CHAPTER 12

Coding and Data Display

Learning Objectives

- ✓ Define coding as a strategy for data analysis
- ✓ Describe strategies for coding including individual and group coding
- ✓ Understand different strategies for data display including temporal, site, and theoretical displays

Cases in Research

Case A—National Helping Hands

Following the interviews and field research, the research project leader for National Helping Hands (NHH) finds herself with a great deal of data. Each interview has produced pages and pages of text on a wide range of subjects. Each site visit team brought back a wide range of information including documents collected from the locations and notes taken during their stay. In all, it looks like there are close to a thousand pages of notes for this research project. What she needs now is a strategy for analyzing these vast sources of data to provide insight into the role of nonprofit organizations in disaster recovery.

Case B—Springfield Housing Department

The Springfield Housing Department has conducted a series of interviews with its clients at four sites across the city. With little idea of what to look for, the department manager wants to address his core research question: "What barriers do

our clients face when applying for housing assistance?" He wants to investigate this general question while also looking for patterns that may emerge from specific sites. What he needs now is a way to analyze the numerous interview responses to facilitate the comparison of the housing department sites and the challenges reported by clients at each location.

Coding Texts

It is often the case that after data collection, you will find yourself overwhelmed with the volume of material that you have collected. One researcher has called this the "10,000 page problem." The very purpose of data analysis is to reduce the complexity of the real world into usable information. If one simply reproduces the complexity of the real world, you have not actually done any useful work. To analyze data means to reduce the complexity of information to produce simple (or at least simpler) conclusions or answers to your research questions. One starting point to the process of reducing the complexity of your data is to use established coding strategies. This chapter will provide an introduction to coding procedures along with some strategies for analyzing coded data using data display techniques.

What Are Codes?

The goal of coding your data is to identify general patterns that emerge from the vast volumes of information you have generated. You can think of this process much like taking notes in a class lecture. Most of us are not capable of writing down every word that a lecturer utters. Instead, you take notes that select key phrases and information from the lecture. You may write down a key word or phrase that you want to remember. At times you may write down extended passages including formal definitions—but these tend to be rare in lecture. You may also draw small pictures to represent relationships between key concepts in a lecture. What one seldom does is simply transcribe the lecture verbatim.

Active note takers will even write down reactions and inspirations to the lecture. If a lecture introduces a concept like selection bias, an active lecturer might write down reactions such as "How often does this apply in the research that I conduct?" or "What can I do about this?" These comments inject your interests and perspectives into the material to which you are listening. The purpose is to go back to your notes, possibly as you study for an exam, and have a shorter, simpler representation of the lecture that boils down to what is most essential for you to remember.

The process of coding your data is much like taking notes on a lecture. You will take notes (with the notes called codes) on sources of data so that you can review your data quickly and search for the areas of the data on which you want to focus on at any given time, etc.

A **CODE** is a meaning or characteristic that you assign to a piece of data. The process of assigning meaning or defining the characteristics of a piece of data—**CODING**—is the essential starting point to data analysis.

What Types of Data Call for Coding?

In some sense, just about any data calls for coding. A survey with closed-response options requires respondents to code their own responses. The purpose of these closed-response options are the same as any other coding strategy—to create comparable and easily interpreted response options. Open-ended response questions in surveys do not interpret themselves, though. If you ask people to identify the most important problem the city faces, you will get a wide variety of responses, and many of these responses may be similar enough to group together. One respondent may say "jobs" while another says "unemployment." Chances are you want to count these two responses as equivalent. If you had used a closed-ended survey question, you may have even had a single option labeled "jobs/unemployment." Everyone who marked that single option would be assigned the same numerical code. With an open-ended survey question, you will have to group responses together on your own. This exercise is a classic example of coding.

Other types of data more obviously call for coding. In some interviews, people provide answers in natural language. Like with open-ended survey questions, you will have to assign meaning and characteristics to these responses. You will have to decide whether discussing "jobs" as a problem is equivalent to "unemployment." The richness of responses when people are allowed to speak for themselves provides many opportunities for attributing meaning and assigning characteristics. Rather than simply lumping terms like "jobs" and "unemployment" together, you may read the surrounding discussion to investigate whether respondents have subtly different meaning to these two terms. While the terms in isolation could be difficult to distinguish, in context you may see differences emerge. It could be that people who refer to "jobs" tend to discuss the problem in reference to specific friends and family who are having difficulty finding jobs while those who refer to "unemployment" discuss the problem in abstract and depersonalized terms. It could be that friends and families lose jobs while faceless other people experience unemployment.

The process of coding is not limited to interview texts. One could code meaning and descriptions to just about any source of data. If during a field visit one were to collect documents from the research site, one could code these documents. One could analyze photos or videos taken from the site in a similar way. The world is full of potential meanings that can reveal an underlying process of relevance to public managers in a variety of ways. Coding is the process of extracting or imposing meaning on the world around us.

Codes and the Unit of Analysis

With the diversity of potential types of media for coding, it is important to consider what it is, exactly, that one is coding. The correct answer depends on the nature of the research question one seeks to answer. If you have a photo of a number of people, one may study the photo as a whole ("How many people are waiting for the train?", "Where

are people sitting while they wait for the train?"), or on each specific individual in the photo ("What did each person bring with him or her to the train station?").

Similarly, one could focus on different units for interviews. One could argue that every word is deliberately chosen and thus code individual words. Alternatively, one could focus on entire sentences as wholes. In written documents, one can focus on paragraphs, sections, or entire documents. Your choice of the unit of analysis will depend on your research question. Some researchers will focus on incredibly precise units of analysis—even counting the number of seconds one waits to respond to a question. If one's theoretical approach suggests that the answer to your research question operates on so small a scale, this may be appropriate. If, however, your research question and theoretical approach directs you to focus on the context of words or meanings that only emerge as part of entire sentences, then you will want to focus on that larger unit of analysis. In the end, you have to choose a unit of analysis that has the meanings and characteristics that you want to investigate. Some researchers refer to the basic units that they are studying as **CHUNKS**.

Types of Codes

We can start with two general phases of coding, each representing a different relationship between the code and the data source. Understanding the types of codes is essential to understanding how coding can be a tool for answering research questions in public affairs. Initial efforts to analyze the data will rely on descriptions of the text. Later analysis may start to group these descriptions together to identify patterns between the described elements. These coding approaches are known as descriptive and pattern coding, respectively.

Descriptive Coding

The first type of code is a **DESCRIPTIVE CODE**. Descriptive codes are uniform descriptive terms that a researcher assigns to material he or she is coding. One could describe many aspects of a chunk. As an initial part of your research, you will need to define the codes that you will use. We will discuss the development of this list in the next section. For now, just imagine that you have such a list at hand. Your purpose is to assign short descriptive terms to each chunk for easy reference later. Your aspiration is to later be able to gather all of the chunks that include similar topics.

Putting Knowledge into Practice: Descriptive Coding Partner Identification in Recovery Operations

The director of National Helping Hands has collected a wide variety of documents related to disaster recovery operations following some recent floods. These docu-

ments include news reports of recovery efforts, official documents related to these efforts, and follow-up reports called "after-action reports." She wants to know which nonprofits (and which types of nonprofits) are active in these recovery efforts. This requires coding this diverse set of documents to identify all the nonprofit participants. She reads through each document marking each mention of a nonprofit organization (possibly keeping a separate list of her own). For each document, she creates a roster of nonprofit actors involved. She also creates an overall list of all actors mentioned in any document. This reveals the diversity of nonprofit organizations in these recovery activities (within the limitations of the available sample of documents).

Description can be a complicated process. The key is using descriptions that are relevant to your research project. Sometimes descriptive codes are easy to implement. It is not too hard to identify the actor's names within a document (though you may sometimes find sources referring to actors by different names and will have to develop a strategy for reconciling these differences). Other descriptions may be more subjective. If you were studying public opinion related to a proposed local ordinance, you could conduct a series of interviews. You could then code the transcripts for positive, negative, or neutral statements related to the ordinance. If a respondent mentioned that the ordinance was potentially costly to the city, you could mark the passage with a negative code. However, there may be some uncertainty as to whether a statement is positive or negative. We will later discuss some techniques for assessing the subjectivity in codes.

One strict approach to descriptive coding, **IN VIVO CODING**, requires limiting codes to the specific language used within the original data source. If a respondent refers to "jobs" as a public problem, the code would need to be "jobs" and not combined under a general heading like "economics" or even "unemployment." This disciplined approach limits the potential for coders to read meaning into the original data that may not be there. On the other hand, in vivo coding may artificially distinguish codes that refer to what is best treated as a single subject. In vivo coding also limits the ability to link codes to the language of research questions because you are limited to the language of respondents—which may be less formal or less precise than a good research question.

The context of your codes will depend entirely on your substantive research question. While we cannot define all of the different types of codes that you may use, consider these examples of types of codes.

Attribute Codes: These codes refer to the attributes of the source of the data you are coding. You may have many basic attributes from the data collection sheets you generated for data management. For example, you may have recorded the name, position title, gender, age, site, or other attributes of interview subjects. These become attribute codes for data analysis purposes. If you wanted to look at all interview transcripts from

people at a given site, the attribute code would allow you to quickly reference all of the relevant data.

Descriptive Codes: These codes refer to the topic raised within a data source. If an interview respondent discussed the challenges of getting reimbursed for disaster recovery activities, you may code that chunk as "financial" or "reimbursement" (or some other code motivated by your research question). This type of code is not limited to interview data. You could also code notes taken by field researchers based on the subject they are describing in their notes. If they describe the behavior of people in the waiting room at the housing office (are people sitting or pacing, etc.), you can code these notes as well. You could potentially code the field researcher's observations on the layout of the room or any other description that seems relevant. Within an audio, video, or photographic recording, you could use codes to describe the content of the record.

Valence/Magnitude Codes: You can step beyond descriptive coding to attribute meaning to aspects of the record. You could assign a value to statements as being positive or negative in relation to a subject. For example, if interview subjects from the housing office complain about their long waits in the lobby, you could code this as a negative report. A statement about the politeness of the people with whom the interview subject interacted in the office would be a positive report. You could go further to code the degree of positivity of negativity as the magnitude of a report. One could separately code mildly positive and negative responses ("The people at the front desk were helpful to me.") and strongly positive and negative responses ("I have never been treated so rudely in my life."). Magnitude codes can be as precise as you want—though you want to ensure reliable coding of magnitudes. We will discuss reliability of codes in the next section.

Pattern Coding

The second type of coding is **PATTERN CODING**. Pattern coding involves identifying relationships between elements within the coded data. This requires reading beyond the immediate data to inferences about these relationships. In some ways, pattern coding represents the coding of descriptive codes. Pattern codes can look for two different patterns within the (description) coded text. First, you can look for clusters of description codes. Clusters of codes are a method for collecting similar codes together for further analysis. Of course, what counts as similar depends on your research project.

Putting Knowledge into Practice: Pattern Coding Barriers to Housing Application Descriptive Code Clusters

After initial descriptive coding of the interviews with housing applicants, the manager wants to define the types of barriers that respondents report. The first

step is to create a list of hypothesized clusters. The manager decides to focus on three initial clusters: (1) transportation barriers, (2) administrative barriers, and (3) personal barriers. The first barrier includes all references to difficulty traveling to or from the housing office. The second barrier involves complaints regarding the paperwork and procedures related to housing assistance. The final barrier involves statements regarding personal conflicts that make getting to the housing office difficult. Examples of the latter may include difficulties securing child care or getting off of work. The manager considers a fourth cluster—psychological barriers. The manager is concerned that people may be reluctant to apply for housing assistance out a sense of shame. Instead of including this as a fourth code, the manager includes an "uncategorized" code. The manager plans to go back following the pattern coding exercise to look at all of the "uncategorized" codes to look for clusters within this group—including psychological barriers or something completely unanticipated.

The second type of pattern code is a relationship code. These codes look for relationships between descriptive codes. This type of codes can accommodate a wide range of relationships. While clusters codes look for descriptions that are all of one type of phenomenon, relationship codes look for connections between different codes. For example, if you see an example of what you think may be a cause and effect, you may code it as such. These codes are considered preliminary and suggestive. These codes represent your reactions to the descriptive codes and the accounts included in the chunks you are coding.

Developing Codes

The examples of coding in the previous section took for granted that the research already has a set of codes ready to apply in a research project. Of course, one has to develop a list of codes just as one would develop a sample of interview subjects, write survey questions, or decide what documents to collect from a site. There are two predominant approaches to this process. This section will focus on theoretically directed processes for generating codes. This approach parallels deductive reasoning.

The starting point for developing codes is in the language of one's hypotheses or research questions. What are the key concepts that your research questions involve?

Putting Knowledge into Practice: Developing Codes Related to Types of Nonprofits in Disaster Recovery

The director of National Helping Hands is interested by the results of her initial assessment of nonprofit organizations active in disaster recovery. However, she

feels that a long list of names is of limited use. She wants to look for patterns in the types of nonprofits that are active in the disaster recovery event. This requires identifying the types of nonprofit actors that interest her. In her previous work, she knows that there are nonprofits with primary affiliations to religious institutions such as Lutheran Disaster Response. One code that interests her, then, is religiously affiliated disaster nonprofits. Further brainstorming provides some other potentially interesting codes: local/nonlocal, disaster-focused nonprofits, service provided (housing, feeding/nutrition, employment, health, education, etc.). She notes that these codes are not mutually exclusive but will enable her to quickly find references to nonprofit organizations of specific types rather than having to look for specific nonprofit organizations.

Coding Strategies

The work of coding involves analyzing data sources and applying codes where they are appropriate. While this is a relatively simple operation, care and attention are vital to produce useful research. A well-designed process for coding your data will provide more consistent data. This data will then be more amenable to analysis. This section will discuss some specific strategies to ensure consistent codes amenable to later analysis.

Individual Coding Strategies

The first set of strategies involves individual coding. These are the approaches one can employ to ensure effective and consistent coding. The goal is to set up your coding procedure to ensure that you read every chunk in the same way, with the same care, and employing the same potential codes. The key is consistency. You want to code every piece of data with consistent rules.

A good coding process may start before you actually look at the data. Consistency in coding requires, at the very least, a consistent understanding of the codes themselves. If you change the meaning of the codes as you are reading, you will treat later chunks in a different manner. This compromises the consistency of coding. Of course, achieving perfect consistency in the meaning of codes is often difficult and may be impossible. We often use coding techniques when we want to understand concepts better (or the meaning of concepts in certain contexts). In these situations, it is natural that our understanding of concepts and the meaning of codes changes over time. Our goal though is to minimize these changes in most situations—especially avoidable changes.

To promote consistency in codes, it is best to invest a great deal of thought into defining your codes in as much detail as possible. This requires developing a set of codes along with a formal definition. It is best if you take the time to write out the definitions of each code. This exercise will force you to confront ambiguities in the codes and to take a stand on specific meanings. This process can supplement your brainstorming efforts. Where you identify incompatible definitions of a candidate code, you can split the code into two mutually exclusive codes. This exercise will both promote consistency in your codes and provide a more detailed set of codes from which to choose.

Once you have a code list with clear definitions, you can begin the actual coding process. How you code is largely a matter of individual style and workflow as well as the nature of the chunks you are coding. We will discuss some basic procedures for coding for three common types of chunks: interview transcripts, documents, and video. This discussion will provide simple strategies that one can implement with standard software or no software at all. The final section of this chapter will discuss how to use specially designed software for coding and data analysis.

Coding interview transcripts can use strategies similar to taking notes in a class. We recommend a split-page note-taking framework. In such a framework, you only print the data source on one side of the page. Typically, the right side of the page is left blank—possibly because of the preponderance of right handedness. It does not really matter which side you leave free, though. You can then number the chunks along the edge of your free space. Next to each chunk, you can list the codes that you feel apply (if any). The setup of the page should resemble Exhibit 12.1

Exhibit 12.1—Coding Interview Transcripts

Questioner—How did you get to the housing office to apply for assistance?	
Respondent—I took the bus . . . and walked some of the way. [1]	Chunk [1] Codes: mode (bus), mode (walk-in)
Questioner—Was it difficult to get to the office?	
Respondent—Getting to the office was kinda' hard. I had to find care for my kids. I did not want to drag them along and then wait in the lobby. The office is in an odd place. The buses do not run near there. I had to catch two buses and then still walk to get here. [2]	Chunk [2] Codes: barrier (child care), barrier (waiting time), barrier (bus stops), barrier (walking distance)

Notice in Exhibit 12.1 that you can apply multiple codes to a given chunk of data. In this case, the answer to a direct question about the difficulty of getting to the office generated four different codes. This raises the question of what the actual chunk size should be. In interviews, people rarely answer questions in easily identifiable paragraphs. The exhibit uses the entire response as a chunk. For some purposes, you may want to look at each sentence or even each word. The choice of chunk all depends on your theory and hypothesis. If you think that the choice of specific words is informative to your study, the word could be a good chunk size. If you think that sentences are the natural unit for your study, you can use sentences. It is always easier to aggregate up from codes applied to smaller chunks to larger chunks (say, from words to sentences) than it is to do the reverse. We have found it difficult to attribute meaning to words and sentences in isolation, so we tend to use clusters of sentences (responses) as the chunk. In other fields, other chunk sizes may be more common. Some scholars of psychology, for example, are intensely interested in the time (in seconds or smaller units) that people pause between answers or when they interrupt each other. In these studies, chunks may be much smaller units. The choice depends entirely on the questions you seek to answer.

The basic ideas behind coding interview transcripts are instructive for coding other types of media. The clearest analogue to interview transcripts is a formal document. This set would include any documents you collect on site or that you otherwise feel are relevant to your research. You may not want to change the format of documents to force them into a split-page format similar to what we recommend for coding interview transcripts. If you want to use a process as similar a possible, it may be possible to rotate the document 90 degrees and print it on one half of a printed page. Alternatively, you may create shorthand for your codes. For each chunk, you can write a small number indexing the chunk along with a brief notation indicating codes. If the margins of the document are generous, you can write a word or words in the margins themselves. If the margins are thin or you have many codes, you may need to number your codes or create alphanumeric representation code (e.g., "7" or BAR-CHLD to represent the barrier of child care needs). Such shorthand is easy to use but may make the codes harder to interpret later (especially by people who were not involved in coding earlier).

Coding videos has a number of differences. As a form of media that does not have a space you can reserve for written codes (even shorthand), you will need to create a separate document for storing your codes. The trick to coding video data is to create a correspondence between the specific code and a video chunk. Most often, this is done through the use of time codes. A time code is simply an indication of the starting time and ending time for a segment of the video. You can use these time codes to create chunks in what would otherwise be one continuous stream of video. These chunks do not need to be of uniform length or spaced uniformly through the video. Think of this strategy as defining scenes (as in a movie). Some scenes may be shorter or longer. The key is having the scenes be discrete units. In movies, this may involve having scenes broken up by the location of the events. For other types of videos, you can separate the chunks however you like. You can define the chunk (say, "from the 30 second mark to the 48 second mark") and then associate codes with this chunk. Typically, you will create

a separate document that lists all of the chunks and any associated codes. The same logic holds for photos. While photos do not require time codes, they require some unique identifier for each photo (presuming each photo is a chunk) and a separate sheet linking the unique identifier to codes.

It is generally a good idea to set aside time when you have completed coding the specific series of chunks to write a summary memo. Memos provide a quick reference to the content of the data. You can include a summary list of the codes from the specific data source. You may move beyond a simple summary to include your reactions to the data overall, comparisons between this data source and other data sources, and thoughts about future research. The memos provide a nice starting point if you need to look back at the data without reading through each of the data sources themselves.

Group Coding Strategies

Larger projects will require the use of entire teams of coders for data analysis. Spreading the work over a number of people allows you to code a great deal more data in a shorter period of time than would otherwise be possible. However, having an entire team of coders requires taking special precautions. Again, the focus of the process is ensuring consistent coding across all of the data. The last section discussed steps to ensure consistency when just one person is coding the data. The challenge grows when trying to ensure consistency across multiple coders.

The basics of coding remain the same. Each coder acts as an individual coder. However, there are two steps you can take to promote consistency in a team coding environment: training and mid-stream team meetings.

Coder training is essential for group coding processes. In the case of individual coders, we recommended that you take great pains to define your codes. The team training exercises are the equivalent of group coding processes. Of course, defining the codes will still be essential. However, you now need to ensure that your code definitions are clearly and consistently understood by all of your coders. In addition to clear definitions, it is useful to provide as many examples as possible as well as training passages to test whether coders are employing the codes as you expect. It is best if you keep these training passages as similar as possible to the material you are coding without drawing them from your sample itself (because your training passage would get far more scrutiny than any other passages if they were singled out for use in training).

The need to gather all of the coders does not end following training and the beginning of actual coding activities. To ensure that your team of coders continues to use the same definition and to deal with problems that emerge in the coding process, it is best to have coding team meetings throughout the process. At these coding team meetings you can ask for problems that the coders are encountering. These problems may be in the coding management process (time management, paperwork, etc.) or ambiguities in the meaning of codes. This is a great time to listen to the coders to hear what difficulties they are having assigning codes to chunks. They can tell you whether they feel that there are important concepts for which they do not have a code or if they are using generic or miscellaneous codes where they feel new, specific codes are in order. The

coders can also tell you if they are having difficulty deciding between two codes where both seem to apply to a given situation. This can help you update your definitions to clarify the meaning and use of codes. It helps that these meetings are also a time where you can disseminate any changes to coding procedures, code lists, or code definitions to all of your coders.

Reliability Assessment

The previous sections have discussed steps you can take to promote consistency in the coding process. While you should implement such strategies, they are no guarantee of consistency. Instead, you should also check the consistency of coding once the process is complete. This process is referred to as reliability assessment. For the most part, people use reliability assessment in group coding situations. They can assess inter-coder reliability—that is, whether different coders are identifying the same codes from the same passages.

There are many approaches to assessing inter-coder reliability. The simplest approach is percent agreement. The starting point is to have two coders analyze an identical passage or series of chunks. After each coder analyzes each chunk, you compare the codes they have assigned. For each coder, count the number of codes they applied that were also applied by the other coder. Each of these is an agreement. Then, count the number of codes applied by each that were not matched by the other coder. Each of these is a disagreement. You can calculate percent agreement with the following simple equation:

Percent Agreement = Agreements / Agreements + Disagreements

This approach to calculating reliability is simple but has some serious limitations. The approach works best when there are a small number of potential codes to apply. In the simplest case where you have one code to apply (or not) to every paragraph, the measure provides a nice summary of the agreement between two coders. This is the case with most codes that will eventually be part of a quantitative assessment—as discussed in future chapters. Typically, people want to see reliability in the 80–90% range with these conditions. However, as the number of codes increases, the number of potential disagreements increases. Sets of codes with fine distinctions make disagreement much more likely. The standards for such research are not lower (people still want close to 80% agreement), but these situations will likely require extensive training to prepare for consistent coding.

With projects in which there are sufficient resources, one can have two coders analyze every chunk and only retain those codes on which there is agreement. You should still conduct reliability assessments to see how many disagreements are present, but this approach provides a great deal of confidence in the codes you do use. If it is not possible to allocate two coders to every chunk, you should still choose a small sample on which to conduct such an assessment. This can represent only a small number of chunks or even 10% of the sample. You need to select enough to ensure confidence that the codes are applied reliably—or to redesign the process and retrain the coders until you can be confident.

Data Display Strategies

Fundamentals of Data Display

You can think of codes as an essential bridge between raw data (even 10,000 pages of raw data) and hypotheses. The real world is messy and confusing. It only reveals itself to researchers slowly and seldom directly. Instead, we have to look for meaning in messy data. Your hypotheses should be stated precisely and clearly. Coding translates raw data into a format in which you can compare your expectations to what you actually observe.

The resulting coded passages are useful as a means to test hypotheses. Each chunk is an observation and an opportunity to falsify your predictions about relationships. Even thoroughly coded texts are not easy to analyze. You still find yourself with pages and pages of text and associated codes. With codes in place, you can begin analysis of the data. The actual analysis should begin with an exploratory display of the data. This section will discuss a variety of strategies to display your codes in a way to facilitate hypothesis testing.

Summary Display

The starting point is to summarize your codes. You have a list of codes with which you have analyzed text. It is natural to first ask which, if any, of the codes were used in the analysis and with what frequency. You can start with summaries of text characteristics. If you have conducted interviews, you can provide simple summaries of the number of interviews with people with various characteristics. How many were men and how many were women? How many interviews did you conduct at each site? These are characteristics that you did not have to code chunk by chunk but are characteristics of entire interviews.

Exhibit 12.2—Summary of Housing Interviews

Total:	22 Interviews
Gender:	
Female Interview Subject	17 Interviews
Male Interview Subject	5 Interviews
Site:	
Site A	9 Interviews
Site B	6 Interviews
Site C	7 Interviews

Barriers mentioned:

Transportation	16 Interviews
Waiting time	4 Interviews
Paperwork difficulties	7 Interviews
Child care	14 Interviews
Time off of work	5 Interviews

The resulting summary can provide some insight into the subjects under study and provides the foundation on which many of the other data display techniques will be built.

It is important to note that there are serious limitations on what you can learn from these basic descriptions of the data. A sample of 22 interviews is small. If you had chosen 22 people off the street for questions about their political opinions, the answers would not provide a great deal of insight into the proportion of the greater public that holds specific opinions. Even if 11 people had said they supported increasing spending on education, you should be hesitant to suggest that 50% of the general public support the same. The proportions and frequencies of responses within such small samples should not be taken to provide clear evidence of proportions and frequencies in the general population.

However, there are some conclusions you can reach from even small samples. The most obvious is the existence of some opinion. For example, if a respondent mentions that he had a problem getting time off work to get to the agency, it is safe to conclude that this is a problem within the general population. You can't infer how frequently this problem occurs or whether it is more common than other problems. The results provide a view of the existence and range of responses. More detailed research questions about frequencies and ratios should be left to other research tools.

Site Display

Basic summaries can be useful, but there are other tools that can provide insight into more nuanced patterns within the data. In these cases, use displays to organize responses and visualize patterns. There are a wide variety of potential display strategies, but we will focus on a few that are particularly useful for public management research.

Public management research often involves the comparison of different offices or locations. If you want to compare different neighborhood offices or school buildings, you can break up the display of information along the lines of the sites you want to compare. The pattern of responses you discover when you consider responses site by site may reveal information suggestive of hypotheses and warrant further investigation.

Putting Knowledge into Practice: A Site Comparison of Housing Office Site

The housing manager wonders if the barriers that people encounter when applying for housing assistance vary based on the location of the offices themselves.

She wonders, specifically, if there are some offices where applicants face more barriers to transportation. Based on the basic summary of interview responses, she knows that there were between 6 and 9 interviews at each site. She decides to break down the barrier responses by site to see how many specific barriers are mentioned in interviews with clients at specific sites.

Site A	9 Interviews
Barriers mentioned:	
Transportation	8 Interviews
Waiting time	4 Interviews
Paperwork difficulties	3 Interviews
Child care	5 Interviews
Time off of work	2 Interviews
Site B	6 Interviews
Barriers mentioned:	
Transportation	1 Interview
Waiting Time	0 Interviews
Paperwork difficulties	2 Interviews
Child care	4 Interviews
Time off of work	2 Interviews
Site C	7 Interviews
Barriers mentioned:	
Transportation	6 Interviews
Waiting time	0 Interviews
Paperwork difficulties	2 Interviews
Child care	5 Interviews
Time off of work	1 Interview

There are a couple of interesting patterns. First note that waiting time was only mentioned as a barrier in one of the sites. This is an indication that further investigation into the waiting time at that location is in order. Something seems to be setting site A apart—though these data are far from definitive in that they are based on a small sample of interviews. The results are more inspiring than conclusive. Notice also that only one respondent mentions transportation barriers at site B. This also warrants further investigation. What is it about this location that seems to be easier to reach than the other sites?

As the example illustrates, reporting codes based on different sites can be revealing. You can see patterns in the differences in codes appearing in different sites. The results are not conclusive or definitive. However, they can be inspiring.

Patterns that emerge even from these small samples can justify more detailed investigation—often justifying more attention. Maybe more importantly, such displays can help you prioritize investigations by identifying patterns that stand out most obviously. In the case of site displays, they can help you target sites for more detailed investigation (possibly more interviews or an entirely different data collection strategy).

Time Display

There are a variety of other simple data display strategies that have proven useful for public management research. It may be that you are not interested in diversity across different sites but across different time periods. When your theory suggests that conditions may change over time, displaying codes across time is a potentially useful tool. This may be the case where you expect phenomenon to develop over time. It could be that the implementation of a new program will only slowly reveal problems. It may be that you want to compare responses at different points in the calendar year. This could be particularly important in situations like education management where the nature of work changes dramatically in the summer.

These two theoretical inspirations undergird two types of time displays: calendar displays and duration displays. In a calendar display, you break up codes by the calendar day, week, or month of the data source. When your research question involves the relationship between calendar date and a code, these displays can be useful. Duration displays involve arbitrary starting dates and then durations removed from these starting dates. You could, for example, look at codes one month or six months following the implementation of new software in an office. It does not matter whether the first month was January or July. What matters is the separation of the data source from the start date of the program.

Putting Knowledge into Practice: Disaster Recovery Partners by Time

The manager of National Helping Hands is curious as to whether the nonprofit organizations that help with disaster recovery change over time. Are the organizations that help out in the first month following a disaster the same ones that continue to work in the community six to nine months later? Here the specific calendar month is not the central concern (though that could be interesting to look at later). The central concern is the change in the roster of nonprofit partners at different points in time after a disaster event. Previous efforts coded documents describing disaster response and recovery efforts following a flood. The manager has decided to

use a time display to compare the organizations named in the reports immediately following event (the first 30 days) to those who were mentioned as acting two or more months after the event. The resulting display may look like the following:

Month 1	Month 2	Months 3–6	Months 6–12
Red Cross	Local Church A	Local Synagogue A	Local Church A
Salvation Army	Local Synagogue A	Local Temple A	Local Temple A
Local Temple A			

The manager notices that as time passes, the number of active organizations drops off. This supports her hypothesis that the composition of nonprofit groups changes as time passes in the response and recovery phase. While not definitive, this evidence suggests further investigation may be warranted.

Theoretical Display

The logic of data displays is not limited to the previous examples of summaries, site displays, and time displays. These types of displays are motivated by specific theoretical concerns. Any theoretical concern that inspires a hypothesis about how different types of units may exhibit different coded characteristics can be the basis of a theoretically tailored display strategy. The key is having some independent variables that you suspect explain some of the variation in some dependent variable—a very general condition. Whatever the independent variable is, you can use that as a dimension in your data display. In the past two examples of display techniques, the independent variables were site and time. You can use many other potential factors. If you believe gender affects the barriers to housing applications, you can simply separate barrier codes by gender. You could alternatively use age, race, level of education, or any of a variety of other characteristics. The logic of data display with codes arrayed and ordered based on an independent variable is fully generalizable to just about any theory.

Putting Knowledge into Practice: Disaster Recovery Partners by Role and Time

The manager of National Helping Hands was encouraged by her initial display of nonprofit partners across time. Clearly there are differences across time. To more formally display some of the potential differences, the manager wants to combine a

display across time with a display separated by role. This adds a new dimension to the prior display. Now the arrangement is time by role:

	Month 1	Month 2	Months 3–6	Months 6–12
Evacuation	2	4	1	0
Health Care	8	2	2	2
Counseling	0	1	1	0
Economic Support	1	2	3	4

The pattern the manager finds is not entirely surprising but may be useful for planning purposes. The quick exit of health care organizations may inspire efforts to keep those organizations involved for a longer duration. The slow build-up of economic support organizations may be exactly as planned. The delayed introduction of counseling organizations may inspire efforts to recruit more of these organizations and to bring them into recovery efforts faster.

Software for Coding and Data Display

The preceding discussion has not assumed anything about the technology you have available for data analysis and display. You could perform any of the analytic techniques described earlier with paper and a pencil. This section will introduce the basics of software that can assist with coding and these basic data display techniques.

It is natural to ask why one would bother using software to assist in coding and data display. If you can accomplish your goals without sometimes expensive software that requires additional training, why adopt the software at all? The answer is simple. The software, while requiring some additional efforts, facilitates the process and makes some advanced techniques much simpler. Software suited to these tasks is known as computer-assisted qualitative data analysis software (**CAQDAS**).

There are two general types of computer software packages for coding and basic data display. There are freely available open-source packages that provide a limited set of options. For researchers that need a more robust toolset, commercial packages are also available. This section will conclude by describing the most common features of these two types of software packages and the most prominent representatives of each. You should note that the world of software is constantly changing. Some of the packages most popular today were not even in existence five years ago. The descriptions of the merits of each type of software, the options available in each, and the names of specific software packages may all change rapidly. The general features in CAQDAS, however, are likely to be of persistent interest to those using techniques like those described in this chapter.

CAQDAS software generally includes the following: search tools, coding tools, note-taking tools, and query tools. Other tools are available in some packages, but these

functions represent the core. The basic operations of the software involves, first, loading a data source into the software. If you have interview transcripts or documents, this will create searchable text. If you have photographs or video, searching the media directly is not generally a concern (or an option). Search functionality is useful for finding segments of texts but is not itself an analytic technique or particularly unique to CAQDAS software. You can load documents into just about any viewer or word processor to accomplish the same goals. CAQDAS software does a lot more than this, though.

The first unique software function of CAQDAS software is the development and management of codes. CAQDAS software is designed to facilitate the cataloging of these codes as well as, in many cases, the clustering and hierarchical arrangement of the codes. The software lets you keep track of formal definitions of codes along with the relationships between codes. The software keeps track of such relations as having a set of codes for barriers to applying for housing assistance where each is distinct (child care, transportation, etc.) while also being a type of barrier. This software generally also helps the development and elaboration of codes throughout the analysis process.

The second unique function is assistance in coding. The specifics of coding vary from package to package, but the basics are consistent. The software allows you to read through and assign codes to the various chunks of your data. This operation typically involves processes like you might find in "track changes" operations in word processing software. In fact, many "track changes" visualization styles look just like codes attached in CAQDAS. The differences are vital, though. CAQDAS software knows that what you are doing is assigning codes rather than attaching generic annotations. In fact, many CAQDAS software packages allow either notes or codes. Codes are something special, though. The software keeps track of which codes are assigned to which chunks. It knows that codes are going to be used repeatedly through the analysis.

Some CAQDAS software provides key stroke substitutions to make coding easier. The goal is to make coding fast and reliable. To this end, some of the software lets you assign keys to specific codes. You can read rapidly through the chunks and assign codes with a single keystroke. The result is faster coding once you are familiar with the keystrokes. This is particularly important for projects that involve a great deal of coding. Such automation can also help with team coding efforts. The software also facilitates the application of multiple codes to a single chunk.

The assignment of codes and the software recognition of codes are essential to the central distinguishing function of CAQDAS. Up to this point, the functionality of CAQDAS software largely overlaps with what you can do with basic word processing software using a combination of search and annotation functions. CAQDAS, however, allows for complex searches of the codes rather than searching the text itself. You can search for all references to a specific code indicating a reference to child care as a barrier to housing applications. The structured query will return all chunks with this code. This is the point where CAQDAS pays off uniquely. Once the coding is done, you can now do a variety of comparisons of the coded text with ease. You could reproduce the tables and data displays in the previous sections with ease. Instead of just listing counts of references, you could include entire chunks organized by site, time, gender, or topic. This is the function that justifies all of the start-up costs in formalizing codes development, defining the hierarchy of codes, and using this specific software for analysis.

This basic functionality is available in a wide variety of software packages. You can find open-source software to serve many of these functions. Attractive options include WeftQDA and RQDA. There is also an online program to facilitate group coding—Coding Analysis Toolkit (CAT). These open-source packages offer basic annotation, coding, and queries. Their chief limitations are with the sorts of media that the software can analyze and the complexity of the queries that one can use on the data. Open-source packages typically allow for simple text format (.txt files) and open document standards (like .pdf files). They typically do not allow proprietary document standards (like .docx files), video, or graphic data sources. For these more complicated data sources (or more complicated queries), you will need more complex, commercial software.

There are a variety of commercial QACDAS packages, but the most popular are NVIVO and Atlas.ti. These packages provide an ever-expanding set of functions including complex queries, rich text integration, and options to output quantitative queries (like those that one would use to create the tables in earlier parts of the chapter) to standard statistical packages. These statistical packages can cost thousands of dollars. In the end, you will have to assess whether the cost of the commercial packages is worth the investment for your specific project.

Conclusion

One of the challenges of conducting thorough research is that you end up with a great deal of data. Rather than being frustrated, you need a strategy to use the data to inform your decisions. This chapter has provided some initial strategies for organizing the data you collect in ways that will help reveal patterns. These strategies will be particularly useful for data like open-ended questions from interviews or field notes. Future chapters will expand your strategies for analyzing a broad range of data.

Suggested Readings

Coding and data displays are techniques on which fewer books have been written (compared to other topics like conducting interviews, writing surveys, etc.). These are often techniques that people learn by doing them. We find two texts to be most helpful for the development of coding strategies and data displays. Each of these books go far beyond the material discussed here and would be excellent material to consult as you embark on coding and data display exercise.

First, Johnny Saldaña's excellent *Coding Manual for Qualitative Researchers* provides dozens of potential types of codes appropriate at various stages of analysis. While the book draws from a field somewhat distant from much public affairs research (theater and education pedagogy), the diversity of coding strategies discussed is superb and inspirational.

Second, Miles and Huberman's *Qualitative Data Analysis: An Expanded Sourcebook (Second Edition)* provides a strong overview of qualitative coding and data analysis. While the text is starting to show its age, it provides diverse examples of data displays on a wide range of research topics—many of which will be comfortable to students in public affairs with their focus on education reform implementation and education management.

Those interested in CAQDAS software could begin by looking into the CAQDAS Networking Project (caqdas.soc.surrey.ac.uk). This group of researchers at the University of Surrey provides a number of resources to help explain and compare CAQDAS software. This is also a useful site to keep up with new software releases and training opportunities.

Vocabulary

Flashcards & MCQs

Useful Websites

CAQDAS—Computer-Assisted Qualitative Data Analysis Software.
CHUNKS—The basic unit of analysis of coding.
CODE—A meaning or characteristic that you assign to a piece of data.
CODING—The process of assigning codes to segments of text.
DESCRIPTIVE CODE—A code based on the specific meaning or characteristic of a chunk.
IN VIVO CODING—Coding that only uses language used within the original chunk.
PATTERN CODING—Identifying relationships between elements within the coded data.

Review Questions

- What distinguishes descriptive coding and pattern coding? Which occurs first?
- Identify four different types of chunks and when they may be appropriate.
- Under what conditions would a site display be most appropriate?

Discussion Questions

- Develop a list of descriptive codes for a project in which a public manager interviews teachers following the implementation of a new accountability system in their schools. The goal of the project is to explain why the program seems much more popular in some schools than others.
- Under what circumstances might you want to limit yourself to in vivo coding?
- Imagine that you have transcripts of a series of focus groups conducted with representative of a local neighborhood where Walmart is considering a new location. Each focus group was conducted at a different location and consisted of different representatives. What sorts of displays would you want to use to analyze the data?