

A Proc Compare Enhancement Macro

Chris Weiner, Bristol Myers Squibb Corporation, New Brunswick, NJ, USA

ABSTRACT

Proc compare is commonly used in clinical trial data validation for SDTM, ADaM, and TLF datasets, but its default outputs can be both difficult to interpret and incomplete, especially when investigating discrepancies. This enhanced macro offers significant improvements, including more readable reports, detailed summaries of non-matching ID records, and output datasets for unmatched and duplicate observations. It provides advanced comparative analyses, such as univariate statistics, percent missing, and distinct value comparisons for specified variables. String differences are clearly highlighted—even for nonprintable special characters—making it easier to spot subtle issues in long text fields. The macro generates comprehensive duplicate reports, indicating which variables differentiate duplicates and summarizing results in a one-record output dataset for easier downstream confirmation and tracking. By automating these enhancements, users can streamline follow-up tasks, improve transparency in discrepancy analysis, and facilitate validation workflows for clinical data, reducing manual effort and increasing reliability.

INTRODUCTION

The requirement to compare datasets is central to the validation process. As clinical trials programmer analysts, we need to report these findings effectively both the validation programmer and to those outside of statistical programming handling the clinical data. Proc compare has become the standard for validation of SDTM, ADaM, TLFs and other data from clinical trials. It has existed since version 5 in the 1980s. Some of the proc compare generated reports require further follow-up work to effectively investigate and communicate and summarize differences. This macro intends to reduce some of this follow up work. The lengthier the initial validation findings, the more worthwhile these improvements and added features will be to the user. For situations with few or no differences and for which proc compare is generating what we need for these differences, we recommend proc compare only. The qcompare macro is good for a first pass at a messy comparison.

In case of a complete match, qcompare runs an internal proc compare to confirm. This complies with any department requirements to use the SAS proc compare procedure as final criterion for validation.

LIMITATIONS OF PROC COMPARE

- Proc compare truncates to 20 characters in its listings, though not in its output datasets. We quite commonly have subject identifiers over 20 characters, and this requires extra work to manipulate the output dataset to clearly list out differences, particularly if we want to report to others to further investigate and resolve these cases. Better to have well formatted list we can email directly out of the qcompare output.
- The proc compare listing for non-matching ID records between base and compare could be clearer and more concise.
- Proc compare does not save this non-matching ID variable group list into a dataset, although these might be of great importance as a validation finding to pass on to others for further investigation and resolution.
- Proc compare reports if duplicated records exist within specific ID groups, without saving an output dataset containing these duplicates. Does not print all duplicates in output report, only the first one. Providing a clearly readable full listing of all duplicate records, and providing some analysis of which of the columns duplicate and which do not, can bring us close to understanding and resolving unacceptable duplicates.
- In case a base character string and a compare character string differ on nonprintable characters, proc compare correctly reports two strings as different. However in the proc compare differences listing, the strings look the same, without reporting exactly what distinguishes base and compare.

ADDITIONAL COMPARATIVE REPORTS PROVIDED BY QCOMPARE

- A comparison of univariate statistics for numerical variables, including percent missing
- A comparison of distinct values between base and compare for a designated variable list. For example, if comparing two lab datasets each with many records per subject and parameter and date, we might want to know if entire subjects or entire paramcds are not common between base and compare.
- Differences in base and compare strings are highlighted to assist in identifying small differences in very long strings
- A one record output dataset summarizing whether base and compare agree so that this can be read in by other programs to check validation status more easily

EXAMPLE MACRO CALL

Our example is a simplified adlb dataset that requires validation. The dataset has the subject, lab parameter and date as the key required to be unique to each record by the ADaM data specifications. We will allow an absolute difference of 10^{-7} between source and base values to be counted as a match, otherwise report a difference. It will also do checks on distinct values of IDs, which we always need to check in such a qc should there be any differences. We want to know if there are any subjects not common to the base and compare, any paramcds not common to the base and compare, and if there are any usubjid-paramcd combinations not common to base and compare. Duplicates, matches and differences will all be saved as output datasets. If any difference listing exceeds 500 records, it will only print the first 500 records. A footnote stating X of total Y records have been printed will be output in such cases.

```
%qcompare(  
  base = basesub,  
  compare = comparesub,  
  idvars = usubjid paramcd adt,  
  abs_tolerance = .0000001,  
  pct_tolerance = ,  
  idgroups = usubjid | paramcd | usubjid paramcd,  
  dup_base_output_ds = base_dups,  
  dup_compare_output_ds = compare_dups,  
  diff_output_ds = diffs_output,  
  match_output_ds = match,  
  max_obs_to_print = 500  
);
```

PARAMETER DEFINITION

Parameter Name	Description	Required
Base	Base dataset name	Yes
Compare	Compare dataset name	Yes
Idvars	ID variables	No. Defaults to observation number match
Abs_tolerance	Tolerance level to consider two numeric values equal	No ,default 0
Pct_tolerance	Percent tolerance level to consider two numeric values equal	No, default 0
Idgroups	Check the agreement of base and compare datasets of these groupings. Any number of groupings can be specified, delimited by .	No
Dup_base_output_ds	Base duplicate output dataset name	No, default to no dataset generated
Dup_compare_output_ds	Compare output dataset name	No, default to no dataset generated
Diff_output_ds	Differences output dataset name	No, default to no dataset generated
Max_obs_to_print	Maximum observations to print in any one report	No, default 100
Outfile	Output Filename	No, default to program name

If both absolute and percent tolerance are specified, then a base and compare values will have to be within tolerance level for both percent and absolute value in order to be considered a match.

EXAMPLE DATA

The square character is how SAS studio renders unprintable characters in Table 1 within the long comment. Base dataset has exactly duplicated records in yellow, a long comment in blue and a nonprintable character in red.

Table 1. Basesub Dataset

usubjid	paramcd	adtcmt	abfl	ittfl	avalpkfl
STUDY-001-1234567890123	ALB	6/12/2019	Y	Y	4
STUDY-001-1234567890124	ALB	6/13/2019	Y	Y	5
STUDY-001-1234567890124	ALB	6/13/2019	Y	Y	5
STUDY-001-1234567890125	ALB	6/13/2019	Y	Y	5
STUDY-001-1234567890127	HCT	6/15/2019	Y	Y	5
This is a very long comment intended to illustrate how the qcompare macro handles very long strings and special characters					
STUDY-001-1234567890128	HGB	6/16/2019	Y	Y	7
STUDY-001-1234567890129	ALP	6/22/2019	Y	Y	7

The compare dataset in Table 2 has

- Records duplicated on usubjid paramcd adt ID variables with distinct values in the aval and atoxtrn variables in light blue
- Records duplicated on usubjid paramcd adt ID variables with distinct values in the anl01fl and aphase variables in green
- Long comment with a slight difference from the base long comment comment in dark blue

Table 2. Comparesub Dataset

usubjid	paramcd	adtcmt	saffl	pcfl	base	aval	abfl	atoxgrn	atoxtrn	anl01fl	aphasen
STUDY-001-1234567890123	ALB	6/12/2019	Y			5	Y	1			
STUDY-001-1234567890123	ALB	6/12/2019	Y			4	Y	1	2		
STUDY-001-1234567890124	ALB	6/14/2019	Y			4	Y	1	2		
STUDY-001-1234567890125	ALB	6/13/2019	Y			4	Y	1	2		
STUDY-001-1234567890126	ALB	6/13/2019	Y			5	Y	1	2	Y	1
STUDY-001-1234567890126	ALB	6/13/2019	Y			5	Y	1	2	N	2
STUDY-001-1234567890127	HCT	6/15/2019	Y			8	Y	1	2	Y	2
This is a very long comment intended to illustrate how the qcompare macro handles some very long strings and special characters											
STUDY-001-1234567890128	HGB	6/16/2019	Y			88888888	Y	1	2	Y	2

EXAMPLE OUTPUT

Nonprintable characters are highlighted in green along with ascii code in parentheses. For example, (13) represents a carriage return.

INITIAL REPORTS FROM QCOMPARE ARE SIMILAR TO THOSE FROM PROC COMPARE

We see from Tables 3-6 that there is some data structure mismatch on record counts. Also, some variables are not common. The length and label attributes differences will need to be addressed in order to validate the compare dataset. Any label with special characters will be reported.

Table 3. Data structure comparison

Data Set Summary				
Dataset	NObs	Created	Modified	NVar
WORK.BASESUB	7	16JAN26:15:53:51	16JAN26:15:53:51	8
WORK.COMPARESUB	8	16JAN26:15:53:51	16JAN26:15:53:51	13

Table 4. Variable comparison

Variables Summary	
Number of Variables in Common: 6	
Number of Variables in WORK.BASESUB but not in WORK.COMPARESUB: 2	
Number of Variables in WORK.COMPARESUB but not in WORK.BASESUB: 7	
Number of ID Variables: 3	

Table 5. Attribute comparison

Listing of Common Variables with Differing Attributes			
Variable	Attribute	WORK.BASESUB	WORK.COMPARESUB
ABLFL	LABEL	Baseline~Record Flag	ablfl
PARAMCD	LABEL	Parameter Code	Param(13)Code
PARAMCD	LENGTH	100	10000
USUBJID	LABEL	Subjid Identifier	
USUBJID	LENGTH	100	10000

Table 6. Base variables unmatched with compare

Listing of Variables in WORK.BASESUB but not in WORK.COMPARESUB		
Variable Name	Type	Length
ITTFL	Char	1
PKFL	Char	1

Table 7. Compare variables unmatched with base

Listing of Variables in WORK.COMPARESUB but not in WORK.BASESUB		
Variable Name	Type	Length
ANL01FL	Char	1
APHASEN	Num	8
ATOXGRN	Num	8
ATOXTRN	Num	8
BASE	Num	8
PCFL	Char	1
SAFFL	Char	1

Table 8. Any labels with nonprintable characters

Nonprintable or nonstandard ASCII variable labels in WORK.COMPARESUB	
Variable Name	Label
PARAMCD	Param(13)Code

ID DIFFERENCES IN A TABULAR FASHION RATHER THAN A VERBAL LIST

All ID values found to be not matching between base and compare will be reported by qcompare. From Tables 9 and 10, we can see that there are two usubjid-paramcd-adt combinations in basesub and two usubjid-paramcd-adt combinations in comparesub that are not common to the two datasets. Also, it happens that one of the usubjid-paramcd-adt combinations in basesub and one in comparesub are not unique within each of these datasets. A complete listing of all such duplicates within base and compare is described later.

Table 9. ID values only in base

usubjid-paramcd-adt records in WORK.BASESUB and not found in WORK.COMPARESUB					
usubjid	paramcd	adt	Observation Count	Observation Numbers	Duplicate ID Value?
STUDY-001-1234567890124	ALB	13JUN2019	2	2,3	Yes, WARNING
STUDY-001-1234567890129	ALP	22JUN2019	1	7	

Table 10. ID values only in compare

usubjid-paramcd-adt records in WORK.COMPARESUB and not found in WORK.BASESUB					
usubjid	paramcd	adt	Observation Count	Observation Numbers	Duplicate ID Value?
STUDY-001-1234567890124	ALB	14JUN2019	1	3	
STUDY-001-1234567890126	ALB	13JUN2019	2	5,6	Yes, WARNING

The proc compare report in Table 11 states same as qcompare report in Table 10, though it is in a less concise description rather than a simple tabular summary. The STUDY-001-1234567890123/ALB/12JUN2019 is handled a little differently in proc compare than qcompare, as follows.

OBSERVATION MATCHING IN THE PRESENCE OF DUPLICATES WITHIN ID VALUES

Proc compare reports the STUDY-001-1234567890123/ALB/12JUN2019 as both a duplicate and mismatched ID in Table 11. Qcompare reports STUDY-001-1234567890123/ALB/12JUN2019 as only a duplicate and not an ID mismatch. Since base and bcompare datasets both have these IDs in common, for our QC purposes we do not need to report a mismatch. The QC issue in our validation reconciliation context is only to resolve the Compare dataset duplicates, not to resolve any mismatch.

Proc compare is matching base and compare in a one-to-one fashion based on their sorted order within the ID combination, where compare does a sql full join within duplicate.

Table 11. Proc compare ID value mismatch listing

WARNING: The data set WORK.COMPARESUB contains a duplicate observation at observation number 2. NOTE: At observation 2 the current and previous ID values are: usubjid=STUDY-001-1234567890123 paramcd=ALB adt=12JUN2019. NOTE: Further warnings for duplicate observations in this data set will not be printed. Comparison Results for Observations Observation 2 in WORK.COMPARESUB not found in WORK.BASESUB: usubjid=STUDY-001-1234567890123 paramcd=ALB adt=12JUN2019. Observation 2 in WORK.BASESUB not found in WORK.COMPARESUB: usubjid=STUDY-001-1234567890124 paramcd=ALB adt=13JUN2019. WARNING: The data set WORK.BASESUB contains a duplicate observation at observation number 3. NOTE: At observation 3 the current and previous ID values are: usubjid=STUDY-001-1234567890124 paramcd=ALB adt=13JUN2019. NOTE: Further warnings for duplicate observations in this data set will not be printed. Observation 3 in WORK.BASESUB not found in WORK.COMPARESUB: usubjid=STUDY-001-1234567890124 paramcd=ALB adt=13JUN2019. Observation 3 in WORK.COMPARESUB not found in WORK.BASESUB: usubjid=STUDY-001-1234567890124 paramcd=ALB adt=14JUN2019. Observation 5 in WORK.COMPARESUB not found in WORK.BASESUB: usubjid=STUDY-001-1234567890126 paramcd=ALB adt=13JUN2019. Observation 6 in WORK.COMPARESUB not found in WORK.BASESUB: usubjid=STUDY-001-1234567890126 paramcd=ALB adt=13JUN2019. Observation 7 in WORK.BASESUB not found in WORK.COMPARESUB: usubjid=STUDY-001-1234567890129 paramcd=ALP adt=22JUN2019.
--

VALUE COMPARISONS

The observation summary tells us that there are some other variables that mismatch in the cases where the ID combination matches, a summary breaks these down by variable, and a detail report lists the mismatching values. These details reports have different structure depending upon whether the variable is numeric or character.

Table 12. Data value comparison

Observation Summary	
Total Number of Observations Read from WORK.BASESUB:	7
Total Number of Observations Read from WORK.COMPARESUB:	8
Number of Observations with Some Compared Variables Unequal:	4
Number of Observations with All Compared Variables Equal:	0
Number of Observations in Common:	4
Number of Observations in WORK.BASESUB but not in WORK.COMPARESUB:	3

Table 13. Data value comparison summary

Variables with Unequal Values		
Variable	Number of Value Differences	
AVAL	4	
COMMENT	1	

Table 14. Data value comparison details for a numeric variable

Value Comparison Results for Variables						
AVAL						
usubjid	paramcd	adt	AVAL WORK.BASESUB	AVAL WORK.COMPARESUB	diff	pct_diff
STUDY-001-1234567890123	ALB	12JUN2019	4	5	1	25
STUDY-001-1234567890125	ALB	13JUN2019	5	4	-1	-20
STUDY-001-1234567890127	HCT	15JUN2019	5	8	3	60
STUDY-001-1234567890128	HGB	16JUN2019	7	88888888	88888881	1269841157.14285

Table 15. Data value comparison details for a character variable

COMMENT				
usubjid	paramcd	adt	COMMENT WORK.BASESUB	COMMENT WORK.COMPARESUB
STUDY-001-1234567890127	HCT	15JUN2019	This is a very long comment intended to illustrate how (16) the qcompare macro handles very long strings and special characters	This is a very long comment intended to illustrate how (18) the qcompare macro handles some very long strings and special characters

The base dataset has 'illustrate', compare has the misspelling 'ilustrate' in Table 15. The compare dataset also has a 'some' not in base. The qcompare finds maximum matching string of characters working in from each end and uses black font. Starting from first difference going in from each end, the font is red. If we had a very long narrative comment discrepancy, this would help us find the first and last difference in spelling between the base and compare comment values. The unprintable characters, an ASCII code 16 in base and a code 18 in compare, are identified in parenthesis.

Table 16. Proc compare data value comparison details

The COMPARE Procedure

Comparison of WORK.BASESUB with WORK.COMPARESUB

(Method=EXACT)

Value Comparison Results for Variables

usubjid	paramcd	adt	Base Value comment	Compare Value comment
			+	+
STUDY-001-1234567890	HCT	15JUN2019	This is a very long	This is a very long

usubjid	paramcd	adt	Base aval	Compare aval	Diff.	% Diff
STUDY-001-1234567890	ALB	12JUN2019	4.0000	5.0000	1.0000	25.0000
STUDY-001-1234567890	ALB	13JUN2019	5.0000	4.0000	-1.0000	-20.0000
STUDY-001-1234567890	HCT	15JUN2019	5.0000	8.0000	3.0000	60.0000
STUDY-001-1234567890	HGB	16JUN2019	7.0000	88888888	88888881	1.26984E9

In Table 16, the standard proc compare report truncates both usubjid and comment since they are longer than 20 characters. The proc compare output dataset in Table 17, however, does not truncate columns. Even though we have the complete strings in output dataset, the proc compare difference dataset is not easy to read and not ready to report to others when summarizing and communicating differences for further investigation. For very long strings we will still require the qcompare word wrapping feature, as direct viewing of the proc compare output dataset in SAS Studio does not do well with very long strings. Note that the DIF row is not aligning with BASE and COMPARE rows when viewed in SAS Studio, making it a bit difficult to map to the string positions in base and compare. Also note that although proc compare gives us '+' to indicate that the comment column has been truncated, there is no such indication that usubjid column has been truncated. So we might mistakenly assume by absence of the '+' that the usubjid is STUDY-001-1234567890, whereas it is actually 3 characters longer in all cases (STUDY-001-1234567890123, STUDY-001-1234567890125, STUDY-001-1234567890127, STUDY-001-1234567890128).

Table 17. Proc compare data value comparison output dataset

Obs	_TYPE_	_OBS_	comment	usubjid	paramcd	adtl	abfl	aval
1	BASE	1		STUDY-001-1234567890123	ALB	12JUN2019	Y	4
2	COMPARE	1		STUDY-001-1234567890123	ALB	12JUN2019	Y	5
3	DIF	1		STUDY-001-1234567890123	ALB	12JUN2019	.	1
4	COMPARE	2		STUDY-001-1234567890123	ALB	12JUN2019	Y	4
5	BASE	2		STUDY-001-1234567890124	ALB	13JUN2019	Y	5
6	BASE	3		STUDY-001-1234567890124	ALB	13JUN2019	Y	5
7	COMPARE	3		STUDY-001-1234567890124	ALB	14JUN2019	Y	4
8	BASE	4		STUDY-001-1234567890125	ALB	13JUN2019	Y	5
9	COMPARE	4		STUDY-001-1234567890125	ALB	13JUN2019	Y	4
10	DIF	2		STUDY-001-1234567890125	ALB	13JUN2019	.	-1
11	COMPARE	5		STUDY-001-1234567890126	ALB	13JUN2019	Y	5
12	COMPARE	6		STUDY-001-1234567890126	ALB	13JUN2019	Y	5
13	BASE	5	This is a very long comment intended to illustrate how □ the qcompare macro handles very long strings and special characters	STUDY-001-1234567890127	HCT	15JUN2019	Y	5
14	COMPARE	7	This is a very long comment intended to illustrate how □ the qcompare macro handles some very long strings and special characters	STUDY-001-1234567890127	HCT	15JUN2019	Y	8
15	DIF	3	XX.....	STUDY-001-1234567890127	HCT	15JUN2019	.	3
16	BASE	6		STUDY-001-1234567890128	HGB	16JUN2019	Y	7
17	COMPARE	8		STUDY-001-1234567890128	HGB	16JUN2019	Y	88888888
18	DIF	4		STUDY-001-1234567890128	HGB	16JUN2019	.	88888881
19	BASE	7		STUDY-001-1234567890129	ALP	22JUN2019	Y	7

ADDITIONAL UNIVARIATE STATISTICS COMPARISON

Another comparative report that can help spot data issues quickly is the univariate statistics comparison. Some variables are not expected have any missing values, and typically this is specified in adam spec. This is a necessary basic check we do in validation work, this is automated with qcompare. We can also see min and max values that are not consistent between the two datasets. Empty variables are highlighted in red in Table 16, as they usually warrant confirmation.

Table 18. Univariate comparison

Univariate statistics comparison between WORK.BASESUB and WORK.COMPARESUB Sorted by descending pct missing in base dataset																
Variable	WORK.BASESUB								WORK.COMPARESUB							
	Records nonmissing	Records missing	Total records	% Missing	Min	Max	Median	Mean	Records nonmissing	Records missing	Total records	% Missing	Min	Max	Median	Mean
adt	7	0	7	0.00%	21712	21722	21713	21714.857	8	0	8	0.00%	21712	21716	21713	21713.5
aval	7	0	7	0.00%	4	7	5	5.4285714	8	0	8	0.00%	4	88888888	5	11111115
aphasen	-	-	-	-	-	-	-	-	4	4	8	50.00%	1	2	2	1.75
atoxgm	-	-	-	-	-	-	-	-	8	0	8	0.00%	1	1	1	1
atoxtrn	-	-	-	-	-	-	-	-	7	1	8	12.50%	2	2	2	2
base	-	-	-	-	-	-	-	-	0	8	8	100.0%	-	-	-	-

IDGROUPS ADD FUNCTIONALITY

If a variable list is in the idgroups argument, the macro will compare distinct values of that variable list in base with distinct values of that variable in compare dataset. This secondary comparison is common in validation work, to see if for example entire patients or entire paramcds are not common to base and compare. Also we may want to see if entire usubjid-paramcd combinations are not common to base and compare. This is done for each variable groups delimited by |. We would want to do this if the overall obs count is not equal between base and compare, to see if a simple explanation in terms of entire subjects, paramcds or usubjid-paramcds can be reported. The focus for further investigation is then is just this one subject value and not the entire list of mismatching records corresponding to this subject treated independently and individually and requiring tracing each of these many records.

Table 19. Usubjid distinct value comparison

usubjid not common to WORK.BASESUB and WORK.COMPARESUB		
usubjid	Value in WORK.BASESUB Only	Value in WORK.COMPARESUB Only
STUDY-001-1234567890126	Y	
STUDY-001-1234567890129		Y

Table 19 tells us that the entire usubjid=STUDY-001-1234567890129 set of records is only in the COMPARE dataset and the entire usubjid =STUDY-001-1234567890126 set of records is only in the BASE dataset

Table 20. Paramcd distinct value comparison

paramcd not common to WORK.BASESUB and WORK.COMPARESUB		
paramcd	Value in WORK.BASESUB Only	Value in WORK.COMPARESUB Only
ALP		Y

Table 20 demonstrates that the entire paramcd=ALP set of records is only in the compare dataset. The entire HCT parameter is absent in base.

Table 21. Paramcd distinct value comparison

usubjid-paramcd not common to WORK.BASESUB and WORK.COMPARESUB			
usubjid	paramcd	Value in WORK.BASESUB Only	Value in WORK.COMPARESUB Only
STUDY-001-1234567890126	ALB	Y	
STUDY-001-1234567890129	ALP		Y

In Table 21 we see that subject STUDY-001-1234567890126 has missing ALB in compare and STUDY-001-1234567890129 has missing ALP in base. In this case, it could be that QC and source both need to address these findings to resolve the issues by changing programming on both sides..

DUPLICATES ANALYSIS

Per adam specifications, suppose we require only one record per subject per paramcd per date. We would like to report this finding to those clinical data management, to see if these can be resolved at raw data level, or else to the statistician, to see if we can reduce the 2 records to 1 record by taking mean or max or min values (for example). We must resolve the duplicate somehow in order to have a valid ADaM dataset to use for further analysis.

The following listings summarize the situation, counting how many and which variables distinguish the duplicated usubjid-paramcd-dates records. Table 22 tells us that the two records in BASE dataset are exact copies for STUDY-001-01234567890124/ALB/13JUN2019.

Table 22. Duplicate ID value summary in base dataset

Duplicate usubjid-paramcd-adt groups in WORK.BASESUB dataset - Summary - WARNING					
usubjid	paramcd	adt	Distinguishing Variables	Observation Numbers	Observation Count
STUDY-001-1234567890124	ALB	13JUN2019	**Exact Copies**	2,3	2

Table 23 tells us that the compare data set has duplicated the usubjid-paramcd-adt ID column values across two records in two cases, and that the entire records were not exact copies, and which variables distinguished values within the ID. One pair of records was on STUDY-001-1234567890123/ALB/12JUN2019 and the other pair was on The STUDY-001-1234567890126/ALB/13JUN2019.

Table 23. Duplicate ID values summary in compare dataset

Duplicate usubjid-paramcd-adt groups in WORK.COMPARESUB dataset - Summary - WARNING					
usubjid	paramcd	adt	Distinguishing Variables	Observation Numbers	Observation Count
STUDY-001-1234567890123	ALB	12JUN2019	aval	1,2	2
STUDY-001-1234567890126	ALB	13JUN2019	anl01fl, aphasen	5,6	2

Further details on distinct values of distinguishing variables are provided in Table 24. We can see that the STUDY-001-1234567890123/ALB/12JUN2019 duplicates were only distinguished from one another by the aval column. One of the two STUDY-001-1234567890123/ALB/12JUN2019 records had aval = 5 and the other one had aval = 4. The STUDY-001-1234567890126/ALB/13JUN2019 pair of duplicated records could only be distinguished from one another by anl01fl and aphasen. One of the two STUDY-001-1234567890126/ALB/13JUN2019 records had an anl01fl = 'Y' with an aphasen = 1, and the other one had an anl01fl = 'N' with an aphasen = 2. This information is very helpful in pinpointing what has to be done to reconcile and resolve these duplicates. Especially helpful when dataset has hundreds of variables all of which duplicate across some records except for one or two variables.

Table 24. Duplicate ID value details in compare dataset

Duplicate usubjid-paramcd-adt groups in WORK.COMPARESUB dataset - Details					
usubjid	paramcd	adt	Variable Distinguishing the Duplicate	Record Number within Duplicate	Value Distinguishing the Duplicate
STUDY-001-1234567890123	ALB	12JUN2019	aval	1	5
			aval	2	4
STUDY-001-1234567890126	ALB	13JUN2019	anl01fl	1	Y
			anl01fl	2	N
			aphasen	1	1
			aphasen	2	2

SUMMARY OF BASE VS COMPARE MATCH

The final results are reported in html listing as an overall match Y/N such as Table 25. The same can be output in a one record dataset.

Table 25. Overall match report where qcompare determines base is not matching compare

Final results							
Qcompare:WORK.BASESUB WORK.COMPARESUB match?	date	time	Maximum difference tolerance	Maximum percent difference tolerance	Maximum difference observed	Maximum percent difference observed	
N	16JAN26	21:53	.0000001	N/A	88888881	12698412	

In case of a match, confirming proc compare results are also reported by qcompare. To determine whether proc compare matches, the listing file is scanned for certain strings. Equivalently, user can also choose to reference the sysinfo to see whether it is 0 to confirm base v compare match.

Table 26. Overall match report where qcompare determines base is matching compare

Final results								
Qcompare:WORK.BASESUB WORK.COMPARESUB match?	date	time	Maximum difference tolerance	Maximum percent difference tolerance	Maximum difference observed	Maximum percent difference observed	Proc Compare:WORK.BASESUB WORK.COMPARESUB match?	sysinfo
Y	16JAN26	21:53	.0000001	N/A	0	0	Y	0

CONCLUSION

The qcompare macro streamlines the data set comparisons for the specific purposes of validation of datasets in clinical trials settings. It generates an easy-to-read tabular html output. In addition to the standard proc compare reports that list discrepancies in dataset structure and dataset attributes ID levels and value differences, it can compare very long character strings and can identify nonprintable characters. Extra reports are provided for comparison of distinct values of alternative ID groups and for duplicate ID records. The qcompare macro does not intend to replace proc compare, only to extend it for early stages of the validation in cases where proc compare is reporting a variety of differences that are hard to understand and summarize.

REFERENCES

- [1] SAS Institute Inc. (2023). *SAS® 9.4 Procedures Guide: Statistical Procedures, PROC COMPARE*. Cary, NC: SAS Institute Inc.
- [2] Auld, M. (2016). "Go Compare: Flagging up some underused options in PROC COMPARE." *PharmaSUG*. Paper BB04.
- [3] Smith, J (2006) "When is New Really New? Proc Compare Gives the Details" NESUG 2006 Paper DA28
- [4] Anusuiya, G and Kumar, R (2015) 'Proc Compare: Wonderful Procedure!' PharmaSug 2015 Paper IB11

ACKNOWLEDGEMENT

The author would like to thank his management and colleagues who provided valuable input and comments on this paper.

CONTACT INFORMATION

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

Author Name: Chris Weiner
Company: Bristol Myers Squibb
Address: 1 Squibb Drive, New Brunswick NJ 08901
Work Phone: (512) 552-0352
Email: chris.weiner@bms.com
Website: <https://www.linkedin.com/in/chris-weiner/>

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.

Brand and product names are trademarks of their respective companies.