PharmaSUG 2021 - Paper DV-086

Creating Patient Profiles Using SAS® Software: Proc Report & SAS ODS

Ballari Sen, Agios Pharmaceuticals,INC.

ABSTRACT

Patient profiles provide comprehensive information for a single subject participating in a clinical study.
The profile template includes relevant clinical data for a subject that can help understand adverse events, concomitant medications, exposure, lab findings and other significant events and findings as a narrative or a visual report.

The Reports are constructed to convey information about a single patient in a concise output within a company for data review. It acts as a key to identifying why a subject experienced adverse event, why the subject took concomitant medication or the effects of dosage on a subject for the investigational treatment. These different requirements and specifications could only be possible through certain amount of customized programming.

SAS provides rich procedures to efficiently create these important reports. This paper will discuss how to produce patient profiles utilizing SAS® software proc procedures: PROC REPORT with the data Step. These patient profiles can serve as a reliable and effective data output source for references in writing a study report, analyzing output for communicating between data management and clinical departments. This paper also utilizes SAS® ODS ability to sort out the actual data and create the output in PDF document with bookmarks embedded in the profile template.

INTRODUCTION

Patient Profile listing are created by programmers for internal medical data review. These patient profiles provide the regulatory authorities the ability to view the listing available for each patient in a clinical study to check data consistency and to ensure high quality. The domain information is usually stored in different datasets and data type formats. Some domain datasets may contain only one record per subject while others may contain multiple records per subject. To further complicate matters, different data standards are implemented for the appearance of patient profile tables and listings, such data demands lend themselves to multi-data presentation that can be easily programmed utilizing ODS & PROC REPORT in SAS [5].

SAS® ODS software provides a means for creating attractive and concise patient profile report listings. ODS being an object-oriented technology, provides greater flexibility in generating, storing, and reproducing SAS procedure and DATA step output and encourages programming beyond template descriptions to creation of complete output layouts. Together with ODS, PROC REPORT provides a fully customizable framework for displaying table data in a format within the SAS system [6].

DESCRIPTION

This paper represents data in a format similar to that of the Clinical Data Interchange Standards Consortium (CDISC) Study Data Tabulation Model (SDTM).
The profile template includes only four relatively simple tables. Each of the patient profile template represents one of the three general observation class: Interventions, Events & Findings & also the subject-level data [2].
The first table includes patient-level demographic information: Including Actual Arm Description, Age, Sex, Weight, Race & Date/Time of Informed Consent.
The next table represents exposure to study-drug dosage information: Cycle/Visit Information, Start/End date of treatment, Dosage, Dosage Units. The final two tables present planned protocol milestone information (e.g. Randomization, study completion and occurrences, conditions, or incidents independent of planned study

evaluations occurring during the trial (e.g., adverse events) or prior to the trial (e.g., medical history)) [2] & different types of findings (laboratory & vital signs).

METHOD

The output will be produced utilizing SAS® software ODS PDF File coding and profile template created in PROC REPORT.

Step 1: Define the Program Style Template.

- The first step in the programming process is to define the PROC REPORT style.
  The below code defines:
  - The protocol Name
  - The Company Name
  - Creation Date of the SDTM Compliant datasets
  - where the profile template will be stored (Out-path location)
  - Creation Date of the patient profile was implemented & programmed.
  - Site name for the study.
  - ARM for the study.
  - SEX, RACE & AGE for a particular subject in the patient profile.
  - Exposure to Study Drug Start Date.
  - Exposure to Study Drug End Date.
  - Protocol Version of the study.

```
title; footnote;
%let protocol= %str(XXXXX-X-XXX);
%let company = %str(XXXXXX XXXXX);
%let crdate='';
%let outpath=%str(:\XXXX\XXXXX\XXXX\);
%let datasetdate='23APRIL2020';
%let sitename='';
%let arm='';
%let sex='';
%let race='';
%let age =' ';
%let dosedate = ' ';
%let doseendte = ' ';
%let protvers = ' ';
```

Step 2: Program Datasets Creation Date

- The next step is to program the datasets creation date populated from the patient-level demographics dataset.

```
proc contents data=work.dm noprint out=crdate(keep=crdate); run;
proc sql;
create table work.crdate as
select put(datepart(cdate),date11.) || ' ' || put(timepart(cdate),hhmm.)
as crdate
from (select max(crdate) as cdate from crdate);
quit;

%put &crdate;
data _null_;
set CDATE;
call symput ("crdate", crdate);
```

```
run;
```

Step 3: Produce the Patient Profile Report in PDF Document with Bookmarks

- The third step is to produce the patient profile for a single patient.
- The DO-While loop iteration will include all the domain records for a particular subject in the profile template utilizing the subjcount variable from the Headerrecord dataset.
- Dataset Headerrecord dataset is to produce the header information for the patient profile template: usubjid, sitename,sex,race,age,protocolversion,study Drug exposure start date & last dosage date.
- Programming the output in PDF Document by implementing ODS PDF File procedure.
- Defining the header & Footnote template Format for the patient profile.

```
  /**Macro Loopoverdatasets will run a single patient profile
                      for USUBJID=X ***/

               %macro loopOverDatasets();
                   proc sql noprint;

   /**Output Unique subject count into the macro variable**/

      select count(*) into:subjCount from work.Headerrecord;
                         quit;

  /*Initiate the loop iteration for populating all the
   domain records for a subject in the patient profile*/
                     %let iter=1;

    /*let subjCount=1;/*Produces one or two patient
                  profile for testing********/

            %do %while (&iter.<= &subjCount.);

 /** Dataset HeaderRecord produces the header information for
  the patient profile template / **Defined in Step 1 above***/;

              data _null_;
              set work.Headerrecord(firstobs=&iter. obs=&iter.);
              call symputx("usubjid",'usubjid'n);
              call symputx("sitename ",'sitename'n);
              call symputx("sex",'sex'n);
              call symputx("race",'race'n);
              call symputx("age",'age'n);
               call symputx("protvers",'protvers'n);
              call symputx("arm",'arm'n);
              call symputx("assndose",'assndose'n);
              call symputx("exstdtc",'exstdtc'n);
              call symputx("lddtc",'lddtc'n); run;

  /** Setup the output path utilizing ODS PDF File*/
                       ods listing close;
                       ods escapechar '^';
  options orientation =landscape nodate nonumber nobyline nocenter;
                       ods noresults;
/**Producing the output in PDF Document with bookmarks attached **/

 ods pdf file ="&outpath\output_PDF\&USUBJID.&datacut..pdf" contents=yes
                          bookmarklist=hide ;
```

```
options orientation=landscape pdfpageview=FITPAGE nocenter nobyline;

/*Header Template Information: J = means Justify, L= Left, C= centre,
                        R = Right******/
      /** ^ is the ODS escape character defined above */

             title1 height=1 j=l "&company." j=r "Site Name
                     %sysfunc(strip(&SITENAME.))";

              title2 height=4 bold j=l "Patient Profiles"
                    j=r "Subject:%sysfunc(strip(&usubjid.))";

                title4 j=l "Sex : %sysfunc(strip(&Sex.))"
           j=c "Protocol Version : %sysfunc(strip(&PROTVERS.))"
                                j=r " ";


              title5 j=l "Race : %sysfunc(strip(&Race.))"
              j=c "Treatment Arm : %sysfunc(strip(&ARM.))"
            j=r "First Dose Date : %sysfunc(strip(&EXSTDTC.))";

            title6 j=l "AGE : %sysfunc(strip(&AGE.))"
            j=r "Last Dose Date : %sysfunc(strip(&LDDTC.))";


       /**Footnote Information: Included the Dataset
      Creation date & the patient profile creation Date**/
            footnote1 height=1 j=l "Dataset Date: &CRDATE." j=r
            "Created on: %sysfunc(date(),date11.)
            %sysfunc(time(),hhmm.) (Confidential)";
```

Step 4: Define Table Template

- The next step defines the three table templates. The first template (Work.DM) represents the patient level demographic data (Figure 1) with 12 columns (12 variables).
- The Second template (work.EX) represents the exposure to study drug data (Figure 2) .
- The third template & fourth template (Work.AE) & (Work.LB) represents Adverse events & Lab findings data.
- Each column relates to a specific variable in the data and will have a column heading predefined in the template.
- The code for these three templates is shown below:

```
 /*Setting the number of rows which displayed in the patient profile*/
                     %let rownum=15;
/**Programming to Merge Main domain dataset & Suppqual domain**/


                %macro SUPP2PAR(list=);
           %do i=1 %to %sysfunc(countw(&list.));
           %let domain = %scan(&list,&i,' ');

  /**Defining the Domain & Suppqual as DM & SUPPDM as an example***/

              %if &domain=DM %then %do;
              proc sort data=sdtm.suppdm out=suppdm;
              by studyid rdomain usubjid; run;
              proc sort data=sdtm.dm out=dm;
```

```
                by studyid domain usubjid; run;

                proc transpose data=suppdm(rename=(RDOMAIN=domain))
                out= suppdmt(drop=_NAME_ _LABEL_);
                by studyid domain usubjid;
                var qval;
                id qnam;
                idlabel qlabel; run;

      /**Merging the main domain and Suppqual domain**/
                   data dm;
                   merge dm suppdmt;
                   page=1;
                   by studyid domain usubjid;
                   run; %end; %else %do;
     /*Programming to run the Macro SUPP2PAR for all the
      other domains as per study (Non-Subject Level) datasets**/
            proc sort data=SDTM.SUPP&domain out=SUPP&domain;
            by studyid rdomain usubjid idvarval ; run;
            proc sort data=sdtm.&domain out=&domain;
            by studyid domain usubjid &domain.seq; run;


       proc transpose data=supp&domain(rename=(rdomain=domain))
out=supp&domain.T(drop=_NAME_ _LABEL_);
       by studyid domain usubjid idvarval; var qval;
       id qnam; idlabel qlabel; run ;



              data supp&domain.T;
              set supp&domain.T;
              &domain.seq=input(idvarval, best.); run;

        proc sort data=supp&domain.T; by studyid domain usubjid
&domain.SEQ; run;

              data &domain;
              merge &domain(in=in1) supp&domain.T;
              by studyid domain usubjid &domain.seq;

              if in1; run;

         /**Programming Events Class: Adverse Events &
       Findings class : Laboratory Findings as an example**/

                %if &domain=sdtm.ae %then %do;
                proc sort data=ae;
                by usubjid aeterm aestdtc; run;

                      data &domain;
                      set &domain;
                      by usubjid aeterm aestdtc;
                      if first.usubjid then count=0;
                      count+1;
                      page=ceil(count/&rownum);
                      run;
                   %end;
                   %else %if &domain= sdtm.LB %then %do;
```

```
                proc sort data=sdtm.lb;
               by usubjid visitnum lbtestcd; run;

                      data &domain;
                      set &domain;
                      by usubjid visitnum lbtestcd;
                      if first.usubjid then count=0;
                      count+1;
                      page=ceil(count/&rownum);
                      run;
                     %end;
                  %else %do ;

                 proc sort data=&domain; by USUBJID &domain.SEQ; run;
                      data &domain;
                      set &domain;
                      by USUBJID &domain.SEQ;
                      if first.usubjid then count=0;
                      count+1;
                      page=ceil(count/&rownum);
                      run;
                     %end;
                     %end;
                     %end;
                   %mend SUPP2PAR;

/*Call Macro SUPP2PAR & List the all the domains to be included in the
patient profile template*/


  %SUPP2PAR(list=%str(dm ae ex lb));

/**Provide the header information for the patient Profiles**/

     Proc sql;
     create table work.Headerrecord as
     select t1.usubjid,T1.sex,T1.rficdtc,T1.race,T1.age,s
     t1.sitename,t5.lddtc,
     case
     when t2.protvers is not null then left(trim(t2.protvers))
     else '-'
     end as protvers label = "Protocol Version",
     case
     when t1.arm is not null then left(trim(T1.arm))
     else '-'
     end as arm label = "Treatment Arm",
     case
     when t3.exdose is not null
     then
     left(trim(t3.extrt)) || '   ' ||compress(left(trim(put(T3.exdose,8.)))||
left(trim(t3.exdosu)))||'  ' || left(trim(t3.exdur))
     else '-'
     end as ASSNDOSE label = "Assigned Dose",
     case
     when t4.exstdtc = '' then '-'
     else t4.exstdtc
     end as exstdtc label = "First Dose Date",
     case
```

```
     when t5.lddtc = '' then '-'
     else t5.lddtc
     end as dsstdtc label = "Last Dose Date"
     from work.dm T1
     left join sdtm.ds t2 on (t1.usubjid = t2.usubjid)
     left join sdtm.EX t3 on (t1.usubjid = t3.usubjid)
     left join sdtm.ex_day1 t4 on (t1.usubjid = t4.usubjid)
     left join sdtm.DS_day t5 on (t1.usubjid = t5.usubjid) ;
     quit;

                /** Patient-Level Table **;/
                    /**Demographics**/
   /** Populating Height and Baseline Weight variable from the VS SDTM **/
                proc sql;
                create table vs_htwt as select distinct
                usubjid,height_raw,weight_raw from
                sdtm.vssdtm;
                run;

/*Programming a left join between Demographics dataset & VSSDTM dataset to
populate only the height & weight variable from VS_HTWT that matches with
the(i.e. Baseline Height & Weight value) DM_Join dataset*/

          proc sql;
          create table dm_join as select A.*,B.height_raw,B.weight_raw
          from SDTM.DM A left join vs_htwt B
          on A.subjid = B.subject;
          run;


/**Renaming Variable Height & Weight Variable and storing it in the work
library**/

      Data work.dm_f(rename= (height_raw = height weight_raw = weight));
      SET dm_join;
      label Height="Height";
      label Weight="Weight";
      run;
/**Programing PROC REPORT for the Demographics data**/

ods proclabel ='Demographics';
proc report data= WORK.DM_F center nowd headline headskip spacing=1
split="|" missing
style(header)={background=pink}style(report)={outputwidth=100%};
where usubjid="&usubjid";
column ("^S={borderbottomcolor=black} Demographics" page ACTARM AGE SEX HEIGHT
WEIGHT RACE RFICDTC);
define page / order noprint;
define ACTARM / display style(header)={just=left vjust=top}
style(column)={cellwidth=7% just=left vjust=bottom};
define AGE / display style(header)={just=left vjust=top}
style(column)={cellwidth=3% just=left vjust=bottom};
define SEX / display style(header)={just=left vjust=top}
style(column)={cellwidth=6% just=left vjust=bottom};
define HEIGHT / display style(header)={just=left vjust=top}
style(column)={cellwidth=5% just=left vjust=bottom};
define WEIGHT / display style(header)={just=left vjust=top}
style(column)={cellwidth=5% just=left vjust=bottom};
define RACE / display style(header)={just=left vjust=top}
```

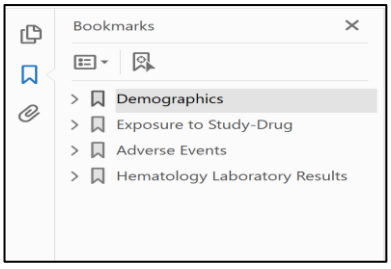
```
style(column)={cellwidth=5% just=left vjust=bottom};
define RFICDTC / display   style(header)={just=left vjust=top}
style(column)={cellwidth=7% just=left vjust=bottom}; break after page / page;
run;

        /*Creating the table for exposure data**/

    proc sql;
    create table work.ex as select a1.usubjid,
    ((A1.exstdtc)) as exstdtc label = "Start Date of Treatment",
    ((A1.exendtc)) as exendtc label = "End Date of Treatment",
    ((A1.exd1)) as exd1 label = "Duration of this period (days)",
    left((put(A1.exdose,8.))) AS exdose label = "Dose",
    left((A1.exdosu)) AS exdosu label = "Dose Units",
    left((A1.dosestat)) AS dosestat label = "Action on Drug",
    from work.ex A1 ;
    quit;

   /**Programming Left-Join between subject_visit date
   & Exposure data to extract cycle/Visit variable****/
      proc sql;
      create table EX_SV_JOIN as select A.*,B.VISIT
      FROM work.EX_F1 A LEFT JOIN WORK.SV B
      ON A.USUBJID = B.USUBJID AND A.EXSTDTC = B.SVSTDTC ;
      RUN;

/***Updating the LABEL for visit variable in Exposure dataset***/
             data ex_f;
             SET ex_sv_join ;
             label visit ="Cycle";run;
/**Sorting the Exposure data subject & dosage start date*/
                data ex_f;
                set ex_f;
                by usubjid exstdtc;
                if first.usubjid then count=0;
                count+1;
                page=ceil(count/&rownum);run;

/**Programing PROC REPORT for the Study Drug Exposure data**/

ODS PROCLABEL='Exposure to STUDY-DRUG-XXX';
proc report data=Work.EX_F center nowd headline headskip spacing=1 split="|"
missing spanrows style(header)={background=pink}
style(report)={outputwidth=100%};
where usubjid="&usubjid";
column ("^S={borderbottomcolor=black} Exposure to STUDY-DRUG-XXX" (page VISIT
EXSTDTC EXENDTC EXDOSE EXDOSU DOSESTAT);
define page    / order noprint;
define VISIT      / display style(header)={just=left vjust=top}
style(column)={cellwidth=7% just=left vjust=top};
define EXSTDTC / display style(header)={just=left vjust=top}
style(column)={cellwidth=7% just=left vjust=top};
define EXENDTC / display style(header)={just=left vjust=top}
style(column)={cellwidth=7% just=left vjust=top};
define EXDOSE     / display style(header)={just=left vjust=top}
style(column)={cellwidth=7% just=left vjust=top};
define EXDOSU     / display style(header)={just=left vjust=top}
style(column)={cellwidth=6% just=left vjust=top};
define DOSESTAT / display style(header)={just=left vjust=top}
```

```
style(column)={cellwidth=7% just=left vjust=top}; break after page / page;run;

/**Programing PROC REPORT for the adverse events data**/

ODS PROCLABEL='Adverse Events';
proc report data=Work.AE center nowd headline headskip spacing=1 split="|"
missing spanrows style(header)={background=pink}
style(report)={outputwidth=100%};
where usubjid="&usubjid";
column ("^S={borderbottomcolor=black} Adverse Events" (page AETERM AEDECOD
AESTDTC AEENDTC AETOXGR AEOUT AESER AEREL AEACN));
define page  / order noprint;
define AETERM     / id display   style(header)={just=left vjust=top}
style(column)={cellwidth=8% just=left vjust=top};
define AEDECOD / id display   style(header)={just=left vjust=top}
style(column)={cellwidth=8% just=left vjust=top};
define AESTDTC  / display   style(header)={just=left vjust=top}
style(column)={cellwidth=8%  just=left vjust=top};
define AEENDTC  / display   style(header)={just=left vjust=top}
style(column)={cellwidth=8%  just=left vjust=top};
define AETOXGR / display   style(header)={just=left vjust=top}
style(column)={cellwidth=8% just=left vjust=top};
define AEOUT      / display style(header)={just=left vjust=top}
style(column)={cellwidth=8% just=left vjust=top};
define AESER      / display style(header)={just=left vjust=top}
style(column)={cellwidth=8% just=left vjust=top};
define AEREL      / display style(header)={just=left vjust=top}
style(column)={cellwidth=8% just=left vjust=top};
define AEACN      / display   style(header)={just=left vjust=top}
style(column)={cellwidth=8% just=left vjust=top};break after page / page; run;

       /*Creating table for Laboratory: Hematology data**/

PROC SQL;
CREATE TABLE LB_HEM AS SELECT A1.USUBJID,
left((A1.VISIT)) AS VISIT LABEL = "Visit Name",
LEFT((A1.LBDTC)) AS LBDTC LABEL = "Date of Specimen Collection",
left((A1.Hemoglobin_LBORRES)) AS Hemoglobin_LBORRES LABEL = "Hemoglobin (g/dL)",
LEFT((A1.Leukocytes_LBORRES)) AS Leukocytes_LBORRES LABEL = "White Blood Cell
(K/uL)",
LEFT((A1.Neutrophils_LBORRES)) AS Neutrophils_LBORRES LABEL = "Absolute
Neutrophil count (K/uL)",
LEFT((A1.Platelets_LBORRES)) AS Platelets_LBORRES LABEL = "Platelet Count
(K/uL)",
LEFT((A1.Lymphocytes_LBORRES)) AS Lymphocytes_LBORRES LABEL = "Absolute
Lymphocyte Count (K/uL)",
LEFT((A1.Monocytes_LBORRES)) AS Monocytes_LBORRES LABEL = "Absolute Monocyte
Count(K/u   L)"
FROM WORK.LB A1 WHERE (Hemoglobin_LBORRES OR Leukocytes_LBORRES OR
Neutrophils_LBORRES OR
Platelets_LBORRES OR Lymphocytes_LBORRES OR Monocytes_LBORRES IS NOT NULL);QUIT;
/**Renaming the Results variable***/
Data LB_HEM_RENAM(rename=(Hemoglobin_LBORRES = HEMORRES
Leukocytes_LBORRES=LEUORRES Neutrophils_LBORRES = NEUORRES
Platelets_LBORRES = PLATORRES Lymphocytes_LBORRES = LYMPORRES Monocytes_LBORRES
= MONOORRES)) ;
SET LB_HEM ;
RUN;
```

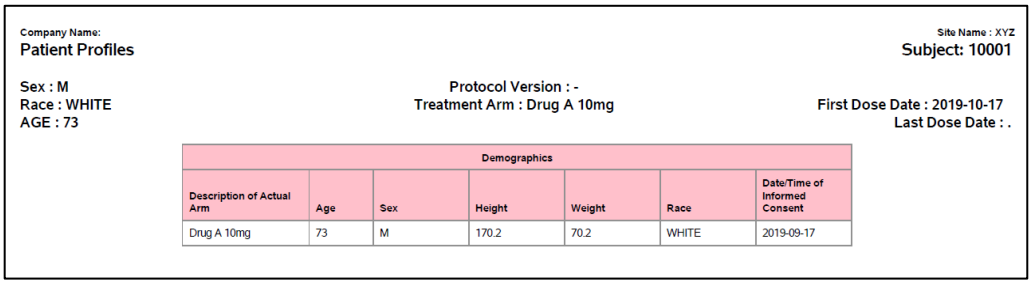
```
     /**Sorting the hematology data by subject & test start date*/

            proc sort data=LB_HEM_F; by  usubjid lbdtc; run;
            data LB_HEM_RENAM;
            set LB_HEM_RENAM;
            by usubjid lbdtc ;
            if first.usubjid then count=0;
            count+1;
            page=ceil(count/&rownum);
            run;

/**Programing PROC REPORT for the laboratory Hematology data**/

ODS PROCLABEL='Hematology';
proc report data=work. LB_HEM_F center nowd headline headskip spacing=1
split="|" missing spanrows style(header)={background=pink}
style(report)={outputwidth=100%};
where usubjid="&usubjid";
column (' ^R/RTF"\i\b\fs18\cf2 Hematology' (page VISIT LBDTC HEMORRES LEUORRES
NEUORRES PLATORRES LYMPORRES MONOORRES));
define page / order noprint;
define VISIT / display style(header)={just=left vjust=top}
style(column)={cellwidth=8% just=left vjust=top};
define LBDTC / display style(header)={just=left vjust=top}
style(column)={cellwidth=7% just=left vjust=bottom};
define HEMORRES / display style(header)={just=left vjust=top}
style(column)={cellwidth=5% just=left vjust=bottom};
define LEUORRES /  display   style(header)={just=left vjust=top}
style(column)={cellwidth=7% just=left vjust=bottom};
define NEUORRES /   display   style(header)={just=left vjust=top}
style(column)={cellwidth=9% just=left vjust=bottom};
define PLATORRES / display   style(header)={just=left vjust=top}
style(column)={cellwidth=9% just=left vjust=bottom};
define LYMPORRES / display   style(header)={just=left vjust=top}
style(column)={cellwidth=8% just=left vjust=bottom};
define MONOORRES /  display   style(header)={just=left vjust=top}
style(column)={cellwidth=10% just=left vjust=bottom};break after page/ page;run;
ods pdf close;
%let iter=%eval(&iter.+1);
%end;
%mend;
/**Call the macro**/
%loopOverDatasets;
```

RESULTS

The resulting patient profile output is shown below in Figure 1: Demographics Profile Data ;
Figure 2: Exposure to Study Drug ; Figure 3: Adverse Events & Figure 4: Laboratory Hematology Data.



PDF Output with Bookmarks



Company Name:
Patient Profiles

Site Name : XYZ
Subject: 10001

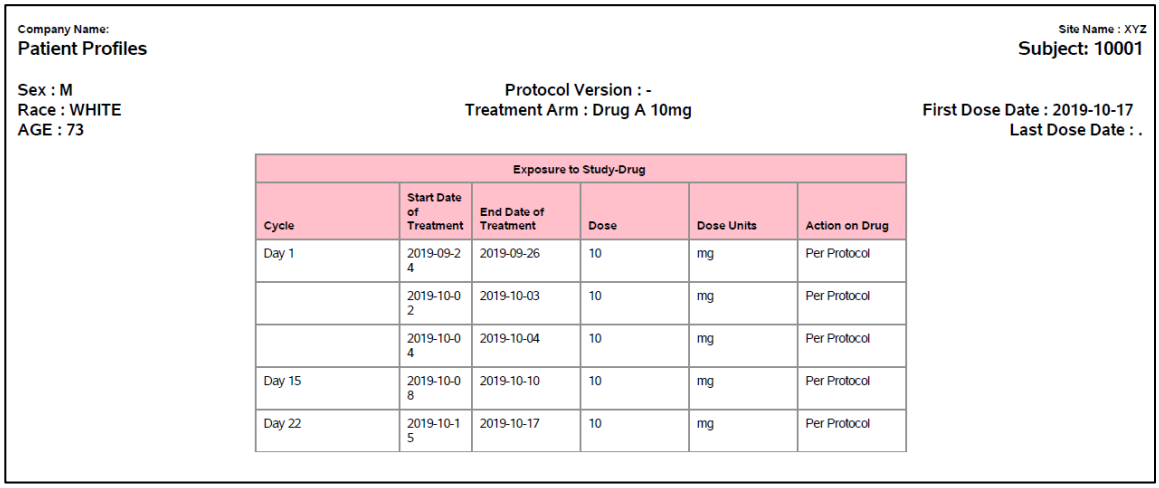
Sex : M
Race : WHITE
AGE : 73

Protocol Version : -
Treatment Arm : Drug A 10mg

First Dose Date : 2019-10-17
Last Dose Date : .

Demographics						
Description of Actual Arm	Age	Sex	Height	Weight	Race	Date/Time of Informed Consent
Drug A 10mg	73	M	170.2	70.2	WHITE	2019-09-17

Figure 1: Demographics Data



Company Name:
Patient Profiles

Site Name : XYZ
Subject: 10001

Sex : M
Race : WHITE
AGE : 73

Protocol Version : -
Treatment Arm : Drug A 10mg

First Dose Date : 2019-10-17
Last Dose Date : .

Exposure to Study-Drug					
Cycle	Start Date of Treatment	End Date of Treatment	Dose	Dose Units	Action on Drug
Day 1	2019-09-24	2019-09-26	10	mg	Per Protocol
	2019-10-02	2019-10-03	10	mg	Per Protocol
	2019-10-04	2019-10-04	10	mg	Per Protocol
Day 15	2019-10-08	2019-10-10	10	mg	Per Protocol
Day 22	2019-10-15	2019-10-17	10	mg	Per Protocol

Figure 2: Exposure to Study Drug

Company Name:
Patient Profiles

Site Name : XYZ
Subject: 10001

Sex : M
Race : WHITE
AGE : 73

Protocol Version : -
Treatment Arm : Drug A 10mg

First Dose Date : 2019-10-17
Last Dose Date : .

Adverse Events

Reported Term for the Adverse Event	Dictionary -Derived Term	Start Date of Adverse Event	End Date of Adverse Event	Standard Toxicity Grade	Outcome of Adverse Event	Serious Event	Causality	Action Taken with Study Treatment
DIARRHEA	Diarrhea	2019-09-27	2019-10-01	2	RECOVERED/ RESOLVED	N	NOT RELATED	DOSE NOT CHANGED
DIARRHEA	Diarrhea	2019-10-01	2019-10-15	1	RECOVERED/ RESOLVED WITH SEQUELAE	N	NOT RELATED	DOSE NOT CHANGED
DIARRHEA	Diarrhea	2019-10-15	2019-10-22	2	RECOVERED/ RESOLVED WITH SEQUELAE	N	NOT RELATED	DOSE NOT CHANGED
DIARRHEA	Diarrhea	2019-10-22	2019-11-12	2	NOT RECOVERED/ NOT RESOLVED	N	NOT RELATED	NOT APPLICABLE
DIARRHEA	Diarrhea	2019-11-12	2019-11-20	4	RECOVERED/ RESOLVED	Y	NOT RELATED	NOT APPLICABLE

Figure 3: Adverse Events

Company Name:
Patient Profiles

Site Name : XYZ
Subject: 10001

Sex : M
Race : WHITE
AGE : 73

Protocol Version : -
Treatment Arm : Drug A 10mg

First Dose Date : 2019-10-17
Last Dose Date : .

Hematology

Visit	Date of Specimen Collection	Hemoglobin (g/dL)	White Blood Cell (K/uL)	Absolute Neutrophil count (K/uL)	Platelet Count (K/uL)	Absolute Lymphocyte Count (K/uL)	Absolute Monocyte Count (K/u L)
Screening	2019-09-17	10.3	7.6	7.3	205	0.2	0.0
Day 1	2019-09-24	10.6	8.0	6.8	181	0.4	0.8
Unscheduled	2019-09-27	11.8	5.4	4.3	144	0.5	0.6
Day 8	2019-10-02	10.5	6.3	5.7	166	0.2	0.3
Day 15	2019-10-08	11.8	7.5	6.9	186	0.2	0.3
Day 22	2019-10-15	11.1	7.2	6.3	156	0.3	0.5
End of Treatment	2019-10-22	10.4	6.0	5.2	145	0.2	0.5
Follow-up	2019-11-14	9.7	5.4	4.3	57	0.0	0.6

Figure 4: Laboratory Hematology Data

CONCLUSION

Patient profiles created with SAS® ODS and PROC REPORT procedure provides a simplistic method of how the SAS® software may be used to produce one of the complicated clinical study data requested in clinical trials.

There are benefits of programming SAS Patient profile in macros, so that it can be extended, reproduced & enhanced in multiple studies to meet the requirements. Also since clinical trial have a lot of table, listing & findings data, all of them combined together for review by programmers & statisticians is tedious, therefore programming with SAS® ODS and creating PDF patient profile outputs with bookmarks provides a quick method for reviewers to find what they want to navigate.

REFERENCES

1. O' Connor, Daniel, Matange, Sanjay 2010." Create Comprehensive Patient Profiles with SAS". Cary, NC. PharmaSug 2010, Paper CC-SAS01.
2. Clinical Data Interchange Standards Consortium. 2020. Study Data Tabulation Model, version 3.1.2 http://www.cdisc.org/

3. Light, S., Gilbert, P. and Genereux, G., (2001). A Novel Approach to Developing a Patient Profile Reporting Application SUGI 26, Paper P042-26.
4. Shu, H. and Zhuge, Y. (2005). Use ODS Generating Patient Profiles. PharmaSUG 2005, Paper TT09.
5. Ritter, A. (2011). Creating Customized Patient Profiles using SAS ODS RTF and PROC TEMPLATE PharmaSUG, 2011, Paper TT03.
6. Okerson, Barbara. “Pleasing the Client: Creating Custom Reports with SAS® ODS LAYOUT and Proc REPORT,” SESUG Proceedings, 2009.
7. Janet Stuelpner. J.J. Hantsch. 2012. One at a Time; Producing Patient Profiles and Narratives. SAS Global Forum.

ACKNOWLEDGMENTS

I would like to thank my manager, Sneha Sundet, for her constant encouragement and critical Review. I would also like to thank Robert O’ Connor for providing me the opportunity to program and work on the Patient Profiles and for his suggestions .

CONTACT INFORMATION

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

Name: Ballari Sen

Company : Agios Pharmaceuticals,INC.

Email : ballari.sen@agios.com

Phone: 7165330372