Table of Contents

A step-by-step introduction to coding and analysis of community insights using a coding framework

This guideline is meant to provide you with all the information needed to start coding and analysing the feedback received from communities. Don't be intimidated by the number of steps presented here – once you start practising these processes, you will soon realise how intuitive it is. Both the feedback collection form and coding framework already exist. To build confidence, it is strongly recommended to practice coding – this could be done individually or as a team. If you have any questions not covered in this document, don't hesitate to contact your IFRC CEA focal points listed at the end of this guide. The last chapter is also your space to add questions and solutions so others can learn from your experience.

What you need to code your community feedback data

1. the Microsoft Office 365 package. Specifically, Excel. We recommend the latest version of Excel to avoid any compatibility issues.
2. Two Kobo accounts: one account for data the feedback manager/administrator, one kobo account data collectors. If you don't have a Kobo account yet, follow this short guide on how to create one.
3. It is important to have at least one designated focal point for managing the community feedback data. It is strongly recommended that the National Society appoints an additional designated focal point for sensitive feedback.

Before you start reading this guide, please make sure you have the following files downloaded:

Part 1: Understanding the importance of the process

Importance of collecting community feedback

Community feedback is any insight shared by community members and can include any type of information, such as questions, suggestions, observations, beliefs, perceptions, concerns, complaints, and statements of thanks. These insights are a crucial aspect of humanitarian work, and we want to ensure the voices of those we aim to serve are heard and considered in decision-making processes. To learn more about the importance of community insights, please refer to ***the IFRC Feedback Kit, Module 1 Community feedback - why bother?***

Importance of coding community feedback

Some feedback can be responded to or acted on immediately during interactions with community members, others might need further follow-ups. In both cases, we want to make sure that all feedback is recorded and analysed. Coding or categorising community insights is vital for systematic analysis and facilitating understanding of what communities think and feel. By systematically organising feedback into themes, we make the data manageable and understandable. Coding ensures that:

- All feedback is valued! Each insight is different even if the story behind it may seem similar. Coding ensures that all information is investigated and not only responded to but also further analysed.

- Data becomes actionable as coded feedback is then ready to be analysed which allows us to see patterns, identify needs, and make informed decisions.

Importance of analysing community feedback

The final step — analysis — transforms coded feedback into actionable insights. Through this process, we can:

- Identify trends and patterns, spot common issues, concerns, or suggestions across different demographics and geographical locations, guiding strategic decisions and policy development.
- Measure impact and assess the effectiveness of interventions and initiatives - identifying areas of success and those requiring further attention - by tracking changes in community feedback over time.
- Prioritise actions by allocating resources and adjusting strategies based on the most critical community needs.
- Enhance engagement and develop more effective ways to engage with communities by understanding their preferences, concerns, and feedback mechanisms.

- Facilitate reporting and communication by streamlining the creation of reports and updates for both internal and external stakeholders, highlighting key findings and actions taken.
- Strengthen advocacy by using evidence-based insights from feedback analysis to advocate for community needs, influence policy, and engage donors.

Part 2: Managing the feedback form

This chapter is meant for feedback/data managers who are responsible for the upload and maintenance of the feedback form, as well as granting access to the form, and data export. Volunteers/data collectors should refer to the slide deck explaining in detail the feedback form questions, and the instruction on how to download the form to mobile devices and start recording community insights.

If you want to learn more about Kobo, check out this website with Frequently Asked Questions.

Admin and user accounts

The ideal yet simple setup is to have two separate accounts – one for the data or feedback manager, and another for volunteers interacting with community members. The difference between these two accounts Is the level of access to data. The manager can see all feedback, edit it (e.g. after a follow-up), and validate it, whilst the volunteers or data collectors will only be able to use the form for data collection (without being able to see, edit or validate the total amount of submitted feedback).

This setup and division are recommended for:

- **Data security and confidentiality** - limiting access to sensitive data helps protect the privacy and confidentiality of the community members providing feedback. The data/feedback manager can oversee who has access to this information, ensuring that it is only available to individuals who need it for analysis, reporting, and decision-making processes. Additionally, by restricting volunteers' access to data entry only, we can prevent unauthorised or accidental modifications to the collected data.

- **Efficient data management** - assigning the task of data verification and validation to a dedicated data/feedback manager ensures a consistent and systematic review process. This manager can oversee the entire dataset, identify errors or inconsistencies, and validate the data before analysis. Volunteers focus solely on gathering data without the complexities of data management, thanks to which they can concentrate on engaging with community members and accurately recording their feedback.

- **For ethical considerations** - knowing that their feedback is handled confidentially and professionally can encourage community members to share honest and candid feedback.

- **For compliance with legal requirements** - many countries have strict laws and regulations governing personal data protection. Having separate accounts helps ensure compliance with these regulations by controlling access to personal data and implementing appropriate safeguards.
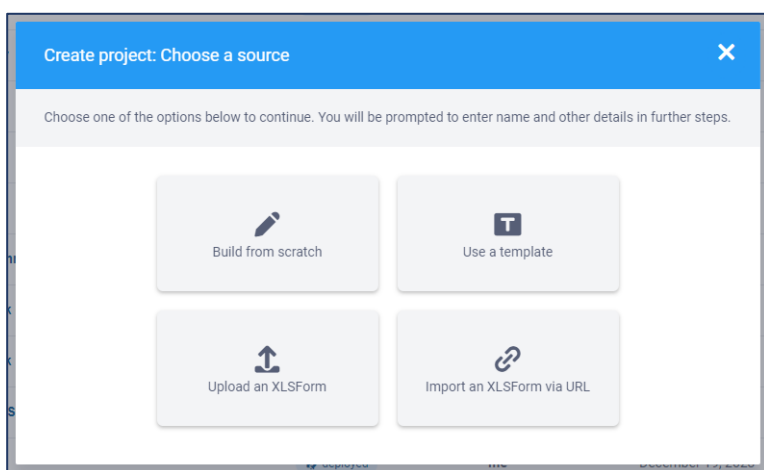
How to upload the feedback form

To make the feedback form available to volunteers/data collectors, it needs to be uploaded to the data manager's Kobo account.

- Sign in to your Kobo account.
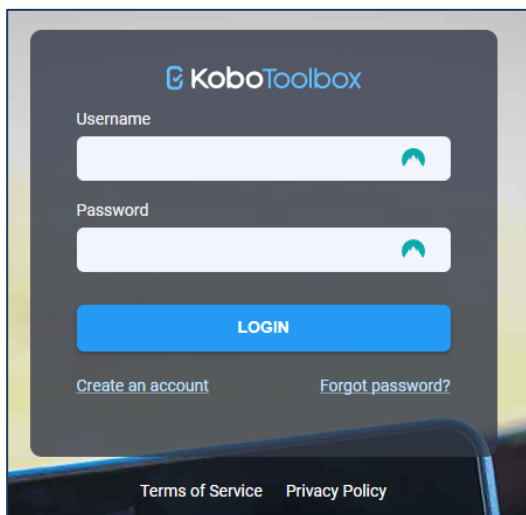
- Locate the blue "*New*" button on your dashboard – it should be visible in the top left corner.

- By clicking it, you should see a pop-up window shown below. Out of the four options available, you should select the "*Upload an XLSForm*" button at the bottom left corner of the pop-up.


- The following pop-up will ask you to select a form that you want to upload. Locate the Feedback form provided (Excel file, XLSForm) and either drag and drop it into the window or click on it and select the file.
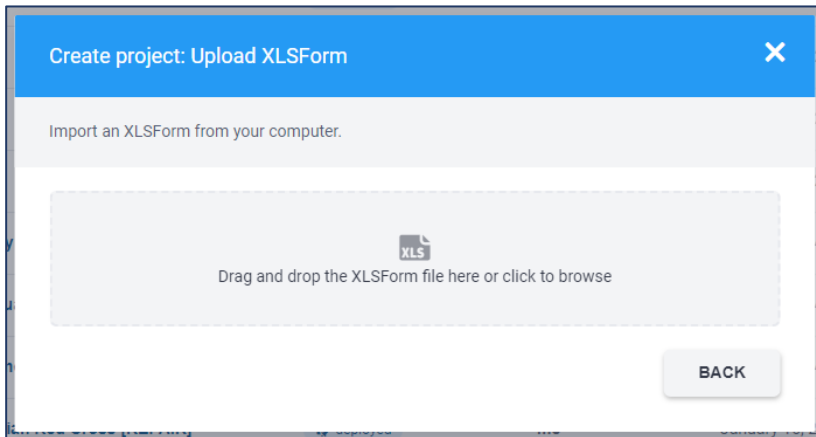
- In the next pop-up window, you will be asked to complete some details of the form. Under the "*Project name*" section, you need to add the name of the form. The name is not predefined so you can come up with yours, but make sure it's easy to understand by other colleagues using the form. This field is mandatory. "*Description*" is a short text explaining the nature and purpose of the form. It allows other colleagues to better understand in what situations this form should be used. This field is not mandatory. "*Sector*" we can already pre-define and set as "*Community Engagement and Accountability*". For "*Country*" fill in the country name and note that in the case of cross-country activities, you can add more than one country. The last two fields are mandatory to complete.

- Once all the information is complete, click the blue "*Create project*" button to be redirected to the form dashboard.
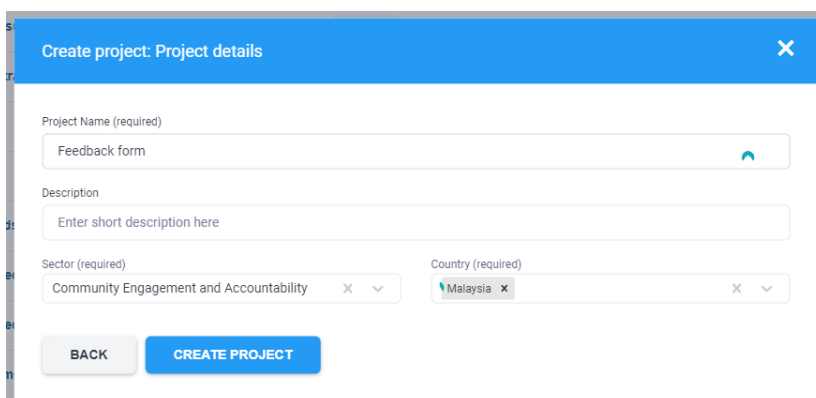

- Now, the form is uploaded to the data/feedback manager's account but making it available to volunteers/data collected requires an additional step. To make the form available – or deployed – click the blue "*Deploy*" button. Once it's done, the orange ribbon should disappear.
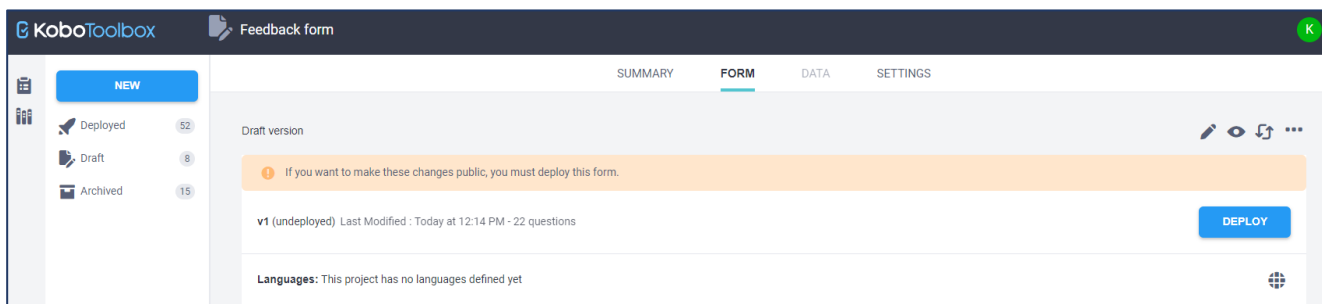
How to manage access to the form

We don't want anyone outside from our own volunteers and staff to use the form and record community insights, that is why we need to restrict access and grant it only to the account created for volunteers/data collectors. To do so, locate "*Settings*" in the top ribbon, select it, and then select "Sharing" from the left panel of the new view.

- Below your account's name, you should see an empty field (in the picture it's filled with Xs) – this is where you should add the username of the volunteers' account. Make sure you add a username and not an email address as this is a common mistake preventing people from sharing the form with other colleagues.

- From the list of different activity access, select "*View form*" and "*Add submissions*" only. Next, click the blue "*Grant permissions*" button. Now, when data collectors log into their account, they should be able to see, download and complete the form.
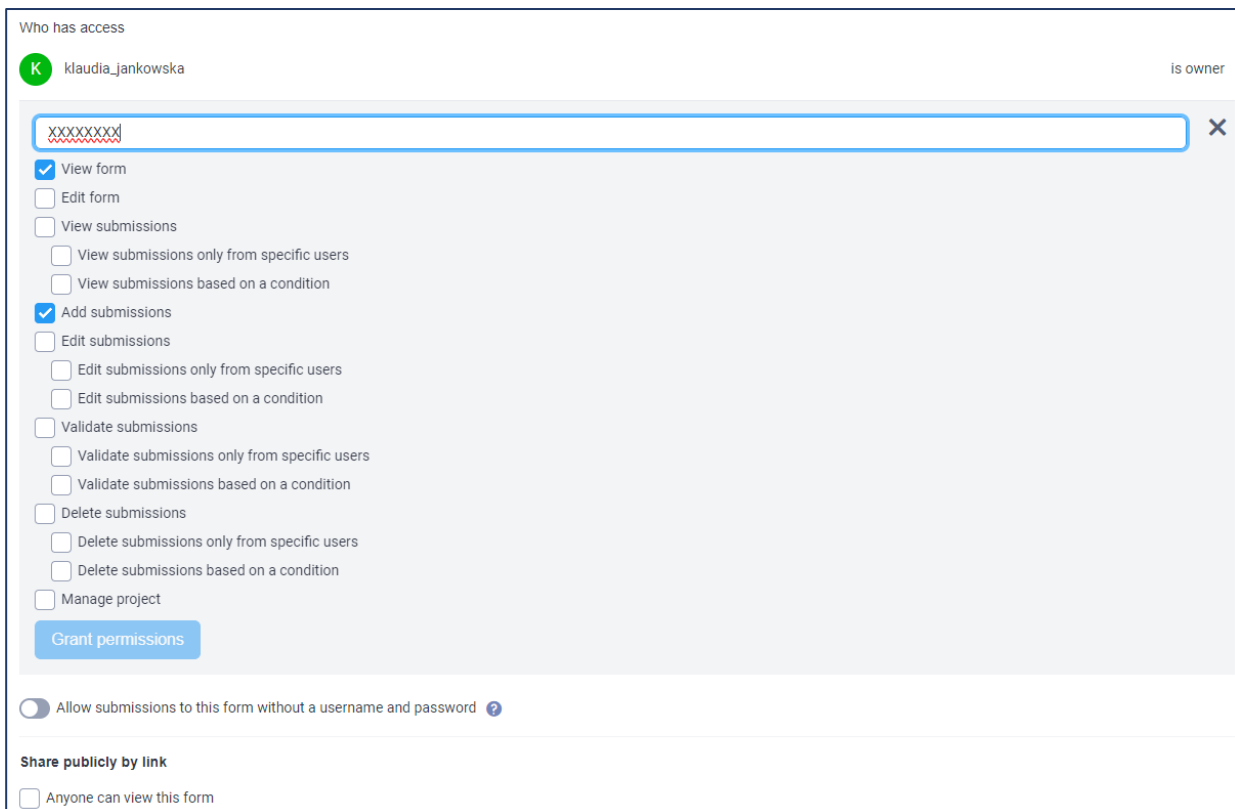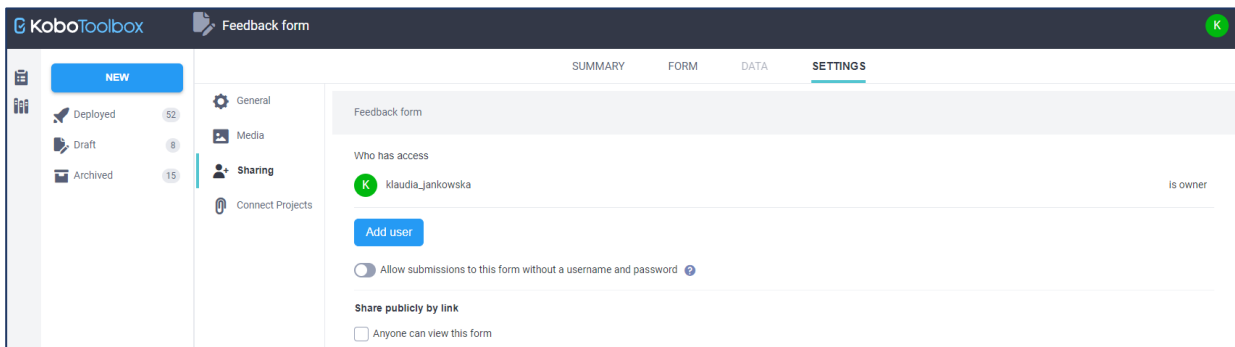
- In the picture above, "*Allow submissions to this form […]*" and "*Anyone can view this form*" are **not** selected. This is because we are restricting access to the form. In case you want to collect feedback using a website form – e.g. collecting insights via hotline phone calls and recording the information on a computer's screen, you can select or make these two options active to allow anyone to record feedback without the need to log into a Kobo account.

How to verify and export feedback data

To conduct data analysis of the community insights, you first need to see what was collected. Feedback should be analysed regularly and responded to as soon as possible.

- By selecting "*Data*" from the top ribbon, you will be able to see all community feedback recorded by volunteers. Although the default table view is comprehensive and covers all the gathered details, it is incredibly difficult to analyse the data in such a form. To make this process easier we use an Excel-based coding framework and feedback data exported from Kobo. At the same time, this table overview can already be used to look for any inconsistencies and outliers.
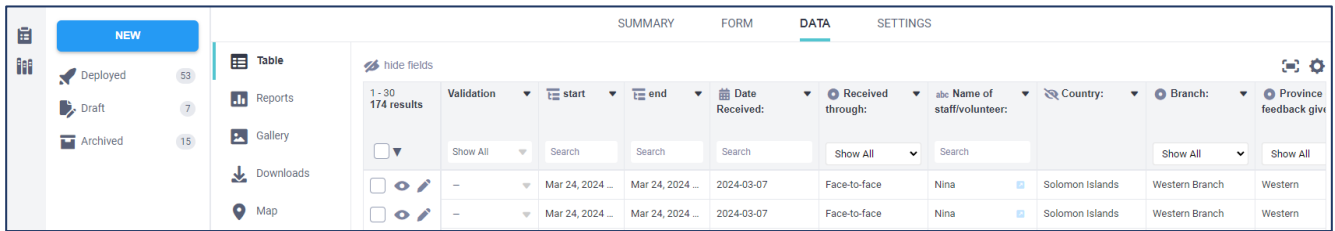
- By clicking "*Downloads*" in the left-hand panel, you will see various export options. The default settings presented in the screenshot below are what we need, which is why there is no need to modify any of the settings. If the feedback form is available in more than one language, an additional option for the export language selection will be available (but note that free text, e.g. the feedback itself will be exported in its original language, free text input is not automatically translated). Confirm the selection by clicking the blue "*Export*" button.

- Depending on the volume of collected insights and internet speed, the process of exporting might not be immediate so be patient. Once the export is ready, you will see an additional line under the "*Exports*" section. To download the Excel version of the collected feedback, click the white "*Download*" button. This file by default will be downloaded into your "*Downloaded*" folder.

- Once the file is downloaded, you can share it with the person(s) responsible for data analysis.
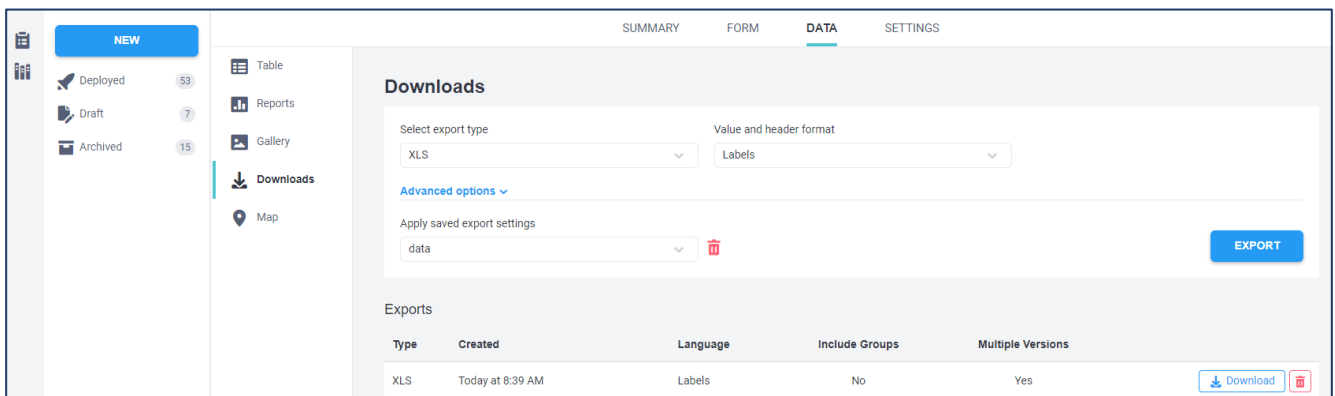
Part 3: How to code feedback using the feedback coding framework

What is the coding framework?

The coding framework is a structured approach used to categorise and analyse qualitative community feedback data. By assigning specific codes to collected community feedback, they can be organised, summarised, and interpreted. This process can then help identify common trends and concerns and make it easier for others to follow up on community feedback and take decisions.

The coding framework is a simple Excel file that can be used to combine community feedback data from different sources and channels. The file you received is designed to be fully in line with the feedback form, which is why changes done to one must be reflected in the other.

Structure of the file on the coding framework

The Excel file consists of seven tabs of which three are hidden:

- **README** tab – instructions and general information about the coding framework, including a detailed explanation of the excel structure. Please make sure you familiarise yourself with this tab before you start adding and coding feedback.

- **LOGBOOK** – the primary sheet for entering and coding feedback. This will be your main space when working with community insights. You will notice that the logbook is colour-coded:

  - *Dark blue* columns are meant for the direct copy-paste of the data exported from Kobo.

  - *Light blue* columns are completed by the feedback manager, based on the data provided under the dark blue columns.

  - *Green* columns contain automatically generated input, which means that no input is needed but also that **no data should be edited, added or removed from these columns**. Don't touch the green columns.
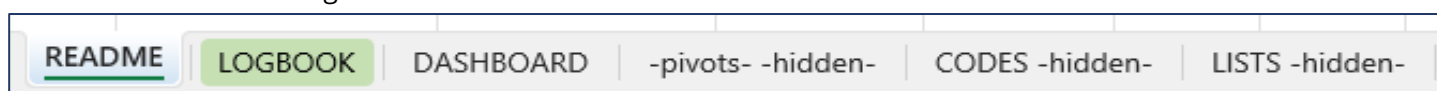
- **DASHBOARD** – graphs and visual representations of the coded feedback for your analysis.

- **CODING SCHEMA** – a list of all codes used in the logbook.

- **Codes, pivots and lists tabs** – hidden sheets containing the actual codes and lists that define the coding schema. The hidden sheets should not be modified, instead, any modification should be need-based, e.g. when we notice that many insights are coded as "Other" due to the lack of relevant codes. These changes should be done by IM or somebody with solid excel knowledge.
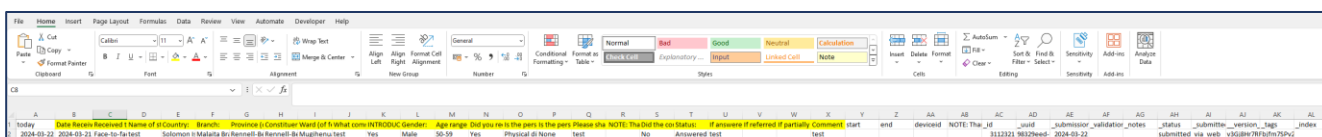
Preparing to work with community feedback data

1. Start by reading the README sheet to understand the framework's purpose, definitions, and general structure. You can always refer to this tab if you have any questions about the logbook's structure.

2. Familiarise yourself with codes and lists. The CODING SCHEMA tab is crucial for your understanding of the codes that are used to categorise feedback. The longer you work on feedback coding, the more confident and comfortable you will be in analysing using the coding frame. To unhide the hidden sheets, you need to right-click on any of the visible tabs and click "*Unhide*", then select which sheets you would like to see.

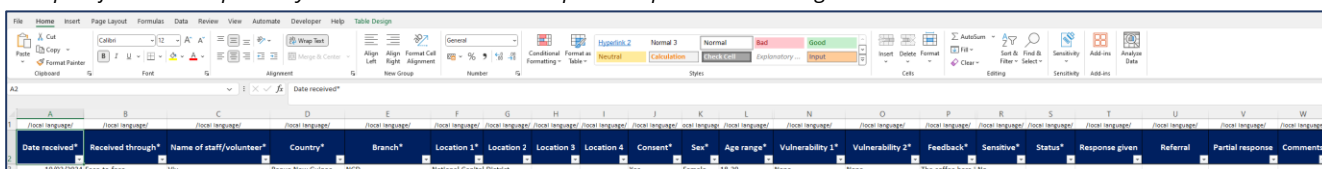## Copying the Kobo export into the logbook

1. Open the exported Excel file and review the data to ensure it includes all the feedback you intended to code. Note that the export will include all data collected with this form, and if some of it has already been coded, it's enough to copy over just the new feedback.

2. In the coding framework file, navigate to the LOGBOOK tab and determine where in this sheet you will be pasting the new data. It's crucial to maintain any existing data integrity, so make sure the columns in both files (the Kobo and export and the framework) match and you paste the data into the next available row without overwriting any previous records or leaving empty rows.

3. The Kobo export and the coding framework structures should match unless you implement any changes in one file without reflecting them in the other. To copy the new feedback data into the logbook you need to highlight everything (apart from the column headers) that is under columns from B ("*Date Received*") to X ("*Comment*") and copy it. You can do it by either clicking the keyboard combination of CTRL+C or right clicking the highlighted data and selecting "*copy*". Next, you need to open the logbook, find the first available/empty row and right-click under column A selecting "*Paste values*" under "*Paste options*". You can also do so by using the keyboard shortcut of CTRL+ALT+V. It's important to paste values only as it minimises the risk of overwriting any predefined formulas and the file crashing.

4. After pasting, verify that all data aligns correctly with the LOGBOOK's column headings and that no formatting issues have occurred. Once you're satisfied that the data has been correctly copied over, save the Excel file.

5. If you want to delete any old data that no longer needs to be stored in the logbook, highlight it and press DELETE on your keyboard. Remove all the unnecessary data from under the dark and light blue columns but make sure that you don't highlight/remove anything from under the green columns.

Example of the Kobo export. In yellow – data that is to be copied and pasted into the logbook.

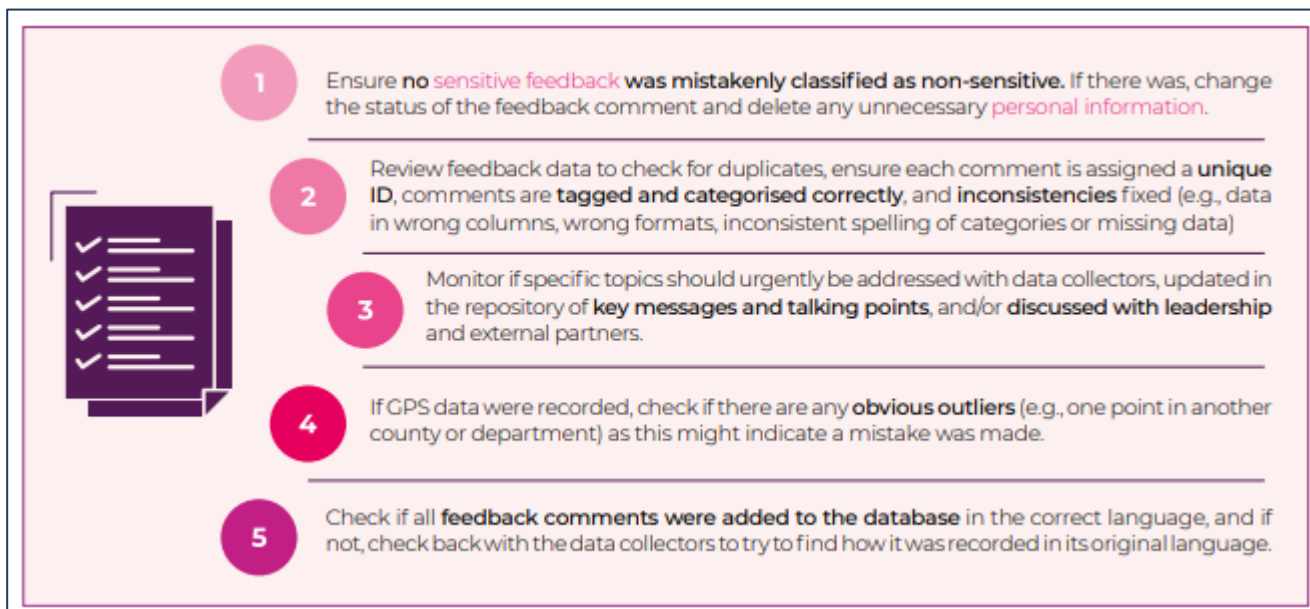Coding framework/logbook preview. In dark blue – where you should paste the copied data.

If you completed filling in the dark blue columns with feedback data, you can start the coding process.

Begin the coding process

Once again, before coding any feedback, familiarise yourself with the coding schema and understand the meaning of each code and the type of feedback it corresponds to. Codes are organised into types, topics, sub-topics, and sectors - ensure you understand the hierarchy and how codes relate to each other.

1. For each piece of feedback in the LOGBOOK, read the entry carefully and try to understand what main message the community member wanted to convey. While reading, identify the main/most important themes or issues mentioned. Look for keywords, phrases, or sentiments that indicate the feedback's focus. If you are unsure on what code to use think on what kind of follow up may be needed and code according to the most important follow up. If you have any doubts about the meaning, you can always contact the person who recorded the feedback and ask for more details.

2. Move to the light blue columns – most of them have pre-defined drop-down lists. What it means is that you have access to lists of codes and categories you can choose from instead of coming up with your own codes. All you need to do is click on the arrow in each cell that contains such a list and select the appropriate code. This approach ensures that coding is done in a consistent way and much easier to analyse.

3. Do not worry if you feel like the coding exercise takes a very long time at the start. This is normal when you have only just begun and still learn the structure and codes. The more you practice, the easier it will become. And don't hesitate to ask your more experienced colleagues for advice and support.

4. You may notice that under some of the light blue columns you cannot select or add anything unless you complete the previous column. This is because some columns depend on the input from another column. To avoid any issues and mistakes, make sure you always start coding from the first light blue column and move one by one to the right.

5. Apply codes consistently across feedback entries. If similar feedback has been coded with a specific code previously, use the same code for new, similar feedback to maintain consistency. To better understand feedback types, please refer to the IFRC Feedback Kit, Module 1, *Types of open community feedback*.

6. Ideally, the coding should not be done by one person only. It might be extremely difficult if you collect a large volume of data, but also one person is more prone to making mistakes without noticing them. You and your colleague/s should always review coded feedback to ensure codes have been applied accurately and consistently within the team. Adjust codes if you find discrepancies or if further analysis reveals a more appropriate categorisation. If working as part of a team, regularly discuss coding decisions with team members to ensure everyone uses the coding schema consistently.

7. The coding process is repetitive As new types of feedback are received, the coding schema may need adjustments or expansions. Be open to revising the schema[1] to capture the full range of feedback accurately. But only adjust if you have found enough evidence to add a code (i.e. your 'other' category has 50 or more entries of the same topic).

Part 4: Analysing community insights

To analyse data means to systematically examine it to uncover patterns, trends, relationships, or insights that can inform decision-making, guide actions, and contribute to understanding complex issues within a given context. To facilitate the feedback analysis, we can use a simple dashboard you can find under the "DASHBOARD" tab.

A dashboard is a visual representation of key data points that allows users to quickly grasp complex information through charts, graphs, and summaries. In the context of feedback analysis, a dashboard provides a snapshot of the feedback data, potentially highlighting trends, patterns, and areas of concern that require attention or further investigation. Dashboards can help better understand the data at a glance, even without delving into detailed information.

---

[1] The coding schema can be found under the "CODES -hidden" tab. Because the list are based on multiple dependencies, it is recommended to contact and consult an Information Management (or Excel) specialist colleague who will help you adjust the lists in accordance with your needs. Although the process is not very complicated, mistakes are easy to be made and these would affect all the drop-down lists used in the logbook. For more information about dependent lists, you can check out this video or follow the steps explained in the annex.

Before you start analysing the data and information presented in the dashboard, make sure the view correctly reflects the logbook entries. To do this, click "Refresh All" under "Data" in the top ribbon. It's a good practice to do it every time you enter or analyse data.
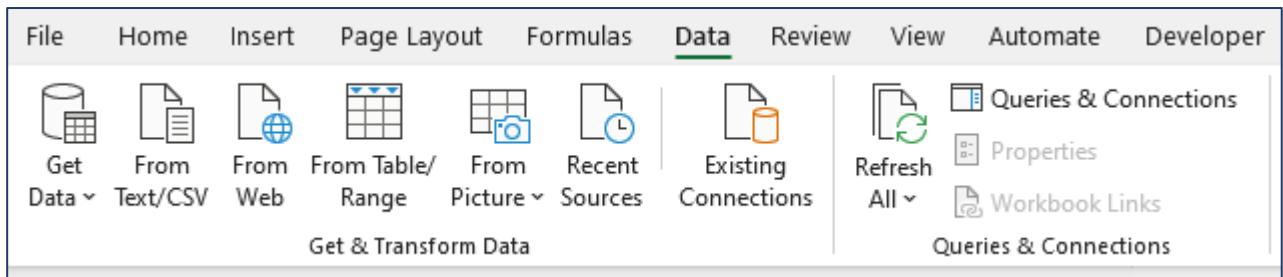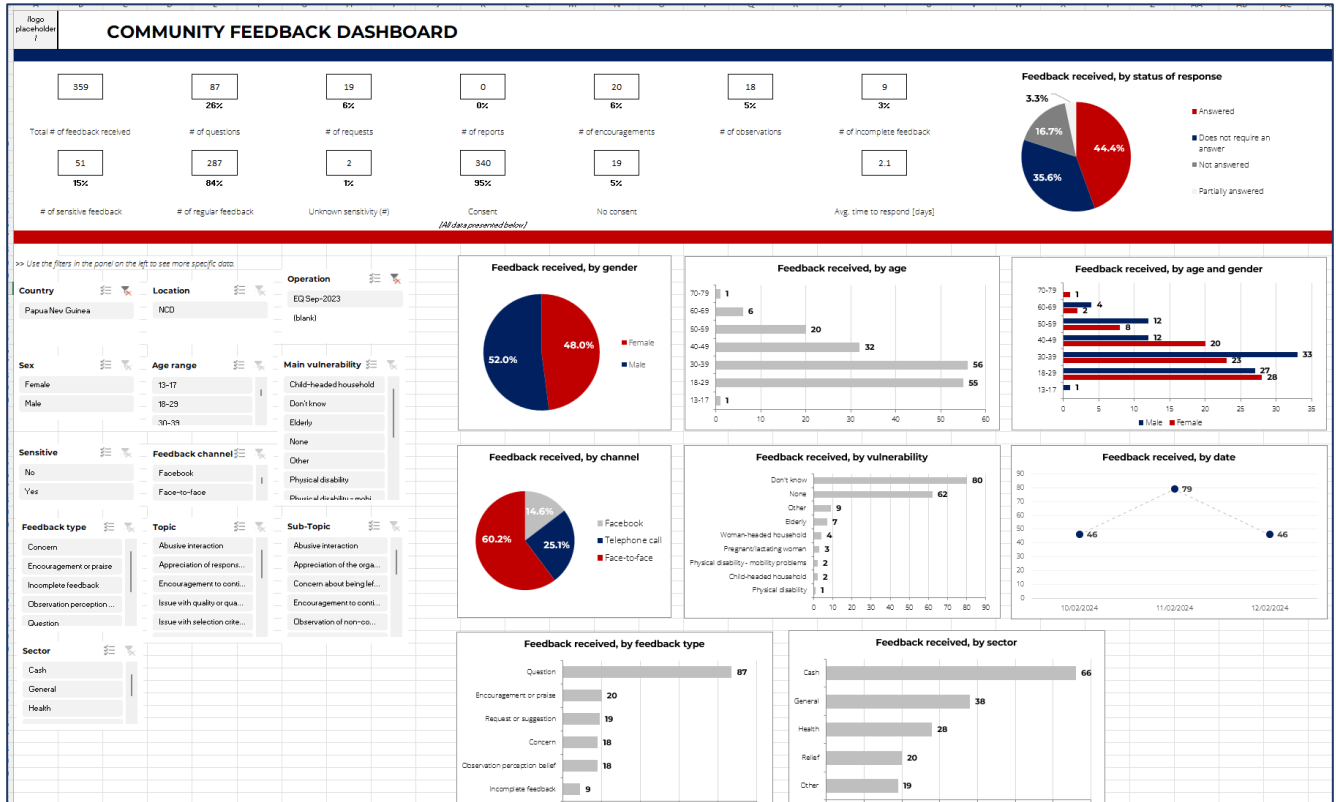
In the top part of the dashboard, you can find what are called key figures. Key figures typically include high-level data such as the total number of feedback entries, number of entries per feedback type, average response time or the number of feedback categorised as sensitive. These figures provide an immediate sense of the scale, scope, and general sentiment of the feedback collected. These figures will change when you change the logbook data but will not be affected by the dashboard filters you can find in the left-hand side panel.

Filters are a crucial feature of dashboards that allow you to display only the data relevant to your specific question or interest. By applying filters, you can narrow down the dataset to specific time periods, geographic locations, feedback categories, or any other criteria defined in your coding framework. Filters enable a more focused analysis by highlighting targeted data. You can use filters (by clicking on various options under different filter boxes) to filter specific pre-defined data or categories, or simply play

around with them taking a more exploratory approach and trying to see any potential trends or patterns. Don't hesitate to adjust your filters to explore the data from different angles or to investigate more specific areas of interest. Once filters are applied, review the charts and graphs. These visualisations will update to reflect the filtered data, helping you identify trends or data that stands out within the specific subset of feedback.

Here are some examples of questions you might seek answers to by using the dashboard:

- *What are the most common types of feedback received in the last quarter?* By filtering the data by the last three months, you can identify the main themes or issues that have emerged recently.

- *Has there been an increase in complaints about a specific sector?* Applying filters to identify feedback related to that sector over time can help you track trends and address any small issues before they turn into reputational risk.

- *Which geographic areas are reporting the highest number of sensitive feedback cases?* Filtering by geographic location and feedback sensitivity can highlight areas in need of potential urgent interventions.

- *Are certain types of feedback more prevalent among specific age groups or genders?* By filtering for demographic information, you can tailor programmes and services to better meet the needs of different community groups.

The dashboard is not just a tool for internal analysis but also a valuable resource for internal and external reporting. It simplifies the process of translating complex data into understandable insights that can be shared not only with team members but also with donors and other stakeholders. By using the dashboard's functions, you can easily filter specific data to highlight relevant information and then directly copy-paste charts and graphs into your reporting template. This approach ensures that reports are not only data-driven but also visually engaging, making it easier for readers to grasp the impact of the work being done, understand trends, and support decision-making processes.

Part 5: Ethical considerations and data privacy

In working with community members, oftentimes considered vulnerable, the do-no-harm principle is the most important. Ensuring the data and information they share with us is protected should always be our priority - by upholding these standards, we not only protect those we serve but also reinforce the values of trust, respect, and dignity that guide our efforts. Each country has its own legislations related to data protection and these are not covered in this document. Instead, you can find here a short explanation of basic terms as well as best practices of how to ensure the data we work with is protected.

Understanding confidentiality and privacy

Confidentiality refers to our responsibility to keep personal and sensitive information private. This means not disclosing information shared in trust by community members without their explicit consent. Acknowledging how critical it is to ensure personal and sensitive information is protected, as a mitigation

measure, we do not collect such data unless absolutely necessary, e.g. we would ask for personal information such as name, surname and contact details only when we know we definitely need to follow up with this person to close the feedback loop. This requires some training of the volunteers who collect feedback.

Privacy involves respecting individuals' rights to control their personal information and how it's used. This is why we always ask for consent before noting anything down, as well as explain why we collect data and how the information is processed and used.

Why ethical data handling matters

Ethical handling of information builds trust between the National Society and the communities they serve. People are more likely to share honest feedback when they believe their privacy is respected and protected.

Sensitive information, if mishandled, can potentially harm individuals or communities. Ethical considerations ensure we respect and protect the dignity and safety of these people.

Following data protection laws and regulations protects the organisation and the individuals it serves from legal repercussions.

Measures for protecting data and ensuring ethical data handling

Data minimisation - collect only the information necessary, reducing the risk of unnecessary data exposure. Do not add more questions to the feedback form because you consider such information interesting or potentially useful in the future; data collection should have a specific purpose all data collectors should stick to.

Secure storage and transmission – make sure that the feedback data is stored securely, especially if the file contains personal and/or sensitive information. Do not store it in public folders, make sure that you can password-protect the file or folder.

Access controls - ensure only authorised colleagues have access to sensitive information, and strictly on a need-to-know basis. This is why we recommend separating data manager and collector Kobo accounts and sharing access to data only with those colleagues directly involved in the feedback management process.

Anonymising Personal Data - when possible, remove personal identifiers in feedback data to protect individuals' identities, especially when analysing or sharing data for reports. Once feedback has been addressed, you can remove or anonymise the corresponding personal data.

Consent and transparency - obtain explicit consent from community members for the collection and use of their data, providing clear information about how their information will be used. It is important that you don't just rely on the consent text in the kobo feedback form but train volunteers on consent so that they can explain consent and know how to engage with communities when collecting feedback.

Training and awareness - regularly train and after training follow up with volunteers and staff on data protection practices and the ethical considerations in handling sensitive information, emphasizing the importance of these practices in their daily work.

Incident response plan - have a clear procedure for responding to data breaches or privacy incidents, including immediate steps to mitigate harm and comply with legal reporting obligations.

Part 6: Contact information for support

- National Society, CEA Focal Points: [Focal points dashboard](#) (please ask the regional CEA coordinator for the password)

- National Society, Volunteer Manager: Name Surname, position, email address.

- IFRC, CEA Coordinator: Viviane L. Fluck, regional CEA coordinator, viviane.fluck@ifrc.org

- IFRC, IM Coordinator: Dedi Junadi, regional information management coordinator, dedi.junadi@ifrc.org

Annex: How to edit the coding schema

The coding schema can be found under the "CODES -hidden-" tab. Because the lists are based on multiple dependencies, it is recommended to contact and consult an Information Management (or Excel) specialist colleague who will help you adjust the lists in accordance with your needs. Although the process is not very complicated, mistakes are easy to be made and these would affect all the drop-down lists used in the logbook. For more information about dependent lists, you can check out [this video](#) and follow the steps explained in this annex.

What is a dependent list in Excel?

Imagine you are making choices that follow a sequence, where what you choose first determines your options for the next choice. A **dependent list** in Excel or any other context works similarly. It's a list where the options available are directly influenced by the selection made in a previous list.

Here is an easy way to picture it:

You want to record items distributed to individuals after a flooding in the village X.

- **First choice (List 1):** You choose the gender of the item recipient (*Female*, *Male*, *Other*).
- **Second choice (List 2):** Based on the previous selection, now you need to select what the person received. The items available depend on the gender you selected – for example, if you selected *Male* under List 1, you would not see *sanitary pads* under List 2.

Taking the concept further, **multiple dependent lists** involve a chain of lists where each subsequent list's available options depend on the choice made in the immediately preceding list. This can continue across several levels of choices, creating a multi-level decision tree.

Let's extend the previous example:

- **First choice (List 1):** You choose the gender of the item recipient (_Female_, _Male_, _Other_).
- **Second choice (List 2):** From the previous list, you selected _Female_ as the gender of the item recipient. Because of that, now you can select _sanitary pads_ as the item received (during this distribution, sanitary pads were given only to women).
- **Third choice (List 3):** Now, based on your previous item selection, you can choose more details. In our case of the _sanitary pads_, you can select _big package (16 pieces)_ or _small package (8 pieces)_.

In the coding framework, the dependencies follow the same pattern:

- First choice – List 1 – **_Feedback type,_**
- Second choice – List 2 – **_Topic_**,
- Third choice – List 3 – **_Sub-topic._**

In Excel, creating dependent lists involves a combination of data validation rules and formulas (in this case, _INDIRECT_) to ensure that the options in a list are updated based on the user's selection in the previous list. This concept, while a bit more advanced, opens up powerful possibilities for managing data input and interaction in your spreadsheets.

How to edit the lists in the coding framework file

The coding framework is a living document, which means that the coding schema should be updated every time we notice it no longer reflects what communities are talking about sharing their insights. Below, we will consider three options – how to remove items from the lists, how to edit them, and how to add new one. In all cases, we will working with the hidden tab "CODES -hidden-".

Types	Question	Request or suggestion	Concern	Encouragement or praise
Question	Question about blood services	Request for support	Barrier to access	Appreciation of activities or services
Request or suggestion	Question about COVID19	Request for information	Issue with quality or quantity or timeliness	Appreciation of response actors
Concern	Question about cash programmes	Suggestion on response improvements	Issue with selection criteria or list	Encouragement to continue the work
Encouragement or praise	Question about response activities	Other request or suggestion	Report of incident	Other encouragement or praise
Observation perception belief	Question about the crisis		Allegation of misconduct	
Incomplete feedback	Question about Red Cross Red Crescent		Allegation of SEA	
	Question about other response actors		Other report	
	Question about preventative behaviours			
	Other question			

LOGBOOK · DASHBOARD · CODING SCHEMA · CODES -hidden- · +

To add, edit or remove an item from a Sub-topic list:

These actions are straightforward and do not require any advanced knowledge of Excel . In addition – and fortunately - sub-topics are also these categories that may need to be modified the most often.

1. For all cases, first you need to locate the list you want to modify.
2. To edit an already existing sub-topic, simply locate it and apply the required changes, e.g. fix a typo, change wording, etc. The changes should be automatically reflected in the drop-down lists in the coding framework.
3. To add a new sub-topic, first you need to find a corresponding *Topic* column. Next, find the first empty cell under the Topic list, and type in the new sub-topic. Under each *Topic* column, you can add up to 10 new sub-topics. These will be automatically reflected in the drop-down lists in the coding framework.
4. To delete a sub-topic, right-click the cell you want to remove and select Delete. If you deleted individual cells from the middle of the column and want to remove the empty space it created, you can highlight everything below the empty space, right-click it and select *Cut* (or select Ctrl+Z on the keyboard), then right-click the empty space and select *Paste* (or Ctrl+V).

To remove an item from a Feedback type or Topic list:

1. Find the *Feedback type/Topic* you want to delete.
2. If you want to remove the whole column, right-click the letter above the columns' header and select *Delete*.
3. If you want to remove an individual *Feedback type/Topic,* right-click the cell you want to remove and select Delete.
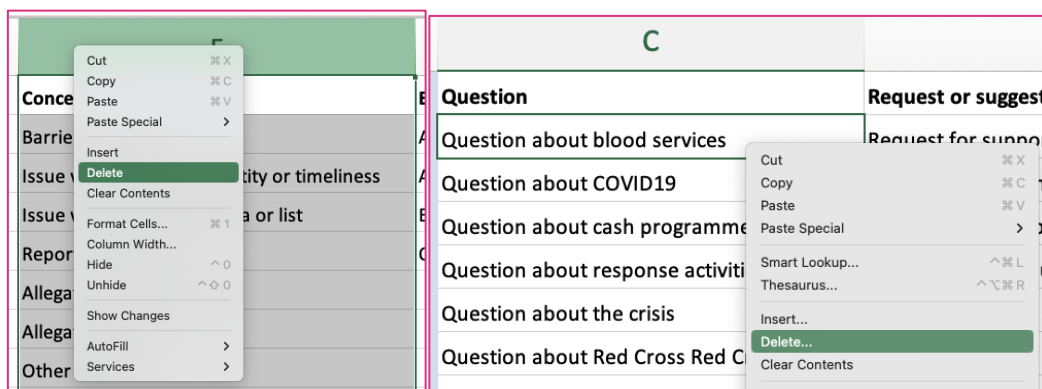
4. If you remove any *Feedback type* or *Topic*, remember to also delete corresponding categories under the subsequent *Topic* or *Sub-topic* columns.

For example: Imagine you want to delete the "*Question about blood services*" Topic from under the column C. You need find a corresponding Sub-topic column to remove. Look for a header with this exact text. In our coding framework, it is column J. As this sub-topic will no longer be needed after we have removed the corresponding topic, you can right-click the J letter and select *Delete*.

5. If you deleted individual cells from the middle of the column and want to remove the empty space it created, you can highlight everything below the empty space, right-click it and select *Cut* (or select Ctrl+Z on the keyboard), then right-click the empty space and select *Paste* (or Ctrl+V).
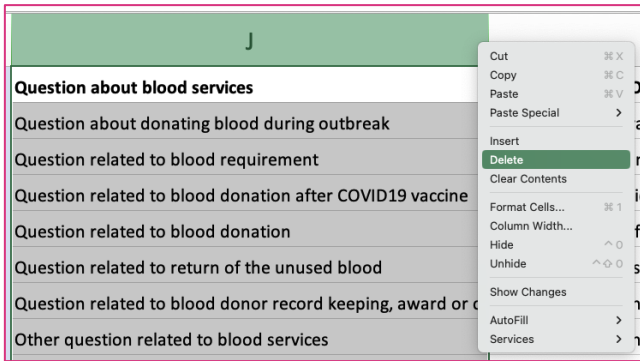
To edit an item in a Feedback type or Topic list:

1. Find the *Feedback type/Topic* you want to edit.
2. Make all the necessary changes, e.g. fix typos, change wording, etc.
3. If you made changes to any cells, remember to also apply the same changes to the dependent cells or column headers.

For example: Imagine you want to change the "*Question about blood services*" Topic from under the column C to "*Question about blood services and donation*". Once this change is done to the header, you now need to update the list it corresponds to. In this case, the list means everything that's below the header. The name of the list can be found in the top left corner. In this scenario, we would need to replace *Question_about_blood_services* to *Question_about_blood_services_and_donation*. The list name must have all white spaces replaced with underscores for the dependencies to work correctly. Apart from this swap, the list name must be the same as the corresponding header.

Sometimes Excel files do not cooperate very well, so if you come across problems with updating the list name by trying to edit the field in the top left corner, you can go to *Formulas*, then *Name Manager*, find the relevant list and edit it there.

19
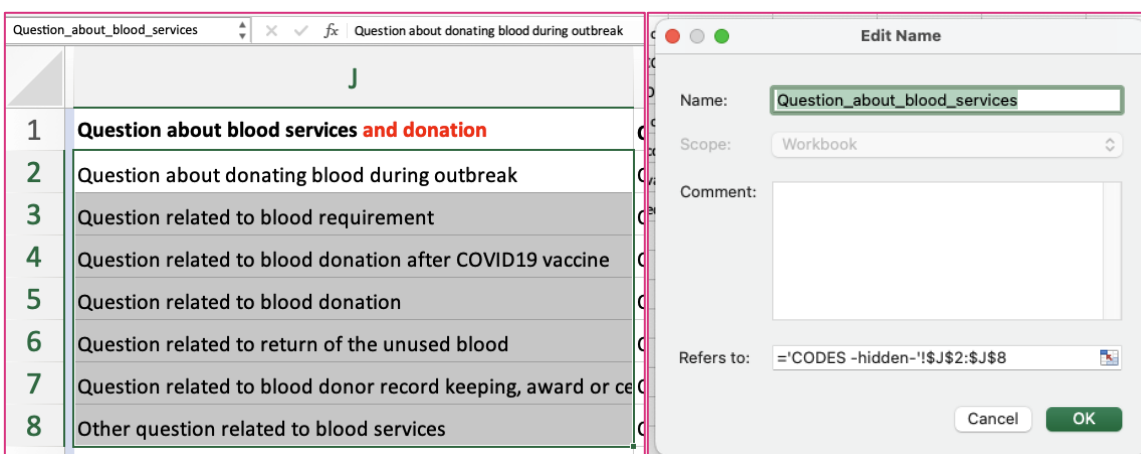
4. Now imagine you want to change the *Feedback type* "*Concern*" to "*Problem*". In this case, you will first need to change the feedback type in the A4 cell, next apply the same change to the Topic's header under E1. And finally, you will highlight all cells from E2 to E8 and change this list's name in the top left corner (using the *Name Manager* under *Formulas*).

Concern				fx	Barrier to access	
	A	**B**	**C**	**D**	**E**	
1	Types		Question	Request or suggestion	Concern	E
2	Question		Question about blood services	Request for support	Barrier to access	A
3	Request or suggestion		Question about COVID19	Request for information	Issue with quality or quantity or timeliness	A
4	Concern		Question about cash programmes	Suggestion on response improvements	Issue with selection criteria or list	E
5	Encouragement or praise		Question about response activities	Other request or suggestion	Report of incident	C
6	Observation perception belief		Question about the crisis		Allegation of misconduct	
7	Incomplete feedback		Question about Red Cross Red Crescent		Allegation of SEA	
8			Question about other response actors		Other report	

To add a new item to a Feedback type or Topic list:

Adding new items is quite like editing items in the dependent lists as again all changes need to be reflected on all levels.

1. First, add your new item/s at the bottom of the relevant list or as a new list.
2. In the *Formulas* tab, go to *Name Manager* and define a new name for your new list – even if you only added one item to an already existing list. You can also define the name by highlighting the whole list and adding its name in the top left corner. This name must reflect the heard of the column.
3. Ensure that other levels are linked to the new items or list.
4. For example: Imagine that the *Concern* feedback type does not exist in our coding framework, and you want to create it. Feedback type is our Level 1 list which means that this addition will affect Level 2 – *Topic* and Level 2 – *Sub-topic*. First add *Concern* under column A. Next, make sure that the whole list – including the new *Concern* feedback type – under *Types* is named *Types.* Use the *Name Manager* to complete it. The next step is to add corresponding *Concern* topics. You basically need to create what you can see in the screenshot added above, under the E column. This list (and its header) will need to be named *Concern.* As the final step, you will need to create sub-topics for all the topics you have just created – you should end of with what we have under columns from Y to AE.

Navigating through the creation and management of multiple dependent lists in Excel can initially seem like navigating a maze—complex and a bit daunting. But remember, every expert in anything was once a beginner. Mistakes and challenges are part of the learning process, providing lessons that contribute to your growth and mastery. So, take a deep breath, give yourself grace to learn at your own pace, and approach each challenge as an opportunity to become more adept at navigating Excel's complexities.

It's important to acknowledge that getting comfortable with dependent lists, especially multiple dependent ones, isn't just about technical skill—it's also about understanding the logic and structure of your data. And this understanding naturally takes time to develop.

And please don't hesitate to seek support from colleagues who have a bit more experience with Excel.