Regular Article

Coding Qualitative Data at Scale: Guidance for Large Coder Teams Based on 18 Studies

Melissa Beresford[1] ⓘ, Amber Wutich[2] ⓘ, Margaret V. du Bray[3], Alissa Ruth[2], Rhian Stotts[2], Cindi SturtzSreetharan[2], and Alexandra Brewis[2]

Abstract
We outline a process for using large coder teams (10 + coders) to code large-scale qualitative data sets. The process reflects experience recruiting and managing large teams of novice and trainee coders for 18 projects in the last decade, each engaging a coding team of 12 (minimum) to 54 (maximum) coders. We identify four unique challenges to large coder teams that are not presently discussed in the methodological literature: (1) recruiting and training coders, (2) providing coder compensation and incentives, (3) maintaining data quality and ensuring coding reliability at scale, and (4) building team cohesion and morale. For each challenge, we provide associated guidance. We conclude with a discussion of advantages and disadvantages of large coder teams for qualitative research and provide notes of caution for anyone considering hiring and/or managing large coder teams for research (whether in academia, government and non-profit sectors, or industry).

Keywords
qualitative coding, team-based coding, qualitative data analysis, text analysis, intercoder reliability, intercoder agreement, collaborative research

Introduction

Coding text is one of the most common methodological approaches for qualitative data analysis across a wide range of academic disciplines. As texts available for qualitative research expand in form and volume (e.g., social media posts and digitized text repositories), there is an increasing need for techniques that enable coding qualitative data *at scale*. Computational and machine learning techniques that use computers to code textual data via key words or algorithmic learning are advancing quickly and will help with basic issues of volume (Nelson et al., 2021). But many qualitative researchers find that human coders alone can read and detect the subtle, contextualized, and latent themes within texts that are often the primary interest of qualitative data analysis (Bernard et al., 2016; Braun and Clarke 2014; Nelson et al., 2021). So, alongside advances in computational/machine learning, there is a significant need to advance methods that will assist with both scaling and (relatedly) accelerating techniques for humans to read, interpret, and apply codes to textual data (Benoit et al., 2016; Cascio et al., 2019; Liggett et al., 1994).

Team-based coding is one approach that enables researchers to code qualitative data at higher volumes and with increased speed

(Burla et al., 2008; Casio et a. 2019; Campbell et al., 2013; Giesen and Roeser 2020; Hruschka 2004; MacQueen et al., 1998). Simply put, more human coders facilitate data analysis in a shorter period of time by sharing the labor cooperatively.

Yet, team-based coding also presents many challenges, and these challenges are only amplified as more team members are brought on to code larger volumes of data. In this paper, we outline unique challenges to working with large coder teams based on 18 projects in the last decade. Each of these projects engaged a team of 12 (minimum) to 54 (maximum) individuals to code a large-scale qualitative data set. For each challenge identified, we detail

[1]Department of Anthropology, San José State University, San Jose, CA, USA
[2]School of Human Evolution and Social Change, Arizona State University, Tempe, AZ, USA
[3]Hollins University, Roanoke, VA, USA

Corresponding Author:
Melissa Beresford, Department of Anthropology, San José State University, Clark Hall Suite 469, One Washington Square, San Jose, CA 95112-3613, USA.
Email: melissa.beresford@sjsu.edu

examples of problems we encountered and solutions we devised so that other researchers can more easily mobilize large coder teams for the analysis of large-scale qualitative data sets in academia, government and non-profit sectors, or industry. The same techniques can be applied to smaller-scale datasets in cases where rapid processing is needed, such as piloting for time-sensitive intervention and/or development projects.

Team-based approaches to qualitative data analysis

Over the past 25 years the methodological literature on team-based coding has grown substantially as more scholars recognize the benefits of using multiple coders to analyze qualitative data. Team-based coding typically involves, first, developing a codebook and assessing intercoder consensus or reliability in some way, and then, splitting up the data among multiple coders so that each coder applies the codes to a portion of the dataset (Burla et al., 2008; Campbell et al., 2013; Carey et al., 1996; Giesen and Roeser 2020; Hruschka et al., 2004; Kurasaki, 2000; MacQueen 1998). Most coder teams discussed in the methods literature consist of two to four coders (Campbell et al., 2013; Carey et al., 1996; Giesen and Roeser 2020; Hruschka et al., 2004; Kurasaki, 2000). The number of coders needed for a project, however, depends on multiple variables, including the size and/or complexity of the dataset; the training, ability, and experience of the coders; language and cultural expertise required; the dispersion of the analytically significant themes in the dataset; the number of times the themes of interest appear in the dataset; the difficultly of detecting the theme in the text; and the levels of specificity the researcher wishes to achieve (Ryan 1999; Bernard et al., 2016).While most scholars cite increased speed and efficiency as their primary motivation for team-based coding (Burla et al., 2008; Cascio et al., 2019; Giesen and Roeser 2020; Hruschka et al., 2004; Lichtenstein & Rucks-Ahidiana, 2021; MacQueen et al., 1998), using multiple coders also provides several other key advantages.

First, team-based coding encourages analytical precision by forcing researchers to clarify exactly what a theme means so that everyone on the team can code consistently (Hruschka et al., 2004; MacQueen et al., 1998). For example, differing interpretations or understandings of themes that arise during codebook development can help the research team to refine thematic codes and establish inclusion and exclusion criteria for codes through iterative discussion and resolution (Cascio et al., 2019; Hruschka 2004; MacQueen et al., 1998).

Second, team-based coding helps to ensure coding reliability (Benoit et al., 2016; Burla et al., 2008; Carey et al., 1996; Hruschka et al., 2004; Krippendorff, 2018; MacQueen et al., 1998). Ensuring agreement among coders allows researchers to demonstrate that different people are able to apply the codebook in the same way and, by extension, that individual coders are more likely to use the codebook in a consistent way over time (Cascio et al., 2019; Hruschka et al., 2004; MacQueen et al., 1998).

Third, multiple coders can help establish the validity (Hruschka 2004; Kurasaki, 2000; Moret et al., 2007) or credibility (Tracy,

2010) of the analysis process. Agreement among multiple coders indicates the themes identified are recognizable by multiple people and not simply figments of one researcher's imagination (Bernard et al., 2016). Moreover, the emic validity (Whitehead, 2005) of the coding system can be enhanced if the coding team includes participants who possess cultural, linguistic, or local expertise relevant to the phenomenon being studied (Bernard et al., 2016).

Finally, using multiple coders can help researchers to identify typicality among coded data segments (Ryan 1999). For example, passages that all coders code consistently for a particular theme capture that theme's core, while passages with less coder agreement generally represent atypical exemplars, or the edges of, a theme (*ibid*). Additionally, high coder agreement can also help to identify exemplary quotes by systematically identifying passages of text that best represent a theme (*ibid*).

Despite these significant advantages, team-based coding also presents many challenges. Team-based coding is prone to communication difficulties, especially among coders with many differences in perspective, opinion, personality, or workstyle (Bozeman et al., 1999; Hall et al., 2005). Good communication in a coder-team requires effective management to ensure team members work efficiently, cooperatively, and on time, with minimal duplication and error (Bozeman et al., 1999; Hall et al., 2005; Richards, 1999).

Training coders—especially novice coders—is time consuming (Cascio et al., 2019; Hall et al., 2005; Hruschka et al., 2004; MacQueen et al., 1998). The amount of time required to train coders *and* complete all the project coding may outstrip the amount of time that coders have available to dedicate to the project (Campbell et al., 2013; Hall et al., 2005). One way to address this problem is to use rotating coder teams (Cascio et al., 2019), but this approach requires multiple rounds of new training sessions and may present difficulty in ensuring reliability across different coder teams.

Difficulties often arise in determining compensation, attributing contributions, and managing the various competing interests and goals of team members (Bozeman et al., 1999; Hall et al., 2005; Liggett et al., 1994; Richards, 1999). Such problems can be resolved through close communication and a clear articulation and understanding of the project goals among team members (Bozeman et al., 1999; Hall et al., 2005; Giesen & Roeser, 2020). But, all these challenges and complexities to team-based qualitative analysis become amplified as more team-members join the process (Giessen & Roser 2020). Solutions that work for managing a team of two to four coders may not work well with 10 + coders.

A research informed approach to defining challenges and guidance for large coder teams

Below we outline four major challenges faced by large coder teams (10 + coders) and offer guidance based on our experiences. Our lab (Culture, Health, and Environment Laboratory [CHELab]) is a large training lab at Arizona State University. Founded in 2006, it is collaboratively led by four social science faculty members and a part-time lab manager.

The projects we run through the lab are tied to each faculty member's larger research agenda. In any given semester, we typically work on two to four substantial text-based analyses. At the time of writing, we have completed 18 qualitative and mixed-methods projects that trained and deployed large coder teams (12–54 coders) (see Table 1).

As reflected in Table 1, our lab projects primarily focus on cross-cultural research. The special conditions that drive our theoretical frameworks and code and codebook development are outlined in Wutich and Brewis (2019) and Wutich et al. (2021). However, in some cases we conduct more conventional single-site projects. In those cases, the theoretical frameworks and codes are determined by the lead PI of the project (e.g., Brewis et al. 2019; Roque et al., 2021; Ruth et al., 2021; Trainer et al., 2021). IRB oversight for all these projects was provided by Arizona State University.

Based on our experiences leading and training coders for these 18 studies, we identify four key recurring challenges for large coder teams: (1) recruiting and training coders, (2) providing coder compensation and incentives, (3) maintaining data quality and ensuring reliability at scale, and (4) building team cohesion and morale. We consider these four challenges to be the most salient challenges for large coder-teams that are not presently discussed in the methodological literature. Our identification of these challenges occurred through inductive, iterative reflection and analysis of training manuals and documents we have developed over the past 15 years and lab notes we have taken on the processes and procedures of past projects. We conclude with our collective observations on the unique advantages of employing large coder teams despite these challenges, and we highlight three notes of caution based on the problems we have yet to solve.

Challenge 1: Recruiting and training coders.

Smaller-scale coding teams typically consist of project leads and trained research assistants who are intimately familiar with the project research questions and data set (Burla et al., 2008; Cascio et al., 2019; Giesen and Reoser 2020; Hruschka 2004). This expertise is ideal, but it is rarely available at scale. Large coding teams typically require recruiting novice coders who need training. However, recruiting large pools of potential coders who are able to join a project team and willing to learn how to code qualitative data is a challenge (see Table 2 for summary).

Guidance for recruiting and training coders

Recognize potential pools of coders and target recruitment. Advertising for paid research assistants may be the most obvious choice for compiling a research team, but we have found it useful to think broadly about other potential pools of coders that may be available to join a project. Potential pools of coders may include undergraduate and/or graduate students who are eager to gain hands-on research experience; engaged community members with a stake or vested interest in the research outcomes; or interdisciplinary research collaborators who are untrained but interested in qualitative research.

As university-based researchers, we most frequently recruit undergraduate students for our coder teams. We do this in two ways: (1) through lab-based internships and (2) through practicum course experiences. When recruiting for lab-based internships, we put out a general call to student email list-serves advertising our lab internship and describing our research studies. Generally, over the course of an academic year, our lab houses two to four projects and we typically assign lab interns to one of these ongoing projects (i.e., students work on the same project over the course of the semester or academic year).

Practicum course experiences involve structuring a university course around a specific research project and turning the whole class into a coder team. For example, for a study on children's perceptions of water futures in the United States (Vins et al., 2014), we crafted the data analysis schedule around the learning goals of an upper-division course. The 54 students enrolled in the course became the coder team, refining the codebook and coding a data set of 3,120 pieces of children's art over the course of the semester. By structuring a university course around coding and analyzing data, we were able to process far more data than would be possible on a small research team. Importantly, this enabled students to obtain an unparalleled hands-on, collaborative research experience in order to learn the social science research process. In fact, the lead author of our academic publication was an undergraduate student enrolled in the practicum.

Clearly articulate coder benefits and incentives. With planning, we find it is possible to align the coders' needs with our project's research goals, learning outcomes, and compensation. For example, if recruiting students, PIs should highlight the types of research skills and experiences coders will gain. If recruiting engaged community members, it may be more important to highlight the broader impacts of the research and competitive pay rates.

Take-Home Tip 1: Highlight individualized benefits to facilitate coder recruitment

When recruiting student coders to join either our lab as interns on multiple ongoing projects, or our practicum courses for a specific project, we outline pertinent details of the project(s), including the project goals, community partners, and general research strategy. We highlight the concrete skills students gain upon completion of the course/project, the credits students would earn toward their degree, the curricular requirements that the course fills in the students' degree program, and the amount of time outside of class students need to dedicate to the project (e.g., class homework). This information allows students to make an informed decision as to whether or not they wish to join the course/project.

Table 1. 18 Studies Using Large Coder Teams for the Analysis of Qualitative Data.

	Data Collection Year	Project	Data source(s)	Data Collection Sites	# Total Trainee Coders	# Codes in Codebook	Key Citations
1	2006	Water quality perceptions	Free list interviews with 131 residents	4 neighborhoods in Phoenix, Arizona, USA	~ 20	32	Gartin et al., 2010
2	2007	Climate change perceptions and local ecological knowledge	Free list interviews with 279 adults in 6 countries	Local sites in Fiji, Ecuador, New Zealand, Australia, USA, UK	~ 20	36	Crona et al., 2013
3	2009	Perceptions of water-borne disease transmission	Open and closed-ended interviews with 468 adults in 9 countries	Local sites in Tanzania, Bangladesh, Guatemala, China, Fiji, Paraguay, New Zealand, UK, USA	~ 20	4	Brewis et al. 2013
4	2011	Neighborhood stigma	Open and closed ended interviews and free list with 300 respondents	Two neighborhoods in Phoenix, Arizona, USA	15	4	Wutich et al., 2014
5	2010	Science of water art	Drawings by 1,650 schoolchildren (2 per participant)	55 Arizona grade schools, USA	54	10	Vins et al., 2014
6	2011	Justice in water institutions	Semi-structured interviews with 132 people in 4 countries	Local sites in Fiji, Bolivia, New Zealand, USA	~ 20	5	Wutich et al., 2012; 2013; Larson et al., 2016
7	2012	"Fat" by any other name	Open-ended questions and elicited free-list word terms with 264 students, plus 15 in-depth follow up interviews	University campus + clinics, gyms, and other weight-related sites in metropolitan Phoenix, Arizona, USA	12	15	Trainer, Brewis, Williams, & Chavez, 2015a, 2015b
8	2012	Uncertainty and climate science	401 surveys collected from respondents in 6 countries	Local sites in Mexico, Fiji, China, New Zealand, USA, Australia	~ 20	13	Gartin et al., 2020
9	2013	Social acceptability and wastewater recycling	387 semi-structured interviews, including an illustrative portion collected in 4 countries	Local sites in Guatemala, Spain, Fiji, USA	15	35	Rice et al. 2019; Stotts et al., 2019
10	2014	Uncertainty and climate distress	Structured open- and close-ended questions with 469 people in 5 countries	Local sites in Cyprus, Fiji, New Zealand, UK, Australia and 3-site design in USA (AK, AL, AZ)	13	102	du Bray et al., 2017a, 2017b
11	2015	Hygiene norms and stigma	267 structured interviews with community members in 4 countries	Local sites in Guatemala, Fiji, New Zealand, USA	13	54	Brewis et al. 2019
12	2016	Ecosystem services and urban rivers	Structured open-ended questions in interviews with 312 residents of riverine communities in 5 countries	Fiji, UK, USA, New Zealand, Australia	12	53	du Bray et al., 2017b, 2019
13	2017	Household water insecurity	Closed and open-ended responses from 3,033 household surveys about water insecurity in 16 countries	Local sites in Haiti, Guatemala, Mexico, Bolivia, Colombia, Tajikistan, Lebanon, Pakistan, Nepal, Ethiopia, Nigeria, Uganda, Kenya, Tanzania, Malawi, Ghana	16	26	Schuster et al. (2020)

Table 1. (continued)

Data Collection Year	Project	Data source(s)	Data Collection Sites	# Total Trainee Coders	# Codes in Code-book	Key Citations
14 2017	Fat talk in everyday settings	494 recorded instances of fat talk conversations	Local sites greater Phoenix, AZ, USA	22	1	SturtzSreetharan et al., 2019; Agostini et al., 2019
15 2018	Citizen science and structural awareness	Observations of 130 possible sites of social structural exclusion	9 sites in Tempe, Arizona, USA	31	4	SturtzSreetharan et al., 2021
16 2018	Structural competency in pre-health students	Open-ended responses to vignette elicitations from 27 students in pre- and post-tests	Undergraduate students at a US university	31	2	Ruth et al. 2021
17 2018	Water sharing in the wake of disasters	Open-ended questions in interviews with 81 people (27 per site)	3 local sites in Puerto Rico	22	2	Roque et al., 2021
18 2019	Drinking water and social disparities	Open-ended responses to vignettes by 154 respondents	4 neighborhoods in phoenix, Arizona	42	18	Brewis et al. 2021

Table 2. Strategies and Examples for Recruiting and Training Coders.

Challenge	Some Strategies	Examples
Recruiting, hiring and training coders	a) Look for early-career researchers who can be trained, supervised, and mentored over a longer period of time b) Prioritize coders who will especially benefit from the long-term experience of working in a coding team (e.g., first-generation students or new graduates looking for career experience) c) Recruit coders from data collection teams whenever possible (and rely on recommendations from supervisors from data collection teams); this is further helpful for understanding the context in which the data was collected d) Prioritize hiring new coders who have been recruited or recommended by existing coders	We target coder recruitment to freshman and sophomore students who then work with us in our lab for three to four years. We preference students who have experience in collecting data through other fieldwork or research experiences. Lab interns start out on simple data management tasks, and move on to more complicated data entry and coding tasks. After multiple years coding on multiple team projects, these coders become experts and begin working on complex datasets, including data collected in other languages (e.g., Spanish, Portuguese), and with more complex assignments (e.g., supervisory tasks).

Table 3. Strategies and Examples for Providing Coder Compensation and Incentives.

Challenge	Some Strategies	Examples
Providing coder compensation and incentives	a) Provide job application training and feedback on resumes and cover letters b) Provide letters of recommendation for coders, particularly those who work long-term c) Provide beneficial connections to senior PIs and supervisors (e.g., helping identify and apply to relevant professional opportunities by virtue of networks) d) Help student coders navigate the "hidden curriculum" (e.g., provide advice on what courses might matter to employers and graduate schools) or university employees	We have collectively written over 1000 letters of recommendation for student coders over 15 years. It is common to write between 10-35 letters of recommendation per coder for jobs, fellowships, scholarships, and advanced degree programs. Keeping files of: (1) a standard template explaining skills each coder gained, (2) a self-evaluation form each coder completes, and (3) an annual evaluation completed by the coder's supervisor helps in providing excellent letters of recommendation at scale. Coders from our lab have been gone on to lead major studies at NGOs, in tenure-track positions, and so forth.

Target coder training according to both project needs and coder incentives. We recognize three major strategies to training a coder team, based on the types of coders hired: (a) expert: hire technical experts with experience in qualitative coding and give project-specific training; (b) targeted training: hire coders who are technical novices and give targeted methodological training; and (c) full training: hire coders who are technical novices and make a major methodological investment to train them as full collaborators.

The expert strategy (a) typically involves hiring paid assistants who have technical expertise. This strategy is financially costly and not always feasible at scale. The full training strategy (c) requires significant time and resources, and normally represents the process of training a graduate student over a number of years or training a community partner who collaborates on a long-term project or a series of projects. Due to the high costs of both strategies, they may not be feasible for large-scale coder teams.

The targeted training strategy (b) is most common for the purposes of compiling a large coder team for a specific project. This means that training should be targeted to the specific project, *but* training should also ideally advance the coders' needs and goals. For example, in working with student coders,

we always provide training on how to articulate their newly acquired research skills in a cover letter or job interview, and how discuss the applicability of those research skills to different professional settings. For community partners, we ensure that they understand the full method well enough to use the findings as a platform for political self-advocacy.

Challenge 2: Providing coder compensation and incentives

Coding qualitative data is labor-intensive (Bernard et al., 2016; Giesen and Roeser 2020; Hruschka et al., 2004); we try to compensate coders in a way they consider fair and equitable. PIs who have the financial resources to pay a large team of coders (e.g., within industry and private sector contexts) can and should compensate coders with money. However, within financially-constrained research contexts and community-based participatory projects, monetary compensation for a large number of coders is a challenge. To address this, we find that compensation can take a variety of forms depending on the needs and expectations of the potential coders (see Table 3 for summary).

Table 4. In Coders' Own Words: The most valuable lesson or skill coders say they gained working on a coding team

"The most valuable skills that I have gained from my experience are learning how to do data entry, Coding/grouping data, and how to work with others, specifically explaining why I interpret the data to be grouped under a Certain code and Coming to a Consensus on what to code it as."—White, male coder

"I now see how refined my research skills have become … Specifically, I am now very confident in my ability to accurately record and code data in a professional lab setting, using real raw data… I not only increased my computer skills but also improved my teamwork and collaborating abilities."—Asian, female coder

"Having to sit down and do the transcribing and coding was immensely valuable for me. I eventually ended up doing ethnographic work in Mongolia after graduating and having had this class as a first experience was extremely useful."—White, male coder

Guidance for providing coder compensation and incentives

Appropriate compensation. If compensating coders with pay, we study local salary ranges and compensation practices for professionals and assistants. This includes expectations that people may have for paid or unpaid time off, as well as local cultural practices like bonus pay during certain times of year (e.g., *aguinaldo*, an extra month of pay given in December across Latin America). When working in multiple countries, languages, and cultures, we find this can be a significant challenge that may require careful study and consultation. We often work in a collaborative framework, and consult our local research partners to design pay schedules. That being said, for university-based researchers who have constrained research budgets, compensating student coders with course credit and opportunities for educational and professional advancement can be developed ethically as an alternative approach.

Whether we compensate coders using paid or applied course-credit arrangements, it is necessary to let coders know what compensation they can expect. In paid positions, this is straightforward. In our lab-based practicum courses and internships, students earn course credit that counts towards graduation and degree requirements. In our research lab, student interns receive course credits based on the amount of time they dedicate to the lab each week. In course-based classes, all enrolled students receive a set amount of course credits for completing the course. The provided course credits can meet degree requirements, but students have alternative options to meet requirements. For instance, students who did not want to participate in the advertised project or internships can choose a different course to fulfill those requirements. The key is to be clear and upfront about (a) the amount of expected work involved, including expected hours, and (b) what student coders can expect in return for that work.

> **Take-Home Tip 2: Develop multiple modes for compensating coders**
> We use a variety of monetary and non-monetary means of compensation to ensure that the value of coding work is recognized and compensated. When budgets allow, we pay coders. When budgets are constrained, we provide compensation for student coders in the form of course credit for the amount of hours worked on the project. We also facilitate an atmosphere in which students are aware that professional support, including networking advice, job, and graduate school application preparation, and letters of recommendation are benefits to joining a coding team.

Offer career mentoring. Compensation alone, in money or course credit, is not sufficient to create a sense of investment and dedication to a team and project. Being part of a coder team is almost always a temporary job. We find coders are more invested when their training and experience helps them achieve their career or educational goals. We try to create opportunities for educational and professional development for all coders, so that their duties align with their long-term goals.

For all of our paid and lab-based interns and course-based research experiences, we dedicate time to (a) learn about coders' long-term career and educational goals, and (b) tailor training opportunities and professional development time to help advance those goals. For example, we teach coders how to articulate their experiences and knowledge in job interviews and cover letters, and how to list their experiences, knowledge, and skills on resumes and/or CVs (Table 4).

Create an incentive structure for promotions and increased responsibilities. Large coding teams will inevitably consist of coders with a range of competencies, interests, and goals. We train all coders so they meet a basic level of competency to accurately code the specific project data set (Campbell et al., 2013; Carey et al., 1996; Hruschka et al., 2004; Krippendorf, 2018; MacQueen et al., 1998). Many coders who join a team have busy lives and other interests, and they wish to be involved in the project only to this baseline extent. But often, a number of coders on any one team demonstrate interests and abilities that exceed this baseline standard. We recognize and reward these interests and abilities through promotions to higher-level tasks and supervisory roles, paired with appropriate mentorship for higher-level positions.

One approach we use for large coder teams is to promote such coders to a "coding supervisor" position in which they

Table 5. Strategies and Examples for Maintaining Data Quality and Ensuring Reliability at Scale.

Challenge	Some Strategies	Examples
Maintaining data quality and ensuring coder reliability at scale	a) Build barriers to original data (e.g., never giving coders a file where they could accidentally unsort columns or delete data; use a database form rather than coding software) b) Use a version control system to check on "versions" of different files (saved over or saved as a new file) c) Require coders to use project equipment, rather than their own computer, so that data can be automatically saved back to project folders, rather than being lost or compromised on personal devices d) Use timesheets that allow coders to report on their daily activities (also allows double-checking of work) e) Reassign trainee coders to duplicate/non-essential datasets if repeated errors are made and cannot be remediated	Trainee coders may join the team and, despite multiple interventions, demonstrate an unwillingness or inability to code reliably. Some trainee coders have been unable to code reliably enough to clear our rigorous internal data quality checks, including intercoder reliability assessments. Multiple interventions include: encouraging coders to prioritize precision over speed, ensuring they read the data carefully, and going through and explaining the discrepancy between their coding and the codebook. Some coders ultimately had to be re-assigned to duplicate or non-essential datasets. This gives trainee coders the opportunity to learn about the coding process, while also protecting the integrity of the data and the project. Such measures would be reflected, however, in supervisor evaluations and recommendations. Typically, once training is complete, we reassign these trainees to another task that is better suited to their skills.

help to supervise and mentor other coders on the team, are charged with higher-level tasks such as setting up and calculating intercoder reliability tests, and help research leads to choose typical exemplars for project reporting and publication. Along with these increased responsibilities, promoted coders receive increased compensation. For coders in paid positions, this means raising their pay. For student coders earning course credit, this often means promoting them to paid positions or recommending them for paid fellowships. When possible, we nominate student coders for prestigious awards that allow them to develop their own independent projects, with the support of our lab research infrastructure. Such incentive structures not only create opportunities for coders to be further invested in the project (if they wish), but also provide an added benefit of ensuring data quality (discussed further below).

Challenge 3: Maintaining data quality and ensuring reliability at scale

Data quality is perhaps the biggest challenge in working with large coding teams. Smaller-scale coding teams have the benefit of close and constant communication in addition to generally consisting of coders who all have a high degree of training and investment in the project (Cascio et al., 2019; Hall et al., 2005; Hruschka 2004; Kurasaki, 2000; Giesen and Roeser 2020). Large coder teams that consist of primarily novice coders present numerous potential threats to data quality, including accidentally deleting, misplacing, duplicating, or rearranging data due to a lack of experience with data management. Additionally, working with multiple coders requires strategies to ensure coder reliability and/or trustworthiness (i.e., making sure coders are all applying codes consistently across the data set) (Burla et al., 1998; Cascio et al., 2019; Hruschka 2004; MacQueen et al., 1998). While a significant literature covers strategies for measuring and ensuring coder reliability among smaller-scale coder teams (Carey et al., 2008; Cascio et al., 2019; Hruschka et al., 2004; Krippendorf, 2018; White et al., 2012), these strategies can be challenging for large coder teams (see Table 5 for summary).

Guidance for maintaining data quality and ensuring reliability at scale

Build barriers to original data access. Maintaining data integrity is a key concern for all researchers. If too many people have access to the original primary data, they can be compromised. Such access can lead to data being changed, re-arranged, or deleted due to oversight and general confusion around who is accessing what data and when. To address this, we build barriers to original data access using various technological tools and clear procedures. Regardless of the specific software that we use (see Note 1) to apply and keep track of codes, only the project PIs and research leads can access original data files. One technique that has worked particularly well for us is to have coders enter their coding into web-based forms (e.g., Google Forms, Qualtrics, SurveyMonkey) that are set up and collated by research team leads. The web-based form lists each unit of analysis and either a drop-down menu or box for coders to enter their codes. These procedures limit the number of people with access to original data files, which drastically lowers the likelihood of accidental data tampering or loss.

Research team leads provide digital copies of data files to coders; the coders edit and/or code data on the duplicate copies.

Note 1: Software Practices for Large Coder Teams

We use a variety of software tools to help coders record and keep track of codes. For projects that have a smaller group of coders (typically < 10) we use VERBI Software MAXQDA to tag and keep track of codes in text because we find it is the easiest QDA program on which to train novice coders quickly. In the past, we have also used Atlas.ti and NVivo for this purpose. However, for large coder teams we do not find it cost effective or practical to have large number of QDA software licenses. In this case, we use a modified software strategy in which we segment or unitize texts in Microsoft Excel and ask coders to record the presence or absence of codes for each segment of text either in a new column in the Excel document or via entry into a web-based form (as described in Challenge 3).

Establish pilot periods and re-assign coders when necessary. Pilot periods allow PIs to asses coder reliability, attitude, flexibility, and ability to work on a team. We establish a pilot period immediately following the training period, and allow for coders to complete a full task arc (i.e., all of the duties they are expected to perform). For example, we often work first with one code from our codebook—building, refining, and testing intercoder reliability, and applying that one code to a selected subset of documents before moving on to repeat these procedures for the rest of the codes in our codebook. This process allows us to (a) ensure that all coders on the team are performing tasks at the basic level of competency the project requires and, (b) promote coders who show interest and the ability to take on higher level and mentorship roles (i.e., as coding supervisors, as mentioned above), and (c) re-assign coders who do not meet the basic level of competence to other tasks. We lay out clear ground rules for the pilot period, including the duration of the period, the compensation structure of the period, and the performance expectations and incentive structure. We also establish evaluation procedures for the pilot period, including whether or not evaluation of the pilot period will be formal or informal. We clearly communicate with coders about all the options that will occur after the pilot period (e.g., assigned as coder, promoted to coding supervisor, reassigned to other tasks, as described in Table 5).

Use a "lead coder" approach for assessing intercoder reliability. Debates over ways to measure intercoder agreement or reliability are discussed in the voluminous literature on the

Table 6. Strategies and Examples for Building Team Cohesion and Morale.

Challenge	Strategies	Examples
Building team cohesion and morale	a) Develop rapport between project leads, coding supervisors, and coders and support career path development b) Create the feeling of a "home base" at a large organization, with a sense of community among coders c) Encourage senior/skilled coders to help intervene if junior coders are struggling with tasks d) Ensure that there is equivalent work across coders, even if some coders must be re-assigned to non-essential work e) Ensure that there is time and opportunity for community building (e.g., short breaks during work time, celebrating the completion of a complex task or project)	Coders often enjoy our lab's atmosphere so much that they recommend the experience to friends. In a large campus, we provide a comfortable setting, with a walk desk, refrigerator, coffee maker, and kitchen access. We hold regular cookie parties, informal chats, and celebrate student achievements outside of the lab (e.g., awards and entry into graduate school). Each year, some coders apply for new jobs, fellowships, or graduate school; we organize time when these coders could work together on personal statements, and then subsequently have them edit and revise reciprocally. Supervisors provide mentorship and feedback as well (also part of incentive structure).

topic (e.g., Armstrong et al., 1997; Campbell et al., 2013; Guba, 1981; LeCompte and Goetz 1982; MacPhail et al., 2016; Schwandt et al., 2007; Tracy, 2010). Common methods include using statistical measures such as Cohen's Kappa (Hruschka et al., 2004) or Krippendorf's Alpha (Krippendorf, 2018) and coming to intercoder consensus through repeated dialog and discussion over coding disagreements (Bernard et al., 2016; Cascio et al., 2019; Campbell et al., 2013). These strategies typically use teams of two to four coders. Calculating intercoder agreement and managing the process for achieving intercoder reliability becomes a tremendous challenge when working with teams of 10 + coders. While there are merits and drawbacks to all techniques for calculating intercoder reliability, we find quantitative measures useful when working on very large coder teams because they serve as an efficient baseline at which we can enter conversations about agreements and disagreements around coding (see Hruschka et al., 2004). On a team of 2–4 coders, it is much easier to detect agreement and have consensus based conversations (Cascio et al., 2019), but these methods can become burdensome and unproductive on a team of 20 + coders. To navigate this, we employ a "lead coder" approach to measure and come to intercoder agreement with a large coder team.

In the lead coder approach, the project leadership team constructs the initial version of the project codebook following MacQueen and colleagues' (1998) method to create detailed and structured codebook definitions for each code. After extensive pre-testing and refinement, we distribute this initial version of the codebook to the whole coding team to review. The "lead coder" (usually one person on the project leadership team) then samples the data set to create a coding test for the purposes of coder training and codebook refinement (usually about 25 text coding units from the data set). The lead coder, working with another member of the leadership team, codes the test set of data, following the initial codebook. Agreement is assessed, and any differences (typically, rare and minor at this point) are resolved through discussion. A final "test set" is then produced and used to onboard new coders to the project. Each new coder uses the initial codebook to code the test

set independently. Each coder then measures their coding agreement for each code with the "lead coder" (we most often use Cohen's kappa to assess intercoder reliability, but when this measure is inappropriate to the data set and research question, we employ alternative techniques [see Barbour, 2001; Krippendorf, 2018; Tracy, 2010]). Depending on amount of disagreement, the size of the team, and coders' topic/site expertise, the whole coding team (including the lead coder) may collectively discusses coding agreements and disagreements and revise the codebook accordingly, following the process outlined by Campbell et al. (2013). In this scenario, the lead coder then re-samples the data set to create a new coding test and the process is repeated until coders achieve an acceptable level of agreement for each code with the lead coder. Key to our process here is our commitment that the "lead coder" does not have the authority to mandate that their coding is "correct" in the initial rounds of codebook development. Coding disagreements are discussed and mutually rectified between the lead coder, project leadership team, and coders with relevant topic/site expertise. In this way, the lead coder and each team member can be considered a dyad of independent coders who come acceptable levels of negotiated agreement (Campbell et al., 2013).

This process of using a lead coder has several advantages. (1) An experienced lead coder with detailed knowledge of the data set and the codebook provides hands-on training and imparts conceptual knowledge to trainee coders through the codebook refinement process. (2) Open discussion enables novice coders to develop more nuanced conceptual understandings of the code by hearing the ways that other coders had thought through and applied the codes (including the lead coder). (3) Points of disagreement among coders and the lead coder help to refine the codebook as coders and the lead coder collectively discuss and reconcile their disagreements. (4) This process provides a test of coder competence. Coders who are not able to achieve an acceptable level of intercoder agreement with the lead coder after multiple rounds of codebook refinement are re-assigned to other tasks.

Create 100% redundancy in coding procedures. In addition to evaluations of coder competence, we build 100% coding

redundancy into our coding procedures when working with large coder teams. For example, on the project that consisted of a team of 54 student coders who were tasked with coding 3,120 drawings, we assigned each of the 54 coders two sets of 58 drawings to code (116 drawings total, per coder). This occurred after 10 + rounds of codebook revision and refinement and all coders reaching acceptable levels of intercoder reliability with the lead coder, as described above. By assigning each coder two sets of drawings to code, each drawing in the data set ($n = 3,120$) was coded twice, by two different coders. Coders entered their codes into a web-based form, which produced a matrix containing the unique IDs for each drawing and the presence or absence count for each of the eight codes in the codebook. This reporting and documentation strategy allowed the leadership team to easily compare codes for each unique drawing across the two coders. Any coding disagreements between the two coders were rectified by the lead coder given their expertise and contextual knowledge of the project (Campbell et al., 2013; Krippendorf, 2018). This redundancy in coding ensured that the final data set contained few, if any, coding errors. Furthermore, this double-coding structure encouraged coders to be more accountable for their coding because they were aware that accuracy checks were built into the labor structure.

Foster a culture that prioritizes data quality and ethical social science. Strategies and procedures for ensuring data quality on large coder teams work best, we find, when they operate in a work culture that prioritizes data quality and ethical social science. We build such a culture very intentionally through formal and informal norms: affirming ethical commitments on all project and team documents, training periods dedicated to social science research ethics, reiterating the role of data quality in ethical social science research, and setting aside time to discuss data quality and ethical research procedures collectively as a team. When problems arise in the research process, we problem-solve by centering approaches that uphold our ethical commitments and the quality of our data. A culture that prioritizes data quality and ethical research creates countless informal reminders to coders to maintain and protect data quality as an ethical responsibility at all times.

Take-Home Tip 3: Devise multiple layers of data protection to maintain data quality

We find there is not one main technique to ensure data quality when working with large coder teams. We layer data protections by using technological tools, providing intensive coder training, creating redundancy in our coding procedures, and fostering an atmosphere that prioritizes data quality and ethical social science research. Different means of data protection will be more or less appropriate for different types of teams, but ensuring data quality through multiple angles has been key to our success.

Challenge 4: Building team cohesion and morale

Human coding can be boring and mentally taxing—especially when done correctly (Cascio et al., 2019; Giesen and Roeser 2020; Hruschka 2004; Lichtenstein and Rucks-Ahidiana 2021). Motivation to stay focused and on task is crucial. Team spirit and camaraderie help cultivate an environment in which coders are invested in and excited about the project. This environment helps to prevent coder disaffection, ensure efficient work flow, preemptively avoid conflict, and creates a productive atmosphere for dealing with conflict when it inevitably arises. In essence, team spirit is the ideal of any team-based research, but maintaining team spirit and cohesion on a team with so many coders—who may be with the project for only a few months—can be an enormous challenge (see Table 6 for summary).

Guidance for building team cohesion and morale

Prioritize and schedule social connection as part of team coding efforts. Coding qualitative data is arduous work that requires great focus and can take a long time. We set realistic timeframes for coders to complete their assigned coding (that allow for coders to take breaks and do not require too much coding in any one day), but also plan time for coders to decompress, socialize, and talk about their ongoing work. These periods of informal social time—in which coders may bring up the curiosities they have encountered in texts, or the examples they see over and over—can assist with team bonding and fostering team spirit. We have found that they also can lead to important new insights in the data or potential new spinoff projects. We build in times for team bonding, such as having coffee hours or allowing 10–15 minutes of free chat before a team meeting formally begins.

Take-Home Tip 4: Prioritize team-spirit to build team cohesion and morale

There are many different ways to build team cohesion and morale, but we find that deliberately fostering a strong sense of team-spirit in the research process is key to navigating the inevitable setbacks, pitfalls, and communication difficulties of team-based research. We cultivate this through clear communication and established procedures and repeatedly emphasizing (and recognizing) all team contributions to the research process.

Emphasize the team-based nature of research as a whole. We ensure that all team members know upfront the team rules,

procedures, and expectations, including the baseline levels of competency we expect for coding a particular project. If a coder does not meet those competency levels for a particular project and needs to be reassigned (as described in Table 5) we maintain morale by emphasizing the importance of all research tasks and explain coding to be one element of the research team.

Institute consistent communication (in-person and virtual) with well-established procedures, values, expectations, codes of conduct around scholarly contributions. We maintain a consistent commitment to appropriately recognizing the contributions of coders in our lab work. Recognition can take many forms (e.g., authorship credit, author order, acknowledgements, etc.). Since we founded our lab, norms around recognition of work have shifted in the academy. Lab work by undergraduates is now increasingly likely to be credited with authorship—or expected to be.

We rely heavily on external guides for how to assign credit, in part to maintain consistency through time when dealing with multiple undergraduate collaborators, while also being able to change as norms shift. For some years, we have used the International Committee of Medical Journal Editors (ICMJE)'s roles and responsibilities guidance (ICMJE, 2021) as our baseline for explaining and sharing transparent expectations around co-authorship. We follow HWISE guidelines for forming consortia to share co-authorship across large international collaborative teams (Jepson et al., 2020).

Given the centrality of anti-racism work to our lab philosophy, we also consider the Civic Laboratory for Environmental Action Research (CLEAR) guidelines (Liboiron et al., 2017). These guidelines help us prioritize junior and marginalized scholars in the recognition and ordering of co-authorship contributions. Being able to share and follow external guidelines, particularly as they are updated, allows adjustments without creating confusion or inconsistencies between lab members and through time.

Note 2: Managing Large Coder Teams During Large-Scale Disruptions

At the time of this writing, we are 18 months into the COVID-19 global pandemic. While the guidance provided here has been developed from our fully-completed projects (which we consider as those where the analyses have appeared in at least one peer-reviewed publication), we have continued to conduct these same lab activities on projects throughout 2020 and 2021 through lockdowns and other disruptions, switching in March 2020 to a synchronous online-only modality. All meetings occurred over zoom. In August 2021, we switched back to in-person lab activities. But the online modality was sufficiently successful that we are continuing to also offer parallel synchronous online options for students moving forward. In adapting these processes to online, we

used a "work alongside" strategy where coders worked remotely but met in prescribed two-hour blocks through the week over Zoom with a faculty member or lab manager in those time blocks. This way, there was always someone available to answer questions as they coded in real time, and a sense of access to and engagement with others in the lab. Overall, for those students unable to be physically on campus, this worked well for all involved. Given disruption is inevitable in global collaboration, strategies for shifting large coder team management online are crucial to have in place.

Research benefits of large coder teams

Despite significant challenges posed by large coder teams, we find that they offer tremendous advantages for qualitative data analysis. If effective, efficient, and equitable procedures for recruitment and training are in place, large coder teams undoubtedly enable researchers to process qualitative data in much less time than a smaller team. In an era of big data, well-run large coder teams open up possibilities to analyze new research questions and work with new data sets that may not have been possible for a small team of coders to tackle. But, perhaps more significantly, large coder teams also have the potential to include a far greater amount of diversity of insights into the analysis process, especially through procedures for codebook refinement. More coders often mean more perspectives and ideas that can be incorporated into the process of refining codes—especially if researchers make a concerted effort to recruit coders from a diversity of backgrounds, cultures, language expertise, and experiences. This process often translates into deeper and more nuanced codes and being able to explain analytical constructs in more concrete ways when reporting research results.

Notes of caution

Despite the significant advantages of large coder teams, we highlight three notes of caution for researchers looking to mobilize large coder teams in their own research.

First, we find that large coder teams are best suited for highly structured coding. In our experience, analysis with a large coder team works best when a smaller team of researchers works to develop the initial version of a codebook (inductively or deductively), and then bring on a larger team of coders to refine the codebook via initial coding tests and discussion. During the initial process of codebook development, too many team members can result in too many ideas that end up creating overly complex codes and codebooks. Thus, highly inductive projects, such as grounded theory or schema analysis projects, are not well suited to large coding teams.

Second, large coder teams require time and resources. While one of the primary goals of employing a large coding

team to process data is to *save* time and code data more efficiently, it is important that researchers recognize that large coder teams nonetheless require significant time investments in training and resources, including communication procedures, preparation of meeting materials, and back-end quality checking. We have found that the time savings from using large coder teams typically occur after these administrative processes and procedures are in place and repurposed for multiple studies. Therefore, we caution researchers to consider whether employing a large coder team will save time, especially if they may be doing so for only one project or without standing research infrastructure in place. Cutting corners in any of these areas when there is not enough time or resources available can lead to disastrous consequences, including poor data quality, disaffected team members, and inability to finish a project.

Third, in our experience, burdensome oversight structures are typically necessary to ensure high-quality analysis with large coder teams. We have found hierarchical leadership structures to be necessary in order to establish chains of supervision, reporting, and accountability as well as to ensure such vital functions as safety, consistency, follow-through, and record-keeping. The reputation of the research group depends on its ability to consistently deliver high-quality coding and analysis. While such hierarchies can be burdensome for all involved, oversight is crucial to ensuring that coding errors are identified and corrected in a timely manner. That said, overly rigid hierarchies are unhelpful to building collaborative teams, and we encourage avoiding leadership hubris and being open to the contribution of ideas, feedback, and criticism from every team member.

Conclusion

Human coding of text—especially large volumes of text or when using many codes—is a laborious and often boring task. In qualitative projects, it can represent a large percentage of the cost of conducting research, in time or funds. While machine-based coding helps with challenges of volume, many researchers recognize that the insights they seek require the subtleties only trained coders can provide. We have provided some perspectives, based on our collective experience in a large qualitative lab leading 18 cross-cultural studies, on how large coding teams can be activated, managed, and sustained to move coding forward faster and at greater scale. These solutions are imperfect and evolving, and will not suit everyone, but we hope this provides researchers an additional option to consider as they expand the reach and scale of their research.

ORCID iDs

Melissa Beresford https://orcid.org/0000-0002-5707-3943
Amber Wutich https://orcid.org/0000-0003-4164-1632

References

Agostini, G., SturtzSreetharan, C., Wutich, A., Williams, D., & Brewis, A. (2019). Citizen Sociolinguistics: A new method for understanding fat talk and other sociolinguistic phenomena. *PLoS ONE*, *14*(5), e0217618. https://doi.org/10.1371/journal.pone.0217618

Armstrong, D., Gosling, A., Weinman, J., & Marteau, T. (1997). The place of inter-rater reliability in qualitative research: An empirical study. *Sociology*, *31*(3), 597–606. https://doi.org/10.1177/0038038597031003015

Barbour, R. S. (2001). Checklists for improving rigour in qualitative research: a case of the tail wagging the dog?. *Bmj*, *322*(7294), 1115–1117. https://doi.org/10.1136/bmj.322.7294.1115

Benoit, K., Conway, D., Lauderdale, B. E., Laver, M., & Mikhaylov, S. (2016). Crowd-sourced text analysis: Reproducible and agile production of political data. *American Political Science Review*, *110*(2), 278–295. https://doi.org/10.1017/s0003055416000058

Bernard, H. R., Wutich, A., & Ryan, G. W. (2016). *Analyzing qualitative data: Systematic approaches*. SAGE publications.

Bozeman, D. P., Street, M. D., & Fiorito, J. (1999). Positive and negative coauthor behaviors in the process of research collaboration. *Journal of Social Behavior and Personality*, *14*(2), 159.

Braun, V., & Clarke, V. (2014). What can "thematic analysis" offer health and wellbeing researchers?. *International Journal of Qualitative Studies on Health and Well-being*, *9*(1), 26152. https://doi.org/10.3402/qhw.v9.26152.

Brewis, A. A, Gartin, M., Wutich, A., & Young, A. (2013). Global convergence in ethnotheories of water and disease. *Global Public Health*, *8*(1), 13–36. https://doi.org/10.1080/17441692.2012.758298

Brewis, A., Meehan, K., Beresford, M., & Wutich, A. (2021). Anticipating elite capture: the social devaluation of municipal tap

water users in the Phoenix metropolitan area. *Water International*, *46*(6), 1–20. https://doi.org/10.1080/02508060.2021.1898765

Brewis, A, Wutich, A, du Bray, MV, Maupin, J, Schuster, RC, & Gervais, MM (2019). Community hygiene norm violators are consistently stigmatized: Evidence from four global sites and implications for sanitation interventions. *Social Science & Medicine*, *220*, 12–21. https://doi.org/10.1016/j.socscimed.2018.10.020

Burla, L, Knierim, B, Barth, J, Liewald, K, Duetz, M, & Abel, T (2008). From text to codings: intercoder reliability assessment in qualitative content analysis. *Nursing Research*, *57*(2), 113–117. https://doi.org/10.1097/01.NNR.0000313482.33917.7d

Campbell, J. L., Quincy, C., Osserman, J., & Pedersen, O. K. (2013). Coding in-depth semistructured interviews: Problems of unitization and intercoder reliability and agreement. *Sociological Methods & Research*, *42*(3), 294–320. https://doi.org/10.1177/0049124113500475

Carey, J. W., Morgan, M., & Oxtoby, M. J. (1996). Intercoder agreement in analysis of responses to open-ended interview questions: Examples from tuberculosis research. *CAM Journal*, *8*(3), 1–5. https://doi.org/10.1177/1525822x960080030101

Cascio, M. A., Lee, E., Vaudrin, N., & Freedman, D. A. (2019). A team-based approach to open coding: Considerations for creating intercoder consensus. *Field Methods*, *31*(2), 116–130. https://doi.org/10.1177/1525822x19838237

Crona, B., Wutich, A., Brewis, A., & Gartin, M. (2013). Perceptions of climate change: Linking local and global perceptions through a cultural knowledge approach. *Climatic Change*, *119*(2), 519–531. https//:doi.org/10.1007/s10584-013-0708-5

du Bray, M. V., Wutich, A., & Brewis, A. (2017a). Hope and Worry: Gendered Emotional Geographies of Climate Change in Three Vulnerable US Communities. Weather, Climate. *and Society*, *9*(2), 285-297. https://doi.org/10.1175/wcas-d-16-0077.1

du Bray, M. V, Wutich, A, Larson, K. L, White, D. D, & Brewis, A (2017b). Emotion, coping, and climate change in island nations: Implications for environmental justice. *Environmental Justice*, *10*(4), 102-107.

du Bray, M., Wutich, A., Larson, K. L., White, D. D., & Brewis, A. (2019). Anger and sadness: Gendered emotional responses to climate threats in four island nations. *Cross-Cultural Research*, *53*(1), 58–86. https://doi.org/10.1177/1069397118759252

Gartin, M., Crona, B., Wutich, A., & Westerhoff, P. (2010). Urban ethnohydrology: cultural knowledge of water quality and water management in a desert city. *Ecology and Society*, *15*(4). https://doi.org/10.5751/es-03808-150436

Gartin, M, Larson, KL, Brewis, A, Stotts, R, Wutich, A, White, D, & du Bray, M (2020). Climate change as an involuntary exposure: A comparative risk perception study from six countries across the global development gradient. *International Journal of Environmental Research and Public Health*, *17*(6), 1894. https://doi.org/10.3390/ijerph17061894

Giesen, L., & Roeser, A. (2020). Structuring a Team-Based Approach to Coding Qualitative Data. *International Journal of Qualitative Methods*, *19*, 1609406920968700. https://doi.org/10.1177/1609406920968700

Guba, E. G. (1981). Criteria for assessing the trustworthiness of naturalistic inquiries. *ECTJ*, *29*(2), 75–91. https://doi.org/10.1007/bf02766777

Hall, W. A, Long, B., Bermbach, N., Jordan, S., & Patterson, K. (2005). Qualitative teamwork issues and strategies: Coordination through mutual adjustment. *Qualitative Health Research*, *15*(3), 394–410. https://doi.org/10.1177/1049732304272015

Hruschka, D. J., Schwartz, D., St.John, D. C., Picone-Decaro, E., Jenkins, R. A., & Carey, J. W. (2004). Reliability in coding open-ended data: Lessons learned from HIV behavioral research. *Field Methods*, *16*(3), 307–331. https://doi.org/10.1177/1525822x04266540

ICMJE (2021). *Defining the roles of authors and contributors. International Committee of Medical Journal.* http://www.icmje.org/recommendations/browse/roles-and-responsibilities/defining-the-role-of-authors-and-contributors.html

Jepson, W., Stoler, J., Young, S., & Wutich, A. (2020). *NSF Household Water Insecurity Experiences (HWISE) - Research coordination network (RCN) data use and Co-authorship guidelines.* https://hwise-rcn.org/hwise-community/collaboration/guidelines-principles/.

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology.* Sage publications.

Kurasaki, K. S. (2000). Intercoder reliability for validating conclusions drawn from open-ended interview data. *Field Methods*, *12*(3), 179–194. https://doi.org/10.1177/1525822x0001200301

Larson, K. L., Stotts, R., Wutich, A., Brewis, A., & White, D. (2016). Cross-cultural perceptions of water risks and solutions across select sites. *Society & Natural Resources*, *29*(9), 1049–1064. https://doi.org/10.1080/08941920.2015.1122132

LeCompte, M. D., & Goetz, J. P. (1982). Problems of reliability and validity in ethnographic research. *Review of Educational Research*, *52*(1), 31–60. https://doi.org/10.3102/00346543052001031

Liboiron, M., Ammendolia, J., Winsor, K., Zahara, A., Bradshaw, H., Melvin, J., Mather, C., Dawe, N., Wells, E., Liboiron, F., Fürst, B., Coyle, C., Saturno, J., Novacefski, M., Westscott, S., & Liboiron, G. (2017). Equity in author order: a feminist laboratory's approach. *Catalyst: Feminism, Theory, Technoscience*, *3*(2), 1–17. https://doi.org/10.28968/cftt.v3i2.28850

Lichtenstein, M, & Rucks-Ahidiana, Z (2021). Contextual Text Coding: A Mixed-methods Approach for Large-scale Textual Data. *Sociological Methods & Research*, 0049124120986191.

Liggett, A. M., Glesne, C. E., Johnston, A. P., Hasazi, S. B., & Schattman, R. A. (1994). Teaming in qualitative research: Lessons learned. *Qualitative Studies in Education*, *7*(1), 77–88. https://doi.org/10.1080/0951839940070106

MacPhail, C., Khoza, N., Abler, L., & Ranganathan, M. (2016). Process guidelines for establishing intercoder reliability in qualitative studies. *Qualitative Research*, *16*(2), 198–212. https://doi.org/10.1177/1468794115577012

MacQueen, K. M., McLellan, E., Kay, K., & Milstein, B. (1998). Codebook development for team-based qualitative analysis. *Cam Journal*, *10*(2), 31–36. https://doi.org/10.1177/1525822x980100020301

Moret, M., Reuzel, R., Van Der Wilt, G. J., & Grin, J. (2007). Validity and reliability of qualitative data analysis: Interobserver

agreement in reconstructing interpretative frames. *Field Methods*, *19*(1), 24–39. https://doi.org/10.1177/1525822x06295630

Nelson, L. K., Burk, D., Knudsen, M., & McCall, L. (2021). The future of coding: A comparison of hand-coding and three types of computer-assisted text analysis methods. *Sociological Methods & Research*, *50*(1), 202–237. https://doi.org/10.1177/0049124118769114

Pokorny, JJ, Norman, A, Zanesco, AP, Bauer-Wu, S, Sahdra, BK, & Saron, CD (2018). Network analysis for the visualization and analysis of qualitative data. *Psychological Methods*, *23*(1), 169–183. https://doi.org/10.1037/met0000129

Rice, J., Stotts, R., Wutich, A., White, D., Maupin, J., & Brewis, A. (2019). Motivators for treated wastewater acceptance across developed and developing contexts. *Journal of Water, Sanitation and Hygiene for Development*, *9*(1), 1–6. https://doi.org/10.2166/washdev.2018.285

Richards, L. (1999). Qualitative teamwork: Making it work. *Qualitative Health Research*, *9*(1), 7–10. https://doi.org/10.1177/104973299129121659

Robins, C. S., & Eisen, K. (2017). Strategies for the effective use of NVivo in a large-scale study: Qualitative analysis and the repeal of Don't Ask, Don't Tell. *Qualitative Inquiry*, *23*(10), 768–778. https://doi.org/10.1177/1077800417731089

Roque, A, Wutich, A, Brewis, A, Beresford, M, García-Quijano, C, Lloréns, H, & Jepson, W (2021). Autogestión and water sharing networks in Puerto Rico after Hurricane María. *Water International*, *46*(6), 938-955.

Ruth, A., Brewis, A., & SturtzSreetharan, C. (2021). Effectiveness of social science research opportunities: A study of course-based undergraduate research experiences (CUREs). *Teaching in Higher Education*. https://doi.org/10.1080/13562517.2021.1903853

Ruth, A., Wutich, A., & Brewis, A. (2016). The global ethnohydrology study: Integrating global health undergraduates in collaborative research. *Practicing Anthropology. October*, *38*(4), 16–18.

Ruth, A., Wutich, A., & Brewis, A. (2019). A model for scaling undergraduate research experiences: The global ethnohydrology study. *International Journal of Mass Emergencies and DisastersMarch*, *37*(1), 25–34.

Ryan, G. (1999). Measuring the typicality of text: Using multiple coders for more than just reliability and validity checks (*58*(3), 313–322). *Human Organization*. https://doi.org/10.17730/humo.58.3.g224147522545rln

Ryan, G. W., & Bernard, H. R. (2003). Techniques to identify themes. *Field Methods*, *15*(1), 85–109. https://doi.org/10.1177/1525822x02239569

Schuster, R. C., Butler, M. S., Wutich, A., Miller, J. D., & Young, S. L. (2020). Household Water Insecurity Experiences-Research Coordination Network (HWISE-RCN), ... & Workman, C "If there is no water, we cannot feed our children": The far-reaching consequences of water insecurity on infant feeding practices and infant health across 16 low-and middle-income countries. *American Journal of Human Biology*, *32*(1), e23357. https://doi.org/10.1002/ajhb.23357

Schwandt, T. A., Lincoln, Y. S., & Guba, E. G. (2007). Judging interpretations: But is it rigorous? Trustworthiness and authenticity in naturalistic evaluation. *New Directions for Evaluation*, *2007*(114), 11–25. https://doi.org/10.1002/ev.223

Stotts, R., Rice, J., Wutich, A., Brewis, A., White, D., & Maupin, J. (2019). Cross-cultural knowledge and acceptance of wastewater reclamation and reuse processes across select sites. *Human Organization*, *78*(4), 311–324. https://doi.org/10.17730/0018-7259.78.4.311

SturtzSreetharan, C. L., Agostini, G., Brewis, A. A., & Wutich, A. (2019). Fat talk: A citizen sociolinguistic approach. *Journal of Sociolinguistics*, *23*(3), 263–283. https://doi.org/10.1111/josl.12342

SturtzSreetharan, C, Ruth, A, Wutich, A, Glegziabher, M, Mitchell, C, Bernard, H R, & Brewis, A (2021). Citizen Social Scientists' Observations on Complex Tasks Match Trained Research Assistants', Suggesting Lived Experiences are Valuable in Data Collection. *Citizen Science: Theory and Practice*, *6*(1).

Tracy, S. J. (2010). Qualitative quality: Eight "big-tent" criteria for excellent qualitative research. *Qualitative Inquiry*, *16*(10), 837–851. https://doi.org/10.1177/1077800410383121

Trainer, S, Brewis, A, Hruschka, D, & Williams, D (2015a). Translating obesity: Navigating the front lines of the "war on fat". *American Journal of Human Biology*, *27*(1), 61–-68. https://doi.org/10.1002/ajhb.22623

Trainer, S., Brewis, A., Williams, D., & Chavez, J. R. (2015b). Obese, fat, or "just big"? Young adult deployment of and reactions to weight terms. *Human Organization*, *74*(3), 266–275. https://doi.org/10.17730/0018-7259-74.3.266

Trainer, S., Brewis, A., & Wutich, A. (2021). *Extreme weight loss: Life before and after bariatric surgery*. NYU Press, pp. 213.

Vins, H., Wutich, A., Brewis, A., Beresford, M., Ruth, A., & Roberts, C. (2014). Children's perceived water futures in the United States southwest, *73*(3), 235–246). Human Organization. https://doi.org/10.17730/humo.73.3.68101441563654w7

Whitehead, T. L. (2005). Basic classical ethnographic research methods. ethnographically informed community and cultural assessment research systems (EICCARS) working paper series. *Cultural Ecology of Health and Change*, *1*, 1–29.

White, D. E., Oelke, N. D., & Friesen, S. (2012). Management of a large qualitative data set: Establishing trustworthiness of the data. *International Journal of Qualitative Methods*, *11*(3), 244–258. https://doi.org/10.1177/160940691201100305

Wutich, A., Brewis, A., York, A. M., & Stotts, R. (2013). Rules, norms, and injustice: a cross-cultural study of perceptions of justice in water institutions. *Society & Natural Resources*, *26*(7), 795–809. https://doi.org/10.1080/08941920.2012.723302

Wutich, A., Ruth, A., Brewis, A., & Boone, C. (2014). Stigmatized neighborhoods, social bonding, and health. *Medical Anthropology Quarterly*, *28*(4), 556–577. https://doi.org/10.1111/maq.12124

Wutich, A., York, A. M, Brewis, A., Stotts, R., & Roberts, C. M. (2012). Shared cultural norms for justice in water institutions: Results from Fiji, Ecuador, Paraguay, New Zealand, and the US. *Journal of Environmental Management*, *113*, 370–376. https://doi.org/10.1016/j.jenvman.2012.09.010