In review

An XLSForm method for fast offline collection of full-community social network data

Augusto Dalla Ragione[1], Jeffrey Andrews[1], Marina Elodie Sedjro Djekinnou[1,2], Doukpè Nadège Adikodé[1,2,3], Iranlowo Abakpin Elodie F Lawin[1,2], Tossou Jeandro Simplice Kakpo[1], Tcheoun Jean Akpo[1,2,4], As'Ad Olawale Ichola Assani[1,2], Daniel Redhead[1,5,6], and Cody T. Ross[1]

Abstract

The collection of social network data from small-scale communities has been an increasingly important part of long-form research projects on resilience, adaptation, and well-being. Various tools are available to facilitate such data collection protocols. Roster-based methods yield high quality data, but typically scale poorly with sample size. Name-generator-based methods, in contrast, scale well with sample size; however, they entail a large amount of highly error-prone post-processing after data collection in order to link elicited names to unique identifiers. Here, we introduce a hybrid XLSForm method for collection of social network data using name-generator prompts that are dynamically linked to a full-community photograph roster. The name-generator prompts result in fast data collection *in situ*, and a confirmation step using photographs certifies that elicited names refer to the correct individuals. We provide both an example XLSForm template, and an R package, `XLSFormulatoR`, that facilitates the automatic creation of larger multi-question XLSForm social network surveys. We present a complete methodological workflow using KoboCollect to implement our survey design. To validate our methodology empirically, we conducted a network study in two rural Beninese communities ($n = 614$): we compare the friendship networks elicited using our app with friendship networks elicited from the same individuals using a roster-based design.

Introduction

Social network data have become increasingly important in field-oriented disciplines, such as anthropology. Harnessing such data allows researchers to study important aspects of social life, including cooperative norms (Starkweather et al. 2023; Pisor et al. 2020), food sharing relationships (Koster and Leckie 2014; Gettler et al. 2023), religiosity (Power 2017), fertility preferences (Colleran 2020), and friendship (Redhead et al. 2023a). Given the size of the social settings where anthropologists usually work, it is often possible to aim for the collection of full (sociocentric) network data, as opposed to egocentric networks, where the information on social ties comes from only a sample of individuals within the larger social group. The type of social network being collected (e.g., sociocentric versus egocentric) may in turn entail different data collection strategies (Cornwell and Hoagland 2014), with trade-offs between data quality and interview duration.

Both roster-based and free-recall/name-generator-based survey tools are commonly employed (Ross and Redhead 2021). In roster-based methods, the researcher presents the respondent with images or names of all individuals in the group, and asks each person about their social ties with each and every possible alter. In name-generator-based methods, the respondent can freely list any individual as part of their network (Marsden 1990). Roster-based methods yield high quality data, but typically scale poorly with sample size (but see Ross and Redhead 2023, for scalable photo-roster-based methods). With large rosters, the duration of each interview may grow to be unacceptable. Or, the maximum number of network layers that can be elicited during an interview may be small in larger communities. Name-generator-based methods, in contrast, scale well with sample size; however, they entail a large amount of post-processing after data collection is complete. It has also been noted that name-generator methods often have higher risk of: 1) respondent recall bias and forgetting relative to "recognition-based" or roster methods (Sudman 1985; Brewer 2000), and 2) errors in the post-processing phase, where reported nominations have to be cross-checked and linked to unique individuals in the sample (Ross and Redhead 2021).

Our goal here is to remove the error-prone post-processing phase of free-list social network data collection, by developing a workflow to collect name-generator-based

[1]Department of Human Behavior, Ecology and Culture, Max Planck Institute for Evolutionary Anthropology, Leipzig, Germany.
[2]African School of Economics, Abomey-Calavi, Benin.
[3]Department of Economics, University of Ottawa, Canada.
[4]Department of Economics, University of Warwick, Coventry, United Kingdom.
[5]Department of Sociology, University of Groningen, The Netherlands.
[6]Inter-University Center for Social Science Theory and Methodology, University of Groningen, The Netherlands.

Corresponding author:
Augusto Dalla Ragione
Email: augusto_dalla_ragione@eva.mpg.de

data, while simultaneously integrating a photo-based identity confirmation step performed by respondents themselves. The approach combines some of the features of roster-based methods (i.e., use of a photo-roster, and the absence of a post-processing record-linkage phase), while retaining the fundamental strengths of the name-generator method (i.e., its favorable scaling with sample size and the fact that it minimizes interviewee fatigue). Nevertheless, the approach we outline here is a recall-based, rather than recognition-based method, as the photograph roster is used to confirm the identity of freely-listed ties, rather than elicit ties directly. We design a simple network-data questionnaire using the XLSForm standard (Open Data Kit Community 2024), and implement it on the free KoboToolbox platform and its associated KoboCollect app. We also include an R package, `XLSFormulatoR`, that can be used to automate the creation of the XLSForm used by KoboCollect. The template forms and the synthetic data presented here are included in the supplementary materials, and the R package is available at: `https://github.com/ADR1993/XLSFormulatoR`.

A brief tutorial

KoboToolbox

KoboToolbox is an open access platform for collecting and managing data (Kobo Toolbox 2024). KoboToolbox maintains an app called KoboCollect, which can be used for offline data collection. After data collection, finalized interview forms can be submitted to the KoboToolbox server from the app, as soon as the surveyor has access to an internet connection. Interview forms are created via the XLSForm standard, and uploaded to the online platform, along with any associated materials (e.g., photograph rosters) that may be needed to support data collection using the KoboCollect app.

Before beginning

We assume that the end-user already has a KoboToolbox account (standard accounts are free) and has already opened a new project. To reproduce the steps in our tutorial, the end-user will also need to have the `names.csv` file, the `photos.zip` file, and a `network_collect.xlsx` file prepared for their survey. Example materials are included in the supplementary appendix to this article, and they can be modified by the end-user as needed.

Creating a KoboToolbox project

After logging in, the end-user can click on the blue button called "New", which will open a prompt to create a new project. A pop-up box will appear, with four options. The user must select "Upload an XLSForm". The user then simply uploads the `network_collect.xlsx` file for their project, and KoboToolbox will create the final data collection app (see Figure 1).

Uploading file attachments

Once a KoboToolbox project is created as described above, the user must add the supplementary files upon which the data collection app depends. These files can be attached to

Figure 1. Creating a new KoboToolbox project.

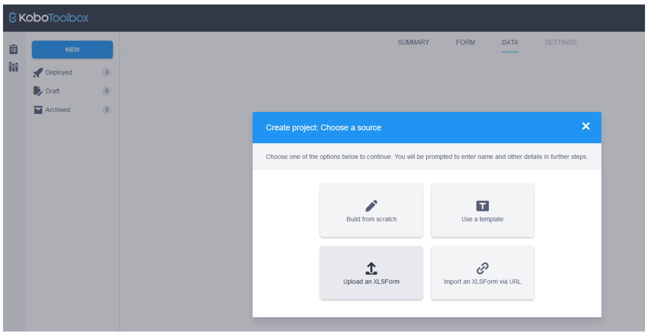
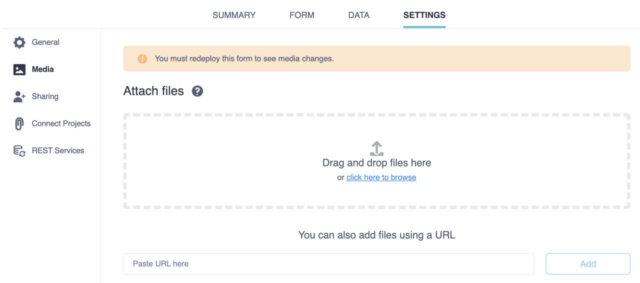


Figure 2. Attaching files in the Media section of the project.



a KoboToolbox project by uploading them in the "Media" section of the "Settings" tab. See Figure 2. The csv file containing the list of names and ID codes, `names.csv`, and the unzipped version of the photos included in the `photos.zip` folder all need to be added to the project from here. After uploading the attachments, the user has to deploy the project so that the updated form can be pushed to all the KoboCollect devices connected to the project.

Creating an XLSForm through R

Creating XLSForms by hand is a highly technical process, and so it will typically be easier for end users to compile a social network questionnaire using our R package, `XLSFormulatoR`. We will provide an example tutorial below for eliciting friendship and sharing networks, as they are collected according to the ENDOW project's protocol: `https://endowproject.github.io/`.

Installation and loading of `XLSFormulatoR` is simple. Just run three lines of code from R:

```
library(devtools)
install_github("ADR1993/XLSFormulatoR")
library(XLSFormulatoR)
```

Then create a labeled list of questions/prompts to be read to each respondent. In this case, we are only considering friendship and double-sampled resource transfer networks (i.e., where there are separate prompts for giving and receiving resource; Ready and Power 2021), so the list will only contain three elements. However, the process can accommodate an arbitrary number of labeled questions.

```
questions = list(
"friendship" = "Please list the names
   of your closest friends.",
"giving" = "Please list the names of
   the people who you have given money
    to in the last 30 days.",
"receving" = "Please list the names of
   the people who have given you money
    in the last 30 days.")
```

Then run the following code to compile the final XSLForm, which includes the basic header questions, and repeat-group network questions for each prompt in the list:

```
compile_xlsform(
 questions,
 type = "jpeg",
 photo_confirm = "all")
```

The function will save a file `network_collect.xlsx` in R's current working directory. The argument `photo_confirm` controls whether or not the photo confirmation step (see Figure 4) is deployed. This argument can be set to: "all" in order to always deploy the photo confirmation step, "only_focal" to photo-confirm only the identity of the focal respondents, or "none" to never confirm identities using photographs. This last option is useful if the researcher has a name roster, but no photo roster. The file outputted by this function can then be uploaded, as shown in Figure 1, to KoboToolbox to build the final survey app.

Visualizing the KoboCollect workflow

After beginning the survey form, the researcher will be prompted to ask the respondent for their name (see Figure 3). The app will display a search bar, with minimal autocomplete, so that the respondent's name can be quickly found in the community roster. After selecting a name, the app will display the respondent's photo, so that the researcher can verify that the respondent's name has been correctly identified.

Then, the first network question will be displayed as shown in Figure 4. The respondent will provide the names of their network connections, and these will be entered using the same search bar with minimal auto-complete shown earlier. After a name is selected, a photo confirmation step is completed, as shown in the latter frames of Figure 4. After all nominations for a given question have been supplied and confirmed, the app will progress to the next question/network layer, and the researcher will repeat the process until the respondent has responded to all question in the survey. At this point, KoboCollect will prompt the researcher to save the interview.

Data download and visualization

After submitting interviews collected via KoboCollect, end-users can download the full database from the KoboToolbox platform via the "Downloads" page in the "Data" section. Users must choose "XLS" as export type, and "XML" as format for values and headers. The resulting export file will contain a main sheet named after the project, plus additional sheets for each repeat group (i.e., each question/network layer) contained in the form. Such data are already organized in an edge-list-like file structure based on individual IDs, sheet-by-sheet.

In order to process the data with standard tools, like `igraph` (Csardi and Nepusz 2006), `XLSFormulatoR` provides a convenience function for compiling the multi-sheet KoboToolbox dataset into a simpler, flat, question-labeled edge-list. Simply download the data from KoboCollect, then read it into R using:

```
path = file.choose()
d = kobo_to_edgelist(path, question,
     save = NULL)
```

The function will save the edge-list in memory as an object, `d` in this case. If desired, the argument `save` can be set to a string like "TarsonisNetworkData", in which case a file called `TarsonisNetworkData.csv` will be saved in R's current working directory.

Field-testing the workflow

In order to field-test our social network data collection protocol, we deployed the app in two rural communities in the Mono and Atlantique departments in Benin (West Africa) to study the structure of friendships, resource transfers, and social support networks. The subsistence of these communities is largely based on artisanal fishing in coastal lagoons, with some individuals involved in wage labor, tourism, or small-scale salt production. A team of six local research assistants collected the data, administering the survey in the local languages. Figure 5 displays the Atlantique friendship network, as collected through the name-generator method on KoboCollect. Figure 6 displays the degree distribution of the Atlantique community as elicited with both roster and free-list methods.

We assess the quality of the data collected using our KoboCollect app by comparing friendship network data elicited through the workflow described here, with comparable friendship network data collected in the same populations using a physical photo-roster-based method (see: Ross and Redhead 2023) two months prior. In Atlantique, $n = 362$ participants completed the KoboCollect-based survey and $n = 384$ completed the roster-based survey. A large fraction of respondents, $n = 352$ completed both surveys. In Mono, $n = 203$ participants completed the KoboCollect-based survey and $n = 212$ completed the roster-based survey. Again, a large fraction of respondents, $n = 195$ completed both surveys. For the comparisons discussed below, we report on only the sub-networks comprised of individuals who completed both surveys. We focus on measuring the *concordance* between reported friendships $i \rightarrow j$ and $j \rightarrow i$, which we define here by calculating the fraction of ties for which $F_{[j,i]} = 1$, conditional on $F_{[i,j]} = 1$, where F is the friendship adjacency matrix.

In Atlantique, the roster-based method led to a greater number of elicited ties: $4,691$, or about $\mu = 13.4$ ties per person on average ($\sigma = 16.6$). The free-list method led to 754 ties in total, or about $\mu = 2.1$ ties per person on average ($\sigma = 1.8$). The concordance of the roster method was fairly high, with 946 of $4,691$ (20%) friendship ties being nominated by both members of a dyad. Concordance

Figure 3. Focal ID. In the first two frames, the researcher searches for the respondent's name using a search bar with minimal auto-complete. In the third frame, the tablet displays the selected name. In the fourth frame, a final photo confirmation step is completed, so that the researcher can confirm that the respondent is indeed the person in the photo.

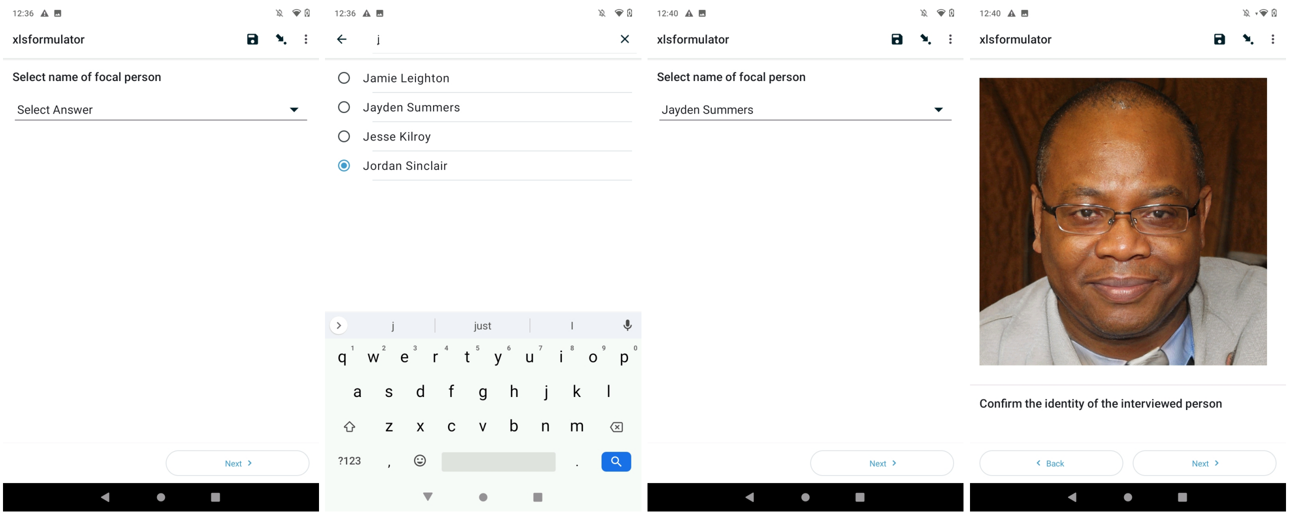
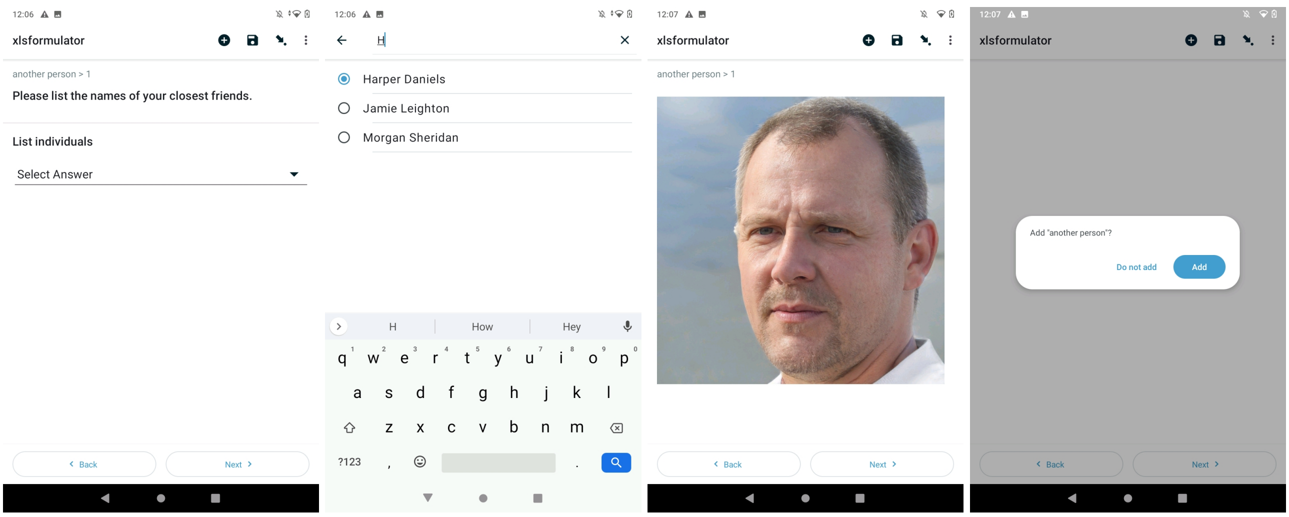


Figure 4. Network data. In the first frame, the network question is displayed. In the second frame, the researcher searches for elicited names using a search bar with minimal auto-complete. In the third frame, the tablet displays displays the photo of the person nominated as a network tie, so that the researcher can verify their identity. In the fourth frame, the researcher is promoted to add additional nominations. In this case, if the respondent has any other friends, the researcher should click "add" to record those ties, using the same process described here. Otherwise, if the list is complete, the researcher should click "do not add", and the form will skip to the next question/network layer.



was slightly higher using the free-list method, with 204 of 754 (27%) friendship ties being nominated by both members of a dyad. Lastly, we compare nodal degree across methods: that is, we use nodal in-degree calculated via the free-list data to predict the same measure calculated from the roster data. We find highly significant associations for both in-degree ($\beta = 2.33, R^2 = 0.20, p < 0.001$) and out-degree ($\beta = 1.89, R^2 = 0.04, p < 0.001$).

In Mono, the roster-based method led, surprisingly, to fewer elicited ties: 484, or about $\mu = 2.5$ ties per person on average ($\sigma = 5.5$). The free-list method led to $1,616$ ties in total, or about $\mu = 8.3$ ties per person on average ($\sigma = 5.8$). The concordance of the roster method was fairly low, with only 66 of 484 (13%) friendship ties being nominated by both members of a dyad. Concordance was much higher using the free-list method, with 742 of $1,616$ (46.0%) friendship ties being nominated by both members of a dyad. Comparing nodal degree across methods, we again find significant associations for both in-degree ($\beta = 0.25, R^2 = 0.45, p < 0.001$) and out-degree ($\beta = 0.16, R^2 = 0.02, p < 0.016$).

Lastly, we compare the Beninese results with publicly available data from a coastal Afrocolombian community, whose subsistence is also based on artisanal fishing (Pisor et al. 2020). In Colombia, however, free-list name-generator data were collected using pencil-and-paper, and no photo-verification step was completed. Thus, record-linkage was performed using only the elicited names. The record-linkage process for the multi-question survey took several days of intensive work, and involved linking unique ID codes to names by hand, and iteratively scanning the full edge-list multiple times in an attempt to ensure that the number of duplications (i.e., the same person being assigned multiple unique ID codes) and collisions (i.e., multiple people being assigned the same unique ID code) was as small as possible. In total, $n = 149$ respondents participated, and made 262 friendship nominations, about $\mu = 1.8$ ties per person on average ($\sigma = 1.7$). Concordance was comparable with the Beninese data, with 66 of 262 (25%) friendship ties being nominated by both members of a dyad.

In general, levels of concordance in human self-report network data are normally low. Early work, often based on small samples, found concordance in network nominations to range as high as 40 to 60% (Marsden 1990). However, in samples with greater coverage, concordance estimates average only about 10% (Ready and Power 2021). Comparing against this baseline, the concordance estimates for the Beninese field-sites are quite high, especially using our KoboCollect protocol. Although, one might expect full concordance in "high quality" data, inconsistent reports do not necessarily reflect data collection errors or even inaccuracies in recall (Ready and Power 2021). Individuals might have different perceptions of a relationships, and/or different thresholds for making nominations (Carley and Krackhardt 1996). Nevertheless, forgetting (Brewer 2000), recall biases (Sudman 1985), impatience with long network questionnaires, as well as simple heterogeneity in nomination rates, are know to affect estimates of concordance/reciprocity, and tools such as the social relations model (Kenny and La Voie 1984) and latent network models (Redhead et al. 2023b) can be used to derive estimates of reciprocity that at least partially account for these factors.

Thus, as a final check on data quality, we used the social relations model (Kenny and La Voie 1984; Back and Kenny 2010) to estimate dyadic reciprocity after accounting for inter-individual variability in making and receiving nominations (see Figure 6 for the degree distributions for both sites, for both network elicitation methods). In the Atlantique site, dyadic reciprocity in friendship nominations was quite high: $\rho = 0.91$ $(95\%CI : 0.85, 0.97)$ when estimated using the free-list data, and $\rho = 0.76$ $(95\%CI : 0.68, 0.84)$ when estimated using the roster data. In the Mono site, dyadic reciprocity in friendship nominations was $\rho = 0.95$ $(95\%CI : 0.91, 0.98)$ when estimated using the free-list data, and $\rho = 0.85$ $(95\%CI : 0.73, 0.95)$ when estimated using the roster data. The dyadic reciprocity in free-list friendship nominations in Colombia was $\rho = 0.84$ $(95\%CI : 0.74, 0.96)$.

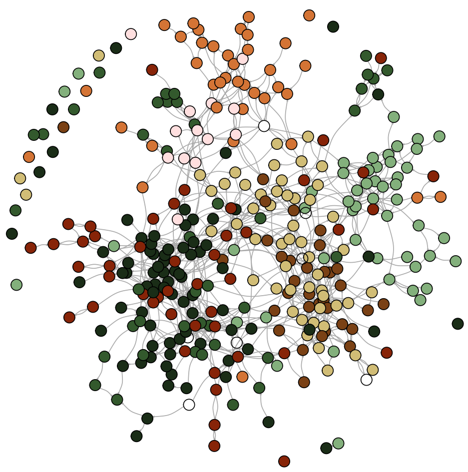
To summarize the comparison between the KoboCollect methodology introduced here, the photo-roster method introduced by Ross and Redhead (2023), and the pencil-and-paper approach used in Colombia, we can start by comparing the methods in terms of data collection time, data entry time, and data cleaning time. Data collection was probably easiest in Colombia, where a simple-pencil-and paper protocol was employed. However, the roster-based search feature in KoboCollect made the data collection nearly as simple, although the photo confirmation step (if used) increases the interview by a few seconds per tie, per layer. For long surveys, it may end up taking up to 30-50% of the interview time. The roster-based method used in Benin was also simple to run in the field, requiring about the same time per interview as the KoboCollect approach. However, transcription and database entry of paper-and-pencil data is by far the least efficient method, taking weeks or months to process datasets of a similar size of those used here. The photo-roster method allows automated AI processing of cell-phone images of tokens place on a physical photo-roster, and data entry takes only a few days. The KoboCollect approach, however, requires only a few clicks, making it the methods with the most efficient "backend".

In terms of data quality, our descriptive statistics and analyses with the social relations model suggest that the KoboCollect method and roster method yield data of comparable quality. Interestingly, average degree was not larger for the roster-method in both sites, as might be expected from past work (Sudman 1985). Instead, in Atlantique the roster-method yielded more reported ties, while in Mono the free-list method did so. This likely highlights the key role played by variation in researcher behavior. Using either method, long lists of names can be elicited, and much will depend on the details: are research assistants instructed to aim for a certain number of ties per question, or cap responses after a certain limit? Or, are they instructed to ask: "Is there anybody else you would like to mention?" until the respond completely exhausts all possible ties?

Limitations

The data collection procedure introduced here is not without limitations and constraints, however. Since it depends on a roster of photos and IDs, the data collection process is forced to be sequential, with social network questionnaires being administered only after the completion of a community census, which should include taking pictures of interviewees. This will not typically be a problem, because the collection of names and facial photographs can be performed when

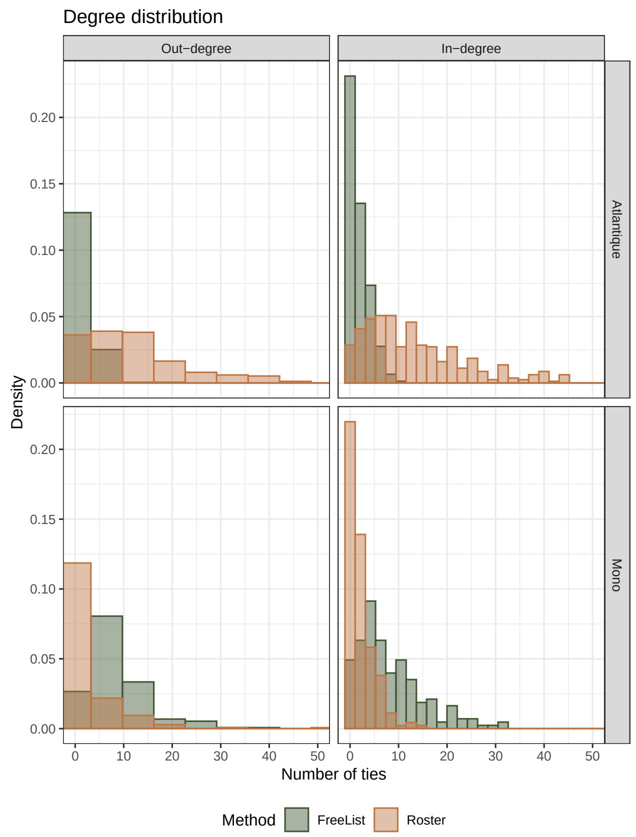
Figure 5. Friendship network ($n = 384$) in Atlantique, from a community living in a small coastal archipelago. Nodes are colored by island. The clear clustering of friendship ties by island indicates that close friendships are more likely to occur between individuals who reside on the same island.



explaining the study and securing informed consent for the entire research protocol. Likewise, it is fairly common for anthropologists working in small-scale communities to maintain name- and photo-rosters, specifically for uses such as social network data collection. However, the procedure of taking pictures is—privacy-wise—inherently invasive to a certain degree, and local cultural sensitivities towards it may vary substantially. We therefore recognize that this workflow may not be feasible in all field situations. Because of this, our R package provides an option to omit all photo confirmation steps from KoboCollect, which allows network data to be collected using only a roster of names.

There are also some small technical limitations with the name search methods—i.e., minimal auto-complete—available in KoboCollect. As field researchers know, names are messy. Individuals may report different first names, or different family names, or nicknames, or they may invert the order in which names are reported. If the use of nicknames in a society is common, then those name must be included in the name string provided to the app, perhaps inside of parentheses after the full legal name of each respondent. Additionally, the possible lack of fixed spelling conventions in local languages, combined with typing errors from the researchers, may further complicate the process of finding specific individuals in large rosters. Name matching algorithms based on various types of string distances, or even phonetic matching, may provide extremely valuable in the future. At present, however, no fuzzy string matching functionality is available in the KoboToolbox/XLSForm ecosystem.

Figure 6. Degree distribution of the Beninese friendship networks as elicited with both roster and free-list methodologies.



Finally, it is advisable to have the definitive survey form downloaded in the devices before heading out to the field. The process of loading surveys (and associated data files) into KoboCollect on a given device requires an internet connection. Bad internet connections can cause frequent form download failures, especially with a large number of photographs (as the first author knows too well).

Conclusion

The KoboCollect-based method for collecting social network data that we introduce here complements roster-based tools that we have developed and deployed in the past (Ross and Redhead 2023). The tutorial we have provided should be useful for anthropologists and other social scientists interested in the collection of sociocentric network data. By delegating the task of photo-based record linkage to the interviewees themselves, our approach to network data collection comes with the advantages of: 1) minimizing errors in identity resolution, and 2) maintaining a rapid interview, which makes efficient use of respondents' time. Furthermore, the specific questionnaire design we implement results in the immediate storage of social network information in a data structure (edge-list), that is already suited for analysis with standard tools like the `STRAND` R package (Redhead et al. 2023b; Ross et al. 2024). As calls for social scientists to include more diverse populations in their research grow (Henrich et al. 2010), we hope that the simple field methods that we have presented here help field researchers overcome some of the difficulties associated with collecting social network data in rural settings.

Funding Acknowledgements

This research was funded by the Max Planck Institute for Evolutionary Anthropology. Daniel Redhead was supported by the 'Societal Transitions and Behavioural Change' sector plan from the ministry of Science and Culture of The Netherlands.

Declaration of conflicting interests

The authors declare no competing interests.

Ethics statement

The research project was approved by the Department of Human Behavior, Ecology, and Culture at the Max Planck Institute for Evolutionary Anthropology. Local research approvals in Benin were granted by INStaD (Institut National de la Statistique et de la Démographie). Consent to conduct research at the community level was granted by the village chief, a legal representative of the community. Participation in the study was voluntary at the individual level, and each participant provided informed consent prior to participating in the research project.

References

Back, M. D. and Kenny, D. A. (2010). The social relations model: How to understand dyadic processes. *Social and Personality Psychology Compass*, 4(10):855–870.

Brewer, D. D. (2000). Forgetting in the recall-based elicitation of personal and social networks. *Social networks*, 22(1):29–43.
Carley, K. M. and Krackhardt, D. (1996). Cognitive inconsistencies and non-symmetric friendship. *Social networks*, 18(1):1–27.
Colleran, H. (2020). Market integration reduces kin density in women's ego-networks in rural Poland. *Nature communications*, 11(1):266.
Cornwell, B. and Hoagland, E. (2014). Survey methods for social network research. *Health survey methods*, pages 275–313.
Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *Complex Systems*, page 1695. https://igraph.org.
Gettler, L. T., Redhead, D., Dzabatou, A., and Lew-Levy, S. (2023). BaYaka forager food sharing networks in the Congo Basin: The roles of gender homophily and kin sharing. *American Journal of Biological Anthropology*, 181(1):59–69.
Henrich, J., Heine, S. J., and Norenzayan, A. (2010). Most people are not WEIRD. *Nature*, 466(7302):29–29.
Kenny, D. A. and La Voie, L. (1984). The social relations model. In *Advances in experimental social psychology*, volume 18, pages 141–182. Elsevier.
Kobo Toolbox (2024). KoboToolbox | Data Collection Tools for Challenging Environments. https://kobotoolbox.org/.
Koster, J. M. and Leckie, G. (2014). Food sharing networks in lowland Nicaragua: an application of the social relations model to count data. *Social Networks*, 38:100–110.
Marsden, P. V. (1990). Network data and measurement. *Annual review of sociology*, 16(1):435–463.
Open Data Kit Community (2024). XLSForm Standard. https://xlsform.org/.
Pisor, A. C., Gervais, M. M., Purzycki, B. G., and Ross, C. T. (2020). Preferences and constraints: the value of economic games for studying human behaviour. *Royal Society open science*, 7(6):192090.
Power, E. A. (2017). Social support networks and religiosity in rural South India. *Nature Human Behaviour*, 1(3):0057.
Ready, E. and Power, E. A. (2021). Measuring reciprocity: Double sampling, concordance, and network construction. *Network Science*, 9(4):387–402.
Redhead, D., Dalla Ragione, A., and Ross, C. T. (2023a). Friendship and partner choice in rural Colombia. *Evolution and Human Behavior*, 44(5):430–441.
Redhead, D., McElreath, R., and Ross, C. T. (2023b). Reliable network inference from unreliable data: A tutorial on latent network modeling using STRAND. *Psychological methods*.
Ross, C. T., McElreath, R., and Redhead, D. (2024). Modelling animal network data in R using STRAND. *Journal of Animal Ecology*, 93(3):254–266.
Ross, C. T. and Redhead, D. (2021). DieTryin: An R package for data collection, automated data entry, and post-processing of network-structured economic games, social networks, and other roster-based dyadic data. *Behavior Research Methods*, pages 1–21.
Ross, C. T. and Redhead, D. (2023). Automatic entry and coding of social networks and dyadic peer ratings. *Methodological Innovations*, 16(2):138–148.
Starkweather, K. E., Reynolds, A., Zohora, F., and Alam, N. (2023). Shodagor women cooperate across domains of work and childcare to solve an adaptive problem. *Philosophical Transactions of the Royal Society B*, 378(1868):20210433.
Sudman, S. (1985). Experiments in the measurement of the size of social networks. *Social networks*, 7(2):127–151.

Appendix

Hand-editing the XLSForm

Many readers will not have much experience coding apps using the XLSForm standard, and so we generally recommend using the XLSFormulatoR R package to build questionnaires automatically. However, in an appreciable fraction of cases, we assume that readers may want to modify the network_collect.xlsx form—perhaps by adding additional demographic questions—before uploading it to KoboToolbox to create the app. This is possible. As such, we provide a brief overview on the XLSForm syntax contained in the output of XLSFormulatoR for the benefit of readers who may need to edit the form by hand.

KoboToolbox questionnaires are built via XLSForm, a standard which allows the construction of interactive apps using an simple Excel template. The XLSForm template can then be compiled for web and Android platforms that then handle data collection. Readers who are interested in a deeper understanding of the XLSForm standard, as well as the app infrastructure, are referred to: https://xlsform.org/ and https://community.KoboToolbox.org/.

An XLSForm contains two required sheets called survey and choices, and may contain a third sheet called settings. The survey sheet structures the form, and contains all questions, while the choices sheet provides lists of option values that can be retrieved or referenced by questions or statements in the survey sheet. Three columns in survey are mandatory: type, name, label. The type column specifies the question type to be implemented, and name assigns the variable name. The values in label are what will appear on-screen, either online or in the app. The column relevant specifies the skip logic, allowing the definition of conditional dependencies between questions: for example, we may want to ask a follow-up question only if a previous question received a specific response. If the cell is left blank, there is no relevance condition, and the question will always appear in the final survey form.

An XLSForm walkthrough example

The structure of the external names.csv (or any attached external csv file) is the same as the choices sheet in the XLSForm: list_name defines the options group, which we do not need in this specific example, as we do not have any choice filter (we filled this column with the string "individual"); label contains the full legal names (and even nicknames) of individuals in the sample, while name lists their associated 3-letter personal identifier code. We understand that this choice in column naming might be confusing, but name is the XLSForm column that stores the unique-identifier values in the database (we thus need this to refer to unique ID codes, rather than legal names, which may not be unique), while label contains the value appearing on-screen (which should be the names of the community members). The survey sheet in the network_collect.xlsx file appears as shown in Figure 7. Compilation in XLSForm is sequential: the first 4 question types specified in Figure 7 collect metadata on the starting and ending time of the interview, the date of the interview, and the KoboToolbox username of the interviewer.

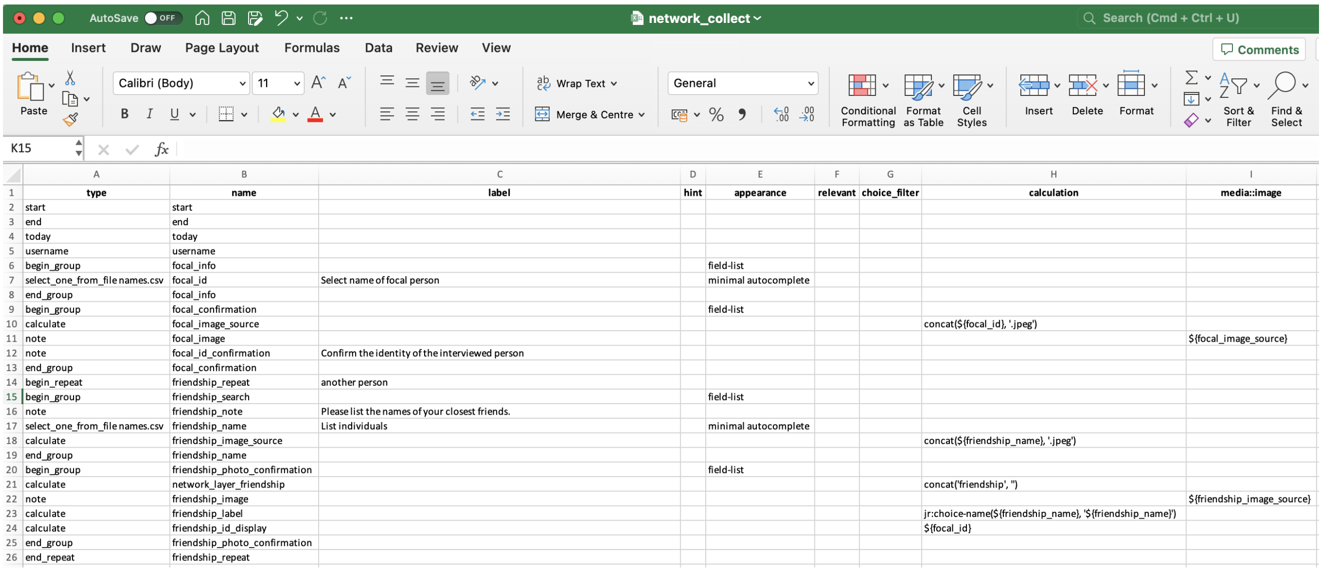
Questions in survey can be grouped together by enclosing them between the begin_group and end_group statements in the type column. In addition, the field-list appearance specification at the start of the group will make all the grouped questions appear in the same screen in the app. Arbitrary questions can be added to our template, prior to collecting network data, using the XLSForm standard. The main contribution we make here, is providing three network-oriented question modules. The first one, a group called focal_info, is the basic building block of all XLSForms produced by XLSFormulatoR. The focal_info module contains only one question, focal_id, whose purpose is to connect the identity of the interviewee to the IDs already present in the names.csv database file. The command select_one_from_file names.csv instructs KoboCollect to retrieve all the entries in the names.csv attachment, and display them as options for the interviewee to select. The values that will be displayed on-screen are retrieved from the label column of the external file (the individual full names); however, the value that will be *stored* in the database after interview submission will be the name column entry—i.e., the personal ID code. The minimal autocomplete appearance provides a pull-down menu with a search bar, so that the interviewer may start typing a name, and the menu will be shortened to include only the subset of names containing the entered text.

The focal_confirmation module serves the purpose of retrieving the photo associated with the ID selected at the focal_id question. This is done by setting the type field to calculate, and specifying the calculation to be performed in the calculation field. In this case, the calculation to perform is concat(${focal_id}, '.jpeg'). This function concatenates together the value selected in the focal_id question (the focal ID) and the ".jpeg" suffix, which is the photographs' file extension. The output of the calculation will be a filename—e.g., "R6T.jpeg"—that gets stored under the variable name focal_image_source. In the next row, the app will use the focal_image_source value to retrieve the corresponding image from the attached media files, and will display it on screen. This is achieved by setting type to note, and retrieving the image by setting ${focal_image_source} in the media::image field. The statement focal_id_confirmation is simply a written text note that reminds the interviewer to confirm that the identity of the interviewee matches the retrieved photo, so that we can safely connect the interviewee to the ID in the database.

Repeat-groups

The XLSForm standard allows the creation of questions or groups of questions that can be dynamically repeated. This feature can be implemented by setting the number of repeats via the value of a previous answer: for example, we might ask an individual how many children they have, and then repeat in-detail questions on each child according to the numeric value given in the previous question. Alternatively, we can design an open-ended repeat-group where the interviewee

Figure 7. The XLSForm used in this article, generated with `XLSFormulatoR`.



	type	name	label	hint	appearance	relevant	choice_filter	calculation	media::image
2	start	start							
3	end	end							
4	today	today							
5	username	username							
6	begin_group	focal_info			field-list				
7	select_one_from_file names.csv	focal_id	Select name of focal person		minimal autocomplete				
8	end_group	focal_info							
9	begin_group	focal_confirmation			field-list				
10	calculate	focal_image_source						concat(${focal_id}, '.jpeg')	
11	note	focal_image							${focal_image_source}
12	note	focal_id_confirmation	Confirm the identity of the interviewed person						
13	end_group	focal_confirmation							
14	begin_repeat	friendship_repeat	another person						
15	begin_group	friendship_search			field-list				
16	note	friendship_note	Please list the names of your closest friends.						
17	select_one_from_file names.csv	friendship_name	List individuals		minimal autocomplete				
18	calculate	friendship_image_source						concat(${friendship_name}, '.jpeg')	
19	end_group	friendship_name							
20	begin_group	friendship_photo_confirmation			field-list				
21	calculate	network_layer_friendship						concat('friendship', '')	
22	note	friendship_image							${friendship_image_source}
23	calculate	friendship_label						jr:choice-name(${friendship_name}, '${friendship_name}')	
24	calculate	friendship_id_display						${focal_id}	
25	end_group	friendship_photo_confirmation							
26	end_repeat	friendship_repeat							

has the possibility of adding another item after the previous one has been answered. This allows the repeat-group to be flexibly extended as needed, and be unconstrained by previous responses. In this example, we use the latter method, allowing interviewees to iteratively report as many network ties as they want.

In Fig. 7, the `friendship_repeat` group defines the repeat instance, and sets the boundaries of the repeat group with the `begin_repeat` and the `end_repeat` specifications in the `type` field. The repeat is further partitioned into two groups called `friendship_search` and `friendship_photo_confirmation`, both with a `field-list` appearance. `friendship_search` has three entries: `friendship_note` is a text note prompting the interviewee to nominate alters as friends. `friendship_name` has the same structure as `focal_id` above, retrieving all the IDs and names from the external CSV file and allowing the selection of one of the `label` values in the list that will be displayed on-screen. `friendship_image_source` concatenates the selected `name` value (i.e., the individual's ID code) and the '.jpeg' suffix, creating the filename of the image to retrieve. The `photo_confirmation` group contains three calculations and one note. The variable starting with the `network_layer` prefix simply stores the network layer that the current tie refers to (in this case, friendship). Since we are dealing with friendship networks, we repeat the label *friendship* by plugging `concat('friendship', '')` in the `calculation` field. `friend_image` is a `note` item that displays the image associated with the filename created in `friendship_group`, retrieving it via the `${friend_image_source}` call in the `media::image` field.

Two more calculations are performed in the `friendship_photo_confirmation` group, whose output is not shown on-screen, but is stored in the data table that will be produced by KoboToolbox. The first is `friend_label`, which returns the full name of the alter by using `jr:choice-name(${friend_name}, '${friend_name}')` and stores it. This allows the end-user to potentially display the list of alter names reported as network ties by the focal individual. The second is `focal_id_display`, which returns the interviewee's ID via `${focal_id}` and stores it in the database. Doing so, the resulting data table will have a column of alter IDs and a column of focal IDs, which makes the transformation into an edge-list smooth once the data are downloaded from the platform. The last two rows in the XLSForm are ending statements to close the `friendship_photo_confirmation` group and the `friendship_repeat` repeat group.