2020

Coding and Analysis: Deciding on Software Needs

Katherine Gregory
CUNY New York City College of Technology

Chapter 19

Coding and Analysis:

Deciding on Software Needs

Katherine Gregory

INTRODUCTION

Qualitative data analysis software (QDAS) tools have been developed in large part to assist researchers with data management, coding, and analysis of their data sets.[1] Once the data collection phase is completed—whether it is in the form of interviews, observational field notes, visual and/or textual content, audio and video formats, or a mix of source materials—the researcher must decide how to approach their data set for coding and analysis. Coding refers to assigning text-based themes to the source materials and then discovering patterns that emerge from the data set. The themes inform findings and are the crux of a researcher's final analysis of their data. When the data set is large or involves more than one research method, a software product may be useful for facilitating the sorting and labeling of those excerpts or images. When the data set or sample size is small or uncomplicated, it may not be necessary to utilize a software application and the data can simply be coded by hand.

As a researcher and scholar for the past twenty years, I have, in various capacities, provided methodological and instructional support to many faculty, researchers, and students regarding their research methods and qualitative data analysis software needs. Over a two-and-a-half-year period, I was the qualitative data analysis and survey design lead for a data services department at a Research I university library. Reasons why faculty, graduate students, and researchers sought my advice varied. Sometimes researchers were experimenting with a new qualitative method; other times they were learning to use these QDAS tools after years of manual coding. In my role, I provided one-on-one instructional support and I taught stand-alone lectures in graduate research methods courses. In these situations, I often found myself suggesting best practices regarding steps researchers needed to take with their qualitative research while still in the design phase. I advised them to consider their potential needs for coding and analysis while drafting their research proposals. I also stressed that QDAS—NVivo, ATLAS.ti, Dedoose—required the manual entry of codes, as the software would not generate these codes for them. Therefore, in addition to providing technical support during research consultations, I also learned how to manage user expectations concerning the limitations of the software. This chapter outlines several key considerations for researchers when choosing a QDAS.

KEY CONSIDERATIONS

Coding Needs

The scope of the project should not be underestimated when selecting a QDAS or opting to hand code. The scope of a project can include but is not limited to the project timeline, the complexity of deliverables, and the number of researchers designated to code materials. Novice researchers, or those new to using QDAS, will want to build in time to learn how to use the software.

Projects that include fewer than ten interviews or observations can be coded in MS Word or Excel, which could save novice researchers the time it would take to learn a complex QDAS. Coding in MS Word involves working from any Word or Excel document and highlighting excerpts using the color-coded highlighter function, different font types, or simply highlighting a section. Themes can be written in the margins using the comment function to signify a code. It is also possible to perform this same process after printing out a data set by using colored ink pens or highlighters to identify excerpts or content and writing in the margin comments to identify a theme in your research.

Careful consideration of the size and complexity of the data sources or primary documents is important. For example, during an initial consultation, I inquire from researchers what kinds of source materials they plan to incorporate in their coding and analysis phase. The answer to this question can determine what product matches their needs. On some occasions, researchers eagerly collect more data because their QDAS has the capacity; however, in some instances, more data may not actually enrich findings or guide researchers to answer a sought-after research question.[2] This means that during the research design phase, researchers must carefully consider what each data source will bring to the project and whether that source contributes to answering a research question, adds dimension to understanding a phenomenon, or simply takes a research project in an unnecessary direction. Thus, the types of data collected should be part of the researcher's overarching design before the coding and analysis stage.

For simple transcription files or open-ended data from online surveys, source materials can effortlessly be uploaded into Atlas.ti or Dedoose without complications. In the event that there are many source materials of different formats or with multiple segregated coding blocks, researchers may want to consider using NVivo. Researchers with large-scale projects—particularly with many different types of data formats, source materials, and coding systems—could benefit from using NVivo for organizing, coding, and analyzing data. The organizational tools provided by NVivo can facilitate structure when building directories for research materials. Over time, I saw a need for supporting NVivo because researchers came in needing assistance with larger, intricate projects with source materials in myriad formats, including audio-video materials, social media data, spreadsheets, images, and more.

It is relevant to mention that all coding terminology rest entirely with the researcher's interpretation. The researcher must create their own codes, sometimes called nodes or themes depending upon the software, either on the fly, with a codebook, or after reviewing their source materials. There have been times when researchers arrived at a session with their data set already organized around social demographic information collected about their participants and clustered together along with responses from an open-ended survey, interview excerpts, or even in the following case, a close-ended question on a given topic. In the latter instance, the researcher wanted the software to "answer a question" beyond the depth of his inquiry and expected the software to provide an interpretation that could never have emerged from the original data. Based on how the researcher organized his data set, he could tell me how many female-identified participants expressed a certain opinion, but this correlation was based on what results he had extracted from his data set and then had copied to a separate document without the corresponding transcriptions. In effect, the researcher had brought with him what looked like the beginnings of a QDAS deliverable after inputting sorting criteria. Had the researcher uploaded his transcripts into Atlas.ti or NVivo, he could have generated a similar document using the "query" tool; however, without his actual transcripts, he was unable to elicit meaning beyond what he had identified as support for or against a particular educational mandate. In the end, the "results" were only as good as his data set.

Coding can be a very personal process for any researcher. Sometimes novice researchers assume more codes mean a more complex analysis, but this is not always the case. As a researcher reads their source materials for the first time, new themes that were never conceptualized before could emerge from the data; however, there may still be a need for some organization of primary codes and sub-codes with the creation of a codebook. I am still haunted by my discovery of a team's coding strategy during a routine consultation. They had been using outdated software stored remotely on a university shared drive that could barely carry the weight of what lurked in the project. The researchers involved explained to me that they were each discovering new codes as they proceeded independently to read through the transcripts for the first time. This coding on the fly without a team consensus fostered the production of hundreds of codes assigned to each transcript, thus making the aggregation of thematic patterns challenging. In effect, their method of coding failed to organize around any prevailing themes to produce a hierarchical organization of ideas and therefore brought no coherency linking different types of associations to the order of flattened themes. Despite the risk of crashing the entire operation, the project progressed as each team member continued to devise new themes as they went through their transcripts. From my perspective, this was chaos and would have been difficult to make sense for retrieval purposes or during the analysis stage. Too many codes can make for difficult analysis.

It is also fair to ask how one makes meaning and coherency out of hundreds and hundreds of codes, without giving them some priority for interpretation. With so many themes, how should we organize them? If the taxonomies are similar, they could fall under a single category or "family" of themes. Think of it as a theme of a higher order, like a "meta" theme. That is how a hierarchy of codes, or nodes in the case of NVivo, operates. In effect, what this demonstrates is the need for hierarchical coding schemes and the foresight to design a codebook before the coding process begins.

In this context of hierarchical coding, it is also meaningful to mention that most products on the market assign a different naming convention to their hierarchical coding tools and, for that matter, naming conventions used for designating a theme as a code or a node. Not all features, however, allow for the same breadth of structural depth across products. What will transpire, depending upon the QDAS, is the potential for rich coding structures. Sometimes they are called "families," as available in Atlas.ti, or reflect a multi-generational family structure with a "child," "parent," and "grandparent" coding system, as with NVivo. If I am constructing a very complex coding scheme, drilling downward with "multiple generations" of codes, NVivo would be my first choice. These tools are there to assist the researcher in prioritization, organization, and overall making sense of their findings. For a complex coding scheme, NVivo has sophisticated coding features; however, some projects do not require such elaborate coding, so Atlas.ti or Dedoose would be sufficient.

The scope of a project can also include the number of coders who will be working on a given project. This often translates to the concept of inter-rater reliability. Inter-rater reliability is a method for diminishing bias in coding. It can be performed when multiple researchers code in isolation from each other, but once done, they will compare their coding to determine if there was a consensus assigning a code to an excerpt or phenomenon found in the data set. In the instance where there is a large team of coders—or even a pair of coders, for that matter—post-data collection organization will usually require the crafting of a codebook. A codebook will identify coding definitions and support inter-rater reliability as multiple research group members code identical data sets using the same coding scheme. From this point on, the project administrator can either merge codes from each researcher to ensure all researchers are assigning a code to a particular phenomenon or transcript excerpt or print out their coded work to manually determine a consensus regarding interpretation of excerpts and coding designation. For this example, NVivo, while more complex, would be suited for such a project, as it has some built-in inter-rater reliability functions specifically suited for research teams.

This is not to suggest that coding can only be accomplished with a codebook. It is possible for two coders to "blind code" by designating codes based on their interpretation and without consultation with each other about their shared or consensual understanding of the data set. But this type of exercise is tedious and time-consuming. However, it also could demonstrate divergent interpretations of the data or lead to new discoveries beyond answering research questions.

Visualizations, Graphs, and Other Figures

As the researcher continues to code and begins to find patterns in their data, it is meaningful to ask what inspires insight about their research findings. The labor of coding can live entirely on a computer as a saved project file, but the coding can also be aggregated a number of ways to produce, depending upon needs and learning style and different types of QDAS deliverables showcasing relationships or associations between assigned codes and excerpts. Choosing an output is a highly subjective decision and does not commit the researcher to any single sorting criteria for linking different types of associations. The output, or reports in the case of NVivo, depends entirely on the needs of the researcher and whether these visualizations inform the way they conceptualize their results. Here, visual learners have the luxury of a plethora of outputs and query schemes in NVivo, Atlas.ti, and Dedoose that can be produced by selecting an assortment of software functions from "output" and "query" to "export" and "report" at any stage of the coding process. For some kinesthetic learners, the printout of their work remains as close as they will get to touching the visualization of their data through a selection of code terms and linking them to quoted excerpts, memos, or other codes to produce a network map of codes that make data tangible. This leads to the question of whether or not the visualization speaks to the researcher. Whether these deliverables inform the way the researcher analyzes and derives insights from the data is entirely subjective.

One researcher sought my assistance as she began conducting a visual analysis and needed to upload hundreds of high-resolution digital photographs in Atlas.ti. As the project grew, there was a need to see some semantic representation linking code terms to specific images. This coded network represented ways to link conceptually the different images to textual interpretation created during the coding process. Yet, those images "felt" decontextualized as mapping together myriad ideas for such a large project seemed unwieldy, if not a bit futile. It simply was not possible to experience the coherency of the entire project in the form of a single networked map of textual ideas and images to gain insight from that deliverable. In the end, use of this tool, while demonstrating a visual context, was simply overwhelming based on the scale of the data set.

Determining what kinds of deliverables, reports, or outputs are required of your project lends to meaningful consideration when calculating the scope of the project. QDAS selection, in this case, can hinge on whether the deliverables are simple and straightforward or require an array of visualizations from diverse data sets. After all, not all visualizations aid in the communication or interpretation of results. Visualization output depends on the researcher's needs and how their audience will understand those representations. This imparts researchers with the need to reflect on what types of data they have configured. In this case, Atlas.ti or Dedoose can produce simple outputs that identify lists of codes and aggregate excerpts; for large-scale, complex projects, NVivo can produce visualizations and reports which can enhance understanding and sharing of the results.

Technical Aspects

When I began assisting researchers who had their own copy of Atlas.ti or NVivo software, I was working exclusively within a Windows ecosystem. On my work Mac, to assist researchers, I used a version of Bootcamp for splitting my hard drive. This allowed me to install Atlas.ti on the PC side of my Mac. Times have changed. I waited in anticipation for the Mac versions to roll out for Atlas.ti and NVivo, but first-generation products did not initially provide an identical interface or features that I was accustomed to navigating with my Windows versions. Slowly, interface integration of the two software applications has occurred. The larger problem involved sharing bundles or projects across operating systems. This lack of compatibility required research groups to work in silos. The good news is that, as of this writing, NVivo allows for shareability across platforms. Researchers can copy their projects in one format that is readable to others using different operating systems. In Atlas.ti, files extensions for bundles created in one version can now be uploaded and read across platforms.

I must stress that when working with a team, researchers should consider the collaborative features of each software that meet their needs. This also should require checking the operating systems of each computer being used, software that will work for all research members, and consideration for project naming conventions. Moreover, working on a group project using Google Docs can also be problematic when the cloud system performs an overwrite, putting work-in-progress in jeopardy of being erased. This leads me to suggest that cloud-based products like Dedoose might be the safest option for team projects.

As we move toward more cloud-based products, gone are the days of backup bundles and fears about file extensions as Dedoose provides greater flexibility for multiple users working on versions of the same project at remote locations. Of course, other issues arise with cloud-based products: the availability of the internet, bandwidth, and Wi-Fi access in remote areas. There are also multiple ethical issues to address regarding the protection of sensitive institutional data that may require additional authentication and two-factor encryption to be secured on a cloud-based product or prohibited altogether. Check if your home institution has protocols in place regarding any limitations about compliance and what data should or should not be stored on cloud-based software.[3]

REFLECTIONS

This chapter covers a number of key issues that must be considered before selecting QDAS. Let us not forget that software costs can be an inhibitive factor for individual researchers and institutions with limited resources. I would be remiss not to suggest that many of the products identified in this book chapter can be expensive. No doubt, it is a luxury to work at a university that makes QDAS available to students, staff, and faculty at their campus software labs or virtual computation center. Individual departments or university staff members may purchase or lease these products with an educational discount; however, costs can still be prohibitive and may not be necessary to complete a project. Sometimes a university agreement can be a one-year lease or permanent downloadable software that can be shared on more than one computer. At the time of this writing, NVivo costs roughly $800 with an educational license; a single user license for Atlas.ti runs at about $630; Dedoose pricing for a cloud based-service costs $14.95 per month.[4] These software costs can be prohibitive, but don't let them deter you from coding. You can also explore open source software found online; however, these products tend not to have any technical support, and the burden is on the researcher to learn how to install and use the software.

Researchers must not underestimate the scope of their project and the significant role this plays when deciding on what QDAS to use. Data set size, file format, and scale of source materials must be evaluated beforehand. When considering complexity and size of source materials, some products are better equipped for large data sets and storage of ancillary materials, while other products provide remote or local storage of source materials. Some storage requirements necessitate two-factor authentication or encryption for the privacy protection of sensitive material, like protected health information or student information, so it is imperative to comply with the requirements set by your institution's IRB. Other aspects to bear in mind include the following:

- Cost—QDAS is expensive. If you do not have access to the software, old-fashion manual coding of data set printouts or while working in Word will get the job done.
- Your skill level and your acumen with software. Time must be set aside for learning the software and it is important to build this time into your project timeline. Functionality of the products can be as simple as labeling excerpts to theme terms or as complicated as finding a co-efficient between themes. Either way, learning how to use these tools requires time.
- The scope of your project. Is it a large data set? Does it require compiling complex source materials? If it's simple and small scale, do you need to use these tools to get the job done?

The author would like to thank Sarah DeMott, PhD, for the useful discussions on this subject.

NOTES

1. Judith Davidson, Trena M. Paulus, and Kristi Jackson, "Speculating on the Future of Digital Tools for Qualitative Research," *Qualitative Inquiry* 22, no.7 (2016): 606, doi:10.1177/1077800415622505.
2. Cynthia S. Robins and Karla Eisen, "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, no. 10 (2017): 768, doi:10.1177/1077800417731089.
3. "Dedoose," University of Michigan Safe Computing, Sensitive Data Guide, University of Michigan Safe Computing, University of Michigan, Retrieved October 1, 2018, https://www.safecomputing.umich.edu/dataguide/?q=node/231.
4. NVivo, accessed February 4, 2019, https://www.qsrinternational.com/; "Technical Information," ATLAS.ti, accessed February 4, 2019, https://atlasti.com/product/technical-info/; Dedoose, accessed February 4, 2019, https://www.dedoose.com/.

BIBLIOGRAPHY

ATLAS.ti. "Technical Information." Accessed February 4, 2019. https://atlasti.com/product/technical-info/

Davidson, Judith, Trena M. Paulus, and Kristi Jackson. "Speculating on the Future of Digital Tools for Qualitative Research." *Qualitative Inquiry* 22, no.7 (2016): 606–10. doi:10.1177/1077800415622505.

Dedoose. Accessed February 4, 2019. https://www.dedoose.com/.

Evers, Jeanine C. "Current Issues in Qualitative Data Analysis Software (QDAS): A User and Developer Perspective." *The Qualitative Report*, Volume 23, Number 13, Article 5 (2018).

Le Blanc, Amana Marie. "Disruptive Meaning-Making: Qualitative Data Analysis Software and Postmodern Pastiche." *Qualitative Inquiry* 23, no. 10 (2017): 789–98. doi:10.1177/1077800417731087.

NVivo. Accessed February 4, 2019. https://www.qsrinternational.com/.

Paulus, Trena M., Kristi Jackson, and Judith Davidson. "Digital Tools for Qualitative Research: Disruptions and Entanglements." *Qualitative Inquiry* 23, no. 10 (2017): 751–56. https://doi.org/10.1177/1077800417731080.

Robins, Cynthia S., and Karla Eisen. "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, no. 10 (2017): 768–78. doi:10.1177/1077800417731089.

University of Michigan Safe Computing. Sensitive Data Guide. "Dedoose." University of Michigan. Retrieved October 1, 2018. https://www.safecomputing.umich.edu/dataguide/?q=node/231.