Paper 6-2010

Comparing JMP and SAS® for Validating Clinical Trials

Sandra D. Schlotzhauer, Chapel Hill, NC

Abstract

When validating clinical trial analyses, an independent programmer typically confirms the results. Most companies use SAS as the standard software for performing analyses and generating results to submit to regulatory agencies. Validation requires replicating the content but not the appearance of the results. For example, the validator confirms a *p*-value, but does not format the results to match the appearance in a table submitted to the FDA. This paper discusses experiences in using both SAS and JMP for validation. The paper discusses strengths of JMP and SAS.

Introduction

The phrase "validating clinical trials" encompasses many tasks, including those shown in the list below:

- Ensuring that the variables used to measure efficacy are appropriate for the disease or condition
- Designing the trial to comply with regulatory guidelines (e.g., Good Clinical Practices, or GCP)
- Sizing the trial to allow for appropriate power to detect a significant difference
- Ensuring that all variables identified in the protocol are included in the Case Report Forms (CRFs), and subsequently that all CRF variables are included in a data set
- Defining variable names to meet appropriate standards (e.g. CDISC)
- Developing edit-checks to find potential errors in data sets for the trial
- Writing SAS programs to create data sets for analysis, and to perform analyses that generate Tables, Listings, and Figures (TLFs)
- Ensuring that all analyses identified in the Statistical Analysis Plan (SAP) have been performed
- Verifying titles, footnotes, and other appearance specifications from the mock TLFs
- Performing independent programming to validate breaking of the blind, especially for crossover trials
- Verifying all derived variables and handling of missing data
- Performing independent programming to validate the TLFs.

In addition, each organization typically has standard operating procedures (SOPs), work practice guidelines (WPs), and SAS programming guidelines. Validation can include ensuring that activities conform to these guidelines as well as regulatory guidelines, such as ICH E9. Many of these validation activities are process-oriented rather than software oriented. An excellent reference that combines discussion of process activities and SAS for many of the items above is *Validating Clinical Trial Data Reporting with SAS®*. This paper discusses process activities experienced when using both JMP and SAS for validation, and focuses on the software activities for the last two items in the above list.

This paper starts with a discussion of planning. Planning is an essential part of validation work, regardless of the software you use. However, when you know that you might use a combination of software packages, the planning activity becomes even more important. Key steps in planning include reviewing the SAP and mock TLFs and assessing software applicability, creating a validation plan that outlines activities, and building file structures in advance.

After planning is complete, validation activities typically occur in phases. Phase 1 involves validation of blinded data, where the actual treatments for each patient are unknown. Instead, a dummy randomization assigns each patient to a mock treatment. Phase 1 analysis and validation typically uncovers many of the potential issues with data algorithms or analysis. Since no one involved knows the actual treatment for a patient, the team makes objective decisions and is protected from making decisions that might be perceived to benefit one treatment over another. At this stage, you want team agreement on how to handle any data issues uncovered during the actual conduct of the trial, such as missing values, improper inclusion in the study, mistakes in execution, and so on. Phase 2 involves Draft validation of unblinded data, where the actual treatments for each patient are known. In a perfect world, Phase 3 of running the Final tables would simply be a re-run of the Draft tables. However, in the real world, teams typically revise the Final TLFs. Sometimes these revisions involve only cosmetic changes, but the revisions can also involve changes in analyses or additional analyses, such as sensitivity work. Some cosmetic changes might require re-running analyses.

For example, suppose the team decides to increase the number of decimal places for descriptive statistics to four from three. If the validation program used previously shows statistics to only three decimal places, this cosmetic change will require re-programming and revalidation activities. However, if the validation program shows statistics to four (or more) decimal places, this cosmetic change will not require re-running any programs. Instead, it will require only rechecking the new table using the previously-generated validation results.

For each phase, validation follows a logical pattern. First, validate the data according to plans. Next, validate the tables and listings. Finally, validate the figures, which typically depend on the tables, listings, or supplemental analyses. This paper follows the same logical pattern. After discussing planning, the paper discusses the strengths of JMP and SAS for validating data, then tables and listings, and finally figures. The paper focuses on JMP, not on the recently released JMP Clinical software. This paper also assumes that validation involves independent programming and duplication of results. In some cases, validation involves peer review of the programs used to create the results—discussing processes and best practices for peer review is another paper.

The paper ends with a summary of strengths for JMP and SAS, and also compares some practical aspects of using each software package as an independent consultant.

Reviewing the SAP and Mock TLFs

The SAP explains the analyses for the trial. This document provides very detailed descriptions of what analyses are performed, on what data, and under what conditions. The document also describes the populations for the analyses, such as “safety” or “intent-to-treat” (ITT), and defines the criteria for inclusion in the populations. When reviewing the SAP, assess software applicability for each validation activity. Reviewing the SAP will help you build the Validation Plan.

First, consider the data that will require validation. Are there calculated variables? Is missing data handled using Last-Observation-Carried-Forward (LOCF) or other approaches? How many populations are there, and how are these populations defined? When creating your Validation Plan, be sure to include discussion of how the data will be validated.

Second, review the analyses and the populations for those analyses. Both JMP and SAS can perform many common analyses for trials, including ANOVA, ANCOVA, logistic regression, Chi-square, survival analyses, and more. However, SAS does provide many analyses that JMP does not. SAS also provides analyses in situations where JMP provides an analysis in a subset of situations. Suppose the SAP proscribes the Fisher’s Exact Test. Both JMP and SAS can run this test for a 2x2 table. However, SAS can also run the test for larger tables. Before deciding which software package is applicable for an analysis, you must also consider the population.

Third, review the mock TLFs and assess the ability to reproduce content. This is usually best performed along with the review of analyses in the SAP. Focus on the descriptive and analytical statistics—many tables contain both. In your review, do not worry about the appearance of the table or listing. When validating results, your focus is on content and not on cosmetics.

Creating the Validation Plan

The Validation Plan is a document that outlines validation activities for a clinical trial. Key inputs include the protocol, annotated Case Report Forms (CRFs), SAP, and mock TLFs. Figure 1 shows the table of contents for a sample Validation Plan.

Most of the topics discussed in a Validation Plan do not depend on the software used for validation activities. For example, the “General Methodology” section typically describes the validation approach for Blinded, Draft, and Final phases, and defines when confirmatory reviews are appropriate.

When you expect to use a combination of JMP and SAS for validation, the “Outputs” section can include text that minimizes the need for revisions to the plan. For example, include the following text:

> The software used for a given activity may be defined as JMP and switched to SAS, or defined as SAS and switched to JMP, without requiring an update to this Validation Plan. The Validation Summary documentation will identify the software used for each activity.

Since obtaining all required signatures for the Validation Plan can be a lengthy process (depending on client SOPs), the text above allows you to change the software used without repeating the signature process.

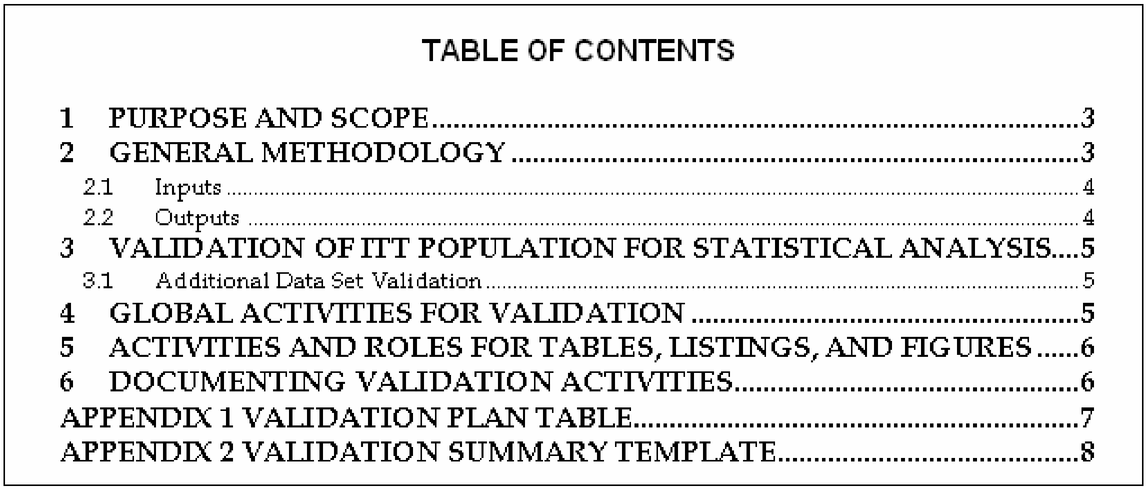


TABLE OF CONTENTS

1 PURPOSE AND SCOPE 3
2 GENERAL METHODOLOGY 3
2.1 Inputs 4
2.2 Outputs 4
3 VALIDATION OF ITT POPULATION FOR STATISTICAL ANALYSIS....5
3.1 Additional Data Set Validation 5
4 GLOBAL ACTIVITIES FOR VALIDATION 5
5 ACTIVITIES AND ROLES FOR TABLES, LISTINGS, AND FIGURES 6
6 DOCUMENTING VALIDATION ACTIVITIES...6
APPENDIX 1 VALIDATION PLAN TABLE..7
APPENDIX 2 VALIDATION SUMMARY TEMPLATE..8

Figure 1. Sample Table of Contents for a Validation Plan

The Validation Plan Table details the validation activities for each data set, table, listing, and figure. Typically, each item forms a row in the table. Typical columns include the

- TLF number
- TLF title
- validation activities
- validator.

The TLF number and title are important because table numbers sometimes change between the plans for a trial and the final clinical study report (CSR). By providing both the number and title, you will be able to trace validation activities throughout the trial.

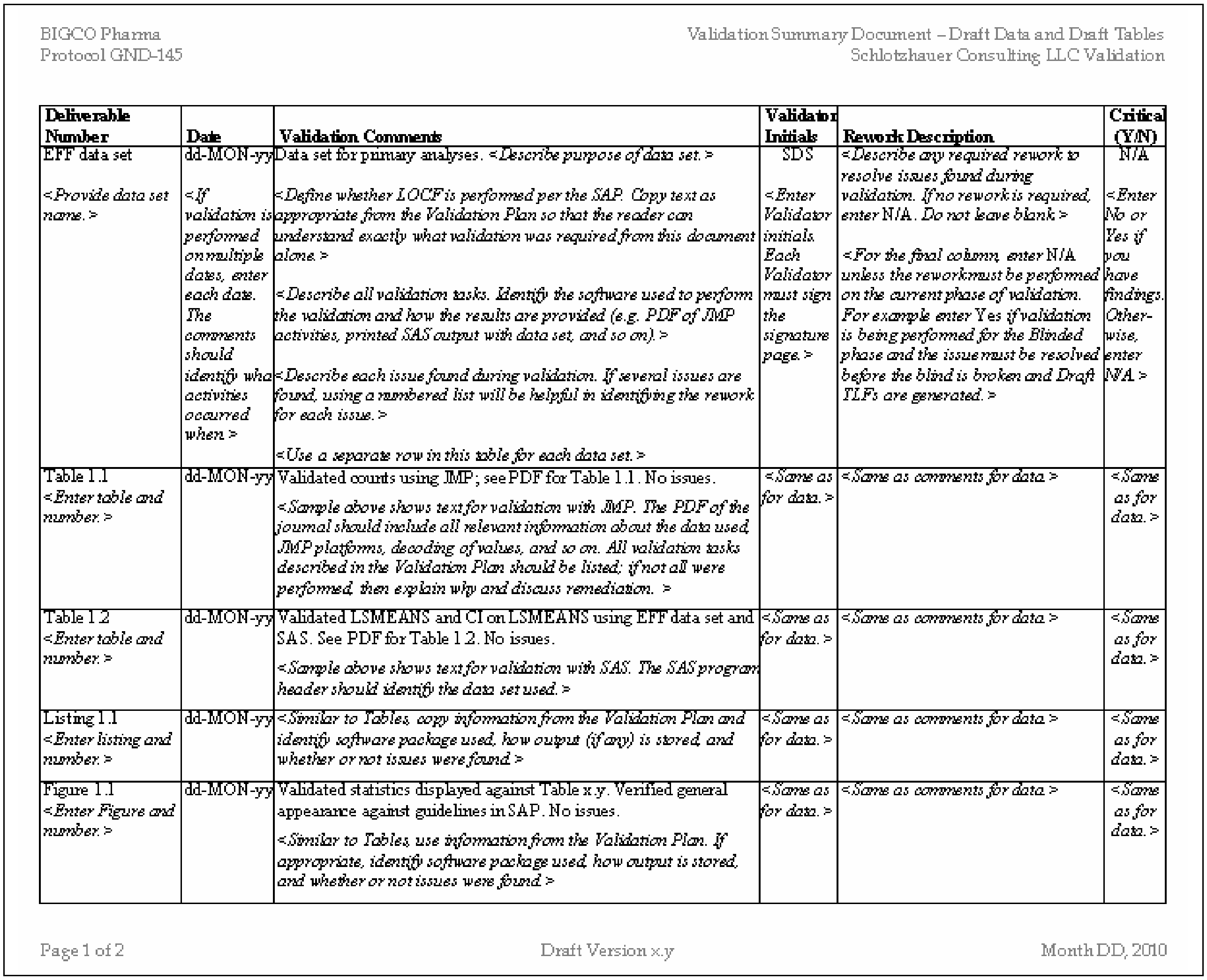
These validation activities should not reproduce the general activities described in the body of the validation plan. The list below shows sample validation activities.

- Verify the counts and frequencies in the table using JMP. Provide a PDF of the JMP results or a JMP journal as documentation.
- Verify the Fisher's Exact Test for the 2xn table using SAS. Provide a PDF of the SAS output as documentation.
- Verify the *p*-values for the ANOVA using JMP and provide a PDF of the JMP results or JMP journal. Confirm that the model used in the analysis is as described in the SAP.
- Verify the *p*-values from the statistical tests using either JMP or SAS, and provide appropriate PDFs to document the results.
- Compare a 10% random sample from each treatment group to the raw data listing.
- Verify the values plotted in the figure against the supporting table (Table x.y).

The items in the list identify the software used, how the results will be documented, and how validation outputs might differ for different software. The fourth item shows an example of describing a validation activity when you do not know in advance whether you will use JMP or SAS for validation. By allowing this flexibility in software choices, you can avoid revisions later. Typically, a Validation Plan is not revised after the blind is broken. In those situations, the Validation Summary must describe any revisions to the planned validation and the reasons for those revisions.

Since most trials involve multiple people performing validation, identify the validator. In some cases, the validators will be from different companies. For example, my experience includes situations where a CRO performs validation on all safety data sets and TLFs, and external statistical consultants perform validation on ITT data sets and TLFs. By clearly identifying validators in the Validation Plan Table, you ensure that all data sets and TLFs are validated.

Figure 2 provides an example of a sample Validation Summary.



BIGCO Pharma
Protocol GND-145

Validation Summary Document – Draft Data and Draft Tables
Schlotzhauer Consulting LLC Validation

Deliverable Number	**Date**	**Validation Comments**	**Validator Initials**	**Rework Description**	**Critical (Y/N)**
EFF data set *<Provide data set name. >*	dd-MON-yy *<If validation is performed on multiple dates, enter each date. The comments should identify wha activities occurred when >*	Data set for primary analyses. *<Describe purpose of data set. >* *<Define whether LOCF is performed per the SAP. Copy text as appropriate from the Validation Plan so that the reader can understand exactly what validation was required from this document alone. >* *<Describe all validation tasks. Identify the software used to perform the validation and how the results are provided (e.g. PDF of JMP activities, printed SAS output with data set, and so on). >* *<Describe each issue found during validation. If several issues are found, using a numbered list will be helpful in identifying the rework for each issue. >* *<Use a separate row in this table for each data set. >*	SDS *<Enter Validator initials. Each Validator must sign the signature page. >*	*<Describe any required rework to resolve issues found during validation. If no rework is required, enter N/A. Do not leave blank. >* *<For the final column, enter N/A unless the rework must be performed on the current phase of validation. For example enter Yes if validation is being performed for the Blinded phase and the issue must be resolved before the blind is broken and Draft TLFs are generated. >*	N/A *<Enter No or Yes if you have findings. Other-wise, enter NA >*
Table 1.1 *<Enter table and number. >*	dd-MON-yy	Validated counts using JMP; see PDF for Table 1.1. No issues. *<Sample above shows text for validation with JMP. The PDF of the journal should include all relevant information about the data used, JMP platforms, decoding of values, and so on. All validation tasks described in the Validation Plan should be listed; if not all were performed, then explain why and discuss remediation. >*	*<Same as for data. >*	*<Same as comments for data >*	*<Same as for data. >*
Table 1.2 *<Enter table and number. >*	dd-MON-yy	Validated LSMEANS and CI on LSMEANS using EFF data set and SAS. See PDF for Table 1.2. No issues. *<Sample above shows text for validation with SAS. The SAS program header should identify the data set used. >*	*<Same as for data. >*	*<Same as comments for data >*	*<Same as for data. >*
Listing 1.1 *<Enter listing and number. >*	dd-MON-yy	*<Similar to Tables, copy information from the Validation Plan and identify software package used, how output (if any) is stored, and whether or not issues were found >*	*<Same as for data. >*	*<Same as comments for data >*	*<Same as for data. >*
Figure 1.1 *<Enter Figure and number. >*	dd-MON-yy	Validated statistics displayed against Table x.y. Verified general appearance against guidelines in SAP. No issues. *<Similar to Tables, use information from the Validation Plan. If appropriate, identify software package used, how output is stored, and whether or not issues were found >*	*<Same as for data. >*	*<Same as comments for data >*	*<Same as for data. >*

Page 1 of 2 Draft Version x.y Month DD, 2010

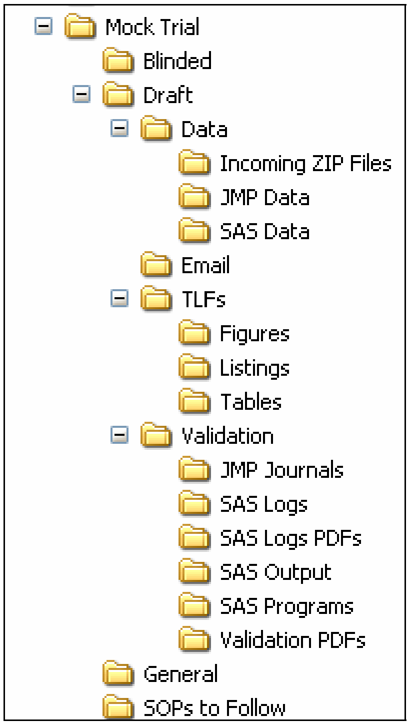
Figure 2. Sample Validation Summary

The sample text in Figure 2 shows one approach to documenting which software package is used, and how the validation results are provided. The row for Table 1.1 describes validation using JMP, and the row for Table 1.2 describes validation using SAS.

Building File Structures in Advance

A final aspect of planning is to build your file structures in advance. This best practice creates an organizational structure of folders that will allow for validation in JMP and in SAS.

Figure 3 shows a sample file structure that is designed for both SAS and JMP to be used in validation. The Draft phase of validation is expanded to show the directory tree. The Blinded folder would contain a similar directory tree, as would the Final folder when it was created. The General folder contains important documents such as the SAP, Validation Plan, CRFs, protocol, and so on. The SOPs folder is important, as you might later need to prove that you followed relevant SOPs in performing your work. With multiple clients, disk space is cheap. Copy the appropriate SOPs for each trial you validate, and you will avoid time spent searching for the relevant SOPs in the future.



- Mock Trial
  - Blinded
  - Draft
    - Data
      - Incoming ZIP Files
      - JMP Data
      - SAS Data
    - Email
    - TLFs
      - Figures
      - Listings
      - Tables
    - Validation
      - JMP Journals
      - SAS Logs
      - SAS Logs PDFs
      - SAS Output
      - SAS Programs
      - Validation PDFs
  - General
  - SOPs to Follow

Figure 3. Sample File Structure

The sample structure allows for incoming data in Zip files. If you receive data directly as SAS data sets or transport files, then you might want to rename folders to fit your needs. Given industry standards, you are unlikely to receive JMP data tables. However, in validation activities, you will almost certainly create JMP data tables. By separating the SAS and JMP data into separate folders, you can separate incoming data and data that you create. If you also create your own SAS data sets, perhaps for troubleshooting problems, you will want to create a separate folder for those data sets also.

The sample structure allows for validation activities using both JMP and SAS. The “Validation PDFs” folder contains the documentation for validation of each data set or TLF. This folder should include all PDFs that are referenced in the Validation Summary Table, and also include the Validation Summary Table document. When naming these PDFs, develop a naming convention that allows for both a JMP PDF and a SAS PDF to support a single table. For example, you might base the naming convention on the assumption that most validation is done with JMP. Your validation files would be of the form “Table x_y.PDF” for JMP validation, and of the form “Table x_y_SAS.PDF” for SAS validation on a table. Place all the raw JMP journals, which require JMP to open, in a folder. Place all the SAS programs in a folder. Similarly, place the raw SAS output and raw SAS logs in folders. For easier auditing, create PDFs of the SAS logs, and place those in another folder.

This sample structure is a starting point. As you validate trials, you might need to add or delete folders to meet your needs. However, starting from a sample structure helps you organize information across trials for a given client, and across clients. By storing all the client delivered information in the “Validation PDFs” folder, you can respond to follow-up requests or audits more quickly in the future.

Validating Data

This paper discusses validating data in the contexts of confirming the accuracy of algorithms for derived variables, and for handling missing data. Programmers create derived variables for many reasons. Some derived variables are simply flag variables that conveniently define whether a patient meets a set of criteria or not. In other cases, derived variables define the change in an efficacy measurement over time. Confirming this second type of derived variable can be more complex when patients have missing data. The SAP defines how to handle missing data. Usually, the process involves some form of carrying forward the last nonmissing information. However, this definition can become complex, and involve multiple decision points. For example, how do you handle a patient who participated in the screening portion of the trial but never actually received treatment? Do you map their initial efficacy measurements (before treatment) to all the post-treatment efficacy timepoints in the trial? Or, do you define the post-treatment measurements as missing? Sometimes, the SAP defines this process; other times, the clinical team makes the decisions on blinded data. If the team makes decisions on blinded data, the Validation Summary should include a discussion of these decisions. In addition, depending on client SOPs, you might need to store additional detail, such as emails or memos-to-file.

The Validation Plan should define whether validation is performed across the data set or on a subset. To clarify, suppose you anticipate some data will be missing. Does the plan define that validation will involve confirming the accuracy for all missing data? Or, does the plan specify selecting a random sample (of 10%, for example) and confirming the accuracy of that sample? In either case, allow for practical decision-making after the Validation Plan is complete. For example, suppose the plan defines a 10% stratified random sample by treatment to confirm LOCF. After the trial is complete, you find that very few patients have missing data—in fact, less than 10%. Instead of simply following the Validation Plan, the best practice is to make a practical decision and perform validation on all the data. The Validation Summary should describe this change from planned validation, and describe the reasons behind the decision for the change. Depending on client SOPs, you might also need additional approval of any changes.

Similarly, the Validation Plan should identify when validation of data sets is not needed. Generally, data is validated at the Blinded and Draft phases. Most of the time, there are very few, if any, data changes between Draft and Final phases. In those situations, you might choose to confirm that the planned changes were made, and choose not to re-validate other data elements. This choice depends on the complexity of the changes. If the data changes are complex enough that they might have impacted other observations, then you are more likely to want to re-validate.

The next two topics discuss validating with JMP and SAS. The JMP topic is more extensive, given the audience for this paper. Considerations in validating data with JMP apply to SAS as well, so the SAS topic focuses on additional aspects of data validation when using SAS. In addition, these two topics focus on validating efficacy data sets. Validating safety data requires the same principles for derived variables.

Validating with JMP

JMP is designed for data exploration using a point-and-click interface. The result of this strength is that you can validate data using existing platforms, data management choices, and by creating formulas. Create a JMP journal to document your activities. In the journal, use bulleted lists or notes to document activities, and the reasons behind those activities. The journal should be complete enough that the reader does not require access to either the data or to JMP to confirm accuracy of your validation.

Consider an example from a cholesterol trial. The SAP defines the ITT population as patients with a baseline measurement and at least one post-baseline measurement. Figure 4 shows an example of the JMP journal that verifies the assignment of patients to the ITT population.

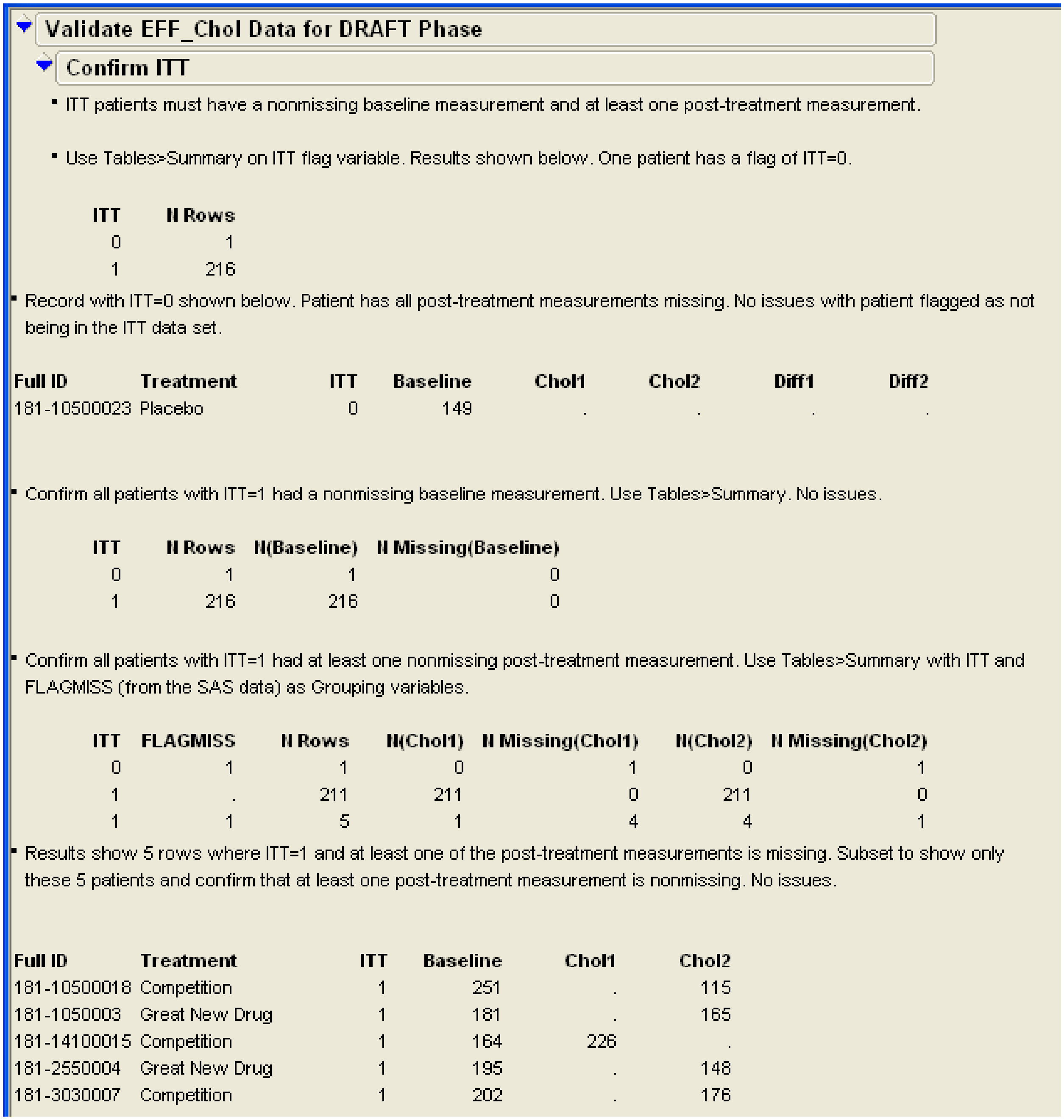
Figure 4 shows the JMP journal, for comparison with later figures that show the PDF files created from journals. This validation uses a combination of **Tables→Summary** and **Subset** to validate the ITT. The journal identifies the data set, phase of validation, and purpose for this journal. Using text items (the bulleted list items), the journal defines the ITT population. Then, the journal shows JMP activities to confirm that patients who are not included in the ITT population are handled correctly, and shows selected variables for the one record with ITT=0. In general, data sets contain many more variables than you need to show in the journal. The second part of the journal confirms that patients who are included in the ITT population meet the criteria.

Continuing the example from the cholesterol trial, suppose the SAP defines that missing efficacy measurements are handled by carrying the last non-missing measurement forward. This is generally called “last-observation-carried-forward,” and abbreviated as LOCF. The SAP further defines that pre-treatment measurements are not carried forward as post-treatment measurements. Figure 5 shows the journal to validate LOCF. The first part of the journal shows validation that can be easily checked visually. With only five patients where LOCF is used, building your own flag variables to confirm LOCF might not be valuable. However, you won’t know in advance how many patients need to be checked. The second part of the journal in Figure 5 shows the results of building flag variables to check. This validation has two steps. First, create a variable that validates the LOCF. Second, use **Tables→Summary** to list the values of the new variable.

Figure 5 also illustrates one issue with JMP when using formulas. The Formula editor is very powerful; however, pasting formulas into a journal does not usually provide an easily readable version of the formula. Instead, copy the formula window to a document, and save the document as a PDF. In general, you can combine all the formulas into a “Process Notes” document. Figure 6 shows the formula used to perform the validation in Figure 5.

Finally, Figure 5 illustrates a process approach. When you create new variables in JMP, use either a prefix or a suffix. This process helps you easily find the variables that you create, and separate those variables from the SAS data set variables.

When you create formulas for data validation, they are saved with the column in the data table. If you create a formula for the Blinded phase, you can simply copy that formula to the same-named variable for the data table in the Draft phase. If you find that you use the same formula across multiple trials, you can create a template. This is simply a data set that contains all the columns needed and has no rows. When you get the new data for a trial, you can create all the new columns, and apply the formulas, by concatenating the template to the data table.



Validate EFF_Chol Data for DRAFT Phase

Confirm ITT

- ITT patients must have a nonmissing baseline measurement and at least one post-treatment measurement.

- Use Tables>Summary on ITT flag variable. Results shown below. One patient has a flag of ITT=0.

ITT	N Rows
0	1
1	216

- Record with ITT=0 shown below. Patient has all post-treatment measurements missing. No issues with patient flagged as not being in the ITT data set.

Full ID	Treatment	ITT	Baseline	Chol1	Chol2	Diff1	Diff2
181-10500023	Placebo	0	149	.	.	.	.

- Confirm all patients with ITT=1 had a nonmissing baseline measurement. Use Tables>Summary. No issues.

ITT	N Rows	N(Baseline)	N Missing(Baseline)
0	1	1	0
1	216	216	0

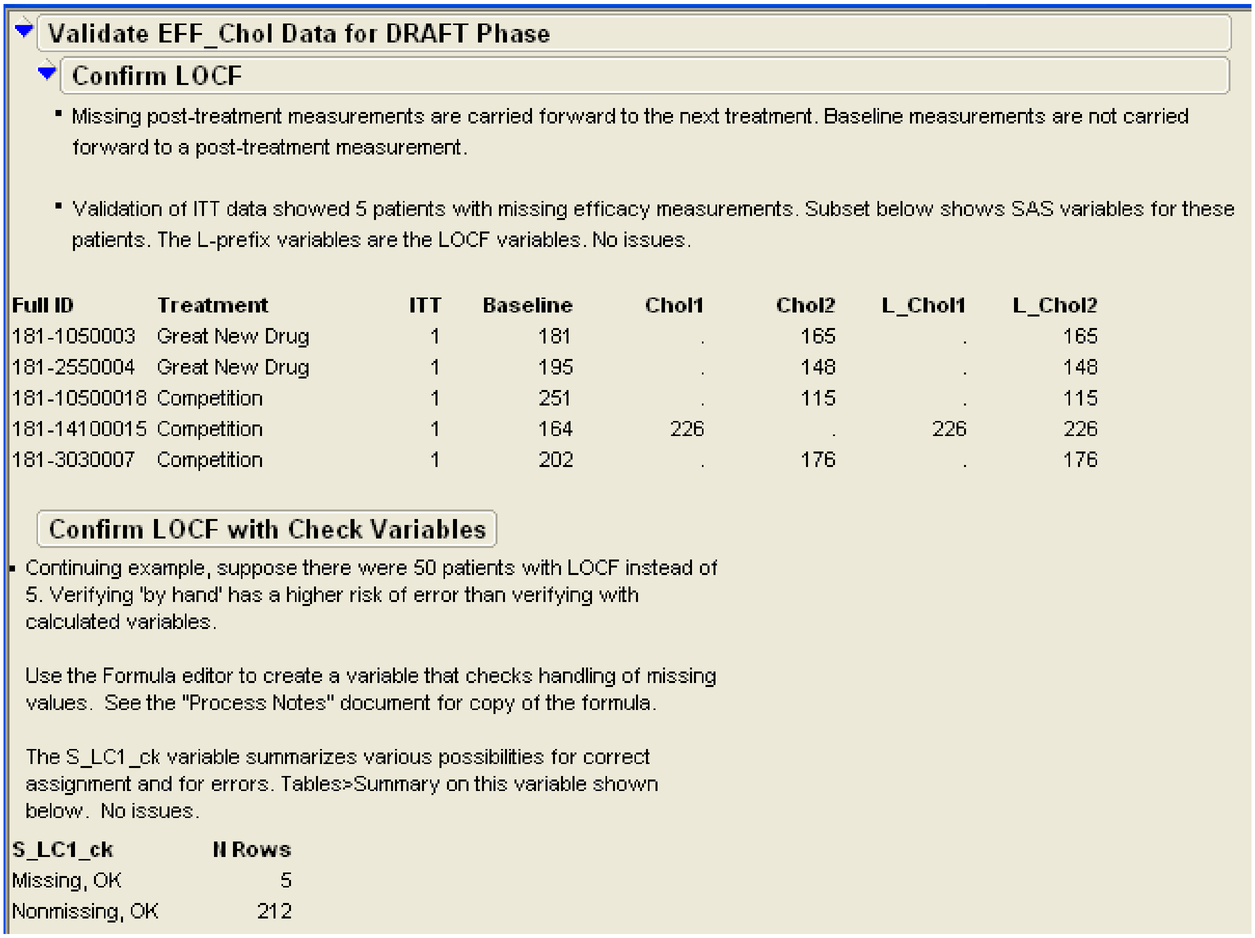
- Confirm all patients with ITT=1 had at least one nonmissing post-treatment measurement. Use Tables>Summary with ITT and FLAGMISS (from the SAS data) as Grouping variables.

ITT	FLAGMISS	N Rows	N(Chol1)	N Missing(Chol1)	N(Chol2)	N Missing(Chol2)
0	1	1	0	1	0	1
1	.	211	211	0	211	0
1	1	5	1	4	4	1

- Results show 5 rows where ITT=1 and at least one of the post-treatment measurements is missing. Subset to show only these 5 patients and confirm that at least one post-treatment measurement is nonmissing. No issues.

Full ID	Treatment	ITT	Baseline	Chol1	Chol2
181-10500018	Competition	1	251	.	115
181-1050003	Great New Drug	1	181	.	165
181-14100015	Competition	1	164	226	.
181-2550004	Great New Drug	1	195	.	148
181-3030007	Competition	1	202	.	176

Figure 4. Sample JMP Journal for Confirming ITT Population



Validate EFF_Chol Data for DRAFT Phase

Confirm LOCF

- Missing post-treatment measurements are carried forward to the next treatment. Baseline measurements are not carried forward to a post-treatment measurement.
- Validation of ITT data showed 5 patients with missing efficacy measurements. Subset below shows SAS variables for these patients. The L-prefix variables are the LOCF variables. No issues.

Full ID	Treatment	ITT	Baseline	Chol1	Chol2	L_Chol1	L_Chol2
181-1050003	Great New Drug	1	181	.	165	.	165
181-2550004	Great New Drug	1	195	.	148	.	148
181-10500018	Competition	1	251	.	115	.	115
181-14100015	Competition	1	164	226	.	226	226
181-3030007	Competition	1	202	.	176	.	176

Confirm LOCF with Check Variables

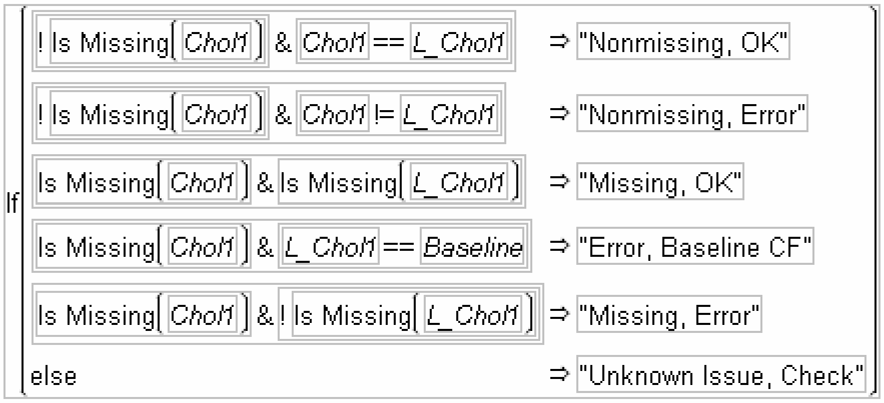
- Continuing example, suppose there were 50 patients with LOCF instead of 5. Verifying 'by hand' has a higher risk of error than verifying with calculated variables.

Use the Formula editor to create a variable that checks handling of missing values. See the "Process Notes" document for copy of the formula.

The S_LC1_ck variable summarizes various possibilities for correct assignment and for errors. Tables>Summary on this variable shown below. No issues.

S_LC1_ck	N Rows
Missing, OK	5
Nonmissing, OK	212

Figure 5. Sample JMP Journal for Confirming LOCF



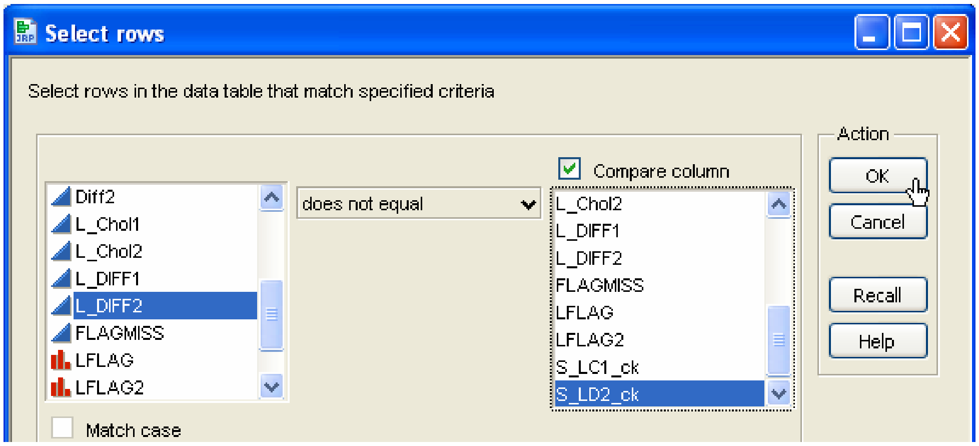
```
If  ! Is Missing( Chol1 ) & Chol1 == L_Chol1             ⇒ "Nonmissing, OK"
    ! Is Missing( Chol1 ) & Chol1 != L_Chol1             ⇒ "Nonmissing, Error"
    Is Missing( Chol1 ) & Is Missing( L_Chol1 )          ⇒ "Missing, OK"
    Is Missing( Chol1 ) & L_Chol1 == Baseline            ⇒ "Error, Baseline CF"
    Is Missing( Chol1 ) & ! Is Missing( L_Chol1 )        ⇒ "Missing, Error"
    else                                                 ⇒ "Unknown Issue, Check"
```

Figure 6. Sample JMP Formula used in Figure 5 for LOCF

Continuing the example, the analysis is performed on a difference variable after accounting for LOCF. If a patient’s measurement is nonmissing, then the difference uses the raw data. If the patient’s measurement is missing, then the difference uses the LOCF. To validate the derived variable, you can create a complex formula that includes IF-THEN conditions for possible calculations, similar to the example in Figure 6. JMP Version 8 provides a new feature that allows for direct comparison of variables using **Rows→Row Selection→Select Where**. For this case, create a simple difference variable,

S_LD2_ck = L_Chol2 - Baseline

where L_Chol2 is the LOCF variable from the SAS data set. This approach assumes you have previously verified the L_Chol2 variable, and you are now simply verifying the calculation of the difference. After creating the variable, use row selection. Figure 7 shows a portion of the JMP window.



Select rows

Select rows in the data table that match specified criteria

L_DIFF2 does not equal

☑ Compare column

S_LD2_ck

□ Match case

Figure 7. Comparing Two Variables using Select Where

Documenting the approach shown in Figure 7 might pose a challenge. The journal can state the activities performed to compare the two variables, and you could create a separate screen capture to show that no rows are selected to meet the criterion. Since this is a new feature with JMP Version 8, my experience in using it has been for data exploration and not for data validation. Future activities for clinical trials will help prove the value of this feature in validation.

These examples provide examples of how to validate data using JMP. My experience has included some very complex definitions for study populations, LOCF, and derived variables. After working with dozens of trials, JMP has been able to perform validation on all of these definitions.

Validating with SAS

Confirming the definitions for derived variables and missing data involves writing SAS programs. You are essentially replicating the DATA step programming performed by the primary programmers. One strength of using SAS is that you could carry validation a step further, and have another person perform independent peer review of both programs. This process could be useful in a large company. For example, suppose the validation programmer uses a different programming approach, which is more efficient. The clinical programming team might choose to use the new (validation) approach in future trials as the primary programming approach.

The same principles discussed for JMP are relevant here. The list below describes additional aspects of validating data with SAS.

- Use DATA step programming and assignment statements to create new variables that confirm the accuracy of derived variables. After independently creating your variables, you can create additional flag variables in the same DATA step. Suppose the derived efficacy variable is EFF1, and your independent variable is V_EFF1. Then, you can create FLAGVE=0 when EFF1=V_EFF1, and FLAGVE=1 otherwise. This helps you be more efficient. If FLAGVE=0 for all records, you do not need to perform further investigation. If FLAGVE=1 for some records, then you need to investigate those records. Perhaps either your programming or the primary programming contains an error, or perhaps there is a real data error to resolve.
- Alternatively, use the WHERE= option in the DATA step to identify records that are potential problems. Then, print only those records and investigate whether the data has been handled correctly or not.
- Use macros for repetitive tasks. For example, suppose the clinical trial occurs over a 12-week period. Develop a program to handle LOCF for the first efficacy timepoint. After performing your own QC on the program, convert the one-timepoints program into a macro that handles LOCF validation for the entire 12 weeks.
- In all the data validation programs, follow good programming practices. Use effective comments to describe the logic behind the data checks. You might need to return to this program much later and the comments will help you understand and remember the decisions.

This paper assumes the audience is more familiar with SAS than JMP, and doesn't discuss the programming aspects discussed above.

A key strength of SAS is the ability to quickly re-run validation programs on later phases for a trial or on future trials. Since a drug development program typically uses the same efficacy variables across pivotal trials, this strength leverages work on one trial to multiple trials, increasing efficiency and reducing programming costs.

Validating Tables and Listings

After validating the data, you can focus on validating tables and listings. Validating listings typically involves creating either a random sample or stratified random sample of the data and verifying the content of the listings against the data sets. Either JMP or SAS can provide random samples. This process differs little between the two software packages, and the paper doesn't discuss it further. Instead, the paper focuses on tables, where the process can differ a lot.

Validating table content can require very different activities depending on your choice of software. For SAS, you will need to create programs, run the programs, and check the output. Good programming practices dictate that you also save the logs and check the logs for errors. For JMP, you interactively create the results and create a journal that contains the JMP tables and graphs for a platform. Since JMP journals can be read only with JMP software, you need to create a PDF of the journal.

For some advanced statistical analyses, SAS is your only choice. Although you can develop analyses with the scripting language in JMP, this moves you into the territory of developing statistical software. In my role as a validator for trials, this is not the focus of my activities. For the purposes of this paper (and for my validation activities), assume that the limits of JMP validation are defined by the features available in the software.

This paper assumes the reader is generally familiar with SAS. Key strengths of using SAS to validate tables include:

- SAS is considered the “gold standard” for pharmaceutical analyses, and can perform virtually any analysis.
- SAS programming guidelines often exist for a client. When they don't, published books and papers provide information on best practices, including program headers, efficient coding, using comments, and more.
- ODS features allow you to control exactly which output tables are created. This can significantly reduce the amount of output that is printed or stored.
- Format libraries, once validated, can help create output that is easily interpreted and matches the output from the primary statistician.
- Using WHERE statements or data set options to control the data used for an analysis can minimize the number of additional data sets that you need to create.
- Tables that use a denominator based on the entire data rather than on actual counts are easier to create than in JMP. For example, tables that show adverse events or concomitant medications typically use a denominator of the number of patients on a treatment. JMP will use the denominator for the number of nonmissing rows.[1] Both SAS and JMP provide the correct counts in these tables. However, the percentages will differ because of the different denominators.
- You can often create a single program and modify it slightly to re-use for other tables in a trial, other trials for a client, and potentially across clients.

Most readers of this paper may be unfamiliar with JMP. Key strengths of using JMP to validate tables include:

- JMP is easy to learn, interactive, flexible, and fast.
- JMP journals provide features for adding significant documentation to the results. Use the journal to identify data sets, subsetting, coding for variables, and comments on the validation results.
- Value orders and value labels for variables allow you to replicate the formats used in SAS and the order of appearance of variable values in a table.
- By setting preferences for an analysis platform, you can minimize the extraneous information created. You can also include only selected results in the journal.
- Also, by setting preferences for reports, you can automatically include the date-time the analysis was run, and the data table used for the analysis on the report.

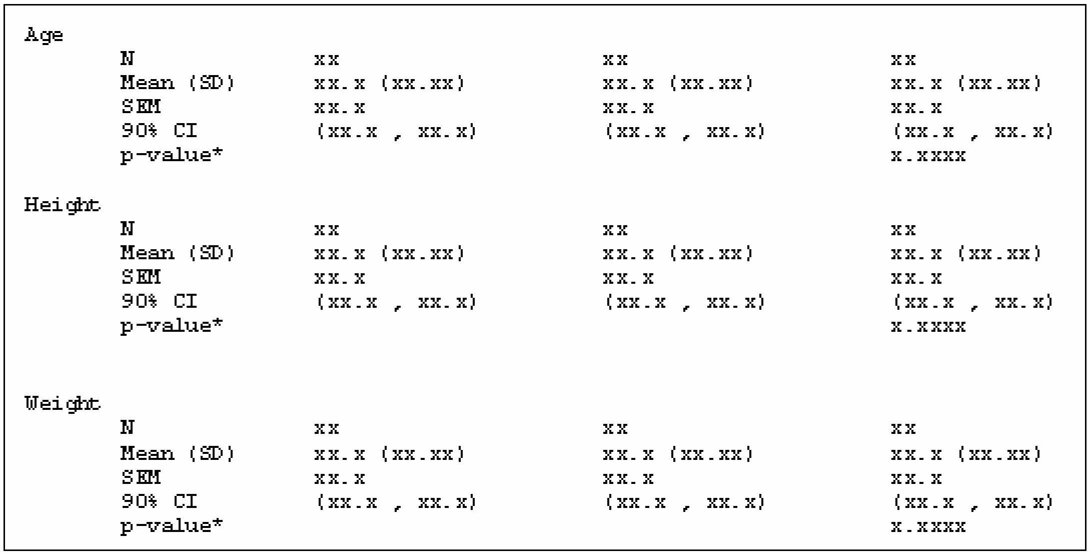
[1] This comment refers to JMP, and not to JMP Clinical. The author does not have direct experience using JMP Clinical to validate trials, and acknowledges that this issue might be mitigated by JMP Clinical.

- By using CTRL-ALT-click on the red-triangle option for analysis results, you can easily add analyses for all combinations of results—for example for an efficacy analysis that is repeated at multiple times post-dose.
- JMP can read SAS data sets and transport files directly. In addition, JMP creates Version 5 transport files (required by the FDA), should you need to provide transport files to a client.
- With Version 8, JMP has a greatly expanded ability to interact directly with SAS procedures. You can create SAS programs, run SAS programs, browse SAS data sets, and more.

Since any clinical trial has many tables, the best way to illustrate the strengths of JMP is with a couple of examples.

Example 1: Demographic Data

Most trials summarize demographic data for patients at the start of the study. The purpose is to show that there are no underlying differences among the treatment groups. In some cases, statistical tests are performed to compare treatment groups. Figure 8 shows a portion of a sample mock table. There are three demographic variables, and the goal is to obtain descriptive statistics for each variable. The columns represent three treatment groups. The printed table shows the treatments in a specific order, and validating will be easier if your results also follow that order. This mock table is modified from what you would use in an actual trial, in order to illustrate JMP features. An actual trial table would have either the confidence interval or the *p*-value from the statistical test, but not both.



Age				
	N	xx	xx	xx
	Mean (SD)	xx.x (xx.xx)	xx.x (xx.xx)	xx.x (xx.xx)
	SEM	xx.x	xx.x	xx.x
	90% CI	(xx.x , xx.x)	(xx.x , xx.x)	(xx.x , xx.x)
	p-value*			x.xxxx
Height				
	N	xx	xx	xx
	Mean (SD)	xx.x (xx.xx)	xx.x (xx.xx)	xx.x (xx.xx)
	SEM	xx.x	xx.x	xx.x
	90% CI	(xx.x , xx.x)	(xx.x , xx.x)	(xx.x , xx.x)
	p-value*			x.xxxx
Weight				
	N	xx	xx	xx
	Mean (SD)	xx.x (xx.xx)	xx.x (xx.xx)	xx.x (xx.xx)
	SEM	xx.x	xx.x	xx.x
	90% CI	(xx.x , xx.x)	(xx.x , xx.x)	(xx.x , xx.x)
	p-value*			x.xxxx

Figure 8. Sample Demographic Table

With SAS, you could replicate the contents of this table using PROC MEANS, PROC UNIVARIATE, PROC ANOVA, or other procedures. You would create a program, run it, check the SAS log for errors or warnings, and save the output. This paper does not show those activities.

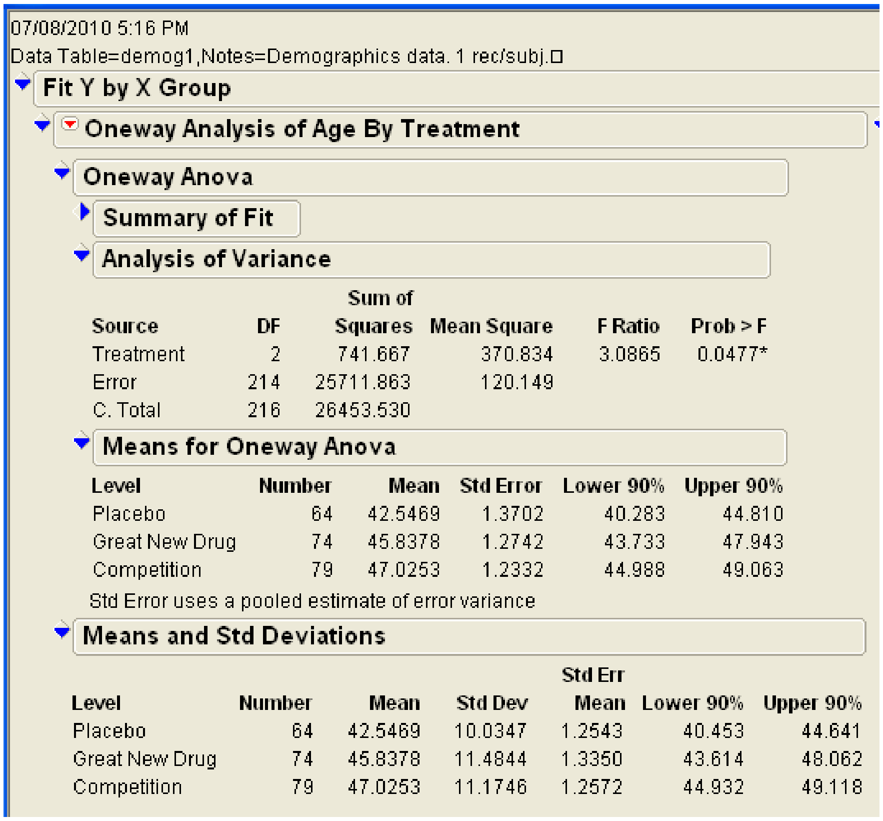
With JMP, you also have multiple choices for replicating the contents. The **Distribution** platform automatically produces most of these results, but not the *p*-value to compare treatment groups. Using **Tables→Summary** also produces descriptive statistics, but not the confidence interval or the *p*-value. Using **Fit Y by X**, with an X variable of Treatment and three Y variables creates a group of results.

Figure 9 shows the results for Age. These are not the automatic results produced by JMP. After the automatic results appear, use CTRL-ALT-click to display all possible options and then apply the options to all analyses (CTRL-option-click on the Mac). Select options to hide graphs, add an ANOVA, display means and descriptive statistics, and set the alpha level to 0.90. In Figure 9, notice the two tables that show means and standard deviations. One table uses the pooled standard deviations from the ANOVA, and another uses the simple standard deviations for each group to create the confidence intervals. This highlights the importance of reviewing the SAP—either approach could be "right," but only one approach will be appropriate for the trial. Figure 9 also shows the effect of setting JMP preferences to display the date-time the analysis was run, and to show the data table used for the analysis.

You could simply include these results into a journal and validate against the published table. However, JMP provides the **Make Combined Data** Table feature that condenses the results and makes validation easier and more efficient. Suppose you want to use the simple standard deviations for the confidence intervals. Right-click in any of the tables and select **Make Combined Data Table**. Figure 10 shows the results. Now, all of the descriptive statistics for the variables are on a single row for each variable-treatment combination and are simple to validate.

Figure 10 shows the automatic formatting for the statistics. Depending on the SAP, you might want to revise the number of decimal places. When doing so, consider the likelihood that the table will be revised. You can simply use the automatic formatting, and validate against the table. However, this requires you to perform any rounding activities on your own. Or, you can adjust the number of decimal places to match the printed table. However, adjusting means that if the client later decides to display more decimal places, you will need to recreate the results.

At this point, the results are accessible only in JMP. The next step is to document the validation efforts and save the results in a format that is easily accessible. Create a JMP journal, and use a combination of outline items and bulleted text items to provide documentation. You can describe the data set, formatting of variables, JMP analysis platforms used, and more. When thinking about documentation, think about what information would be required for a future auditor to replicate your results. Save the JMP journal, and store it in the appropriate folder in your file structure. Finally, to make the journal easily accessible print or save it as a PDF. Figure 11 shows the resulting PDF file.



07/08/2010 5:16 PM

Data Table=demog1,Notes=Demographics data. 1 rec/subj.□

Fit Y by X Group

Oneway Analysis of Age By Treatment

Oneway Anova

Summary of Fit

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Treatment	2	741.667	370.834	3.0865	0.0477*
Error	214	25711.863	120.149		
C. Total	216	26453.530			

Means for Oneway Anova

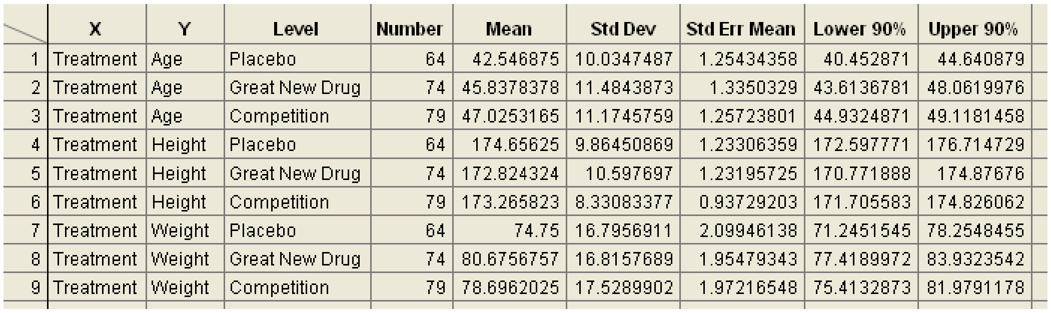
Level	Number	Mean	Std Error	Lower 90%	Upper 90%
Placebo	64	42.5469	1.3702	40.283	44.810
Great New Drug	74	45.8378	1.2742	43.733	47.943
Competition	79	47.0253	1.2332	44.988	49.063

Std Error uses a pooled estimate of error variance

Means and Std Deviations

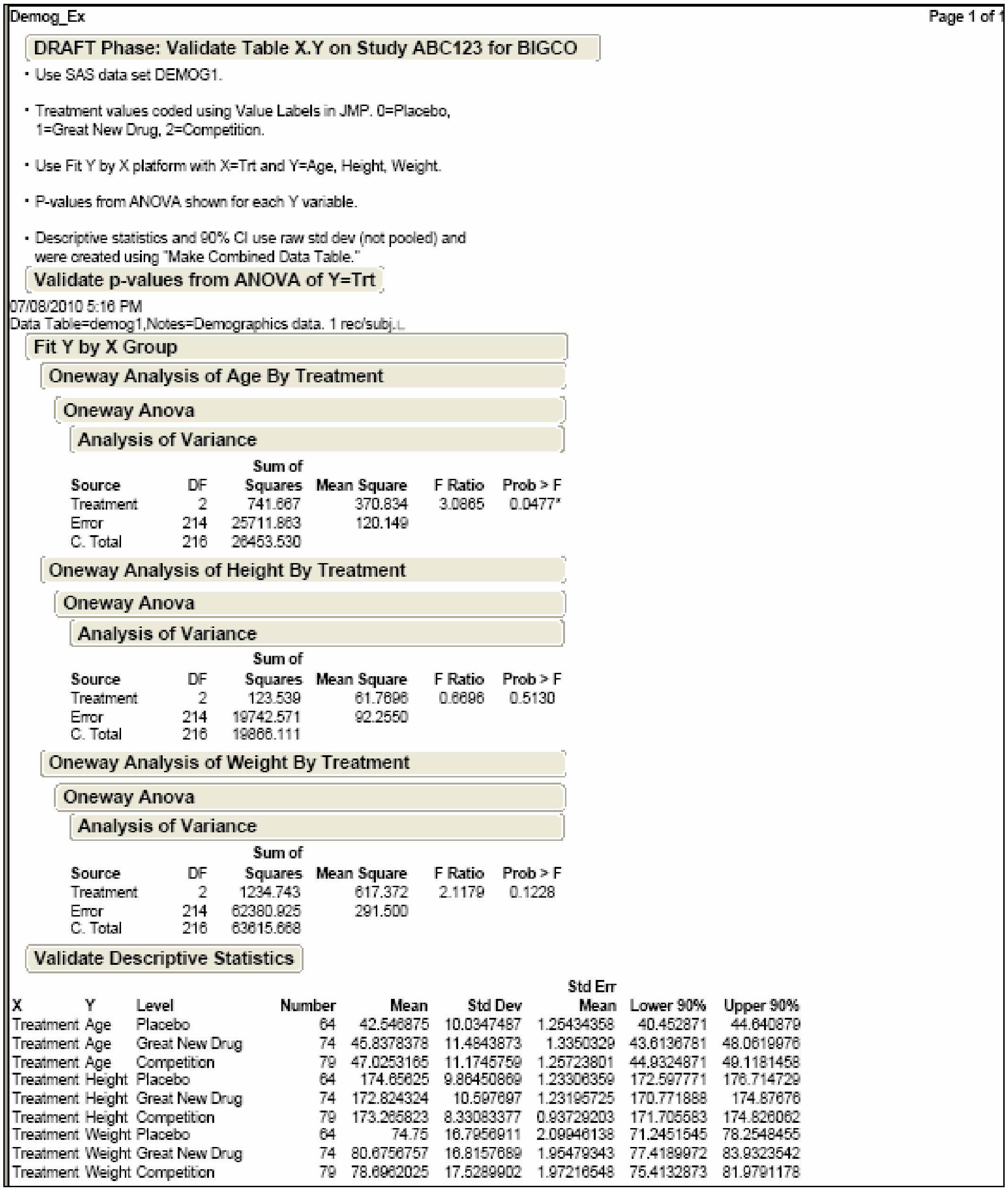
Level	Number	Mean	Std Dev	Std Err Mean	Lower 90%	Upper 90%
Placebo	64	42.5469	10.0347	1.2543	40.453	44.641
Great New Drug	74	45.8378	11.4844	1.3350	43.614	48.062
Competition	79	47.0253	11.1746	1.2572	44.932	49.118

Figure 9. Sample Fit Y by X Results for One Variable



	X	Y	Level	Number	Mean	Std Dev	Std Err Mean	Lower 90%	Upper 90%
1	Treatment	Age	Placebo	64	42.546875	10.0347487	1.25434358	40.452871	44.640879
2	Treatment	Age	Great New Drug	74	45.8378378	11.4843873	1.3350329	43.6136781	48.0619976
3	Treatment	Age	Competition	79	47.0253165	11.1745759	1.25723801	44.9324871	49.1181458
4	Treatment	Height	Placebo	64	174.65625	9.86450869	1.23306359	172.597771	176.714729
5	Treatment	Height	Great New Drug	74	172.824324	10.597697	1.23195725	170.771888	174.87676
6	Treatment	Height	Competition	79	173.265823	8.33083377	0.93729203	171.705583	174.826062
7	Treatment	Weight	Placebo	64	74.75	16.7956911	2.09946138	71.2451545	78.2548455
8	Treatment	Weight	Great New Drug	74	80.6756757	16.8157689	1.95479343	77.4189972	83.9323542
9	Treatment	Weight	Competition	79	78.6962025	17.5289902	1.97216548	75.4132873	81.9791178

Figure 10. Descriptive Statistics Combined into a Data Table



Demog_Ex

Page 1 of 1

DRAFT Phase: Validate Table X.Y on Study ABC123 for BIGCO

- Use SAS data set DEMOG1.
- Treatment values coded using Value Labels in JMP. 0=Placebo, 1=Great New Drug, 2=Competition.
- Use Fit Y by X platform with X=Trt and Y=Age, Height, Weight.
- P-values from ANOVA shown for each Y variable.
- Descriptive statistics and 90% CI use raw std dev (not pooled) and were created using "Make Combined Data Table."

Validate p-values from ANOVA of Y=Trt

07/08/2010 5:16 PM
Data Table=demog1,Notes=Demographics data. 1 rec/subj.

Fit Y by X Group

Oneway Analysis of Age By Treatment

Oneway Anova

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Treatment	2	741.667	370.834	3.0865	0.0477*
Error	214	25711.863	120.149		
C. Total	216	26453.530			

Oneway Analysis of Height By Treatment

Oneway Anova

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Treatment	2	123.539	61.7696	0.6696	0.5130
Error	214	19742.571	92.2550		
C. Total	216	19866.111			

Oneway Analysis of Weight By Treatment

Oneway Anova

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F
Treatment	2	1234.743	617.372	2.1179	0.1228
Error	214	62380.925	291.500		
C. Total	216	63615.668			

Validate Descriptive Statistics

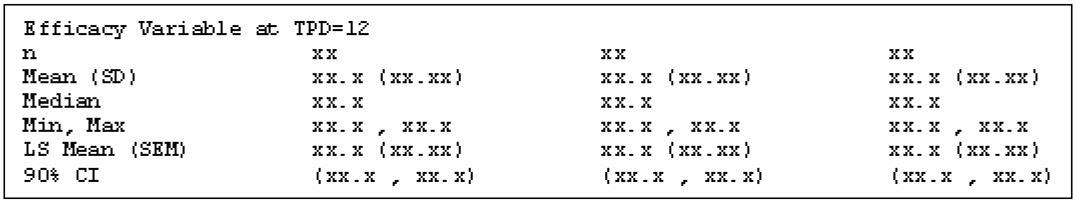
X	Y	Level	Number	Mean	Std Dev	Std Err Mean	Lower 90%	Upper 90%
Treatment	Age	Placebo	64	42.546875	10.0347487	1.25434358	40.452871	44.640879
Treatment	Age	Great New Drug	74	45.8378378	11.4843873	1.3350329	43.6136781	48.0619976
Treatment	Age	Competition	79	47.0253165	11.1745759	1.25723801	44.9324871	49.1181458
Treatment	Height	Placebo	64	174.65625	9.86450869	1.23306359	172.597771	176.714729
Treatment	Height	Great New Drug	74	172.824324	10.597697	1.23195725	170.771888	174.87676
Treatment	Height	Competition	79	173.265823	8.33083377	0.93729203	171.705583	174.826062
Treatment	Weight	Placebo	64	74.75	16.7956911	2.09946138	71.2451545	78.2548455
Treatment	Weight	Great New Drug	74	80.6756757	16.8157689	1.95479343	77.4189972	83.9323542
Treatment	Weight	Competition	79	78.6962025	17.5289902	1.97216548	75.4132873	81.9791178

Figure 11. PDF of JMP Journal for Example 1

JMP can create very elaborate journals, and can also create project files. You could essentially create a single file that contains all the validation activities performed in JMP. However, consider a future audit. You want to be able to answer audit questions about a single data set or TLF without needing to also show results from other activities. As a result, my recommendation is to create a single JMP journal file (and the accompanying PDF) for each data set or TLF validation.

Example 2: Efficacy Analysis

Suppose an efficacy variable is measured at several times after patients receive a treatment dose. This approach is common in many types of trials. For the example, the variable that measures time post-dose is TPD. For each time post-dose, the SAP specifies that tables will contain descriptive statistics, least-squares means (LS Mean) from a complex model, and a 90% confidence interval on the LS Mean. Because of the complexity of the model, you must use SAS to generate the LS Mean. However, you want to use JMP to validate the descriptive statistics. For this example, you will create two PDF files for validation—one that contains the JMP results, and a second that contains the SAS results. Figure 12 shows a portion of a sample mock table.



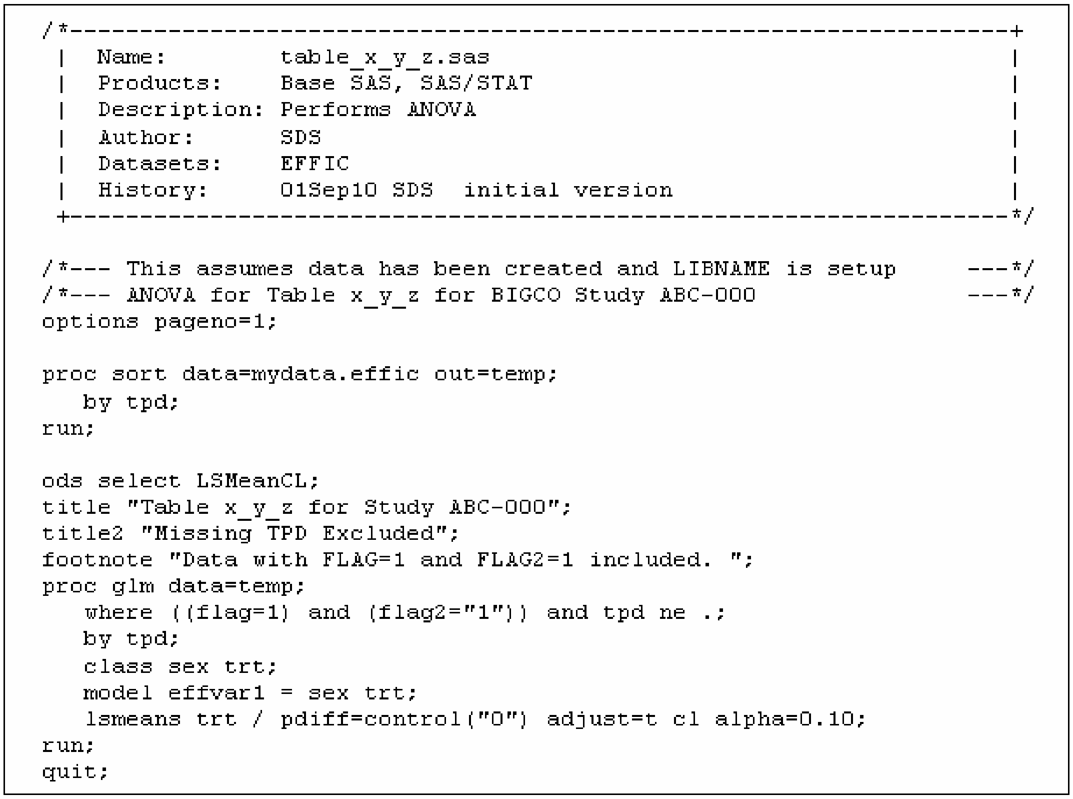
```
Efficacy Variable at TPD=12
n                        xx                  xx                  xx
Mean (SD)                xx.x (xx.xx)        xx.x (xx.xx)        xx.x (xx.xx)
Median                   xx.x                xx.x                xx.x
Min, Max                 xx.x , xx.x         xx.x , xx.x         xx.x , xx.x
LS Mean (SEM)            xx.x (xx.xx)        xx.x (xx.xx)        xx.x (xx.xx)
90% CI                   (xx.x , xx.x)       (xx.x , xx.x)       (xx.x , xx.x)
```

Figure 12. Sample Efficacy Table

You can validate the descriptive statistics using several JMP platforms. Using **Tables→Summary** produces the most compact summary, and one that is easily validated against the printed table. For this example, use a **Group** variable of TPD, and specify the statistics for the efficacy variable. For faster validation, specify the statistics in the order they appear in the printed table. Although JMP automatically produces the **Nrows** variable, you will want to include the **N** statistic for the efficacy variable. For the JMP validation, again create a JMP journal with appropriate documentation and print this journal to a PDF.

For the SAS validation, you need to create a program, run it, check the log for warnings or errors, and save the output. As with JMP, saving the output as a PDF gives the most potential flexibility for accessing it in the future. Although your client has SAS software, as will the regulatory agency, you might be required to show results to an auditor who does not have SAS software.

Figure 13 shows a sample SAS program to create the results. For example purposes, the model is simpler than the complex model actually used in the trial. The program header is abbreviated also, but shows key points of identifying the trial, software and data required, program author, history and so on. The program itself also contains comments, and uses titles and footnotes to provide further documentation. The ODS statement limits the output produced, which makes it easier to validate the results because there is less extraneous information to look through. The WHERE statement limits the data used in analysis, and those limits are explained in the combination of the TITLE2 and FOOTNOTE statements. The DATA step sorts the data by TPD and creates a temporary data set that is used for analysis. This is not strictly necessary, but it will make validation easier because the results will be in the same order as the printed table.



```
/*-------------------------------------------------------------------+
 |  Name:        table_x_y_z.sas                                     |
 |  Products:    Base SAS, SAS/STAT                                  |
 |  Description: Performs ANOVA                                      |
 |  Author:      SDS                                                 |
 |  Datasets:    EFFIC                                               |
 |  History:     01Sep10 SDS  initial version                        |
 +-------------------------------------------------------------------*/

/*--- This assumes data has been created and LIBNAME is setup     ---*/
/*--- ANOVA for Table x_y_z for BIGCO Study ABC-000                ---*/
options pageno=1;

proc sort data=mydata.effic out=temp;
   by tpd;
run;

ods select LSMeanCL;
title "Table x_y_z for Study ABC-000";
title2 "Missing TPD Excluded";
footnote "Data with FLAG=1 and FLAG2=1 included. ";
proc glm data=temp;
   where ((flag=1) and (flag2="1")) and tpd ne .;
   by tpd;
   class sex trt;
   model effvar1 = sex trt;
   lsmeans trt / pdiff=control("0") adjust=t cl alpha=0.10;
run;
quit;
```

Figure 13. Sample SAS Program for Portion of Example 2 Table

Using JMP Scripting

JMP scripts can save time when re-running analyses. This is especially helpful when you create scripts in the Blinded phase that you can use later in the Draft and Final phases. However, you need to balance the amount of time required to create and manage the scripts against the time required to simply re-run the analysis in JMP.

Sometimes, you might find it simpler to set JMP preferences to meet your needs and re-run the analysis. For example, for two nominal variables JMP automatically creates a mosaic plot. JMP does not automatically perform statistical tests. Suppose you do not want the mosaic plot and you do want to perform a Chi-square test. For validation, you can set preferences to omit the plot, and to perform the Chi-square tests.

Look again at Example 1. For the three **Fit Y by X** analyses, you can save each script to the data table. Simply select the red triangle for each analysis and select **Script→Save Script to Data Table**. Later, you can simply click on the script buttons in the data table and re-run the analysis.

However, since you already know you need to re-run all three analyses, you might want to save one script that runs all analyses. Select the red triangle for an analysis and select **Script→Save Script for All Objects**. This saves the script in a JMP scripting window, and you can then save the script with a useful name. Figure 14 shows the script created using these steps for the analyses in Example 1.

Another reason for saving the scripts separate from the data table involves the multiple phases of validation. Once you validate the Blinded phase, you are not likely to re-run the validation. Instead, you will next validate the Draft tables, and will use different data. By saving the script independent of the data table, you can re-run it against other data tables that have the same name and same variables.

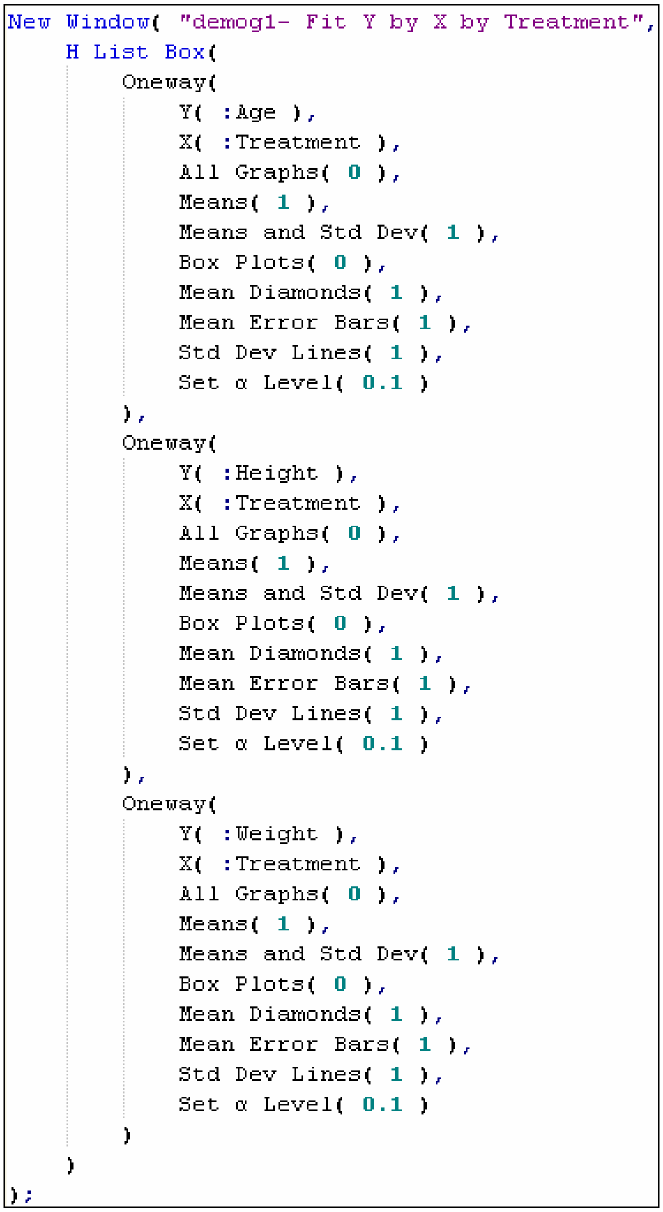
This approach uses very simple JMP scripting—saving an activity. JMP provides very advanced scripting, and allows you to create windows where you specify variables for analyses, build your own analyses, and more. However, you need to balance the work of essentially writing a program against using JMP interactively. If you program advanced analyses, you will need to perform the same type of programming validation that you would in another language. My experience has been that evaluating development activity against simply using JMP as-is or with saved scripts leads to the conclusion not to do programming development. Others, especially those who are very experienced with JMP's scripting language, might make a different choice.

Validating Figures

Validating figures using JMP depends on the figure and the information displayed. In some cases, validation involves cross-checking information displayed against an existing table. In other cases, validation involves performing an analysis and then checking the figure displayed.

Consider the first situation. Suppose the figure displays statistics that are summarized in an existing table. For example, suppose the SAP specifies figures that show the least-squares means and confidence intervals for the efficacy variable at each time post-dose. The statistics plotted in these figures are contained in the table for Example 2. Validation involves simply checking that the appropriate statistics are displayed in the table, and that the appearance requirements in the SAP are met. If the SAP specifies different colors, markers, or line types for each treatment, validation involves confirming the SAP requirements. In general, recreating the same figure in JMP is not necessary.

Consider the second situation. Suppose the SAP specifies survival analysis. The raw SAS output is not provided as a table. Instead, the appropriate survival analysis figures are created and annotated with *p*-values. In this situation, validation involves performing the analysis and generating the figure. For validation purposes, you do not need to recreate the annotation, but you do need to validate the information in the annotation. Validating this figure then becomes similar to validating any of the tables. Evaluate whether JMP can perform the analysis, or whether you need to use SAS. If you can use JMP, create the output table and graph, save the results in a journal, and print the journal to a PDF file. As with the validation of tables, be sure to include enough information in the journal so that others could replicate your results. If you need to use SAS, create a program, run the program, check the log for errors, and save the results in a PDF file.



```
New Window( "demog1- Fit Y by X by Treatment",
    H List Box(
        Oneway(
            Y( :Age ),
            X( :Treatment ),
            All Graphs( 0 ),
            Means( 1 ),
            Means and Std Dev( 1 ),
            Box Plots( 0 ),
            Mean Diamonds( 1 ),
            Mean Error Bars( 1 ),
            Std Dev Lines( 1 ),
            Set α Level( 0.1 )
        ),
        Oneway(
            Y( :Height ),
            X( :Treatment ),
            All Graphs( 0 ),
            Means( 1 ),
            Means and Std Dev( 1 ),
            Box Plots( 0 ),
            Mean Diamonds( 1 ),
            Mean Error Bars( 1 ),
            Std Dev Lines( 1 ),
            Set α Level( 0.1 )
        ),
        Oneway(
            Y( :Weight ),
            X( :Treatment ),
            All Graphs( 0 ),
            Means( 1 ),
            Means and Std Dev( 1 ),
            Box Plots( 0 ),
            Mean Diamonds( 1 ),
            Mean Error Bars( 1 ),
            Std Dev Lines( 1 ),
            Set α Level( 0.1 )
        )
    )
);
```

Figure 14. Sample Script for Example 1 Table

Conclusion

The intent of this paper is to show that JMP can be effectively used for validating clinical trials, or can be used to augment SAS for validation. The table below summarizes the strengths of JMP and SAS for validating data sets and TLFs for clinical trials.

JMP Strengths	SAS Strengths
No programming knowledge required.	Considered the gold standard for analysis.
Performs many analyses.	Can perform virtually any analyses, and uses the same programming language as the original analysis.
Interactive and flexible.	You can re-run programs with virtually no effort.
With selective saving of JMP scripts, you can quickly re-run validation activities on later phases of validation.	With effective header blocks and comments, programs can be self-documenting.
By setting JMP preferences for platforms, you can minimize extraneous information in the results.	With effective use of ODS, you can minimize extraneous information in the results.
With effective comments in journals, results can be self-documenting. Using JMP preferences to include the data table and date-time stamp in the results adds to this documentation.	With effective titles and footnotes, output can be self-documenting. SAS automatically includes the date-time stamp on the output.
Adding value labels and value orders to variables in a data table facilitates validation against the printed table.	Format catalogs used for primary analysis can be used for validation after an initial check of the formatted values.
Creating PDFs of JMP journals makes the results readable by clients who do not have the software.	Creating PDFs of SAS Logs and SAS output makes these files readable by clients who do not have the software.
Focus on data exploration helps find data issues with no programming, and the intuitive formula editor helps in simplifying validation of calculated variables.	Can perform very complex data joins and merges, which can be necessary for some tables.
Excluding rows from analysis can reduce the need for creating additional data tables.	Using a complex WHERE statements can reduce the need for creating additional data sets.
Lower cost of entry and maintenance of software, and simpler installation of software.	Programs easily modified for different trials within a drug development effort.

The obstacles to using each software package are essentially the reverse of the strengths. For example, with JMP, you will find analyses that require SAS. For SAS, you will have a higher cost of entry.

Practical Considerations

In using the software packages to perform validation as an independent consultant, you will also want to consider the practical aspects of using the software. For SAS, initial and ongoing license fees are considerably higher than for JMP. If you are using the software on your own computer, the return-on-investment (ROI) decision depends on how many trials you will validate in a year (and on how much of your other work involves the software). If you are using SAS on a client’s PC or server, the ROI decision is less relevant.

Similarly, installing JMP is simpler than installing SAS. Unlike SAS, a single-user version of JMP does not require an annual license renewal. Since installing and administrative tasks require time, only you can decide whether the time invested is appropriate for the amount of validation activities you perform with each software package.

Recommendations

For many large companies, SAS will likely continue to be the software package used for validating clinical trials. These companies have a significant investment in the software and the infrastructure to support it. In addition, SAS remains the “gold standard” for analyses. For smaller companies, and especially for independent consultants, JMP provides a viable alternative for validating data and TLFs from clinical trials. The recently released JMP Clinical software shows a commitment to expand support for the drug development process.

In making your own decision, consider the strengths of each software package, the amount of your workload that will require the package, and the practical considerations for using JMP or SAS.

Acknowledgements

Special thanks to Steve Horohonich and Dr. Julie Jones first for giving me the opportunity to perform statistical validation on many trials, using both JMP and SAS; and second, for reviewing drafts of this paper.

References

Matthews, Carol I., and Brian C. Shilling, 2008. *Validating Clinical Trial Data Reporting with SAS®*. Cary, NC: SAS Institute Inc.

Recommended Reading

An informed validator is a more effective validator. Familiarize yourself with regulatory guidelines for clinical trials. These include Good Clinical Practices (GCP), ICH E9 for statistical principles, CDISC for data standards, and 21CFR Part 11 for digital signatures.

Contact Information

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

Sandra D. Schlotzhauer
Schlotzhauer Consulting LLC
400 Gibbon Drive
Chapel Hill, NC 27516
919.933.6341
sandra@schlotzcon.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.