PharmaSUG China 2022 - Paper 132

Auto-Creation for Patient Profiles SAS Code

Yuhang Liu, Jie Liu BeiGene Ltd.

ABSTRACT

Patient Profiles(PP) is a report displaying all information of each subject. A clear and concise PP output can help medical monitors and safety team to have a quick glance on clinical data of each subject. However, it is very time consuming and tedious to produce patient profiles by reporting each EDC dataset. Therefore, it is worthy to explore a new method for creating patient profiles automatically.

Architect Loader Specification (ALS) file, which can be exported from RAVE EDC system, describes the database design and contains all metadata information of EDC datasets. This paper will take PP auto-creation as an example to give an attempt and practice on SAS program automated creation based on metadata by using SAS version 9.4 .

INTRODUCTION

PP program mainly contains two aspects about processing raw data and generating PDF file. PDF generation is stable and can refer to a fixed code, however, raw data value transformation process is irregular and changeable as it should follow the actual raw data. Traditional method for value transformation process is to draft form by form manually, which is complicated and time consuming. This paper proposes a method to automatically create PP program code, mainly focus on replacing the tedious and repetitive value transformation process by capturing information from ALS file.

MAIN IDEA

By capturing the metadata information of each field and form from ALS file, the product generates a series piece of code for variable value transformation, and then splices them into an integrated PP program. The product is consisted of 2 parts.

Part 1

The first part contains a program introduction and some macros which will be used in the generated PP code. The macros prepared here include:

- A macro to find the latest raw data for compare from previous data.
- A macro to compare current data with the latest raw data and derive a flag representing new, update or no change.
- A macro to transform date from date11. format into is8601da. format.
- A macro to create title and footnote.
- A macro to count observations for dataset.
- A macro to the table content in report.
- A macro to calculate first dose date and last dose date.
- A macro to generate PDF file.

Part 2

The second part involves to value transformation of each form.

Step 1. Import ‘Forms’ sheet and ‘Fields’ sheet to get ‘DraftFormActive’ information and ‘DraftFieldActive’ information. Select active forms and fields in EDC system.

Step 2. Import ‘MatrixX#MASTERDASHBOARD’ sheet and judge whether should form display ‘InstanceName’ information in final PP report according to the times of forms appears in folders. Specially, ‘InstanceName’ is necessary for ‘Unschduled Visit’ and ‘Survival follow-up’ forms by default.

Step 3. Scan EDC database and capture raw fields and unit information of each field. Then generate a piece of code for target PP code.

Figure 1. Marked below is the process code generated by step 3 for the final PP code.

```
*------------------------------------------------------------*
Coagulation - Local[LB_COAG]
*------------------------------------------------------------*;
%compare2data(  _data   =  LB_COAG,
                _keepvar=  SUBJECT SITEID RECORDID FOLDERNAME INSTANCEID FOLDERSEQ
                           LBPERF LBMRES LBDELYN LBDRES LBALYN LBALRES LBDAT_RAW PT_RAW INR_RAW APTT_RAW PPT_RAW);
data d0_LB_COAG;
    set LB_COAG;
    length subjid_ $50.;
    if recflag=. then call missing(recflag);
    subjid_=tranwrd(subject,'-','_');

    array var{*} $128. PPT_RAW_ APTT_RAW_ INR_RAW_ PT_RAW_;
    array unit{*} PPT_UN APTT_UN INR_UN PT_UN;
    array value{*} PPT_RAW APTT_RAW INR_RAW PT_RAW;
    do i=1 to dim(value);
        if strip(lowcase(value{i})) not in ('','na','nd','n/a') and strip(lowcase(unit{i})) not in ('','na','nd','n/a')
        then var{i}=strip(value{i})||' ('||strip(unit{i})||')';
        else var{i}=strip(value{i});
    end;

    label LBPERF="Was the Sample collected?"
        LBMRES="Reason for assessment missed"
        LBDELYN="Was the assessment delayed?"
        LBDRES="Reason for assessment delayed"
        LBALYN="Was the assessment done at auxiliary location?"
        LBALRES="Reason for assessment done at auxiliary location"
        LBDAT_RAW="Date sample collected"
        PT_RAW_="Prothrombin time"
        INR_RAW_="INR"
        APTT_RAW_="Activated Partial Thromboplastin Time"
        PPT_RAW_="Partial Thromboplastin Time"
        instancename='Visit'
        recordposition='Logline #';

    %datechange(indate=LBDAT_RAW,outdate=sort_d);
    keep subjid_ instancename sort_d folderseq recordid recordposition LBPERF LBMRES LBDELYN LBDRES
        LBALYN LBALRES LBDAT_RAW PT_RAW_ INR_RAW_ APTT_RAW_ PPT_RAW_ recflag;
proc sort;by subjid_ sort_d folderseq recordposition;run;
```

Step 4. Import ‘DataDictionaryEntries’ sheet and identify the other information as long as ‘CodedData’ of ‘DataDictionaryName’ containing ‘other’. Then generate a piece of code to combine field with other information in final PP code.

Figure 2. Marked below is the process code generated by step 4 for the final PP code.

```
*------------------------------------------------------------*
Subject Discontinuation from the Study[SD]
*------------------------------------------------------------*;
%compare2data(  _data    =  SD,
                _keepvar=  SUBJECT SITEID RECORDID FOLDERNAME INSTANCEID FOLDERSEQ
                        EOSRES DSSTDAT_RAW DSDECOD DSDECOTH DDDAT2_RAW DDRES DDRESO DDAUTYN);
data d0_SD;
    set SD;
    length subjid_ $50.;
    if recflag=. then call missing(recflag);
    subjid_=tranwrd(subject,'-','_');

    if compress(lowcase(DSDECOD),,'kad') in ('other','otherspecify') then DSDECOD_="Other: "||strip(DSDECOTH);
    else DSDECOD_=strip(DSDECOD);

    if compress(lowcase(DDRES),,'kad') in ('other','otherspecify') then DDRES_="Other: "||strip(DDRESO);
    else DDRES_=strip(DDRES);

    label EOSRES="Was the end of study related to COVID-19?"
        DSSTDAT_RAW="Date of Subject Discontinuation from the Study"
        DSDECOD_="Indicate primary reason for study discontinuation from the study"
        DDDAT2_RAW="Date of Death"
        DDRES_="Primary Cause of Death"
        DDAUTYN="Was an autopsy performed?"
        instancename='Visit'
        recordposition='Logline #';

    %datechange(indate=DSSTDAT_RAW,outdate=sort_d);
    keep subjid_ instancename sort_d folderseq recordid recordposition EOSRES DSSTDAT_RAW DSDECOD_
    DDDAT2_RAW DDRES_ DDAUTYN recflag;
proc sort;by subjid_ sort_d folderseq recordposition;run;
```

Step 5. Import ‘Fields’ sheet and identify integrated field whose ‘PreText’ value contains ‘check all that apply’ and followed by some fields with ‘ControlType’ value equal to CheckBox. And generate a piece of code to integrate fields which could be concatenated into one field to show the highly integrated informatio.

Figure 3. Marked below is the process code generated by step 5 for the final PP code.

```
*------------------------------------------------------------*
Demographics[DM]
*------------------------------------------------------------*;
%compare2data(  _data    =  DM,
                _keepvar=  SUBJECT SITEID RECORDID FOLDERNAME INSTANCEID FOLDERSEQ
                        BRTHDAT_RAW AGE_RAW SEX DMCBP ETHNIC RACAIAN_RAW RACASN_RAW RACAIND_RAW RACACHI_RAW RACAFIL_RAW RACAJAP_RAW
                        RACAKOR_RAW RACAVIT_RAW RACAOTH_RAW RACAOSPY RACBAA_RAW RACNHOPI_RAW RACWC_RAW RACNR_RAW RACUNK_RAW RACOTH_RAW RACEOTH);
data d0_DM;
    set DM;
    length subjid_ $50.;
    if recflag=. then call missing(recflag);
    subjid_=tranwrd(subject,'-','_');

    if strip(RACAOTH_RAW) in ('1','Yes') then RACAOTH_RAW_="Other: "||strip(RACAOSPY);
    else RACAOTH_RAW_='';

    if strip(RACOTH_RAW) in ('1','Yes') then RACOTH_RAW_="Other: "||strip(RACEOTH);
    else RACOTH_RAW_='';

    length RACASN_RAW_ $12000. ;
    array _RACASN_RAW_{*} RACAIND_RAW RACACHI_RAW RACAFIL_RAW RACAJAP_RAW RACAKOR_RAW RACAVIT_RAW RACAOTH_RAW_;
    do i=1 to dim(_RACASN_RAW_);
        if _RACASN_RAW_{i}='1' then RACASN_RAW_=catx(',',RACASN_RAW_,tranwrd(vlabel(_RACASN_RAW_{i}),'(Character)',''));
        if substr(vname(_RACASN_RAW_{i}),length(vname(_RACASN_RAW_{i})))='_' then RACASN_RAW_=catx(',',RACASN_RAW_,_RACASN_RAW_{i});
    end;
    if not missing(RACASN_RAW_) then RACASN_RAW_='Asian:'||RACASN_RAW_;

    length RACAIAN_RAW_ $12000. ;
    array _RACAIAN_RAW_{*} RACAIAN_RAW RACASN_RAW_ RACBAA_RAW RACNHOPI_RAW RACWC_RAW RACNR_RAW RACUNK_RAW RACOTH_RAW_;
    do i=1 to dim(_RACAIAN_RAW_);
        if _RACAIAN_RAW_{i} in ('1','Yes') then RACAIAN_RAW_=catx(';',RACAIAN_RAW_,tranwrd(vlabel(_RACAIAN_RAW_{i}),'(Character)',''));
        if substr(vname(_RACAIAN_RAW_{i}),length(vname(_RACAIAN_RAW_{i})))='_' then RACAIAN_RAW_=catx(';',RACAIAN_RAW_,_RACAIAN_RAW_{i});
    end;
```

Step 6. For other fields with ‘ControlType’ is CheckBox, the product also generates a piece of code to translate ‘1/0’ into ‘Yes/No’.

Figure 4. Marked below is the process code generated by step 6 for the final PP code.

```
*------------------------------------------------------------*
ECG Local Read (triplicate)[EG]
*------------------------------------------------------------*;
%compare2data(  _data    =  EG,
                _keepvar=  SUBJECT SITEID RECORDID FOLDERNAME INSTANCEID FOLDERSEQ
                           EG1PERF EGMRES EGDELYN EGDRES EGALYN EGALRES EGT1TPT EG1ND_RAW EGT1DAT_RAW EG1QTCF_RAW);
data d0_EG;
    set EG;
    length subjid_ $50.;
    if recflag=. then call missing(recflag);
    subjid_=tranwrd(subject,'-','_');

    if strip(EG1ND_RAW)='1' then EG1ND_RAW_='Yes';
    else if strip(EG1ND_RAW)='0' then EG1ND_RAW_='No';
    else EG1ND_RAW_=' ';

    label EG1PERF="Was a 12-Lead ECG performed?"
        EGMRES="Reason for assessment missed"
        EGDELYN="Was the assessment delayed?"
        EGDRES="Reason for assessment delayed"
        EGALYN="Was the assessment done at auxiliary location?"
        EGALRES="Reason for assessment done at auxiliary location"
        EGT1TPT="Timepoint"
        EG1ND_RAW_="Not Done"
        EGT1DAT_RAW="Date of Assessment"
        EG1QTCF_RAW="QTcF interval (msec)"
        instancename='Visit'
        recordposition='Logline #';

    %datechange(indate=EGT1DAT_RAW,outdate=sort_d);
    keep subjid_ instancename sort_d folderseq recordid recordposition EG1PERF EGMRES EGDELYN EGDRES EGALYN EGALRES
         EGT1TPT EG1ND_RAW_ EGT1DAT_RAW EG1QTCF_RAW recflag;
proc sort;by subjid_ sort_d folderseq recordposition;run;
```

Step 7. Import 'CoderConfiguration' sheet to select those fields with coding information. And then, generate a piece of code to combine raw data with coding information.

Figure 5. Marked below is the process code generated by step 7 for the final PP code.

```
*------------------------------------------------------------*
Medical History[MH]
*------------------------------------------------------------*;
%compare2data(  _data    =  MH,
                _keepvar=  SUBJECT SITEID RECORDID FOLDERNAME INSTANCEID FOLDERSEQ
                           MHTERM MHSTDAT_RAW MHONGO MHENDAT_RAW MHTOXGR);
data d0_MH;
    set MH;
    length subjid_ $50.;
    if recflag=. then call missing(recflag);
    subjid_=tranwrd(subject,'-','_');

    length MHTERM_ $8000.;
    MHTERM_='Prior medical history: '||strip(MHTERM)||';`nSOC: '||strip(MHTERM_SOC)||';`nPT: '||strip(MHTERM_PT)||';';

    label MHTERM_="Prior medical history;~System Organ Class;~Preferred Term;"
        MHSTDAT_RAW="Date Started"
        MHONGO="Ongoing at the time of first dose?"
        MHENDAT_RAW="Date Stopped"
        MHTOXGR="NCI-CTC severity grade at baseline"
        instancename='Visit'
        recordposition='Logline #';

    %datechange(indate=MHSTDAT_RAW,outdate=sort_d);
    keep subjid_ instancename sort_d folderseq recordid recordposition MHTERM_ MHSTDAT_RAW MHONGO MHENDAT_RAW MHTOXGR recflag;
proc sort;by subjid_ sort_d folderseq recordposition;run;
```

Step 8. Generate a piece of code to re-assigns label for each field.

Figure 6. Marked below is the process code generated by step 8 for the final PP code.

```
*------------------------------------------------------------*
Medical History[MH]
*------------------------------------------------------------*;
%compare2data(  _data    =  MH,
                _keepvar=  SUBJECT SITEID RECORDID FOLDERNAME INSTANCEID FOLDERSEQ
                        MHTERM MHSTDAT_RAW MHONGO MHENDAT_RAW MHTOXGR);
data d0_MH;
    set MH;
    length subjid_ $50.;
    if recflag=. then call missing(recflag);
    subjid_=tranwrd(subject,'-','_');

    length MHTERM_ $8000.;
    MHTERM_='Prior medical history: '||strip(MHTERM)||';`nSOC: '||strip(MHTERM_SOC)||';`nPT: '||strip(MHTERM_PT)||';';

    label MHTERM_="Prior medical history;~System Organ Class;~Preferred Term;"
        MHSTDAT_RAW="Date Started"
        MHONGO="Ongoing at the time of first dose?"
        MHENDAT_RAW="Date Stopped"
        MHTOXGR="NCI-CTC severity grade at baseline"
        instancename='Visit'
        recordposition='Logline #';

    %datechange(indate=MHSTDAT_RAW,outdate=sort_d);
    keep subjid_ instancename sort_d folderseq recordid recordposition MHTERM_ MHSTDAT_RAW MHONGO MHENDAT_RAW MHTOXGR recflag;
proc sort;by subjid_ sort_d folderseq recordposition;run;
```

Step 9. Captures the first time-variable within forms in Fields sheet as the ordering variable, and if there is no time-variable, InstanceName instead. All the time-variables will be outputted to an excel file for manual check. Then, the product will generate a piece of code for sorting.

Figure 7. Marked below is the process code generated by step 9 for the final PP code.

```
*------------------------------------------------------------*
Substance Use[SU]
*------------------------------------------------------------*;
%compare2data(  _data    =  SU,
                _keepvar=  SUBJECT SITEID RECORDID FOLDERNAME INSTANCEID FOLDERSEQ
                        SUNCF SUTRT);
data d0_SU;
    set SU;
    length subjid_ $50.;
    if recflag=. then call missing(recflag);
    subjid_=tranwrd(subject,'-','_');

    label SUNCF="Substance use?"
        SUTRT="Category of Substance"
        instancename='Visit'
        recordposition='Logline #';

    if prxmatch('/\d\d?\s*\-*[a-zA-Z]{3}\s*\-*\d\d\d\d/',instancename) then do;
        b=prxparse('/\d\d?\s*\-*[a-zA-Z]{3}\s*\-*\d\d\d\d/');
        call prxsubstr(b,instancename,start,length);
        sort_d_=substr(strip(instancename),start,length);
        %datechange(indate=sort_d_,outdate=sort_d);
    end;
    else sort_d=put(input(compress(instancename,,'kd'),best.),z8.);
    keep subjid_ instancename sort_d folderseq recordid recordposition SUNCF SUTRT recflag;
proc sort;by subjid_ folderseq sort_d recordposition;run;
```

Step 10. Split a form into parts as some forms contain too many fields to display in one page.

Figure 8. Marked below is the process code generated by step 10 for the final PP code.

```
*------------------------------------------------------------*
ECG Local Read (triplicate)[EG]
*------------------------------------------------------------*;
%compare2data(  _data    =  EG,
                _keepvar=  SUBJECT SITEID RECORDID FOLDERNAME INSTANCEID FOLDERSEQ
                        EG1PERF EGMRES EGDELYN EGDRES EGALYN EGALRES EGT1TPT EG1ND_RAW EGT1DAT_RAW EG1QTCF_RAW);
data d0_EG;
    set EG;
    length subjid_ $50.;
    if recflag=. then call missing(recflag);
    subjid_=tranwrd(subject,'-','_');

    if strip(EG1ND_RAW)='1' then EG1ND_RAW_='Yes';
    else if strip(EG1ND_RAW)='0' then EG1ND_RAW_='No';
    else EG1ND_RAW_=' ';

    label EG1PERF="Was a 12-Lead ECG performed?"
        EGMRES="Reason for assessment missed"
        EGDELYN="Was the assessment delayed?"
        EGDRES="Reason for assessment delayed"
        EGALYN="Was the assessment done at auxiliary location?"
        EGALRES="Reason for assessment done at auxiliary location"
        EGT1TPT="Timepoint"
        EG1ND_RAW_="Not Done"
        EGT1DAT_RAW="Date of Assessment"
        EG1QTCF_RAW="QTcF interval (msec)"
        instancename='Visit'
        recordposition='Logline #';

    %datechange(indate=EGT1DAT_RAW,outdate=sort_d);
    keep subjid_ instancename sort_d folderseq recordid recordposition EG1PERF EGMRES EGDELYN EGDRES EGALYN EGALRES
         EGT1TPT EG1ND_RAW_ EGT1DAT_RAW EG1QTCF_RAW recflag;
proc sort;by subjid_ sort_d folderseq recordposition;run;

data    d0_EG_part1(keep=subjid_ recordposition instancename EG1PERF EGMRES EGDELYN EGDRES EGALYN EGALRES recflag)
        d0_EG_part2(keep=subjid_ recordposition instancename EGT1TPT EG1ND_RAW_ EGT1DAT_RAW EG1QTCF_RAW recflag);
    set d0_EG;
run;
```

Step 11. Add a comparison function between current data with latest data and generate a flag according to comparison results. And there is also a macro variable to control whether need to execute comparison function.

Figure 9. Marked below is the process code generated by step 11 for the final PP code.

```
*------------------------------------------------------------*
Substance Use[SU]
*------------------------------------------------------------*;
%compare2data(  _data    =  SU,
                _keepvar=  SUBJECT SITEID RECORDID FOLDERNAME INSTANCEID FOLDERSEQ
                        SUNCF SUTRT);
data d0_SU;
    set SU;
    length subjid_ $50.;
    if recflag=. then call missing(recflag);
    subjid_=tranwrd(subject,'-','_');

    label SUNCF="Substance use?"
        SUTRT="Category of Substance"
        instancename='Visit'
        recordposition='Logline #';

    if prxmatch('/\d\d?\s*\-*[a-zA-Z]{3}\s*\-*\d\d\d\d/',instancename) then do;
        b=prxparse('/\d\d?\s*\-*[a-zA-Z]{3}\s*\-*\d\d\d\d/');
        call prxsubstr(b,instancename,start,length);
        sort_d_=substr(strip(instancename),start,length);
        %datechange(indate=sort_d_,outdate=sort_d);
    end;
    else sort_d=put(input(compress(instancename,,'kd'),best.),z8.);
    keep subjid_ instancename sort_d folderseq recordid recordposition SUNCF SUTRT recflag;
proc sort;by subjid_ folderseq sort_d recordposition;run;
```

This part will generate a completed Data procedure for each form eventually.

Finally, the product splices these codes from part 1 and part 2 and generates a Rich Text Format (RTF) file. You just need copy the content to a new SAS program and you will get the final PP program with little modification.

CONCLUSION

In this work, we have proposed a method that generates the PP program efficiently. With loading the latest ALS file, this method is applicable for any studies and for any updates.

CONTACT INFORMATION

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

Yuhang Liu
BeiGene
Email: yuhang.liu@beigene.com

Jie Liu
BeiGene
Email: jie1.liu@beigene.com