

Qualitative Methods with Nvivo Software: A Practical Guide for Analyzing Qualitative Data

David B. Allsop^{1,*} , Joe M. Chelladurai², Elisabeth R. Kimball², Loren D. Marks²  and Justin J. Hendricks² 

¹ Department of Psychology and Neuroscience, Dalhousie University, Life Sciences Centre Rm 3263, 1355 Oxford St., Halifax, NS B3H 4R2, Canada

² School of Family Life, Brigham Young University, Joseph F. Smith Building Rm 2086, Provo, UT 84602, USA; joe.chelladurai@live.in (J.M.C.); ellierose2014@live.com (E.R.K.); loren_marks@byu.edu (L.D.M.); justin_hendricks@byu.edu (J.J.H.)

* Correspondence: david.allsop@dal.ca

Abstract: From 1995–2016, there has been a 15-fold increase in qualitative scholarship in the social sciences, but the rigor and quality of published work has ranged widely. Little scholarship provides concrete, pragmatic explanations of (and directions regarding) the execution of systematic, high-rigor qualitative analysis. The present article guides the developing qualitative researcher through technical and procedural aspects of analyzing qualitative data with specific attention to reliability and rigor. Guidance addressing transcription, importing data, forming coding pairs, performing initial/open coding (examples of three types), determining core themes, systematic team-based coding, maintaining a data audit trail, creating a Numeric Content Analysis (NCA) table, and preparing work for publication is provided. Materials include several tables and figures that offer practical demonstrations on how to use Nvivo in data analysis. Transcription tips and outsourcing benefits and cautions are also offered. Altogether, the present article provides qualitative researchers practical guidance for executing multiple stages of qualitative analysis.



Citation: Allsop, D.B.; Chelladurai, J.M.; Kimball, E.R.; Marks, L.D.; Hendricks, J.J. Qualitative Methods with Nvivo Software: A Practical Guide for Analyzing Qualitative Data. *Psych* **2022**, *4*, 142–159. <https://doi.org/10.3390/psych4020013>

Academic Editor: Broderick Patricia

Received: 24 January 2022

Accepted: 18 March 2022

Published: 22 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: qualitative methods; coding; interviews; semi-structured interview; Nvivo

1. Introduction

From 1995 to 2016, there has been a 15-fold increase in qualitative scholarship in the social sciences [1]. The rise in qualitative scholarship has been helpful for social science in general. Scholars have remarked that qualitative inquiry has brought pluralist orientations to theory and practice and has inspired theory building, fostered greater inclusion of minorities, and promoted interdisciplinary collaborations [2]. Some journals that were once reluctant to publish qualitative research are becoming more methodologically inclusive and encouraging high-rigor qualitative publications, though some journals have remained impermeable by qualitative work. The field has faced the twin circumstance of a dramatic increase in qualitative publications—based on data collected both through traditional and online sources see [3]—as well as a wide array of quality and systematic rigor in such work [4]. This influx has established the need for related reporting standards introduced by the American Psychological Association [5]. With more consumption, production, and standardization of this mode of inquiry, and calls for more widespread education in this method among undergraduate and graduate students [6,7], we hope to see a rise, not only in quantity, but also in accountability (see [8]), rigor, and standards reflected in qualitative scholarship generally.

Marks [4] has noted that a “core question that qualitative researchers need to address—first to themselves and then to their readers [and reviewers]” is the following: “*Is your research method ‘science’ and if so, why?*” (p. 495, emphasis in original). A primary issue is that of replicability. Marks [4] proposed a systematic, team-based approach to foster replicability and rigor in qualitative analysis that they co-developed over two decades.

During the interim time, however, qualitative software has seen significant development and advancement that allows researchers to document reliability, rigor, and replicability, when intentionally utilized to do so. Marks [4] expressed skepticism regarding highly idiosyncratic approaches, where a single qualitative researcher produced an entire study from beginning to end without external “checks and balances” in place—referring to such work as the “monk/nun in a cell” method (p. 497).

With the advent of computer technology in qualitative analysis, we now have various software packages that offer ways to transcribe, input, and conduct preliminary to sophisticated analysis. Critics of software usage in qualitative analysis cite that the software creates distance by the computer “doing the analysis” instead of doing it “by hand” [9] (p. 1). Rebuttals to this argument refer to the fact that software packages are not necessarily meant to analyze data, but to aid the analysis process [10].

While the usage of qualitative software is increasing, there is little to no report of how particular software packages are used in conducting a particular analysis of a study. Although such studies cite user guides, or generic procedures of conducting analysis, they do not necessarily mention the exact process of how software is being used to analyze and arrive at findings or themes. There is a lack of specific detail in reporting that may be a cause for disturbance for quantitatively minded, methodologically rigorous reviewers. This common hindrance may not just be avoided but, when addressed through adequate reporting, may provide an opportunity to encourage replication and verification that can further methodological rigor.

Accordingly, the goal of the current paper is to provide step-by-step instructions on Marks’ [4] process for qualitative data analysis with NVivo software. Specifically, we provide guidance on preparing interview data for analysis, finding core themes through open coding (i.e., initial coding) of interviews (our description of open coding equates roughly to Marks [4] Phase Two: steps 5–17 (p. 496)), choosing core themes, systematically coding the data for those themes (our description of systematic coding equates roughly to Marks [4] Phase Two: step 18 (p. 496)), and finally preparing the results for publication.

It is important to note that this manuscript does not include descriptions of several key parts of the qualitative research process. We do not describe the data collection process, interview procedures, or meaning making, and instead refer the reader to Marks [4] (for details on data collection, see p. 496 of Marks [4], Phase Two: steps 1–4. See also p. 500). Our focus is on the technical and procedural aspects of analyzing the data rather than methods of the data collection process. Furthermore, we focus primarily on interviews as the source of qualitative research, as opposed to case studies, observational studies, or other forms of qualitative work (another important qualification is that our steps are described for PCs rather than Macs. However, steps for Mac users would be similar).

2. Data Preparation

2.1. Key Terms

We introduce the reader to some key terms and abbreviations before diving in:

- *Code*—A meaningful word or phrase which represents and conveys the messages and meanings of participant words.
- *Coding*—The process of looking through data to find, and additionally assign, codes to participant words.
- *Node*—A feature in NVivo which provides “Containers for your coding—they let you gather [data which has been assigned a code] in one place” [11] (p. 7).
- *SC*—Single-Click; one left click with a mouse.
- *DC*—Double-Click; two left clicks in quick succession with a mouse.
- *RC*—Right-Click; one right click with a mouse.
- *SC-HD*—Single-Click and Hold-Down; one left click with a mouse, keeping the pressure of the clicking finger held down.
- *R-SC*—Release-Single-Slick; release the pressure of the finger holding a single-click down.

2.2. Preparing Data for Analysis

Data transcription of data and importing data files into Nvivo are the first steps of *Author masked for review's* method with Nvivo. Transcription can range from using a basic audio player and word processing software to more specialized transcription software, including Nvivo itself. Our focus in the current manuscript is on technical procedures using Nvivo—accordingly, we refer the reader to Appendix A for some tips on transcribing (generally, transcription takes many hours to complete one hour of interview. It becomes a faster process for more experienced transcribers and transcription software often helps speed up the process as well, but on average, it may take two to three hours to transcribe one hour of interview with transcription software. Accordingly, outsourcing is a worthwhile consideration). After transcription, text files are imported into Nvivo by taking the following steps:

1. Open Nvivo and create a new, empty Nvivo file;
2. SC “Import” on the ribbon;
3. SC “Files”;
4. Navigate to the files of interest;
5. Select the desired file(s) to import into Nvivo;
6. SC “Open”;
7. SC “Import”;
8. SC “OK”;
9. Save.

3. Performing Open Coding

Qualitative analysis begins through open coding once transcripts are imported. Open coding is “reading through an interview and recording . . . a brief conceptual ‘code’ that reflects what the participant is discussing” [4] (p. 501). (Codes can vary widely. Codes we have utilized have been as short as “gender” or as long as “old ‘normal’ is gone forever [4] (p. 501)”). Codes can be combinations of nouns, verbs, adjectives, and other word forms such gerunds (words ending in “ing”). One important point with open coding is that it focuses more on coding what is being said rather than what the text means. Meaning making comes later in the process once all the data have been viewed, sorted through, and organized. In qualitative research, researchers do not want to put words in participants’ mouths as it would be mostly authors speaking rather than the participants. By focusing on what is said during the open coding process, it makes the remaining processes easier, as researcher thoughts and words have not been mixed in with the actual data.

Open coding differs across three methods. The first method is the *entire interview method* that applies to projects focused on analyzing transcripts from the beginning to end, each and every word. Most often (but not always), this method is applied when no prior coding work on transcripts has been performed. The second is the *deep dive method* that applies to projects which are focused on analyzing already coded work more deeply. The third method is the *keyword method* that applies to projects focused on a specific aspect of transcripts found through keyword searches. An example of the entire interview method can be found in [12], an example of the deep dive method can be found in [13], and an example of the keyword method can be found in [14].

We describe open coding beginning with the entire interview method as the deep dive method and keyword method build upon the entire interview method. Procedures discussed in the entire interview method, including details on the fundamentals of open coding, apply to the other two methods. Accordingly, we underscore the importance of grasping the entire interview method first before focusing on the other two methods.

We emphasize that in Marks’ method [4], multiple researchers conduct the coding work—including open coding and subsequently described systematic coding and partner review (see [4], pp. 497–498 for more on the value of multiple researchers coding a project). Before proceeding, two or more researchers should be available.

3.1. Open Coding through the Entire Interview Method

Leading up to this point, each individual researcher should have non-coded transcripts imported into an NVivo file and the file opened with Nvivo. They then open the code by performing the following steps (see also Figures 1 and 2):

1. SC “Files” in the left navigation menu to display a list of files (transcripts);
2. Open the transcript by performing a DC on the file name. Alternatively, RC the file name then SC the “open document” option;
3. SC inside the window with the text of the transcript so that “document tools” appears on the ribbon. Under “document tools”, (1) SC coding stripes > SC “all coding” and (2) SC “highlight” > SC “all coding”;
 - a. Note—coding stripes make it easy to view and track coding conducted by researchers
4. SC “nodes” in the left navigation menu. The node navigation pane will appear (which should be empty at this point);
5. Read through a section of text (about a paragraph) and determine a “code” that fits it adequately;
6. Once a code is decided upon, highlight the text and SC-HD at the start of the code > drag to the end of the code > R-SC;
7. SC-HD the highlighted text > drag mouse to empty space in the nodes pane > R-SC;
 - a. Note—there are additional ways to assign a text to a node—see [11] (p. 25)
8. In the new window that popped up type in the name of the code decided in step 4;
9. SC “okay”. A new node appears in the nodes pane with the name of the code;
10. Continue the process until the end of interview:
 - a. If the same code is present more than once (as it likely will be), repeat steps 5–6, but at step six drag the mouse to the re-occurring node and then R-SC.
 - b. If a new code becomes present (as it likely will), repeat steps 5–9.
11. Once you have finished coding the interview, go through it quickly one more time to ensure that no codes were missed. If necessary, repeat the steps 10a or 10b;
12. Save, open the next interview to be coded, and perform steps 10a or 10b. Do not delete the nodes in the nodes pane;
13. Perform step 12 until all interviews are coded.

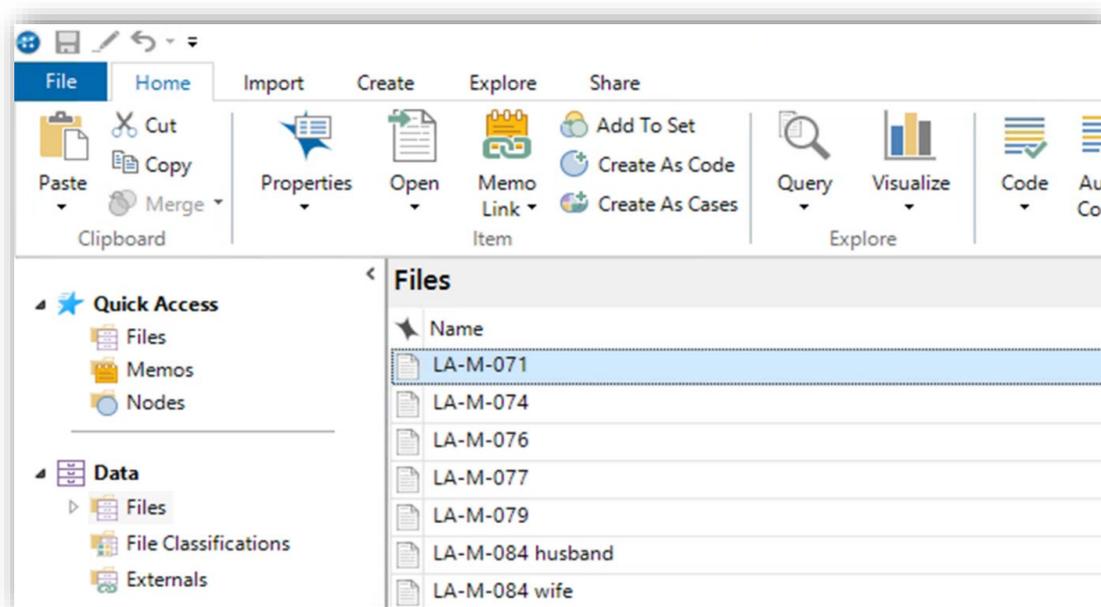


Figure 1. Open coding with the entire interview method screenshot 1 (Steps 1–3).

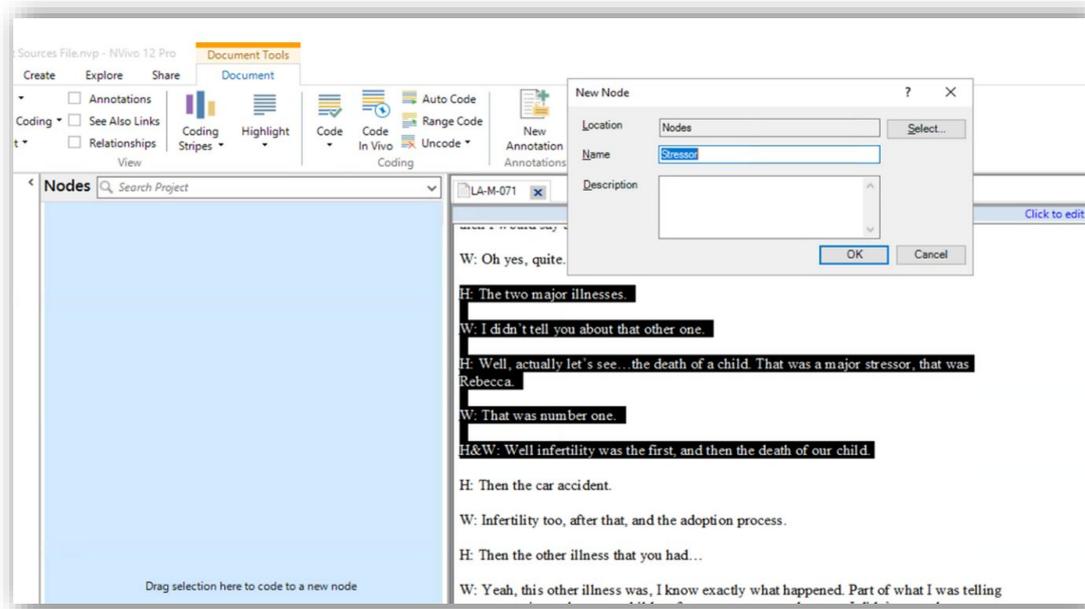


Figure 2. Open coding with the entire interview method screenshot 2 (Steps 4–8).

3.2. Open Coding through the Deep Dive Method

Coding using the deep dive method is similar to the entire interview method. The key difference is that instead of open-coding a collection of interviews, the researcher open-codes previously coded nodes. To perform open coding using this method, a researcher should have an Nvivo file open that contains at least one node that is ready to be analyzed further and then takes the following steps:

1. SC “Nodes” in the left navigation menu to display a list of nodes;
2. Open the node corresponding to the code you are performing a “deep dive” on by performing a DC on the node name. Alternatively, RC the node name then SC the “open node” option;
3. SC inside the window with the text of the transcript so that “document tools” appears on the ribbon. Under “document tools”, (1) SC coding stripes > SC “all coding” and (2) SC “highlight” > SC “all coding”;
4. SC “nodes” in the left navigation menu. The node navigation pane will appear (which should be empty at this point);
5. Read through a section of text (about a paragraph) and determine a “code” that fits it adequately. Ensure you only look for new codes (do not code at the node you are doing the deep dive on);
6. Once a code is decided upon, highlight the text and SC-HD at the start of the code > drag to the end of the code > R-SC;
7. SC-HD the highlighted text > drag mouse to empty space in the nodes pane > R-SC;
8. In the new window that popped up type in the name of the code decided in step 4;
9. SC “okay”. A new node appears in the nodes pane with the name of the code;
10. Continue the process until the end of node:
 - a. If the same code is present more than once (as it likely will be), repeat steps 5–6 but at step six drag the mouse to the re-occurring node and then R-SC.
 - b. If a new code becomes present (as it likely will), repeat steps 5–9.
11. Save.

3.3. Open Coding through the Keyword Method

Open coding through the keyword method is similar to the deep dive method. The key difference is that the results of a search are transformed into a node, and that node is

further analyzed. To perform open coding using this method, a researcher should have transcripts (either non-coded or coded) imported into an Nvivo file and the file opened with Nvivo. They should also have identified (a) a key construct or idea they wish to search for (e.g., health behaviors) and (b) the search terms related to the construct or idea (e.g., exercise, diet, sleep). Researchers should include as many relevant search terms as possible (e.g., expanding the initial list of search terms through use of a thesaurus). Open coding through this step is broken up into steps performed by a single researcher on behalf of the entire coding team and subsequent steps performed by all coders. Open coding through the keyword method begins with a single researcher taking the following steps (see Figure 3 for illustration):

1. SC "Explore" on the ribbon to display the query command group;
2. SC "Text Search";
3. SC inside the "search for" box;
4. Ensuring that the "find" slider is set at "exact matches" and the "spread to" drop down box is set to "none", enter keywords into the "search for" box utilizing the wildcard "*" after (and potentially before) keywords;
 - a. Note—the wildcard "*" enables stems of words to be found instead of only the exact word itself. There is an option on the slider to utilize "with stemmed words". However, utilizing the wildcard "*" instead of the "with stemmed words" option on the slider yields more search hits. For instance, searching one of our datasets with the criteria "health* disease* sick*" and the slider set on "exact matches" yielded 140 matching files and 507 matching references. However, searching our dataset with the criteria "health disease sick" with the slider on "stemmed words" yielded 121 files and 389 references. Accordingly, we recommend using wildcards and setting the find slider set to "exact matches"
 - b. Note—there is an option on the "find" slider to include synonyms by selecting "with synonyms". We recommend not using this option and rather finding synonyms beforehand and using the "exact matches" option. Our reasoning is (1) the researcher knows exactly what words Nvivo used in the search and (2) the researcher has the exact words used for searching available to report
5. SC "Run Query";
6. Review search results by SC on either "reference" or "word tree" on the right-hand ribbon next to the search results ("reference" shown in Figure 3) and adjust search terms as necessary;
 - a. Note—finalizing the search iterative process and adding and subtracting words through trial and error is helpful
 - b. Note—Word trees are helpful for visualizing the search and are one aesthetic way to display search terms for reports, posters, presentations and other publications (for more information on word trees, see http://help-nv11.qsrinternational.com/desktop/procedures/run_a_text_search_query.htm; accessed 17 March 2022)
7. If desired, save word tree by RC in the window with the word tree > SC "Export Word Tree" > Select File Location > SC "Save" (Word trees cannot be saved when coding is spread as is performed in step 8);
8. In the "Spread To" drop down menu, SC custom context > SC the "Number of Words" bubble > enter "30" in the "Number of Words" box > SC "OK"
 - a. Note—Our recommendation for the minimum number of words is 30; more can be added if desired
9. SC "Run Query";
10. In Nvivo (or a separate program), record the keywords utilized in the search for reference and later reporting;
11. SC "Save Results";

12. SC “Select” next to the “location” box > SC “nodes” > SC “OK”;
 - a. Note—Another option would be to save it in the “query results” location. The downside of this approach is that you cannot uncode irrelevant search hits)
13. Enter the name of the search results (e.g., Health Project Search Hits);
14. SC “OK”. A new node is created, opened, and is accessible under “codes” on the left navigation pane in “nodes”;
15. Save;
16. Share the file with each team member performing open coding.

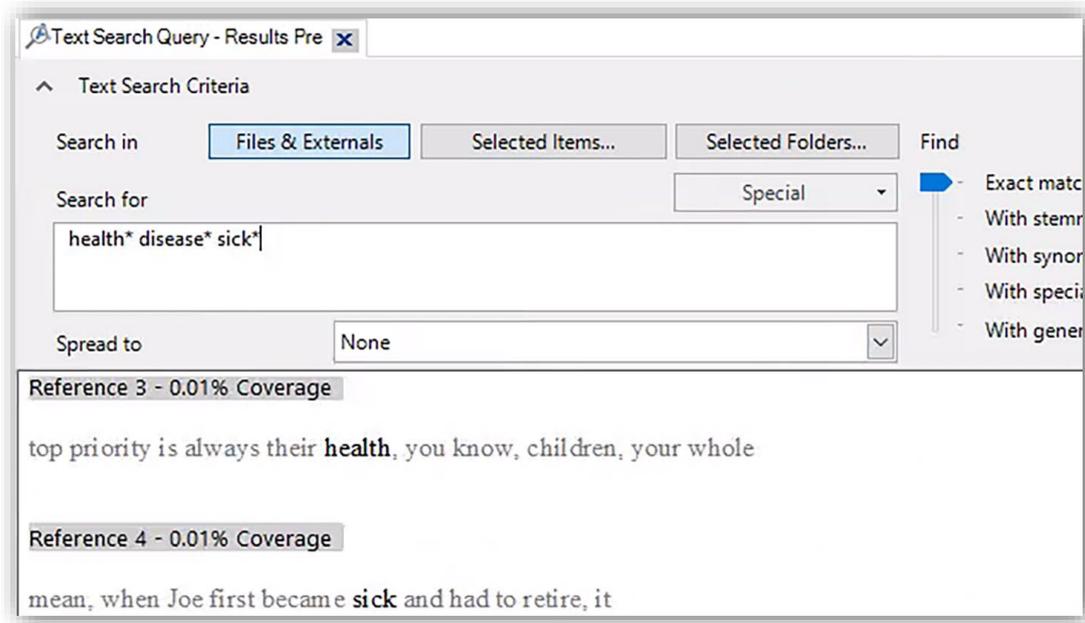


Figure 3. Open coding steps with the entire interview method screenshot (Steps 1–5).

All coders then perform the next steps:

17. SC inside the window with the text of the transcript so that “document tools” appears on the ribbon. Under “document tools”, (1) SC coding stripes > SC “all coding” and (2) SC “highlight” > SC “all coding”;
18. SC “nodes” in the left navigation menu. The node navigation pane will appear (which should be empty at this point);
19. Read through a section of text (about a paragraph) and determine a “code” that fits it adequately. Ensure you only look for new codes (do not code at the keyword node);
20. Once a code is decided upon, highlight the text and SC-HD at the start of the code > drag to the end of the code > R-SC;
21. SC-HD the highlighted text > drag mouse to empty space in the nodes pane > R-SC;
22. In the new window that popped up type in the name of the code decided in step 4;
23. SC “okay”. A new node appears in the nodes pane with the name of the code;
24. Continue the process until the end of node:
 - a. If the same code is present more than once (as it likely will be), repeat steps 19–20 but at step 20 drag the mouse to the re-occurring node and then R-SC.
 - b. If a new code becomes present (as it likely will), repeat steps 19–23.
25. Save.

3.4. A Tip for Speeding Up Open Coding . . . and a Caution

It is not necessary to open code all data. Theoretical saturation, where little to no new information about key themes is gleaned from further coding, will be reached at a certain point during open coding. From a methodological standpoint, one reason limiting the

data utilized for open coding is acceptable is that all data *will eventually be analyzed multiple times*—systematic coding (described later) ensures this. Researchers recommend coding anywhere between 9 and 24 interviews [15–17] to achieve theoretical saturation. We raise a caution in this limiting process: do not overly cut time at the expense of quality. Validity of the key themes depends on the quality of open coding—leaning towards doing more open coding rather than less is better. We suggest considering only limiting open coding if there are more than 20 interviews, or approximately 200,000 words. Recommendations about limiting open coding data are presented in Table 1.

Table 1. Decisions around limiting the amount of data open coded.

Step	Question and Decision
Entire interview method	–
1a	Are there more than 20 interviews? No: Open code all interviews Yes: Proceed to Step 2a
2a	Does the word count of all interviews exceed or equal 200,000? ^a No: Open code all interviews Yes: Proceed to Step 3a
3a	Do you wish to limit data coded in order to speed up open coding? No: Open Code all interviews Yes: Cautiously decide which of the 20 interviews will be open coded. Record decision rationale for later reporting
Deep dive or keyword methods	–
1b	Does the word count of all text in the deep dive node or keyword node exceed or equal 200,000? ^a No: Open code the entire deep dive or keyword node Yes: Proceed to Step 2b
2b	Do you wish to limit data coded in order to speed up open coding? No: Open Code all interviews Yes: Cautiously decide how to limit coding, aiming to open code at least 200,000 words or more. Record decision rationale for later reporting

Note. ^a The 200,000 word benchmark is derived from the following logic: (1) the average number of syllables spoken per minute in American English is 250 [18]; (2) the average number of syllables per word in adults is 1.5 [19]; (3) therefore, the average number of words per minute is 250 over 1.5 which equals 166.67; (4) therefore, an hour-long interview would contain 10,000 words (166.67 words per minute times 60 min); (5) therefore 20 interviews would contain 200,000 words (10,000 words times 20).

4. Consolidating Open Codes in Preparation for Systematic Coding

Open coding is followed by consolidating the various codes team members have identified. This is best accomplished through an approximately hour-long research team meeting (consolidating open coding may require more than an hour depending on the amount of data, number of team members or the complexity of the project). It is important that *all* the coding team members attend as well as the coding team supervisor or faculty member—even if that supervisor/faculty did not perform any open coding. Diverse perspectives (the main contribution provided by members of the coding team) and a vision of the project (provided primarily by the supervisor/faculty) should both be present as codes are deliberated upon and finalized. During this meeting, each member of the coding team takes turns sharing (a) the names of the codes they identified, (b) a brief definition/description of the codes, and (c) how many instances of that code were present in the work they performed (these counts can be found through looking at the nodes pane in Nvivo which lists the number of times a node has been coded and (where applicable) the number of interviews that contain that node). A scribe records the codes (and associated counts) each team member shares on a whiteboard or projector.

After each team member has shared, the goal of the meeting is to make a preliminary decision about which codes are the key or core themes (for more on core themes, see [4] *masked for review* p. 502, first paragraph). There are two effective strategies for achieving

consensus around the core themes. The first is *combination*, or *the merging of two closely related open codes under a single code*. It may not be necessary to employ this strategy, particularly if there are few codes (less than 6–7 total codes) presented by researchers and the codes do not significantly overlap conceptually. However, in our experience many codes (10+ total codes) are typically presented by team members and usually do overlap—making the use of this strategy necessary. This strategy is employed when team members' codes significantly overlap conceptually.

The second strategy is *elimination*, or *removing “pretenders (i.e., concepts that may seem important at first, but are not supported across interviews) [emphasis added]”* so as to focus on *“contenders (i.e., salient and frequent concepts that do emerge across interviews) [emphasis added]”* [4] (p. 496). Elimination is used when team members' codes are infrequent (e.g., found less than five times) or irrelevant (outside the scope of the project). Irrelevancy only applies as a justifying reason in the deep dive or keyword methods but not in the entire interview method. This is because in the entire interview method, researchers are performing a clean exploration of the data whereas the other two methods have a preconceived focus.

At this point some researchers may be wondering why elimination is even necessary. After all, in general, in research it is good practice to not discard data. The first reason that it may be necessary is simply practicality. Experience has shown that it is difficult to perform the next steps of the coding process (systematic coding and partner review) when many codes are present. The second reason is saliency. Marks' method [4], as well as research generally, is reductionistic in nature and focuses on observing and presenting the loudest findings. Indeed, it is focused looking at on the “core” rather than “surface” themes. The last reason is digestibility. Readers and stake holders may find it difficult to digest an overabundance of codes.

Our experience is that “consensus is typically achieved around a small number of themes (typically 4–6)” [4] (p. 501). We advise not exceeding seven core themes. We also advise using combination over elimination where feasible. Elimination should only be performed where combination is not allowable in order to save as much information as possible. Combination may not be possible when codes are reasonably distinct from one another. When this condition is present, elimination can be turned to as a tool.

Once consensus around the core themes has been reached, clear definitions of each of these themes are recorded in a document accessible to all team members. Based on personal experience, we note that the following steps (systematic coding and partner review) proceed more smoothly and quickly if coders referenced and had on hand the list of definitions created when coding and reviewing. Accordingly, the importance of the list of code definitions cannot be overstated. The next phase of coding can then begin once this document is created.

5. Dividing Team Members into Pairs When Coding Teams Are Larger than Two

To speed up the project, the work of systematic coding and partner review (both explained in the next sections) of interviews can be divided up amongst team members when coding teams have more than two individuals. For instance, if a team consists of four researchers and there are fifty interviews (or nodes for the deep dive or keyword methods), then two team members may be assigned to work as a pair on interviews 1–25 and the other pair assigned interviews 26–50. Of note, dividing up team members' work is possible across all three methods, as text and nodes are always displayed in Nvivo by interview (the number of interviews that are part of an overarching deep dive or keyword node can be seen by SC the overarching node > and then SC “summary” on the right navigation pane). We note that, in trio work, each team member has a chance to review all the interviews assigned to their trio. This is preferable to having one team member review the same portion of interviews twice as it prevents bias due to comparing the coding of two team members to one another.

The division of work differs among even- and odd-numbered teams. In even-numbered teams, each team member works with a partner on a portion of interviews (for systematic coding and partner review work; again, explained next). However, a three-member team works together in a trio, and an odd-numbered greater than three (e.g., five, seven) works together in pairs except for one trio. For instance, if a trio were assigned 30 interviews to code and review, each trio member could systematically code all 30 interviews. Then, review work could be divided as follows:

- Researcher one could review interviews 1–15 of researcher two and interviews 16–30 of researcher three.
- Researcher two could review interviews 1–15 of researcher three and interviews 16–30 of researcher one
- Researcher three could review interviews 1–15 of researcher one and interviews 16–30 of researcher two

6. Systematic Coding

Researchers perform systematic coding by once again coding the words of participants after the core themes have been finalized and the teams divided if necessary. Systemic coding is focused on confirming the presence of already identified codes. Systematic coding (a) confirms how many times each of the core themes is present, (b) identifies all participant excerpts that correspond to a particular code, and (c) continues to prepare research for dissemination.

6.1. Preparing a File for Systematic Coding

First, a new file needs to be prepared for team members to perform systematic coding on. To achieve this, only one researcher needs to perform the steps outlined in Table 2.

Table 2. How to prepare files for systematic coding.

Step	Description
1	Have the list of codes and definitions (from the consolidation meeting) nearby.
–	<i>Steps to take if a clean copy^a is not available.</i>
2	Create a copy of one of the researchers' Nvivo files being utilized for the project (preferably designating in the file name that the file is a master, un-coded copy). If conducting the deep dive or keyword methods, ensure that the copied file contains the overarching node.
3	SC "nodes" in the left navigation menu. The node navigation pane will appear.
4	If performing the entire interview method: Delete all nodes (SC a node > on the keyboard press delete key; alternatively RC on the node > SC "delete"). If performing the deep dive or keyword methods: Delete all nodes <i>except</i> the overarching node guiding the project (SC a node > on the keyboard press delete key; alternatively, RC on the node > SC "delete").
–	<i>Steps once a clean copy is made.</i>
5	Create new nodes which correspond to the names of the core themes (RC in the nodes navigation pane > SC "new node" > type name of node > SC "okay"; alternatively, use keyboard shortcut ctrl + shift + n > type name of node > SC "okay"). Double check spelling.
6	If performing the entire interview method: Proceed to step 7. If performing the deep dive or keyword methods: Select all nodes except the overarching node and organize them under the overarching node (hold ctrl on the keyboard down > SC all nodes except overarching node > on the last node, SCHD > drag mouse to the overarching node > R-SC. To display the nodes now organized under the overarching node, SC the small box with the "+" sign next to the overarching nodes name).
7	Save the file.
8	Create copies of the file for each team member (preferably designating each team members name in their copy of the file).
9	Distribute copies of the files to team members.

Note. ^a clean copy would include (a) no pre-existing nodes in the entire interview method or (b) only the overarching or keyword node in the deep dive or keyword methods.

6.2. The Process of Systematic Coding

Each individual researcher can begin systematic coding on their designated file once the file is prepared. Before opening their file, the researcher ensures that they have the names of core themes identified as well as the definitions of the core themes nearby. It is important to have the definition of the core themes nearby and refer to them during systematic coding; this will ensure that they are accurate when coding participant excerpts, and makes the next step (reviewing partner work) smoother. A researcher opens their file and performs the following steps listed in Table 3 to systematically code.

Table 3. The process of systematic coding.

Step	Description
1	Depending on the method, either open the first interview ^a or the overarching node or keyword node ^b and optionally turn on highlighting and coding stripes.
2	Read through a section of text (about a paragraph) and determine if it adequately fits the codes (nodes) listed in the node navigation pane.
3	If the section of text adequately fits the codes (nodes) listed in the navigation pane, highlight the text by SC-HD at the start of the code > drag to the end of the code > R-SC. Then, SC-HD the highlighted text > drag mouse to corresponding node > R-SC. If the section of text matches multiple codes, repeat this step accordingly.
4	If the section of text does not adequately fit the code, proceed to the next section of text.
5	Entire interview method: follow steps 2–4 until the end of the interview. Save often. Deep dive or keyword methods: follow steps 2–4 on until the end of the text contained in the deep dive node or keyword node. Save often.
6	Open the next interview to be coded and repeat steps 1–5 until all interviews are coded. –

Note. ^a Steps 1–3 of the open coding steps for the entire interview method (the difference being the node navigation pane will not be empty). ^b Steps 1–3 of the open coding steps for the deep dive method (if performing the keyword method, open the node corresponding to the search hits).

6.3. Common Questions When Conducting Systematic Coding

The question often comes up about how broad the text excerpt should be when coded (Step 3 in Table 3) (e.g., the relevant text, the whole paragraph, or multiple paragraphs). Codes should be broad enough to capture the relevant context without being excessive. Inside the code, we recommend researchers capture (a) the name of the participant, (b) the beginning of the thought the participant shares, and (c) the end of the thought the participant shares. The other question that usually comes up is if researchers may code the same section of text as multiple codes—the answer is yes. This is common as participant excerpts often do correspond with multiple codes (nodes).

7. Reviewing Partner Work

7.1. Individual Review of a Partner's Work

Researchers review one another's work once they have systemically coded their own work. To begin this process, team members exchange the Nvivo files they have completed systematic coding on with their partner/trio. Then, team members, ensuring their own coded file is closed so as to not overly bias their review, open a partner's file in Nvivo and perform the following steps:

1. Review the core themes and the definitions of the core themes;
2. Open (or create and then open) the memo or document used for recording partner review notes (see proceeding section in this manuscript "Effective partner memos and audit trails" for more on memos);
3. Depending on the method, either open the first interview or the overarching node or keyword node and turn on highlighting and coding stripes;
4. Beginning at the first interview or the beginning of the overarching or keyword node, read through the paragraph/section and decide if you agree or disagree with the decision made by a partner to either assign or not assign this paragraph/section of text to a particular node. Evaluate partner decisions based if one believes a partner:

- a. Correctly coded something that should be coded (their code is correct).
 - b. Incorrectly coded something that should not be coded (their code needs to be removed).
 - c. Correctly did not code something that should not have been coded (their decision to not code was correct).
 - d. Incorrectly did not code something that should have been coded (a code they missed should be added).
5. If you agree, note it on your partner review memo. If you disagree, similarly note it, and additionally record a brief sentence indicating why you disagree;
 6. Repeat Steps 4–5 until all text is reviewed, saving often.

7.2. Meeting Together to Discuss Partner Work

After partners have reviewed the work of their partner on their own, partners are ready to meet and share agreements, resolve disagreements, and calculate the inter-rater-reliability of their partnership. We explain how partners go through the review process as follows (and for illustration, in Table 4 we provide an example of what partners may say and do when they are meeting together to discuss partner work):

1. Both partners sit side by side with their own individual computer in front of them and (a) open Nvivo, (b) open their partner's coded file and the relevant nodes or interviews depending on the method, and (c) open the memo or document used for recording partner review notes;
 - a. Note—even in a trio, this process is conducted with *a* partner—so that only two members of the trio perform this process at a time together
2. The reviewer, looking at their reviewee's coded file open in Nvivo and their own memo or review document, and beginning at the first interview, shares the first instance where they disagreed or thought a code was missed by the reviewee. If there are no disagreements, move directly to step 10;
 - a. Note—the reviewer is the partner providing feedback and suggestion and the reviewee is the partner the reviewer is evaluating
 - b. Note—all methods, including the deep dive and keyword methods, are grouped by interviews when displayed in Nvivo. Accordingly, we suggest going interview by interview during a partner review session
3. The reviewer waits for the reviewee to identify the spot the reviewer is referring to;
4. The reviewer shares (a) what they disagreed with, or thought should be added, (b) why they disagreed or thought a code should be added and (c) what action they recommend taking (with the possible actions being removal of a code or addition of a code that was missed);
5. The reviewee responds, sharing their rationale as to why they did (in the case of a removal disagreement) or did not (in the case of a code addition disagreement) code a section as they did, potentially rebutting the suggestion of the reviewer;
6. Partners continue to discuss the disagreement until agreeing, accepting, or conceding occurs. Agreeing, accepting, and conceding are defined below:
 - a. *Agreeing*: The act of the reviewee wholeheartedly and enthusiastically viewing a suggested change by the reviewer as correct.
 - b. *Accepting*: The act of the reviewee viewing a suggested change by the reviewer as incorrect but reservedly and reluctantly accepting the suggested change.
 - c. *Conceding*: The act of the reviewer viewing the rebuttal from the reviewee about their suggested change to the reviewee as incorrect, but reservedly and reluctantly dismissing the change they (the reviewer) suggested.
7. Partners tally the results of the disagreement for later counting, with agreeing counting towards *agreements* and accepting and conceding counting towards *unresolvable disagreements*;

8. Partners make the necessary change to the reviewee’s file, either removing or adding code(s). To remove, RC the coding stripe corresponding to the relevant node > SC “uncode”. To add a code, SC-HD at the start of the code > drag to the end of the code > R-SC. Then, SC-HD the highlighted text > drag mouse to corresponding node > R-SC;
9. Keeping the same roles as reviewer or reviewee respectively, repeat steps 2–8, for the rest of the disagreements in the first interview;
10. Switch roles as reviewer and reviewee and repeat steps 2–9 for the text in the first interview;
11. Count up the number of agreements and unresolvable disagreements. Include in the count of agreements the number of times the reviewer agreed with the reviewee’s codes, but which were not already discussed (because there was no disagreement);
12. Record on the inter-rater reliability spreadsheet the count of agreements and unresolvable disagreements;
13. Save both partners Nvivo files as well as the inter-rater reliability spreadsheet;
14. Repeat steps 2–13 for the rest of the interviews where partners reviewed one another.

Table 4. Meeting and consolidating partner reviews—example conversation.

Step	Example Conversation					
1	–					
2	Reviewer: “My first disagreement is at line 55 where the husband said, ‘I had to take a little more than a week off my job due to the illness. It was disheartening not just for me, but for my whole family’. Do you see that spot?”.					
3	Reviewee: “Okay, I see it”.					
	Example of Code Removal Disagreement			Example of Code Addition Disagreement		
4	Reviewer: “From line 55 to line 56 you coded this section as ‘debilitating illness.’ However, I do not feel it matches up with that code. I think we should remove it”.			Reviewer: “From line 55 to line 70 I think we should add the code ‘depression’. The husband references how it was ‘disheartening,’ and is conveying how it was depressing to have to be sick and take off work. Subsequently, we should code that section as ‘depression’”.		
5	Reviewee: “I see. At line 55 he mentions how he had to take work off for about a week. To me, that matches our definition of ‘debilitating illness’.			Reviewee: “I see. I did not initially feel that spot fit the ‘depression’ code—making it fit seemed like too big of a stretch”.		
	Ex. of Agreeing	Ex. of Accepting	Ex. of Conceding	Ex. of Agreeing	Ex. of Accepting	Ex. of Conceding
6	Reviewee: “But, now that I think about it more, I do not think it fits like you suggest. Let’s remove it”.	Reviewee: “But, I see your point though. I still think that code belongs there, but we can remove it as you suggest”.	Reviewee: “Based on that point I shared, I think we should keep it”. Reviewer: “I’m still not on board, but I see what you mean and we don’t need to remove it”.	Reviewee: “But, now that I think about it more, I do think it fits like you suggest. Let’s add it”.	Reviewee: “But, I see your point though. I still think that code doesn’t belong, but we can add it as you suggest”.	Reviewee: “Based on that point I shared, I think we should not add it”. Reviewer: “I’m still not on board, but I see what you mean and we don’t need to add it”.
7	Agree/Disagree Count “1 Agreement”	Agree/Disagree Count “1 Unresolvable Disagreement”	Agree/Disagree Count “1 Unresolvable Disagreement”	Agree/Disagree Count “1 Agreement”	Agree/Disagree Count “1 Unresolvable Disagreement”	Agree/Disagree Count “1 Unresolvable Disagreement”
8	Action Taken	Action Taken	Action Taken	Action Taken	Action Taken	Action Taken
	Uncode ‘debilitating illness’ at lines 55–56 in the Reviewee’s file	Uncode ‘debilitating illness’ from at 55–56 in the Reviewee’s file	No changes necessary	Code ‘depression’ at lines 55–56 in the Reviewee’s file	Code ‘depression’ at lines 55–56 in the Reviewee’s file	No changes necessary

Table 4. Cont.

Step	Example Conversation
9	–
10	Reviewer: “Ok, now it’s your turn to be the reviewer and share your review of my coding”. Reviewee: “Let’s see . . . so we had 1 agreement where you suggested I remove something and I agreed, 1 unresolvable disagreement where you suggested we add something and I only accepted, and 1 unresolvable disagreement where you suggested we add something and you conceded”.
11	Reviewer: “That’s what I counted too. You also had 3 codes that I agreed with that we did not need to discuss”. Reviewee: “So . . . in total that makes for 4 agreements and 2 unresolvable disagreements”. Reviewer: “That’s what I count too”.
12	Reviewee: “Alright, on our spreadsheet, for the interview ‘Hernández’ with Alexa as the reviewee and Joseph as the reviewer, we’ll put 4 under agreements in cell ‘e7’ and 2 under unresolvable disagreements in cell ‘f7’.
13	–
14	–

Note. Steps correspond to the steps in the section “Meeting together to discuss partner work”.

Keeping count of agreements and unresolvable disagreements when partners are sharing their reviews of each other’s work is the basis of inter-rater reliability. Inter-rater reliability (IRR) is equal to the sum of agreements and unresolvable disagreements divided by the number of agreements. One effective way to track and calculate IRR is utilizing computer spreadsheet software and a spreadsheet, such as the one provided in the Supplementary Material associated with this article. We note that organizing the rows of the spreadsheet by interviews works for all three methods, not just the entire interview method. This is because when a node is opened in Nvivo (for instance in the deep dive or keyword methods) the text is grouped by interviews. Accordingly, we suggest organizing the spreadsheet by interviews.

7.3. Effective Partner Memos and Audit Trails

In order to stay organized when partner reviewing, it is important to keep an audit trail of what codes reviewers believe should added, removed, or unchanged, and what happens as a result of meeting with a partner. One way to keep an effective audit trail is to take advantage of the memo function in NVivo. To create a new memo, SC “Notes” > RC “Memos” > SC “New Memo” > Type the desired memo name > SC “OK” (the best location for the review memo of a reviewer is in the file the reviewee coded in Nvivo, rather than the file the reviewer coded). Researchers may follow the following memo format: (a) Provide a header with the interview name bolded and underlined; (b) when noting disagreements, record the line number, what is disagreed with, why the researcher disagrees, and what they think should be changed (i.e., added or removed); (c) when meeting with a partner, record in a color besides black (e.g., red) what was changed after discussing a disagreement and how it was changed. If nothing was changed, simply record “unchanged”. We provide an example of a partner review memo in Figure 4.

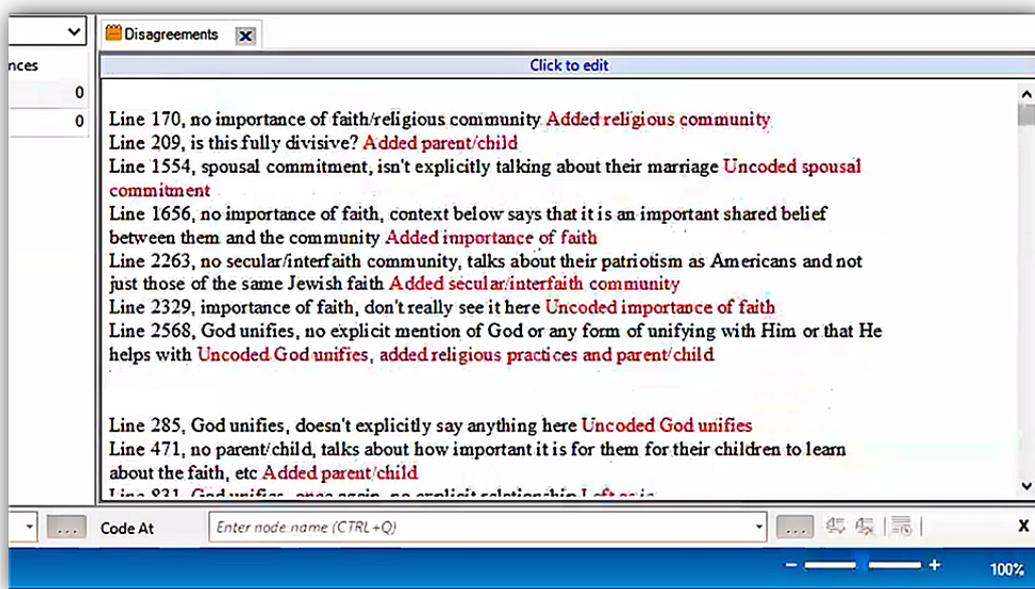


Figure 4. Example of a partner review memo with red text indicating where partner's had unresolvable disagreements.

8. Preparing Work for Publication

8.1. Merging Partner Files

Work can be prepared for publication and dissemination once systemic coding and partner review is complete. While getting ready to share work, it is useful to have a single, merged file that contains the coding of all researchers involved in the project. Prior to merging researchers' files, all nodes should have the exact same names in all files or merging will not work properly. A single researcher follows the steps outlined below, for each set of coding pairs files, to merge coded files:

1. Open one member of the coding pairs files in Nvivo;
2. Save a copy, ensuring the name of the new file designates a merge and which partnership (e.g., healthcoding-Alexa-Joseph-merge.nvp);
3. Opening and using the new, copied file, SC "import" on the ribbon > SC "project" > browse to the other partner's file location > select the file to be imported > SC "open";
4. Ensuring that the defaults, "all (including content)" and "merge into existing item" are selected, SC "import":
 - a. At this point, the nodes are merged; however, for the purposes of Marks' [4] method, the reference counts are not accurate—Nvivo is double counting overlap between coders (where they both coded the same section of text) as multiple nodes when in reality they only should count as one. The following steps remedy this.
5. Create a new node by RC in a blank area of the nodes window (or use the shortcut CTRL + Shift + N) and name it (e.g., "correct count");
6. RC one of the nodes > SC "create as" > SC "create as code";
7. In the "select location" window that appears, SC the plus sign under nodes > SC the recently created code (e.g., SC "correct count") > SC "ok";
8. In the "new node" window that appears, provide a name for the node (the name of the old node will suffice) > SC "ok":
 - a. Note how the reference count for the new node is less as compared to the old node it was created from. This is correct and will happen in nearly every case.
9. Repeat this step for each node (including the overarching deep dive or keyword node). If necessary (i.e., using the deep dive or keyword methods), organize nodes under the overarching deep dive or keyword node (hold ctrl on the keyboard down > SC

- all nodes except overarching node > on the last node, SCHD > drag mouse to the overarching node > R-SC. To display the nodes now organized under the overarching node, SC the small box with the “+” sign next to the overarching nodes name);
10. Delete the nodes with the incorrect counts (e.g., not part of the “correct count node”) by selecting the appropriate nodes > press the delete key (alternatively RC > SC “delete”) > SC “yes” in the delete confirmation window that appears;
 11. Save the file;
 12. Repeat steps 1–11 for each coding pair’s files.

8.2. Our Perspective on Numbers and How to Prepare a Numerical Content Analysis

One practice we have found beneficial is sharing the number of times each core theme was found, known as a numerical content analysis (NCA). Some researchers argue that reporting numbers misrepresents qualitative findings, as the essence of qualitative research is diluted. Others approach qualitative research from a purely content analysis paradigm that almost parallels quantitative research. We suggest that qualitative research can benefit by using numeric content analysis, which adds methodological rigor. Quantitative reports, such as the NCA, should be seen as complementary and supplementary rather than present inferences of salience based on frequency. For an example of an NCA table, see [14], Table 2.

Preparing an NCA can be achieved by simply opening the final, merged coded file in Nvivo and recording (on a text document, spreadsheet, etc.) the counts of the nodes. Nvivo also offers the ability to export these counts to a spreadsheet; a function we have found helpful but not a requisite step. To perform this optional step, open the desired file and navigate to “nodes” under “codes” in the navigation pane, and perform the following steps:

1. Select all nodes desire for export (so that all nodes a highlighted);
2. RC one of the highlighted nodes > SC “export” > SC “export list”;
3. In the save as dialog box which appears, navigate to the desired folder to save the file in (if necessary), change the file name (if desired), and change the file type if desired (e.g., .xlsx, .docx, .rtf, .txt, etc.);
4. SC “save”. Use the information in the saved file to create an NCA table.

8.3. Preparing and Trimming Gems

In the final phase, findings are presented in the form of exemplary participant quotations. When presenting findings, it would be impractical to present raw data to represent participant quotations due to typical word count or page restrictions. Therefore, trimming is usually necessary. Suggestions for preparing quotes include: (1) trim quotations using ellipses (. . .) (e.g., tangential information); (2) remove non-verbal phrases and pauses, when appropriate; (3) in joint interviews, wherever possible, include within participant discussions and dialogues; (4) summarize background information and present participants’ takeaway statements before or after quotations; (5) group participant quotations that are similar and string phrases together. Selected quotes can vary by length depending on depth of context and can be presented as both block and in-text quotations.

We note that typically, in qualitative reports, the balance between how often the voice of researchers is used and how often the voices of participants are used has been tilted towards researchers. This is unfortunate—much valuable time has been spent on interviewing, collecting, transcribing, and analyzing data. We advocate more participant-voiced findings sections so as to better convey the lived experiences of participants.

9. Conclusions

We have provided step-by-step guidance for how to use Marks [4] method with Nvivo. The procedures we have outlined provide a structure for efficiently and effectively performing qualitative inquiry that is replicable and rigorous. A key function of qualitative research—and all research generally—is to understand the lived experiences of individuals. Understanding individuals’ lived experiences is valuable because, as James [20] stated, “*that which produces*

effects within another reality must be termed a reality itself" (pp. 339–400, emphasis added) and, as Thomas [21] indicated, if a situation is perceived as real, it is real in its consequences. We hope our methodological guidance contributes towards a better understanding of individuals' realities and the associated consequences of their perceived situations.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/psych4020013/s1>.

Author Contributions: Conceptualization, D.B.A., J.M.C., E.R.K., L.D.M. and J.J.H.; Methodology, D.B.A., J.M.C., E.R.K., L.D.M. and J.J.H.; Software, D.B.A., J.M.C., E.R.K. and J.J.H.; Resources, L.D.M.; Data Curation, D.B.A., J.M.C., E.R.K., L.D.M. and J.J.H.; Writing—Original Draft, D.B.A.; Writing—Review & Editing, J.M.C., E.R.K., L.D.M. and J.J.H.; Supervision, D.B.A. and L.D.M.; Project Administration, D.B.A. and L.D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Data examples used in the manuscript were gathered following procedures approved by institutional review boards at Brigham Young University and Louisiana State University.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1. Tips for Transcribing

We offer several tips for transcribing: (1) be consistent in labeling, marking the timestamp whenever you stop if taking a break, and always remember to save your progress; (2) use abbreviations for whoever is speaking, and, to keep with anonymity, use letters to represent the speaker, such as "H" for a husband, "W" for a wife, and "I" for an interviewer; (3) put [inaudible] in the transcription to mark parts of the interview where the transcriber cannot understand what is being said; (4) when possible, have a transcribing partner review initial transcriptions, including simultaneously listening to the audio and going over the written transcription; (5) omit filler words, such as "uh", "like", "um" unless it is important to what the individual is saying—these distract from the meaning of what will eventually be coded; (6) include important actions or descriptions of how things are said to give future coders context (e.g., if someone laughs when/after saying something, be sure to insert "laugh", or when someone is saying something sarcastic, insert "sarcastic").

Appendix A.2. Outsourcing Benefits and Cautions

Outsourcing transcription can save hours of time. A key argument for outsourcing is that transcription is a draining process and, therefore, outsourcing is an invaluable resource [22]. Arguments against outsourcing include distancing of the researcher from the data [23] and not capturing the tone and nonverbal aspects of the interview see [24]. Outsourcing can be valuable, but the decision to outsource should be made cautiously.

References

1. Marks, L.D.; Kelley, H.H.; Galbraith, Q. Explosion or much ado about little? A quantitative examination of qualitative publications from 1995–2017. *Qual. Res. Psychol.* **2021**, *1*, 1–19. [[CrossRef](#)]
2. Gergen, K.J.; Josselson, R.; Freeman, M. The promises of qualitative inquiry. *Am. Psychol.* **2015**, *70*, 1–9. [[CrossRef](#)] [[PubMed](#)]
3. Morison, T.; Gibson, A.F.; Wigginton, B.; Crabb, S. Online Research Methods in Psychology: Methodological Opportunities for Critical Qualitative Research. *Qual. Res. Psychol.* **2015**, *12*, 223–232. [[CrossRef](#)]
4. Marks, L.D. A Pragmatic, Step-by-Step Guide for Qualitative Methods: Capturing the Disaster and Long-Term Recovery Stories of Katrina and Rita. *Curr. Psychol.* **2015**, *34*, 494–505. [[CrossRef](#)]

5. Levitt, H.M.; Bamberg, M.; Creswell, J.W.; Frost, D.M.; Josselson, R.; Suárez-Orozco, C. Journal article reporting standards for qualitative primary, qualitative meta-analytic, and mixed methods research in psychology: The APA Publications and Communications Board task force report. *Am. Psychol.* **2018**, *73*, 26–46. [[CrossRef](#)] [[PubMed](#)]
6. Mitchell, T.; Friesen, M.; Friesen, D.; Rose, R. Learning Against the Grain: Reflections on the Challenges and Revelations of Studying Qualitative Research Methods in an Undergraduate Psychology Course. *Qual. Res. Psychol.* **2007**, *4*, 227–240. [[CrossRef](#)]
7. Ponterotto, J.G. Integrating qualitative research requirements into professional psychology training programs in North America: Rationale and curriculum model. *Qual. Res. Psychol.* **2005**, *2*, 97–116. [[CrossRef](#)]
8. Parker, I. Criteria for qualitative research in psychology. *Qual. Res. Psychol.* **2004**, *1*, 95–106. [[CrossRef](#)]
9. Fielding, N.G.; Lee, R.M. *Computer Analysis and Qualitative Research*; SAGE Publications: Thousand Oaks, CA, USA, 1999.
10. Zamawe, F.C. The Implication of Using NVivo Software in Qualitative Data Analysis: Evidence-Based Reflections. *Malawi Med. J.* **2015**, *27*, 13–15. [[CrossRef](#)]
11. QSR International. NVivo 10 for Windows: Getting Started. 2014. Available online: <http://download.qsrinternational.com/Document/NVivo10/NVivo10-Getting-Started-Guide.pdf> (accessed on 17 March 2022).
12. LeBaron, A.B.; Hill, E.J.; Rosa, C.M.; Marks, L.D. Whats and Hows of Family Financial Socialization: Retrospective Reports of Emerging Adults, Parents, and Grandparents. *Fam. Relat.* **2018**, *67*, 497–509. [[CrossRef](#)]
13. Jorgensen, B.L.; Allsop, D.B.; Runyan, S.D.; Wheeler, B.E.; Evans, D.A.; Marks, L.D. Forming Financial Vision: How Parents Prepare Young Adults for Financial Success. *J. Fam. Econ. Issues* **2019**, *40*, 553–563. [[CrossRef](#)]
14. Allsop, D.B.; Leavitt, C.E.; Clarke, R.W.; Driggs, S.M.; Gurr, J.B.; Marks, L.D.; Dollahite, D.C. Perspectives from Highly Religious Families on Boundaries and Rules About Sex. *J. Relig. Health* **2021**, *60*, 1576–1599. [[CrossRef](#)] [[PubMed](#)]
15. Ando, H.; Cousins, R.; Young, C. Achieving Saturation in Thematic Analysis: Development and Refinement of a Codebook. *Compr. Psychol.* **2014**, *3*, 03–CP. [[CrossRef](#)]
16. Hennink, M.M.; Kaiser, B.N.; Marconi, V.C. Code Saturation Versus Meaning Saturation. *Qual. Health Res.* **2017**, *27*, 591–608. [[CrossRef](#)] [[PubMed](#)]
17. Saunders, B.; Sim, J.; Kingstone, T.; Baker, S.; Waterfield, J.; Bartlam, B.; Burroughs, H.; Jinks, C. Saturation in qualitative research: Exploring its conceptualization and operationalization. *Qual. Quant.* **2018**, *52*, 1893–1907. [[CrossRef](#)] [[PubMed](#)]
18. Robb, M.P.; Maclagan, M.A.; Chen, Y. Speaking rates of American and New Zealand varieties of English. *Clin. Linguist. Phon.* **2004**, *18*, 1–15. [[CrossRef](#)] [[PubMed](#)]
19. Johnson, W.; Darley, F.L.; Spriesterbach, D.C. *Diagnostic Methods in Speech Pathology*; Harper & Row: Oxford, UK, 1963; p. xv, 347.
20. James, W. *The Varieties of Religious Experience: A Study in Human Nature*; Longmans, Green and Co.: New York, NY, USA, 1902.
21. Thomas, W.I. *The Unadjusted Girl: With Cases and Standpoint for Behavior Analysis*; Little, Brown, and Co.: Boston, MA, USA, 1923.
22. Matheson, J. The Voice Transcription Technique: Use of Voice Recognition Software to Transcribe Digital Interview Data in Qualitative Research. *Qual. Rep.* **2015**, *12*, 547–560. [[CrossRef](#)]
23. Tilley, S.A. “Challenging” Research Practices: Turning a Critical Lens on the Work of Transcription. *Qual. Inq.* **2003**, *9*, 750–773. [[CrossRef](#)]
24. Bird, C.M. How I Stopped Dreading and Learned to Love Transcription. *Qual. Inq.* **2005**, *11*, 226–248. [[CrossRef](#)]