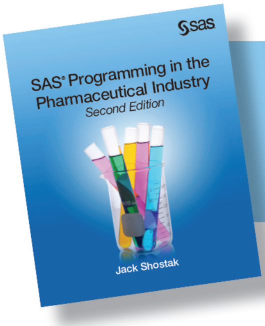
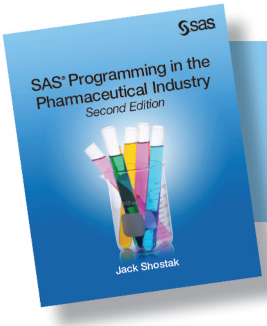
SAS® Programming in the Pharmaceutical Industry

Second Edition

Jack Shostak


SAS
SAS® Programming in the Pharmaceutical Industry
Second Edition
Jack Shostak

Contents



Chapter 2 Preparing and Classifying Clinical Trial Data

This chapter describes the key clinical data preparation issues and the different classes of clinical data that are found in clinical trials. Each class of data brings with it a different set of challenges and special handling issues. Sample case report form (CRF) pages are provided. These pages are loosely based on the Clinical Data Interchange Standards Consortium’s (CDISC) Clinical Data Acquisition Standards Harmonization (CDASH) data collection standard. They are provided with each type of data to aid you in visualizing what the data in the CDISC Study Data Tabulation Model (SDTM) standard would look like. The key data preparation issues presented are concepts that apply universally across the various classes of clinical trial data.

Preparing Clinical Trial Data

Clinical trial data come to the statistical programmer in two basic forms: numeric variables and character string (text) variables. With this in mind, there are two considerations for all numeric and text variables. All data should be cleaned if they are needed for analyses, and any data entered as *free-text variables* should be coded or categorized if they are needed for analyses. Generally speaking, it is much more preferable if the data is coded either inherently by data collection design or later by clinical data management before it ever is sent to a statistical programmer.

“Clean” the Data If They Are Needed for Analysis

If data will be summarized or analyzed as part of the protocol-defined statistical analysis, they should be cleaned first. “Cleaned” in this context means that erroneous data that have been entered into a variable are repaired before data analysis. Under the direction of the statistics group and based on the needs of the statistical analysis plan, the data management group is responsible for cleaning the clinical data.

Before the statistical programmer receives data that are ready for analysis, the clinical data management group cleans the data. This is done through a query process, which is built into the clinical data management system. The clinical data management query process usually looks like this:

1. A programmatic or manual investigation of the data finds an errant data point.
2. A “query” or data clarification form (DCF) for that data point is sent to the clinical site.
3. The clinical site responds to the query. If the data is collected via an electronic data capture system, the site may fix the data issue.
4. If the clinical site does not fix the data issue themselves, then the clinical data management group updates the database or CRF based on the response from the clinical site.

Depending on the size and complexity of the clinical trial, queries sent to sites can easily number in the thousands. Because the cost of reconciling these queries quickly rises, it is important to be judicious when creating them. It is worth noting that electronic data capture (EDC) systems may reduce the number of queries needed, because the entry screens are often programmed so that errant data cannot be entered. It is also worth noting that if the clinical data is placed into the CDISC SDTM format, there can be a large number of automatic data queries generated because standard queries and cross data type queries are easy to generate from the SDTM data model.

In order to reduce unnecessary data queries, the statistics group should be consulted early in the clinical database development process to identify variables that are critical for data analysis. Optimally, the statistical analysis plan would already be written by the time of database development so that the queries could be designed based on the critical variables indicated in the analysis plan. However, at the database development stage, usually only the clinical protocol exists to guide the statistics and clinical data management departments in developing the query or data management plan.

How clean the data must be depends on the importance of the data. Critical analysis variables must be clean, so this is where the site and data management groups should focus their resources. If the data are “dirty” at the time of statistical analysis, many inefficient and costly workarounds may need to be applied in the statistical programming, and the quality of the data analysis could suffer. However, if a variable is not important to the statistical analysis, then it is better to save the expense of cleaning that variable.

Categorize Data If Necessary

Clinical trial data come in two basic forms: numeric variables and text variables. Numeric variables are easy for the statistical programmer to handle. Numbers can be analyzed with SAS in a continuous or categorical fashion without much effort. If a numeric variable needs categorization, it is easy enough to categorize the data within SAS. For example, if you had to classify patient age, a simple DATA step such as the following might serve well.

Program 2.1 Categorizing Numeric Data

```
data adsl;
   set adsl;

      if . < age <= 18 then
         agegr1n = 1;
      else if 18 < age <= 60 then
         agegr1n = 2;
      else if 60 < age then
         agegr1n = 3;
run;
```

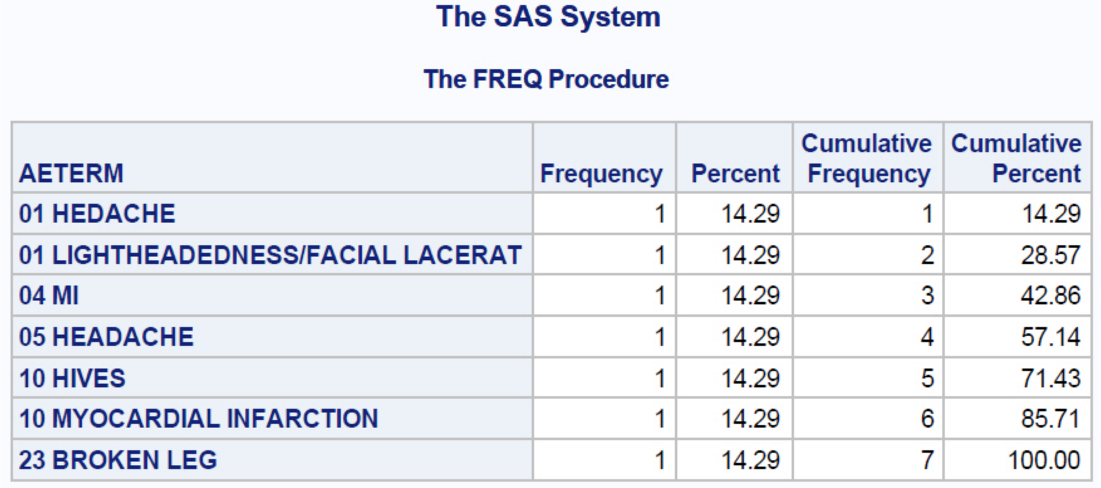
The problem for the statistical programmer in categorizing data comes from text variables or, more specifically, free-text variables. A “free-text” variable is one that may contain any characters and is typically limited only in length. As an example, let’s say you need to summarize the adverse events for a set of patients in a trial. The following SAS code shows the data and a quick summarization of the adverse events.

Program 2.2 Summarizing Free-Text Adverse Event Data

```
data AE;
input USUBJID $ 1-7 AETERM $ 9-41;
datalines;
100-101 HEDACHE
100-105 HEADACHE
100-110 MYOCARDIAL INFARCTION
200-004 MI
300-023 BROKEN LEG
400-010 HIVES
500-001 LIGHTHEADEDNESS/FACIAL LACERATION
;
run;
```

```
options nodate nonumber missing = ' ';
ods escapechar='#';
ods pdf style=htmlblue file='program2.2.pdf';

proc freq
   data = ae;
   tables aeterm;
run;

ods pdf close;
```

Program 2.2 yields the following output.



The SAS System

The FREQ Procedure

AETERM	Frequency	Percent	Cumulative Frequency	Cumulative Percent
01 HEDACHE	1	14.29	1	14.29
01 LIGHTHEADEDNESS/FACIAL LACERAT	1	14.29	2	28.57
04 MI	1	14.29	3	42.86
05 HEADACHE	1	14.29	4	57.14
10 HIVES	1	14.29	5	71.43
10 MYOCARDIAL INFARCTION	1	14.29	6	85.71
23 BROKEN LEG	1	14.29	7	100.00

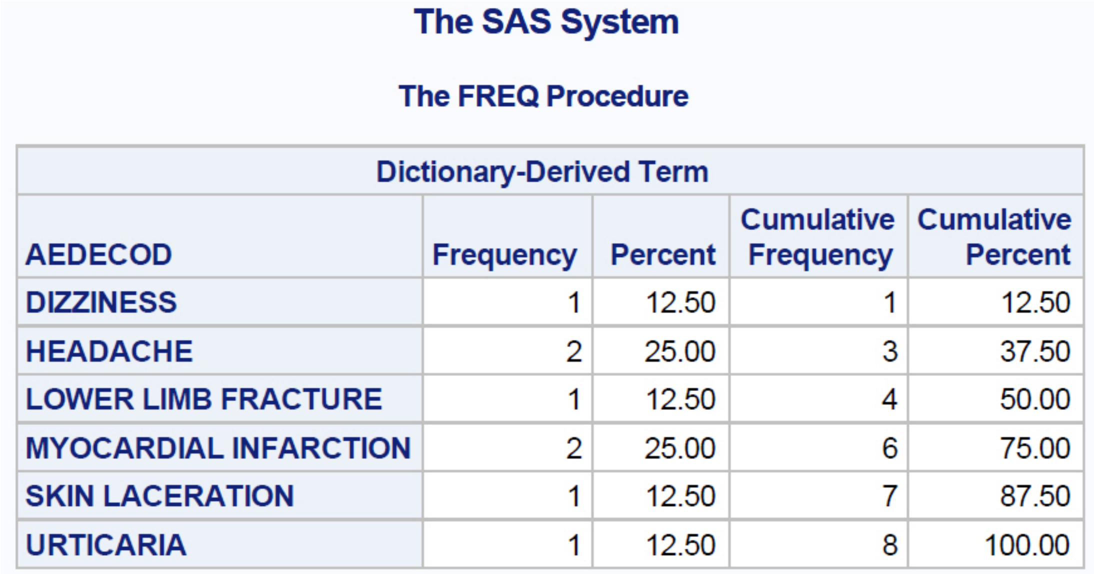
There are three problems with this adverse events summary. First, “HEADACHE” and “HEDACHE” are counted as separate events even though it is clear that the latter is simply a misspelling of the former. Second, “MI” and “MYOCARDIAL INFARCTION” are considered as separate events even though the former is simply an abbreviation of the latter. Finally, “LIGHTHEADEDNESS/FACIAL LACERATION” refers to perhaps related but different adverse events that need to be counted separately. All three of these problems exist because the data were entered in free-text fashion and summarized from the free-text variable AETERM.

There is only one good solution to handling free-text variables that are needed for statistical analysis. The free-text variables need to be coded by clinical data management in the clinical database. If the adverse events were coded with a dictionary, such as *MedDRA*, which will be explored further in Chapter 4, the previous example might look like Program 2.3.

Program 2.3 Summarizing Coded Adverse Event Data

```
data ae;
label USUBJID  = "Unique Subject Identifier"
      AEPTCD   = "Preferred Term Code"
      AETERM   = "Reported Term for the Adverse Event"
      AEDECOD  = "Dictionary-Derived Term";

input USUBJID $ 1-7 AEPTCD $ 9-16
      AETERM $ 18-38 AEDECOD $ 40-60;

datalines;
100-101 10019211 HEDACHE               HEADACHE
100-105 10019211 HEADACHE              HEADACHE
100-110 10028596 MYOCARDIAL INFARCTION MYOCARDIAL INFARCTION
200-004 10028596 MI                    MYOCARDIAL INFARCTION
300-023 10061599 BROKEN LEG            LOWER LIMB FRACTURE
400-010 10046735 HIVES                 URTICARIA
500-001 10013573 LIGHTHEADEDNESS       DIZZINESS
500-001 10058818 FACIAL LACERATION     SKIN LACERATION
;
run;

options nodate nonumber missing = ' ';
ods escapechar='#';
ods pdf style=htmlblue file='program2.3.pdf';

proc freq
   data = ae;
   tables aeterm_aedecod;
run;

ods pdf close;
```

Program 2.3 yields the following output.



The SAS System

The FREQ Procedure

Dictionary-Derived Term				
AEDECOD	Frequency	Percent	Cumulative Frequency	Cumulative Percent
DIZZINESS	1	12.50	1	12.50
HEADACHE	2	25.00	3	37.50
LOWER LIMB FRACTURE	1	12.50	4	50.00
MYOCARDIAL INFARCTION	2	25.00	6	75.00
SKIN LACERATION	1	12.50	7	87.50
URTICARIA	1	12.50	8	100.00

You can see the benefit of coding the adverse events in the resulting summary. The headaches and myocardial infarctions are grouped appropriately, and splitting lightheadedness and facial laceration into separate events leads to those data being summarized separately as well.

However, there are some alternative, albeit poor, solutions to the free-text variable problem. One option is to *hardcode* events so that they are categorized properly. We will discuss hardcoding further in the next section, but it is generally a practice to be avoided as much as possible. Another option is to use a SAS DATA step string function such as SOUNDEX, INDEX, INDEXW, or SUBSTR to try to categorize data in like groups. This approach is very risky, because you cannot be guaranteed to capture all free-text data and categorize them the same way with these text-scanning tools. If the free-text data are unimportant, then such tools can be used. However, if the data are unimportant, then they probably should not be analyzed anyway and at best should be presented in some type of data listing.

Avoid Hardcoding Data

Sometimes even after clinical data management make a good attempt at cleaning and coding the data, you may find that the data still contain some undesired or discrepant values. Perhaps a variable was left uncoded, or perhaps there is a serious adverse event known to have occurred that has not yet been entered in the clinical database. When this happens, the statistical programmer may result to hardcoding. Hardcoding is explicitly stating the value of a symbolic object or variable in a program. An example of hardcoding follows.

Program 2.4 A Hardcoding Example

```
data endstudy;
   set endstudy;

   if subjid = “101-1002” then
      discterm = “Death”;
run;
```

In this example, it is known from non-database sources that at study termination, subject 101-1002 died. That information is hardcoded into the program and overrides the information coming from the clinical data management system. Here are two reasons why hardcoding is a bad practice:

- Hardcoding overrides the database controls in a clinical data management system. With hardcoding, there is no clear audit trail of data change, and CFR 21 – Part 11 controls might be considered compromised.
- Data often change in a trial over time, and the hardcode that is written today may not be valid in the future. Unfortunately, a hardcode may be forgotten and left in the SAS program, and that can lead to an incorrect database change.

Many organizations expressly forbid hardcoding in their SAS programming standard operating procedures, while others allow the practice. Occasionally, there may be a justifiable reason for hardcoding. For instance, there may be an upcoming *data safety and monitoring board* (*DSMB*) or *independent data monitoring committee* (*IDMC*) meeting where the clinical trial must be monitored for safety information using the best available data. If there is a critical adverse event that the statistical staff is aware of but it cannot be entered in the clinical data management system in time, then perhaps that would justify hardcoding. However, it is better to avoid hardcoding at all costs and instead correct data in the clinical data management system. If hardcoding must be done, then an approach like the following might be used.

Program 2.5 An Improved Hardcoding Example

```
data endstudy;
   set endstudy;

   **** HARDCODE APPROVED BY DR. NAME AT SPONSOR ON 02/02/2012;
   if subjid = "101-1002" and "&sysdate" <= "01MAY2012"d then
      do;
        discterm = "Death";
        put "Subject " subjid "hardcoded to termination reason"
             discterm;
      end;
run;
```

Note that this program uses SAS code comment text to indicate that hardcoding is being used and with whose approval. Requiring a keyword such as “HARDCODE” in the comment facilitates searches for hardcodes later. Also, note that a PUT statement is provided to the SAS log, verifying during program execution that hardcoding has been used. The hardcode in Program 2.5 has an expiration date. For example, if you know that you have an upcoming IDMC date next year, you can program the hardcodes to expire in the month that precedes the IDMC meeting.

In summary, for data to be useful in clinical trial analyses, they need to be quantifiable. The data must be either a continuous measure or a categorical value. Free text poses a problem for analysis, and, if it is a valuable variable for the statistical analyses, it really must be coded. Finally, hardcoding should be used only when absolutely necessary, because it is inherently problematic. Organizations that do allow hardcoding should document in their standard operating procedures (SOPs) that it is an approved business practice and how it is to be used.

Classifying Clinical Trial Data

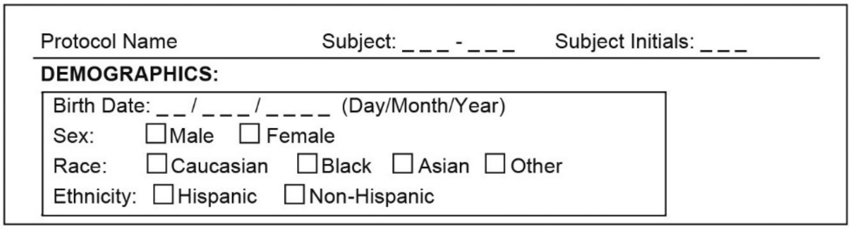
There are different ways to classify clinical trial data. As mentioned earlier, data can be classified by their physical nature into discrete chunks or as a more continuous measurable quantity. In clinical trials, there are other important contextual ways of grouping data as well. For instance, clinical trials are primarily focused on determining two things about a drug, biologic, or device: Is it efficacious, and is it safe? The data that help to answer these questions are broadly classified as *efficacy data* and *safety data,* respectively.

The Clinical Data Interchange Standards Consortium (CDISC) and its Submission Data Standards group have provided another way to broadly categorize clinical trial data. They have categorized data into *interventions class*, *events class*, *findings class,* and other special-purpose "*domains*" such as demographics. Interventions are the drug administration and surgical procedures that the patient receives during the course of the trial. Events are the unplanned clinical occurrences that the patient experiences over the course of the trial. Findings capture the planned examinations of the patient over the course of the trial. The demographics of a patient are that person's essential baseline characteristics.

The following sample CRF forms have been made to align with the CDISC CDASH standard.

Demographics and Trial-Specific Baseline Data

Here is a typical demographics CRF:



Protocol Name Subject: _ _ _ - _ _ _ Subject Initials: _ _ _

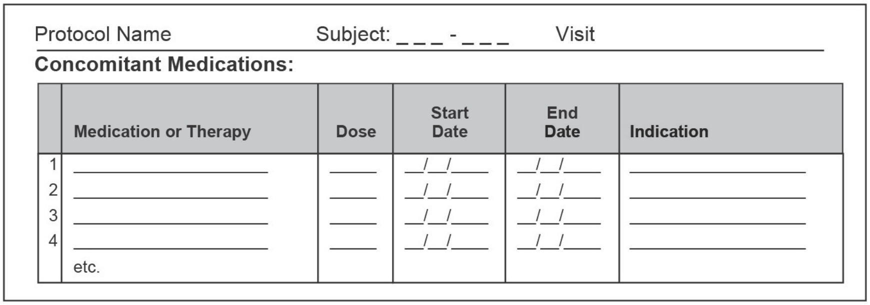
DEMOGRAPHICS:

Birth Date: _ _ / _ _ _ / _ _ _ _ (Day/Month/Year)
Sex: □Male □ Female
Race: □Caucasian □Black □Asian □Other
Ethnicity: □Hispanic □Non-Hispanic

Trial-specific patient characteristics may be included with the demographics data as well. Height, weight, smoking status, and sometimes vital signs are common additions. These measures are collected because they may be relevant to the therapeutic intervention and could be used to stratify the statistical analysis. Demographic and other baseline characteristics are used to define patient groupings, or *strata*, for subpopulation analyses, or they may be used as *covariates* during *inferential analyses*. Demographic and baseline characteristics are also commonly used to show that the therapeutic treatments under study have comparable populations at baseline. Demographics data fall into the special purpose demographics SDTM domain and play a part in efficacy and safety analyses, because either may be stratified by demographics and baseline characteristics. Other baseline subject characteristics would get stored in the subject characteristics SDTM domain or in the appropriate SDTM domain (e.g., blood pressure in vital signs).

Concomitant or Prior Medication Data

Concomitant medications and *prior medications* are collected in one of two forms: a list-type free-text format where the medications get coded later by data management, or a pre-categorized data format. Here is the free-text CRF format:

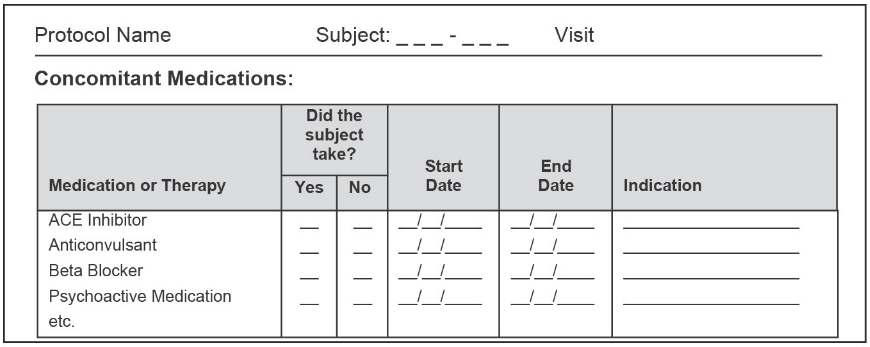


Protocol Name Subject: _ _ _ - _ _ _ Visit

Concomitant Medications:

	Medication or Therapy	Dose	Start Date	End Date	Indication
1	________________	____	__/__/____	__/__/____	________________
2	________________	____	__/__/____	__/__/____	________________
3	________________	____	__/__/____	__/__/____	________________
4	________________	____	__/__/____	__/__/____	________________
	etc.				

Here is the pre-categorized per protocol CRF format:



Protocol Name Subject: _ _ _ - _ _ _ Visit

Concomitant Medications:

Medication or Therapy	Did the subject take?		Start Date	End Date	Indication
	Yes	No			
ACE Inhibitor	__	__	__/__/____	__/__/____	____________________
Anticonvulsant	__	__	__/__/____	__/__/____	____________________
Beta Blocker	__	__	__/__/____	__/__/____	____________________
Psychoactive Medication	__	__	__/__/____	__/__/____	____________________
etc.					

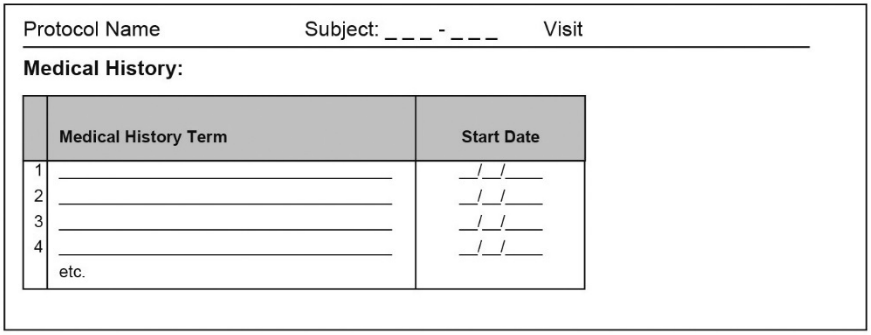
The free-text CRF format is useful in that it allows for an explicit description of the medication taken, whereas the pre-categorized format omits that detail. However, the free-text list format necessitates additional coding with a coding dictionary such as *WHOdrug* in order to be useful for analyses. The pre-categorized format has the benefit of capturing only the medications of concern for the given protocol and therapy and eliminates the cost of additional coding.

An essential detail for the statistical programmer to watch for in prior or concomitant medications data is whether or not the start and end dates are important for analyses. Unfortunately, it is often the case that the importance of the timing of prior or concomitant medications is not determined until after much of the data have been entered or even after the database is closed to entry. For instance, it may be decided later that a specific concomitant medication has to be watched carefully for interaction with a medication used in the study. If insufficient attention was placed on the quality of the medication start and end dates, then determining whether there is overlap with study medication is difficult if not impossible.

Concomitant or prior medications may be used in either safety or efficacy analyses. The presence of specific medications may be used as covariates for inferential analyses. Also, medications are often summarized to show that the therapies under study come from medically comparable populations. Medications may be used to determine protocol compliance and to help define a protocol-compliant study population. Concomitant medications may be examined to determine whether they interact with study therapy or whether they can explain the presence of certain adverse events. From a CDISC SDTM perspective, concomitant medications are considered an intervention.

Medical History Data

Like concomitant medication data, patient *medical history* data are collected in one of two forms: a list-type free-text format where the histories get coded, or a pre-categorized data format. Here is the free-text CRF format:

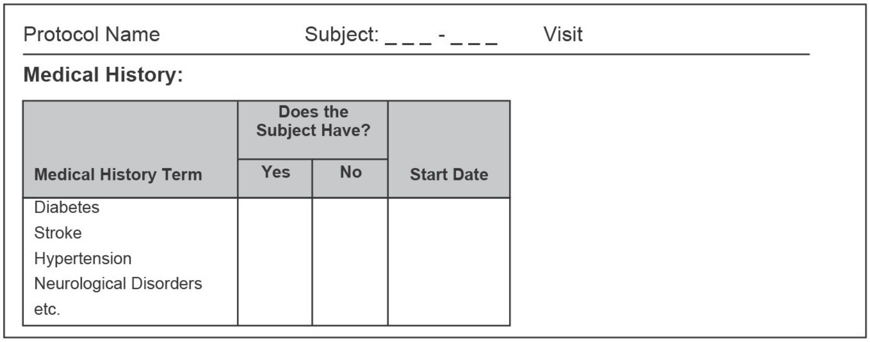


Protocol Name Subject: _ _ _ - _ _ _ Visit

Medical History:

	Medical History Term	Start Date
1	______________________________	__/__/____
2	______________________________	__/__/____
3	______________________________	__/__/____
4	______________________________	__/__/____
	etc.	

Here is the pre-categorized medical history CRF format:



Protocol Name Subject: _ _ _ - _ _ _ Visit

Medical History:

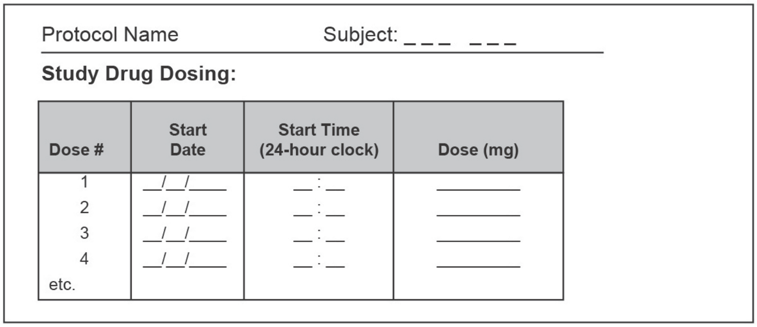
	Does the Subject Have?		
Medical History Term	Yes	No	Start Date
Diabetes Stroke Hypertension Neurological Disorders etc.			

Again, the free-text CRF format is useful in that it allows for explicit description of the historical condition, whereas the pre-categorized CRF format omits that detail. However, the free-text list format necessitates coding with a coding dictionary such as MedDRA in order to be useful for analyses. The pre-categorized format is useful here, because only medical history relevant to the investigational therapy can be captured and the cost of additional coding of the history data is eliminated entirely.

Medical history data may be used in either safety or efficacy analyses. The presence of historical medical conditions may be used as covariates for inferential analyses. Also, medical histories are typically summarized to show that the therapies under study come from study populations with comparable disease histories. Medical histories may be used to determine protocol compliance and to help define a protocol-compliant study population. Medical history is considered a finding from a CDISC SDTM perspective.

Investigational Therapy Drug Log

Drug logs, or drug exposure data, capture the investigational drug dosing times. Here is a sample drug log CRF form:



Protocol Name Subject: ___ ___

Study Drug Dosing:

Dose #	Start Date	Start Time (24-hour clock)	Dose (mg)
1	__/__/____	__ : __	________
2	__/__/____	__ : __	________
3	__/__/____	__ : __	________
4	__/__/____	__ : __	________
etc.			

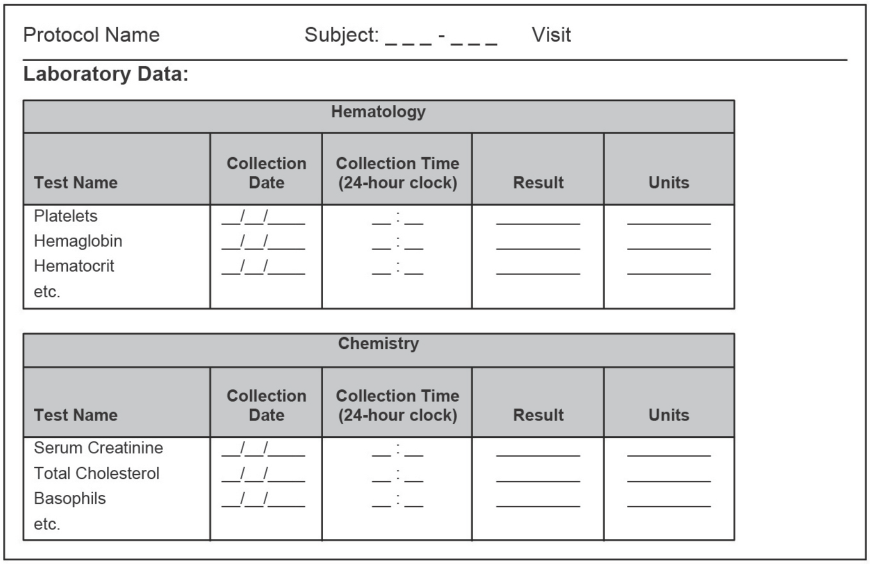
The investigational therapy drug log can be a source of problems for the statistical programmer. Here again, dates and times of dosing may be critical for effective use of this data. Missing dosing records, start times, or stop times can seriously hinder the quality of the reporting of dosing data. It is important to look at the analysis plan to determine if the dosing data are important to analysis. If they are important, then data management should clean the data to ensure the quality of the medication start and stop times.

Drug log or exposure data are used in many ways for both efficacy and safety analyses. As a safety issue, the drug record is often used in conjunction with adverse events to determine whether adverse events were treatment-emergent. In other words, did the patient have an adverse event that might have been caused by the investigational therapy? Also, drug log data may be used for safety analysis purposes to watch for abnormal laboratory values or other clinical events after dosing. Finally, drug log data are useful for determining protocol violations and can be used to determine treatment compliance. The drug log data are categorized as an intervention from a CDISC SDTM perspective.

Associated with drug log or drug exposure data is another type of data called drug accountability. This data captures the disposition of the study drug. It is not concerned with whether a patient was exposed to the drug but where the drug went. Drug accountability tracks data such as how many pills a patient was sent home with and how many they returned. It can be used to calculate protocol dosing compliance and is categorized as a finding from a CDISC SDTM perspective. Because the data is so interrelated, it is not uncommon to find data collection forms merge or integrate information from drug exposure and drug accountability.

Laboratory Data

Laboratory data may consist of many different collections of tests, such as ECG laboratory tests, microbiologic laboratory tests, and other therapeutic-indication-specific clinical lab tests. However, laboratory data traditionally consist of results from urinalysis, hematology, and blood chemistry tests. Traditional laboratory data can come from what are called local laboratories, which are labs at the clinical site, or from central laboratories where the clinical sites send their samples for centralized analysis. Often when the laboratory data come from a central laboratory, there is no CRF page for the data, and they are loaded into the clinical data management system directly from an electronic file. Local laboratory data may be represented with a CRF page such as this:



Protocol Name Subject: _ _ _ - _ _ _ Visit

Laboratory Data:

Hematology				
Test Name	**Collection Date**	**Collection Time (24-hour clock)**	**Result**	**Units**
Platelets	__/__/____	__ : __	_________	_________
Hemaglobin	__/__/____	__ : __	_________	_________
Hematocrit	__/__/____	__ : __	_________	_________
etc.				

Chemistry				
Test Name	**Collection Date**	**Collection Time (24-hour clock)**	**Result**	**Units**
Serum Creatinine	__/__/____	__ : __	_________	_________
Total Cholesterol	__/__/____	__ : __	_________	_________
Basophils	__/__/____	__ : __	_________	_________
etc.				

Laboratory data can pose a challenge to the statistical programmer in many ways. Simply obtaining the data can sometimes be difficult. Occasionally you have to work with a specialized local laboratory, and sometimes just getting the data to the statistics group in a usable format can be hard if CDISC CDASH and SDTM standards are not used. For example, the local laboratory staff may have used Microsoft Excel for data entry, and when they entered the data they entered rows within the columnar data with inconsistent formats, making machine readability of the resulting data file difficult. Another common issue is found within the "units" variable shown above. If local labs were used, it is likely that the lab units will have to be converted to a common unit for each laboratory test. Finally, laboratory values often need to be flagged as outside the normal range or perhaps outside the "clinical concern"/"panic range," where the latter is just a more extreme version of the former. Sometimes, the local or central laboratory flags these records, but it is not uncommon for the statistical programmer to have to make these assignments as well.

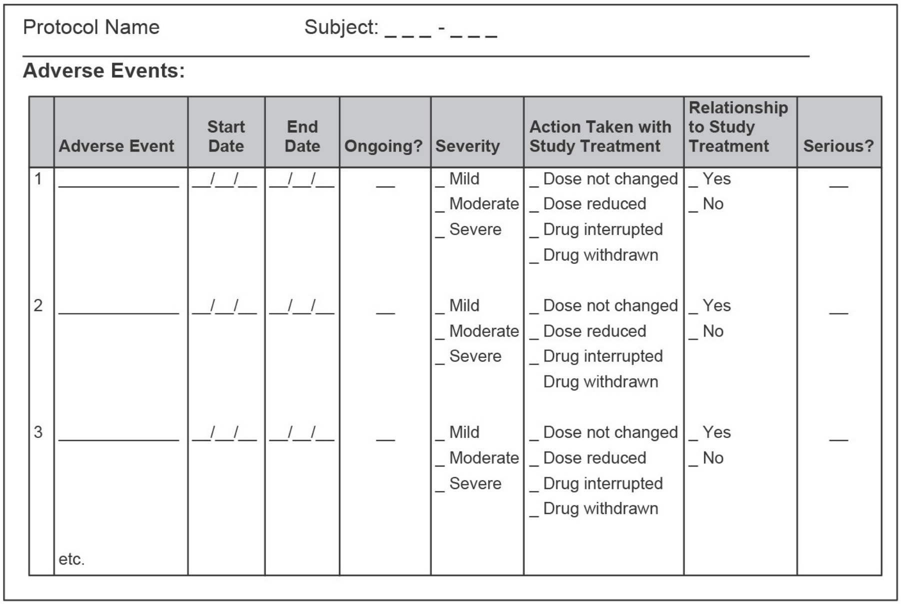
Laboratory data are most often associated with safety analyses, but they may play a part in efficacy analyses as well, especially if the laboratory data are part of the clinical endpoint definition. From a CDISC SDTM perspective, laboratory data are a finding, because they are a planned assessment. The CDISC SDTM has a number of specialized laboratory-like data domains besides LB for laboratory data. These domains that are very laboratory-like include EG for ECG data, VS for vital signs data, MB for microbiology, and PC and PP for pharmacokinetic data.

Adverse Event Data

In the FDA's "Guidance for Industry E6 Good Clinical Practice: Consolidated Guidance," an adverse event is defined as follows:

> Any untoward medical occurrence in a patient or clinical investigation subject administered a pharmaceutical product and that does not necessarily have a causal relationship with this treatment. An AE can therefore be any unfavorable and unintended sign (including an abnormal laboratory finding), symptom, or disease temporally associated with the use of a medicinal (investigational) product, whether or not related to the medicinal (investigational) product.

The adverse event form is fairly standard across clinical trials. The form consists of a list of events for which data are entered as free text and are later coded with a dictionary such as MedDRA and some associated event attribute variables. In just about any clinical trial, an adverse event form similar to the following sample will be found.



Protocol Name   Subject: _ _ _ - _ _ _

Adverse Events:

	Adverse Event	Start Date	End Date	Ongoing?	Severity	Action Taken with Study Treatment	Relationship to Study Treatment	Serious?
1	____________	__/__/__	__/__/__	__	_ Mild _ Moderate _ Severe	_ Dose not changed _ Dose reduced _ Drug interrupted _ Drug withdrawn	_ Yes _ No	__
2	____________	__/__/__	__/__/__	__	_ Mild _ Moderate _ Severe	_ Dose not changed _ Dose reduced _ Drug interrupted Drug withdrawn	_ Yes _ No	__
3	____________	__/__/__	__/__/__	__	_ Mild _ Moderate _ Severe	_ Dose not changed _ Dose reduced _ Drug interrupted _ Drug withdrawn	_ Yes _ No	__
	etc.							

The adverse event form is a cornerstone of patient safety monitoring, and as such it contains very important data. There are several data issues for the statistical programmer to be concerned about here.

Treatment-Emergent Signs and Symptoms

In guidance document ICH E3, "Structure and Content of Clinical Study Reports," the FDA defines *treatment-emergent signs and symptoms (TESS)* as "events not seen at baseline and events that worsened even if present at baseline." As simple as that may sound, it can sometimes be quite difficult to implement in programming. The important data variables that come into play are dosing record dates and times, adverse event start and stop times, and adverse event severity. All of these data variables need to be completed accurately for TESS to be calculated properly.

Serious Adverse Event Reconciliation

Just as there is an adverse event form, there is usually a *serious adverse event* (*SAE*) form. Note here that "serious" as defined by the FDA is different from "severe" on the adverse event form. A patient can have a "severe" headache that may not be considered "serious." The ICH guideline (also in ICH E3) entitled "Clinical Safety Data Management: Definitions and Standards for Expedited Reporting" defines serious adverse events as follows:

> A serious adverse event (experience) or reaction is any untoward medical occurrence that at any dose: results in death, is life-threatening, requires inpatient hospitalization or prolongation of existing hospitalization, results in persistent or significant disability/incapacity, or is a congenital anomaly/birth defect.

Historically, a separate CRF is used to capture serious adverse events, because those often must be reported to the FDA within 24 hours. Often, this means that the serious adverse events CRF data and the regular trial CRF adverse events are collected in different data tables, if not entirely different software systems. Pharmaceutical companies often want to reconcile the two databases to ensure that all serious adverse events appear in the regular-trial CRF adverse events database and that any event in the serious adverse events database is flagged properly as serious in the regular CRF adverse events database.

The problem is that the regular-trial adverse events database and the serious adverse events database do not join well if at all programmatically. You can attempt to join or merge the two databases by event start date and coded term, and that will join many regular-trial adverse events to the serious events. However, this is far from foolproof, because of mismatches in adverse event start dates and because the adverse events may have been coded slightly differently in the two systems. The best way to link the serious adverse events and adverse events databases is to have the clinical data management system create a linking variable key for you. In lieu of that, the only way to reliably link the two data sources is manually.

The good news is that with modern electronic data capture systems and the upcoming absorption of electronic health care data into clinical trials databases, the problem of reconciling adverse events to serious adverse event data will be fixed. Many electronic data capture systems now collect the serious and regular adverse event data in the same electronic form, which makes integration of the data unnecessary.

Concomitant Medication Reconciliation

Additional concomitant medication may be given in response to an adverse event, and especially with serious adverse events. Often you want to know precisely which medication was taken, but because that information may not be well captured on the adverse event form, there needs to be a linkage with the concomitant medications form. Once again, this is not something than can reliably be done with a program unless the clinical data management system creates a linking variable key behind the adverse event and concomitant medications forms. Some data management systems do this and, again, with electronic data capture, this is becoming more prevalent.

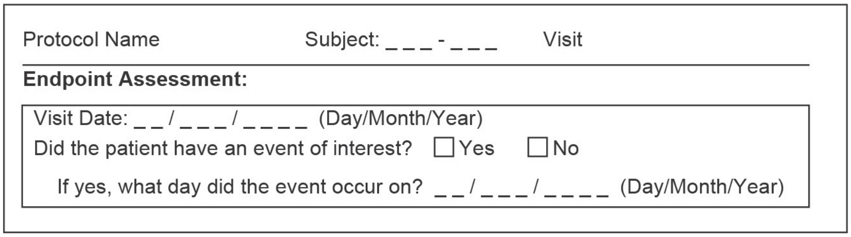
Laboratory Data Reconciliation

The adverse event for a patient may indicate a medical condition such as hypercholestimia, so there may be a request to ensure that there are elevated cholesterol laboratory data that can verify such a claim. You can sometimes make this kind of verification with programming if you know precisely which lab tests are involved and what level indicates a probable adverse event.

In the end, because of the importance of the data, it is imperative that the entire adverse event form data are cleaned. Reconciling the adverse event data with other clinical data in the clinical data management system can be very difficult if the data management system does not provide variable keys for linking such data. Adverse event data fall into the safety area of statistical analyses and are considered an event from a CDISC SDTM perspective.

Endpoint/Event Assessment Data

Endpoint or event assessments typically capture what the clinical trial was designed to study. For example, if a clinical trial were studying an anti-epilepsy medication, then the event form would likely collect seizure information. The endpoint or event assessment form is designed to collect data after the investigational drug or device intervention so that these data can be statistically compared to data from the patient's state before the drug or device intervention. Endpoint or event collection pages vary widely because of the broad range of ways to measure clinical disease, but here is a simplified sample endpoint collection page:



Protocol Name   Subject: _ _ _ - _ _ _   Visit

Endpoint Assessment:

Visit Date: _ _ / _ _ _ / _ _ _ _ (Day/Month/Year)

Did the patient have an event of interest? □ Yes □ No

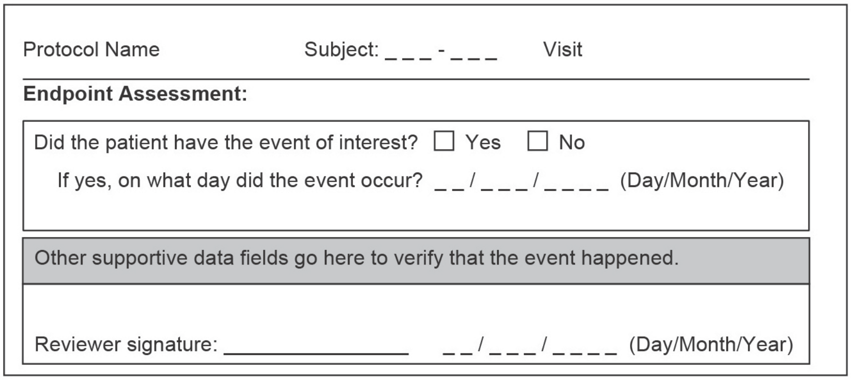
If yes, what day did the event occur on? _ _ / _ _ _ / _ _ _ _ (Day/Month/Year)

In this form, "event" would be replaced by some clinical finding such as "myocardial infarction," "stroke," "seizure," or the like. This example form is extremely simplified, because there are usually a number of associated event qualifying data variables captured as well. The event/endpoint page data must be clean, because it likely captures the primary efficacy data for the clinical trial.

The problem with endpoint data usually occurs when they need to be reconciled against data that are collected by the *clinical endpoint committee* (*CEC*), which we discuss next. The endpoint/event data are almost always used for efficacy analyses but may be used for safety analyses as well. From a CDISC SDTM perspective, the endpoint/assessment is often considered a finding, because it is a planned examination, but it could also be considered an unplanned event.

Clinical Endpoint Committee (CEC) Data

It is often the case that the endpoint/event form captures data that are not entirely objective because they contain some level of clinical judgment. For instance, when precisely is a cold cured, was an event truly a myocardial infarction, or did any given event truly occur? The clinical site investigator may decide, using his or her clinical judgment, that a given event occurred, but often it is necessary to have an independent assessment of that event by another physician. This independent review helps to ensure that events are reported in a consistent way across multiple clinical sites for a clinical trial. Usually what happens is that a condition on the regular case report form "triggers" the release of a CEC form to be sent to the CEC. The CEC then takes the CEC form and verifies whether or not an actual event occurred based on the data available in the patient's clinical records at the given site. A sample CEC form follows:



Protocol Name   Subject: _ _ _ - _ _ _   Visit

Endpoint Assessment:

Did the patient have the event of interest? □ Yes □ No

If yes, on what day did the event occur? _ _ / _ _ _ / _ _ _ _ (Day/Month/Year)

Other supportive data fields go here to verify that the event happened.

Reviewer signature: ________________   _ _ / _ _ _ / _ _ _ _ (Day/Month/Year)

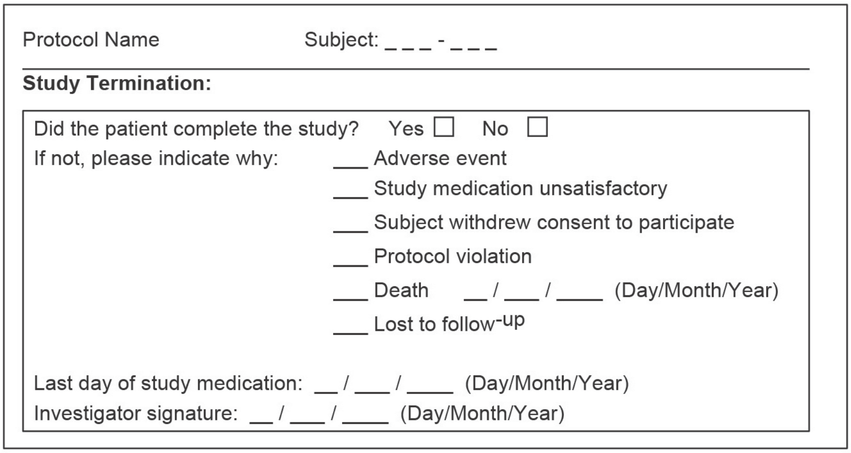
In this CEC form, "event" would be replaced by some clinical finding such as "myocardial infarction," "stroke," "seizure," or the like. Once again, this form is extremely simplified, and there are usually a number of associated data variables captured that help to support the existence of the event.

The biggest problem for the statistical programmer when using CEC data is reconciling these data against the regular CRF endpoint/event data. This can be a difficult task, especially when you consider that a patient may have more than one event on a given day. Fortunately, because the endpoint/event data are so critical to a clinical trial, the quality of the reconciliation from the CEC form to the CRF form is not often relegated to some form of fuzzy data join. Usually there will be a definitive linkage via a key mapping data set that links the CEC event data to the CRF event data. However, if that key data set does not exist, then the statistical programmer must prepare for some difficult programming. It is also worth noting that the data from the adverse event forms, laboratory forms, and other forms, as well as a specific "event" form, may in fact trigger clinical events. This may add to the complexity of the reconciliation programming.

The clinical endpoint committee data are almost always used for efficacy analyses, but they may also be used for safety analyses. From a CDISC SDTM perspective, the endpoint/assessment is considered a finding, as it is a planned examination.

Study Termination Data

The study termination form collects patient exit information from the clinical trial. Here is a sample study termination form:



Protocol Name Subject: _ _ _ - _ _ _

Study Termination:

Did the patient complete the study? Yes □ No □

If not, please indicate why:

___ Adverse event

___ Study medication unsatisfactory

___ Subject withdrew consent to participate

___ Protocol violation

___ Death __ / ___ / ____ (Day/Month/Year)

___ Lost to follow-up

Last day of study medication: __ / ___ / ____ (Day/Month/Year)

Investigator signature: __ / ___ / ____ (Day/Month/Year)

The study termination form data may be used for efficacy or safety analysis purposes. With regard to safety, if patients discontinue a study medication earlier than patients on standard therapy or placebo, then that is important to know. For efficacy analyses, patients who withdraw due to a lack of efficacy or adverse event may be precluded from being considered a treatment responder or success. Also, often the study termination date is used as a censor date in time-to-event analyses for therapy efficacy. Study termination forms play a key role in patient disposition summaries found at the start of a clinical study report. From a CDISC SDTM perspective, the study termination form is a finding.

Treatment Randomization Data

The *randomization* of a patient to a given therapy is the cornerstone of a randomized clinical trial. You may find these data in more than one place. They are often found within some form of *Interactive Voice Response System* (*IVRS*), but they may also be found in an electronic file that contains the treatment assignments or on the CRF itself. If randomization data are found on the CRF, they usually consist only of the date of randomization for treatment-blinded trials. IVRS data are often found outside the confines of the clinical data management system and usually consist of the following three types of data tables.

Randomization Scheme Data Set

The *randomization scheme* assigns a therapy randomly across a study population based on various stratification factors such as site, *blocking factor*, and perhaps subject demographics. There is no actual patient assignment information in this data table. Here is an example of a randomization scheme with a blocking factor size of four and a *treatment ratio* of 2:2:

Index	Site	Block	Treatment
1	101	1	Study Medication
2	101	1	Placebo
3	101	1	Study Medication
4	101	1	Placebo
5	101	2	Placebo
6	101	2	Placebo
7	101	2	Study Medication
8	101	2	Study Medication

Notice that treatment is randomly assigned within the given blocks and that there are two placebos and two study medications in each block. Also notice the “index” variable. The order of the randomization scheme is critical to the usefulness of the scheme, because that is the order in which patients are assigned treatment. If the order of the scheme is altered in any way, then the scheme is damaged.

Drug Kit List Data Set

The *drug kit list* is simply a list that shows which drug container/kit label goes with which study medication. It might look something like this:

Kit Number	Treatment
10000001	Study Medication
10000002	Study Medication
10000003	Study Medication
10000004	Study Medication
10000005	Study Medication

Drug Assignment Data Set

The *drug assignment data set* indicates which patient got which drug. It might look something like this:

Site	Subject	Treatment
101	0001	Study Medication
101	0002	Study Medication
101	0003	Placebo
101	0004	Placebo

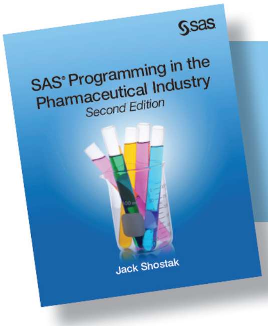
Note that the drug assignment data may not exactly match the order in the randomization scheme, because different patients pass screening procedures and are eligible for randomization at different times. Sometimes there are errors in treatment assignment, due to drug kits being misallocated or lost, that lead to a discrepancy between the drug assignment and the randomization scheme.

Other data sets may be found within the IVRS system that prove useful to the statistical programmer as well. Often the IVRS collects several baseline patient characteristics that are used in the stratification of the randomization scheme and subsequent assignment of study therapy. Finally, the preceding examples show in detail what the treatment variable is, in the “treatment” column. It is more often the case that the treatment variable is coded, such as “A” or “B” or “C.” It is of paramount importance that you know with absolute certainty how the treatment code can be properly interpreted.

The randomization data are used in both efficacy and safety analyses, because they are typically the key stratification variable for the trial. The randomization data allow you to answer the question of whether patients who are getting the study therapy fare better than the alternative. The CDISC SDTM allocated that actual treatment assignment information to the special demographics domain. The study therapy kit number would go in the CDISC SDTM DA domain.

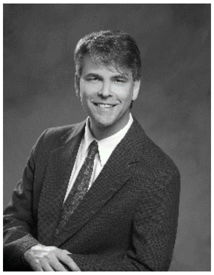
Quality-of-Life Data

Sometimes you may also see *quality-of-life* (*QOL*) *data* collected for your clinical trial. Quality-of-life data are collected to measure the overall physical and mental well-being of a patient. These data are usually collected with a multiple-question patient questionnaire and may be summed up in an aggregate patient score for analysis. Some commonly used quality-of-life questionnaires are the SF-36 and SF-12 Health Survey, but there are quite a few disease-specific QOL questionnaires available to clinical researchers. Quality-of-life data are often a subset of a type of data called patient-reported outcomes. They are patient-reported outcomes, because many times the patient reports them directly into a data collection tool, such as a website, themselves. From a CDISC SDTM standpoint, questionnaire data is classified as a finding.



Index

About The Author



Jack Shostak, Associate Director of Statistics, manages a group of statistical programmers at the Duke Clinical Research Institute. A SAS user since 1985, he is the author of *SAS Programming in the Pharmaceutical Industry*, and coauthor of *Common Statistical Methods for Clinical Research with SAS Examples, Third Edition*, as well as *Implementing CDISC Using SAS: An End-to-End Guide*. Shostak has published papers for the Pharmaceutical SAS Users Group (PharmaSUG) and the NorthEast SAS Users Group (NESUG), and he contributed a chapter, "Reporting and SAS Tool Selection," in the book *Reporting from the Field*. He is active in the Clinical Data Interchange Standards Consortium (CDISC) community, contributing to the development of Analysis Data Model (ADaM), and he serves as an ADaM trainer for CDISC.