Constructing Kirq, software for set-theoretic social research: A software development travelogue

Claude Rubinson
University of Houston—Downtown
rubinsonc@uhd.edu
cjr@grundrisse.org
http://grundrisse.org

PyHou
Houston, Texas
March 19, 2013
Overview

- Introduction to QCA
- History of QCA software
- First (mis-)steps in developing Kirq: the fsQCA package for R
- Use cases and design goals for acq and Kirq
- Python's role in meeting these design goals
- Lessons learned: Using Python for academic software projects
What is Qualitative Comparative Analysis?

• A method of conducting social research by analyzing subset relationships, using Boolean algebra
What is Qualitative Comparative Analysis?

- A method of conducting social research by analyzing subset relationships, using Boolean algebra.
- Example: Religious fundamentalists tend to be politically conservative.
What is Qualitative Comparative Analysis?

- A method of conducting social research by analyzing subset relationships, using Boolean algebra
- Example: Wealthy individuals tend to come from privileged families.
What is Qualitative Comparative Analysis?

• Particularly concerned with two types of causal relationships: necessary conditions and sufficient conditions
What is Qualitative Comparative Analysis?

- Necessary condition: cause must be present for outcome to occur
- Example: Must be exposed to HIV to contract AIDS
What is Qualitative Comparative Analysis?

- Sufficient condition: if cause occurs, outcome will occur
- Example: Abortion or miscarriage will terminate pregnancy
What is Qualitative Comparative Analysis?

- Sufficient condition: if cause occurs, outcome will occur

Recently deported women who do not plan to cross again (Outcome)

High SES women who haven't lived in the U.S. and aren't traveling with family ($X_1$)

High SES women who haven't lived in the U.S., have only attempted cross a few times and felt that their last crossing experience was very dangerous ($X_2$)
What is Qualitative Comparative Analysis?

- **Sufficient condition**: if cause occurs, outcome will occur

Recently deported women who do not plan to cross again (Outcome)

High SES women who haven't lived in the U.S. and aren't traveling with family ($X_1$)

Women belonging to sets $X_1$ and $X_2$

High SES women who haven't lived in the U.S., have only attempted to cross a few times and felt that their last crossing experience was very dangerous ($X_2$)
What is Qualitative Comparative Analysis?

• Challenges conventional statistical analysis, which is based upon a linear-additive model

• Complements other set-theoretic research methods (e.g., SNA and QNA)

• Does not depend upon degrees of freedom, so is useful for small-, medium-, and large-N studies

• Encourages research process that is “retroductive” and “case-oriented”
What is Qualitative Comparative Analysis?  
Example: Brown and Boswell (1995)

Truth Table with Contradiction (from Table 4 of Brown and Boswell 1995)

<table>
<thead>
<tr>
<th>Recent Black Migrants</th>
<th>Weak Union</th>
<th>Black Strikebreaking</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>East Chicago, Pittsburgh, Youngstown</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>Con</td>
<td>Buffalo, Chicago, Gary, Johnstown, [Cleveland]</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>Bethlehem, Joliet, McKeesport, Milwaukee, New Castle, Reading</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Decatur, Wheeling</td>
</tr>
</tbody>
</table>
**What is Qualitative Comparative Analysis?**

**Example: Brown and Boswell (1995)**

Revised Truth Table without Contradiction (from Table 5 of Brown and Boswell 1995)

<table>
<thead>
<tr>
<th>Recent Black Migration</th>
<th>Weak Union</th>
<th>Local Govt Repression</th>
<th>Black Strikebreaking</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>East Chicago, Pittsburgh, Youngstown</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>—</td>
<td>Buffalo, Chicago, Gary, Johnstown</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Cleveland</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>Bethlehem, Joliet, McKeesport, New Castle, Reading</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Milwaukee</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>Decatur</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Wheeling</td>
</tr>
</tbody>
</table>
What is Qualitative Comparative Analysis?  
Example: Brown and Boswell (1995)

Revised Truth Table without Contradiction (from Table 5 of Brown and Boswell 1995)

<table>
<thead>
<tr>
<th>Recent Black Migration</th>
<th>Weak Union</th>
<th>Local Govt Repression</th>
<th>Black Strikebreaking</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>East Chicago, Pittsburgh, Youngstown</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Buffalo, Chicago, Gary, Johnstown</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Cleveland</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>Bethlehem, Joliet, McKeesport, New Castle, Reading</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Milwaukee</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>Decatur</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Wheeling</td>
</tr>
</tbody>
</table>
What is Qualitative Comparative Analysis?
Example: Brown and Boswell (1995)

Revised Truth Table without Contradiction (from Table 5 of Brown and Boswell 1995)

<table>
<thead>
<tr>
<th>Recent Black Migration</th>
<th>Weak Union</th>
<th>Local Govt Repression</th>
<th>Black Strikebreaking</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>East Chicago, Pittsburgh, Youngstown</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Buffalo, Chicago, Gary, Johnstown</td>
</tr>
</tbody>
</table>

RBM * WU * LGR +
RBM * ~WU * LGR = Black Strikebreaking
What is Qualitative Comparative Analysis?  
Example: Brown and Boswell (1995)

Revised Truth Table without Contradiction (from Table 5 of Brown and Boswell 1995)

<table>
<thead>
<tr>
<th>Recent Black Migration</th>
<th>Weak Union</th>
<th>Local Govt Repression</th>
<th>Black Strikebreaking</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>East Chicago, Pittsburgh, Youngstown</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Buffalo, Chicago, Gary, Johnstown</td>
</tr>
</tbody>
</table>

\[
\text{RBM} \times \text{WU} \times \text{LGR} + \\
\text{RBM} \times \sim\text{WU} \times \text{LGR} = \text{Black Strikebreaking}
\]

\[
\text{RBM} \times \text{LGR} = \text{Black Strikebreaking}
\]
Technical and Usability Challenges

• QCA algorithms are:
  – NP-hard (no exact algebraic solution)
  – $O(2^N)$ complexity, where $N$ indicates the number of variables (not observations) in the data set

• Because data sets tend to be small and matrix algebra isn't used, no use for NumPy

• How to maintain and encourage retroductive, case-oriented research process?

• How to make software that's efficient, useful, and usable?
Lineage of QCA Software

QCA
(Drass and Ragin 1992)

TOSMANA
(Cronqvist 2011)

QCA (R)
(Dusa and Thiem 2012)

QCA3 (R)
(Huang 2012)

fs/QCA
(Ragin, Drass, and Davey 2009)

fuzzy (Stata)
(Longest and Vaisey 2008)

acq & Kirq
(Rubinson and Reichert 2012)
“Plan to throw one away; you will, anyhow”

- fsQCA module for R
  - Cross-platform, but requires R
  - Not user-friendly
  - Too slow
  - R programming “considered harmful”
  - But: allowed me to realize that the user interface should be task-oriented
“Plan to throw one away; you will, anyhow”

- **fsQCA module for R**
  - Cross-platform, but requires R
  - Not user-friendly
  - Too slow
  - R programming “considered harmful”
  - But: allowed me to realize that the user interface should be task-oriented
“Plan to throw one away; you will, anyhow”

- fsQCA module for R
  - Cross-platform, but requires R
  - Not user-friendly
  - **Too slow**
  - R programming “considered harmful”
  - But: allowed me to realize that the user interface should be task-oriented
“Plan to throw one away; you will, anyhow”

- fsQCA module for R
  - Cross-platform, but requires R
  - Not user-friendly
  - Too slow
  - **R programming “considered harmful”**
  - But: allowed me to realize that the user interface should be task-oriented
“Plan to throw one away; you will, anyhow”

- fsQCA module for R
  - Cross-platform, but requires R
  - Not user-friendly
  - Too slow
  - R programming “considered harmful”
  - **But: allowed me to realize that the user interface should be task-oriented**
Use Cases for acq and Kirq

• **acq**: QCA at the Unix commandline
  – a “scratch my own itch” project

• **Kirq**: QCA for everybody else
  – a user-friendly, crossplatform GUI program
Design Goals for acq and Kirq

- Software that is efficient, useful, and usable:
  - Conform to the Unix philosophy
  - Good “out of the box” performance, plus ability to optimize performance
  - Support and encourage good QCA research practices
  - Kirq should be crossplatform and user-friendly

- Also important: Avoid sucking up all of my time
Why Python?

• The surrounding ecosystem
  – Ability to hire others
  – Confidence that the supporting environments is stable and will continue to be maintained
  – Python is *lingua franca* in academia
  – Rich environment for GUI toolkits, installers, etc.
  – Chose Qt for GUI toolkit and PyInstaller for installer
Design Goal: Avoiding a time sink

- Relatively easy to recruit and hire good programmers
- Easy to mix procedural and OOP programming
- Official online documentation remains top notch
- The core Python language remains relatively compact
  - but not the standard library, and certainly not the surrounding environment (PyPI, etc)
Design Goal: Conform to the Unix Philosophy

- Build a prototypes as soon as possible
- Small is beautiful/do one thing well
  - acq's GUI scripts: gtt and concov
  - have resisted adding a data editor to Kirq
  - decided to offload Kirq's visualizations
- Make every program a filter
  - Because Kirq can read data from the commandline, it's easy for other programs to call out to it
Design Goal: Good “out of the box” performance and ability to optimize

- acq had fewer lines of code than fsQCA for R module, and was faster
  - compare to QCA module for R
- Good tools for profiling
- Some standard, well documented practices for improving performance, although Python optimization often requires expertise
- Promise of projects such as PyPy
Design Goal: Support and encourage good QCA research practices

- Less concern for performance means more attention to user-interface issues
- Writing acq as Unix shell scripts helped me streamline the QCA analysis; both acq and Kirq make it easy to modify and rerun analyses
- Have designed Kirq to facilitate interrogation and comparisons of solutions
- Lots of GUI niceties, such as tooltips and pop-out windows
- Importance of “eating your own dogfood”
Design Goal: Kirq should be cross-platform and user-friendly

- Standardized on Python 2.7, for PyQt
- Only minor compatibility issues with PyQt bindings and OSX, and none with Windows – Kirq always feels native
- Could never build Qt for OSX; used MacPorts instead (slow, but works well)
- PyInstaller works really well, but needed to use development branch for OSX (dev branch is now stable)
- Session history is Kirq's killer feature
<table>
<thead>
<tr>
<th>City</th>
<th>M</th>
<th>U</th>
<th>R</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bethlehem</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Buffalo</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Chicago</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Cleveland</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Decatur</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>East_Chicago</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Gary</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Johnstown</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Joliet</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>McKeesport</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Milwaukee</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>New_Castle</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Pittsburgh</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Reading</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Wheeling</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Youngstown</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Row</td>
<td>M</td>
<td>U</td>
<td>N</td>
<td>Consist</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>---------</td>
</tr>
<tr>
<td>1</td>
<td>True</td>
<td>True</td>
<td>3</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>True</td>
<td>False</td>
<td>5</td>
<td>0.80</td>
</tr>
<tr>
<td>3</td>
<td>False</td>
<td>True</td>
<td>6</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>False</td>
<td>False</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td>Term</td>
<td>Consist</td>
<td>RawCov</td>
<td>UniqCov</td>
<td>ObsConsist</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>--------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>M*R</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>Buffalo;Chicago;Gary;Johnstown;East_Chicago;Pittsburgh;Youngstown</td>
</tr>
</tbody>
</table>

Solution:

| Solution | 1.00 | 1.00 | NA | NA | NA |
Lessons Learned – Python's Advantages (for Academic Projects)

- Core language is relatively compact, with excellent documentation
- Relatively easy to find developers
- Strong, well-developed environment of GUI toolkits, installers, etc.
- Good performance out of the box, with ability to optimize if necessary
Lessons Learned – Python's Disadvantages (for Academic Projects)

- Package distribution is a mess, as is associated documentation
- Churn in the standard library is too rapid to keep up with for a part-time developer
- Dead tree documentation is lousy
- Online signal-to-noise ratio is low
- Python community online is too insular; overly concerned with “idiomatic Python”