It's About Time!
A Primer on Time-Slotting of Data Using SAS ®

Maria Y. Reiss, Wyeth Research, Collegeville, PA

INTRODUCTION

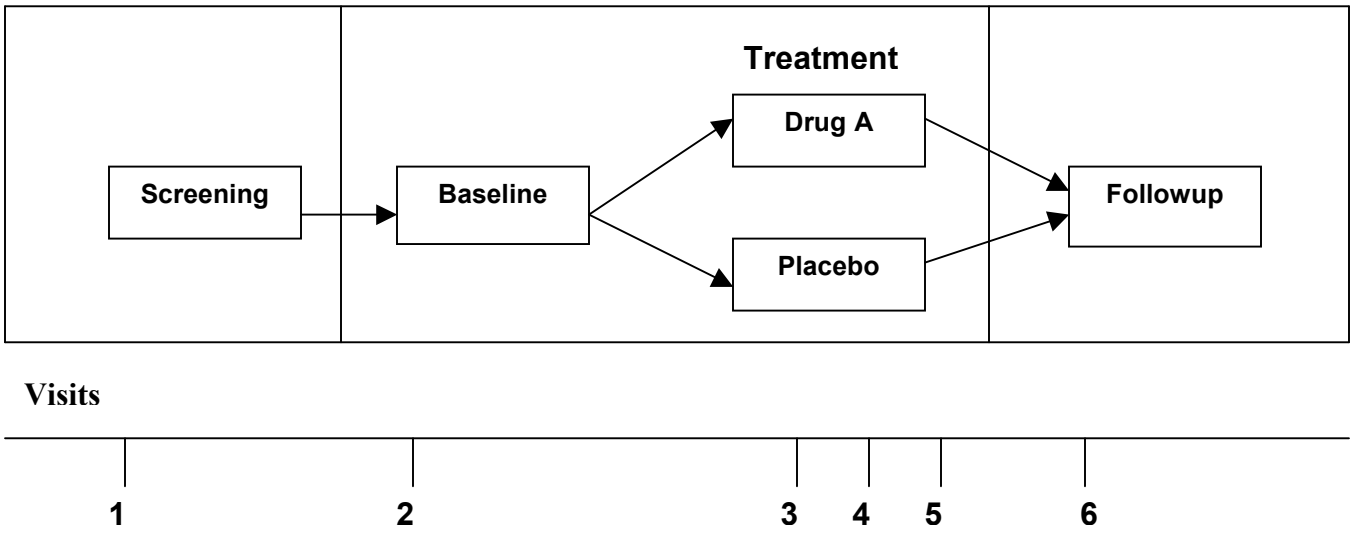
Many research studies have a set schedule for data collection. For example, in a clinical trial, the subjects are supposed to follow a set schedule of physician visits, laboratory tests, physical exams, etc. Consequently, the data collected during a clinical trial has a time component to it. A laboratory test result from the start of a clinical trial will be analyzed differently than a laboratory test result taken during the middle or the end of a clinical trial. Also, it can be important in the analysis of data to only compare events that occurred during the same relative time period. For these reasons, it is often necessary to slot data into time periods in order to properly analyze it. This paper explains the most common algorithms for slotting data into time periods. The paper focuses on the techniques used in the pharmaceutical industry since clinical drug trials are a classic example of research studies with multiple time periods. Even though the examples presented are from the pharmaceutical industry, the main concepts and techniques still apply to any research study with a time component.

This paper also addresses problems encountered while slotting data into time periods, such as handling of data from unscheduled visits and handling dirty or out-of-sequence data. The paper also presents sample SAS code for generating status reports and timeline plots that display a research study's actual schedule of data collection compared to its planned schedule of data collection.

Lastly, the paper will briefly discuss the future direction of time-slotting of data in the pharmaceutical industry. The Clinical Data Interchange Standards Consortium (CDISC) has recently developed a new data model for encapsulating information about the structure and design of clinical trials called the ***CDISC Trial Design Model*** (CDISC Submission Data Standards Team, 2005). This new model is likely to affect how time-slotting of clinical data will be handled in the future.

EXAMPLE DATA

Throughout this paper, a simple, fictional clinical trial will be used to illustrate concepts. This fictional trial, the ***Placebo-Controlled Parallel Study of Drug A***, consists of the following time periods.



During the Treatment Period, a subject will be randomized to take either Drug A or Placebo. The scheduled procedures for this study are:

Study Procedures	**Days –14 to –1**[a]	**Day –1**	**Day 1**	**Day 7**	**Day 14**	**Day 28**	**Day 42**
Study Interval	Screening	Baseline	Double-Blind Treatment Days 1 - 28				Followup
Visit ID	**1**	**2**		**3**	**4**	**5**	**6**
Informed consent	X						
Medical history	X						
Eligibility assessment	X	X					
Randomization		X					
Dispense study med. (Drug A or Placebo)	X	X		X	X		
First dose of study med			X				
Double-blind study med administration				X----------X			
Drug Accountability		X		X	X	X	
Physical examination	X					X	
Vital signs	X	X		X	X	X	
Laboratory evaluation	X			X	X	X	
Serum pregnancy	X					X	
Record prior/concomitant treatments	X	X		X	X	X	
Record adverse events	X	X		X	X	X	X
Efficacy Assessment Scale #1		X		X	X	X	
Efficacy Assessment Scale #2		X		X	X	X	
Followup Questionnaire							X

CURRENT METHODS OF TIME-SLOTTING DATA

I. Slotting Based on Sequence Number

The simplest method of slotting data into time periods is to assign a sequence number to each timepoint in the collection schedule. In the pharmaceutical industry, this sequence number is called a Visit Number. Each data record will contain the Visit Number of the visit when the data was collected. This way, the programmer or statistician can use the Visit Number variable to select data from a particular timepoint. If all data from the trial is collected at the scheduled times, then this system works very well. However, in reality, data may not be collected at the scheduled times. For example, in the hypothetical clinical trial, Visit 4 is supposed to occur after the subject has been taking the investigational new drug for 2 weeks. However, the subject may not come into the clinic for Visit 4 at the proper time. He or she may come in for her Visit 4 a week early due to conflicts with his or her personal schedule. When analyzing data, it may be important to only compare data from subjects who have taken the investigational new drug for the same amount of time. In this case, grouping data based on the Visit Number may not be suitable due to the fact that the subjects may not have adhered to the planned schedule of visits.

Another problem with slotting data based on Visit Number only is that extra, unplanned data may be collected. For example, during a clinical trial, a physician may order additional lab tests for a subject who is experiencing adverse events. Thus, the unplanned data will not have a Visit Number or it may be assigned a Visit Number such as “98”.

II. Slotting Based on Actual Date/time Values

The drawbacks with the method of slotting data based on Visit Number have led some pharmaceutical companies to adopt a more complicated but more precise method of slotting data into time periods – slotting data into time periods based on actual date/time values. The date/time values that are used to slot the data can be either collection date/times (such as laboratory test collection date/times) or event start date/times (such as Adverse Event start date/times). This method allows more accurate comparisons of data. For example, with this method, lab test results that were collected during the same relative time periods can be easily selected and compared. Also, this method can easily handle unscheduled visits. The main idea behind this method is to create meta-data about the time periods in the study. This is the same idea behind the CDISC Trial Design Model (CDISC Submission Data Standards Team, 2005).

At Wyeth Research, we follow these general steps in order to slot data based on actual date/time values:

1. Determine the major timepoints and the planned time periods of the study. At Wyeth Research, the major timepoints are called milestones.
2. Create a ***Milestone*** meta-data dataset containing each subject's actual milestone date/times.
3. Create a ***Time Period*** meta-data dataset containing each subject's actual schedule of time periods during the study.
4. Join the ***Time Period*** meta-data dataset with the data from Case Report Forms to assign Time Period information to each record in the Case Report Form data.

1. Determine the major timepoints and the planned time periods of the study.

The planned time periods for a study will be based on the analysis needs of the study. As Jack Shostak wrote in his book, "***SAS® Programming in the Pharmaceutical Industry***", "The definitions for valid windows are somewhat arbitrary and are typically based on clinical and statistical judgment from trial to trial." (Shostak, 2005).

For our hypothetical study, the ***Placebo-Controlled Parallel Study of Drug A***, I define the time periods to be: **Screening**, **Baseline**, **Treatment**, and **Followup**.

2. Create a Milestone meta-data dataset containing each subject's actual milestone date/times.

For our hypothetical study, the ***Placebo-Controlled Parallel Study of Drug A***, the milestone dates are Study Start, Baseline Visit, Therapy Start, Therapy Stop, and Study Stop. At Wyeth Research, we build a Milestone dataset with one record per subject per milestone:

Example of Time Milestones Meta-Data

Subject	Milestone Name	Milestone Datetime	Source of Milestone Datetime
001	Study Start	07FEB05	Informed Consent date
001	Baseline Visit	18FEB05	Randomization date
001	Therapy Start	19FEB05:08:00	First dose of study medication datetime
001	Therapy Stop	15MAR05:15:30	Last dose of study medication datetime
001	Study Stop	29MAR05	Conclusion of Study date

3. Create a Time Period meta-data dataset containing each subject's actual schedule of time periods during the study.

At Wyeth Research, we build this dataset from the milestone dataset created in Step 2. The dataset of actual time periods contains one record per time period per subject with the actual start and stop date/times of each time period. Note that the time periods in this example are inclusive on the start date/times and exclusive on the stop date/times. This Time Period meta-data dataset will later be joined with Case Report form data to apply time period labels to every record in the Case Report form data.

Example of Time Period Meta-Data

Subject	Time Period	Time Period Start Datetime	Time Period Stop Datetime	Planned Length of Time Period	Actual Length of Time Period
001	Screening	Study Start: 07FEB05	Baseline Visit: 18FEB05	14 days	11 days
001	Baseline	Baseline Visit: 18FEB05	Therapy Start: 19FEB05:08:00	1 day	1 day
001	Treatment	Therapy Start: 19FEB05:08:00	Therapy Stop + 2 days: 17MAR05:15:30	28 days	~26 days 15.5 hours
001	Followup	Therapy Stop: 17MAR05:15:30	Study Stop: 29MAR05	14 days	~12 days

Note that two days are added to Therapy Stop to create the ending time for the Treatment Period. It is a common practice in the pharmaceutical industry to add a lag time to the last dose of study medication to account for the time that the drug stays in the body of the subject. One convention is to extend the Treatment Period by a length of time equal to 5 times the half-life of the drug in the human body. Following this convention for our hypothetical clinical trial, I add a lag time of two days to the datetime of last dose of study medication.

The datetime variables in the Milestone and Time Period meta-data datasets are dynamically determined from the date/times recorded on Case Report forms. Since all data from Case Report forms may not be received at the same time, the meta-data datasets can change content as new Case Report Form data arrives. For example, if information about study medication doses is gradually added to the database and Therapy Start is determined from the earliest study medication dose and Therapy Stop is determined from the latest study medication dose in the database, the length of the treatment period will change every time new data about study medication doses arrives. At the beginning of the study, the treatment period may extend from Dose 1 to Dose 2, then later extend from Dose 1 to Dose 3, etc. Thus, the time-slotting meta-data for each subject will not be complete until all of the raw data arrives for the subject. Because of this, Wyeth Research uses temporary "override" milestone date/times to create time periods until the actual date/times become available in the database. For example, if all of a subject's study medication records are not yet in the database, a temporary Therapy Stop value of Therapy Start + 28 days can be used until the study medication records are complete. Or, if a subject's actual Study Stop date is not yet known, a date value of Therapy Start + 42 days can be temporarily used as the Study Stop date. Using this technique, all the Time Periods for a study can be created even if the actual milestone date/times are not yet available in the database. This technique also handles the problem of Case Report Form data arriving in the database out-of-sequence. This is not so much of a problem with electronic Case Report Forms, but with paper Case Report Forms, this was common. For example, Followup Questionnaires and Efficacy Questionnaires might arrive in the database prior to the study medication records. With the technique of temporary override dates, the dates of the Followup and Efficacy Questionnaires can still slot to the correct time periods.

Another consequence of dynamically creating the Milestone and Time Period meta-data from the date/times of Case Report forms is that dirty date/time data can cause incorrect time intervals to be created. For example, if the wrong year is recorded on a study medication form, a nonsensical Treatment period may be created. This type of error can cause errors such as the start of the Followup period preceding the start of the Treatment period. It is good practice to run edit checks on the Time Period meta-data to find gaps and overlaps in the Time Periods. If these are found, then the milestone date/times must be incorrect, and efforts should be made to correct the underlying Case Report form date/times. After the Case Report form data is complete and clean, then the Time Period meta-data will also be correct.

The Time Period metadata example listed earlier contains one large time window for the entire treatment period. Generally, a large window for the Treatment Period is sufficient for safety analyses of a drug. However, for efficacy analyses, it may be desirable to create narrower windows of time for the Treatment Period. An alternate scheme is:

Subject	Time Period	Time Period Start Datetime	Time Period Stop Datetime	Planned Length of Time Period	Actual Length of Time Period
001	Screening	07FEB05	18FEB05	14 days	11 days
001	Baseline	18FEB05	19FEB05:08:00	1 day	1 day
001	Treatment Week 1	19FEB05:08:00	26FEB05:08:00	7 days	7 days
001	Treatment Week 2	26FEB05:08:00	05MAR05:08:00	7 days	7 days
001	Treatment Week 3	05MAR05:08:00	12MAR05:08:00	7 days	7 days
001	Treatment Week 4	12MAR05:08:00	17MAR05:15:30	7 days	5 days 7.5 hours
001	Followup	17MAR05:15:30	29MAR05	14 days	~12 days

The starting and ending datetimes for the Week 1 through Week 4 treatment time periods can either be determined from Case Report Forms, or they can be determined by dividing up the treatment period by a formula (such as ¼ of treatment period in each week or including 7 days of treatment period in each week until one runs out of days in the treatment period).

The Time Period meta-data table provides information about each subject's actual schedule of events during a clinical trial. Thus, it is useful in providing status information about the trial. It is also enormously useful in that it can be joined with the Case Report Form datasets to apply time labels to each record from the Case Report Form datasets. At Wyeth Research, we have found that a few additional enhancements to the Time Period meta-data dataset are very helpful when joining the Time Period meta-data with the Case Report Form datasets:

- adding a Time Period ordering variable.
- adding a column for the actual treatment received during a time period.
- adding the time periods of Pre-study and Post-study. This time periods are useful for Medical History data and for Adverse Events reported after the end of the Study.

So with these revisions, sample Time Period meta-data tables look like:

Safety Time Periods (SAFTIME)

Subject	Randomized Treatment Group	Time Period	Time Period Order	Time Period Start Datetime	Time Period Stop Datetime	Actual Treatment Received
SUBJID	TRTGROUP	TMPERIOD	TMPERORD	TMSTART	TMSTOP	REGIMEN
001	Drug A	Pre-Study	1	07FEB95	07FEB05	None
001	Drug A	Screening	2	07FEB05	18FEB05	None
001	Drug A	Baseline	3	18FEB05	19FEB05	None
001	Drug A	Treatment	4	19FEB05:08:00	17MAR05:15:30	Drug A
001	Drug A	Followup	5	17MAR05:15:30	29MAR05	None
001	Drug A	Post-Study	6	29MAR05	29MAR15	None

Efficacy Time Periods (EFFTIME)

Subject	Randomized Treatment Group	Time Period	Time Period Order	Time Period Start Datetime	Time Period Stop Datetime	Actual Treatment Received
SUBJID	TRTGROUP	TMPERIOD	TMPERORD	TMSTART	TMSTOP	None
001	Drug A	Pre-Study	1	07FEB95	07FEB05	None
001	Drug A	Screening	2	07FEB05	18FEB05	None
001	Drug A	Baseline	3	18FEB05	19FEB05	None
001	Drug A	Treatment Week 1	4	19FEB05:08:00	26FEB05:08:00	Drug A
001	Drug A	Treatment Week 2	5	26FEB05:08:00	05MAR05:08:00	Drug A
001	Drug A	Treatment Week 3	6	05MAR05:08:00	12MAR05:08:00	Drug A
001	Drug A	Treatment Week 4	7	12MAR05:08:00	17MAR05:15:30	Drug A
001	Drug A	Followup	8	17MAR05:15:30	29MAR05	None
001	Drug A	Post-Study	9	29MAR05	29MAR15	None

These tables could also be combined into one table with an additional variable to distinguish the two types of time periods.

4. Join the Time Period meta-data dataset with the data from Case Report Forms to assign Time Period information to each record in the Case Report Form data.

Some types of Case Report Form data records may span multiple time periods (e.g. an Adverse Event begins during Screening, but it does not end until the Followup period). Despite this fact, it is still sometimes clearer to assign a single time period to a record. The Adverse Event that spans multiple time periods may be assigned the time period label for the time period when the Adverse Event started. To assign a single time period label to a record, the first step is to decide which date/time variable to use when slotting the record. Then a proc sql join statement can be used to join the Time Period meta-data dataset with the Case Report Form dataset. For example, a dataset storing Adverse Event records, ADVERSE, contains the start datetime of the Adverse Event in the variable STARTDT. This dataset can be joined with the Safety Time Period dataset, SAFTIME, as follows:

```
proc sql;
  create table newadverse as
  select a.*, b.tmperiod, b.tmperord, b.regimen
  from adverse as a
  left join saftime as b
  on (a.subjid = b.subjid) and (b.tmstart <= a.startdt < b.tmstop);
quit;
```

The output dataset, NEWADVERSE, will be identical to the original ADVERSE dataset except that each record will contain the additional variables TMPERIOD, TMPERORD, and REGIMEN. These new variables will indicate the time period when each Adverse Event records started and the subject's actual treatment when the Adverse Event started.

For example, the following record could be in the output dataset, NEWADVERSE.

Subject	Adverse Event Verbatim	AE Start Datetime	AE Stop Datetime	Time Period	Time Period Order	Actual Treatment Received
SUBJID	AEX	STARTDT	STOPDT	TMPERIOD	TMPERORD	REGIMEN
001	Headache	01MAR05		Treatment	4	Drug A

Applying the time labels to each record in this way allows a programmer or statistician to easily select data records that slot to a specific time period. In the example above, if a programmer wanted to select all Adverse Event records that started in the Treatment period, he or should would simply select NEWADVERSE records where tmperord = 4.

Sometimes, it beneficial to know all of the time periods that a record spans, not just the time period when the record started. For example, when tabulating Adverse Events by Time Period, you may want to count an Adverse Event in every time period that it spans. So you may want the Headache listed in the table above counted in the Treatment, Followup, and Post-Study periods, not counted only in the Treatment period. One technique for counting all of the time periods that a record spans is to join the Time Period meta-data dataset with the Case Report Form dataset as follows:

```
proc sql;
  create table newadverse as
  select distinct coalesce(trim(left(d.patient))), d.tmperiod, d.tmperord, d.regimen, e.*
  from saftime as d
  inner join adverse as e
  on d.patient = e.patient and
  ( d.tmstop >= d.tmstart or d.tmstop <= .Z ) and ( e.stopdt >= e.startdt or e.stopdt <= .Z )
  and ( e.startdt < d.tmstop or d.tmstop <= .Z ) and ( d.tmstart <= e.stopdt or e.stopdt <= .Z ) ;
```

The output dataset, NEWADVERSE will contain a record for every time period that the Adverse Event spans with the new TMPERIOD, TMPERORD, and REGIMEN variables. These new variables will indicate the time periods that the Adverse Event records span and the subject's actual treatment during those time periods.

Subject	Adverse Event Verbatim	AE Start Datetime	AE Stop Datetime	Time Period	Time Period Order	Actual Treatment Received
SUBJID	AEX	STARTDT	STOPDT	TMPERIOD	TMPERORD	REGIMEN
001	Headache	01MAR05		Treatment	4	Drug A
001	Headache	01MAR05		Followup	5	None
001	Headache	01MAR05		Post-Study	6	None

As stated in the introduction, extra data is sometimes collected during a study. Since a particular time period may contain multiple data records, a programmer or statistician may need to either average all the data records in a time period or choose a single record from the time period. In his book "***SAS® Programming in the Pharmaceutical Industry***", Jack Shostak describes a technique for choosing a single record. Essentially, he defines time intervals for visits with the center of the interval being the "ideal" target time for the visit. Then, he chooses the data record that is closest to the ideal target time as the official record for the visit. His book contains sample SAS code for selecting the target data record. (Shostak, 2005).

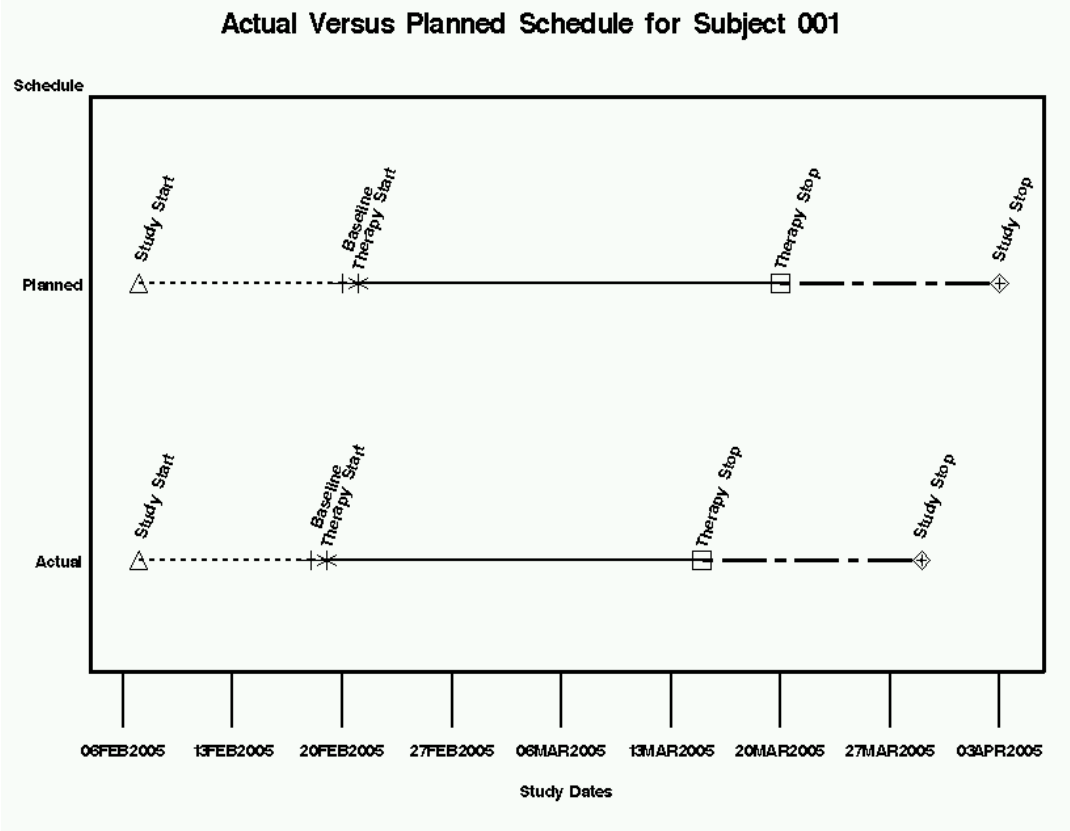
The method presented in this section, **Slotting Based on Actual Date/time Values**, is more complicated that Visit Number slotting, but it has several advantages:

- It allows easy selection and comparison of data collected during the same relative time periods.
- Creating a Time Period meta-data dataset for your study encapsulates information about the design and execution of your clinical trial and matches the CDISC Trial Design Model paradigm.
- The Time Period meta-data dataset can be joined with data that spans multiple time periods in tabulation programs.

STATUS REPORTS AND TIMELINE PLOTS

In this section, I am providing some example reports that can be generated from the Milestone or Time Period meta-data datasets.

Example Timeline Plot Comparing a Subject's Planned and Actual Schedule in a Study



This timeline plot was produced from a modified Time Period meta-data dataset using the annotate facility with PROC GPLOT. The SAS code for generating this timeline plot is in Appendix A of this paper. For an excellent explanation of how to produce timeline plots using SAS, see the paper "*Fun With Timelines: Doing More With SAS/Graph® PROC GPLOT*" by Barbara B. Okerson (Okerson, 2002). Also, if you would like more information about this timeline plot or the SAS code for creating it, please contact me using the Contact Information provided at the end of the paper.

Example Status Report for Subjects in a Two Period Crossover Clinical Trial

This report is produced using data step manipulation of a Milestone meta-data dataset and PROC REPORT.

Subject Participation Status
Dates of Each Study Period

INVESTIGATOR: 001, USA, Smith, John
TREATMENT SEQUENCE: Drug A 40 mg/Drug B 40 mg

	Screening		_____Period 1_____			Wash Out	_____Period 2_____				
Subject	Start Date	Duration	Start Date	Stop Date	Duration	Duration	Start Date	Stop Date	Duration	Conclusion Date	Followup Date
000001	21JUL05	18	07AUG05 08:07	13AUG05 08:32	7	17	30AUG05 08:00	05SEP05 08:40	7	19SEP05	19SEP05
000002	22JUL05	20	10AUG05 07:55	16AUG05 08:30	7	16	01SEP05 08:05	07SEP05 08:30	7	24SEP05	24SEP05
000003	22JUL05	15	05AUG05 08:15	11AUG05 08:35	7	17	28AUG05 08:45	03SEP05 08:37	7	17SEP05	17SEP05
000004	22JUL05	15	05AUG05 08:30	11AUG05 08:30	7	17	28AUG05 08:00	03SEP05 08:35	7	18SEP05	18SEP05
000005	24JUL05	18	10AUG05 08:00	16AUG05 08:32	7	16	01SEP05 08:12	07SEP05 08:32	7	24SEP05	24SEP05

FUTURE DIRECTION OF TIME-SLOTTING IN PHARMACEUTICAL INDUSTRY

The CDISC Study Data Tabulation Model V3.1.1 contains a new standard model for describing the "key aspects of the planned conduct of a clinical trial in a standardized way", the *Trial Design Model*. One of the purposes of this model is to allow reviewers to "compare planned and actual treatments and visits for subjects in a clinical trial." Due to space and time limitations, I can only highlight the key features of the Trial Design Model, but as pharmaceutical companies adopt CDISC, the Trial Design Model will provide a standard way of reporting the time component of clinical trials.

The CDISC Trial Design Model introduces several new terms for the planned schedule of a clinical trial:

- Element – "a basic building block for time within a clinical trial" with "a description of what happens to the subject during the Element", "a definition of the start of the Element", and "A rule for ending the Element."
- Arm – "a planned sequence of Elements, typically equivalent to a treatment group."
- Epoch – "the portion of a blinded trial that corresponds to an individual element".
- Visit – "a clinical encounter that encompasses planned and unplanned trial interventions, procedures, and assessments that may be performed on a subject."

(CDISC Submission Data Standards Team, 2005).

The **Elements** of our example clinical trial, the ***Placebo-Controlled Parallel Study of Drug A*** are Pre-Study, Screening, Baseline Visit, Drug A, Placebo, Followup, and Post-Study. Our example clinical trial has two **Arms**:

1) Pre-Study, Screening, Baseline Visit, Drug A, Followup, Post-Study
2) Pre-Study, Screening, Baseline Visit, Placebo, Followup, Post-Study

while the **Epochs** are: Pre-Study, Screening, Baseline Visit, Treatment, Followup, and Post-Study. The **Visits** are, of course, Visits 1 through 6.

The CDISC Trial Design Model describes three meta-data tables that contain the data for the planned schedule of a clinical trial:

- Trial Elements Table
- Trial Arms Table
- Trial Visits Table

The layout of these tables is described in Chapter 7 of the ***CDISC SDTM Implementation Guide (Version 3.1.1)*** that is available for downloading from the CDISC web site (http://www.cdisc.org).

The CDISC Trial Design Model also describes two meta-data tables for storing information on a subject's actual data collection during a trial.

- Subject Visits Table

The Subject Visits Table contains one record per Subject per Actual Visit.

- Subject Elements Table

The Subject Element Table contains one record per Subject per Element
As an example, a Subject Element Table for our example clinical trial might look like:

Study Identifier	Unique Subject Identifier	Subject Element Code	Description of Subject Element	Start Date/Time of Element	End Date/Time of Element
STUDYID	USUBJID	ETCD	ELEMENT	SESTDTC	SEENDTC
123-456	001	01	Pre-Study	1995-02-07 T00:00:00	2005-02-07 T00:00:00
123-456	001	02	Screening	2005-02-07 T00:00:00	2005-02-18 T00:00:00
123-456	001	03	Baseline Visit	2005-02-18 T00:00:00	2005-02-19 T00:00:00
123-456	001	04	Drug A	2005-02-19 T08:00:00	2005-03-17 T15:30:00
123-456	001	05	Followup	2005-03-17 T15:30:00	2005-03-29 T00:00:00
123-456	001	06	Post-Study	2005-03-29 T00:00:00	2015-03-29 T00:00:00

Note: Only a subset of the CDISC prescribed columns are listed here. See the ***CDISC SDTM Implementation Guide (Version 3.1.1)*** for a full description of this table.

As you can see, the Subject Elements table closely resembles the Time Period meta-data table described in the **Slotting Based on Actual Date/time Values** section of this paper. This table could be used in the same manner as the Time Period meta-data table to add time period labels to Case Report Form data.

CONCLUSION

Many types of data have a time component. When analyzing such data, it is crucial to understand and consider the time component. This paper summarizes techniques that one can use to create metadata about the time aspects of data and to slot data into time periods. Although there are issues that one must consider when using these techniques, such as dirty or out-of-sequence data, the use of these techniques makes it easier to analyze the time aspect of data.

ACKNOWLEDGEMENTS

I would like to thank Jeff Goodrich, Ellen Meier, and Chao Luo of Wyeth Research for all they have taught me on the topic of time-slotting of data, clinical data, and SAS programming in general.

REFERENCES

CDISC Submission Data Standards Team. 2005. *Study Data Tabulation Model Implementation Guide: Human Clinical Trials (Version 3.1.1)*. Clinical Data Interchange Standards Consortium.

Okerson, Barbara. 2002. *Fun with Timelines: Doing More With SAS/GRAPH® PROC GPLOT*. SUGI 27, Paper 218-27.

SAS Institute Inc. 2005. SAS Institute FAQ. *How Do I Create a Timeline Plot with SAS/GRAPH®*? http://support.sas.com/faq/020/FAQ02016.html.

Shostak, Jack. 2005. *SAS® Programming in the Pharmaceutical Industry*. Cary, NC, USA: SAS Institute Press.

CONTACT INFORMATION

I welcome your comments and questions. You may contact me at:

Maria Y. Reiss
Wyeth Research
500 Arcola Road
Collegeville, PA 19426
Email: reissm@wyeth.com
Phone: (484) 865-5694

APPENDIX A

Sample SAS code for generating a timeline plot from Time Period meta-data using the annotate facility with PROC GPLOT is listed below. For simplicity, the data and code for processing only one subject is listed. The code could easily be modified to handle multiple subjects.

The input Time Period meta-data is in a dataset called MILESTN in the format:

Subjid	Type	Study Start	Baseline Visit	Therapy Start	Therapy Stop	Study Stop
001	Actual	07FEB2005	18FEB2005	19FEB2005	15MAR2005	29MAR2005
001	Planned	07FEB2005	20FEB2005	21FEB2005	20MAR2005	03APR2005

Type is either Actual or Planned.

```
%* Start of Program;

goptions reset=all ftext=zapf ftext=swissb htext=0.9 htitle=1.5;

proc format;
    value $typefmt
      'Actual'  = 1
      'Planned' = 2;
    value typefmt
      1 = 'Actual'
      2 = 'Planned';
run;


data anno;
  length function $8 text $20;
  retain xsys ysys '2' size 2 height 2 angle 70;
  set milestn end=last;
  line=2;
  function='move'; x=milestn1; y=typen; output;
  function='symbol'; size=2; text='triangle'; output;
  function='LABEL'; size=.9; text="Study Start"; x=milestn1+.8;
y=typen+.24; output;
  function='draw'; size=2; x=milestn2; y=typen; output;
  function='symbol'; size=2; text='plus'; output;
  function='LABEL'; size=.9; text="Baseline"; x=milestn2+.8;
y=typen+.24; output;
  line=2;
  function='move'; x=milestn2; y=typen; output;
  function='draw'; size=2; x=milestn3; y=typen; output;
  function='symbol'; size=2; text='star'; output;
  function='LABEL'; size=.9; text="Therapy Start"; x=milestn3+.8;
y=typen+.24; output;
  line=1;
  function='move'; x=milestn3; y=typen; output;
  function='draw'; size=2; x=milestn4; y=typen; output;
  function='symbol'; size=2; text='square'; output;
  function='LABEL'; size=.9; text="Therapy Stop"; x=milestn4+.8;
y=typen+.24; output;
  line=9;
  function='move'; x=milestn4; y=typen; output;
  function='draw'; size=2; x=milestn5; y=typen; output;
```

```
  function='symbol'; size=2; text='diamond'; output;
  function='LABEL'; size=.9; text="Study Stop"; x=milestn5+.6;
y=typen+.24; output;
  format typen typefmt.;
  run;


  axis1 order=(1 to 2) offset=(6,10) minor=none major=none
label=("Schedule");
  axis2 order=('06feb2005'd to '03apr2005'd by week) offset=(3,4)
major=(height=3 width=2)
       minor=none width=3 label=("Study Dates");


  title1 'Actual Versus Planned Schedule for Subject 001';

  proc gplot annotate=anno;
  plot typen*milestn5 / vaxis=axis1 haxis=axis2 legend=legend1;
  run;
```