PharmaSUG 2013 - Paper PO17

Validating Listing Output: A Better Way

Hunter Vega, Stat-Tech Services, LLC, Chapel Hill, NC
James Kniffen Jr., Stat-Tech Services, LLC, Chapel Hill, NC

ABSTRACT

When using the SAS® System to create data listings for a clinical study, RTF output can range in size from one page to thousands of pages. In the past, this output was evaluated for accuracy using either the process of parallel programming and manual evaluation of selected records, or a comparison of output data sets using PROC COMPARE. While manual checks of long listings are demanding, they help identify errors such as truncation or incorrect formats that can be missed in a purely programmatic approach. The LISTCOMP macro extracts data directly from an RTF file, compares it to a parallel programmer's data set, and then provides a summary of each of the columns in the listings. Use of the macro is not dependent upon the version of SAS, but implementing the macro may require a moderate level of programming expertise. This approach not only ensures that the actual RTF output matches the parallel programmer's output, but that the reporting of information displayed in the RTF output allows for rapid review and identification of values that are impossible, incorrectly formatted, unformatted, or truncated. The LISTCOMP macro both reduces the validation time needed and increases the likelihood that problems in the output are identified.

INTRODUCTION

Clinical drug and device studies often ask for data listings summarizing each subject's data in ways that can be easily referenced by reviewers and medical writers. In a demographics listing, only one record is displayed per patient, summarizing data such as age, gender, weight, etc. However, laboratory results listings frequently include many pages of output for each subject. As the output becomes larger, it becomes harder for the statistician or data manager to verify that all of the data is correctly displayed.

The LISTCOMP macro creates a way to compare the RTF output to a parallel programmer's output, while also allowing a third party reviewer to examine the listing for any inaccurate displays of the data that would not have been considered incorrect when using only PROC COMPARE. With these comparisons, the statistician can focus on reviewing the output for inconsistencies in number of records, number of subjects displayed, and actual data values.

MACRO SYNTAX

```
%LISTCOMP
     (file=..\..\output\listing\demog.lis.rtf,
     dset=demog_val,
     splitc=%bquote(*),
     varord=subjid subj icdate date age num
            sex cat comments char,
     datefmt=date9.
);
```

→filepath for RTF listing

→parallel programmer's data set

→if a split character is used in the parallel programming, this translates any instances of "\line" in the RTF file into the split character.

→this lists the variables in the parallel programmer's data set in the order the data is displayed in the RTF file followed by the variable type: SUBJ, CHAR, NUM, CAT, or DATE

→if there are any date variables, this gives the date format, such as "date9." or "mmddyy10.".

PHASE 1: IMPORTING THE RTF LISTING

When the RTF file is imported into SAS, the RTF code must be parsed to find the actual data that is displayed in the listing. The macro looks for a dummy RTF tag (in this case, "\bkmstart") in the RTF file to indicate where the data display is beginning in the listing. A "\row" tag indicates that a new line of data is starting. The macro also accounts for any ordering variables in the RTF file so that there are no missing values for the ID variables in the imported data. The macro also looks for an RTF tag ("\SubjectID") to indicate the patient number or data record identifier.

The parallel programmer's data set (or "QC data set") must have any formats (dates, specific numeric formats, etc.) already applied to the variables before calling the macro. The macro reads in the QC data set, applies all formats, and puts the variables in the proper order in the QC data set to ensure an accurate comparison.

PHASE 2: SUBJECT AND OBSERVATION REPORTING

The "\SubjectID" tag in the RTF file and the "SUBJ" variable type from the VARORD statement let the macro locate the subject identifier values. A frequency analysis of these values for both the RTF file and QC data set produce a report of the number of subjects and observations found from both data sources. A list of all subjects found (in either one or both sources) is also provided to ensure accuracy.

Once the RTF tags have been processed, the macro creates a data set of the RTF listing output which can be used to compare to a parallel programmer's data set. The QC data set will need to sort the listing records in exactly the same order as they appear in the RTF listing. The macro then compares these independently created data sets and gives a summary of the number of subjects, the number of records, and a display of non-matching subject identifier values if applicable.

```
IP Listing Observations found: 83
IP Listing Subjects found: 83

QC Listing Observations found: 82
QC Listing Subjects found: 82
```

```
    Patients Inventory Report
1 Subject(s) in IP but not in QCP



               3089
```

Phase 2 Example. SAS Output for Subject Comparison

PHASE 3: PROC COMPARE

The processed RTF and QC data sets are then compared using PROC COMPARE to ensure that the data values match. If they match, a message is printed indicating that the compare came back clean. If they do not match, a list of non-matching variables is printed along with a value-by-value indication of discrepant values. To keep the output succinct, only the first 100 discrepancies are printed from the PROC COMPARE output using the "maxprint=100" option.

```
The following variables do not match:
Column 1: subjid
Column 2: icdate
Column 3: age
Column 4: sex
Column 5: comments
```

Obs	Base Value icdate	Compare Value icdate
6	31JUL2009	05AUG2009
7	05AUG2009	16OCT2009
8	16OCT2009	07AUG2009
9	07AUG2009	13AUG2009
10	13AUG2009	21SEP2009
11	21SEP2009	05SEP2009

Phase 3 Example. SAS Output for Data Conflicts

PHASE 4: VARIABLE REPORT

The final phase of the macro is a variable-by-variable report of the RTF output versus the parallel programmer's data. This is produced to allow the reviewer to easily spot problems that PROC COMPARE might miss, such as an incorrect format or a long character string that is being truncated in both the RTF and QC data sets. There are four variable types supported in the macro:

- **Numeric values:** The five lowest and five highest non-missing values are displayed for both the RTF and QC data sets. A count of the missing values in both the RTF and QC data is also displayed.
- **Text strings:** The five longest non-missing text strings are displayed in descending order of length. A count of the number of missing text strings in both the RTF and QC data is displayed.
- **Categorical values:** A frequency analysis of both the RTF and QC data is displayed, along with a count of missing values in the RTF and QC data.
- **Dates:** The five earliest and five latest dates that are not missing or partial (i.e.; 11/UNK/2012) are displayed. A count of partial or missing dates in both the RTF and QC data are also displayed.

In the example output shown here, you can see that the parallel programmer's data set has one patient with missing values for each of the variables displayed. You can also see that the formatted values for the "sex" variable are different between the RTF and QC, and that the missing patient from the QC output creates visible discrepancies in the "age" and "comments" variables.

Variable	IP Dataset	QC Dataset
Column 2: icdate	Extreme Date Values	Extreme Date Values
	10JUL2009	10JUL2009
	12JUL2009	12JUL2009
	24JUL2009	24JUL2009
	24JUL2009	24JUL2009
	25JUL2009	25JUL2009
	15DEC2009	15DEC2009
	15DEC2009	15DEC2009
	17DEC2009	17DEC2009
	17DEC2009	17DEC2009
	20DEC2009	20DEC2009
	Partial or missing = 11	Partial or missing = 12
Column 3: age	Extreme Numeric Values	Extreme Numeric Values
	20	20
	21	21
	21	21
	21	21
	21	21
	59	58
	59	59
	59	59
	59	59
	60	60
	Missing = 0	Missing = 1

Variable	IP Dataset	QC Dataset
Column 4: sex	F: 0 Female: 73 (88.0%) M: 0 Male: 10 (12.0%) Missing = 0	F: 72 (87.8%) Female: 0 M: 9 (11.0%) Male: 0 Missing = 1
Column 5: comments	Text Strings (Length) String (127) Subject showed adverse reaction (illness) to device 36 hours after installation. Device removed and subject discontinued study. (107) Subject failed exclusion criteria 4, but continued in study. Device installed normally 27oct09 without uade (43) Subject lost to follow-up at 12-month visit (43) Subject lost to follow-up at 12-month visit (43) Subject lost to follow-up at 12-month visit Missing = 15	Text Strings (Length) String (127) Subject showed adverse reaction (illness) to device 36 hours after installation. Device removed and subject discontinued study. (107) Subject failed exclusion criteria 4, but continued in study. Device installed normally 27oct09 without uade (43) Subject lost to follow-up at 12-month visit (43) Subject lost to follow-up at 12-month visit (34) Subject failed at IE; Not treated. Missing = 16

Phase 4 Example. SAS Output Showing Content Summaries

CONCLUSION

A vital component of a clinical trial is the accurate collection and analysis of subject data; therefore, validating the data for a study is a crucial step in the process of completing any trial. Since studies are constantly increasing in size, the amount of data collected continues to grow. Even though data listings can be thousands of pages long, longer listings should still be as accurate as a short output files.

The LISTCOMP macro can import RTF data listings and evaluate the validity of the listing to a parallel programmer's data. Since the macro takes considerably less time to run than someone manually reviewing a sample of the listing records, the time savings will be significant. Confidence can be placed in the accuracy of the data displayed as the LISTCOMP macro also allows for a reviewer to check for any data discrepancies. This programmatic method for verifying data listings will help expedite the delivery of a package from CRO to client, saving both time and money.

CONTACT INFORMATION

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

Name: Hunter Vega
Company: Stat-Tech Services, LLC
Address: 501 Eastowne Drive, Suite 230
City, State ZIP: Chapel Hill, NC 27514
Work Phone: 919-929-5015
Fax: 919-928-9320
E-mail: hvega@stattechservices.com
Web: www.stattechservices.com