Next Generation Edit-Checks

PEER REVIEWED

Classic Data Cleaning (Based on Classic Edit-Checks)

Once upon a time things were rather simple: during a clinical trial, all the data was collected on paper and three (or even four) carbon copies were made. Data was then transferred to a database, including all investigators' errors (thanks to the double data entry method) and then turned over to the data manager. On this same database a control software was then run, which contained all the edit-checks detailed in the data management plan.

The result of this process was a long list of 'inconsistencies' pointing to data that were invalid. Data managers relied on this list to transfer specific queries to monitors who in turn sent the queries over to investigators.

Investigators responded to the queries, and the data manager was then able to make the necessary corrections. This was the final stage of the data cleaning, which was the last step prior to the statistical analysis of the data.

This scheme was simple but not very efficient, especially if you compare it to what can be accomplised with e-CRFs.

The first real problem was that an investigator could write whatever he/she chose (including their shopping list!) on the original paper form (and on carbon copies, of course). So, you needed really a lot of edit-checks to offset the consequences of this extreme freedom.

However, an electronic CRF's structure is much more rigorous: if a field is designed for numbers, you can't type letters or words in it. Furthermore, all numbers can be validated as soon as they are inserted, and they will be rejected if they are not in the specified range.

The second problem with the old system was that the cleaning stage occurs long after the investigators have inserted the data; therefore data cleaning does not offer any help with data insertion. It is obvious that you need a lot more "manpower" hours to clean a database without any input data rules than one where these rules are embedded in the system itself.

'Nice' Edit Checks

One more consideration before briefy analysing the options available today: the use of e-CRF by non-technical people and in particular by clinicians. As you may have experienced, there is software that is 'friendly' or easy to use, and makes you feel very comfortable, and then there is 'hostile' software that is not user-friendly. For example, we have all encountered nightmare forms on the internet that require large amounts of repetitive information to be inputted and then do not allow us to save because a field is missing or incomplete. At this point, the human reaction is to throw the PC out the window; however, being a civilised human being, one does not give in to the impulse but just closes the PC and curses the programmer.

Although investigators are usually compensated for the time required to enter data, their compliance depends in large part on their comfort level with the system. It is not sufficient to simply have proper data controls, but also the controls must act in a synergistic manner with the investigator. The controls need to be helpful without being tedious or cumbersome.

Advanced Data Cleaning (and the New Edit-Check Menagerie)

Today, the compact army of old time edit-checks can be subdivided into specific categories with rather different characteristics and uses.

Here are some of them.

Dates in the Future

According to an old saying, "Hindsight is 20/20". In an e-CRF it should never be possible to insert a date in the future. The obvious reason is that one is gathering facts and not hypotheses. The only certain data points are the ones that have already occurred. If an investigator inputs a future date, the system rejects it and highlights the error. It is a simple control that saves a lot of oversight and recognises only one exception: the necessity to register in the e-CRF future appointments.

Date Sequence

It is evident that the dates of past visits should be in sequential order based on time. It is inconceivable that the second visit would take place before the first visit. Less obvious is what happens when a investigator is inserting a date for the second visit which is earlier than the date of the first visit. A system that automatically rejects the second date as wrong will be perceived as stupid and hostile. Why? Because the investigator might have inserted a wrong date for the first visit! The system should thus accept the new date and just highlight the inconsistency with the date already present. This gives the investigator an opportunty to fix the error and he/she will be grateful for the help.

Usually visits occur at a preset intervals (e.g. five weeks). In this case, a good system should use a colour-coded alert: after a date is entered, colours can pop up, such as green for a correct interval, yellow for a date that is outside the interval range but still within acceptable limits, and red for visit intervals that are outside acceptable limits.

In the case of preset visits, the system will also indicate when successive visits should occur using a 'neutral' colour (e.g. white).

An SMS reminder can then be sent to investigators and/or patients about upcoming visits.

In-page Alerts

When it is necessary to make sure that two data points on the same form are consistent (e.g. verifying that the systolic pressure is higher than the diastolic pressure, as it should always be), then the correct control method is a

direct in-page alert. An alert message indicating the error should pop up and ask that the discrepancy be corrected before saving to the database. Naturally, no data points should be removed from a partially completed form.

An investigator should be able to correct just the errant data point without having to reenter any other information and finally send the data set to the database. Controls of this type have a limit because they are performed on the investigator's computer/device. Often computers malfunction or have viruses that can corrupt this mechanism, so an error check should be repeated on the central database server (using validation and authentication techniques) so that only trustworthy systems (as central servers) should have the final word on the data accepted. By the way: all systems inputting data on the web should be viewed as potentially untrustworthy systems or devices.

Mandatory Fields

One way of guaranteeing that essential data points are complete is to make them 'mandatory'. This does not mean that the system will reject the entire data set if one mandatory data field is left empty (this would be the nightmare form we were talking about earlier). The system would not allow the completion and the progression to the next form. This will prompt the investigator to provide essential information (even at a later stage, or whenever it's available) without restricting or blocking the entire data entry.

Dependent Fields
Many data inconsistencies are tied to questions or data points that are interdependent. For example, a patient is designated as never having been a smoker, but in the field requiring the number of cigarettes smoked per day, there is a number other than zero. This type of incoherence is avoided by simply placing a hierarchy of questions that will allow a subsequent field to appear only if the answer in the previous field requires it. If a patient answers yes to being a smoker, then the question of "how many cigarettes per day" will appear. Otherwise, this field will simply not appear and will never be given a value.

Automatic Control of Unit Measure and Plausibility Control
A very tedious part of filling in forms is the necessity for a clinician to specify every unit of measure and ranges of normality for every lab result value. A smart e-CRF should allow investigators to fill in these units and ranges only once and for all patients being seen at that research site.
Not only is this convenient, but it is indispensable if the system is to verify firstly that the values inputted are plausible and secondly whether they fall in the 'normal' range for the specific test.

Another major benefit is in regard to the data manager and statistician: they will have to act only on 'normalised' values and they will save countless hours of work verifying and checking the validity of laboratory test values by hand. According to GCP standards, there will be a table that reports how the data was entered, but there will also be a table with normalised data for analysis.

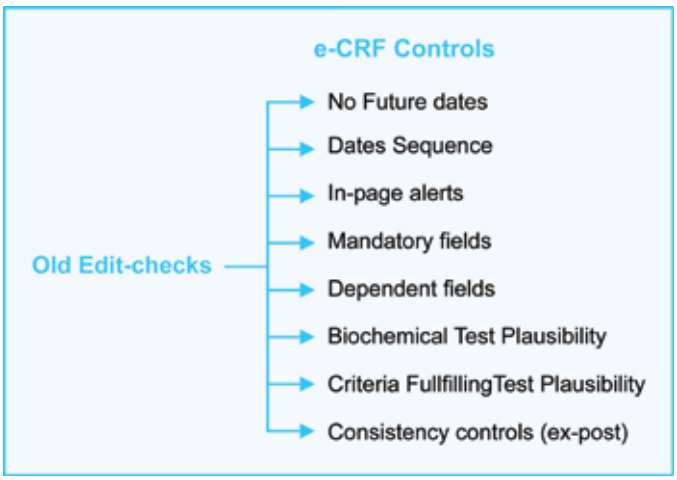
Is the Patient Fulfilling Inclusion/Exclusion Criteria?
Even if GCP requires that a clinician assumes complete responsibility for respecting inclusion and exclusion criteria, a quality e-CRF can and should help him/her in this task. How? Simply by providing a screening form that precedes the inclusion and exclusion criteria forms.

In this screening form, there should be key questions correlated to inclusion or exclusion, for example the value for creatinine that would indicate renal impairment, thus excluding the patient from the study. If a value is entered that is incompatible with the relevant criteria, the investigator is advised of the incompatibility. If the value is confirmed, then the patient will not be enrolled in the study and will be flagged as a criteria violator.

Another example of the system automation is the patient's birthdate. When this date is inserted, if the patient's age exceeds the age limit for entering the study, the system notifies the investigator and he/she can act accordingly.

Consistency Controls (Ex-post)
When inconstistencies result from data points inserted in different forms or at different times, and none of the previously listed strategies can be used, then post-completion coherency controls come into play (they are very much like the traditional edit-checks).

These controls, that are specifically designed for every study, can be carried out at any time by the study monitors



or data managers, and not only after the database lock. The results are summed up in a special chart and are extremely valuable when preparing monitoring visits or for generating queries.

Conclusions
Since total control is still utopia, the role of the monitor and data manager still remains fundamental. There is a great deal that can be done to at least guarantee that monitors and data managers have to deal only with a limited range of errors.

During a visit to a site, the real-time control system allows the monitors to dedicate their time to problem-solving rather than to problem-finding (as happened with paper CRFs or poor designed e-CRFs). This obviously makes on-site visits a lot more efficient. This may also lead to less frequent visits.

Andrea DeRosa is the founder of Airon Telematica (Air-Tel). He has over 10 years of experience in eCRF design and optimisation. After graduating with a medical degree, he held positions as a Medical Manager at Roche, Italy and at Roussel Uclaf from 1981 to 1989. In the 1990s, as a founding member of the medical marketing and communication agency, Airon Communication, he began to take advantage of his expertise in information technology to support companies in their transition from traditional to web-based medical information. In 2000, he founded Air-Tel, a company dedicated to supplying eCRF SaaS (Software as a Service) for sponsored and non-profit clinical research.
Since 2009, he has been a Lecturer on the topic of e-CRF on the Postgraduate Master's degree programme in Clinical Research and Drug Development at Bicocca University, Milan. Email a.derosa@air-tel.it