

# ShinyQDA

## Qualitative Data Analysis with R Shiny

**Jason Bryer, Ph.D.**

Assistant Professor and Associate Director

CUNY SPS Data Science and Information Systems

April 11, 2025



# Overview

shinQDA is designed to work with a collection of documents, or texts. It was originally designed to help us analyze essays from the **Diagnostic Assessment and Achievement of College Skills (DAACS)**. For DAACS, students are asked to write a minimum of a 350 word essay where they summarize their self-regulated learning results and identify strategies they will use to be successful students.

The core data entry features include:

- Selecting parts of the text to tag with codes.
- Provide scaffolded questions about the highlighted text.
- Enter codings and comments at the text level (i.e. the entire document).
- Score the text using a pre-defined rubric.

Analysis features include:

- Sentiment analysis both at the individual text level as well as across the entire corpus.
- Tokenization
- Word frequencies
- Code frequencies
- Co-occurrence plots
- Inter-rater reliability
- Topic modeling



# Demo



# Creating a new project

You can download `shinyQDA` from [Github](#) using the `remotes` package. Note the addition of the `dependencies` parameter to ensure all the suggested packages are installed too (this includes among other things, the packages to do sentiment analysis).

```
remotes::install_github('jbryer/ShinyQDA', dependencies = c('Depends', 'Imports', 'Suggests'))
```

We will initialize a new ShinyQDA project using the `daacs_data` included in the package. Change that with your dataset and ensure that `id_column` and `text_column` correspond to your data frame.

```
data("daacs_data", package = 'ShinyQDA')
ShinyQDA::new_app(name = 'daacs_demo',
                  dir = getwd(),
                  qda_data = daacs_data,
                  id_column = 'id',
                  text_column = 'qda_text',
                  initialize_sentiment_dictionaries = TRUE,
                  run_app = FALSE)
```



# Adding a rubric

Currently rubrics can only be added through the command line. The contents of the rubric (i.e. each cell) can be edited afterwards.

```
data("daacs_rubric", package = 'ShinyQDA')
daacs_qda$add_rubric(
  rubric_name = 'daacs',
  description = 'DAACS Scoring Rubric',
  rubric = daacs_rubric)
```



# The qda object

The `shinyQDA` package uses a SQLite database to store all the data. You can use any SQLite viewer application to explore (e.g. [DB Browser for SQLite](#), [RSQLite](#) R package). The Shiny application also has a "raw" view. However I recommend all interactions (i.e. data read and write) occur through the [qda](#) function.

```
daacs_qda <- ShinyQDA::qda(paste0(find.package('ShinyQDA'), '/daacs_demo/qda.sqlite'))
ls(daacs_qda)
```

```
## [1] "add_code_question"           "add_code_question_response"   "add_codes"
## [4] "add_coding"                  "add_rubric"                  "add_rubric_response"
## [7] "add_text"                   "add_text_question"           "add_text_question_response"
## [10] "db_conn"                    "db_file"                     "delete_code_question"
## [13] "delete_code_question_responses" "delete_coding"               "delete_rubric_response"
## [16] "delete_text_question"        "delete_text_question_responses" "get_assignments"
## [19] "get_code_question_responses" "get_code_questions"          "get_coders"
## [22] "get_codes"                  "get_codings"                 "get_last_update"
## [25] "get_log"                    "get_rubric"                  "get_rubric_responses"
## [28] "get_rubrics"                "get_text"                     "get_text_question_responses"
## [31] "get_text_questions"          "log"                         "methods_docs"
## [34] "update_code"                "update_code_question_response" "update_coding"
## [37] "update_rubric_criteria"     "update_text_question_response" "users_passphrase"
```



# Run the app

You can run the application locally.

```
shiny:::runApp('daacs_demo')
```

Or you can deploy to a Shiny server.

- shinyQDA will not work with [shinyapps.io](#) since that service will not preserve the data between sessions.
- DigitalOcean is a relatively affordable option (the 2GB / 2CPUs tier is usually sufficient and is currently \$18 per month). [Click here for installation directions](#)
- Share using Dropbox (or similar). Be warned that more than one person working at a time could cause conflicts in the data file.



# Setting up the codebook.

Now you can setup the codes by clicking Setup -> Codebook.

The screenshot shows the ShinyQDA web application interface. At the top, there are three colored window control buttons (red, yellow, green). The title bar displays the URL <http://127.0.0.1:6361>, the page title [~/Dropbox \(SPS\)/ShinyQDA/inst/daacs\\_demo - Shiny](#), and a [Open in Browser](#) button. On the right, there is a [Publish](#) button with a dropdown arrow. Below the title bar, the navigation menu includes links for [ShinyQDA](#), [Data](#), [Coding](#), [Setup](#) (which is currently selected), [Analysis](#), and [My Info](#).

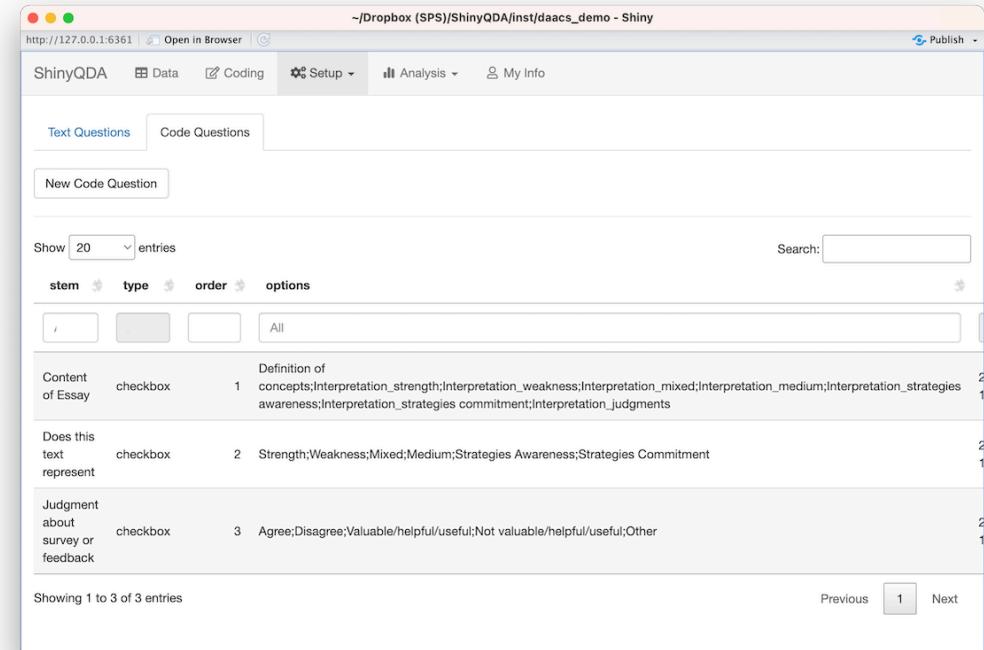
The main content area is divided into two sections. The left section contains buttons for [Add Code](#), [Collapse All](#), and [Expand All](#). Below these buttons is a hierarchical tree view of codes. The root node, [metacognition](#), is expanded and highlighted with a blue background. Its children are [planning](#), [monitoring](#), and [evaluation](#). Below this is another collapsed node, [motivation](#), which has children [mindset](#), [test anxiety](#), and [mastery orientation](#). At the bottom of the tree view is another collapsed node, [self efficacy](#), which has children [self efficacy for online learning](#), [self efficacy for writing](#), and [self efficacy for mathematics](#).

The right section displays details for the selected code, [metacognition](#). It includes fields for **Code:** (containing "metacognition"), **Description:** (containing "NA"), and **Color:** (containing the hex code "#8DD3C7"). A small icon in the bottom right corner depicts a person standing in front of a screen displaying a large letter "R".



# Setting up code questions

Clicking Setup -> Questions will allow you to add specific questions (beyond the codes) that can be applied to either the entire text (i.e. each question can be answered once for each text document) or highlighted text.



The screenshot shows the ShinyQDA software interface with the 'Code Questions' tab selected. The main area displays a table of three entries:

Content of Essay	checkbox	Definition of concepts;Interpretation_strength;Interpretation_weakness;Interpretation_mixed;Interpretation_medium;Interpretation_strategies awareness;Interpretation_strategies commitment;Interpretation_judgments
Does this text represent	checkbox	Strength;Weakness;Mixed;Medium;Strategies Awareness;Strategies Commitment
Judgment about survey or feedback	checkbox	Agree;Disagree;Valuable/helpful/useful;Not valuable/helpful/useful;Other

Below the table, it says "Showing 1 to 3 of 3 entries". At the bottom right, there are "Previous" and "Next" buttons, with the number "1" indicating the current page.



# Now you can get coding...

ShinyQDA

Text Coding

Text Data

Data

My Info

## Select text:

259: According to the responses I received from the Self-Regulate...

Code Editor

[View All Codes](#)

[Add Code](#)

### Codes assigned to this text

metacognition

### Additional comments about the text

This is a good essay!

According to the responses I received from the Self-Regulated Learning Assessment, I possess many – if not all – the skills necessary for successful study and learning at the college level.

To begin, I scored strongly with regard to metacognition. When I first saw the word, I was curious about the precise meaning of this neologism, wondering even whether it was accurate. I asked myself, “What kind of thinking is ‘beyond thought’?” Later, when I read the description given in the evaluation results, I was pleased to see that “metacognition” signifies reflection upon one’s own thinking: a king of “thinking thought.”

In fact, such self-reflection has always been a strong personal characteristic. As a young college student, a friend introduced me to the PQRST method of study: or, preview-question-read-study-test. I quickly mastered it and began applying it to my work, with great success. Thus, the essential elements of metacognitive thought – planning, monitoring, and evaluating – have been an integral part of my conscious thought process for many years.

Regarding strategies, meanwhile, I see that I could improve my use of certain tools to increase the efficacy of my work. Although I have set up my home office in an ideal way and have a keen sense of time management, I nonetheless could be more regular in seeking assistance, particularly on the technical level. Accordingly, I have made a resolution to that effect.

Finally, I am highly motivated, in all aspects of my life. I strongly believe in the efficient potential of a positive mindset, and I daily practice mindfulness exercises. Because my goals are clearly defined and divided into achievable steps, I am efficient regarding time and energy, measuring both according to the task at hand. When that involves a test or examination, I am more efficient, rising to the challenge.



# Moving forward...

If you encounter any issues or want to make a feature request you can do so on Github here:  
<https://github.com/jbryer/ShinyQDA/issues>

Here are some things I'm working on...

- Increase performance, in particular the topic modeling module.
- Incorporate LLM for summarizing text(s).
- Regression model (for predicting an outcome).
- Ability to add documents in the Shiny application.
- Add/edit rubrics from the Shiny application.



# Questions?



# Thank You!

✉ jason.bryer@cuny.edu

⌚ @jbryer

Ⓜ️ @jbryer@vis.social

🔗 jbryer.github.io/ShinyQDA/

linkedin.com/in/jasonbryer/

