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. *Strike* 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.

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.
   \[\rightarrow\] 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.
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.

We also define **properties** of a code.

Each property has **dimensions**.

(a range of values it can take on)

<table>
<thead>
<tr>
<th>Properties</th>
<th>Dimensions</th>
<th>from other notes</th>
<th>from ①</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Decision</td>
<td>(Easy ← ③) Difficult</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Duration</td>
<td>Short (←) Long )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Options</td>
<td>(Home Alone , Home w/ Relative, Home w/ Visiting Nurse) Nursing Home</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 programmers
  - didn't find answer
**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 \text{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 \text{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.