**Template for a Data Cleaning SOP**

< This template is to be used to create a study specific guideline or Standard Operating Procedure (SOP) for data cleaning. It can apply to the entire study/trial or to one analysis/database lock in a larger study/trial. >

< This document provides an overview of the main things to consider when writing such a guideline and/or when planning your data cleaning >

< Information provided like this (between < and > and purple text) are guidance provided to the author of the document and should be actioned and then deleted. >

< This document provides an outline and description of required sections, but should be modified based on the specific section. Sections that are not applicable can be deleted. Note that this list is not comprehensive as each study/trial will have its own study-specific needs regarding key data and essential data checks to be performed. >

< When we refer to database lock, we refer to the finalisation of a dataset for a specific analysis. The study could be continuing and the database may still be updated, but a final “locked” version of the database would exist for a specific analysis or interim analysis. >

|  |  |
| --- | --- |
| Author |  |
| Reviewer |  |
| Version |  |
| Effective date |  |

# General information

It is important to ensure the accuracy of collected study data, which necessitate cleaning the data that has been entered in the database prior to any data analysis. This document describes the process of data cleaning for < provide study name >.

# Aim

The aim of this document is to outline key considerations involved in data cleaning, which is the process of data checking to enable detection of discrepancies and generation of data queries, and appropriate processes to follow when applying corrections or exceptions

# Responsibilities

< Describe who will be responsible for what aspects of data cleaning>

< Studies should generally have a data manager or someone who is responsible for overseeing the data collection and management and this section will describe this person’s responsibilities >

It is the responsibility of the data manager (or person responsible for data management tasks) to:

* Ensure data are entered into a database with an audit trail facility, such as REDCap
* Incorporate data validation checks into the design of the database. These are data entry rules for each field, such as:
  + Ensuring the primary identifier is unique
  + Allowing only legal values in categorical fields
  + Including range checks for continuous fields
  + Setting fields to be required, where appropriate
  + Utilising branching logic to hide fields when not applicable
  + Maintaining appropriate separation between research data and data that is identifying of research participants

More complex data checks, for example, checking the sequential ordering of dates, may be incorporated into the study database or may be performed in a statistical package after data have been entered. This may be programmed by the data manager or data management team or may be programmed by the study statistician.

It is the responsibility of < specify the person – probably the data manager > to ensure that all data checks as described in this guidance are completed.

It is the responsibility of the Principal Investigator to review the data checking SOP with the study statistician to ensure that all important data checks are included in this SOP.

# Data sources

< It is important to make it clear where all of the electronic data for the study will be stored. Ideally all data would be in a single study specific database (for example in REDCap). If some data (e.g. laboratory data) comes from elsewhere this should be detailed >

< List all data sources: For example (delete those not applicable) >

* Participant data entered into REDCap
* Laboratory data in laboratory data management systems

< Also specify what fields are collected from what data sources >

< Refer to the project’s Data Management Plan >

# Software and record keeping/file management

< Within the data cleaning SOP, it is important to provide details for the data checking and cleaning process. Here we provide some pointers for how this should be conducted: >

**Data checking**

* A statistical package (like Stata) is recommended for conducting data checking. This must be done via a command file (e.g. Stata do-file) to ensure a documented record of the checks and manipulations performed is saved.
* Datasets and command files associated with data checking should be clearly named and saved in the appropriate folder for the study. Refer to the Good Stata Programming SOP (https://doi.org/10.25374/MCRI.14741058)
* All consistency checks run in the statistical command file should be preceded by a comment within the command file clearly stating exactly what is being checked.
* Some queries, missing data or data errors may never get resolved (see Section 7 below). These must be noted within the query report (Section 8) and data checking/preparation command file. Data checking commands should be modified so that known unresolvable data queries are recorded as exceptions and not listed as open queries. They should also be noted within the analysis files and, if necessary, for completeness of information, also in the analysis report. This will avoid queries being raised repeatedly that will never be resolved.
* Always preserve the data exported from the database in its raw form so it is always possible to re-run a script and obtain the same output.

**Data correction**

* All changes to data must be recorded in an audit trail, electronic, or otherwise.
* Data corrections should be made in the study database via the study’s query process (cf. Sections 7 and 8). All data queries and their resolutions should be documented. If the data are stored in an appropriate database, such as REDCap, the edits will be recorded in the audit trail. Capturing reason for change in the audit trail is recommended (e.g. using REDCap’s “Reason for change” function).
* Data corrections should not be made in the command file of the statistical package unless identified following database lock. In this scenario, data corrections must be documented with a detailed comment explaining why each change has been made, who requested the change and the date of the request or change. The requests must be documented, e.g. via an email from the project manager.

# Prepare data for checking

* Remove any test or otherwise redundant records (e.g. test records or records captured in error if agreed with research team).
* Ensure variable names, labels and value labels are consistent with sections 11 below
* Ensure variables are in the correct format (e.g. check that numeric variables are read into your statistical package as numeric variables, date variables as dates, datetime variables as datetimes) and convert as required
* If dates have been read into your statistical package as string variables, convert them to date variables.
* A copy of the data will need to be exported for data checking. Variables containing participant-identifying data (e.g. name, address, phone number) must not be exported from the database. Check that no identifying variables were accidentally exported from the database, remove variables containing identifying information that are found. Pay particular attention to the contents of any free text fields.

# Data checking

< In this section we list some of the checks that you may wish to conduct on your data. There may be other, study-specific checks that are required. You can delete checks that are not applicable to your data >

## Integrity of dataset(s)

* Confirm that there are no duplicate records.
* Confirm that the number of records and participants are as expected.
* Confirm records have a unique identifier (primary key). This may be a single variable, such as patient ID. Alternatively, it may be a combination of two or more variables, e.g. Patient ID and visit number, or Patient ID and hospital admission date.
* Confirm that all participants meet the eligibility criteria (discuss discrepancies with the research team).
* If combining datasets from multiple sources, cross-check basic demographics where possible to confirm that record IDs correspond across the various sources (e.g. check that the date of birth and sex match in the various datasets for each ID).

## Identify data queries (when data are entered by study staff)

Query codes are codes to be entered into a field in the database when there is uncertainty about the value, e.g. if is it currently unknown and being followed up with the site, or when there is a missing value that is never going to be recovered. Query codes should be defined by the study team/data manager. The values of query codes are determined by the data field type, e.g. a number for numeric fields or a date for date fields, and must not be a plausible value for that field. Examples are -1 for missing data that are never going to be recovered and -2 for currently unknown values that are being investigated for positive, numeric variables, and 01/01/1000 = missing data that are never going to be recovered and 02/02/2000 = currently unknown values that are being investigated for date fields. These query codes must be resolved before data cleaning is finalised and the database is locked (where applicable) (aside from those representing data that are never going to be recovered). Non-resolvable query codes should be documented as such and replaced with missing data indicators.

## Identify missing data

< Ideally the database should be set up in a manner that would automatically raise a query if required data are missing. >

* If branching logic is used in the data collection instrument, it is important to incorporate this in checks to distinguish between records with no data for a given variable that (1) should have data and (2) should not have data as the variable was meant to be skipped (e.g. pregnancy questions skipped for a male participant). Only the former should be considered missing data.
* For data entered by staff from a paper form: Where a value is missing on a paper form, data entry staff should have entered the missing code for that field to confirm the value is truly missing. At the data checking stage, check for blanks in the exported data. The only blank values should be those resulting from fields that are hidden by branching logic at the data entry stage. Any other blanks should be listed in the data queries as these may be fields that have been skipped during data entry.
* Studies should have a process in place for identifying any missing case report forms (CRFs)/ data forms or missed visits, based on the visit schedule for a participant as outlined in the study protocol.

## Identify inconsistent data

* Check whether data that are present should have been collected. For example, lab values are marked as not done but then lab values are recorded.
* It may be necessary to review string variables to ensure that responses are intelligible and relevant, i.e. that the response makes sense in the context (rather than simply whether or not a response exists).

< This can be simplified during the design phase by using a drop down list of responses/radio button if only a small number of responses are expected. >

* When reviewing string variables, look for and remove any identifying information included in text responses. A common example is inclusion of people’s names (e.g. of participants, children, medical practitioners, friends, etc.)
* Check whether data with branching logic is consistent, for example if the answer to Question 7 is yes, then Question 8 should be answered, but if the answer to Question 7 is no, then Question 8 should not be answered.
* Check the calculations of any derived variables where possible. Check that these calculations are dealt with as expected where there are missing data.
* Compare related variables to identify any impossible data. For example:
  + Calculate maternal age at child's date of birth to check if mother's age is feasible.
  + Produce a scatter plot of height against weight to identify potentially implausible pairs of values
* Check date order and durations
  + Dates of events that occurred prior to study enrolment (e.g. date of birth, dates associated with medical history) must be before the date of the first visit
  + All visit dates are between the date of study enrolment and the participant's last date in the study (or the date of data download)
  + Date should be in the correct order, i.e. date of birth < date of 1st visit< date of 2nd visit, or date of birth < date baby was introduced to solid food
  + Check that ages and time intervals/durations lie within the expected range

## Identify illegal data (categorical variables)

* Confirm that categorical variables only contain the values that they are supposed to.

< Ideally the database would be designed to only allow legal values.>

## Identify unlikely data (continuous variables)

* Confirm that data from continuous variables are within expected ranges. This includes:
  + Range checks (e.g. height and weight between values appropriate for age, date of birth consistent with inclusion criteria, etc.).   
    < Ideally the range limits would be included within the study database hence may not need to be checked.>
  + Outlier checks (plotting continuous variables and confirming that any outliers are genuine and not the result of errors).
* Where the same variable is measured at several time points for each participant, it is valuable to plot each person’s sequence of recorded values to ensure that they behave reasonably. If there are many participants in your study, this may be impractical. An alternative is to write code identifying any obvious errors, e.g. child’s height has decreased over time (allowing for measurement error).
* If only participants with a baseline record should have data at later time points, check this is the case.
* Check that all applicable continuous data have a unit recorded and convert data captured in the wrong unit (for example birth weight in g if it should have been captured in kg). Check that units are consistent throughout the data.

## Randomisation code for randomised trials

For randomised trials, check the randomisation code in the study database against the original randomisation schedule and possibly the pharmacy records to ensure that the randomisation code has been merged in correctly. This check should only be done after the database has been locked and all data has been cleaned ready for analysis.

< If possible, have a second person review the data checking command file. >

# Data query reports

< Data queries arising from the data checking process may be compiled into reports which are sent to the data manager and/or study sites for resolution. These are usually known as Data Clarification Forms (DCFs), but may also be known as query reports or error reports. >

< Alternatively, queries can be raised directly within the database (e.g. using REDCap’s “Data Resolution Workflow”) and assigned to the relevant person for investigation. >

< Each study has a specific procedure to raise and document queries. The below is one suggestion. Please adapt based on your specific study >

At a minimum, the query reports should include the following for each query raised:

• Trial/Study Number

• Participant ID

• Date of visit and/or data form/CRF name or number

• The question number and/or text

• Brief description of the error

• Space for authorised delegate to respond to the query and then sign and date the updated response (either signing off each response or per page of responses).

A log should be kept of when each query report was sent out e.g. via a query report tracker. It is also recommended that a copy of each query report sent be kept by the data manager.

# Discrepancy management and related record-keeping

< Discrepancies refer to the data errors and inconsistencies identified as described in Section 7 >

* The general steps to follow to resolve discrepancies are as follows (in order):
  1. Check discrepant values against the source data (the original, raw data, e.g. a paper survey or handwritten clinic notes). Any data entry errors should be corrected in the database.
  2. If the discrepant value is not a data entry error, cross-check it against any available alternative data sources (e.g. check date of birth recorded in a second data collection instrument or allergy status derived from clinic observations). This may help inform how to treat the discrepancy. Similarly, any ‘notes’ variables for physical assessment and clinic visits may contain additional information that can help. Again, any data entry errors should be corrected in the database.
  3. If the value is still discrepant and cannot be resolved, a data query code should be attached to the data point. Best practice is to list these as data discrepancies in the final study report.
* The process for discrepancy management should be discussed with, and agreed on, by the research team and included in the data cleaning guidance.
* Ideally any changes to values following data cleaning should be made in the study database and the audit trail function should be used to record the reasons for the changes. If changes are made as part of the analysis code, keep a written record of any changes that have been made in the cleaning command file or other documentation. For example, “DOB was changed to 02/02/2021 as per email with X date 14/12/2021”

# Prepare data for analysis

< Once the data have been cleaned, the database should be locked and exported for analysis >

Always preserve the data exported from the database in its raw form without any data manipulations. It must always be possible to return to the original dataset, re-run the preparation and analysis script and reproduce results. Once a cleaned dataset has been exported and securely stored, begin the process of generating an analysis dataset.

## 10.1 Convert missing values to system missing value codes

* Convert any values in the source dataset that are used to indicate missing data to the missing value codes used in the statistical software (e.g., . .a, .b, etc in Stata.). The values should be converted to system missing value codes so that they do not interfere with subsequent checks and later analyses.

## 10.2 Creating derived variables as required

* Derive variables required for analysis in line with the definitions/algorithms stated in the study protocol or statistical analysis plan.
* It is recommended that new variables constructed during cleaning are identified as such, for example, by appending ‘\_d’ to the variable name to indicate that is derived from the raw data.

# Variable coding and value labelling

* Be consistent with variable coding within and across studies (where applicable). This may require recoding in some situations.
* Yes/No variables should be coded as 0=No 1=Yes.
* Ensure each value is appropriately labelled with text that clearly describes the responses, ideally using the same wording as the questionnaire.

# Check the final analysis dataset

* Check that in the final dataset to be used for analysis, all variables (including variables created during data preparation) have appropriate names, labels and value labels
* Ensure the data dictionary is up to date (including data notes for any constructed variables)