

Using Software in Qualitative Research a step-by-step guide

Christina Silver
& Ann Lewins

second edition



Los Angeles | London | New Delhi
Singapore | Washington DC

1

Qualitative Data Analysis and CAQDAS

This chapter introduces the eclectic field of Computer Assisted Qualitative Data Analysis (CAQDAS) in the context of qualitative research methodology and the techniques of analysis generally. We discuss the practicalities of research in the software context, outline some basic principles and distinctions which resonate throughout the book; discuss software developments, debates and functionality; and discuss selected qualitative approaches. The remaining chapters build from here, describing some core tasks you might undertake using CAQDAS packages, illustrated via three case-study examples (Chapter 2). Our overall emphasis is on the inherent fluidity between the processes involved in analysis and how customised CAQDAS packages reflect and reinforce them.

We discuss analysis in the context of technological possibilities. Table 1.1 lists common analytic tasks enabled by CAQDAS, but software itself does not dictate their sequencing, or whether certain tasks are undertaken or tools are used. These decisions rest entirely with you, informed by the interplay between methodology, analytic strategy, technology and practicality.

Table 1.1 Common tasks of analysis supported by CAQDAS packages

Task	Analytic rationale
Planning and managing your project	Keep together the different aspects of your work. Aid continuity, and build an audit trail. Later, illustrate your process and your rigour through transparent writing.
Writing analytic memos	Manage your developing interpretations by keeping track of ideas as they occur, and building on them as you progress.
Reading, marking and commenting on data	Discover and mark interesting aspects in the data as you see them. Note insights as they strike you, linked to the data that prompted them – enabling retrieval of thoughts together with data.

(Continued)

Table 1.1 (Continued)

Task	Analytic rationale
Searching (for strings, words, phrases etc.)	Explore data according to their content. Discover how content differs across data and considering how familiarising with content helps you understand what is 'going on'.
Developing a coding schema	Manage your ideas about your data by creating and applying codes (that represent themes, concepts, categories etc.). The structure and function of a coding schema depends on methodology, analytic strategy and style of working.
Coding	Capture what is going on in your data. Bring together similar data according to themes, concepts etc. Generate codes from the data level (inductively) or according to existing ideas (deductively) as necessary; define the meaning and application of codes.
Retrieval of coded segments	Revisit coded data to assess similarity and difference, to consider how coding is helping your analysis, and prioritising 'where to go next'.
Recoding	Recode into broader or narrower themes or categories if appropriate and necessary. Perhaps bring data back together and think about them differently.
Organisation of data	Organise data according to known facts and descriptive features to allow consideration of how these aspects play a role in your understanding.
Hyperlinking	Link data to other data segments and/or to other files to track process, contradiction, association etc.
Searching the database and the coding schema	Test ideas, interrogate subsets for similarity and difference, identify anomalies, or generate another level of coding.
Mapping	Manage analytic processes by visualising connections, relationships, patterns, processes, ideas.
Generating output	Report on different aspects of your progress and the project at any stage. Save as files to capture status at an analytic stage, or to work in other applications. Print off to get away from the computer and think and work in more 'traditional' ways.

Qualitative research and data analysis

Qualitative research is a broad field that crosses disciplinary, methodological and sector-based boundaries, and it is important to acknowledge the variety contained within it. Different philosophical, theoretical and methodological traditions underpin the way researchers think about and do analysis. Much work has been done elsewhere to make sense of these – often competing and sometimes complementary – scientific principles. If you are new to the area we point you in the direction of the following in particular:

Bryman and Burgess (1994), Creswell (1998), Mason (2002), Bernard and Ryan (2010), Silverman (2010, 2011), Bazeley (2013) and Saldaña (2013). Neither the scientific and philosophical principles nor the disciplinary and methodological subtleties within approaches to qualitative research and analysis are the focus of discussion in this book. However, reflecting on your ontological and epistemological standpoints (i.e. how you understand the world to work and how you believe it can be investigated) is important in locating and justifying your research. In reading the literature you will come across many different terms used to define the context and manner of inquiry, including *perspective, framework, approach, strategy, methodology, and method*. There are no clear boundaries between or hierarchical structure to these terms; they overlap and are used differently in particular contexts. Categorisation of qualitative research in terms of data collection techniques has a long history, but detailed discussion concerning the *processes* and *procedures* involved in analysis (i.e. what we actually do) has only occurred more recently (Bryman and Burgess, 1994). This book discusses processes and procedures of analysis specifically in the context of customised software use.

The extent of diversity in the field is well illustrated by comparing the work of three authors, all of whom wrote during the 1990s yet conceptualised qualitative research rather differently. Tesch (1990) distinguished 27 *forms* of qualitative research (see Figure 1.1; p. 23). Woolcott (1994) differentiated qualitative research *strategies* according to six styles of collecting data (archival strategies, interview strategies, non-participant observation strategies, participant observation strategies, field study, ethnography). Miles and Huberman (1994: 7) argue that while a ‘core’ of recurring features exist across qualitative research, they are ‘configured and used differently in any particular research tradition’. They distinguish between three traditions: *interpretivism* (including phenomenology, social interactionism, semiotics, deconstructionism, ethnomethodology and hermeneutics); *social anthropology* (including ethnography, life history, grounded theory, ecological psychology, narrative studies and case-study analysis; and *collaborative social research* (action research).

The range of ways used to describe qualitative research and analysis illustrates the difficulty of adequately reflecting the diversity in how general principles intersect to result in specific strategies. Most authors concede there to be much overlap between the distinctions they draw; there is often even blurring between understandings amongst different authors using the same terms. Researchers combine data collection methods in qualitative research design and borrow elements from various approaches in developing specific strategies for investigating new social problems or for using different forms of data.

Problems in categorisation systems are illustrated particularly clearly in contemporary writings about mixed methods. As more authors enter the debate, a tendency to generate increasingly specific categorical systems to reflect diversity ensues. Increasingly subtle differentiations complexify to such a degree that the area can become more difficult to access for novice researchers. Nevertheless, broad overviews and summaries are important in gaining entry to any field of scientific inquiry.

The use of customised software is not *required* in order to conduct robust analysis. But its use enables us to be more transparent in how we go about analysis because

the tasks we engage in, their sequence, role and documentation can be more easily illustrated than when working manually.

The practicalities of research in the software context

The availability of customised qualitative software occurred within a diverse methodological field, which has only become more varied with digital technology, big data and the rise of applied, commercial and citizen-research. In addition there is increasing discussion of mixed methods approaches to research and analysis and the use of visual methods. Reflection about the rise of qualitative software and the implications of its use must be done in the context of the practicalities of research, in which analysis is understood as a core activity throughout an iterative process.

Whatever the characteristics of a particular study, there are certain core elements involved in doing research. Planning is paramount (Box 1.1). Authors usually discuss several aspects in planning and conducting research. Mason (2002), for example, discusses 'questions of strategy', 'generating qualitative data', and 'analysing qualitative data'; Boolsen (2006) distinguishes between 'problem formulation', 'research design', 'data collection' and 'analysis'. In our experience researchers often plan data collection carefully, but neglect to put the same degree of effort into planning the analysis.

In the context of the use of software, much less has been written about research design than in relation to qualitative (and, increasingly, mixed methods) approaches more generally. Di Gregorio and Davidson (2008) wrote the first comprehensive discussion of research design in the specific context of software use that transcends individual products. In further opening up discussion about the role of software in designing and conducting research, we identify six key tasks in setting up a software project to reflect initial research design (Silver and Lewins, 2014). These tasks reflect the sense in which CAQDAS packages are essentially project management tools which can be used from the earliest moments of conceiving a research idea, through all the phases of planning and implementation of analysis to the tasks of writing up an account for publication, preparing for a conference presentation or organising a thesis (Chapter 2).

BOX 1.1

ANALYTIC NOTES

Research design and software project set-up

Designing a research project is all about planning how you intend to carry out the research. What methods of data collection or generation will you employ? Why? And what will be the implications of doing so? What restrictions are there on the way you will proceed, arising

from the circumstances in which you will work? What are the likely consequences of your design choices? What is your analytic focus likely to be? How will you handle changes in focus? These sorts of questions should guide the way you set up a software project. Although it is common to plan research, often this is done primarily in relation to data collection. It is just as important, however, to plan the analysis. Using software from the outset will help with integrating all types of planning into your work.

These tasks are: (i) managing and referencing literature; (ii) defining research topic and questions; (iii) representing theoretical frameworks; (iv) incorporating research materials; (v) defining factual features; (vi) developing analytical areas of interest. They are inherently interrelated, occurring in tandem rather than as discrete stages. One of the main benefits of using qualitative software is that flexibility can be built into analytic designs to reflect changes as projects evolve. This is a common thread through this book. This way of thinking about setting up a software project emphasises the importance of making explicit what you plan to do and how you plan to do it. Woolf (2014a) describes these essential elements as the *strategies* and *tactics* of analysis.

Managing and referencing literature

Reviewing existing literature concerning your broad topic is a fundamental early task. Technological developments mean that this process is changing rapidly and significantly. Many journals have electronic versions providing free or easy access to full-text articles. Bibliographic software has developed to the point that it is quick and easy to transfer reference lists and online material directly into libraries, along with associated metadata. CAQDAS packages have also developed significantly in this area, with several now enabling the direct importation of PDF files and references from bibliographic software. Conducting a literature review within qualitative software is not only feasible, but also incredibly useful. Chapter 5 distinguishes between *direct* and *indirect* handling of literature, via annotating and coding full-text articles and/or developing critical appraisals about and linking within and between them. However you chose to proceed, integrating literature with the rest of your work through a CAQDAS package enables you to later systematically compare existing literature with your analysis.

Formulating the research problem and defining the research questions

Formulating the research problem is more than just deciding on the topic. It is informed by your ontological and epistemological standpoint and your familiarisation with and critiquing of the literature, both of which help you rationalise why

the area you are interested in requires further research. If you use software to facilitate the literature review, it makes sense also to write up your formulation of the research problem and define the initial research questions within the software project (Chapter 5). You can thus be explicit about your interest, assumptions, expectations and prejudices and link your writing to the literature that contributes to your problem formulation.

Representing theoretical frameworks

Whether your intention is to work within a clear theoretical framework, perhaps through applying existing theory or testing hypotheses on new bodies of data or areas of conceptual interest, or to develop theory from empirical data, you will never be working within a theoretical vacuum. Contrasting ways of working can broadly be distinguished according to the *direction* you are working in; whether *top-down* (deductive) or *bottom-up* (inductive). These approaches, and their combination (abduction), are discussed in the context of coding in Chapter 7. Whatever its role, it is important to relate your conceptualisation of the research problem to existing theory, to represent that within the software project at the outset and reflect how it evolves during the project. That might happen via memo-writing (Chapter 10) and/or the visualisation of theoretical contexts in visual maps (Chapter 11). You will be able to refer back to these ideas at later stages and compare initial assumptions and expectations with the analysis as you proceed.

Incorporating research materials

Data collection is all about constructing the best possible dataset in order to investigate the research problem. Under ideal circumstances, what data are required to answer the research questions? What data are available? Could you construct a suitable dataset from existing sources and conduct secondary analysis, or do you need to collect new data? What instruments will you use to generate new data if required? How will you ensure data are of sufficient quality? In the context of setting up a software project, you can create locations for storing data and other research materials early on (Chapter 5). You do not yet need to have data ready to incorporate. You may even change your mind and work with different materials later on, but thinking about data, how they are related to one another and how they will be handled as soon as possible is an important part of research design and software project set-up.

Defining factual features

Factual features are known characteristics about data and respondents (Chapter 12). Depending on your design you may sample on this basis; for example, if conducting a comparative case study in which you are focusing on two or more

organisations, settings or other entities. Alternatively, you may be interested in comparing how individuals with certain socio-demographic characteristics think about, experience or talk about an issue. One-case designs also include comparative elements, although these typically relate to features *within* cases rather than across multiple ones. In addition are analytic facts identified as salient through the processes of interpretation. Factual features often pertain to information which stays constant within a project. However, longitudinal designs include some such features which change over time, and these can constitute core comparative aspects. Either way, these aspects can be well handled in software.

Developing analytical areas of interest

There are many different approaches to qualitative data analysis, some of which we discuss in more detail below, and in Chapter 7. The main focus of this book is on how software packages specifically designed for the purpose may support *your* approach to analysis.¹ As such, this book will not tell you what your approach should be, or what the specific means are by which you will achieve them. Although this chapter discusses some common analytic strategies and the rest of the book discusses how analytic tasks can be supported by CAQDAS packages, this is done in broad terms. You should therefore read this book in tandem with the wider literature on qualitative research and data analysis, if you are not already familiar with it.

‘Analysis’ is often written about or conceived as a discrete *stage* in a research project. This is the result of having to separate phases of work or analytic processes in order to describe and discuss them without causing confusion. In many respects we are doing the same here, in this chapter, and throughout the book. However, conducting research is not a linear, one-directional task (see Figure 2.1; p. 45). The elements that comprise any project are interrelated and fluid. Analysis is not a *stage* of work with clear boundaries. You analyse from the first moments of conceiving the idea of a project, locating it within your ‘world-view’ and formulating the problem through design, data collection and into writing up. Doing a literature review is a form of analysis (Chapter 5). Deciding whether, how and what to transcribe is an analytic act (Chapter 7). Developing a coding scheme (Chapter 9) and linking data and concepts (Chapter 11) are analytic. Writing up an account is a form of analysis (Chapter 10). Designing a research project forces you to be explicit about what you want to analyse and how you intend to do so (see Silver and Lewins, 2014). You will get more out of your project and your use of software tools if you come to software clear in your mind what your analytic strategy is and what processes you need to go through in order to apply it and answer your research questions.

¹There are many other software applications that may facilitate aspects of qualitative data analysis and qualitative research more generally but that were not specifically designed for the purpose. For an overview and discussion of such tools see Silver and Lewins (2013) and Paulus et al. (2013).

CAQDAS packages are project management tools which have potential benefits for all qualitative and mixed methods projects – far and beyond the ways they are most usually described, critiqued and reported. We discuss this further in Chapter 2, in describing our conceptualisation of their role in supporting the many phases of work.

Some basic principles and distinctions

Throughout this book we return to some central ideas which frame the way we perceive analysis and discuss and teach software. In preparing yourself for reading this book, and your general preparation for thinking about and engaging with software the most important of these are the following:

1. Analysis constitutes a series of processes which, although having distinct characteristics in their own right, are fluid and overlapping.
2. Analytic work can be distinguished according to levels of abstraction from data and directions of work, and these are reflected in the way you can work with software.
3. Approaches to analysis are usefully distinguished according to whether they are essentially code-based or non-code-based.
4. Analysis requires the ability to cut through data in different dimensions.

Making reference to the case-study examples (Chapter 2), we illustrate that the use of customised software significantly facilitates these aspects, allowing either the strict adherence to an established and documented analytic strategy, or the creation of one in relation to the specific needs of an individual project.

Analytic processes

Qualitative analysis, rather than being a linear set of procedures which follow on from one another in a logical, one-directional way, is better understood as a *process*. This is not a new idea. Bryman and Burgess noted in the mid-1990s, for example, that the previous decade had seen a number of ‘shifts in emphasis’ in the way research methodology was being discussed, including that ‘stages of social investigation have been replaced with the idea of research as a social process which requires careful scrutiny’ (1994: 1). Since then, many different qualitative approaches have been discussed in similar ways. Software explicitly designed for the purposes of qualitative data analysis reflects and reinforces its non-linear and fluid nature. We prioritise such fluidity in the way we discuss software tools and encourage you to think about and use them. The linear format of this book limits us somewhat in this regard. However, having the idea of *fluidity* in your mind from the outset is important. Refer to our conceptualisation of the ‘core analytic activities’ (Figure 2.1; p. 45) and the sense in which analytic tasks are carried out through the use of software tools iteratively and incrementally. This view of software will set the foundation to enable you to make the most of your chosen package.

Something that has characterised qualitative data analysis generally, and the use of CAQDAS more particularly, is an absence of specific and detailed accounts of *how* analysis proceeds. There is much discussion about qualitative data analysis as a ‘craft’ which has resulted in a sense of *mystery* around what analysts actually do. We illustrate the role of software in ‘opening up’ qualitative data analysis, in making apparent both the *processes* involved, and the impact of technology on methodology. Whatever type of researcher you are, you have a responsibility to document and reflect on what you do, and the impact of your processes on the results. Using CAQDAS will not do this for you, or, in or of itself, make you a ‘better’ researcher. But it will provide you with tools that more directly and immediately allow you to illustrate and justify your analytic strategy. A theme of this book, then, is that CAQDAS provides the tools, but you, the researcher must use them wisely.

Levels and directions of work

Analysis happens at different ‘levels’. In discussing the role of software in analysis we distinguish between four:

Sometimes you work at the *data* level, carefully viewing and reading relevant material and considering its importance in relation to the formulation of the research problem, the existing literature, and the adequacy of the data you have before you. You might make notes (usually called ‘annotations’) about what you see, which help you to identify and reflect upon interesting aspects of the data and decide what to do next, in order to move the analysis forward. You link data segments that appear to be associated.

Much (but not all) qualitative data analysis employs the mechanism of ‘coding’ to organise ideas about what is important in data, in relation to research questions. This might be thought of as the *indexing* level (depending on your analytic approach and methodological context). However described, it refers to the organisation of data according to what you, as the researcher, deem to be interesting in the materials you have before you. In some approaches, this level of work builds on work previously done at the ‘data’ level. Elements of work done at this level of work might also be conceptualised as ‘thematic’ if you consider codes to constitute themes.

If we think of coding (and other means of organising ideas such as linking) data segments as a process of indexing meaning or content; mapping out what is ‘going on’, it follows that at some point we need to ‘move on’ and work at a more *conceptual level*. This often proves challenging when using CAQDAS for the first time; regardless of how the terms ‘codes’, ‘themes’, ‘concepts’ and ‘categories’ are understood and conceived as operating within analysis. Difficulties can arise in doing so for several reasons, not least your confidence in analytic process and confidence in experimenting with software tools such that you can manipulate them to suit your needs. Issues in using the technology can be particularly challenging amongst researchers who are learning about software at the same time as learning about methodology, or amongst experienced analysts who are new to the use of technology for analytic purposes (Silver and Rivers, 2014). This is understandable and to be expected. Yet the tools that allow you to interrogate patterns, relationships and connections within

and between data (Chapter 13) are where the significant potential power of using software, in comparison to working on paper, lie. We hope that this book will give you the confidence to experiment with software tools beyond those that appear most straightforward in the context of your methodological framework, analytic strategy, practical project demands and personal style of working.

The importance of stepping back from data and your developing analysis of them is a recurrent theme in this book. From a process point of view this entails working at a more *abstract* level. Whether your analysis is directly informed by theory (deductive), oriented around theory-building (inductive), or employing elements of both (abductive), creating connections between data, codes and your comments about them is an integral part of analysis. CAQDAS packages provide ‘mapping’, ‘modelling’ or ‘networking’ tools which enable you to interrogate connections according to earlier work, or to create connections according to your current thinking about (sub)sets. However you work, diagrammatic visualisations facilitate thinking in non-linear ways. We see the use of writing tools in tandem with mapping tools as integral to getting the most out of work at the abstract level specifically, and your use of software more generally.

Chapter 7 discusses different approaches to coding, distinguishing between *inductive*, *deductive* and *abductive*. This relates in part to the level at which you start and the direction of analysis. In broad terms, inductive approaches tend to be ‘theory-building’, starting at the data level and working up towards the abstract levels. Deductive approaches are often characterised as ‘theory-testing’ in which the project is driven or informed by an existing theoretical framework which is applied to, or tested on, a new body of data. Abductive approaches combine approaches. This is simplistic of course; rather than seeing different ways of working as mutually exclusive or distinct from one another, analysis often involves working in different ways, at different levels, in different directions, at different times. The use of software facilitates such flexible, iterative and incremental processes.

Code-based and non-code-based approaches

CAQDAS packages largely grew out of social science disciplines and are distinguished from other software tools by the sense in which they offer *qualitative* approaches to qualitative data (see below). Many now offer much more, but given their historical roots and the debates around their use, it is relevant to make some broad comments about their versatility in terms of analytic techniques generally. We do this in part by distinguishing between ‘code-based’ and ‘non-code-based’ approaches.

The packages we discuss in this book all have powerful and sophisticated coding tools. These can be used to facilitate a range of approaches to analysis (Chapter 7). Many methodologies employ coding as a means of organising ideas about data; therefore CAQDAS packages have wide appeal and are extensively used across sectors, disciplines, methodologies and analytic approaches. Amongst early criticisms were contentions that software served to homogenise qualitative data analysis because it prioritised coding, thereby encouraging code-based approaches at the expense of alternative ways of navigating data (Coffey et al., 1996; Lee and Fielding

1996). Although such criticisms have since been superseded by software developments, it is relevant to highlight the role of coding in analysis (whether supported by software or not).

The first point to make is that coding is not *in and of itself* analysis. Whether using software or not, coding is a device to organise ideas about data (Chapters 2 and 7). Coding is a common aspect of analysis and facilitates interpretive thinking. But it is essentially an indexing process, in which you catalogue that a particular segment of data is about, or a general instance of, something in which you are interested. Coding in software has a number of advantages over paper-based coding (Chapters 7–9), but technically a code is simply a position in a database system to which you link data segments. Codes need not, therefore, be conceived of, or used as, *interpretive* devices in the sense in which they are primarily discussed in the literature. They can be used in that way, of course. And many, perhaps most, who use software, do so. But you can use them for a whole range of purposes that transcend customary, methodological ways of working. Thinking about codes and the process of coding in narrow ways might be limiting and homogenising, but that is to do with the user, not the technique or the software itself.

However, some types of associations exist within data for which coding tools are inadequate. A classic example is when a respondent makes a comment which s/he later contradicts. The two statements might be about the same general topic, but they communicate something quite different. Associating them with the same code does not record that level of nuance. Or when reviewing literature, you read something which reminds you of a passage you read in a different article, and you want to note that association. Or in analysing the way a text builds an argument – perhaps in a political speech or other form of public discourse – you want to track how rhetorical or other linguistic devices are used to make a point. These are just some examples of instances in which coding fails to offer sufficient flexibility. Many CAQDAS packages include the additional ability to *hyperlink* between passages of data in addition to coding them (Chapter 6). These devices offer more flexibility in handling non-linear and non-thematic linkages such as those listed above (Silver and Fielding, 2008; Silver and Patashnick, 2011).

Cuts through data

The need to make comparisons *across*, and interrogate patterns and relationships *within* data is inherent to analysis. Distinguishing between the different *cuts* through the dataset you wish to make is a useful device in thinking about how tools might be employed for your analytic needs. Cutting through data 'horizontally' is typically about focusing on one area of interest across all data (or subsets). For example, you might focus on an individual code or theme and retrieve all data segments linked to it, regardless of the data files in which they occur. You are thus able to start thinking about how a particular topic or theme occurs across the whole dataset, to think more analytically and start interrogating more deeply. Cutting through data horizontally in this way allows you to ask questions like: Are the segments coded here

equivalent, or do they need recoding in order to handle peculiarities, contradictions, nuances? Are there some segments which are not adequate instances of the code? Do they need uncoding, reconceptualising, disregarding? How does this sort of retrieval change the way you are thinking about the code? What further questions does cutting through the data in this way raise?

In addition, you may be interested in focusing attention on one particular data file, an interview transcript or set of field notes, for example, and visualising how multiple codes occur in relation to one another, 'vertically', throughout. This might be related to sequence, proximity or embeddedness. You might consider the relative occurrence of all codes, or selectively choose those that you envisage might be related, or will help you identify interesting patterns in the way you have coded. This can lead to further questioning of data, to ascertain, for example, whether a pattern in coding identified in one data file also occurs in other, related data. Language-oriented approaches can benefit in particular from this type of visualisation, or those in which the sequence of code application is of particular interest. For example, in analysing political discourse there may be a focus on how an argument is constructed. Considering the position of codes that capture particular linguistic devices according to the way sentences are structured, for example, would enable a comparison of how different politicians craft an argument. This type of coding can be combined with more content-based, descriptive or thematic types, in order to subsequently investigate whether arguments relating to particular topics are constructed differently. Consideration of the relative position of code occurrence in this way enables you to remain at the data level, although working in this way also allows you to combine the indexing and data levels.

Cutting through data horizontally and vertically need not be distinct activities. Some software packages allow you to visualise both dimensions at the same time. Such concurrent working offers means of 'playing' with data and the connections within in many different ways. You can also combine working with data and codes like this with interrogation on the basis of the factual characteristics pertaining to respondents and data. For example, having sampled for and interviewed respondents with different socio-demographic characteristics (e.g. gender, age, role), you can investigate the differences in how men and women think about, discuss and experience particular topics. This can be done both horizontally and vertically.

Throughout the book we illustrate how different ways of viewing data can facilitate your analysis. These are not restricted to vertical and horizontal cuts, but distinguishing on this basis offers a good starting point for considering how you intend to work with and compare data and explore the patterns and relationships between them.

The rise of qualitative software

Software programs supporting qualitative analysis have a relatively long history. The earliest handling of textual data developed on mainframe computers during the 1960s. These concordance-type tools provided quick listings of word usage, frequency and standardised measurements appropriate for certain types of quantitative

content analysis and other language-oriented approaches. Basic data management systems geared around storing and indexing large volumes of textual information have also been available since the early days of personal computing. There are now many such software programs providing a variety of sophisticated management and linking tools that support the range of digitised information researchers collect to inform their work. The internet 'revolution' has broadened the range of possibilities almost infinitely. However, the tools we discuss in this book, categorised as CAQDAS, developed from the mid-1980s and share particular characteristics arising from their developmental origins.

What types of software do we categorise as CAQDAS?

Tesch (1990) began the process of relating analysis types and software tools. Weitzman and Miles (1995) built on her work, creating a taxonomy of qualitatively oriented software packages. The CAQDAS acronym is well understood across disciplines as broadly referring to software designed to assist the analysis of qualitative data. It was coined by Raymond Lee and Nigel Fielding following the 1989 Surrey Research Methods Conference which first brought together pioneers in the field. The subsequent CAQDAS Networking Project (established in 1994) had the effect of 'fixing' the acronym.²

However, there has been rather a fuzzy conception about which packages CAQDAS includes. We make a broad definition here. Software which falls under the CAQDAS 'umbrella' includes a wide range of packages, but their general principles are concerned with taking a *qualitative* approach to qualitative data. Qualitative data include text, visual and multimedia forms of non-numerical, or unstructured material (Chapter 4). A qualitative approach often includes a need to interpret data through the identification and possibly *coding* of themes, concepts, processes, contexts, etc., in order to build explanations or theories or to test or enlarge a theory (see below and Chapter 7). Qualitative data collection techniques include in-depth interviews, focus groups and participant observation. Approaches to qualitative research and analysis include action research, ethnography, ethnomethodology, hermeneutics and phenomenology. Qualitative analysis strategies include grounded theory, thematic analysis, Framework analysis, conversation and narrative analysis. Different approaches may employ a range of data types and analytic strategies, and the techniques employed and processes followed to undertake analysis cut across approaches and strategies.

The qualitative strategies we refer to are distinct from 'quantitative content analysis' or 'text mining' techniques, in which the statistics of word or phrase frequencies and their occurrence relative to other words or phrases are the basis of analytic work (see Holsti, 1969). We refer to tools that support such approaches in

²The acronym QDAS (qualitative data analysis software) is preferred by some (e.g. di Gregorio and Davidson 2008; Bazeley 2013) but we use CAQDAS because of its historical roots and more general use and acceptance in the field.

varying degrees. Where we include them it is because they have a focus on the qualitative as well. Chapter 3 provides an overview of the packages we are mainly concerned with in this book.

Which is the 'best' CAQDAS package?

This is perhaps the most frequently asked question, yet it is impossible to answer! All the packages we use and teach have tools in common, plus their own distinctive features (Chapter 3). The purpose of software is not to provide a methodological or analytic framework. The tools available support certain tasks differently, and there is some debate about whether individual packages may 'steer' the way analysis is performed. Users should take into account practicalities such as the way software programs are taught, and how much emphasis is placed on an 'ideal' way of using software. To promote only one way of using any package undervalues both the software and the methodological independence of the researcher. Creeping homogeneity helps no one in the long run except the person who is trying to sell you a method or product. As the researcher, you are the expert. You remain in control of the interpretive process and you decide which tools within a software package best facilitate *your* approach to analysis. You also have the responsibility for being transparent about your processes and ensuring the quality of your interpretation (Chapter 10).

Whichever package you choose, you will be able to utilise a selection of tools which will facilitate data management and analysis. Software developments and blurring boundaries mean that tools within a given package may not be appropriate for all qualitative approaches. Equally, just because a function is available does not mean you have to use it. We caution against choosing a package simply because it is the one you have the 'easiest' (e.g. immediate or free) access to, or that seems the most sophisticated. However, if you do not have a choice, you will usually be able to make a package work for you.

Analytic strategies in the context of software use

Methods of data collection cut across methodologies. Approaches may employ broadly interpretive techniques for analysis, but the respective sensitivities and beliefs about the nature of data and knowledge mean that the starting points of interpretation can be quite different. The significance (or not) of language and cultural contexts and also the purposes of research can be so divergent that a sentence of speech or an observed action will have different significance to an ethnographer compared with, for example, a conversation analyst, an ethnomethodologist or a grounded theorist.

Tesch (1990), in rationalising common and differentiating elements in the context of software, distinguished *types* of qualitative research according to where the research interest lies: (i) the characteristics of language; (ii) the discovery of regularities; (iii) the

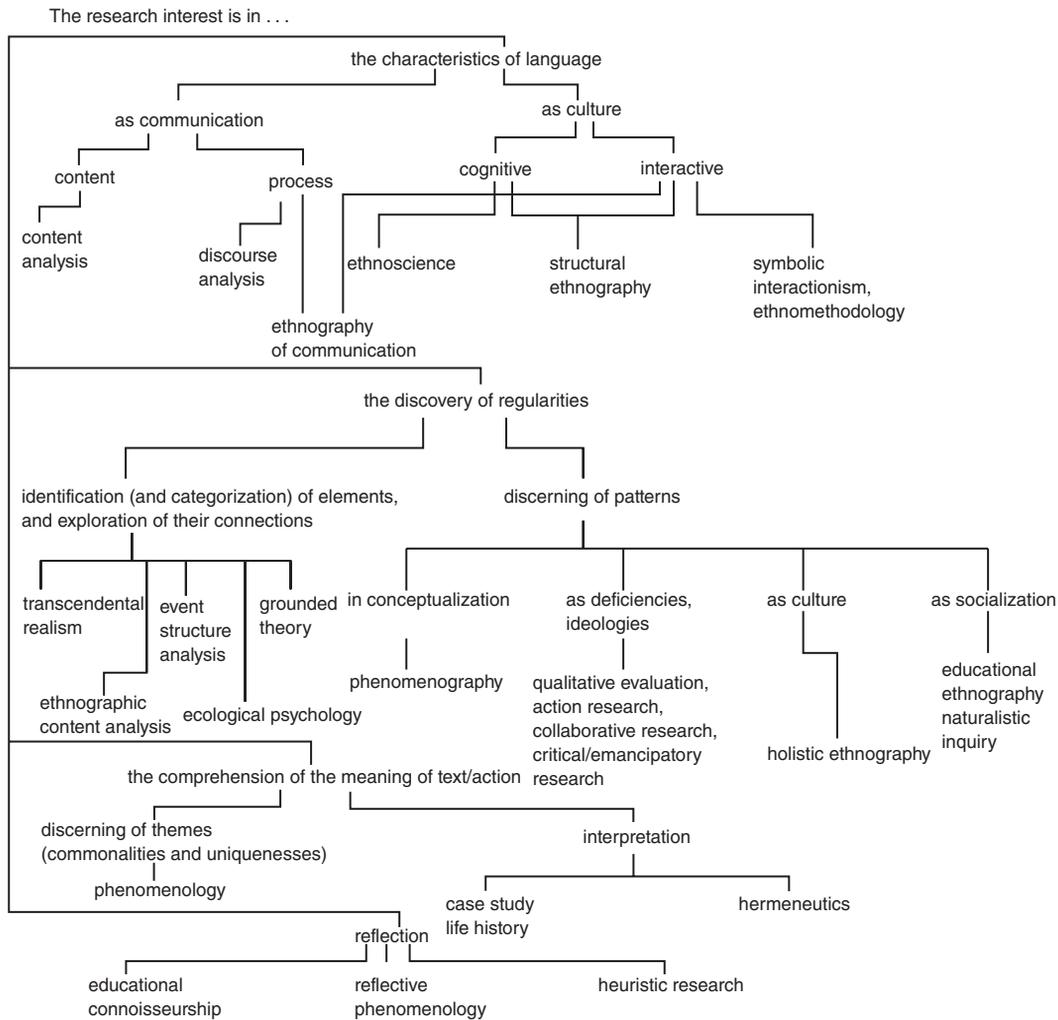


Figure 1.1 Graphic overview of qualitative research types (Tesch, 1990: 72–73)

comprehension of the meaning of text/action; and (iv) reflection (see Figure 1.1). These distinctions reflect overarching philosophical stances and priorities. Although defining qualitative data as ‘any information. . .that is not expressed in numbers’ (1990: 55) Tesch explicitly focused on textual forms in her discussions.

Tesch herself acknowledged difficulties with these classifications, recognising that whilst some are representative of an epistemological stance, others are more about method. Thus at a conceptual level they are not equivalent. Much has changed since she was writing, but Tesch’s work continues to have broad relevance to the intersection between qualitative methodology and technology. It is not within the scope of this book to discuss in detail the range of qualitative research types or systematically illustrate how software may support them. Indeed, our focus is on the qualitative strategies more specifically, and the way software tools can be employed to undertake them.

Therefore, as a lead-in to the task-based chapters that follow, we outline five strategies that can be well supported by CAQDAS packages. These are discourse analysis, narrative inquiry, Framework analysis, grounded theory and thematic analysis. In addition we discuss mixed methods research and visual analysis as broader approaches. Some are more commonly discussed in the context of software than others, and not all are included in Tesch's scheme. Our choices for discussing each in this chapter reflect our observations in one of three respects:

1. they are commonly used by researchers employing CAQDAS;
2. they are well supported by CAQDAS but are infrequently reported upon in the context of software use; and
3. there is significant potential growth areas for qualitative methodology and software in the future.

In discussing these strategies and approaches our aim is to provide a broad introduction. We do not claim to be presenting an exhaustive overview of qualitative research, approaches to analysis or the way software tools are used. However, we add comments regarding the history and context of software use; where appropriate, this includes packages which typically fall outside of the CAQDAS collection. We refer you to specific texts where you can gain more detailed information.

Analysis of discourse

Discourse analysis refers to a broad range of language-based approaches to the analysis of texts that consider the way knowledge is produced and used. This might entail a focus on particular types of discourse (e.g. medical, political, legal); the use of implicit theories to make sense of social action (e.g. economics, power, gender relations); or devices used to structure discourses and their intentions (e.g. rhetoric, linguistic devices, interaction) (Spencer et al., 2003; Silverman, 2001). Researchers across a wide range of disciplines employ variants of discourse analysis to study a multitude of aspects of social life, and in so doing have developed specific strategies. Conversation analysis, Foucauldian discourse analysis (Willig, 2001) and critical discourse analysis are frequently discussed derivatives, but there are many others (for an overview see Gee and Handford, 2012).

Hammersley (2002) in summarising the field, distinguishes between 'types' of discourse analysis in terms of their focus; the sorts of knowledge they claim to make; and in the kinds of technique they employ. Others distinguish more specifically between approaches. Glynos et al. (2009), for example, identify six, highlighting differences and similarities according to the dimensions of ontology, focus and purpose. Dick (2004) in contrast, discusses discourse analysis as a range of approaches, from descriptive variants that aim at understanding conventions such as 'turn-taking' to analytic variants that focus more on generating understandings of the use of language in specific social contexts. Wooffitt (2005) does a good job of demystifying the area, discussing conversation analysis as a key methodological approach to the analysis of verbal interaction, but also outlining distinctive features of various approaches to discourse analysis more

generally, such as discursive psychology, rhetorical psychology, speech act theory, critical discourse analysis and Foucauldian forms of discourse analysis.

A discourse analysis can be conducted on data generated through various methods, including primary forms such as interviews, discussions, life histories and secondary forms such as policy documents, newspaper articles and speeches. Often, relatively small amounts of data and/or numbers of texts are utilised when conducting a discourse analysis, as they are analysed at a very fine level of detail, although with the support of software this need not be the case.

BOX 1.2

FUNCTIONALITY NOTES

Software tools for language-oriented approaches

Language-oriented approaches rely on close consideration of the presence or physical relation of the occurrence of words, phrases and structures. The text-mining type functionality provided by some CAQDAS packages (Chapters 3 and 6) enables such patterns in texts to be reliably found, coded (Chapter 7) and retrieved (Chapter 8), and these are particularly useful for derivatives of discourse analysis. The sophistication of these tools varies quite considerably, so consider options carefully if this type of functionality forms the basis of your work. You might simply need to locate occurrences, mark, write about and output them. You may not code them, but doing so will improve your repeated access to them. Getting the most out of the software might mean coding for certain linguistic devices and coding for the context in which they occur. You can then compare how certain devices are used differently in contextual discourses.

Narrative inquiry, in contrast, is characterised by a focus on the sequencing of textual characteristics rather than data fragmentation or reduction which is inherent in thematic (code-)based approaches. As such, preserving the natural features of texts, including both structural and sequential elements, is paramount and code-based tools may be seen as inappropriate for this task. Some CAQDAS packages provide two-dimensional spaces for 'mapping' connections. These could be useful for approaches needing to create graphic representations of relationships and structures revealed, for example, by conversations in a work or social setting. In narrative analysis, specific methods and formalised traditions vary in terms of how software might help in anything other than improved general management of and access to them. However, hyperlinking devices might provide ways of linking between structures within an account or across several (Chapters 5 and 11). For less structured approaches, for example to life history accounts or the observation of work, links between points in the data may be useful for tracking a chronology or a set of procedures (Chapter 6).

Narrative inquiry

Narrative inquiry is concerned with the structure of accounts, or stories, focusing on how they are constructed, including processes, sequences, intentions and meanings. It is used to investigate experiences, how they are known about, made sense of, and

communicated. Narrative inquiry has increased in popularity in recent decades, frequently attributed to what has been termed the 'linguistic' or 'narrative' turn which is seen to have occurred during the 1990s (Atkinson, 1997; Lieblich et al., 1998; Fenton and Langley, 2011).

Authors differ in the ways they conceptualise the traditions, foci and processes of narrative inquiry and the practicalities of analysis. Bernard and Ryan (2010), for example, identify four major traditions: sociolinguistics, hermeneutics, phenomenology and grounded theory. Daiute and Lightfoot (2004), in contrast, in their edited volume organise the writings of authors from different disciplines according to 'literary readings', 'social-relational readings' and 'readings through the forces of history'. They see these as the three main ways of conceptualising narrative analysis, although they concede that they are neither discrete nor used in the same way by researchers. Lieblich et al. (1998) divide approaches to narrative analysis according to different ways of reading: 'the holistic-content reading'; 'the holistic-form reading'; the 'categorical-content reading'; and the 'categorical-form reading'. Riessman (2005) adds to these by further distinguishing between the performative or dialogical aspect of narrative and visual narratives. These examples of the range of ways of understanding narrative inquiry encapsulate the complexity of the field.

Rather than being a single or uniform approach or method, therefore, narrative analysis is characterised by diversity; not only is it utilised across disciplines and informed by a range of theoretical traditions, but also it constitutes a mix of methodological approaches. The 'texts' that are analysed may be 'naturally occurring' (such as documents generated for other purposes, for example, diaries) or collected through speaking with or interviewing research participants (often, but not exclusively, in the form of 'oral history' type interviewing). Its forms also utilise a variety of analytic strategies, including both quantitative and qualitative practices.

Framework analysis

Framework analysis is a specific method for analysing qualitative datasets. A matrix-based method for ordering and synthesising data, Framework analysis was originally developed during the 1980s at the UK-based National Centre for Social Research and is now a widely used method that supports case-based and thematic approaches to qualitative data analysis. At its core is the idea of a 'thematic framework' which is used to 'classify and organise data according to key themes, concepts and emergent categories' (Ritchie et al., 2003: 220). In contrast to other methods, Framework focuses on the *synthesis* of data, involving the creation of summaries of verbatim data rather than on data reduction activities through the use of coding. Fruber (2010) illustrates a five-phase process involved in undertaking an analysis of pregnant women suffering from mild to moderate psychological distress using the Framework method. This involved phases of (i) data immersion/familiarisation; (ii) developing a theoretical framework; (iii) indexing (coding); (iv) charting (using matrix charts); and (v) synthesising (summarising). Others

break the process down slightly more. Whatever the exact processes involved, and how they progress, the development of summaries is achieved in such a way as to maintain context, language and meaning. Framework analysis has commonalities with other forms of qualitative analysis, including grounded theory and thematic analysis. It is distinct, however, in its focus on summarising and synthesising data and their display and analysis through the use of matrices, rather than on the use of coding as the main basis of analytic work.

BOX 1.3

FUNCTIONALITY NOTES

Software tools for Framework analysis

Having developed it initially for in-house use, NatCen released the FrameWork software in 2009 for sale to the wider academic and applied research community. In 2011, NatCen formed a partnership with QSR International, its functionality was subsumed within NVivo, and FrameWork software taken off the market. There is not a wide literature concerning the framework method, or the degree to which it is facilitated by software applications, and that which does exist tends to originate from NatCen.

If using Framework analysis – or other matrix-based approaches – or indeed, simply needing to summarise or write about what is seen in a general sense, the bespoke summary-writing spaces provided by some packages will work well (Chapter 10). But you can always approximate the functionality through the use of standard memo-writing spaces where the specific tools are not available. You may do this before, after, or to the exclusion of coding. Comparison is likely inherent to these approaches, and factual data organisation (Chapter 12) will enable you to later interrogate writing and coding accordingly (Chapter 13).

Grounded theory

Originated by Glaser and Strauss (1967), grounded theory is a well-known and frequently discussed form of qualitative research. It comprises a methodological approach rather than simply being an analytic or coding strategy. Since the first descriptions there have been many adaptations. In *The Discovery of Grounded Theory* (1967), Glaser and Strauss created an organised and interactive approach to the collection and analysis of data, using what they called the ‘constant comparative method’. The history of their co-operation is interesting and relevant to subsequent modifications. Strauss was instrumental in the development of the Doctorate of Nursing Science (DNS) at the University of California at San Francisco, and both he and Glaser had a shared research interest in chronic illness and dying. The development of grounded theory was in part pragmatic, arising from a need to create a text for DNS students that systematised a way of dealing with qualitative data. It was also a response to perceptions that qualitative data analysis had somehow lost its empirical connection to data in an over-preoccupation with theory. The text included in it new routines but also documented techniques

already in use by the Chicago School. There was little writing of methodological texts up to that point (Morse et al., 2009). Personal descriptions in Morse et al. (2009) reveal how these creative moments in qualitative methodological history occurred and how a range of the ‘second generation’ of grounded theorists applied their own modifications to substantive research projects. Glaser continues to support and reinforce the original principles of the 1967 text and to stress inductive ‘emergence’ of codes and categories and theory-free starting points.

BOX 1.4

ANALYTIC NOTES

Features of grounded theory (1967)

At the heart of the original grounded theory method was a basic principle, the ‘constant comparative method’. Its main features are as follows:

- A coding process (later to become known as *open coding*) consists of annotations in the margin expressed as codes based on social constructs or on the respondent’s own language (later labelled *in vivo* codes).
- Data segments are compared, thus refining ideas about this and subsequent categories.
- Memos are an important aspect, and should be kept updated about the development of each category.
- Collecting, coding and analysing data should occur concurrently; thus ongoing ‘theoretical sampling’ of data is performed to enable further comparisons to be made of different groups and settings.
- Categories are further refined and relationships among them identified.
- Categories are reduced to smaller set of more abstract higher-level concepts – allowing the possibility of generality or the production of formal theory.
- The collection of more data retains the principle of being grounded in the data and permits further incidents to be analysed in the light of these concepts – allowing the modification of these concepts.
- When concepts are not being modified any further, categories are said to be *theoretically saturated*. Theoretical saturation means that the analysis of more incidents is not adding further to ideas, it merely ‘adds bulk to coded data and nothing to theory’.

Amongst the second generation who influenced later strands of grounded theory were several DNS students and associated postdoctoral researchers. We focus on two of these, since they possibly comprise the most influential developments. Firstly, Strauss with Juliet Corbin diverged from the original work and wrote *Basics of Qualitative Research* in 1990, developing grounded theory to such an extent that Glaser questioned whether it had any relationship to the original and suggested it was effectively another qualitative method. Corbin collaborated for 16 years with Anselm Strauss until his death, and continues to apply and adjust grounded theory to current substantive contexts and at a practical level to computer usage, having included reference to MAXQDA

in the 3rd edition of *Basics of Qualitative Research*, published in 2008. Secondly, Kathy Charmaz, in many publications between 1975 and 2002 and with Anthony Bryant (Bryant and Charmaz, 2007) consolidated a constructivist version of grounded theory that accounts for beliefs about the relative, context-laden nature of interpretation. Although maintaining many of the features espoused by Glaser, Strauss and Corbin, constructivist grounded theory attends also to the active role of the researcher in research generally, and particularly in analytic processes; the interplay between researcher and data that results in the use of codes, development of categories and the theoretical account. Thus the effect of prior knowledge and existing literature, as well as the issue of reflexivity, are highlighted. Charmaz makes the point that grounded theory is primarily *a way of thinking about data* and, as such, cannot be standardised. Constructivist grounded theory attends to the ways in which theoretical development is tied to the engagement with epistemological issues, and as such illustrates the role of both deduction and induction in the analytic process (Charmaz, 2006).

BOX 1.5

FUNCTIONALITY NOTES

Software tools for theory-building approaches

Theory-building approaches are characterised by the need to move beyond description through writing (Chapters 6 and 10) and indexing of data through basic coding (Chapter 7) to generate themes, concepts or categories (Chapters 8 and 9). Grounded theory and thematic analysis are examples of theory-building approaches. CAQDAS packages support processes involved in generating theory very well, and you will find most of the tools discussed in this book useful at various moments.

Working deductively, you will likely have a theoretical framework at the outset which can be represented in the software as a map, model or network (Chapter 11). Working inductively, you will be working towards generating such a visual representation of your interpretation or theory. Software facilitates either approach or a combination of both (Chapter 7). You will need to record the factual features of data and respondents (Chapter 12) in order to make comparisons within and between cases (Chapter 13). You will need to write about your processes and analytic insights in identifying patterns and relationships and developing and testing theories.

Thematic analysis

Thematic analysis is a commonly used approach to the analysis of qualitative data, yet is relatively infrequently described or discussed in specific terms. Although those who write about it often understand it to constitute a method of analysis in and of itself (Braun and Clarke, 2013; Fereday and Muir Cochrane, 2006; Attride-Stirling, 2001), its techniques are used in many other approaches, and therefore its status is debated. Outside of these debates, it can be seen as constituting a set of analytic processes applicable in a variety of theoretical contexts, disciplines and topics of investigation (Boyatzis, 1998). This applicability is seen to relate to its inherent flexibility as well

as its independence from theory and epistemology, as contrasted to other approaches such as conversation analysis, interpretive phenomenological analysis and grounded theory which have clear roots in particular traditions (Braun and Clarke, 2006). Indeed, elements inherent to thematic analysis are also evident in these approaches, and as such thematic analysis is applicable to theoretically driven research and more applied approaches. It might be argued that thematic analysis is the definitive ‘code-based’ approach in the sense that it entails a process of *encoding* qualitative information (Fereday and Muir Cochrane, 2006). In attending to an identified absence in the literature concerning specific procedures for conducting thematic analysis, Braun and Clarke (2006) propose a six-phase guide, involving (i) familiarising yourself with data; (ii) generating initial codes; (iii) searching for themes; (iv) reviewing themes; (v) defining and naming themes; and (vi) producing the report. They differentiate ‘types’ of thematic analysis in terms of its form and outcome, on a number of levels:

- Aim of the analysis: whether to develop a rich description of the dataset, or a detailed account of one particular aspect.
- Identification of themes: whether inductive or theoretical.
- ‘Level’ of themes: semantic and latent themes.
- Epistemological underpinnings: essentialist/realist vs constructionist.
- Types of questions being asked: research questions, questions asked of respondents where primary data are collected, questions which guide coding and analysis.

Mixed methods research

Mixed methods is a vast and varied field in social science methodology which, although with a long history in terms of the utilisation of more than one method within a given project (Hesse-Biber, 2010), has been much debated in recent years (Ivankova and Kawamura, 2010). Indeed, it is only seen to have *formally* existed as a field for 10–15 years (Teddlie and Tashakkori, 2012). Technological capabilities afforded by CAQDAS packages have the potential to play an important role in the continued growth of mixed methods, although the majority of discussions in the literature remain concerned with research design rather than the role of software (notable exceptions being Bazeley, 2006; 2011; Fielding, 2012; Kuckartz, 2012).

Mixed methods research involves the use of more than one type of method within a research project. That may involve mixing quantitative methods, mixing qualitative methods or mixing qualitative *and* quantitative methods. The latter type has come to the fore in methodological discussions and is often what is implied by the use of the general term. These approaches involve the collection, analysis and integration of both quantitative and qualitative data within a single study or as part of a longer-term strategy across multiple studies (Creswell, 2003; Kelle, 2006). In considering the appropriateness of employing a mixed methods approach, authors distinguish between paradigmatic, pragmatic and political (or transformative) rationales (Brannen, 2005; Creswell et al., 2011).

Mixed methods approaches are seen as transcending traditional paradigmatic debates between quantitative and qualitative approaches, and have thus been described as constituting a 'third paradigm' (Tashakkori and Teddlie, 2003). Indeed, one of the rationales for their use is the idea that employing and making explicit different philosophical positions is valuable to social science research (Greene, 2007). But there is huge variety in the way projects are designed, and importantly, what is being mixed, why and the stage(s) at which mixing occurs. Some authors therefore question the utility of conceptualising mixed methods in paradigmatic terms (Bazeley, 2009; Mertens and Hesse-Biber, 2012). Nevertheless, utilising both quantitative and qualitative methods is widely seen as enabling the benefits of each to be realised at the same time as minimising their limitations. More pragmatic rationales for the use of mixed methods are thereby discussed in terms of employing methods which best suit the nature of the problem under study. Giving primacy to the importance of the research question and valuing both objective and subjective knowledge are key aspects in employing methods according to 'what works' (Morgan, 2007). This may include the use of multiple researchers in collaborative projects as well as multiple methods. Political or transformative approaches emphasise the role of mixed methods research in improving society in some way (Brannen, 2005; Mertens, 2009; Mertens and Hesse-Biber 2013).

Whatever the rationale for employing mixed methods approaches, terminology is an issue in getting to grips with the literature. 'Mixed methods research' is perhaps the most widely used term to refer to the general field, but others are also employed, including 'mixed research' and 'multiple methods'. Once a close reading of the literature begins, it quickly becomes apparent that, similarly to qualitative research, particular terms are used in quite different ways across contexts. This can be confusing to the novice researcher. Authors develop increasingly specific and nuanced terms as they attempt to differentiate their conceptualisations from those of others. This is seen starkly amongst those who distinguish between types of mixed methods through developing research design categorisation systems, and several authors have called for more consistency in terms used (Bryman, 2008; Johnson, Onwuegbuzie, and Turner, 2007; Tashakkori and Teddlie, 2010). It is not within the scope of this book to discuss the field in detail or to make any further attempts to unravel its complexities; our focus is on the role of software in supporting analytic strategies rather than in rationales or designs *per se*.

BOX 1.6

FUNCTIONALITY NOTES

Use of software for mixed methods

In the context of the use of CAQDAS packages, it is the task of mixing analytic techniques which is relevant, whatever the types of data or design being employed. This can mean employing a quantitative approach to qualitative data, a mixed approach to qualitative data, or a mixed approach to mixed data. Your analytic design will affect the software tools you use.

(Continued)

(Continued)

Software can enable quantitative information about qualitative materials to be imported and linked (Chapter 12). If you conduct coding in a particular way you can count the occurrence of certain features, thereby quantifying qualitative data (Chapters 7 and 8). You can import mixed data in the form of spreadsheets (Chapter 4). You can transform codes into categorical variables. You can export summary frequency information pertaining to qualitative coding to conduct statistical analyses (Chapter 13). Alongside these options you will use many of the other tools depending on the specific analytic design.

Whatever the emphasis, approach or design, conducting mixed methods research is much more than simply taking the 'best' from quantitative and qualitative methods and combining them; and the variety and debate in the literature is testament to this (Bergman, 2008). Indeed, as Bryman (2008) cautions, mixed methods projects are subject to methods-related shortcomings just like those originating within either paradigm, despite often being presented as a means of overcoming them. Considering the role of software in mixed methods research requires, however, moving away from the specifics of research design towards the practicalities and procedures involved in the analysis of data.

Visual analysis

The analysis of still and moving images has a long history, with their use in disciplines such as anthropology and management studies pre-dating the formalisation of visual sociology as a discipline, which occurred from the 1970s (see Schnettler and Raab, 2008, for a historical overview). The use of visual records in empirical research, however, has been advanced by the rise of digital technology and their use is now widespread across academic and applied disciplines. A range of specific theories concerning the use of visual records and strategies for their analysis have developed, although often in isolation from similar work in other disciplines (Hindmarsh, 2008). In addition, existing methodological approaches and analytic strategies have been applied to and adapted for the analysis of visual records. Examples include interaction analysis (Jordan and Henderson, 1995; Heath et al., 2010), visual ethnography (Pink, 2007), visual grounded theory (Konecki, 2011) and visual semiotics.

Approaches to visual analysis are informed by the role and analytic status of data within a given project as well as the methodological and disciplinary traditions which inform design (Silver and Patashnick, 2011). In considering visual analysis in the context of software use for supporting analysis it is useful to draw a number of distinctions:

- whether visual data are being used primarily for illustrative purposes or are construed as data sources in their own right;
- whether visual data are the only or primary data sources or are being used in combination with other data sources;

- whether still or moving images are being used, or both;
- whether the subject of analysis is the content of visual records, interaction contained within them, or indeed both;
- whether the approach to analysis is quantitative, qualitative, or mixed.

Making distinctions on these levels is not to say that approaches or techniques are, or need to be, mutually exclusive in this regard, just that considering these aspects at the outset of designing a project and in reading and evaluating research outputs is useful.

Notwithstanding these distinctions, the emphasis of visual research, where the focus is on some aspect of human behaviour or conduct, is frequently on the micro-analysis of sequences of interaction. This speaks to the affordances of the medium in capturing 'naturally occurring' behaviours, the complexity and multimodality of visual data, in particular video, and the work involved in analysing such material from a practical point of view. It also raises the issue of the means through which visual records are analysed, which necessitates the drawing of a further distinction: whether data are analysed directly, or indirectly through the use of a written transcript (Silver and Patashnick, 2011), leading to the consideration of the role of technology in visual analysis strategies in more detail.

BOX 1.7

FUNCTIONALITY NOTES

Software tools for visual analysis

Working with visual data is very different from working with textual forms. You will first have to decide whether to work directly with the visual media, or indirectly via a written representation. Working directly, annotations, memos and codes are the key tools you will use to record your ideas about what is in the data. Working indirectly, the development of the transcript will constitute an intermediary analytic task. Visual analysis can employ various analytic strategies. Refer therefore to discussions about the other approaches discussed here in reflecting on which software tools will enable your analytic needs to be achieved, appropriately within your research design. That will also be affected by the status of the visual within the larger project; that is, whether it is the main or only form of data, or integrated with or supplementary to other forms.

Writing about and coding visual data may pertain to verbal content and/or non-verbal interaction. Where both are of analytic interest you will need to be particularly systematic in the use of tools for specific purposes. The amount of visual data you have and your analytic focus will affect the reliance on factual data organisation (Chapter 12) and interrogation tools (Chapter 13).

Concluding remarks: a critical yet flexible approach

We encourage the view that you as the researcher draw on elements of methodology and methods to provide the framework of your analytic strategy. You might

be wedded to a particular approach and apply it through your use of software. You might draw on the principles and methods of more than one methodology, and, through your use of software, develop your own analytic strategy, specific to the needs of your project. However you work, throughout you must be thoughtful and transparent about your own role and beliefs (ontology and epistemology) and also be in tune with how working contexts impact on the way you analyse. 'Working contexts' include the use of software – a set of tools which at one level are just like the pencil or highlighter, simply enabling different ways of looking at, and cutting through, data. But software tools provide more potential for flexibility, access and thoroughness than their 'manual' or 'paper' counterparts. Nevertheless, they have to be used competently and appropriately.

You are responsible for ensuring the processes you go through are rigorous and the findings you report are true to your data. Never do something *just* because it is possible. The commercial context within which software packages are developed is worth remembering. Software companies need to make a profit, and although most are still true to their academic roots and the needs of researchers, commercial pressures mean that to a degree they all need to try and meet the needs of a range of researchers. Just like you do not need all the tools of your chosen word-processing application, you will not need all the tools of your chosen CAQDAS package for an individual analysis. We encourage experimentation, but always do so within the boundaries of your methodological requirements. Try to avoid being distracted by fancy or complex options unless they will actually help to achieve an analytic task. The impact of technology on methodology is exciting, but never let it distract you from the ultimate aim of your engagement with software: you need complete your project, therefore you need to focus on the means of achieving your ends. As such, consider how to adopt a critical yet flexible approach in planning for and actually using software.