Data collection in Zambia. Photo: Bryan/Oxfam

GOING DIGITAL

Improving data quality with digital data collection

Three years on from the Going Digital: Using digital technology to conduct Oxfam's Effectiveness Reviews pilots, where we shared learning on the added value of using digital technology to conduct our impact evaluations, Oxfam continues to develop its survey techniques through the use of technology. The second paper in this series, Going Digital: Using and sharing real-time data during fieldwork, demonstrated how Oxfam was sharing real-time data during fieldwork to increase engagement and participation in surveyed communities, as well as improving integration between qualitative and quantitative data collection techniques. In this third paper, we present some of the features enabled by digital data collection technology that we have been piloted and used to improve quality and accuracy of data. It also explores ways in which the ethics of respecting privacy can be improved in survey data collection.

OXFAM

1 INTRODUCTION

In 2015, Oxfam GB's team of Impact Evaluation Advisers started using digital devices to conduct household surveys for Oxfam's Effectiveness Reviews (internal quasi-experimental impact evaluations). In the last three years, we have conducted individual and household surveys using digital devices in more than 20 countries for the majority of our Effectiveness Reviews benefitting from improvements in data security, accountability, accuracy and timing, and even reducing costs.

We documented and shared our experience of making the transition from a paper process to a digital one in Going Digital: Using digital technology to conduct Oxfam's Effectiveness Reviews. We shared our experience in processing and sharing data in real time during fieldwork in Going Digital: Using and sharing real-time data during fieldwork. In this paper, we present some of the features used for improving the quality of data collected through household and individual surveys. In many of these data collection processes we took advantage of features provided by digital devices to pilot and implement solutions that would help to increase data quality. We recognize the limitations of automated technology, and there remains a need for expertise and rigour in monitoring and analysing data, but this paper explores the role that in-built functionality can offer to ensure data meets certain standards.

In this document, we will present some of the digital data collection features employed during survey data collection in the last three years and argue how these enable data-quality improvements to consistently meet or exceed sector standards.[1] In the appendices, we will also provide practical examples with the accompanying code[2] from SurveyCTO, a mobile data collection software application, and Stata, statistical analysis software. Through these examples, and by making the code available, we hope that other colleagues and practitioners will be able to build on our experience to use and extend digital data collection features, ultimately improving data quality and survey practices. We hope this technical and practical guide is useful for colleagues and practitioners implementing survey data collection with digital technologies. The sections of the paper are grouped by survey features that demonstrate the quality checks implemented and why, with a link to the appendix, which shows the associated coding used in SurveyCTO and Stata.

2 PRE-RECORDED CONSENT FORM

One area identified as critical in our experiences collecting data for monitoring and evaluation, is how informed consent is explained by the enumerator and obtained from the respondent. Oxfam's Responsible Program Data Policy defines informed consent as 'a process for getting willing permission to collect data of any kind based on a clear appreciation and understanding of the facts, implications, and consequences of any engagement from participants'.[3] The Responsible Program Data Policy was released in 2015, framed according to the following rights: the right to be counted and heard; the right to dignity and respect; the right to make an informed decision; the right to privacy; and the right to not be put at risk. The right to make an informed decision includes the decision of the respondent as to whether they give informed consent to participate in the survey or any data collection exercise.

All data collection activities use a consent statement that is traditionally read out to, or read by, the respondent. Obtaining consent is more than following a standard template, it involves an open conversation, the ability to ask questions, and is very contextual. Oxfam is exploring ways in which it can improve the consent framework, which is outlined in a blog on 'Early stage prototyping: Consent

and data minimisation for those affected by humanitarian crisis'. However, at the most basic level, there is a concern amongst researchers and monitoring and evaluation practitioners that during data collection processes, the consent statement is not always given enough attention by the enumerators at the beginning of the interview. During training, significant emphasis is put on obtaining consent appropriately, but it has still been found that at times this is cut short or missed altogether. The ICT in Programme team explore how the use of technology offers opportunities to amplify and improve the effectiveness of Oxfam's programme work and have looked at ways in which technology can improve how we explain and obtain informed consent from the respondent.

During a multi-country baseline, Oxfam recorded the consent statement in five different languages required for the survey. At the beginning of each interview the enumerators played an audio of the pre-recorded consent form, explaining the purpose of the survey. An example of this consent code is given in the appendix.

As a way to further ensure quality, the survey team was also able to check if the audio had been played fully before the beginning of the interview, as speed checks on the length of the audio files were introduced. When the time taken was less than the length of the audio recording in the local language, the data export would flag that the audio file had been cut short. This meant that supervisors could follow up with enumerators about playing the full audio file, leading to improvements and ensuring that respondents heard the whole consent form. This process was not without its challenges; some audio files had poor quality sound making it a challenge to hear, and supervisors were unable to find time to do thorough checks and follow up with enumerators when necessary. What the process demonstrated was that often the audio file was not played in full and only where supervisors followed up the speed violations with the enumerators did the process see a marked improvement.

In future data collection processes, it is advisable to combine pre-recorded consent forms with a written version of the consent form to be left with the respondent. This will ensure the respondent also has a written record, including the contact details, in case they decide to withdraw their consent.

3 GENERATING UNIQUE IDENTIFIERS

Unique identifiers, also known as IDs, are essential when data analysis requires the merging of different sources of information, such as questionnaires conducted with different respondents and at different levels (household and individuals, producer group and producer group members, etc.), or several data collection exercises scheduled with the same respondents. Depending on the type of sampling approach, IDs can be preloaded or generated on the spot.[4] For Oxfam's Effectiveness Reviews, identifiers are used to link datasets together, and personal information is also collected to enable future data collection exercises as needs be. However, in some instances, identifiers can be used to minimize the personal data collected and stored, in which case randomization of a series of characters could be used to generate identifiers.

PRELOADING IDENTIFIERS

When it is possible to obtain the full lists of respondents before starting data collection, it is advisable to assign the ID to each respondent before beginning the survey. Preloading allows for information at

hand (such as names and IDs) to be pre-entered in the survey. While preloading of identifier information is possible in a paper-based survey, the process and survey logistics are made more flexible when using digital technologies; it allows the uploading of the complete list of respondents on the devices, is accessible offline, and reduces the risk of mistakes in entering the identifiers. Enumerators can simply filter by geographical area, and a list of contact details with the associated ID will show up for the enumerator to select. An example of the SurveyCTO code is presented in the appendix.

IDENTIFIERS GENERATED ON THE SPOT

In many instances, it is not possible to obtain the full list of sampled units in advance, such as households, making it then impossible to create a preloaded file with unique identifiers. In such cases, the survey team will have to create unique identifiers on the spot, once sampling is done, and assign the identifier consistently across the different survey tools (household and individual surveys in the setting of the Effectiveness Reviews for example). Typically, this creates challenges of duplication and consistency. One way to do this to avoid these mistakes is to combine a variety of information, such as geographical information, enumerator ID, interview date, and sampling order. When generating such ad-hoc identifiers, digital data collection helps by reducing the amount of information that is entered manually, thus decreasing the chances of mistakes. Examples of the SurveyCTO code are presented in the appendix. However, it is important to be aware that, while such a system reduces the chances of errors, it does not prevent them entirely – mistakes in information entered manually can still occur. This is why good tracking tools need to be put in place by the survey team, to ease information flow within the team. Daily monitoring as the data comes in is also key (see section ‘Survey progress monitoring and automated quality checks’ below).

Finally, under both scenarios (preloaded or ad-hoc identifiers), there is a trade-off between reducing the chances of mistakes by automating the generation of IDs and allowing the program to accommodate any unforeseen challenges (such as a village being inaccessible, an enumerator having to leave the team, sudden changes in the composition of teams, etc.). The SurveyCTO codes presented in the appendix allow for flexibility by including ‘other’ categories at every stage of the identification process (at different geographical area levels, household level, and enumerator identification), and enumerators are trained to use these ‘other’ categories as a last resort.

4 HOUSEHOLD ROSTER INTERVIEW FLOW

The household roster is an important component in most household surveys as it contains detailed information of all the household members involved in a survey. Accurately gathering this information is particularly important to ensure quality data to understand the make-up of a household and ensure no one is omitted. At the same time, because the questions are repeated for each individual, the process of collection can take a long time and be error prone.

While digital data collection provides an easier interview process compared with paper-based interviews, there is still a concern that, because of the number of follow-up questions for each household member, respondents might be tempted to avoid mentioning some household members to shorten the length of the survey. To obviate this issue and minimize such risk, Oxfam GB’s Impact Evaluation Advisers were keen to ensure that the household roster section was set up with a specific

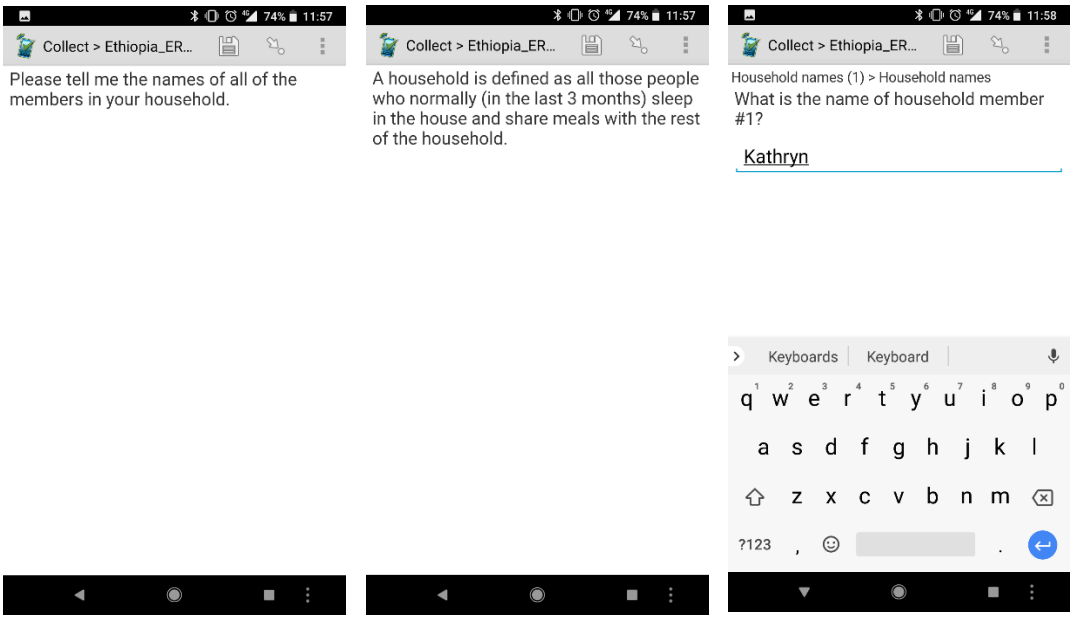
workflow. This flow would require the enumerator asking first how many people lived in the household, followed by asking all their names, and finally following up with the specific questions about each household member.

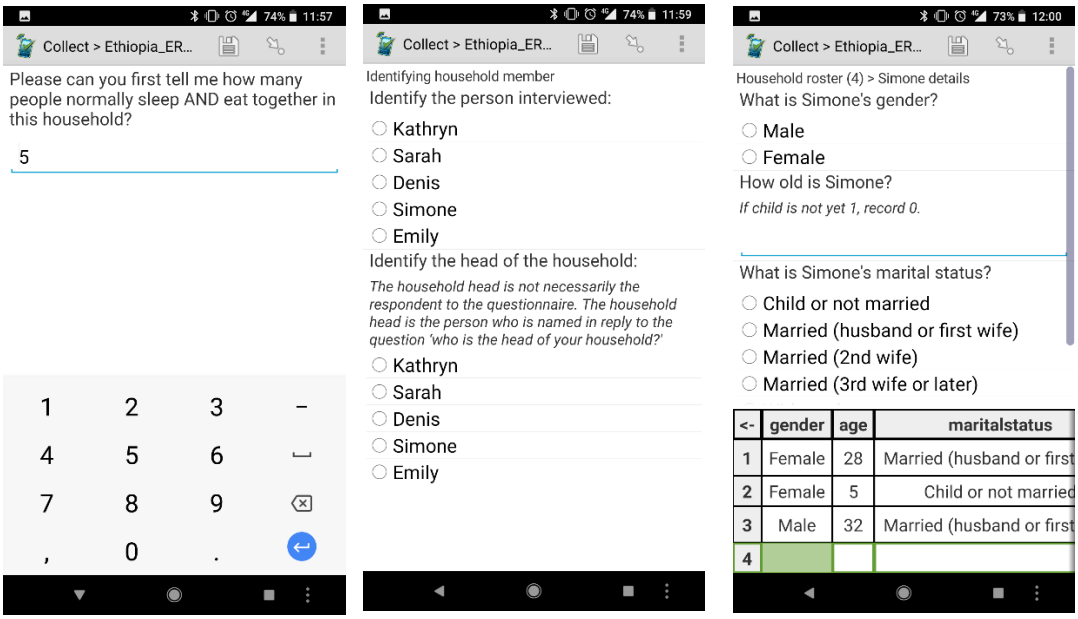
Figure 1: Example household roster – paper survey form

No	101	102	103	104		105	106	107	108	109
	Name of household member	Identify the person interviewed	Identify the head of household.	Gender		What is his/her age? *Enter the approximate age in years.*	What is his/her marital status? *0 = child or not married* *1 = married (husband or first wife)* *2 = married (2nd wife)* *3 = married (3rd or later wife)* *4 = widowed* *5 = separated or divorced* *6 = other*	What is the highest level of education he/she has achieved? *0 = child below school age* *1 = never attended school* *2 = some primary education* *3 = completed primary education* *4 = some secondary education* *5 = completed secondary education* *6 = any post-secondary or university education* *9 = does not know*	Is he/she fit and able to work? *0 = child below working age* *1 = Generally good health: does productive work* *2 = Serious disability or sickness: cannot do any productive work* *3 = Too old to do any productive work.*	If he/she is aged between 7 and 18 years old: Did he/she attend school last week? *1 = Did not attend school* *2 = Attended nursery or primary school* *3 = Attended secondary school* *4 = Attended post-secondary education*
				F	M					
1.		☐	☐	☐	☐	____	__	__	__	__
2.		☐	☐	☐	☐	____	__	__	__	__
3.		☐	☐	☐	☐	____	__	__	__	__

Figures 1 and 2 show the differences in how paper-based and computer-assisted questionnaires look. On a mobile device, the design ensures that this question flow is followed, and the software dynamically pulls through the names entered in question 101, into questions 102–109.

Figure 2: Example household roster – mobile devices form





To further aid respondents and enumerators, the devices also display a summary table with the information entered, to minimize confusion in cases where there are many people within the household. This can be checked with the respondent and can easily be edited if required.

Examples of the SurveyCTO code that enables this flow is available in the appendix.

5 CONSISTENCY CHECKS

Another feature that digital data collection allows is on-the-spot consistency checks. This could be used through hard checks to reduce data-entry errors (such as typos or other types of mistakes), which purposely prevent moving on to the next step, or the survey from being completed, if there are predefined inconsistencies, and to flag potential inconsistencies for the enumerator's attention, or through soft checks, which may not be as rigid and allow movement to the next step, but still present an alert.

Overall, for each survey instrument, there could be the potential for a large number of such checks to be programmed. However, in order to not over-burden programmers and enumerators, and to avoid making the program too rigid, which could create challenges, checks should focus on critical variables for the future data analysis.

HARD CHECKS – DOUBLE-ENTRY AND CONSTRAINTS

Digital data collection allows for data to be entered as the interview is going on, reducing the chances of typos or other mistakes specific to data entry. Such mistakes are more likely to happen in continuous text or text variables, where enumerators do not choose from a list of options. To reduce the chances of typos even further, double-entry of a given survey question can easily be programmed. Hard checks are programmed so that in the case of mismatch between the two separate entries, enumerators have to go back and correct the data. An example of the SurveyCTO code is presented

in the appendix on a continuous variable. The enumerator first asks ‘How many rooms are there in your compound that are used for sleeping by your household members?’ and enters the respondent’s answer; the enumerator is then prompted to enter the value again, and the program checks for the two entries being the same.

Referring to the unique identifiers above, another way to reduce chances of mistakes when these are entered manually is to prompt the enumerator to enter the identifier at the very beginning and end of the survey. Ensuring the correct identifier is submitted is crucial for merging datasets, so if the enumerator enters two different identifiers, they would not be able to move forward. An example of the SurveyCTO code is presented in the appendix.

HARD CHECKS – CALCULATION FUNCTIONS

A clear advantage of using digital devices for data collection is the ability to instantly perform calculations. This is a useful feature when asking questions the answers to which need to add up to a certain number.

The calculation function was used in a data collection exercise investigating sources of income. Figure 4 shows the paper version of the questionnaire. Enumerators asked about the income received in the previous 12 months from a list of several income sources (questions 405A to 405N). At the end, they had to add up all the values received and ask respondents to confirm if that represented the total amount received by the household in the last year. With the paper-based method, this process would involve an interruption in the interview process while the enumerator performed the calculation. However, using digital devices, the calculation (question 406) was conducted automatically. The SurveyCTO code for this is given in the appendix.

Figure 3: Sources of income calculation function – paper survey form

		401	402	403	404	405
		Did any member of your household earn any income from this source during the past 12 months? *1 = Yes* *0 = No*	**Did any member of your household earn any income from this source in the year 2014?** *1 = Yes* *0 = No*	*If 401 = Yes:* **In the last 12 months, who engaged in this activity?** *[list household members]* *Enter codes*	**What was your household’s income from [Source] in the last months?** **JOD**	**What was your household’s income from [Source] in the last 12 months?** **JOD**
A	Cash for Work	|__|	|__|	|__|__|__|__|	|__|__|__|__|	|__|__|__|__|
B	Payments from home production sale (e.g. Sale embroidery, carpet, food preparation, handicrafts, other)	|__|	|__|	|__|__|__|__|	|__|__|__|__|	|__|__|__|__|
C	….	|__|	|__|	|__|__| |__|__|	|__|__|__|__|	|__|__|__|__|
N	Others	|__|	|__|	|__|__| |__|__|	|__|__|__|__|	|__|__|__|__|

Figure 4: Paper-based questionnaire – income questions

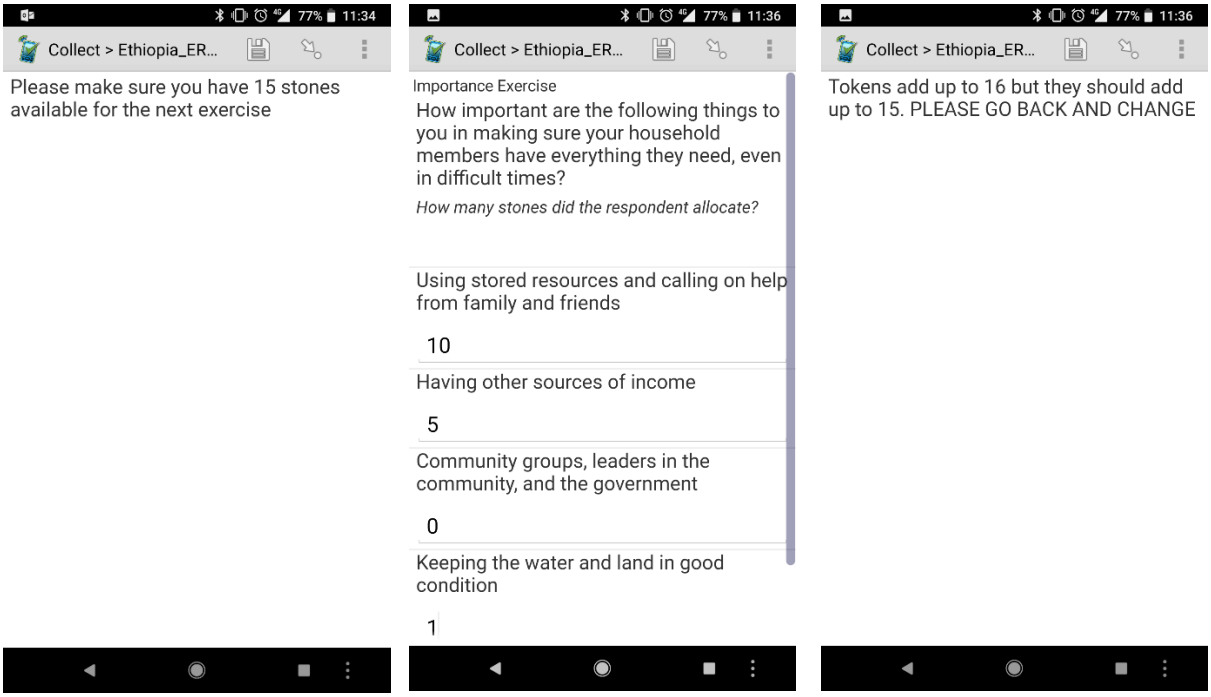
406	**Would you consider [**Make calculation**] JOD to represent correctly the amount received by your household in the last year?**	1 = Yes 0 = No	|__|
407	If 406 = No: **How much would you consider to be the amount received by your household in the last year?**	JOD	|__|__|__|,|__|__|__|

The calculation feature was used for a weighting exercise during the Effectiveness Reviews conducted in Ethiopia, Somaliland and Nepal in 2015/2016. Respondents were asked to complete a 'budget allocation' exercise, designed to elicit their preferences over which dimensions of resilience they considered to be most important, by distributing a number of 'stones' across different dimensions to represent their priorities. Resilience was conceptualized using five dimensions, with each dimension represented by a picture, as shown in Figure 5. Respondents were asked to allocate a fixed number of stones to pictures that represented these dimensions, to express how important they considered each of these dimensions to be.

Figure 5: Weighting worksheet from Ethiopia and Somaliland

The total number of stones available to be allocated across the five dimensions was fixed, so respondents and enumerators were forced to allocate all the stones before being able to proceed with the interview.

Figure 6: Example of weighting calculation – mobile devices form



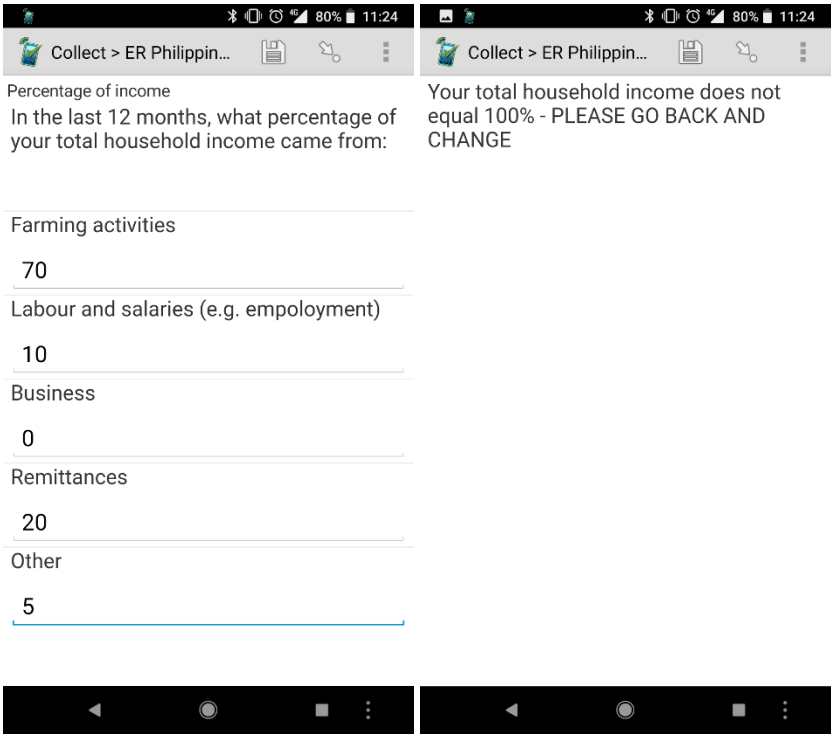
The SurveyCTO code is available in the appendix.

SOFT CHECKS – CALCULATION FUNCTIONS

Soft checks can also be included in the survey to flag potential inconsistencies, which require further prompting of the respondent by the enumerator. When conducting an interview, enumerators have to not only create and maintain a good relationship with the respondent, but also make sure each question is well communicated and understood, and think critically about the consistencies of responses given across the survey to prompt the respondent as needed. Soft checks support the latter. They are particularly useful when some combinations of answers to key questions are unlikely, but not impossible, or when there is a need to allow flexibility. An example of each of these instances is given below.

A very basic use of the calculation function feature was used during a survey where the respondent was asked to estimate the proportion of income coming from different income channels. The question required the sum of the incomes from each channel to add up to 100% of the total income. A SurveyCTO code was developed to show to the enumerator an error message if the total value did not add up to 100%. However, the code was deliberately designed to only flag the issue, allowing the enumerator to continue with the interview if they needed.

Figure 7: Example income calculation – mobile devices form



Another example relying on calculation functions is the combination of age and marital status in a household roster. The younger a household member is, the less likely this person is to be widowed. but there may be a very few cases of a household member being below 20, and already a widow. To make sure the cases recorded as such reflect the actual situation and are not a typo in age or marital status for the specific household member, a soft check can be built into the program. Each time an enumerator enters 'widow' for the marital status of a household member below 20, a message pops-up: 'Enumerator: you just filled that [*Name of the household member*] is a widow and is below 20; please check this information before moving forward.' The SurveyCTO code is presented in the appendix.

Many such combinations of variables are possible, and questionnaire programming will therefore focus on the most critical variables for future data analysis.

6 RANDOMIZING SURVEY FEATURES

Another function that is significantly useful in improving data quality and questionnaire design is the ability to introduce random variation in the survey features, which enables performing A/B testing on questionnaire designs, testing the impact of one design approach versus another.

For example, Oxfam GB's Impact Evaluation team randomized two questionnaire modules on time-use during an Effectiveness Review in Indonesia on a women's empowerment project. Comparing the results from this A/B experiment provided valuable information on the strengths and weaknesses of the two designs. It provided evidence to enable striking a balance between accuracy and time required to collect the data.

Findings suggest that while one version takes longer, on average, to administer, it is also more reliable in reducing enumerators' bias. Full findings, analysis and questionnaire design are reported in the following blog: Measuring time: Comparing questionnaire designs.

A similar approach was used as part of a larger multi-country field experiment on surveyor identity and response bias. We randomized the consent form at respondent level to suggest that the enumerator was conducting the survey either on behalf of Oxfam or a local independent survey company. While both were true, the experiment aimed to test whether individuals' responses were systematically influenced by the type of organization that respondents believed to be conducting the study.[5] The analysis of the data is currently underway and will be shared soon. The SurveyCTO code is in the appendix.

SURVEY PROGRESS MONITORING AND AUTOMATED QUALITY CHECKS

A significant feature of digital data collection technologies is the ability to download and process the data instantly while the collection is still taking place. As we previously argued in Going Digital: Using and sharing real-time data during fieldwork, this has the potential to increase engagement and participation in surveyed communities. Additionally, we argue that this feature also has the potential to improve data quality by monitoring incoming data while data collection is still underway.

In close collaboration with the survey manager, this enables the identification of potential training weaknesses or misunderstanding, enumerators who need additional supervision, and data mistakes and inconsistencies that need correction through call back of respondents or second visits. These checks need to be focused in order for the survey manager to take action on them promptly, while data collection is still ongoing.

While each questionnaire has its own features, and each data collection exercises its own protocol, we provide here a general, but not exhaustive, list of some of the monitoring and automated quality checks we have implemented in the past that can be used in other surveys. There are different ways to do this, and an effective way is to develop a Stata dofile,[6] which automatically runs through the data and outputs descriptive statistics and observations that need further enquiry in a sharable format. The relevant (simple) Stata code examples are presented in the appendix.[7]

There are many ways to write the code to perform the checks presented below and output the results. The output format and the communication flow with the survey team matter as much as performing the check itself.

Some of the monitoring checks include:

1. Checking the response rate and the number of completed interviews:

- by geographical area, to follow survey progress, which may help identifying areas where respondents are harder to interview than others, and get feedback on the reasons why;
- by enumerator, to identify enumerators who have more or less difficulty in conducting surveys;
- by treatment arms, to assess if there are significant differences between intervention and comparison groups due to survey implementation and/or availability or willingness of respondents to take part in the survey.

An example of Stata code for monitoring the number of completed surveys per enumerator and outputting this statistics in an Excel file (dofile 1, line 30 to 49) is presented in the appendix.

2. Checking the survey duration by enumerator

Digital data collection allows for time stamps throughout the survey. This check can be performed for the whole survey duration or for a specific module of interest. Checking for differences in average survey duration per enumerator allows the flagging of enumerators who conduct interviews longer or shorter than the average. Both scenarios will require additional supervision from the survey manager to understand the drivers of this variation and take appropriate action.

An example of the Stata code is given in the appendix (dofile 1, line 50 to 112), in which we look at the average, minimum and maximum survey duration per enumerator, based on the variable survey duration created by the questionnaire program. The code also outputs the share of surveys that lasted less than 15 minutes (an arbitrary threshold that may be needed at the onset of the survey, when very few surveys were completed), less than half the median survey duration, or more than twice the median survey duration, by enumerator. An example of outputting the list of surveys that are particularly short and need more investigation is also given.

3. Duplicates and matching of IDs

Checking on a daily basis for duplicates in IDs as the data comes in allows for communication with the survey team and immediate correction of the mistakes. Similarly, in a setting where several surveys are matched and data collected simultaneously, checking for matching as the data comes in will allow for corrections.

An example of Stata code for a survey protocol in which up to two individual surveys were conducted per household survey are presented in the appendix (dofile2). The code identifies duplicate IDs in the household and individual surveys separately and outputs such cases in an Excel file for communication with the survey team. The code then checks for mismatches between the household survey and individual surveys, and outputs unmatchable IDs for further investigation.

Some of the quality checks include:

1. Checking for the distribution of continuous variables

Looking at the distribution of continuous variables gives valuable information, particularly at the start of the survey, as it can allow the person doing the remote monitoring to spot misunderstandings from specific team members, typos, or geographical specificities that require more investigation (for example a unit conversion factor being different from an area to another). As more data comes in, one can look at outliers based on mean and standard deviation of the distribution (particularly on the right-hand-side of the distribution for positive continuous variables) and the concentration of those outliers per enumerator or per geographical area.

An example of Stata code is presented in the appendix (dofile 1, line 124 to 134). The code looks at the distribution of variables by tabulating them (required at the onset of a survey when very few observations are collected), identifies outliers,[8] and looks at enumerators and geographical locations for which outlier observations are identified. The code then outputs the share of surveys with outlier value per enumerator (similar code can be written by geographical area), and outputs the pseudo-anonymized data of specific observations that require more investigation and correction by the survey team.

2. Checking for internal consistency

Inconsistency, as mentioned above, is combinations of entries that are either impossible (but could not be programmed as hard checks in the survey programming) or possible, but very unlikely. This could be particularly important in cases where different questions about similar topics take place at

different points in the questionnaire, where it is less easy for the enumerator to catch the inconsistency and probe further. For example, an early question asks about participation in a group, while later questions ask about knowledge of the existence of this group. Participants of the group should know of its existence. If this is not built into the questionnaire program as a hard check (or even a soft check), this could be monitored to flag the consistency in answers without having alerted respondents or enumerators at the time of submission. Another example could be the relationship between age and marital status mentioned above. If these are key variables for the analysis, monitoring the likelihood of such cases will ultimately enhance data quality[9] through communication with the survey team.

An example of Stata code is given in the appendix (dofile 1, line 167 to 200). The code flags inconsistent observations (defined when two variables do not equate – the condition will be specific to the questionnaire in hand), then outputs the proportion of surveys with the inconsistency per enumerator and the pseudo-anonymized data of specific observations that require more investigation and correction by the survey team.

3. Checking for variation of key variables by enumerator

In a setting where several enumerators are surveying the same respondents, following the same protocol, once enough data has been collected there would be no reason to observe variation by enumerator if enumerators were conducting surveys in the exactly the same way. Enumerator training on survey tools and protocols is designed to build a common understanding of the survey instrument and common practices. Checking for the distribution of key variables per enumerator will allow checking for whether enumerators are administering the survey in the same way.[10] This could add value for key variables, or gate questions, which answer conditions skipping an entire set of questions.

An example of Stata code is presented in the appendix (dofile 1, line 201 to 226). The code looks at the distribution of key sections being skipped by the enumerator; the code outputs summary statistics per enumerator: share of surveys in which a given section is skipped, average number of sections skipped (out of 2 sections in this example) and total number of surveys. Number of skips and survey duration – check 2 – will be correlated.

7 CONCLUSION

Adding quality checks into the survey design of digitally enabled data collection can significantly improve the quality of survey data collected. This has obvious positive implications for the validity of the study, and it also reduces the time and resources spent on data cleaning, which can instead be invested in other activities, such as feeding results back to the communities (see Going Digital: Using and sharing real-time data during fieldwork).

Many of these quality checks depend on the survey supervisors having enough time, expertise and resources to follow up with enumerators and respondents, and invest time in the re-training of enumerators if required. The choice of which quality checks to perform depends on the purpose of the survey and should be tailored accordingly. This is why carefully thinking and planning at the survey design and planning stage is key to ensuring that the quality checks add value rather than create a burden on the respondent or the survey team. It is important to note that our aim is to work in partnership with the enumerators and researchers who form the survey team. Quality checks never replace human interaction, and any discrepancies are investigated as a team.

Using digital data collection technologies responsibly has the potential to increase data quality, even when resources are limited. We hope that, by making these simple examples available, other colleagues will build on it, unlocking greater potential for knowledge, learning and, ultimately, impact.

APPENDIX

PRE-RECORDED CONSENT FORM

Adding the number of seconds the audio file lasted into the minimum seconds column enabled the use of the speed violations count function. When the audio file for the Hindi language was played for less than 282 seconds this would flag an error within the data export. In the example below, however, we also wanted to find out how long the recording had been played for – recognizing that there is a difference between whether it has been played for five seconds, versus 280, for example. To calculate this, the use of 'calculate_here' was added with the calculation column showing the expression 'once(duration())'. This was added both before the consent question and after, with the calculation function then subtracting the before and start times '${consent_end_secs}-${consent_start_secs}' to get the total time the audio recording was played for. Colours shown in the table below as per SurveyCTO .xls forms.

type	name	label:english	calculation	media:audio:hindi	minimum_seconds
calculate_here	consent_start_secs		once(duration())		
select_one yesno	consent_india	If you have no further questions and would like to participate in the survey, I will record your consent to participate, and we will start the survey. Would you like to continue?		Survey Intro Hindi.wav	282
calculate_here	consent_end_secs		once(duration())		
speed violations count	consent_violation				
calculate	consent_time_total		${consent_end_secs}-${consent_start_secs}		

PRELOADING AND FILTERING UNIQUE IDENTIFIERS

Using the SurveyCTO search function under the appearance column, the lists of geographical identifiers (region, district, village) and household IDs to select from are pulled out from the preload file and attached to the form. Using the 'matches' feature of the search function, each list is filtered out based on the previous selection, so that the enumerator selects household ID among the list of household IDs of respondents from the region, district and village selected. The code allows for flexibility in case there is a need to conduct the survey in areas or households that were not pre-identified, through the 'other' options for district, village and household ID.

Survey sheet

type	name	label:english	appearance	constraint	relevance
select_one regionlist	region	Which region are you in?	search('preload')		
select_one districtlist	district	Which district are you in?	search('preload', 'matches', 'Re-gion_id', ${region})		
text	districtoth	Please write the name of the district you are in			${district} >= 98 and ${district} <= 99
select_one villagelist	village	Which village are you in?	search('preload', 'matches', 'Dis-trict_id', ${district})		
text	villageoth	Please write the name of the village you are in			${village} >= 98 and ${village} <= 99
select_one hhlist	hhidA	Please select the household you are visiting in the list	search('preload', 'matches', 'Vil-lage_id', ${village})		
text	hhidAoth	Please enter manually the HH ID, in case the respondent is not in the list			${hhidA} = "Other"
note	notehhidA	The household ID is ${hhidA}			${hhidA} != "Other"

Choices sheet

list_name	value	label:english
regionlist	Re-gion_id	Region
districtlist	Dis-trict_id	District
villagelist	Vil-lage_id	Village
hhlist	HHID	Head_FM

Preload file structure

Region	Re-gion_id	District	District_id	Village	Village_id	HHID	Head_FM
Region 1	1	BB	1	ZZZ	1	1.01.01	Ayad (1.01.01)
Region 1	1	BB	1	ZZZ	1	Other	Other
Region 1	1	Other	98	Other	98	Other	Other
Region 2	2	AA	2	XXX	2	2.02.01	Jenny (2.02.01)
Region 2	2	AA	2	XXX	2	Other	Other
Region 2	2	Other	99	Other	99	Other	Other

GENERATING AD-HOC UNIQUE IDENTIFIERS

Using household ID following the format '*Region ID.Village ID.Enumerator ID.Sampling order*', region and village IDs are generated when the geographical information is selected in the list, and so is enumerator ID. The only information to enter manually is the sampling order, after having sampled households from hard-copy lists. The calculation function then generates the household ID by concatenating the region ID, village ID, enumerator ID and sampling order, ensuring that each of these is made of two digits. The note then displays the ID to the enumerator.

type	name	label:english	constraint	relevance	required	Calculation
select_one region	region	What is the name of the region			yes	
select_one village	village	What is the name of the village			yes	
text	villageoth	Please specify the name of the village		${village} =99	yes	
select_one enumerator	enumerator	Please select your name:			yes	
begin group	tracking					
integer	sampling	What is the sampling order of this household?	.>0		yes	
calculate	hhid					concat(if(${region}<10,concat(0,${region}),${region}),".", if(${village}<10,concat(0,${village}),${village}),".", if(${enumerator}<10,concat(0,${enumerator}),${enumerator}),".", if(${sampling}<10,concat(0,${sampling}),${sampling}))
note	notehhid	This respondent's ID is ${hhid}				
end group	tracking					

HOUSEHOLD ROSTER INTERVIEW FLOW

The household roster coding ensures that the workflow used on the paper version of the form can be continued in the digital format. The hhnumber field is limited to ensure that the answer is between 1 and 19, although this maximum figure can be changed depending on the context. This is done by using the constraint expression '. < 20 and . > 0'. This number is then used to determine how many times the next set of questions is repeated in the roster. Firstly, the name of each household member is asked and the calculate row is used to remember the position that the names are given. Thus the question appears 'What is the name of household member #1?'. The number increases up until the figure input into the hhnumber field is reached (up to a maximum of 19). 'Indexed-repeat' is used in the calculation fields for each household; this assigns the the number to the name so that the survey knows the order. The use of hhnumber in the choice_filter column creates the list of household names for enumerators to easily select who is the person being interviewed and who is the household head. This saves the names from being typed in for a second time. The next repeating section again uses hhnumber to repeat the set of questions based on the number in the household. However, using the indexed-repeat function as a calculation ensures that the right name from the household is pulled through.

This avoids confusion, particularly with larger households where the order in which the names were given can be forgotten. There, when the question states, ‘what is ${namefromearlier} gender?’, it will change the ${namefromearlier} into the name that was entered in the sequence earlier. Having ‘table’, in the appearance column, brings up a summary table at the bottom of the form page.

type	name	label:english	constraint	Appearance	calculation	repeat_count	choice_filter
integer	hhnumber	Please can you first tell me how many people normally sleep together under the same roof in this household for the last 12 months?	. < 20 and . > 0				
begin repeat	names	Household names				${hhnumber}	
calculate	namenumber				index()		
begin group	page3	Household names					
text	name	What is the name of household member #${namenumber}?					
end group	page3	Household names					
end repeat	names						
calculate	name1				indexed-repeat(${name}, ${names}, 1)		
calculate	name2				indexed-repeat(${name}, ${names}, 2)		
calculate	name3				indexed-repeat(${name}, ${names}, 3)		
calculate	name4				indexed-repeat(${name}, ${names}, 4)		
calculate	name5				indexed-repeat(${name}, ${names}, 5)		
calculate	name6				indexed-repeat(${name}, ${names}, 6)		
calculate	name7				indexed-repeat(${name}, ${names}, 7)		

calculate	name8				indexed-repeat(${name}, ${names}, 8)		
calculate	name9				indexed-repeat(${name}, ${names}, 9)		
calculate	name10				indexed-repeat(${name}, ${names}, 10)		
calculate	name11				indexed-repeat(${name}, ${names}, 11)		
calculate	name12				indexed-repeat(${name}, ${names}, 12)		
calculate	name13				indexed-repeat(${name}, ${names}, 13)		
calculate	name14				indexed-repeat(${name}, ${names}, 14)		
calculate	name15				indexed-repeat(${name}, ${names}, 15)		
calculate	name16				indexed-repeat(${name}, ${names}, 16)		
calculate	name17				indexed-repeat(${name}, ${names}, 17)		
calculate	name18				indexed-repeat(${name}, ${names}, 18)		
calculate	name19				indexed-repeat(${name}, ${names}, 19)		
begin group	page4	Identifying household member					
select_one hhmember	interviewee	Identify the person interviewed					${hhnumber}>=filter
select_one hhmember	headhh	Identify the head of the household					${hhnumber}>=filter
end group	page4	Identifying household member					
begin repeat	hhroster	Household roster		table		${hhnumber}	
calculate	namefromearlier				indexed-repeat(${name}, ${names}, index())		
begin group	page5	${namefromearlier} details					

select_one gender	gender	What is ${namefromearlier} gender?					
integer	age	How old is ${namefromearlier}?	. < 130				
select_one hhrelationship	relationship	How is ${namefromearlier} related with you?					
end group	page5	${namefromearlier} details					
select_one school	schoolqualification	What is the highest grade completed by ${namefromearlier}?					
select_one work	work	Would ${namefromearlier} be sufficiently fit and able to do domestic or livelihood work NOW if they wanted to?					
end repeat	hhroster						

HARD CHECKS – DOUBLE-ENTRY AND CONSTRAINTS

Using the constraint function, the consistency of two entries of the same field is checked. In the case of the two entries not matching, a message is displayed 'The two entries do not match! Please go back, check, and make the necessary correction.' to the enumerator's attention.

type	name	label:english	hint:english	constraint	Constraint message:english
integer	huthouseroomsnow	**How many rooms are there in your compound, which are used for sleeping for your household members?**	*Do not include kitchen and bathroom/toilet areas.*	. >= 0 and . <= 50	
integer	huthouseroomsnow2	**Please re-enter**	*How many rooms are there in your compound, which are used for sleeping for your household members? Do not include kitchen and bathroom/toilet areas.*	. = ${huthouseroomsnow}	The two entries do not match! Please go back, check, and make the necessary correction.

Similarly, a second entry of the survey unique identifier could be performed at the end of the survey, using the constraint function. If the two entries do not match, a message is displayed to the enumerator ‘The Household ID does not match with the one entered at the beginning of the survey!’ so that the enumerator can check on tracking tools and correct.

type	name	label:english	hint:english	con-straint	constraint message:english
text	hhid2	Enumerator please enter the household ID again; kind reminder that this ID starts by ${villenum} :	Reminder: the Household ID follows the format: "Village ID.Enumerator ID.Date.Survey order"	. = ${hhid}	The Household ID does not match with the one entered at the beginning of the survey!

CALCULATION FUNCTIONS 1

The calculation code here uses a repeating section to pull through a csv file that holds the income sources list. Using this csv format means that it is easier to simply change the options in the csv file from project to project rather than having to change the raw coding. The begin repeat row and calculate rows pull the csv income source information through in the right order to appear in the questions. This repeat and calculate is repeated again for the second set of repeat questions, named incomesource_repeat2, with the relevance ‘${incomesource_selected2}=1’ meaning only those who selected that they had earned income from any of the sources in the last 12 months. In this second repeated group, information about monthly and yearly income is obtained for each selected source. All the values are then summed up in calculateincomeyeartot with the calculate function ‘sum(${incomeyear_})’. The respondent is then asked to confirm if the total estimate is correct, and if not, to provide a more accurate estimate.

type	name	label:English	constraint	Relevance	calculation	repeat_count
note	noteincomesource	Now I will ask you some questions about sources of income in which you and other members of your household have engaged in, during the past 12 months.				
begin repeat	incomesourcegroup	Source of income				int(pulldata('incomelist', 'incomecount', 'incomeid_key', '1'))

calculate	incomesource_id1				index()	
calculate	incomesource_name1				pulldata('incomelist', 'incomename:english', 'incomeid_key', ${incomesource_id1})	
begin group	page8	Source of income				
select_one yesno	incomesource_selected_	In the last 12 months, did you or anyone in your household earn any income from ${incomesource_name1}?				
select_one yndk	incomesource_selected2014_	In 2014, did you or anyone in your household earn any income from ${incomesource_name1}?				
end group	page8					
end repeat	incomesourcegroup					
begin repeat	incomesource_repeat2	Crop production				int(pulldata('incomelist', 'incomecount', 'incomeid_key', '1'))
calculate	incomesource_id2				index()	
calculate	incomesource_name2				indexed-repeat(${incomesource_name1}, ${incomesourcegroup}, ${incomesource_id2})	
calculate	incomesource_selected2				indexed-repeat(${incomesource_selected_}, ${ incomesourcegroup}, ${incomesource_id2})	
begin group	page8a	${incomesource_name2}		${incomesource_selected2}=1		

integer	incomemonth_	What was the total household income from ${incomesource_name2} in the LAST MONTH?	. >= 0			
integer	incomeyear_	What was the total household income from ${incomesource_name2} in the LAST 12 MONTHS?	. >= 0			
end group	page8a					
end repeat	incomesource_repeat2					
calculate	calculateincomeyeartot				sum(${incomeyear_})	
select_one yesno	incomeyearconfirm	Would you consider JOD ${calculateincomeyeartot} to correctly represent the total amount earned by your household in the last year?				
integer	incomeyearrevised	How much would you consider to be the amount received by your household in the last year?	. >= 0			

CALCULATION FUNCTIONS 2

Using the SurveyCTO, calculations and relevance functions were added into the design to prevent the enumerator from continuing if the total did not equal to the desired number. This was done by adding a weighting calculation that added the five dimensions together and then a required note field stating ‘Tokens add up to ${weighting_calculation} but they should add up to 15. PLEASE GO BACK AND CHANGE’. This note included relevance, which ensured that it only appeared when the total was not 15 (expressed as ‘${weighting_calculation} != 15’ in the code). The question was also required, meaning that enumerators had to go back and correct the answers before being able to proceed with the interview.

type	name	label	constraint	constraint message
begin group	weighting	Importance Exercise		
note	weighting_exercise	How many tokens did the respondent allocate?		
integer	contingency_re-sources	Using stored resources and calling on help from family and friends		
integer	livelihood_viability	Having other sources of income		
integer	social_institutional	Community groups, leaders in the community, and the gov-ernment		
integer	integrity_environ-ment	Keeping the water and land in good condition		
integer	innovation_potential	Trying new things	${contingency_resources}+${livelihood_via-bility} +${social_institutional}+${integrity_environ-ment} +${innovation_potential}=15	Tokens do not add up to 15. Please go back and change.
end group	weighting			

CALCULATION FUNCTIONS 3

The code below demonstrates a soft check where enumerators could proceed with the survey even if the calculation did not add up to 100%. To do this, the calculation field added together all the integer answers using the code shown below. A note field was then added using the relevance, '${percentageincometotal} != 100', which told the note field to only show if the answer did not add up to 100%. This note field remained an optional field in this instance and therefore enabled the enumerator to continue. This could easily be changed into a hard check by marking the field as required and thereby ensuring the enumerator went back to correct the answer.

Type	Name	label	Relevance	Calculation
begin group	page9	Percentage of income		
note	percentageincome	In the last 12 months, what percentage of your total household income came from:		
integer	share_farming	Farming activities		
integer	share_labour	Labour and salaries (e.g. employment)		
integer	share_business	Business		
integer	share_remitt	Remittances		
integer	share_otherproduct	Other		
end group	page9			
calculate	percentageincometotal			(${share_farming}+${share_labour}+${share_business}+${share_remitt}+${share_otherproduct})
note	percentageincometotal-note	Your total household income does not equal 100% - PLEASE GO BACK AND CHANGE	${percentageincometotal} != 100	

CALCULATION FUNCTION – SOFT CHECK

Using the calculation and relevance functions, each time a household member is identified with characteristics that may need further probing (being below 20 and a widow, in this example), a message is displayed: 'Enumerator: you just filled that "Household member's name" is a widow and is below 20; please check this information before moving forward.' Using a note rather than a constraint makes it a soft check: if this information is correct, the enumerator does not make any change and moves forward with the interview. The code below also calculates the total number of such cases in the household roster (the variable youngwidow_flag), for monitoring purposes.

type	name	label:English	appearance	relevance	calculation	repeat_count
begin repeat	hhroster	Household roster				${hhnumber}
calculate	namefromearlier				indexed-repeat(${name}, ${names}, index())	
begin group	rostergenage	${namefromearlier} details	field-list			
select_one gender	gender	What is ${namefromearlier}'s gender?				
integer	age	How old is ${namefromearlier}?				
select_one marital	marital	What is ${namefromearlier}'s marital status?				
end group	rostergenage		field-list			
calculate	youngwidow	A widow is below 20			if(${marital}=3 and ${age} < 20,1,0)	
note	youngwidownote	Enumerator: you just filled that ${namefromearlier} is a widow and is below 20; please check this information before moving forward.		${youngwidow} = 1		
begin group	rosterschoolwork	${namefromearlier} details	field-list			
select_one highested	highested	What is the highest level of education ${namefromearlier} has achieved?				
select_one fitwork	fitwork	Is ${namefromearlier} able to work?				
select_one attendschool	attendschool	Did ${namefromearlier} attend school in the last week of last term?				
end group	rosterschoolwork	${namefromearlier} details	field-list			
end repeat	hhroster					
calculate	youngwidow_flag	Young widow - Number of cases in the roster			sum(${youngwidow})	

RANDOMIZING QUESTIONS

Below is the code used to perform the A/B testing. This is implemented by using the relevance '${randomtimeAB} <= 0.50' on the two question modules with a calculation 'once(random())' to ensure that respondents were randomly assigned to either group A or group B.

type	Name	label:English	relevance	calculation
calculate_here	startconsent			once(duration())
calculate	randomAB			once(random())
select_one yesnoconsent	consentA	Good morning/afternoon. My name is _____. I am conducting a survey on behalf of OXFAM. OXFAM is an international organization working to develop long-lasting solutions against poverty and promote campaigns for social change. OXFAM is carrying out this survey to evaluate a former OXFAM project and to help understand about the lives of women in this community. I would like to request your participation in a short interview about women's empowerment. Please be aware that no special support will come to your household as a result of your responses to the questions. Any information you provide will be used for research and evaluation purposes. Data will not be shared in a way where you or any other household member could be identified. If you have concerns after the survey, you can withdraw your response at a later date by contacting the offices of OXFAM. Are you willing for us to spend approximately one hour with you carrying out an interview?	${randomAB} <= 0.50	

select_one yesnoconsent	consentB	Good morning/afternoon. My name is _____. I am conducting a survey on behalf of OXFAM in collaboration with other local partner organizations (LET, ATFD, AFTURD). OXFAM is an international organization working to develop long-lasting solutions against poverty and promote campaigns for social change. OXFAM is carrying out this survey to evaluate a former OXFAM project and to help understand about the lives of women in this community. I would like to request your participation in a short interview about women's empowerment. Please be aware that no special support will come to your household as a result of your responses to the questions. Any information you provide will be used for research and evaluation purposes. Data will not be shared in a way where you or any other household member could be identified. If you have concerns after the survey, you can withdraw your response at a later date by contacting the offices of OXFAM. Are you willing for us to spend approximately one hour with you carrying out an interview?	${randomAB} > 0.50	

AUTOMATED QUALITY CHECKS WITH STATA

```
/*****************************************/
/****Do-file 1 – Monitoring data quality****/
/*****************************************/


// Upfronts
      clear
      clear all
      set more off, perm

      local day 27.04 // Monitoring date - automatically rename the output excel files with this date


// Set paths

      global home "C:\Users\apretari1\2_ER\Data"


// Run CTO import dofile

 do "$home\import_er_household_final.do"
 do "$home\import_er_individual_final.do"


// HH level dataset

            u "$home\ER_HOUSEHOLD_FINAL.dta", clear

/***************************/
/*Survey progress monitoring*/
/***************************/

* 1. Check for the number of completed surveys

                  gen complete = (surveystatus == 1)
                  bys enumerator: egen nbcomplete = total(complete)
```

```
                        label var nbcomplete "Number of complete surveys"

                  // Outputting summary statistics by enumerator

                  preserve

                        keep enumerator nbcomplete
                        duplicates drop
                        export excel using "$home\monitoring_`day'.xlsx", firstrow(varl) sheet("HH survey - comple-
tion") sheetreplace

                  restore

* 2. Check of the survey duration by enumerator

            // Turning surveyduration in minutes

                  destring surveyduration, replace
                  summ surveyduration

                  gen surveyduration_min = surveyduration / 60
                  summ surveyduration_min
                  summ surveyduration_min, d

                  gen hhsurveybelow15 = (surveyduration_min <= 15) // shorter than 15 minutes - if willing to set an
arbitrary threshold, at onset of the survey
                  gen hhsurveyshort = (surveyduration_min <= `r(p50)'/2) // shorter than half the median duration
                  gen hhsurveylong = (surveyduration_min >= `r(p50)'*2 & surveyduration_min != .) // longer than twice
the median duration

            // Looking at the data by enumerator

                  bys enumerator: summ surveyduration_min
                        label var surveyduration_min "Survey duration (minutes)"

            // Creating summary statistics by enumerator

                  by enumerator: egen surveydur_mean = mean(surveyduration_min)
                        label var surveydur_mean "Average survey duration"
```

```
                    by enumerator: egen surveydur_min = min(surveyduration_min)
                          label var surveydur_min "Minimum survey duration"
                    by enumerator: egen surveydur_max = max(surveyduration_min)
                          label var surveydur_max "Maximum survey duration"
                    by enumerator: egen surveydur_nb = count(surveyduration_min)
                          label var surveydur_nb "Number of completed surveys"

                    * Share of surveys by enumerator that are:
                          * below 15 min,
                          * shorter than half the median duration
                          * longer than twice the median duration
                    foreach x in below15 short long {
                          by enumerator: egen hhsurvey`x'_mean = mean(hhsurvey`x')
                                label var hhsurvey`x'_mean "Share of `x' surveys"
                          }

              // Outputting summary statistics by enumerator

              preserve

                    keep enumerator surveydur_* hhsurvey*_mean
                    duplicates drop
                    export excel using "$home\monitoring_`day'.xlsx", firstrow(varl) sheet("HH survey - duration")
sheetreplace

              restore

              // Listing the surveys that are too short and needs investigation / call back

              preserve

                    keep if hhsurveyshort == 1
                    keep hhid starttime submissiondate endtime datetimeend surveyduration_min
                    export excel using "$home\monitoring_`day'.xlsx", firstrow(varl) sheet("HH survey - too short!")
sheetreplace

              restore
```

```
* 3. Duplicates and matching of IDs

      // >> See dofile 2


/*****************************************************************/
/*Quality checks: response quality checks and enumerator variation*/
/*****************************************************************/

* 4. Checking for the distribution of continuous variables

            foreach var in var1 var2 var3 {
                  ta `var' // Manual look at distribution at onset of survey
                  summ `var', d
                  gen `var'_out = ((`var' >= `r(mean)' + 3*`r(sd)') & `var' != .) // Focus on the right-hand side of
the distribution
                        label var `var'_out "Dummy for outlier of `var'"
                  ta enumerator if `var'_out == 1 // Manual investigation
                  ta village if `var'_out == 1 // Manual investigation
                  }

            // Creating summary statistics by enumerator

            foreach var in var1 var2 var3 {
                  bys enumerator: egen `var'_out_mean = mean(`var'_out)
                        label var `var'_out_mean "Share of surveys with outlier value of `var'"
                  }

            // Outputting summary statistics by enumerator

            preserve

                  keep enumerator *_out_mean
                  duplicates drop
                  export excel using "$home\monitoring_`day'.xlsx", firstrow(varl) sheet("HH survey - outlier per
enumerator") sheetreplace

            restore
```

```
154             // Listing the surveys with outlier that needs investigation / call back
155
156             preserve
157
158                   keep if (var1_out == 1 | var2_out == 1 | var3_out == 1)
159                   keep enumerator hhid var1 var1_out var2 var2_out var3 var3_out
160                   duplicates drop
161                   export excel using "$home\monitoring_`day'.xlsx", firstrow(varl) sheet("HH survey - outlier correc-
162 tions") sheetreplace
163
164             restore
165
166
167 * 5. Checking for internal consistency
168
169             gen inconsistency = (var1 != var2) if complete == 1
170             ta enumerator if inconsistency == 1  // Looking manually
171
172             // Creating summary statistics by enumerator
173
174             bys enumerator: egen inconsistency_mean = mean(inconsistency)
175                   label var inconsistency_mean "Share of surveys with inconsistency"
176
177             // Outputting summary statistics by enumerator
178
179             preserve
180
181                   keep enumerator inconsistency_mean nbcomplete
182                   duplicates drop
183                   export excel using "$home\monitoring_`day'.xlsx", firstrow(varl) sheet("HH survey - inconsistency")
184 sheetreplace
185
186             restore
187
188             // Listing the surveys that need correction
189
190             preserve
191
192                   keep if inconsistency == 1
```

```
                  keep enumerator hhid var1 var2
                  duplicates drop
                  export excel using "$home\monitoring_`day'.xlsx", firstrow(varl) sheet("HH survey - correction")
sheetreplace

            restore


* 6. Checking for variation by enumerator

            gen skipcrop = (crop != 1) if complete == 1
            gen skiplivestock = (anylivestock != 1) if complete == 1
            gen nbskip = skipcrop + skiplivestock

            // Creating summary statistics by enumerator

            foreach var in skipcrop skiplivestock {
                  bys enumerator: egen `var'_mean = mean(`var')
                        label var `var'_mean "Share of surveys for which `var' == 1"
                  }

            bys enumerator: egen nbskip_mean = mean(nbskip)
                  label var nbskip_mean "Average number of sections being skipped"

            // Outputting summary statistics by enumerator

            preserve

                  keep enumerator *skip*_mean nbcomplete
                  duplicates drop
                  export excel using "$home\monitoring_`day'.xlsx", firstrow(varl) sheet("HH survey - skip") sheet-
replace

            restore
```

```
/*********************************************************/
/****Do-file 2 – Monitoring duplicates and match of IDs****/
/*********************************************************/



// Checking IDs

// Upfronts
      clear
      clear all
      set more off, perm

      local day 27.04 // Monitoring date - automatically rename the output excel files with this date

// Set paths

      global home "C:\Users\apretari1\2_ER\Data"

// Run CTO import dofile

 do "$home\import_er_household_final.do"
 do "$home\import_er_individual_final.do"

// Checking the IDs

            u "$home\ER_HOUSEHOLD_FINAL.dta", clear

            // Keep variables of interest for communicating with survey team

            keep region hhid hhhname enumerator consent surveystatus starttime

            // Tag duplicates

            duplicates tag hhid, gen(dupl_hh)

            // Export duplicates

            egen tot_dupl_hh = total(dupl_hh)
```

```
40              if tot_dupl_hh >= 1 {
41
42                    preserve
43
44                          keep if dupl_hh >= 1
45                          keep hhid hhhname
46                          sort hhid
47                          export excel using "$home\monitoring_`day'.xlsx", firstrow(var) sheet("HH duplicates") sheet-
48  replace
49
50                    restore
51                    }
52
53              sort hhid
54
55              // Add suffix to variables
56
57              ren * *_hh
58              ren hhid_hh hhid
59
60              // Save file for merging
61
62              tempfile check
63              save `check'
64
65
66              u "$home\ER_INDIVIDUAL_FINAL.dta", clear
67
68              // Keep variables of interest for communicating with survey team
69
70              keep region hhid indid respname enumerator consent surveystatus   starttime
71
72              // Tag duplicates
73
74              duplicates tag indid, gen(dup_indid)
75
76              egen tot_dup_indid = total(dup_indid)
77
78              if tot_dup_indid >= 1 {
```

```
79
80                      preserve
81
82                            keep if dup_indid >= 1
83                            keep indid hhid respname
84                            sort indid
85                            export excel using "$home\monitoring_`day'.xlsx", firstrow(var) sheetreplace sheet("Ind du-
86    plicates")
87
88                      restore
89                      }
90
91                sort hhid
92                merge hhid using `check'
93
94                gen doesnotmerge = (_merge != 3)
95
96                preserve
97
98                      keep if doesnotmerge == 1
99
100                     keep hhid indid enumerator enumerator_hh hhhname_hh respname starttime_hh starttime _merge
101
102                     order *_hh
103                     sort hhid indid
104                     export excel using "$home\monitoring_`day'.xlsx", firstrow(var) sheetreplace sheet("Does not merge")
105
106               restore
```

ACKNOWLEDGEMENTS

We would like to thank Jonathan Lain, Andrew Anduko (former Oxfam Impact Evaluation Advisers) and Kristen McCollum, former Impact Evaluation Intern, who contributed by developing and shaping some of the features reported in this paper. Thanks also to the consultants we have used as well as Oxfam Canada who managed the data collection activities where many of these quality checks were initially piloted. Finally, our thanks to the SurveyCTO team, who supported us in the creation of some of these survey features.

RELEVANT LINKS

E. Tomkys and S. Lombardini (2015). *Going Digital: Using digital technology to conduct Oxfam's Effectiveness Reviews*. Oxford: Oxfam GB. http://policy-practice.oxfam.org.uk/publications/goingdigital-using-digital-technology-to-conduct-oxfams-effectiveness-reviews-578816

S. Lombardini and E. Tomkys Valteri (2017). *Going Digital: Using and sharing real-time data during fieldwork*. Oxford: Oxfam GB. https://policy-practice.oxfam.org.uk/publications/going-digital-using-and-sharing-real-time-data-during-fieldwork-620432

Oxfam International (2015). *Oxfam Responsible Program Data Policy*. Oxfam International. https://policy-practice.oxfam.org.uk/publications/oxfam-responsible-program-data-policy-575950

S. Lombardini (2017). *Measuring time: Comparing questionnaire design*. [blog] https://views-voices.oxfam.org.uk/methodology/2017/01/real-geek-measuring-time-comparing-question-naire-designs

E. Tomkys and L. Eldon (2016). *Mobile Survey Toolkit*. Oxford: Oxfam GB. https://policy-practice.oxfam.org.uk/publications/mobile-survey-toolkit-617456

World Bank IE unit Wiki: https://dimewiki.worldbank.org/wiki/Main_Page

J-Pal online resources: https://www.povertyactionlab.org/research-resources/measurement-and-data-collection

NOTES

1 See the World Bank Impact Evaluation Unit wiki, for example https://dimewiki.worldbank.org/wiki/Main_Page which presents guidelines on survey practices, data collection management and monitoring, which are complementary to this publication.

2 Code will be in SurveyCTO and Stata as these are the software packages used by the team. Of course, these solutions can be used by other software e.g. R

3 Oxfam's Responsible Program Data Policy

4 Provided listing data can be sent to the server through an internet connection, the SurveyCTO 'Dataset' function allows automation of the sampling process and the generation of a preload file while the listing households is conducted.

5 Due to ethical concerns, and specifically to ensure informed consent and avoid any deception of respondents, the experiment implemented the following measures. First, the consent form explicitly stated that data would be used for research and evaluation purposes, and would not be shared in a way that any household could be identified. Second, when the field supervisors contacted village authorities to arrange permission to enter each village, they were trained to mention the names of all organizations involved. Third, if a respondent asked for more information on the organization(s) conducting the survey, enumerators were carefully trained to truthfully mention all organizations involved.

6 Or any statistical software that enables coding and performing pre-defined analysis. SurveyCTO Data Explorer is another way to run some of these checks, which does not require access to the whole dataset, but only to the key fields on which to run the checks. This can be particularly useful in dealing with different skill sets within the team, and/or with different team members having access to different information to meet confidentiality requirements.

7 For more on this, see the J-Pal's High Frequency Checks Guide (forthcoming): https://www.povertyactionlab.org/research-resources/measurement-and-data-collection

8 Any value higher than the sum of the mean and 3 standard deviations.

9 The SurveyCTO code presented in the appendix creates a variable flagging the number of household members recorded as widows and below 20 in the household roster; this variable is created in the questionnaire program to be used for monitoring.

10 Testing for statistical differences is doable but requires enough data to be collected before such tests can be performed.

This case study was written by Emily Tomkys Valteri, Alexia Pretari and Simone Lombardini. It is part of a series of papers and reports written to inform public debate on development and humanitarian policy issues.

For further information on the issues raised in this paper please email etomkysvalteri@oxfam.org.uk.

Published by Oxfam GB under ISBN 978-1-78748-307-1 in July 2018.
DOI: 10.21201/2018.3071
Oxfam GB, Oxfam House, John Smith Drive, Cowley, Oxford, OX4 2JY, UK.

OXFAM

Oxfam is an international confederation of 20 organizations networked together in more than 90 countries, as part of a global movement for change, to build a future free from the injustice of poverty. Please write to any of the agencies for further information, or visit www.oxfam.org.

OXFAM