**Cleaning protocol for SOS survey data**

Ana Casas1, Andrew R. Bell 2, Beata Kupiec-Teahan2,3, Amy R Lewis4, Jose Mendozas5,6, Simon Willcock4,7

1 School of Water, Energy and Environment, Cranfield University, Cranfield MK43 0AL, UK

2 Boston University, Boston, USA

3 Bangor Business School, Bangor University, Bangor, Gwynedd LL57 2DG, UK

4 School of Environmental & Natural Sciences, Bangor University, Bangor, Gwynedd LL57 2DG, UK

5 Pontificia Universidad Católica del Perú, Perú

6 Instituto de Estudios Peruanos, Perú

7 Net Zero and Resilient Farming, Rothamsted Research, Harpenden, Hertfordshire AL5 2JQ, UK

**Preprocessing of the data**

1. Data was stored on ODK Central server encrypted. *This has since been removed and deleted.*
2. Each week data was downloaded using RUODK package[[1]](#footnote-1)
3. Empty files were identified and removed. *This often occurred in the first and last weeks of the project.*
4. Data was merged based on task type, and form name was added. *This enabled us to create a distinction of weeks.*
5. Some variables redacted that were identified as not required for early analysis
   1. This was using redact function in R (numbers changed to 9, and text to [redacted])
   2. Phone numbers and household locations removed.
6. Data was appended to the ID KEY, and entries with no match were removed- *this accounts for enumerators practicing, early trials, and those potentially filling in who did not consent to be part of our study.*

**Protocol description for Demographics baseline**

1. Append the Demographics baseline file for the three countries.
2. Identify any duplicates on ID and keep the earliest entry based on end date – *this accounts for people submitting the demographics baseline multiple times. We decided to keep the earliest entry, as this is the entry they completed under supervision in workshops with the teams.*
3. Once we had a final document without duplicates, we checked the user type and the gender for each ID against the information from the consent forms, which had been provided and checked by the local teams. In case of any discrepancies in either gender or user type, we kept the information from the consent form and assumed there was an input error by the participant. In extreme cases where an ID had a specific gender in the consent form but they repeatedly selected a different gender in the demographics baseline, we double-checked the information with the local team.
4. We also checked the age of the participants. For Kenya and Peru, age was not supplied in the consent forms. Therefore, for each ID, we checked the age column for all the entries from the demographics baseline. If all the entries had same/similar ages, we kept the first entry – *this accounts for people who are unsure about their age*. If entries were very different, but there was a particular age that was clearly more frequent than the age stated in the first entry, we kept the most frequent age and assumed there was an input error by the participant. In cases where we were unsure, we checked the information with the local teams.

For South Africa, as the consent forms included the age, we kept the age from the consent form in case of discrepancies.

1. Once all the discrepancies were corrected, we created a final master file. This file is to be used as the baseline to cross-reference questions in the other surveys – for example, in cases where we asked questions that should only be answered by CBS users, we use the demographics master file to remove any non-users who might have answered those questions by mistake. Similarly, in the surveys where we asked about menstrual health management, we use the demographics master file as a reference to remove any male participants who might have answered those questions.

*NOTE – P027 removed as age below age limit (15 yo)*

**Protocol description for the rest of the surveys**

1. Append the data for all the countries.
2. Identify any duplicates on ID+Form version and keep the earliest entry based on end date – *this accounts for people submitting forms multiple times when they don’t receive a confirmation message immediately.*
3. Find all the out-of-range/invalid entries:
   1. The start time is after/on the release date (start\_date in the survey) AND before/on the expiration date AND
   2. The end time is before/on the expiration date.
4. Identify whether the out-of-range entries fit within the active period for any other form version (excluding the baseline), where the active period is defined as any date between the release date and the expiration date for a survey, both inclusive. ***Note that active periods for the same form version might be different for different countries***.

To do this, check whether the start time and end time of that entry are within the release date-expiration date range for any other form version, and check whether there is an existing valid entry for that other form version - we will only assign a new form version number to the out of range entries if there are no valid entries for that new form version in the survey already. T*his accounts for people filling in the wrong form version in a particular week, but still filling in the survey, so it’s still a valid entry, as long as they didn’t fill in two surveys on the same week.*

1. Remove any duplicates in the out of range entries where for the same ID and Form Version we’ve assigned more than one potential new form version. *This accounts for situations where an entry could fit under two different form versions because there’s an overlap in dates in the forms (e.g. in the example shown below, if we have a submission for form version 1 but it was filled in on the 3rd of Feb, it’s out of range for Form 1 but it could fit under form 2 or form 3; if there are no valid entries in the survey for Form 2 or Form 3, the cleaning protocol will assign Form 2 to the out of range entry.*

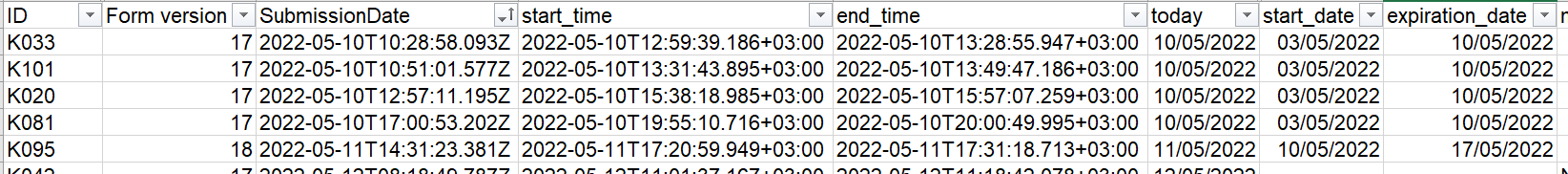
A screenshot of a computer

Description automatically generated with low confidence

1. Remove any duplicates in the out of range entries where the same new form version has been assigned to different ID&Form version combinations. *E.g. This accounts for people filling in multiple surveys in the same day. For example, if someone filled in surveys 2, 3, 4, and 5 on the same day, and that day corresponds to the timeframe for form version 10, we would only keep the entry for form 2 and would reassign it to form 10, the other entries would be removed.*
2. If a match is found in 4-6), correct the form version, release date and expiry date, and correct the status of the entry from out of range/invalid to valid/in range.
3. Filter out all the invalid entries.

**Detail of the data**

The screenshot below shows an example of the columns found in the dataset:



The columns used for the cleaning protocol are described below:

* Form version corresponds to the week number when participants should fill in the form.
* SubmissionDate is the date when the survey reached the server. In cases when participants have internet connection issues or they don’t have any money available in their phones, these dates may have a delay with respect to when participants actually completed the survey.
* Start\_time and end\_time corresponds to the times when participants started and finished filling up the survey.
* start\_date is the date when the survey was released – referred to as “release date” in this document. The survey shouldn’t be available before this date.
* Expiration\_date is the date when the survey closes. The survey shouldn’t be filled in after this date.

1. Mayer, Florian Wendelin. (2020, Nov 19). ruODK: An R Client for the ODK Central API (Version 4.3.1). Zenodo. https://doi.org/10.5281/zenodo.3953158 [↑](#footnote-ref-1)