# OCA-DSI data management

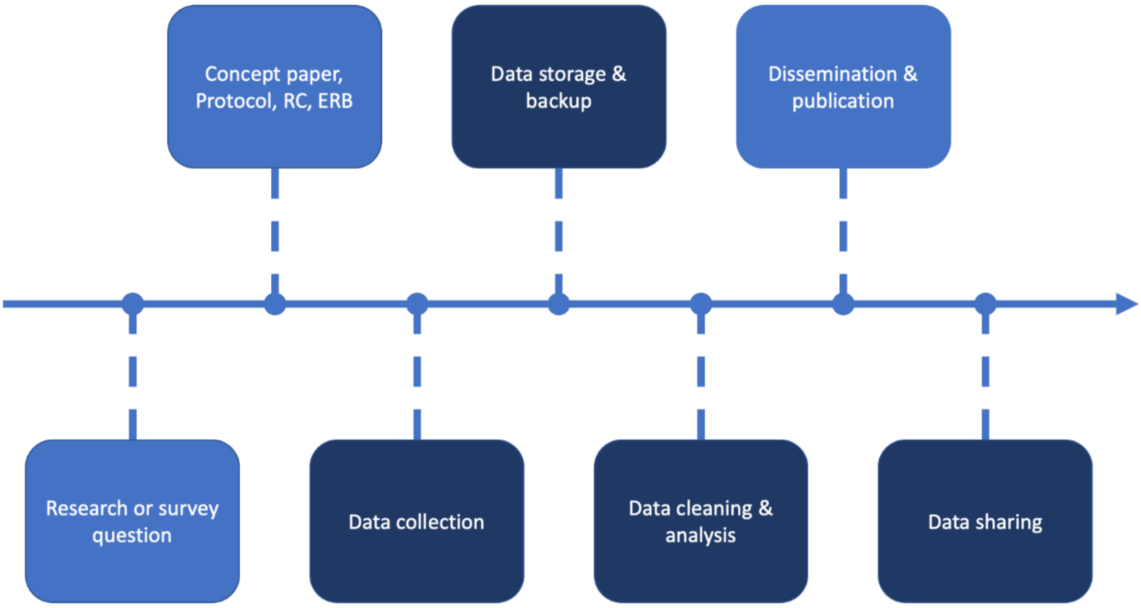
Knowledge page for CKAN platform

## Introduction

This page guides you through the different stages of data management in a study or survey: from collection to sharing. The aim is to help you manage your data efficiently while meeting OCA standards for the eventual sharing of your dataset on the Data Sharing Platform.

## Overview

Data management is at the heart of the wider data journey and consists of four stages highlighted in dark blue in the chart below:


Concept paper, Protocol, RC, ERB
Data storage & backup
Dissemination & publication
Research or survey question
Data collection
Data cleaning & analysis
Data sharing

## Stage 1: Data collection

In this first step, you will define the data to be collected to answer your research or survey question, and the data collection tool you will use. The best approach is to first define the dictionary of variables and then integrate it into an appropriate data collection tool.
A minimal data dictionary consists of a list of variables and, for each variable, the options it can take. For instance, the variable Gender can take options Male, Female and Unknown/unspecified.

Among the many existing data collection tools, OCA recommends to use Kobo Toolbox for surveys and REDCap for research studies.
Kobo Toolbox allows for the design of simple electronic forms that can be completed using a smartphone or tablet, with the interviewer seeing one question per screen. It is best suited to one-off questionnaires, for participants who are only interviewed once.
REDCap allows for the design of more complex electronic forms to be completed using a computer or tablet, as the interviewer sees the entire form on the screen. This system is best suited to studies with several questionnaires, for example longitudinal studies where participants are visited several times.
In practice, it is recommended to contact your Epi Advisor to check which tool would be most appropriate for your study. Other data collection tools (Excel, EpiData, etc.) can be used as a last resort. For instance, if your research study requires MSF Electronic Medical Records, the data can be exported in Excel format. In this case, make sure that you also export or create an appropriate data dictionary.

OCA has developed standardized Kobo templates for retrospective mortality, malnutrition, and vaccine coverage surveys. Those templates are written in Excel and follow the XLSForm standard, which includes a data dictionary. If you are considering conducting one of these surveys, please contact your Epi Advisor. You can also find more details on the standardized surveys on the fieldresearch website.

There are no REDCap templates for research studies at MSF OCA yet, but you can find several examples and training courses online. We recommend the REDCap Training Series from the Institute of Translational Health Sciences.

## Stage 2: Data storage and backup

In terms of project organisation, we recommend keeping all data and scripts related to a given project in a dedicated directory with a descriptive name and a README file. This *one folder = one project* philosophy makes the project self-contained and portable, i.e. it can be moved from one computer to another without losing parts of the scripts or data.
This is handy for making a quick manual backup of your project: copy and paste the project folder into a separate backup directory, adding the date of the backup to the folder name.
For more guidelines we recommend reading the "Good practices guidelines in R projects" developed by Epicentre. For instance, it contains good practice for naming and organising project sub-folders.

There are 3 places where OCA projects can be stored: i) MSF OCA Sharepoint; ii) your personal MSF OneDrive and iii) an encrypted MSF OCA laptop. Note that if you use REDCap or Kobo Toolbox, the data collected on the electronic device will be sent to and stored on another dedicated server (contact your Epi Advisor for more information). This data can then be downloaded and stored in your project folder for further analysis.

According to the OCA guidelines (see English, French and Spanish versions), your project folder must be stored on the MSF OCA Sharepoint site of your Mission. To do this, you must request access from the site manager. Storing your project on Sharepoint has several advantages:

- You can synchronise the project between the MSF Sharepoint server and your MSF laptop using OneDrive. When you work locally, OneDrive synchronises your project in real-time with the Sharepoint server.
- The MSF Sharepoint server acts as a backup: if your laptop is broken, you can re-synchronise your project on another laptop. MSF Sharepoint servers are also automatically backed up, so you don't have to worry about duplicating the data stored on the Sharepoint server.
- Deleted items are stored in the Recycle Bin folder of the Sharepoint site for 30 days and can be restored with one click. So, if you have deleted a file by mistake you can easily recover it. See this link for more details about the Sharepoint Recycle Bin.
- Sharepoint allows Mission MedCo to finely control who can access each item and with what type of access (write, read only, etc.).
- Unlike your personal MSF OneDrive folder which will be deleted when you leave OCA, the project will remain stored in Sharepoint.

However, there are 2 situations where you might need to store your project outside the Mission Sharepoint:

- If you cannot get access to the OCA Mission Sharepoint, you should store your project in your personal MSF OneDrive and transfer it to the OCA Mission Sharepoint as soon as you have access.
- If your internet connection is limited or non-existent, as is often the case in the field, your project on your laptop will not be synchronised with the Sharepoint server in real time. This can be problematic if you lose your laptop as your project in Sharepoint may be out of date. It is therefore important to ensure that you have a regular local backup of your project. OCA guidance is that no external drives should be used to save or back up data. Neither USB nor hard drives. In case there is no internet (i.e. no One Drive, no Sharepoint, etc.), backups should be done on the MSF laptop of the study implementer (which is encrypted) and on the MSF laptop of another colleague. The transfer between the two laptops can use a USB or external hard drive, but the data must be deleted immediately after the transfer.

If you are a first time user of OneDrive or Sharepoint, we recommend you check out the Introduction page on Microsoft website as well as Epicentre tips (in English and French) on managing synchronisation in low bandwidth environments.

### Advanced use-case: version control of code files

Version control systems are software tools that help the study implementer to manage changes to their files containing code (R, Python, Stata, etc.) by tracking and managing changes over time. They are useful to work collaboratively on a code file, to easily revert changes that you regret or to implement something risky that might break your code.

The most widely used version control system is probably Git, but it takes some time to learn (don't do it the day before you leave for the field). Fortunately, there are several good tutorials for getting familiar with Git, see for example this chapter in the Epi-R handbook.

Once you are using Git to track your code files, you can also synchronise with a Git server so that your code and change history are stored online (e.g., for collaborative work). MSF does not have a dedicated Git server but many MSF projects use Github, which is free and user-friendly (see the Epi-R Handbook chapter for an introduction to Github).

Three important things to remember when you use Git/Github:

- Only track code files with Git. Do not track data files, otherwise they will be uploaded in Github, which would be against OCA's guidelines on data storage. Also, avoid tracking binary files (.doc, .xlsx) as Git cannot handle changes in these files.
- Do not track code files that might contain personal information, for instance if your code file contains comments with such information.
- You should place your Git folder in your project directory but remove the One Drive synchronisation for this folder as One Drive does not work very well with Git.

## Stage 3: Data cleaning and analysis

### Data cleaning

Data cleaning poses two main challenges: it takes up a significant amount of your project's time and, if not done well, it can generate errors that can impact on data analysis. It is therefore important to learn how to be efficient while respecting some principles to ensure data integrity.

To be efficient, we recommend you use data validation in the data collection tool to limit data entry errors. For example, you can set minimum and maximum values to avoid outliers. There are several online resources about data validation for Kobo Toolbox and REDCap.

Once data is collected, there can still be errors/inconsistencies and we recommend you run a data cleaning script on your data to check and detect errors or inconsistencies. OCA guidelines recommend that you use the R programming language to write your script. If you are more familiar with Stata, there are dedicated tutorials to help you switch to R (see for instance Chapter 4 of The Epidemiologist R Handbook).

One possible approach is to write one R function per check and run them regularly on your data export to detect errors (don't wait until the end of data collection to start cleaning).

Once you have detected an error there are two options:

- If the data collection tool has an audit trail feature (e.g., REDCap) you can manually edit and correct the electronic record. A full log of the changes (what, when, by who, etc) will be automatically recorded by the Audit Trail. This approach can be time consuming but is recommended when data errors need to be assessed one by one for quality purposes, as in the case of clinical trials.

- If the data collection tool does not have an audit trail feature (e.g., Kobo Toolbox), then instead of modifying the raw data it is preferable to write a script that will either perform cleaning automatically (e.g., replace 3021 by 2021 in all dates with such an error) or will merge an external file that contains the corrected data (e.g., data that requires substantial manual cleaning).

In both cases, it is important to document the errors found and how they were resolved. If data collection is still ongoing, consider adding a new data validation to the data collection tool if this will prevent the error from recurring.

Key principles to ensure data integrity is reproducibility and transparency:

- Reproducibility: never modify raw data (except with Audit Trail), ensure your script works as a pipeline, get clean data as an output
- Transparency: audit trail, documentation of what has been done and why.

You can find more details and practical examples of data cleaning in the Data Management section of the Epidemiologist R Handbook.

### Data analysis

Data analysis should follow the same principles as data cleaning, except that instead of check and corrections, the R script should contain analysis and visualisation functions.

For data analysis it is convenient to use R-Markdown, which enables you to write documents that are fully reproducible. R-Markdown documents mix narrative text and code to produce elegantly formatted output that can be static (e.g., PDF, Word) or dynamics (e.g., HTML).

The advantage of using R and R-Markdown for data analysis is that you can generate tables and figures automatically. This way, if the data change due to data cleaning or additional data collection, the analysis can be updated accordingly. For small analyses, do everything in an R-Markdown. For more complex analyses (large data set, processing time), you can have an R script to analyse and save the results and then an R-Markdown to write your report and insert/visualise the results.

If a data quality issue is detected, add the correction to the data cleaning script, not the data analysis script, and regenerate a clean, updated dataset.

More resources in the Epidemiologist R Handbook and R4Epi projects.

## Stage 4: Data sharing

### Data dictionary

Documenting the variables in a dataset is a crucial part of data management and ensures that a dataset is interpretable by researchers who were not directly involved in study design or data collection — or even those who were involved but are returning to a dataset after some time away.

A data dictionary is a spreadsheet that contains as many rows as there are variables in your dataset and as many columns (fields) as are needed to describe your variables. In order to standardize data dictionaries across all OCA projects, we have defined a minimum set of fields that must be filled in for each variable in your dataset (see table below). Use the exact field names in the table below as column headers in your data dictionary to ensure that the data dictionary passes automated checks built into the data sharing platform.

| **Required field** | **Description** | **Example entry** |
|---|---|---|
| variable_name | Variable name (i.e. exact column name within the corresponding dataset). Variable names must start with a letter, contain only letters, numbers, and/or underscores, and be unique within a dataset | sample_type |
| short_label | Short phrase describing the variable in words | Type of laboratory sample collected |
| type | Variable type. Options:<br>- Numeric<br>- Date<br>- Time<br>- Datetime<br>- Coded list<br>- Free text | Coded list |
| choices | The list of options (pairs of codes and labels) corresponding to a variable of type "Coded list". For the sake of completeness consider entering all options available during data collection, even if only a subset of those options actually appears in the final dataset. | 1, Blood \| 2, Nasal swab \| 3, Throat swab \| 4, Other |
| origin | Was the variable a part of the original data collection instrument, or was it derived after initial data collection (i.e. derived from one or more original variables). Options:<br>- original<br>- derived | original |
| status | Is the variable shared or withheld to reduce disclosure risk. Options:<br>- shared<br>- withheld | shared |

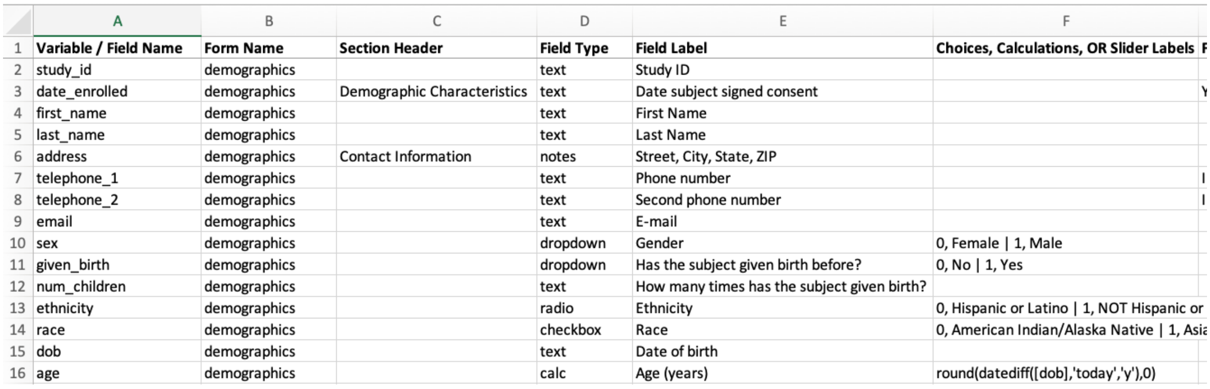
A data dictionary may contain additional columns that you feel are relevant, including fields taken directly from the original data collection tool like REDCap or Kobo/ODK. Some potential examples are included in the table below.

| **Optional field** | **Description** |
|---|---|
| form (or instrument, group, section) | Can be used when variables are separated across multiple forms/instruments (as in REDCap) or sections/groups (as in REDCap or Kobo) |
| branching_logic (or relevance) | Indicates conditional relationships among variables (e.g., variable job_sector is only collected when variable employed is "Yes") |
| required | Yes/No or TRUE/FALSE indicator of whether each variable was required or optional during data collection |
| constraint (or validation) | Numerical or other type of constrain during data collection on the possible values for a given variable (e.g., minimum/maximum allowed values) |

To facilitate the preparation of data dictionaries, we have developed an R package called datadict. The package includes three functions to prepare an OCA-style data dictionary template:

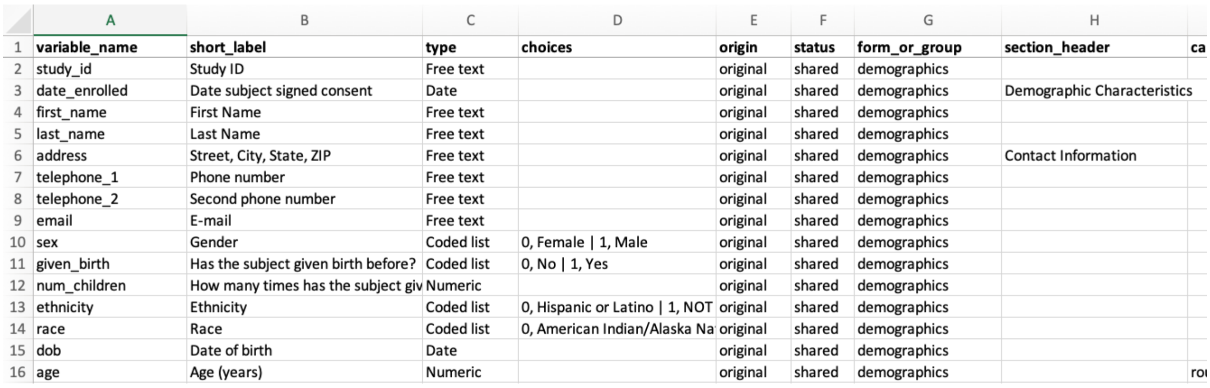
- dict_from_odk(): prepare a dictionary template from an Kobo/ODK dictionary
- dict_from_redcap(): prepare a dictionary template from a REDCap dictionary
- dict_from_data(): prepare a dictionary template from a raw dataset

For example, if your study data was collected using REDCap then you can export the original data dictionary from REDCap as a .csv file, which might look like this:



|  | A | B | C | D | E | F |  |
|---|---|---|---|---|---|---|---|
| 1 | **Variable / Field Name** | **Form Name** | **Section Header** | **Field Type** | **Field Label** | **Choices, Calculations, OR Slider Labels** | **F** |
| 2 | study_id | demographics |  | text | Study ID |  |  |
| 3 | date_enrolled | demographics | Demographic Characteristics | text | Date subject signed consent |  | Y |
| 4 | first_name | demographics |  | text | First Name |  |  |
| 5 | last_name | demographics |  | text | Last Name |  |  |
| 6 | address | demographics | Contact Information | notes | Street, City, State, ZIP |  |  |
| 7 | telephone_1 | demographics |  | text | Phone number |  | I |
| 8 | telephone_2 | demographics |  | text | Second phone number |  | I |
| 9 | email | demographics |  | text | E-mail |  |  |
| 10 | sex | demographics |  | dropdown | Gender | 0, Female \| 1, Male |  |
| 11 | given_birth | demographics |  | dropdown | Has the subject given birth before? | 0, No \| 1, Yes |  |
| 12 | num_children | demographics |  | text | How many times has the subject given birth? |  |  |
| 13 | ethnicity | demographics |  | radio | Ethnicity | 0, Hispanic or Latino \| 1, NOT Hispanic or |  |
| 14 | race | demographics |  | checkbox | Race | 0, American Indian/Alaska Native \| 1, Asia |  |
| 15 | dob | demographics |  | text | Date of birth |  |  |
| 16 | age | demographics |  | calc | Age (years) | round(datediff([dob],'today','y'),0) |  |

Read the .csv file into R as a data frame and use the dict_from_redcap() function to produce an OCA-style dictionary template, which will look like this:



|  | A | B | C | D | E | F | G | H |  |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **variable_name** | **short_label** | **type** | **choices** | **origin** | **status** | **form_or_group** | **section_header** | **ca** |
| 2 | study_id | Study ID | Free text |  | original | shared | demographics |  |  |
| 3 | date_enrolled | Date subject signed consent | Date |  | original | shared | demographics | Demographic Characteristics |  |
| 4 | first_name | First Name | Free text |  | original | shared | demographics |  |  |
| 5 | last_name | Last Name | Free text |  | original | shared | demographics |  |  |
| 6 | address | Street, City, State, ZIP | Free text |  | original | shared | demographics | Contact Information |  |
| 7 | telephone_1 | Phone number | Free text |  | original | shared | demographics |  |  |
| 8 | telephone_2 | Second phone number | Free text |  | original | shared | demographics |  |  |
| 9 | email | E-mail | Free text |  | original | shared | demographics |  |  |
| 10 | sex | Gender | Coded list | 0, Female \| 1, Male | original | shared | demographics |  |  |
| 11 | given_birth | Has the subject given birth before? | Coded list | 0, No \| 1, Yes | original | shared | demographics |  |  |
| 12 | num_children | How many times has the subject giv | Numeric |  | original | shared | demographics |  |  |
| 13 | ethnicity | Ethnicity | Coded list | 0, Hispanic or Latino \| 1, NOT | original | shared | demographics |  |  |
| 14 | race | Race | Coded list | 0, American Indian/Alaska Na | original | shared | demographics |  |  |
| 15 | dob | Date of birth | Date |  | original | shared | demographics |  |  |
| 16 | age | Age (years) | Numeric |  | original | shared | demographics |  | ro |

Note that this data dictionary template might require additional manual editing. For example, if there are additional "derived" variables that you plan to share that were not part of the original data collection, these will need to be manually added to the data dictionary as additional rows. Or, if some of the collected variables will not be shared for the sake of pseudonymisation (see next section), you will need to change the relevant value of the **status** column from "shared" to "withheld".

## Pseudonymisation

Prior to sharing a dataset, it is important to consider whether doing so could result in the disclosure of personally identifiable information about the subjects of the study, and take appropriate measures (i.e., pseudonymisation) as necessary. Here are some key concepts and terms to understand:

*Disclosure*
When a person or organization recognizes or learns something that they did not already know about another identifiable person or organization through released data.

*Disclosure scenarios*

- spontaneous recognition (i.e., someone with knowledge of the sampled population recognizes a unique or particular combination of data values)
- record matching/linkage with other existing datasets (e.g., population registers, electoral rolls, data from specialized firms)

*Sensitive variables*
Variables that contain confidential information that should not be released without suitable methods to limit re-identification risk (e.g., income, religion, political affiliation, health-related variables). Degree of sensitivity may depend on country and context.

*Direct identifiers*
Variables that unambiguously reveal a person's identity (e.g., name, passport number, phone number, physical address, email address). These should always be withheld prior to sharing data.

*Indirect identifiers*

Variables containing information that, when combined with other variables, could lead to re-identification (e.g., sex, age, marital status, occupation). Re-identification risk generally relates to the uniqueness of value combinations *in the population that the study was sampling from*. Note the potential for elevated identifiability risk from extreme (i.e. uncommon) values of continuous variables (height, income, number of children, land area).

*k-anonymity*
A measure of re-identification risk for discrete variables. *k* = the number of records in a dataset containing a certain combination of indirect identifiers (e.g., how many records have *sex* = 'female' and *age_group* = '40-49 years' ?). A higher value of *k* means lower re-identification risk because higher *k* means more records in the dataset with the same combination of indirect identifiers.

The purpose of pseudonymization is to transform a dataset to achieve an "acceptable level" of disclosure risk. The first step is to remove any direct identifiers, like name, address, phone number, etc. Direct identifiers are almost never used in analyses anyway, so this step is straightforward.

The next step is more challenging. We need to consider which variables in our dataset could be indirect identifiers, evaluate the dataset for uncommon combinations of those indirect identifiers, and, if there are combinations that we assess to be "too uncommon" (which corresponds to a relatively high re-identification risk for those individuals), take appropriate steps to transform our dataset as necessary.

For example, imagine that our dataset contains only one record with the following combination of indirect identifiers:

- sex = "Female"
- age_group = "60-69 years"
- occupation = "Doctor"

We have a few different options for how to proceed:

- Suppress the record (i.e. the entire row)
- Suppress one of the relevant variables (i.e. entire column)
- Aggregate levels of one of the variables (e.g., aggregate age groups >60 to "60+")
- Shuffle the values of one of the variables (e.g., shuffle some of the values of occupation among study subjects)

The challenge here is that, unlike with direct identifiers, which are rarely useful for analysis, indirect identifiers are often very useful for analysis. So suppressing, aggregating, or shuffling them to limit disclosure risk may also limit the utility of our shared dataset for analysis. There's an inherent tradeoff between protecting the anonymity of our study subjects and retaining the utility of our shared dataset.

Here is what a typical pseudonymisation workflow will look like:

1. Select a threshold value of *k*-anonymity that will be the minimum acceptable value for combinations of indirect identifiers within the released dataset (e.g., *k* = 5).
2. Assess re-identification risk of each variable (e.g., classify each variable in your dataset as either a direct identifier, indirect identifier, or non-identifying).
3. Assess the utility of each variable for analysis (e.g., high, low, uncertain).
4. Withhold any variables classified as direct identifiers (e.g., names, phone numbers).
5. Merge groups of related indirect identifiers, where possible. E.g., If the dataset contains two age-related variables *age_in_years* and *age_in_months*, merge these two variables into a new derived variable *age*, and withhold the original variables.
6. Review all unique values of "free text" type variables to ensure they do not contain identifying details. Aggregate or withhold as necessary.
7. Discretize any indirect identifiers that are continuous variables (e.g., height in cm -> discrete height categories).
8. Assess re-identification risk criterion (e.g., k-anonymity) using all indirect identifiers.
9. Pseudonymize indirect identifiers to limit re-identification risk (e.g., aggregate, withhold).
10. Repeat steps 8 and 9 until the given risk criterion is met.
11. Ensure that the final pseudonymized dataset and dictionary meet all data-sharing requirements.

Evaluating and limiting disclosure risk is not a process that can be easily automated. It requires careful consideration of each dataset, including the study methodology, and the location and context in which the data was collected.

You can find more information about pseudonymisation in the online guide Statistical Disclosure Control for Microdata.