

A "Big Picture" Data Diagnostic Program for Clinical Trials Data

**Dorothy E. Pugh, ClinTrials Research Inc.
Indrani Nandy, ClinTrials Research Inc.**

ABSTRACT

One of the first things those performing clinical trials data cleaning need to do is to get an overview of the relationships of the visit sequence numbers and the dates at which the various assessments associated with those visits took place. The SAS® program presented here helps show how to capture the relatively sparse anomalies in these relationships by comparing the date of each assessment to one or more user-specified reference dates. This technique can be used 1) to determine which assessment(s) should be used to determine the patient's first and last date in a protocol or on study drug, 2) to determine which assessments were typically performed at unusual times, 3) to catch obvious data errors, and 4) to spot possibly serious protocol violations.

INTRODUCTION

Clinical trials attempt to measure and report the safety and efficacy of a potential medication, called a "study drug." Patients participating in these trials provide written informed consent, then meet with participating physicians, known as "investigators," at regularly scheduled "visits" at which "assessments" such as physical exam and laboratory test data, measurement of compliance with prescribed dose of study drug, "adverse events," etc. are collected. For all these measurements, a date of such measurements is typically recorded on the Case Report Form. In protocols that allow dose to be changed, if the investigator sees the need to change a patient's study drug dose, a special schedule of prescribed doses enters the data. Important measures such as how long a patient was on study drug, how much study drug the patient received, whether the patient was on study drug at the time of an adverse event depend on the correctness of these dates, as does the validity of the clinical trial.

However, identifying date problems is not easy. Many date problems are revealed only when relationships between two dates, often from different files, are examined, and identifying and performing separately all possible comparisons would be an overwhelming task. Furthermore, it

is hard in some cases to develop a universal criterion for "red flags." For example, a patient's apparently being three days late to a visit typically does not present a problem. However, a discrepancy of three weeks could be one, and one of 364 days usually means the date contains the wrong year.

This paper discusses a special approach to identifying problem dates based on a different way of looking at clinical data and a new programming technique to implement this.

KEY CONSIDERATIONS

One important step is to make sure that certain key dates used in calculations are correct. Some dates must match other dates exactly because they are intended to represent the same information. Other dates may differ without an obvious error existing, but the degree to which they differ may determine whether a problem exists.

If the prescribed doses and reported intake differ, it is important to determine whether the existence of data errors or lack of compliance is the problem. Gross discrepancies should get a high priority in the data cleaning process.

Many protocol violations involve steps taken in the wrong sequence. If a patient's first date on study drug precedes the date the patient signed the informed consent form, an important violation exists. If the patient takes study drug before the investigator collects assessment data or baseline data on ongoing medical problems, a common problem with laboratory data, the assessment data is invalid and needs to be recorded for validation purposes during the analysis process.

Finally any assessments that are typically off-schedule would need special analysis consideration. Sometimes this can signify a problem that the project medical monitor needs to know about. Even when this does not represent a departure from this monitor's expectations, when the treatment duration is

crucial to the analysis, off-schedule assessment would need this special consideration.

PROGRAMMING APPROACH

Our approach to identifying "red flag" dates is to determine how they compare to both the scheduled dates and the actual visit date and dates of assessments associated with the same visit or milestone in the study. We have divided dates into three different classes: 1) those associated with events that occur at the start of a study, 2) those associated with events that occur at the end of a study, and 3) those associated with events that are scheduled to take place at each visit. Each date is compared only to other dates belonging to the same class, and in the case of the second class, only to dates associated with the same scheduled visit.

Some "first" dates that we compare include: 1) date patient gave written informed consent, 2) the earliest date on the study drug record (i.e., record of reported intake of study drug), 3) the date medical history and ongoing medical conditions are recorded, and 4) the date baseline characteristics were taken. Some "last" dates include: 1) the last date on the study drug record, 2) last date on the prescribed study drug record (in studies where dose is adjusted on an individual basis), and 3) the date that a patient was told to stop taking the study drug in association with an adverse event. Dates associated with all visits include those of 1) visit dates, 2) laboratory test dates, 3) study drug monitoring dates, 4) physical exam dates, and 5) other medical exam dates, e.g., ECG and EEG dates.

Our goal necessarily becomes one of optimization rather than perfection. We need to generate a "red flag" printout that contains no false negatives, and as few false positives as possible.

The approach we have chosen involves choosing a set of reference date variables and comparing their values with those of other date variables. For example, the first date on the study drug record might serve as a reference date for all "first" dates, and the visit dates might serve as reference dates for all assessments scheduled to be taken at all visits. In the simplest case, all discrepancies would be routed to an output file.

The basic programming approach we have chosen involves these steps: 1) merging all data sets containing the dates of interest by the patient ID and visit sequence number, after dropping all variables from each except the "by" variables and the visit date, putting the reference date data sets first, 2) using PROC TRANSPOSE to transpose the resulting data set BY the patient ID variables, and IDing the visit sequence number, 3) assigning a classification variable to each date based on whether it is a "first" date, etc., 4) sorting the new data set by this classification variable, and 4) using a RETAIN statement and the FIRST.x construction to compare non-reference dates with the reference date they are associated with (the reference date value being carried down by the RETAIN statement), 5) routing discrepancies between each reference date value and those of the other variables in its class to an output file.

THE CODE

This paper describes a programming approach rather a specific program. However, sections of code are given below to present a clearer idea of how we have done this:

So that date variables can be assigned classifications, these lists of date variable names are assigned to macro variables.

```
%let class1='date1' 'date4' ... ;
%let class2='date2' 'date5' ... ;
%let class3='date3' 'date6' ... ;
```

To the get the last date on study drug according to the study drug record, this code creates a data set with the latest date on the file, regardless of which visit it is associated with. Like all other dates in its classification, it is assigned an arbitrary value not associated with scheduled visit sequence numbers.

```
data temp.lastdrug(keep=patient enddte);
  set in.drug_rec(keep=patient startdte enddte
                  where=(enddte ne .));
  by patient startdte enddte;
  if last.patient;
  visseq = 99;
run;
```

Merge all dates of concern. Make sure that references dates are placed before other dates in

their classification to take advantage of the FIRST.X later on.

```
data temp.visits;

merge

%*****
*               Reference dates               *
*****;

indata.drug_rec(keep=patient visit_no startdt
  where=(visit_no eq 1)
  rename=(startdt=date1))
indata.visits(keep=patient visit_no date
  rename=(date=date2))
temp.lastdrug(keep=patient visit_no
  enddt
  rename=(enddt=date3))

%*****
*               Other dates                   *
*****;

indata.assess1(keep=patient visit_no date
  rename=(date=date4))
indata.assess2(keep=patient visit_no date
  rename=(date=date5))
indata.assess3(keep=patient visit_no date
  rename=(date=date6))
.
.
.
;
by patient visit_no;

*****
* Use transposition to compare multiple *
* assessment dates with a reference *
* date for the same visit. *
*****;

proc transpose data=temp.visits
  out=temp.out
  prefix=vis;
by patient;
id visit_no;
```

```
run;

*****
* Assign classifications to observations *
* based on _NAME_ value, the name *
* of the pre-transposition variables. *
*****;

data saswork.out;
  set saswork.out;
  if _name_ in (&class1) then class = 1;
  if _name_ in (&class2) then class = 2;
  if _name_ in (&class3) then class = 3;
run;
```

The output of a program following this general form is displayed in Figure 1.

CONCLUSION

The programming approach described in this paper represents an efficient method for developing easily validated programs that can scan large data sets for user-selected "red flags" and print out easy-to-understand diagnostic messages. We have used it to determine the correctness of many key analysis variables.

AUTHOR CONTACT INFORMATION

Dorothy E. Pugh
Senior Programmer/Analyst
ClinTrials Research Inc.
11000 Weston Blvd.
Cary, NC 27513
dpugh@clintrialsresearch.com

Indrani Nandy, Ph.D.
Senior Biostatistician
ClinTrials Research Inc.
11000 Weston Blvd.
Cary, NC 27513

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Figure 1.

Anomalous scheduled assessment dates (Date)
 Reference dates are CMP/FSTDTE (class 1: very first date only)
 EVO/VISDTE (class 2), and CMP/LSTDTE (class 3: very last date only)

where class eq 2 and daysdiff gt 7

----- Site Number=291 Patient Number=008 CRF Number=040892 -----

OBS	Variable name	Description	CLASS	VISIT	Date	In visit window?
170	URIDTE	urine test date on LABTST_G	2	Visit 2	17JAN1997	Y
171	BLODTE1	blood test date on LABTST_G	2	Visit 2	17JAN1997	Y
172	BLODTE2	date on TROLVL_G	2	Visit 2	17JAN1997	Y
173	INTLSTDT	intake date on TROLVL_R	2	Visit 2	17JAN1997	Y

OBS	Reference date	target date	first date of visit window	last date of visit window	date - target date	date - reference date
170	27DEC1996	30DEC1996	18NOV1996	09FEB1997	18	21
171	27DEC1996	30DEC1996	18NOV1996	09FEB1997	18	21
172	27DEC1996	30DEC1996	18NOV1996	09FEB1997	18	21
173	27DEC1996	30DEC1996	18NOV1996	09FEB1997	18	21