PharmaSUG2011 - Paper HW05

From SAP to BDS: The Nuts and Bolts

Nancy Brucken, i3 Statprobe, Ann Arbor, MI
Paul Slagle, United BioSource Corp., Ann Arbor, MI

ABSTRACT

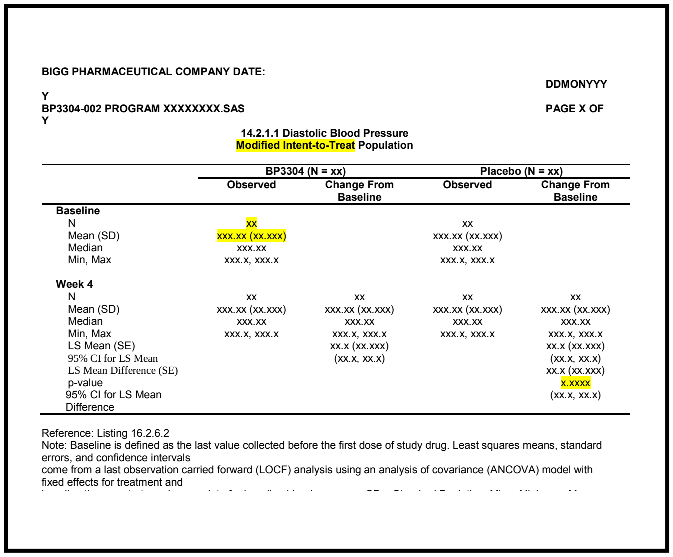
You've just read through the protocol, the analysis plan, and the table shells for the study you were recently assigned to, and have noticed that the primary efficacy analysis is an ANCOVA on change from baseline for one of the vital signs. Furthermore, there are quite a few tables summarizing change from baseline over multiple visits, as well as subjects with abnormal or clinically significant responses. Your manager has informed you that your analysis datasets will need to meet the Analysis Data Model (ADaM) specifications, version 2.1. You know what analysis datasets are, but how do you make them ADaM compliant?

This workshop will walk participants to process in creating the Basic Data Structure (BDS) analysis dataset defined in the ADaM v 2.1 model and implementation guide. Participants will create the ADaM based specifications and metadata, based on the tables being created. The workshop will address such topics as what the required variables are for the Basic Data Structure, understanding when to carry forward SDTM variables, how to add flags and timing variables, and knowing when to create new variables or new records in the analysis dataset.

TIME TO ROLL UP THE SLEEVES?

THE BEST PLACE TO START IS AT THE END

We all know that the source of our data is in the SDTM datasets and we have an SAP and mock up of the table to be created. This is a good starting point. For this exercise, we are going to start on one of the more critical tables to be developed, the ANCOVA analysis of the vital signs data. Specifically, let's start with table 14.2.1.1., the Diastolic Blood Pressure Analysis.



BIGG PHARMACEUTICAL COMPANY DATE: **DDMONYYY**
Y
BP3304-002 PROGRAM XXXXXXXX.SAS **PAGE X OF**
Y

14.2.1.1 Diastolic Blood Pressure
Modified Intent-to-Treat Population

	BP3304 (N = xx)		Placebo (N = xx)	
	Observed	Change From Baseline	Observed	Change From Baseline
Baseline				
N	xx		xx	
Mean (SD)	xxx.xx (xx.xxx)		xxx.xx (xx.xxx)	
Median	xxx.xx		xxx.xx	
Min, Max	xxx.x, xxx.x		xxx.x, xxx.x	
Week 4				
N	xx	xx	xx	xx
Mean (SD)	xxx.xx (xx.xxx)	xxx.xx (xx.xxx)	xxx.xx (xx.xxx)	xxx.xx (xx.xxx)
Median	xxx.xx	xxx.xx	xxx.xx	xxx.xx
Min, Max	xxx.x, xxx.x	xxx.x, xxx.x	xxx.x, xxx.x	xxx.x, xxx.x
LS Mean (SE)		xx.x (xx.xxx)		xx.x (xx.xxx)
95% CI for LS Mean		(xx.x, xx.x)		(xx.x, xx.x)
LS Mean Difference (SE)				xx.x (xx.xxx)
p-value				x.xxxx
95% CI for LS Mean Difference				(xx.x, xx.x)

Reference: Listing 16.2.6.2
Note: Baseline is defined as the last value collected before the first dose of study drug. Least squares means, standard errors, and confidence intervals
come from a last observation carried forward (LOCF) analysis using an analysis of covariance (ANCOVA) model with fixed effects for treatment and

Now that we have the table mock, let's start dissecting it a bit. First, we know the selection criteria are based on the MITT population. Because we are working with the ADaM data structures, we know that this will require the MITTFL flag created in the Subject Level Analysis Dataset (ADSL). Creating the ADSL dataset is not covered in this workshop, but creating it is a fairly straight forward process with each subject having a single row in the dataset. The MITTFL is one of the variables contained in that row, and has a value of either 'Y' or 'N'.

The next thing to look at is the individual analysis. While it is a very simple analysis, it is the first analysis shown on the report and a good place to start. The first item on the table is the number of observed subjects in the modified intent-to-treat population. This analysis is also based on observations at the baseline visit. Since we are probably going to use a single procedure to compute all of the basic descriptive statistics, let's also plan on computing the mean and standard deviation. We now have enough details to start creating the Analysis Results Metadata.

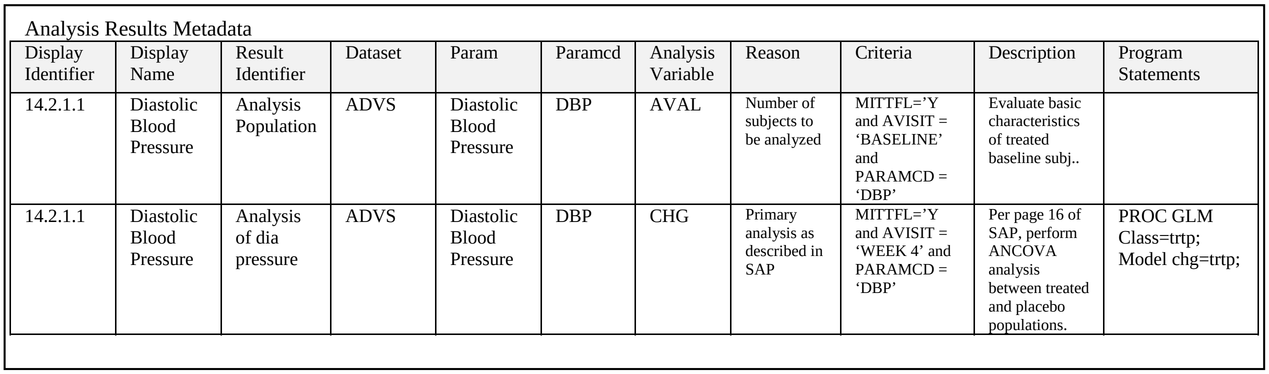
The Analysis Results Metadata describes, for each cell on a report, how the data is collected and analyzed. The details of this Metadata are described on page 22 of the ADaM Specifications Document (not the Implementation Guide). The first two fields to complete are the Display Identifier (table number) and the Display Name (name of the table). The Result Identifier is the unique description for what we are calculating, which is the Number of Subjects Analyzed.

The next four fields we need to create in the Analysis Results Metadata define where we are going to get the data to analyze. First, let's think about where we are going to be getting this data. We will want to create an analysis dataset just for vital signs data. So, let's call that dataset ADVS. We'll talk more about this later, but understand that each value should be a row in the dataset described by the PARAM and PARAMCD variables. So, for our needs we will use the rows where PARAM contains ‘Diastolic Blood Pressure’ and the PARAMCD value is ‘DBP’. Since our highlighted fields describe the actual results, we are going to analyze the values in the AVAL field.

Now that we know what we are going to analyze, we need to describe the why and how. The reason for calculating the ‘n’ is to identify the number of subjects being analyzed while the ‘mean (sd)’ describes the mean and standard deviation of the results in that population. We know that we are going subset the variable based on the MITTFL=‘Y’ and AVISIT = ‘BASELINE’ but we also need to identify that we will only use records with PARAMCD=‘DBP’. The reason we are performing this analysis is to generate basic descriptive statistics to describe the diastolic blood pressure of treated subjects at baseline. We also know that we will be using the UNIVARIATE procedure to produce these results.

That takes care of a simple analysis, but what about something a little more complicated? Let’s take a look at the p-value calculation for the Week 4 analysis. The Display Identifier and the Display Name are both the same as in the prior analysis. We are also going to be analyzing the diastolic blood pressure results, so the DATASET, PARAM, and PARAMCD fields are the same. However, we are analyzing change from baseline and not the actual results. Rather than calculating the change from baseline in the table program, we will use the CHG values stored in the analysis dataset. The reason we are doing the analysis is because it is the primary efficacy analysis as specified in the protocol. The selection criteria can be described as selecting the records where MITTFL=‘Y”, PARAMCD=‘DBP, and AVISIT = ‘VISIT 4’. The description for doing all of this can be found in the SAP. On page 11 of the SAP, we see the first bullet describes this as one of the efficacy parameters to be analyzed. We also see on page 16 of the SAP that we will be analyzing the change in blood pressure using the ANCOVA model.

Now we have all the information needed to complete the Analysis Results Metadata for the variables we have identified in the table. An abbreviated example of the spreadsheet used for this is attached below. Each variable on each unique table is identified in the same manner we have just completed. As you go through this, you are building the information necessary to create the next step of the ADaM structures, the Analysis Variable and Analysis Parameter Metadata.



Analysis Results Metadata

Display Identifier	Display Name	Result Identifier	Dataset	Param	Paramcd	Analysis Variable	Reason	Criteria	Description	Program Statements
14.2.1.1	Diastolic Blood Pressure	Analysis Population	ADVS	Diastolic Blood Pressure	DBP	AVAL	Number of subjects to be analyzed	MITTFL=’Y and AVISIT = ‘BASELINE’ and PARAMCD = ‘DBP’	Evaluate basic characteristics of treated baseline subj..	

14.2.1.1	Diastolic Blood Pressure	Analysis of dia pressure	ADVS	Diastolic Blood Pressure	DBP	CHG	Primary analysis as described in SAP	MITTFL=’Y and AVISIT = ‘WEEK 4’ and PARAMCD = ‘DBP’	Per page 16 of SAP, perform ANCOVA analysis between treated and placebo populations.	PROC GLM
										Class=trtp;
										Model chg=trtp;

PUTTING THE DATA IN ITS PLACE

As you complete the above process, you can start to see what data you are going to need to create the report. Since we started with the Diastolic Blood Pressure as our example, we will continue using that, knowing that other vital sign values will also need to be analyzed.

For what we defined above, we will need the actual values and the change from baseline values for Diastolic Blood Pressure for the Baseline Visit, Week 4, Week 8, Week 12, Week 16, Week 20, Follow-up, and End of Study. In addition to the value and timing information described above, we are also going to need to identify the population each value is to be reported in. Fortunately, we can pull all of the above information from the Vital Signs (VS) Study Data Tabulation Model (SDTM) domain and the Subject Level analysis dataset (ADSL).

Since this workshop is focused on the creation of the Basic Data Structures used to support reporting, we are going to assume that the ADSL dataset has already been created and described. We are also going to assume that the VS SDTM database has also been created. There are other workshops and classes available which will help the participant create these domains.

So let’s get started on creating the ADVS database. Within the BDS definitions, we know that there are 5 required variables for any analysis dataset. These are the STUDYID, USUBJID, PARAM, PARAMCD, and either AVAL or AVALC (one for numeric values, the other for character). The STUDYID is self-explanatory and, for this workshop, should be set to ‘BP3304-002’ for all records.

When creating USUBJID, you should consider that all of these datasets may be concatenated sometime in the future to support integrated reporting. For this reason, the unique subject id should be a combination of the study ID and the subject identifier. As an example, for this workshop we would have subject ‘A01’ entered as ‘BP3304-002-A01’. As you can tell, for some studies this can create a rather lengthy value for this field. Because of this, the developer may want to consider including the permitted variable SUBJID, which can be used to just represent the subject identifier, ‘A01’ in this example.

In the past, to create the analysis dataset for a report many programmers would have then created a unique variable for each type of analysis performed. For the ADaM BDS, this is not allowed. The BDS uses a much narrower, normalized data structure where all variables become key parameters used to describe the AVAL value. The purpose of the required fields PARAM and PARAMCD is to describe what AVAL represents. In our example, we would populate PARAM with ‘Diastolic Blood Pressure’ and PARAMCD with ‘DBP’. Each result would then be stored in the AVAL variable. This basic data structure is shown at the right.

Very [very] Basic ADVS dataset

STUDYID	USUBJID	SUBJID	PARAM	PARAMCD	AVALl
BP3304-002	BP3304-002-A01	A01	Diastolic Blood Press	DIABP	80
BP3304-002	BP3304-002-A01	A01	Diastolic Blood Press	DIABP	78
BP3304-002	BP3304-002-A01	A01	Diastolic Blood Press	DIABP	76
BP3304-002	BP3304-002-A01	A01	Diastolic Blood Press	DIABP	79

That is a very [very] basic data structure. The problem is that we are not going to be able to use it to produce the analysis described in the Results Metadata. When do the values occur? Where is baseline? Which subjects are in the MITT population? There are many additional variables that need to be included to meet our reporting needs.

First, let’s take a look at timing. At a minimum, we need to include the AVISIT variable to define when each of these visits occurred. We know, from the SAP, that for each subject we will need a BASELINE value, a WEEK 4 value, and values for each of the other required timepoints. Obviously as you define the AVISIT variables, you are going to be able to identify the baseline visit and derive the change from baseline values. As you compute the change from baseline, that value can be stored in the CHG variable. You can also store the baseline value itself in the BASE variable. This allows the calculation of CHG to be simple and easier for a reviewer later.This is allowed, as CHG describes the actual results stored in AVAL.

Now that you have the visit information and the change from baseline values, you need to add flags to identify the analysis population. As described earlier, that information is already available in the ADSL dataset, so we need to merge to get the MITTFL flag into this dataset. You are also going to need to identify which treatment group they are assigned to.

For that, you will need to populate the TRTPN and TRTP variables. While that data is stored in the ADSL dataset, there you will find the variables TRT01PN and TRT01P. The ADSL dataset is structured so that subjects can be tracked across different treatment groups for the duration of the study. For the individual analysis datasets, however, it is expected that each record represents a single point in time and therefore should only be assigned to one specific treatment. Since our study is simple, with only one treatment per subject, we can assign TRT01PN to TRTPN and TRT01P to TRTP. For more complex studies, you would need to compare the visit date from the VS domain to the treatment period start and end dates (TRxxSDT and TRxxEDT) to determine which TRxxP and TRxxPN values would need to be used.

ADVS Variable Metadata

DS Name	Param Id	Var Id	Var Label	Var Type	Var Format	Codelist	Source / Derivation
ADVS	*ALL*	STUDYID	Study Identifier	Text	$15		ADSL.STUDYID
ADVS	*ALL*	USUBJID	Unique Subject ID	Text	$20		ADSL.USUBJID
ADVS	*ALL*	SUBJID	Study Level Subject ID	Text	$5		ADSL.SUBJID
ADVS	*ALL*	TRTP	Planned Treatment	Text	$10		ADSL.TRT01P
ADVS	*ALL*	TRTPN	Planned Treatment ID	Int	1.0		ADSL.TRT01PN
ADVS	*ALL*	MITTFL	Modified ITT Flag	Text	$1	Y=Yes, N=No	ADSL.MITTFL
ADVS	*ALL*	AVISIT	Visit Identifier	Text	$10	Baseline Week 4 Week 8 [more]	VS.VISIT
ADVS	*default*	PARAM	Parameter Identifier	Text	$20	Diasto..	VS.VSTEST
ADVS	paramcd	PARAMCD	Parameter ID code	Text	$8	DBP=Diastolic..	VS.TESTCD
ADVS	*ALL*	AVAL	Result	Int	8.4		VS.VSORRESU
ADVS	*ALL*	BASE	Baseline Value	Int	8.4		AVAL where AVISIT=BASELINE
ADVS	*ALL*	CHG	Change from Baseline	Int	8.4		AVAL - BASE
ADVS	*ALL*	DTYPE	Derivation Type	Text	$10	LOCF	Identifies if the AVAL is derived from prior visits.

Note: DS=Dataset, Param=Parameter, Var=Variable, Id=Identifier

You now have enough basic information to create the Analysis Variable Metadata. For each of the fields we have described above, you will need to describe the variable including type and variable label. You are also going to have to describe the source of each of these variables. When describing the source, you should either indicate the SDTM domain that the variable comes from, the

ADSL variable that it is used to populate the field, or the method for creating the variable from the SDTM data (or constant). An example of the ADVS Variable Metadata is included to the right. Also note that variable labels are still limited to a maximum of 40 characters.

WHAT ABOUT MISSING RESULTS OR EXTRA VISITS?

Sometimes subjects don't always show up when they are supposed to, or they don't show up at all. The analysis dataset should represent the data going into the analysis, even if it has to be computed. As you read in the SAP (which you should have read many times by this point), there is a specification that missing data should be imputed using the Last Observation Carried Forward method (LOCF). This methodology means that for each subject, the last recorded value is saved for the visit and a new record is created to represent the value at the end of the study . The derived records will only be usedfor analysis tables and not for listing individual visits. You will also need to let the reviewer of the analysis data know which data values were derived, and how the derivations were completed.

After you have assigned the AVAL and CHG values for both the original visit and the derived visit, there is still one more stop. You will need to add the DTYPE field to your ADVS dataset to identify which values are derived and what technique was used. For our purposes, you would assign LOCF to the DTYPE field. For other cases, you may also assign other values to DTYPE depending on the type of derivation performed. In your metadata, you will also need to describe, with as much detail as possible, how the derived value was created. For example, you may need to note that only post-baseline values should be carried forward.

There are other times when the subject shows up more often than is planned. Again, the SAP specifies that the record closest to the scheduled visit is the one that should be used for analysis. Since our ADVS dataset is going to be used for analysis, we will need to evaluate how to handle this data. One approach would be to only include those values to be used for reporting. However, since this data may also be used for generating listings, excluding data is not generally the best approach. Instead of excluding records, use the ANLxxFL flag (which we can call ANL01FL for our purposes) to identify which variables should be used for analysis. With this variable, we can flag the records actually used in our analysis and still allow all values to be stored in the analysis dataset. One thing to remember is that if you use the ANLxxFL in your reporting, be sure to include it in the selection criteria in your Analysis Results Metadata.

For some of our subjects, you will find that they have multiple baseline visits. In the SAP, we specifiy to use the average of all readings to create a baseline value. There isn't another variable to capture the derived baseline value and the BASE value recorded must show in in an individual record in the dataset. For that reason, we are going to need to create a new record in the ADVS dataset with a DTYPE of 'Average' and an AVISIT of BASELINE. This leads to the question of which AVISIT of BASELINE is the BASELINE record? This is solved by including the ABLFL value and flagging the record to be used as baseline with a 'Y'. Be aware that you only need to add a derived baseline record when baseline needs to be derived.

SOME OTHER CRITICAL FLAGS

While there are many flags you will use in creating ADaM datasets, here is one more you will use rather frequently. Take a look at table 14.3.6.2, the Clinically Significant Vital Signs report. Here you will notice that for some of the vital signs parameters we have calculated the frequency that they are above or below some specific criteria. For this report, the medical reviewers have established specific criteria (such as diastolic blood pressure >130 mmHg) for determining critical significance. For some other types of data, the protocol may require the investigator to identify what is critically significant on the CRF form, which would be reported in an SDTM domain.

In the ADaM data structures, we are provided a flag to identify AVAL results that are critically significant. To accomplish this, we actually need to use a combination of variables. The first variable, CRITy, defines the criteria being evaluated for each record. We also need to use the CRITyFL to identify whether or not the criteria specified in the CRITy variable has or has not been met. As you create these variables, you should check that the number specified in y matches between CRITy and CRITyFL.

In our example in table 14.3.6.2, we are actually going to create two criteria. Criteria 1 will be used to define when the AVAL exceeds the upper range and Criteria 2 will be used to identify when the AVAL is below the lower range for each type of vital sign. Following our examples, when the PARAMCD was DBP, we would set CRIT1 to 'Diastolic BP > 130' and then set CRIT1FL to either 'Y' or 'N' if that criteria was met. For the same record we would set CRIT2 to 'DIASTOLIC < 40' and then set CRIT2FL to 'Y' or 'N' if that value was met. If the PARAMCD was SBP, we would set CRIT1 to 'SYSTOLIC > 210' with CRIT1FL set to 'Y' or 'N' if that criteria was met and CRIT2 to 'SYSTOLIC < 80' and set CRIT2FL to 'Y' or 'N' for that criteria.

PARAMETER METADATA

Some of the variables being created are categorical. If you were the reviewer, wouldn't you want to know what values would be expected? For this reason, the Parameter Metadata is created to show what values are allowed for specific variables. Take a look at the AVISIT variable and you will notice the codelist is populated with the expected values (shortened to fit in this document.)

You will also notice that PARAM is one of the categorical fields with all the populations. The Parameter Identifier shows a value of 'DEFAULT' meaning that the fields in the PARAM are self-explanatory. The PARAMCD field, however, is actually a parameterized field. Each entry there represents a value in the PARAM field. For that reason, the PARAMCD value also shows up in the Parameter Identifier field. The codelist then needs to show that DBP is the equivalent of Diastolic Blood Pressure. In this manner, we create the Parameter Metadata within the Variable Metadata.

DEALING WITH DATA THAT IS [CENSORED]

If you look at the table shells, table 14.2.5.1 is the Time to Achievement of <90 mmHG of Diastolic BP. This is also a key analysis of this study. Both the table shells and the SAP specify that the Kaplan-Meier method is to be used for computing the confidence intervals. Because of this, we know we are going to need to use the LIFETEST procedure and customize the data to be used with that procedure.

The way we will do this is a two-step process. The end result will be a whole new dataset with just the data to be used in the LIFETEST procedure. We will call the new dataset ADVSLT. The creation of this will be based on the data within the ADVS dataset.

In the first step, we are going to add another analysis flag variable, as we have already described. This flag will be used to identify the first record for each subject with diastolic blood pressure less than 90 mmHg. For this example, we will create ANL02FL to flag the recorded value used. These records will be used as an input for creating the ADVSLT dataset.

After this flag has been created, the next step is to create the ADVSLT dataset. Starting with the ADVS dataset, we can then select all records where ANL02FL is set to 'Y'. We will then create the PARAM variable with a value of 'Diastolic BP Days till Threshold Val' and a PARAMCD with a value of 'DBPDUR'. The AVAL will be the number of days, which is the study day of the visit.

In addition to the AVAL for each DBPDUR record in ADVSLT, we need to identify if the duration recorded should be censored. For that, we will use the CNSR (censored) variable. If the subject has at least 1 record meeting the criteria, we can expect that the duration is an actual result and set the CNSR to 0. If the subject has no records below the threshold, we would set the CNSR value to 1. This will be used as the censoring value to the LIFETEST procedure.

HAVING FUN WITH LABS

While we have been focusing on creating the vital signs analysis, there is another findings type of data that we should focus on: LABS. In addition to the variables above, here are some that will help you will the lab analysis.

First, let's take a look at 14.3.5.1 which is the shift from baseline table. For this table, we need to know if each post baseline value is either higher or lower than the baseline value. This is often used in conjunction with the upper and lower values (i.e. if the baseline value was above normal but the current value is within normal ranges). We are probably going to be creating a dataset called ADLB to capture the results for each lab result at each expected visit. We will also use the LB domain, which should contain many of the data points we are going to use.

Within the ADLB dataset, we are going to use the STUDYID, USUBJID, and AVISIT much as we did for the ADVS dataset. For the PARAM value, we are going use the laboratory test name but we are going to include the units the result is reported in in the PARAM value. The AVAL will be the actual results and the CHG values will record the change from baseline. In addition to these values, we are going to use some other values to classify our AVAL. For each of the laboratory results, we are going to record the ANRLO and ANRHI variables to record the lower and upper normal range values. If this is the baseline record, we are going to use the BNRIND to classify the AVAL result as either above ANRHI (set BNRIND to 'H') or if the AVAL is below ANRLO (Set BNRIND to 'L'). The BNRIND result should be repeated for each record that the baseline result is compared to. If the AVAL is not associated with the baseline record, the ANRIND should be used and populated with 'H' or 'L' in the same was as the BNRIND variable. Now that we know if the baseline value is normal (as blank), above or below normal and the non-baseline value is normal, above, or below normal, we can also create one more variable. The SHIFT1 variable can be used to combine the BNRIND and ANRIND variables into a single result representing the shift in results.

SOME NOT SO ADAM DATASETS

So now we have just created the vital signs analysis dataset and you have probably realized how easy it is to create these datasets for purposes of analyzing findings types of data. The next question is how many analysis datasets will we need? Related findings should be grouped in the same dataset. In this case, we also need to summarize systolic blood pressure, so those values should also be added to ADVS.

But you are also probably asking yourself about how to handle data, such as adverse events, concomitant medications, and medical history. Are you going to have to create a parameter for each and every variable in the SDTM domains?

Breaking the events domains into one record per variable doesn't make sense, especially if the intent is to reconstruct it within the reporting program. The fact is that, for now, you can just pull the SDTM domains directly into the ADAM domains and add the necessary flags for analysis. The ADaM committee recently released draft specifications for ADAE, and is still working on further details for these domains.

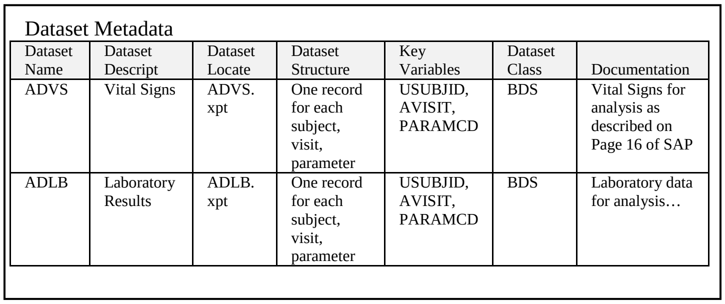
THE SIMPLE PART, THE DATASET METADATA

Now that you have the Analysis Results Metadata, the Variable Metadata, and the Parameter Metadata, the next step is to define the Dataset Metadata. Fortunately, we have saved the simplest metadata for the last.

This process simply describes all of the analysis datasets that are being used. There are just 7 fields that need to be defined for each dataset. The first three simply define the dataset itself with the name, the description of the database, and the file name.

After you have declared the dataset, you need to define more about the contents of the dataset. The dataset structure and key variables work together to let the reviewer know how your dataset should be sorted and analyzed. In our dataset, we created one record for each subject, for each visit, and for each parameter. Describing that, along with the variable names, is all that is necessary.

The next section just describes the type of dataset. In most cases you will define this as a BDS. The other types you might need to use would be ADSL, for the subject level data, and SDTM, which you will probably need when using adverse event, concomitant medications and medical history data. An example of this metadata is at the right.



Dataset Metadata

Dataset Name	Dataset Descript	Dataset Locate	Dataset Structure	Key Variables	Dataset Class	Documentation
ADVS	Vital Signs	ADVS. xpt	One record for each subject, visit, parameter	USUBJID, AVISIT, PARAMCD	BDS	Vital Signs for analysis as described on Page 16 of SAP
ADLB	Laboratory Results	ADLB. xpt	One record for each subject, visit, parameter	USUBJID, AVISIT, PARAMCD	BDS	Laboratory data for analysis…

SOME THINGS TO CONSIDER BEFORE YOU START USING THE METADATA

If you look back over the metadata you have just created, you probably find that everything you need to start coding has been documented. Still, there are some points that might be worth reviewing before you start.

1) Are all fields documented? Check the table shell and make sure everything has been captured and documented.

2) Do you have all the source data identified? Also, is it available? What about handling missing data or date imputation?

3) Are all of the statistical calculations clear? If you are computing inferential statistics, is the model clear on how to compute results? What about censoring rules?

The metadata that you are creating should be written as a standalone document. Consider that the programmer who actually codes the dataset and TLF programs for this output may only have the metadata and SAP for creating it.

CONCLUSION

At this point, you should be talking metadata this, metadata that, metadata, metadata, and more metadata.

What you should be seeing is that the creation of the metadata reflects the standard process that most of us have already been following when creating analysis datasets to be used for generating tables, listings, and figures. Basically, you start by analyzing what data you need to create the reports and follow that by describing where that data is coming from. Once you have those pieces, you add details such as what codes are available for specific results. You then handle unique data conditions, such as censoring.

All that should be happening now is a more structured approach to creating the analysis data. The best way to do this is to jump in and work through it. After a few times, you will find others asking for your opinions on how to create the ADaM datasets for a project.

REFERENCES

ADaM version 2.1 and the ADaM IG 1.0 can be downloaded from http://www.cdisc.org.

CONTACT INFORMATION

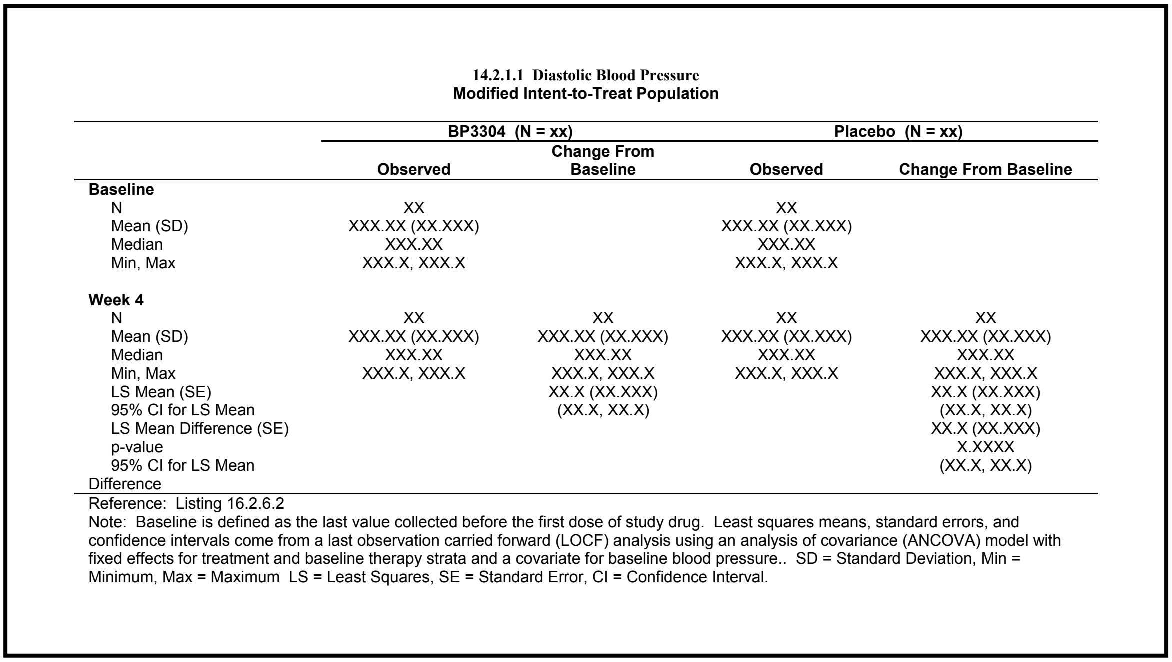
Your comments and questions are valued and encouraged. Contact the authors at:

Nancy Brucken
i3 Statprobe
5430 Data Court, Suite 200
Ann Arbor, MI 48108
734-757-9045
Nancy.Brucken@i3statprobe.com

Paul Slagle
United BioSource Corp
2200 Commonwealth Dr, Suite 100
Ann Arbor, MI 48105
734-994-8940 x1604
Paul.Slagle@unitedbiosource.com

Appendix A

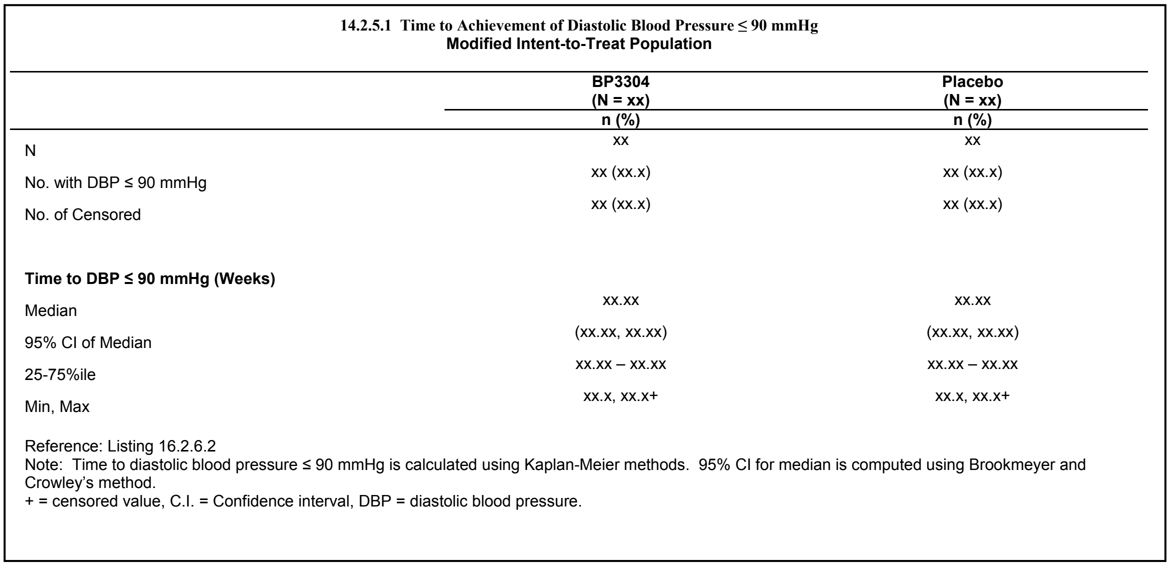
Table shells used in Hands On Workshop



14.2.1.1 Diastolic Blood Pressure
Modified Intent-to-Treat Population

	BP3304 (N = xx)		Placebo (N = xx)	
	Observed	Change From Baseline	Observed	Change From Baseline
Baseline				
N	XX		XX	
Mean (SD)	XXX.XX (XX.XXX)		XXX.XX (XX.XXX)	
Median	XXX.XX		XXX.XX	
Min, Max	XXX.X, XXX.X		XXX.X, XXX.X	
Week 4				
N	XX	XX	XX	XX
Mean (SD)	XXX.XX (XX.XXX)	XXX.XX (XX.XXX)	XXX.XX (XX.XXX)	XXX.XX (XX.XXX)
Median	XXX.XX	XXX.XX	XXX.XX	XXX.XX
Min, Max	XXX.X, XXX.X	XXX.X, XXX.X	XXX.X, XXX.X	XXX.X, XXX.X
LS Mean (SE)		XX.X (XX.XXX)		XX.X (XX.XXX)
95% CI for LS Mean		(XX.X, XX.X)		(XX.X, XX.X)
LS Mean Difference (SE)				XX.X (XX.XXX)
p-value				X.XXXX
95% CI for LS Mean Difference				(XX.X, XX.X)

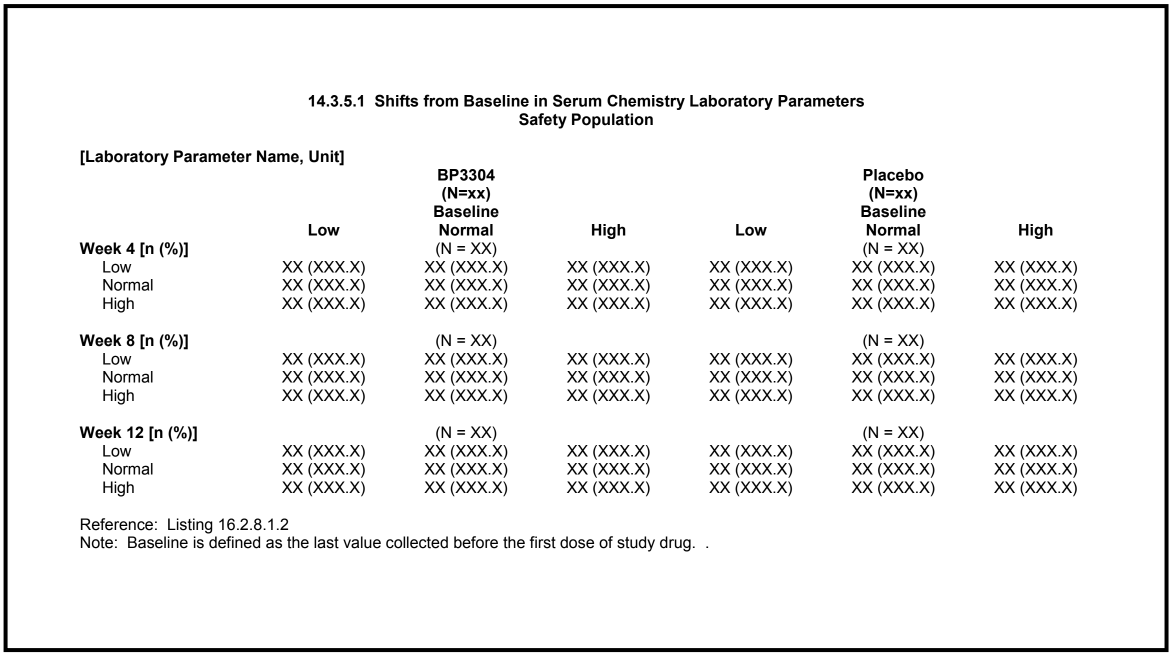
Reference: Listing 16.2.6.2
Note: Baseline is defined as the last value collected before the first dose of study drug. Least squares means, standard errors, and confidence intervals come from a last observation carried forward (LOCF) analysis using an analysis of covariance (ANCOVA) model with fixed effects for treatment and baseline therapy strata and a covariate for baseline blood pressure.. SD = Standard Deviation, Min = Minimum, Max = Maximum LS = Least Squares, SE = Standard Error, CI = Confidence Interval.



14.2.5.1 Time to Achievement of Diastolic Blood Pressure ≤ 90 mmHg
Modified Intent-to-Treat Population

	BP3304 (N = xx)	Placebo (N = xx)
	n (%)	n (%)
N	xx	xx
No. with DBP ≤ 90 mmHg	xx (xx.x)	xx (xx.x)
No. of Censored	xx (xx.x)	xx (xx.x)
Time to DBP ≤ 90 mmHg (Weeks)		
Median	xx.xx	xx.xx
95% CI of Median	(xx.xx, xx.xx)	(xx.xx, xx.xx)
25-75%ile	xx.xx – xx.xx	xx.xx – xx.xx
Min, Max	xx.x, xx.x+	xx.x, xx.x+

Reference: Listing 16.2.6.2
Note: Time to diastolic blood pressure ≤ 90 mmHg is calculated using Kaplan-Meier methods. 95% CI for median is computed using Brookmeyer and Crowley's method.
+ = censored value, C.I. = Confidence interval, DBP = diastolic blood pressure.

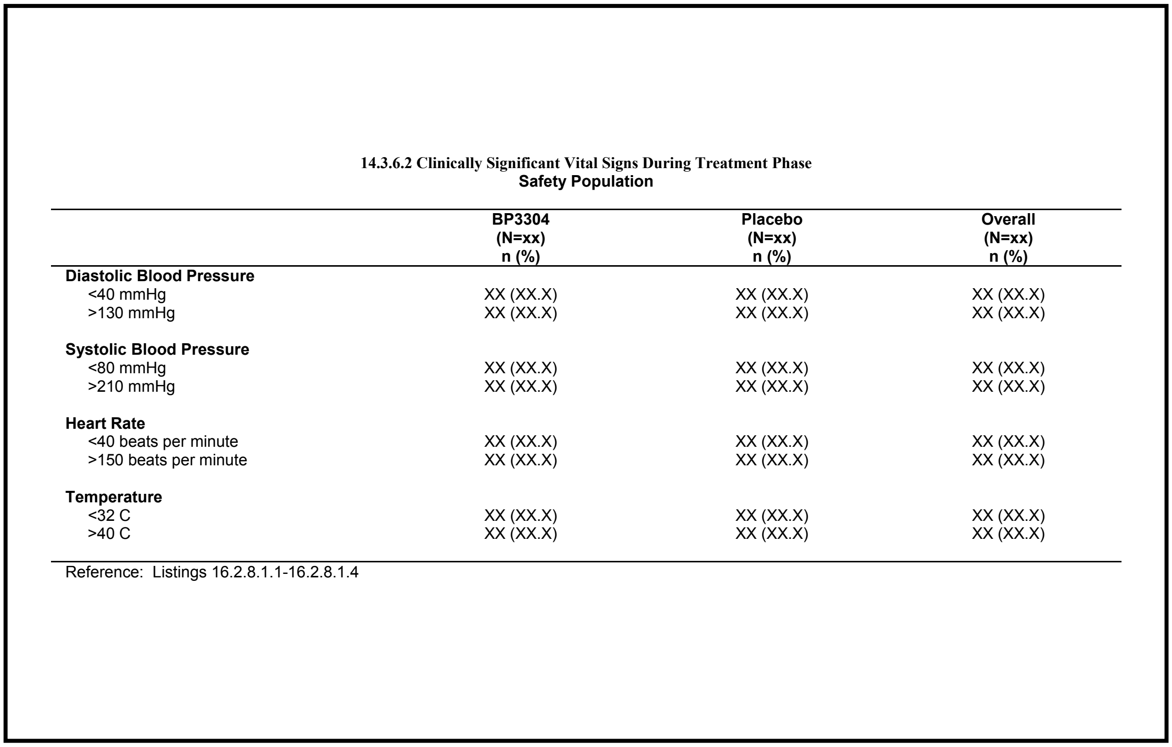


14.3.5.1 Shifts from Baseline in Serum Chemistry Laboratory Parameters
Safety Population

[Laboratory Parameter Name, Unit]

		BP3304			**Placebo**	
		(N=xx)			**(N=xx)**	
		Baseline			**Baseline**	
	Low	**Normal**	**High**	**Low**	**Normal**	**High**
Week 4 [n (%)]		(N = XX)			(N = XX)	
Low	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)
Normal	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)
High	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)
Week 8 [n (%)]		(N = XX)			(N = XX)	
Low	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)
Normal	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)
High	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)
Week 12 [n (%)]		(N = XX)			(N = XX)	
Low	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)
Normal	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)
High	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)	XX (XXX.X)

Reference: Listing 16.2.8.1.2
Note: Baseline is defined as the last value collected before the first dose of study drug. .



14.3.6.2 Clinically Significant Vital Signs During Treatment Phase
Safety Population

	BP3304 **(N=xx)** **n (%)**	**Placebo** **(N=xx)** **n (%)**	**Overall** **(N=xx)** **n (%)**
Diastolic Blood Pressure			
<40 mmHg	XX (XX.X)	XX (XX.X)	XX (XX.X)
>130 mmHg	XX (XX.X)	XX (XX.X)	XX (XX.X)
Systolic Blood Pressure			
<80 mmHg	XX (XX.X)	XX (XX.X)	XX (XX.X)
>210 mmHg	XX (XX.X)	XX (XX.X)	XX (XX.X)
Heart Rate			
<40 beats per minute	XX (XX.X)	XX (XX.X)	XX (XX.X)
>150 beats per minute	XX (XX.X)	XX (XX.X)	XX (XX.X)
Temperature			
<32 C	XX (XX.X)	XX (XX.X)	XX (XX.X)
>40 C	XX (XX.X)	XX (XX.X)	XX (XX.X)

Reference: Listings 16.2.8.1.1-16.2.8.1.4