SAS162-2022

Quick Jumpstart into Pharmaceutical SAS Programming

Matt Becker, SAS Institute Inc.

Jim Box, SAS Institute Inc.

ABSTRACT

SAS analytics is used across many industries: retail, gaming, finance, insurance, and many others. But what is different when it comes to the pharmaceutical industry? What kind of SAS PROCs do we use for safety and efficacy in clinical development? What do safety and efficacy in clinical development mean? In this paper and presentation, we will cover common topics and SAS programming methodologies used in clinical R&D analysis.

INTRODUCTION

The pharmaceutical industry is the discovery, development, and manufacturing of medications by public and private organizations. The creation of these medications spans centuries if not millenniums. But what are some of the key aspects of the pharmaceutical industry that are important to SAS users and how is SAS used in clinical trials? In this paper and presentation, we will spend time on the key aspects of the industry and how analytics is prominent in medical research and development.

Clinical trials are studies the test whether a medical method (strategy, treatment, device) is safe and effective for humans. As the gold standard, they provide the best data for making decisions on safety and effectiveness. In 2019, it was estimated to get a new prescription medicine to market cost nearly $2B US dollars and only ~12% succeed in getting to market. The time it takes to get from ideation to market of a prescription medicine takes ~12 years to regulatory approval.

Within clinical research, there are four (4) phases of clinical trials: Phase I, Phase II, Phase III and Phase IV. In Phase I, the clinical trials are meant to identify what dosage of a medicine is safe. Ten (10) to twenty (20) subjects are enrolled in these studies. Phase II clinical trials are meant to show if the medicine works – does it have efficacy. These studies have twenty (20) to two hundred (200) subjects enrolled. In Phase III, clinical trials are compared to placebo (sugar pill for example) or a comparator drug. These clinical trials have greater than one thousand (1,000) subjects enrolled. Both safety and efficacy are analyzed in these studies. Finally, Phase IV clinical trials look at how the medicine is doing in the ‘real world.’ These trials are not always required and may be competitive.

Double-blind controlled trials are the gold standard for medicinal products. These clinical trials require data access, data management, data integration, analysis, and reporting. To many, SAS is the industry standard for data and analyses. It is a flexible platform that provides the ability to perform ETL (extract transform load) to industry standards such as CDISC SDTM and ADaM, to generate statistical analyses, and to produce tables, listings, and figures for clinical trial submissions. In this paper, we will focus on the execution of clinical trials: namely standards, safety, efficacy, and how SAS is commonly used for these tasks.

CDISC AND DATA STANDARDS

The clinical data interchange standards consortium (CDISC) creates standards for medical research. Standards are instrumental in helping with efficiency in the system interoperability and review of submissions of medicinal products to regulatory agencies. The first release of data standards came about in 1999 with SDS v1.0 (submission data standards) and ODM v0.8 (operational data model).

Within clinical trial programming, CDISC SDTM (standard data tabulation model) and ADaM (analysis data set model) are the two metadata standards we aspire to utilize. Each standard will have “domains” which are data structures for key aspects of clinical trials. Examples of some domains are demographics (DM), adverse events (AE), and laboratory (LB). An example of the SDTM model for a sub-section of the AE domain is shown in Figure 1. Figure 2 represents a portion of the subject level metadata (ADSL) for an ADaM domain.

C	D	E	F	G	H	I	J	K	L	M
Class	Dataset Name	Variable Name	Variable Label	Type	CDISC CT Codelist Code(s	Described Value Domain(s)	Value List	Role	CDISC Notes	Core
Events	AE	STUDYID	Study Identifier	Char				Identifier	Unique identifier for a study.	Req
Events	AE	DOMAIN	Domain Abbreviation	Char			AE	Identifier	Two-character abbreviation for the domain.	Req
Events	AE	USUBJID	Unique Subject Identifier	Char				Identifier	Identifier used to uniquely identify a subject across all studies for all applications or submissions involving the product.	Req
Events	AE	AESEQ	Sequence Number	Num				Identifier	Sequence number given to ensure uniqueness of subject records within a domain. May be any valid number.	Req
Events	AE	AETERM	Reported Term for the Adverse Event	Char				Topic	Verbatim name of the event.	Req
Events	AE	AEDECOD	Dictionary-Derived Term	Char		MedDRA		Synonym Qualifier	Dictionary-derived text description of AETERM or AEMODIFY. Equivalent to the Preferred Term (PT in MedDRA). The sponsor is expected to provide the dictionary name and version used to map the terms utilizing the external codelist element in the Define-XML	Req
Events	AE	AECAT	Category for Adverse Event	Char				Grouping Qualifier	Used to define a category of related records. Examples: "BLEEDING", "NEUROPSYCHIATRIC".	Perm
Events	AE	AEBODSYS	Body System or Organ Class	Char				Record Qualifier	Dictionary derived. Body system or organ class used by the sponsor from the coding dictionary (e.g., MedDRA). When using a multi-axial dictionary such as MedDRA, this should contain the SOC used for the sponsor's analyses and summary tables, which may not necessarily	Exp
Events	AE	AESEV	Severity/Intensity	Char	C66769			Record Qualifier	The severity or intensity of the event. Examples: "MILD", "MODERATE", "SEVERE".	Perm
Events	AE	AESER	Serious Event	Char	C66742			Record Qualifier	Is this a serious event? Valid values are "Y" and "N".	Exp
									Describes changes to the study treatment as a result of the event. AEACN is specifically for the relationship to study treatment. AEACNOTH is for actions unrelated to dose adjustments of study treatment. Examples of	

Figure 1. Adverse Event (AE) SDTM domain

B	C	D	E	F	G	H	I	J
Data Structure Name	Dataset Name	Variable Group	Variable Name	Variable Label	Type	Codelist/Controlled Terms	Core	CDISC Notes
Subject-Level Analysis Dataset	ADSL	Identifier	STUDYID	Study Identifier	Char		Req	DM.STUDYID
Subject-Level Analysis Dataset	ADSL	Identifier	USUBJID	Unique Subject Identifier	Char		Req	DM.USUBJID
Subject-Level Analysis Dataset	ADSL	Subject Demographics	AGE	Age	Num		Req	DM.AGE. If analysis needs require a derived age that does not match DM.AGE, then AAGE must be added
Subject-Level Analysis Dataset	ADSL	Subject Demographics	AGEU	Age Units	Char		Req	DM.AGEU
Subject-Level Analysis Dataset	ADSL	Subject Demographics	AGEGRy	Pooled Age Group y	Char		Perm	Character description of a grouping or pooling of the subject's age for analysis purposes. For example, AGEGR1 might have values of "<18", "18-65", and ">65"; AGEGR2 might have values of "Less than 35 y old" and "At least 35 y old".
Subject-Level Analysis Dataset	ADSL	Subject Demographics	SEX	Sex	Char		Req	DM.SEX.
Subject-Level Analysis Dataset	ADSL	Subject Demographics	RACE	Race	Char		Req	DM.RACE.
Subject-Level Analysis Dataset	ADSL	Population Indicator	SAFFL	Safety Population Flag	Char		Cond	These flags identify whether or not the subject is included in the specified population. A minimum of one subject-level population flag variable is required in ADSL. \n Not all of the indicators listed here need to be included in ADSL. As stated in Section 3.1.4, Flag Variable Conventions, only those indicators corresponding to populations defined in the statistical analysis plan or populations used as a basis for analysis need be included in ADSL. \n This list of flags is not meant to be all-inclusive. Additional population flags may be added. \n The values of subject-level population flags cannot be blank. If a flag is used, the corresponding numeric version (*FN, where 0 = No and 1 = Yes) of the population flag can also be included. Please also refer to Section 3.1.4, Flag Variable Conventions.
Subject-Level Analysis Dataset	ADSL	Treatment	ACTARM	Description of Actual Arm	Char		Perm	DM.ACTARM
Subject-Level Analysis Dataset	ADSL	Treatment Timing	TRTSDT	Date of First Exposure to Treatment	Num		Cond	Date of first exposure to treatment for a subject in a study. TRTSDT and/or TRTSDTM are required if there is an investigational product. Note that TRTSDT is not required to have the

Figure 2. Subject level analysis dataset (ADSL) ADaM domain

For the Pharmaceutical industry, the first programming tasks revolve around taking an input data structure to an SDTM domain. This is called “mapping.” Input data structures could follow another CDISC standard named CDASH (clinical data acquisitions standards harmonization) or a company metadata template. In most cases today, electronic data capture systems (EDC) provide this raw data.

SAFETY

Subject safety in clinical trials looks to ensure unnecessary harm from a medicinal product is emphasized. During the trial, subjects are routinely monitored via investigator visits, diary entries, phone calls, etc. Within our CDISC standard domains, safety data is recorded in demographics (DM, ADSL), labs (LB, ADLB), medications (CM, ADCM), adverse events (AE, ADAE), vital signs (VS, ADVS), and others.

EFFICACY

In clinical trials, efficacy is how well a treatment (diagnostic, medicinal, preventative) achieves the desired outcome. As with safety, subjects are routinely monitored during the trial. Efficacy data is recorded in our CDISC standard domains subject level analysis dataset (ADSL), labs (LB), efficacy analysis dataset (ADEFF), therapeutic area (TA), and others.

MAPPING TO DATA STANDARDS WITH SAS

When mapping to CDISC SDTM or ADaM, we usually start with the standard Excel file and begin documenting our RAW input data variables and which standard variable into which they should be mapped. For example, we receive data from our EDC system with a variable that identifies when a subject started treatment. Upon reviewing the SDTM structure for DM, we realize there is a variable RFSTDTC that identifies when a subject starts treatment (see Figure 1). Thus, we map my EDC variable into RFSTDTC.

Similarly, when creating an ADaM variable, we start with the standards Excel file and begin documenting how each metadata variable should be created. For example, in our ADSL domain, there is a variable named “SAFFL” (see Figure 2). This variable denotes whether that patient is eligible for “safety analysis.” These definitions, what details whether a patient is eligible for safety analysis, are defined within the statistical analysis plan (SAP). Our SAS program incorporates the SAP rules into code to produce the SAFFL variable for each subject.

Mapping code in SAS commonly utilizes data step or Proc SQL, SAS macros for re-usable code, and SAS procedures. Within the data step or Proc SQL, the tasks that are done include variable creation, variable assignment, data integration, and data creation. SAS macros are highly valuable and are regularly used within mapping programs to standard domains. For example, there are many dates that are received from our data capture systems. These dates need to be converted to standard formats – so a macro to do this conversion is regularly used. Lastly, Proc SORT, Proc SQL, Proc Transpose, Proc Freq, Proc Means, and others are used to perform key data transformation or analysis. Figure 3 shows an example mapping program for an adverse event (AE) domain.

```
data ae(drop=visit aestdtc aeendtc aeacn aerel aeout
        rename=(visitnew=visit aestdtc_new=aestdtc aeendtc_new=aeendtc aeacn_new=aeacn aerel_new=aerel aeout_new=aeout));
  set &rawdata..ae;
  length usubjid $25 visitnew $25 aesev aecat $50 aeenrf $15;
  usubjid=trim(left(studyid))||'-'||trim(left(site))||'-'||trim(left(randomno));
  domain='AE';
  if aenone='Y' then aeoccur='N';
  else aeoccur='Y';
  if aeoccur='Y' then do;
    %msdtmdt(datepart=aestdt, datevar=aestdtc_new);
    %msdtmdt(datepart=aeendt, datevar=aeendtc_new);
  end;
  aecat='ADVERSE EVENTS';
  aeseq='14';
  visitnum=visit;
  visitnew=upcase(put(visitnum,visitf.));
  if aeongo='Y' then aeenrf='ONGOING';
  aesev=aetoxgrc;
  aeacn_new=aeacnc;
  aerel_new=aerelc;
  aeout_new=aeoutc;
  aecontrt=aecontr;
run;

%msdtmdy(inds=ae, todate=aestdtc, studyday=aestdy);
%msdtmdy(inds=ae, todate=aeendtc, studyday=aeendy);

proc sort data=ae;
  by aeterm;

proc sort data=raw.codeae out=codeae;
  by aeterm;

data ae;
  merge ae(in=x) codeae(in=y);
  by aeterm;
  length aebodsys aedecod $200;
  if x;
  aebodsys=upcase(socterm);
  aedecod=upcase(prefterm);
run;

proc sort data=ae;
  by studyid domain usubjid aeseq aespid;

%mimpsdtm(micsv=AE, miin=AE, miout=sdtm.ae, mlbl=AE SDTM Dataset);
```

Figure 3. SDTM AE domain mapping program

SAS PROGRAMMING TECHNIQUES FOR LISTINGS

Listings are simple reports of data for a subject by topic. For example, the listing of vital signs provides a list of each subject's vital sign results (e.g. temperature, blood pressure, pulse) by visit. The statistical analysis plan (SAP) will contain "shells" which are visual representations of what each listing will contain. It is then the responsibility of the programmer to create the listing for that topic.

Listing code in SAS commonly utilizes data step or Proc SQL, SAS macros for re-usable code, SAS procedures, and ODS statements. Within the data step or Proc SQL, the tasks that are done include variable creation, variable assignment, and data integration. SAS macros are highly valuable and are used in listings. For example, we may be creating a listing of adverse events, a listing of serious adverse events, and a listing of treatment emergent adverse events. This is the same structure of output. Thus, a macro could be written that has as a parameter the filter for each adverse event listing and could be called to create all 3 listings. To create output files that we can provide for submission, we use ODS statements to identify an output destination. Common destinations are RTF, PDF, and SAS LST files. Lastly, Proc SORT, Proc SQL, Proc Transpose, and Proc Report are the common SAS procedures to create listing output. Figure 4 shows an example medical history listing program.

```
proc sort data=sdtm.mh out=mh;
  by usubjid;

proc sort data=sdtm.dm out=dm(keep=usubjid age);
  by usubjid;

data mh;
  merge mh(in=x) dm(in=y);
  by usubjid;
  if x;
run;

proc sort data=mh;
  by usubjid mhspid;

data mh;
  set mh;
  by usubjid mhspid;
  page=int(_n_/20)+1;
  if mhoccur='N' and mhterm='' then mhterm='NONE';
run;

%mcase(inds=mh, exceptl=%str('USUBJID','MHSTDTC','MHENDTC'));

%mtitle(progid=lmh);

proc report data=mh headline headskip nowindows split='|' missing spacing=1;
  column page usubjid age mhspid mhstdtc mhendtc mhterm;
  define page / order noprint;
  define usubjid / order 'Subject' style={just=left cellwidth=6%};
  define age / order 'Age|(Years)' format=4.1 style={just=center cellwidth=7%};
  define mhspid / order noprint;
  define mhstdtc / display 'Start Date' style={just=left cellwidth=10%};
  define mhendtc / display 'Stop Date' style={just=left cellwidth=10%};
  define mhterm / display 'Description of Condition / Event' flow style={just=left cellwidth=34%};
  break after page / page;
  compute before usubjid;
    line " ";
  endcomp;
run;

ods rtf close;
ods listing;
```

Figure 4. Medical History Listing Program

SAS PROGRAMMING TECHNIQUES FOR TABLES

Tables are aggregate summaries of data across subjects by topic. For example, the table of vital signs provides subject aggregate summaries of vital sign results (e.g. n, mean, median, min, max of diastolic blood pressure). The statistical analysis plan (SAP) will contain “shells” which are visual representations of what each table will contain. It is then the responsibility of the programmer to create the table for that topic.

Table code in SAS commonly utilizes data step or Proc SQL, SAS macros for re-usable code, SAS procedures, and ODS statements. Within the data step or Proc SQL, the tasks that are done include variable creation, variable assignment, and data integration. SAS macros are highly valuable and are used in tables. For example, we need to count the number of males and females in our study. In addition, we need to count the number of specific adverse events (e.g. “Fever”) in our study. Both counts are computing the frequency of categorical variables. If we are doing this for many variables across our tables, we can create a macro to provide these results. To create output files that we can provide for submission, we use ODS statements to identify an output destination. Common destinations are RTF, PDF, and SAS LST files. Lastly, Proc SORT, Proc SQL, Proc Transpose, Proc Report, Proc Means, Proc Freq, Proc Summary, Proc TTest, and Proc Lifetest are the common SAS procedures to create table output. Figure 5 shows an example study drug exposure table program.

```
data ex;
  set &derdata..ex(where=(q_safeas='Y' and exseq in(0,1)));
  &tcond;
  output;
  if atrt in(2,3,4) then do;
    atrt=5;
       output;
  end;
  atrt=6;
  output;
run;

%mtottrt(cond=%str(if q_safeas='Y';) &tcond);

%mss(msdata=ex, msout=vol1, msvar=q_voluse, msstats=n meansd median range, msprec=0, msorder=1);

%mss(msdata=ex, msout=vol2, msvar=q_vtbsa, msstats=n meansd median range, msprec=0, msorder=2);

%mss(msdata=ex, msout=vol3, msvar=q_vtbsat, msstats=n meansd median range, msprec=0, msorder=3);

data final;
  set vol1 vol2 vol3;
  length page 4;
  page=1;
run;

proc sort data=final;
  by page order sorder;
run;

data final;
  set final;
  by page order sorder;
  length firstcol $40;
  if first.order then firstcol=put(order,orderf.);
  else firstcol='';
run;

%mtitle(progid=&progid);

proc report data=final headline headskip nowindows split='|' missing spacing=1 style(header)=[protectspecialchars=off];
  column  page order sorder firstcol text ("Age Group (years) \brdrb\brdrs" trt1 trt2 trt3 trt4 trt5) trt6;
  define page /order noprint;
  define order /order noprint;
  define sorder /order noprint;
  define firstcol /" " style={just=l cellwidth=25%};
  define text/ " " style={just=l cellwidth=10%};
  define trt1 / "     0 - 2 |     (N=&pop1)" style={cellwidth=9% asis=on pretext="\tqdec\tx450 "};
  define trt2 / "     3 - 6 |     (N=&pop2)" style={cellwidth=9% asis=on pretext="\tqdec\tx450 "};
  define trt3 / "     7 - 11 |     (N=&pop3)" style={cellwidth=9% asis=on pretext="\tqdec\tx450 "};
  define trt4 / "     12 - &up_limit |     (N=&pop4)" style={cellwidth=9% asis=on pretext="\tqdec\tx450 "};
  define trt5 / "     3 - &up_limit Total |     (N=&pop5)" style={cellwidth=9% asis=on pretext="\tqdec\tx450 "};
  define trt6 / "     Total |     (N=&pop6)" style={cellwidth=9% asis=on pretext="\tqdec\tx450 "};
  break after page / page;
  compute before order;
    line " ";
  endcomp;
run;
```

Figure 5. Study Drug Exposure Table Program

CONCLUSION

Double-blind controlled trials are the gold standard for clinical trial studies. In most cases, the data for these studies are provided by electronic data capture systems. SAS is considered the pharmaceutical industry standard for submission of data and analyses for medicinal product approval by regulatory agencies. Using SAS, programs can be written to perform data transformation to industry standards such as CDISC SDTM and ADaM, to generate statistical analyses, and to produce tables, listings, and figures for clinical trial submissions.

CONTACT INFORMATION

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

Matt Becker / Jim Box
SAS

matt.becker@sas.com / jim.box@sas.com