WHAT IS THIS DOCUMENT?

This resource can support anyone who wants to code open, unstructured feedback data. It provides an overview of the main steps and provides details of what this will look like depending on the method you are using for coding. [Here](https://docs.google.com/document/d/1fukQjwD4Ia0r0FKcK-vDcjwebJV1dbWB/edit?usp=sharing&ouid=109473232721131860036&rtpof=true&sd=true) is an overview of the different methods for coding open, unstructured feedback data.

HOW TO USE THIS TOOL?

This tool should be used in conjunction with Stage 3 – Referral and Analysis in Module 3 of the IFRC Feedback Kit. The required steps before using this tool are to decide on which method you are going to use for coding your data (see [this](https://docs.google.com/document/d/1fukQjwD4Ia0r0FKcK-vDcjwebJV1dbWB/edit) resource) and to develop or adapt a coding framework for your context (see [this](https://docs.google.com/document/d/1G3x-yQCgmtW8nbzhxcmIEwV0pNyAVrvf/edit?usp=sharing&ouid=109473232721131860036&rtpof=true&sd=true) resource).

# Compile the data you want to code

First, you need to gather all the feedback comments you would like to code. This might be all the feedback comments that were collected during a week or during a certain set of activities.

# Divide the text into the smallest possible, meaningful unit

You then need to divide the text into small units. This makes it easier for you to assign the codes, especially if you would like to code longer text such as transcripts from a focus group discussion.

One unit should contain a statement with one specific meaning, for example:

1. *“We do not like the food you provide. You are providing canned fish, but we are not used to eating fish”*
2. *“Give us money instead, we will then buy what we prefer and know how to prepare”*

If you are using Excel, this means you have one statement in one row. The data will already be structured like this if you used the [Excel feedback logbook](https://docs.google.com/spreadsheets/d/14VSl2D5XWvmeH3ZXahMbQvWIO1szyPp1/edit?usp=sharing&ouid=109473232721131860036&rtpof=true&sd=true) to consolidate the data, or having downloaded the data collected using an online survey, for example with this [KoBo form](https://docs.google.com/spreadsheets/d/1cNpWl0Eraev-FJy0QHakl2qh9VqUYXFw/edit?usp=sharing&ouid=109473232721131860036&rtpof=true&sd=true).

When using sticky notes, you can write each unit on one note. When dividing the text into units, we still want to ensure that the information about who provided the feedback is still attached.

This is what it can look like in an Excel sheet: There is one statement per row, and we have all the demographic information in separate columns in the same row:

Graphical user interface, text, application, email

Description automatically generated

When working with sticky notes you can add the demographic information on the same sticky note. This is what it can look like:

We do not like the food you provide. You are providing canned fish, but we are not used to eating fish

Female, elderly

Give us money instead, we will then buy what we prefer and know how to prepare

Male, adults

# Assign a code

The next step is to read each statement separately and think about which code in your coding framework comes closest to describing the content of the feedback comment. You can look through your codebook, read through the codes' descriptions, and go through the example statements to help you decide on the best code. If you cannot find any code that describes the feedback comment, you can always assign the code "Other".

**Make sure you stick to the literal meaning of the statement and avoid inferring what the person is trying to say.** If the statement is “*We don’t like the canned fish*” this can be coded as “Dissatisfaction with the provided food” NOT “Suggestion to provide other types of food”, as this is not what the person was saying. The provision of another type of food assistance might still be what is decided in response to statements like this, but when analysing what was shared with us, we are only describing what was heard and shouldn’t judge on what needs to happen next yet.

If certain expressions or idioms are used, you will have to interpret what this means in everyday language. Therefore, it is crucial for the data to be coded and interpreted by local staff and volunteers who master the local language and understand the cultural and linguistic nuances.

Once you have decided on the best code to use, you can assign it to your statement by writing it next to it, choosing it from a dropdown menu, or grouping your sticky notes together under the specific code. When coding the data using Excel with dropdown-menus, you will first have to choose the type, the category, and then the specific code as the next step.

This is what it can look like in Excel:

*Choosing the type from the dropdown menu:*

A picture containing table

Description automatically generated

*Choosing the category from the dropdown menu:*

Table

Description automatically generated

*Choosing the code from the dropdown menu:*

Graphical user interface, text, application

Description automatically generated

This is what it can look like if you are working with sticky notes:

**Dissatisfaction with the provided food:**

We do not like the food you provide. You are providing canned fish, but we are not used to eating fish

Female, elderly

This food you are giving us is really difficult for us to eat

Female, adult

I cannot stand the small of the fish

Male, elderly

# Assign two codes, but only when you have two unique meanings

A general rule is that if you have two codes that would fit your feedback comment, but one is more general, and another one is more specific, then the comment should only be coded under the more specific code.

For example, we use the same statement as before:

*“We don’t like the canned fish”*

Suppose you have two codes:

* ‘Dissatisfaction with the assistance provided’ – according to your codebook this is used for any comments mentioning discontent with any assistance, if there isn’t a more specific code
* ‘Dissatisfaction with the provided food’ – according to your codebook this one is used for any comments mentioning dissatisfaction specifically

Both of your codes would work, but you will only use the more specific code (‘Dissatisfaction with the provided food’)

But sometimes a statement might have two unique meanings that correspond with more than one code.

If the comment has two distinct meanings and of the two codes that would fit, none is more specific than the other one, the comment can be coded using both. We call this ‘**double-coding’**.

For example, the following statement has two distinct meanings:

*“We don’t like the canned fish, and the latrines are not working either”*

In this one sentence you have two distinct messages (“We don’t like the canned fish” and “the latrines are not working either”). We would like to capture both and can use two different codes for the same statement.

Assigning two codes to the same statement is easy if you are writing the codes next to the text in your notebook for example. If you are working with sticky notes, you can copy the statement and add them to the correct groups of codes.

**Dissatisfaction with the provided food: Issues related to WASH:**

*“We don’t like the canned fish, and the latrines are not working either”*

Female, elderly

*“We don’t like the canned fish, and the latrines are not working either”*

Female, elderly

If you are working in an Excel spreadsheet, or a table in a word processing software, you can copy the row of the statement and paste it underneath. In one row you can assign the first code, and in the next row you can assign the second code.

Graphical user interface, text, application

Description automatically generated

# Have a review process

Once a person has finished the coding, a second person should review the coding. This is not only necessary as human error will always occur, but also important to identify and discuss differences in the understanding of the different codes. It takes a while to familiarize yourself with the coding frame, and is normal to make a few mistakes in the beginning, or to have some discussions about the definitions of the different codes. While in the beginning, all the coding should be reviewed, spot checks might suffice at a later stage.