

PhUSE 2014

Paper PP15

QC:manual vs program – a personal view

Helen Nicholson, Cmed, Horsham, UK

ABSTRACT

When producing any output we need to know that it is right. It does not matter if the output is to be used by another department, or are the final tables, figures and listings for the clinical study report. Unless the output is produced from a fully automated system, the outputs need checking in some form every time they are produced to ensure that they are correct. Normally this will involve a method of programming which will re-produce the data from the output in some form which can then be compared against the original. This poster will look at some of the most common ways of doing this comparison, either by eye or programmatically and present the pros and cons of each method.

INTRODUCTION

Whenever something is produced, no matter what it is, dataset to be passed on to someone else, table figure or listing (TFL) for the final report, or ad-hoc output used within the department or group, you need to know that is right. This is done by checking what has been produced so that it can be used with confidence knowing that it is of a high enough quality to be used – in other words it goes through quality checking (QC).

There are lots of different ways to QC what is produced and there are lots of different things to check. You need to check that the content is correct, that it contains the information that was expected, for example if you are looking at adverse events it is no good producing a demographics table. If a shell or template was provided then you need to check that what is produced matched the shell or template – often within given conditions.

One of the biggest questions is how do you do the checking – do you do it by hand (or eye) or do you do it programmatically. In reality the answer is normally a combination of both, it is just a question of how much of each, and that may depend on what it is that is being QCed.

Who does the QC is another question and not one that is discussed here. This paper assumes that the output on question has been checked by the original programmer and it is an independent person doing the QC.

TABLES, FIGURES and LISTINGS

Manual checking

Even if you think you are doing the QC programmatically the truth is there are always going to be some manual checking – it is just a question of how much. You always need to check that the output looks 'right', that it is actually doing the job it was meant to be doing and that nothing has appeared that you were not expecting.

The output needs to be compared to the original shell, template or specification to check that it matches, that the titles and footnotes say the right thing and are in the right place. This is normally done by looking at what has been produced and what was expected, and checking that it matches.

This could be done programmatically but the time taken to read in the shell then read in the output and check the two match is unlikely to be quicker than just looking. And if one of the titles or footnotes contain the run date/time then you would need to take this into consideration when checking them programmatically. Also in my experience a lot of shells or template displays have a standard way of putting in the table or listing number that is not always the same format that is wanted on the final output. There are often extra titles and footnotes that are not displayed in the shells because they are the same across all the outputs for the study.

One of the checks that you will do, normally without thinking about it, is to check the spelling of everything on the output. If there are spelling mistakes in the data then there is not a lot you can do about it, but if it is in either a column or row heading then you get a chance to change it before it goes out.

You need to check the contents of the TFL. If for example it is a demographic table you may want to check that the number of females in each treatment group is right. So either looking at a listing or the dataset you count how many females there are in each group and hope it matches what is produced in the table. Unless it is a very small amount of data this is very time consuming and prone to errors. However in the past it has been known for

three or four people to be entering values into a calculator that one person is reading out and checking that everyone go the same number at the end. This stage is now largely done programmatically.

If your output goes over more than one page then you need to check that the page breaks fall in a sensible place, ideally you don't want just one odd line at the bottom of a page and the rest of that section on the next, also you don't want just one line on a separate page.

Depending on how the final output is going to look you need to check that all the columns are right or left aligned, that decimal places lines up and that there are the right number of decimal places for the data. If a percentage is being displayed do you have the % sign next to the data or in the column or row header? Is it obvious how the percentage is calculated, for example if it is a number followed by a percentage is it obvious what the denominator is?

Are all the numbers present – are there any blank spaces where you are expecting something – does a blank space make sense or should it be a zero? And are there places where it should really be a blank space rather than a 0 – for example if you have some simple statistics of n, mean, standard deviation and the range and n is 0 then it should be blank spaces for the rest rather than 0. Or are any numbers too large for the space given so they are shown as asterisks?

If you have text in the output is it all present – has any been truncated because the space is not wide enough? If the text has to wrap does it always wrap at a sensible place?

Consistency

One final check that is normally done manually is a consistency check across the outputs. This is not only checking that if one table has x number of subjects in the safety population that all the outputs have the same number of subjects in the safety population. You also need to know that the outputs look consistent so that a united package is produced. This includes, but is not limited to, checking that if on one table the % sign is in the column header, then it is in all other columns that contain percentages, that is columns are left aligned in one table they are left aligned in all the outputs, that the capitalisation and any formatting of the column or row headers is consistent and that any indentation across tables is consistent and reasonable.

Proof of checking

Once you have done you checking you need to prove that you have done it. This is normally done by ticking and signing the output. The tick may mean different things on different tables. Does one tick on a page mean you have checked the whole page or should you tick each number you have checked? Do you have to check every number on the page or can you just check certain columns or rows? There are all things that have to be considered – how much checking and ticking may well depend on what you are checking and any standard operating procedures or working practices that are in place.

.

Electronic checking

In order to speed things up you may do some checks electronically or programmatically. The amount that is done this way will depend on how you are doing your QC. One way of doing the QC electronically is to reprogram the output either fully or partially. You then need to know how you are going to compare what you have produced with the original output.

Dual program the data but not the layout

When you first start dual programming, unless you are following established procedures, you normally aim just to reproduce the numbers or text of TFL, but make no attempt to match the layout of the TFL. You then have to manually check that the numbers match what was produced. So you still have to do some form of manual checking.

These dual programs can be quick to write as often most of the original program is in the layout to produce the final output. The trouble with doing this is that if there is a large number of items to check and it might take time, and unless time was taken in the QC program it is often not in the same order as what is being checked. And although the original dual programmer may know the order that the output is in for anyone else they have to sit and work it out.

Another issue is that not all the information may be programmed for speed. So although the numbers may be present often the percentage are not and the QCer has to work them out by hand.

To actually compare the output to the original is a manual process so you have all the same issues. This means that you have a lot of the same issues that you have when you do purely manual checking, how much do you check, and how do you indicate what has been checked. You still have to sign the output to indicate that it has been checked, keeping the original copy and also the output that you have checked the output against.

Dual program the data and the layout

The next logical step is to re write the QC program so that it reproduces the TFL exactly including the layout. The QC output then needs to be compared to the original which can be done manually but could also be done electronically. There are several products on the market that allow two files to be compared, allowing you to select that type of differences that are shown.

As both the production programmer and the QC programmer are using the same shells the same layout should be able to be produced. Unless you have very detailed shells that include the width of all the columns, there may need to be some back and forth between the two programmers to get identical looking output. Both programmers need to make sure that they are wrapping text, either column or row headers or within the body of the TFL, at exactly the same point. If a column has a percentage included then agreement need to be reached as to how many spaces between the number and the percentage.

Doing it this way all title and footnotes would be included on both output which can bring its own problems. If there is a date time stamp on the output then unless both are run at exactly the same time, then if you are using a package to compare the outputs you are going to get differences that you may need to explain – unless you can tell the package to ignore certain things.

When you actually run the compare between the original and the production there is normally a way of saving electronically the differences so there is no need to print out anything to prove there are no differences, however what you save and print will depend on what is required by your organisation.

Dual program data with programmed compare of the output

One way around the issue of date time stamps in the title and or footnotes is to just compare the content of the table, including column and row headings but excluding the titles and footnotes. This can be done by producing an output dataset of the TFL and then reproducing that dataset as part of the QC program and using PROC COMAPRE within SAS to check that the two datasets are identical. Each column across the page becomes a variable and each row down the page becomes an observation.

As this means that the variables that are being compared have to be the same it means that there has to be either some standard naming conventions in place for the columns or details have to be given on the shell. Although this can be implemented after the production program has been written, with just the addition of the output to a permanent dataset added in afterwards it is easiest if it is as done as part of the original programming.

One of the easiest ways to produce an output dataset is to add an OUT= to the PROC REPORT in the production program. Then you know that the output dataset contains the same information that is in the report. This means that the variables in the output dataset are the ones in the COLUMN statement plus a `_break_` variable that give the breakpoints as specified by any BREAK statements. You also do not have to worry about labels – they are what are given in any DEFINE statements.

The QC program would not have to match the layout of the PROC REPORT but it would have to match the column headers as these are the labels for the variables. Using standard names for the columns makes things a lot easier, so for example if there are five columns on the output the COL1 to COL5 would make sense. If it is a listing and you are putting out the accrual variables then you might not need to rename them – it will all depend on the rules you have.

When doing the QC this way there are a few things you have to take into consideration. There may be some nonprinting columns that are used to get the sort order correct, these have to be in the QC dataset as well. You have to decide if you are going to match the `_BREAK_` variables, and it can take a long time to get the blank spaces right in the columns. One way round the last issue is to compress the variables to remove the spaces – you are after all interested in the data that is in there not necessarily the number of spaces before or after the data.

Proof o checking

As with doing manual QC you need to know how you are going to prove that you have done the QC and what you need to keep to prove it.

If you have dual programmed the data only then still need to do the manual compare of the output against the original. The accepted norm is that you tick the numbers you check and then sign and date the output to indicate you have done the checking. This could be at the same time the titles et are checked.

If you have dual programmed the layout as well and used a package to compare the two outputs then this normally produces an output report that you can print of and sign. You can normally keep the report electronically which many mean that you don't need to print it off.

If you have dual programmed and then used PROC COMAPRE on the datasets you could print of the compare report, sign and date it, or again you can save the report electronically so avoiding the need to print and sign. If you put the TFL number and other details in title statements for you RPOC COMAPRE you can include the username so it is obvious who actually run the final compare.

Where the compare has been done electronically and the results stored electronically then you normally need to sign something to say you have done that. This could be part of a tracking sheet allowing you to sign against each output at the time of delivery, or maybe just one signature to say that you have actually done what the tracking sheet says you have. The exact details of what you sign and when will depend on your standard operating procedures or working practises.

Other Options

Over the years other methods of QC have been tried for different things. One that I have used, although a good few years ago, is to read the listings back into SAS and produce summary tables on the information that is in them to compare against any tables that have been produced. This only works if you know that the listings are right, in my case it was when data management used the listings to check that the data had been entered correctly.

I should perhaps add that this was done in the time before PROC REPORT, when the output file was created with a FILE PRINT statement in a DATA _NULL_. This is not a method I would recommend as it can take a lot of code to read in the data into the correct format.

Manual vs Electronic

Whether you go for a fully manual QC or a full or partially electronic QC will depend on a number of factors. A manual QC, where you tick and sign the original output, can be very quick for small TFL's but can be very time consuming for large outputs. And if what you are producing has to be re done at a late date then the full QC has to be done again

There does not seem to be a way round doing some manual checking – the table matches the shell and does what it is meant to. You should always page through an output to check that there is nothing that is obviously wrong – asterisks where there should be number because the width is too narrow, data wrapping at strange points or truncating. These things might not show up in an RPOC COMPARE or other electronic comparison, as they tend to be report specific.

Where you dual program data only the programs can be quick and easy to do, after all often a lot of time is spent in getting the output to look right, however for long tables this can again be very time consuming and it is very easy to make a mistake when checking long column of data.

When you dual program including the layout and then use a package to compare the original and the QC output you still need to do the manual checks mentioned above.. It can be very time consuming getting the layout to match exactly and you need to allow for the different run date time stamps. You still need to look though the results to check it is only the date time differences.

If you are producing an output dataset for the TFL and then dual coding that and comparing the two it can be very easy to get so involved in getting the dataset to match that you forget to look at what you are actually producing. You need to remember to always look at what you are producing and do a quick mental check – does it make sense. It can also take a lot of work to get the columns to match. However once it has been done once, if you need to re-run the production code (with no changes) then it is very quick to re QC the output.

DATASETS

Manual checking

Although it may not seem obvious there are still manual checks that you can and should do on a dataset. You still need to look at the dataset produced and check that it makes sense. If the dataset contains some decodes of coded variables then you need to check that you have got them the right way round – it is very easy to get the yes/no decodes the wrong way round – and sometime the specifications do not make it clear which one you should use. You need to check that you are using the right decode list – this is important if you are using a format to produce the value. You should not assume that the format you used last time is the right one to use for this study – there may have an extra option this time round that you did not have last time.

You need to check that you have the correct number of observations present. Have you combined the data in the correct manner, if you have used a PROC SQL have you checked that you are not adding in spurious or duplicate observations because the data is not in the form that you were expecting. A lot of this you would think would be checked by the dual coding, but if mistakes can be made once (say in production code) what is to stop you making the same mistake in the QC code. It might be that the specifications are not detailed enough and need clarification.

You need to be able to check the formats for the variables, the length and the labels of the variables. When looking at the labels you need to check that they make sense, you don't want a label saying that the variable is the Body mass index or BMI when if you look at the calculation it is really the body surface area. This may seem a bit extreme but how often is the calculation for a variable changed without thinking about the label.

You should look at the content of the datasets – if when looking at it you feel that something is not right or looks odd then it probably needs looking at just to confirm that it is OK. If one of the variables is almost always empty is there a reason for it or has something gone wrong in the programming that has blanked it out for some reason.

Electronic checking

Most people now dual program datasets to make sure that they are correct, this is done by an independent programmer following the same specifications that the production programmer used. The normal process is that the QC program does not produce a permanent dataset but at the end uses PROC COMPARE to compare the production and the QC dataset. However it is possible for the dual program to produce a permanent dataset and the comparison is actually done by a third programmer who just compare the two datasets and decides which one (if either) is right.

The ideal is to get a 100% match on the datasets with that line saying no differences found. That is both datasets have the same number of observations, the same number of variables and the contents of the variables/observations match. For the variables it is not just names but also the length, format, informat and label. Any dataset label should also match.

One of the advantages of dual programming the datasets completely is if after everything has been taken into consideration it is felt that the production dataset is wrong and the QC dataset is right then the code can be swapped over, with the production becoming the QC code and the QC code becoming the production. How you do this would depend on you organisation, and in some places it would not be allowed.

It is possible to keep the dual programmed or QC datasets as permanent dataset and program the TFL QC from these so effectively have two stream of programming going on at the same time with them only coming together at points to make sure they match.

Some of the most common problems with dual programming the datasets is the specifications are not clear enough or do not go into enough detail about some of the calculations. If there is any ambiguity in the specification you can be sure that the two programmers will read it two different ways and so get two different results. Also a specification can be written without any thought to the practicalities of how something is going to be produced. It can be common for the same variable to be in more than one dataset with slightly different definitions in each.

Manual vs Electronic

When checking a dataset it is a lot easier to dual code it rather than check it by hand. One reason for this is that when you get new data then you can just re run the QC program and provided that you have coded for all possibly data choices, and some of the more common data issues (like partial dates) can QC the dataset in seconds rather than hours.

You should still look at the dataset to make sure that it looks reasonable, there are not a lot of duplicated observation present (unless there is a good reason for them), that no observation or variable contains a lot of missing information that cannot be explained.

If you use proc compare then you can store the return code from the compare in a dataset that also has the dataset name in it and the name of the person doing the compare. This means that you can see the status of all the datasets that have had a compare run on them simply by opening one dataset. A very quick way of knowing the status of your study QC.

As when doing the tables it is very easy to get so assume that the production is right and to spend a long time getting the QC dataset to match the production one. This is something that the QC programmer should be aware of and stop themselves if they find they are doing that and take a step back and decide which is right.

If there a complicated calculations in the dataset then it can be difficult to make then match exactly, it will depend on when and where the rounding was done. Ideally this will be detailed in the specification but is it not then this may be one source of differences. The two numbers may look

identical but if you put them both out to a high number of decimal places then you may find they differ in the 18th decimal place. The other trick is to put them out in HEX format – sometimes it is easier to spot the difference then. This is a particular problem if the production is done on one platform and the QC on another,

CONCLUSION – which one for me

When it comes to TFL's I have experienced a great many variations in what is checked and how it is checked. There always needs to be some form of manual checking – and there should always be that last manual sanity check before it goes out. For small one-off outputs I think the quickest way is to dual program the content but not the layout, and then manually check the QC output against the production output.

For anything else my preferred way is, currently, to produce an output dataset and dual code that. This is made a lot easier if you have standards for the variable names, for example column1 for the first column, column2 for the second etc., and the variables are compressed, or left aligned, so that you do not have to match on white space.

For datasets I think they have to be dual programmed and then compared. My preference is to get 100% match although when I only differ on three points in a dataset with 50+ variables and 150 thousand records I have been known to question if it matters that if we differ on less than 0.005% of the data!

And if you make sure the name of whoever is running the QC is recorder in the compare when it is done together with the date and the result is stored electronically you have your proof. Adding information to a tracking sheet makes it easier to track and gives a place to sign if required