td>
<td>%Crea_date(d=v1dt_dd,m=v1dt %Crea_date(d=v5dt %Crea_date(d=prtee %Crea_date(d=prtee</td>
</tr>
<tr>
<td>VS</td>
<td>Data</td>
<td></td>
<td>input1 input1 input2</td>
</tr>
<tr>
<td>VS</td>
<td>Subgroup</td>
<td></td>
<td>where formName ne &quot;Neonatal&quot; and formName eq &quot;Neonatal Vital Signs&quot; and vsp in (&quot;Y&quot;</td>
</tr>
<tr>
<td>VS</td>
<td>TRANSPOSE_VAR</td>
<td>Num</td>
<td>hr, syssu, diasu, syss, diast, wt, hrm sys dia rr weigh headcir apgar</td>
</tr>
<tr>
<td>VS</td>
<td>USUBJID</td>
<td>Char</td>
<td>SITEID1=substr(siteID,1,4); usu same</td>
</tr>
<tr>
<td>VS</td>
<td>DOMAIN</td>
<td>Char</td>
<td>&quot;VS&quot; same same</td>
</tr>
<tr>
<td>VS</td>
<td>VSSCAT</td>
<td>Char</td>
<td>&quot;NEONATAL ASSESSMENTS&quot;</td>
</tr>
<tr>
<td>VS</td>
<td>VSTESTCD</td>
<td>Char</td>
<td>VSTESTCD=put(<em>NAME</em>,$VSTES same</td>
</tr>
</tbody>
</table>
The Program

Example of a program:

```r
%createSDTM(domain=SC);
/*Optional AD HOC steps*/
%createDATASET (datashell=SC, dataset=Union);
```

Variable xxSEQ:

Normally the program computes it automatically. If the variable is not needed just edit the call of the macro this way:

```r
%createSDTM(domain=DM, seq=N);
```
If no errors, then the draft dataset can be checked and finalized
The Macro %createSDTM

1. The import of the Excel dataset;
2. The creation of support datasets;
3. Work on each of the raw data separately:
   3a. The import into temporary library of the raw dataset;
   3b. The transpose of the dataset, if necessary;
   3c. The creation of the SDTM variables;
4. The union of all the modified input dataset into one, ‘Union’.
The Macro `%createSDTM`

1. EXCEL INPUT
2. CREATION SUPPORT DATASETS
   - RAW DATA #1 ELABORATION
   - RAW DATA #2 ELABORATION
   - ... 
   - RAW DATA #n ELABORATION
3. OUTPUT DATASET ‘UNION’

3A. RAW DATA
3B. VARIABLES TO TRANSPOSE?
   - Yes
   - TRANSPOSE
   - CREATION OF SDTM VARIABLES
   - No
Limits

1. It doesn’t work with Trial Domains and Supp Domains
2. Selection of records (Ex: IE domain)
3. The presence of variables in the raw data with the same name of SDTM variables to create
4. Hard to add new records

Solutions

1. Not required
2. Editing the dataset Union
3. The program automatically renames the variable in the raw data by adding an underscore before it
4. Uncommon for SDTM, possible with ad hoc space, maybe not to use this tool
Conclusion

- This process covers many situations
  - Many input datasets
  - Both structure of input datasets (either if they need to be transposed or not)
  - Every mapping way is possible (algorithms, string...)
- More efficient
  - Less code to write → less likely to make mistakes
  - Easier for a Junior programmer (more about SDTM theory)
Questions