

How to process qualitative data?

- Data usually in the form of text, video, or other media

① Break data into pieces.

each piece is one independent

thought / step / episode / intent / idea

Label each piece

User #, Session #, Piece #

② Assign each piece a "code"

(Label or Concept)

1. open coding - induce category names as you go.

2. fixed coding - start with a fixed set of category names
deductive

3. stare at ~~data~~ and then come up with a ~~fixed coding scheme~~ - bleah :-)

A General Way to Code Data

- (but it is incomplete)

- ① start with an uncategorized piece
assign it a category
- ② find all other pieces in the
same category
- ③ Repeat ①-② until all data in a
category.
- ④ Organize categories into themes.
- ⑤ Organize themes into
theoretical constructs
(together form a theory)

⑥ Continue research
by deliberately choosing
samples with a good chance
of confirming, deepening, or
disconfirming your theory
→ THEORETICAL SAMPLING

⑦ Code new data using existing
categories as a starting point
(can add/modify categories as
needed) — a little deductive

Grounded Theory

by ~~Glaser~~ and Strauss 1960s

a theory that was induced

(grounded in the data)

→ something that accounts for a pattern of behavior

Strauss - style grounded theory is very systematic:

- ① constant comparison
- ② coding paradigm
- ③ theoretical sampling

Constant Comparison

- ① Start by comparing one data piece to others. What words are the same? Different?
- ② Begin to assign codes to data pieces.
(concept names)
- ③ As codes are assigned, compare each new data piece to all others with the same code (concept)
Begin to craft definitions for each code.

- ④ Now compare new data pieces
to the corresponding code definition.
→ sharpen definition of the
code
- ⑤ Build categories out of concepts.
(themes)

Example of Constant Comparison

from Corbin and Strauss 3rd ed.

Basics of Qualitative Research.

Field Notes ① and ② : on next page

COMPARISON:

① is about PLACEMENT.

② is about LOSS

Last sentence of ① suggests relationship between PLACEMENT and LOSS.

Later on, compare field notes marked PLACEMENT with each other to sharpen your definition of a code

Field Note ①

It was a very difficult decision to put my husband in a nursing home. But I'm 85 and I was unable to care for him physically and emotionally. He seemed alright but 6 months after entering the nursing home he died, and so now I regret the decision.

Field Note ②

I feel so much loss at the death of my husband. Even when he was in the nursing home, I knew he was there. Now I'm alone.

Coding Paradigm

Coding is not just the identification of a one-phrase code.

Strauss recommends defining

a code in detail and also noting

① causal condition: *when is this relevant?*

what caused this to occur?

② interactions among actors

③ strategies

④ consequences

Often ①-④ are pointers to other codes data pieces,

FIRST-CUT
CODE DEFINITION, from Field Note ①

PLACEMENT : putting someone somewhere

causal condition:

existing living condition not
sustainable due to CARETAKING
requirements

interactions: 1. CARETAKING provider

unable to continue due to
advanced age

2. Wife has AUTHORITY to put
patient somewhere

Strategies: Wife puts husband in
nursing home

consequences: LOSS six months later
(as perceived by wife)

As more field notes get labelled

PLACEMENT, the definition of the code is improved.

- add more options to each of the 4 elements of the coding paradigm.

We also define properties of a code.

Each property has dimensions.

(a range of values it can take on)

Possible props/dims for PLACEMENT:

Properties	Dimensions
1. Decision	(Easy \longleftrightarrow Difficult)
2. Duration	Short \longleftrightarrow Long
3. Options	(Home Alone, Home w/ Relative, Home w/ Visiting Nurse) Nursing Home

Another Example of a Well-Defined Code

READING SOURCE

- cause: new CHECK-IN,
been away a few days,

DEBUGGING

- interactions: author stopping by
to notify of changes (NOTIFICATION)
- strategies:
 - start at main,
follow control flow
 - read newly edited first
 - compare diffs w/ Tool (TOOL USE)
- consequences:
 - aware of additional changes to
be made (AWARENESS)
 - QUESTION-ASKING to other programmer
 - didn't find answer

READING SOURCE defn, cont.

properties with dimensions:

	props	dimensional range
Reading Code	frequency	often — never
	concentration	hi — lo
	duration	long — short

Common properties of interest

- ① Actor(s) or Subject(s)
- ② Emotions Expressed
- ③ Time
- ④ Location

Grounded Theory Process is Iterative

- continually review code assignments, code definitions, coding paradigm, properties and dimensions.
- may review notes from one category at a time (axial coding) to sharpen definition of that category

At some point,
code definitions stop changing.

You can choose a CORE CATEGORY now
— and only assign codes from
here on out that relate to the
core category

(relate if part of
core category or
part of connected
category)

example: choose to focus on

interactions with source

(reading/editing/building)

as CORE,

drop attention to effect of
desk layout.

MEMOS

- ① relationships between concepts,
between categories
- ② influence of your personal experience or personal study
- ③ anytime you are thinking instead of coding.