MWSUG 2017 – TT06

From Device Text Data to a Quality Dataset

Laurie Smith, Cincinnati Children's Hospital Medical Center, Cincinnati, OH

ABSTRACT

Data quality in research is important. It may be necessary to obtain data from a device to be used a research. Often it is read from an external text file and entered onto a CRF. The data is then read from the CRF and entered into a database. This process introduces many opportunities for data quality to be compromised. The quality of device data used in a study can be greatly improved if the data can be read directly from a device's output file into a dataset. If the device outputs results into a text file that can be saved electronically, SAS® can be used to read the data needed from the results and saved into a dataset. In addition to improving data quality, data collection and monitoring time can also be reduced by taking advantage of these electronic files as opposed to recapturing this data on a CRF.

INTRODUCTION

The methods introduced will allow a SAS® Programmer, with basic SAS® programming skills, use SAS® to retain desired test results from a results report in fixed text format generated by a testing device.

EXTRACTING DATA FROM TEXT RESULTS REPORTS GENERATED BY A DEVICE

DEVICE DATA AS A TEXT FILE

Some devices generate text files containing results from testing. Each file is usually presented in a fixed format with one file per subject per test.

```
Test 2 RESULTS REPORT

Generated:                                    8/11/2017 1:25:03 PM
------------------------------------------------------------------

PATIENT INFORMATION

Patient Initials:                             LB
Patient ID:                                   0101
Patient gender:                               F
Birth date:                                   2/11/1973
Patient weight:                               91 kg
Patient height:                               178 cm
BSA:                                          2.12 m2
Heart rate:                                   88 bpm

SUMMARY

Test Result 1:                          11.12 cm/s
Test Result 2:                          1.70 mmHg
Test Result 3:                          801.00 ms
Test Result 4:                          2.42 l/min
Test Result 5:                          0.00 %
Test Result 6:                          33.62 ml/beat
Test Result 7:                          0.00 l/beat
```

Display 1. Example of Device data text file

The results may contain several results for a subject grouped by a particular characteristic.

```
Test 1 RESULTS REPORT

Generated:                                          8/11/2017 1:22:35 PM
-------------------------------------------------------------------

PATIENT INFORMATION

Patient Initials:                                   LB
Patient ID:                                         0101
Patient gender:                                     F
Birth date:                                         2/11/1973
Patient weight:                                     91 kg
Patient height:                                     178 cm
Heart rate:                                         88 bpm


LEFT REGION RESULTS

Body Surface Area:      2.12 m2
Test Result 1:          10.44 ml
Test Result 2:          7.54 ml/m2
Test Result 3:          8.33 ml
Test Result 4:          6.21 ml/m2
Test Result 5:          157.93 ml/min
Test Result 6:          20.05 %
Test Result 7:          19.27 g
Test Result 8:          14.02 g/m2
Test Result 9:          12.61 g/m


RIGHT REGION RESULTS

Body Surface Area:      2.12 m2
Test Result 1:          142.00 ml
Test Result 2:          103.62 ml/m2
Test Result 3:          72.55 ml
Test Result 4:          53.28 ml/m2
Test Result 5:          5.12 l/min
Test Result 6:          48.47 %
Test Result 7:          65.61 g
Test Result 8:          47.51 g/m2
Test Result 9:          43.10 g/m
```

Display 2. Example of Device data text file

USING SAS® TO EXTRACT DESIRED DATA

The process of manually entering this data onto a CRF, then into a database can be eliminated if SAS® is used to extract desired data directly from the text report.

It is best to present this program as a macro with a parameter equal to the name of the text file, since data for multiple subjects will most likely be imported.

Extraction from a simple report

Extracting data from a report as seen in Display 1 involves reading the txt file in a data step using an infile statement, starting at line 3 (where the data starts). For this extraction, only the testing date (Generated), Patient Initials, Subject ID (Patient ID), Test Heart Rate (Heart rate), Test Result 6, and Test Result 7 are needed for the final dataset. The subject ID is assigned to a macro variable to serve as an identifier for later use in the macro. The data step finds the line containing the desired data value, reads in the value and units and stores it in a character variable.

```
%macro Tst2(filename);
      data Tst2x;
```

```
            infile "(Data Location)\&filename..txt" firstobs=3 truncover
                   scanover;
            input
                  @'Generated:' itstdt $100.
                  @'Patient Initials:' iinit $100.
                  @'Patient ID:' isubjid $100.
                  @'Heart rate:' itst2hr $100.
                  @'Test Result 6:' itstrslt6 $100.
                  @'Test Result 7:' itstrslt7 $100.;
            call symput('subjid',strip(trim(isubjid)));
      run;
```

The next data step in the macro creates a subject specific dataset that isolates the numeric values from each character test result variable from the dataset above (dropping the character variables).

```
data tst2&subjid. (drop=iinit itstdt isubjid itstrslt6 itstrslt7 itst2hr);
            retain subjid;
            length  subjid $25. init $3.;
            format tstdt date9.;
            set tst2x;
      tstdt=input(scan(itstdt,1,' '),mmddyy10.);
      subjid=strip(trim(isubjid));
      init=strip(trim(iinit));
      tst2hr=input(scan(itst2hr,1,' '),best12.);
      tst2rslt6=input(scan(itstrslt6,1,' '),best12.);
      tst2rslt7=input(scan(itstrslt7,1,' '),best12.);
      run;
      proc sort data=tst2&subjid; by subjid tstdt; run;
%mend;
```

Call the macro with the filename as the parameter:

```
%Tst2(Subj 0101 LB Test 2);
```

Resulting in the following output:

subjid	init	tstdt	tst2hr	tst2rslt6	tst2rslt7
0101	LB	11AUG2017	88	33.62	0

Output 1. Test 2 Result

Extraction from a grouped report

Extracting data from a report as seen in Display 2 is similar to the above extraction except data must be extracted per section. The data values needed for the final dataset are the testing date (Generated), Patient Initials, Subject ID (Patient ID), Test Heart Rate (Heart rate), Left Region Test Result 1, Left Region Test Result 3, Left Region Test Result 6, Left Region Test Result 8, Right Region Test Result 1, Right Region Test Result 3, Right Region Test Result 6, and Right Region Test Result 8.

Since the test descriptions for the Left and Right Regions are the same, they will have to be extracted in separate data steps. The identifiers (testing date, Patient Initials, and Subject ID) will be extracted for Left and Right Regions such that the Left and Right Region data can be merged to create one dataset. Test Heart Rate will be extracted with the Left Region data only.

Left Region Extraction:

```
%macro Tst1lft(filename);
      data Tst1lftx;
            infile "(Data Location)\&filename..txt" firstobs=3 truncover
                  scanover;
```

```
            input
                  @'Generated:' itstdt $100.
                  @'Patient Initials:' iinit $100.
                  @'Patient ID:' isubjid $100.
                  @'Heart rate:' itst1hr $100.
                  @'LEFT' lft $100.;
                  if trim(lft)='REGION RESULTS' then do;
                        input
                              @'Test Result 1:' itstrslt1 $100.
                              @'Test Result 3:' itstrslt3 $100.
                              @'Test Result 6:' itstrslt6 $100.
                              @'Test Result 8:' itstrslt8 $100.;
                  end;
            call symput('subjid',strip(trim(isubjid)));
run;
```

Similar to the previous section, the next data step in the macro creates a subject specific dataset that isolates the numeric values from each character test result variable from the dataset above (dropping the character variables).

```
      data Tst1lft&subjid. (drop=itstdt iinit isubjid lft itstrslt1 itstrslt3
                                 itstrslt6 itstrslt8 itst1hr);
            retain subjid;
            length  subjid $25. init $3.;
            format tstdt date9.;
            set Tst1lftx;
      tstdt=input(scan(itstdt,1,' '),mmddyy10.);
      subjid=strip(trim(isubjid));
      init=strip(trim(iinit));
      tst1hr=input(scan(itst1hr,1,' '),best12.);
      lfttst1rslt1=input(scan(itstrslt1,1,' '),best12.);
      lfttst1rslt3=input(scan(itstrslt3,1,' '),best12.);
      lfttst1rslt6=input(scan(itstrslt6,1,' '),best12.);
      lfttst1rslt8=input(scan(itstrslt8,1,' '),best12.);
      run;
      proc sort data=Tst1lft&subjid; by subjid init tstdt; run;
%mend;
```

Call this macro with the filename as the parameter:

```
%Tst1rgt(Subj 0101 LB Test 1);
```

Resulting in the following output:

subjid	init	tstdt	tst1hr	lfttst1rslt1	lfttst1rslt3	lfttst1rslt6	lfttst1rslt8
0101	LB	11AUG2017	88	10.44	8.33	20.05	14.02

Output 2. Test 1 Left Region Result

Right Region Extraction macro:

```
%macro Tst1rgt(filename,subjid);
      data Tst1rgtx;
            infile "(Data Location)\&filename..txt" firstobs=3 truncover
                   scanover;
            input
                  @'Generated:' itstdt $100.
```

```
                  @'Patient Initials:' iinit $100.
                  @'Patient ID:' isubjid $100.
                  @'RIGHT' rgt $100.;
                  if trim(rgt)='REGION RESULTS' then do;
                        input
                              @'Test Result 1:' itstrslt1 $100.
                              @'Test Result 3:' itstrslt3 $100.
                              @'Test Result 6:' itstrslt6 $100.
                              @'Test Result 8:' itstrslt8 $100.;
                  end;
            call symput('subjid',strip(trim(isubjid)));
      run;

      data Tst1rgt&subjid. (drop=itstdt iinit isubjid rgt itstrslt1 itstrslt3
                                 itstrslt6 itstrslt8);
            retain subjid;
            length  subjid $25. init $3.;
            format tstdt date9.;
            set Tst1rgtx;
      tstdt=input(scan(itstdt,1,' '),mmddyy10.);
      subjid=strip(trim(isubjid));
      init=strip(trim(iinit));
      rgttst1rslt1=input(scan(itstrslt1,1,' '),best12.);
      rgttst1rslt3=input(scan(itstrslt3,1,' '),best12.);
      rgttst1rslt6=input(scan(itstrslt6,1,' '),best12.);
      rgttst1rslt8=input(scan(itstrslt8,1,' '),best12.);
      run;
      proc sort data=Tst1rgt&subjid; by subjid init tstdt; run;
%mend;
```

Call this macro with the filename as the parameter:

```
%Tst1rgt(Subj 0101 LB Test 1);
```

Resulting in the following output:

subjid	init	tstdt	rgttst1rslt1	rgttst1rslt3	rgttst1rslt6	rgttst1rslt8
0101	LB	11AUG2017	142	72.55	48.47	47.51

Output 3. Test 1 Right Region Result

Create final dataset

All extracted data for the subject can now be merged into one dataset. Again, defined as a macro where the parameter is the Subject ID, in order to more easily merge all data per subject for multiple subjects.

```
%macro mergeDData(subjid);
      data DeviceTextData&subjid;
            merge tst1lft&subjid.
                  tst1rgt&subjid.
                  tst2&subjid.;
      by subjid init tstdt;
label subjid='Subject ID'
        init='Subject Initials'
```

```
        tstdt='Test Date'
        tst1hr='Subject Test 1 Heart Rate'
        lfttst1rslt1='Left Region Test 1 Result 1'
        lfttst1rslt3='Left Region Test 1 Result 3'
        lfttst1rslt6='Left Region Test 1 Result 6'
        lfttst1rslt8='Left Region Test 1 Result 8'
        rgttst1rslt1='Right Region Test 1 Result 1'
        rgttst1rslt3='Right Region Test 1 Result 3'
        rgttst1rslt6='Right Region Test 1 Result 6'
        rgttst1rslt8='Right Region Test 1 Result 8'
        tst2hr='Subject Test 2 Heart Rate'
        tst2rslt6='Test 2 Result 6'
        tst2rslt7='Test 2 Result 7';
run;
%mend mergeDData;
```

Call this macro with the subid as the parameter:

```
%mergeDData(0101);
```

Resulting in the following output:

Subject ID	Subject Initials	Test Date	Subject Test 1 Heart Rate	Left Region Test 1 Result 1	Left Region Test 1 Result 3	Left Region Test 1 Result 6	Left Region Test 1 Result 8	Right Region Test 1 Result 1	Right Region Test 1 Result 3	Right Region Test 1 Result 6	Right Region Test 1 Result 8	Subject Test 2 Heart Rate	Test 2 Result 6	Test 2 Result 7
0101	LB	11AUG2017	88	10.44	8.33	20.05	14.02	142	72.55	48.47	47.51	88	33.62	0

Output 4. Final dataset for subject 0101

CONCLUSION

It is often practice to transcribe the results needed for research from device results in txt format from the text reports onto a CRF, where the data on the CRF are later entered into a database. Using SAS® to read the desired data values directly from the text report into a dataset can eliminate the need for this process.

This process can easily be adjusted to retain units for value conversions to one standard unit in case results provided are presented in different units.

CONTACT INFORMATION

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

Laurie Smith
Cincinnati Children's Hospital Medical Center
(513) 803-9001
laurie.bishop@cchmc.org