Package: qcoder (via r-universe)

October 29, 2026

Title Lightweight Qualitative Coding

Version 0.1.0

Description A free, lightweight, open source option for analyzing text-based qualitative data. Enables analysis of interview transcripts, observation notes, memos, and other sources. Supports the work of social scientists, historians, humanists, and other researchers who use qualitative methods. Addresses the unique challenges faced in analyzing qualitative data analysis. Provides opportunities for researchers who otherwise might not develop software to build software development skills.

Depends R (>= 3.4.0)

License GPL-3 + file LICENSE

Encoding UTF-8

LazyData true

Imports assertthat, magrittr, readr, shiny, stringi, stringr

Suggests knitr, rmarkdown, testthat (>= 2.0.0), withr, covr, pander, here, textreadr, dplyr, DT, shinyAce (>= 0.4.0), shinyFiles, shinyjs, shinythemes, zip (>= 2.0.3)

URL https://github.com/ropenscilabs/qcoder

BugReports https://github.com/ropenscilabs/qcoder/issues

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Collate 'read_in_files.R' 'parse_qcodes.R' 'qcoder-package.R' 'utils.R' 'selection.R'

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libx11-dev zlib1g-dev

Repository https://ropensci.r-universe.dev

Date/Publication 2026-01-13 17:44:11 UTC

RemoteUrl https://github.com/ropenscilabs/qcoder

RemoteRef HEAD

RemoteSha b08f531d4288d489268c598b39fcdcfe3eb92146

Contents

add_code . . . 2
add_codes_to_selection . . . 3
add_discovered_code . . . 3
add_new_documents . . . 4
add_unit . . . 4
build_paths . . . 5
create_empty_code_file . . . 6
create_empty_docs_file . . . 6
create_empty_unit_doc_file . . . 7
create_empty_units_file . . . 7
create_qcoder_project . . . 8
do_update_document . . . 8
error_check . . . 9
get_codes . . . 10
import_project_data . . . 10
parse_one_document . . . 11
parse_qcodes . . . 11
parse_splititem . . . 12
qcode . . . 13
qcode_custom . . . 13
read_code_data . . . 14
read_data . . . 14
read_documents_data . . . 15
read_unit_data . . . 16
read_unit_document_map_data . . . 16
txt2html . . . 17
update_links . . . 18
validate_project . . . 18
validate_project_files . . . 19

Index **20**

`add_code` *Add code Append a new unit record to the existing data frame*

Description

Add code Append a new unit record to the existing data frame

Usage

```
add_code(codes_df, new_code, new_code_desc, codes_df_path)
```

Arguments

codes_df	Existing codes data frame
new_code	text name of a new code (single name only)
new_code_desc	text description of the code
codes_df_path	full path to the codes data frame

Examples

```
unlink("./_my_qcoder_project", recursive=TRUE)
```

---

add_codes_to_selection *Adds codes surrounding the selected text*

Description

Adds codes surrounding the selected text

Usage

```
add_codes_to_selection(selection, codes)
```

Arguments

selection	The selection of text to be coded
codes	The code or codes to be added to the document

---

add_discovered_code *Update codes data frame Add discovered codes to the codes data frame*

Description

Update codes data frame Add discovered codes to the codes data frame

Usage

```
add_discovered_code(
  codes_list = "",
  code_data_frame = NULL,
  codes_df_path = ""
)
```

Arguments

codes_list	A list of codes (usually from a coded document)
code_data_frame	Existing data frame of QCODE codes
codes_df_path	The path where the updated code data frame should be saved

---

add_new_documents *Add new documents Adds new document or documents to an existing documents data frame.*

Description

Add new documents Adds new document or documents to an existing documents data frame.

Usage

```
add_new_documents(files, docs_df_path = "", file_path = "")
```

Arguments

files	Vector of new files to be added
docs_df_path	Path to existing data frame of text documents
file_path	Full path to the data set of documents including trailing slash

Examples

```
create_qcoder_project(project_name = "my_qcoder_project", sample = TRUE)

unlink("./my_qcoder_project", recursive=TRUE)
```

---

add_unit *Add unit Append a new unit record to the existing data frame*

Description

Add unit Append a new unit record to the existing data frame

Usage

```
add_unit(units_df, new_unit, units_df_path)
```

Arguments

units_df	Existing units data frame
new_unit	text name of a new unit (single name only)
units_df_path	full path to the units data frame

Examples

```
unlink("./_my_qcoder_project", recursive=TRUE)
```

---

build_paths *Build the paths for file creation*

Description

Builds the paths to the data to be imported and to the data frame where the imported data is to be stored. The project name is required, all other parameters may be set. These each represent a segment of the path to a file. If a project path is not set or set to "" it will be set to the current working directory via getwd(). The Shiny qcode application assumes that the data frame folder will be "data_frames" and that any new documents to be imported will be in a folder called "documents".

Usage

```
build_paths(
  project_name,
  data_path = "",
  data_frame_name = "",
  df_path = "data_frames",
  project_path = ""
)
```

Arguments

project_name	Name of the project
data_path	Path segment to the data. The format for this may depend on the function using the paths.
data_frame_name	Name of the data frame that will contain the data
df_path	path segment(s) to the created data frame file from the project path
project_path	Path to the project (not including project_name). This will be set to getwd() if a value of "" is passed in.

Value

A named list of paths. "data_frame_path" is the path to the data frame and "data" is the path to the data.

create_empty_code_file

Create an empty codes data set

Description

Used to create a codes data frame with no data but that can have data added. File is placed in the data_frames folder.

Usage

```
create_empty_code_file(path)
```

Arguments

path	Full path to data frame to be created.

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project")
path <- file.path(getwd(),
  "_my_qcoder_project/data_frames/qcoder_codes__my_qcoder_project")
create_empty_docs_file(path)
unlink("./_my_qcoder_project", recursive=TRUE)
```

create_empty_docs_file

Create an empty documents data set

Description

Used to create a codes data frame with no data but that can have data added. File is placed in the data_frames folder.

Usage

```
create_empty_docs_file(path)
```

Arguments

path	Full path to data frame to be created.

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project")
path <- file.path(getwd(),
  "_my_qcoder_project/data_frames/qcoder_docs__my_qcoder_project")
create_empty_docs_file(path)
unlink("./_my_qcoder_project", recursive=TRUE)
```

---

create_empty_unit_doc_file

Define an empty many to many unit to document map

Description

Define an empty many to many unit to document map

Usage

```
create_empty_unit_doc_file(path)
```

Arguments

path	Full path to data frame to be created.

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project")
path <- file.path(getwd(),
  "_my_qcoder_project/data_frames/qcoder_units_document_map__my_qcoder_project")
create_empty_docs_file(path)
unlink("./_my_qcoder_project", recursive=TRUE)
```

---

create_empty_units_file

Define an empty units data frame

Description

Define an empty units data frame

Usage

```
create_empty_units_file(path)
```

Arguments

path	Full path to data frame to be created.

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project")
path <- file.path(getwd(),
  "_my_qcoder_project/data_frames/qcoder_units__my_qcoder_project")
create_empty_docs_file(path)
unlink("./_my_qcoder_project", recursive=TRUE)
```

`create_qcoder_project` *Create a standard set of folders for a QCoder project*

Description

Create a standard set of folders for a QCoder project

Usage

```
create_qcoder_project(project_name, sample = FALSE)
```

Arguments

`project_name`	A string project name to be located in the current working directory or a path to a project folder.
`sample`	Logical that indicates that the sample data should be copied to the project.

Examples

```
create_qcoder_project(project_name = "my_qcoder_project")
unlink("./my_qcoder_project", recursive=TRUE)
```

`do_update_document` *Update document Updates the text field of the documents data frame, typically after pressing Save button in the Shiny App. May also be used in the console.*

Description

Update document Updates the text field of the documents data frame, typically after pressing Save button in the Shiny App. May also be used in the console.

Usage

```
do_update_document(updated, docs_df_path, this_doc_path)
```

Arguments

updated	The updated text as a character string
docs_df_path	Location of the documents rds file.
this_doc_path	Name of record to be updated, as recorded in "doc_path" field of data frame.

Examples

```
unlink("./_my_qcoder_project", recursive=TRUE)
```

---

error_check *Check for coding errors*

Description

Checks the current document for coding errors.

Usage

```
error_check(document)
```

Arguments

document	A string to be scanned for errors.

Details

This function takes a string (such as the contents of a document), and conducts some basic linting. It returns a warning if there aren't a matching number of (QCODE) tags, or if text has been marked to be captured but the capture is missing a tag (missing {#my_tag}).

Value

A warning message as a character string.

Examples

```
error_check("An (QCODE)unmatched set of (QCODE) gives (/QCODE){#tag} a warning.")
error_check("A (QCODE) qcode with a missing tag gives a warning.")
```

`get_codes` *Extract codes from text Take coded text and extract the codes, assuming they are correctly formatted.*

Description

Extract codes from text Take coded text and extract the codes, assuming they are correctly formatted.

Usage

```
get_codes(doc_text)
```

Arguments

`doc_text`	The text data for a single document

Examples

```
unlink("./my_qcoder_project", recursive=TRUE)
```

`import_project_data` *Read data into a project Convenience method to read raw data from standard locations and using standard names in a project folder structure.*

Description

Read data into a project Convenience method to read raw data from standard locations and using standard names in a project folder structure.

Usage

```
import_project_data(project_name)
```

Arguments

`project_name`	The project name. This should represent the folder holding the project.

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project", sample = TRUE)
import_project_data("_my_qcoder_project")
unlink("./_my_qcoder_project", recursive=TRUE)
```

parse_one_document *Parse one document*

Description

Parse one document

Usage

```
parse_one_document(doc, df, qcoder_documents, dots = NULL)
```

Arguments

doc	A single document from qcoder_data
df	The data frame that will contain the parsed data
qcoder_documents	The full documents data frame
dots	Other parameters that may be passed in.

Examples

```
unlink("./my_qcoder_project", recursive=TRUE)
```

parse_qcodes *Parse coded text*

Description

Take a data frame of coded text documents and return a data frame of the codes captured within.

Usage

```
parse_qcodes(x, ...)
```

Arguments

x	A data frame containing the text to be coded; requires columns "doc_id" and "document_text"
...	Other parameters optionally passed in

Details

This function takes a text document containing coded text of the form:

```
"stuff to ignore (QCODE) coded text we care about (/QCODE){#my_code}
more stuff to ignore"
```

and turns it into a data frame with one row per coded item, of the form: `docid,qcode,text`

`parse_qcodes` assumes that it is being passed a data frame, the `parse_one_document` function is called to do the heavy lifting extracting the coded text from the `document_text` column.

Newline characters are replaced with an HTML <br> in the captured text.

If no valid qcodes are found, `parse_qcodes` returns an empty data frame (no rows).

Value

If the data frame contains coded text in the `document_text` column, output will be a data frame with three columns: "doc", "qcode", and "text".

```
The "doc" column is the corresponding "doc_id" value from the input
data frame.

"qcode" is the code that the captured text was marked up with.

"text" is the text that was captured.
```

Examples

```
my_documents <- data.frame("doc_id" = 1, "document_text" =
"(QCODE)Parts of above text is licensed {CC BY-SA 4.0}.(/QCODE){#license} Credit to SRCCON.
","doc_path" = "mypath.txt")
parse_qcodes(my_documents)
```

---

parse_splititem *Parse a single item within a document*

Description

Parse a single item within a document

Usage

```
parse_splititem(splititem, df, doc_id, dots)
```

Arguments

splititem	String usually generated by parse_one_document()
df	Data frame for storing parsed data
doc_id	The doc_id for the document this string is part of.
dots	List of additional options passed in.

---

qcode *This launches the coder Shiny app*

Description

This launches the coder Shiny app

Usage

```
qcode(use_wd = TRUE)
```

Arguments

use_wd	Whether or not the current working directory when launching qcoder should be used as the base from which the project file is selected.

Examples

```
if (interactive()) {
 qcode()
}
```

---

qcode_custom *This launches the coder custom Shiny app*

Description

This launches the coder custom Shiny app

Usage

```
qcode_custom()
```

Examples

```
if (interactive()) {
  qcode_custom()
}
```

read_code_data *Create a file of codes from csv file Use this if you have a spreadsheet of codes already created.*

Description

Create a file of codes from csv file Use this if you have a spreadsheet of codes already created.

Usage

```
read_code_data(
  project_name,
  data_path = "codes/codes.csv",
  df_path = "data_frames",
  data_frame_name = "qcoder_codes",
  project_path = ""
)
```

Arguments

project_name	Name of the project, which matches folder name
data_path	Path to a file containing code data in csv.
df_path	Full path to the codes data frame.
data_frame_name	The name of the RDS file that the data frame will be stored in.
project_path	Full path to the project folder

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project", sample = TRUE)
read_code_data(project_name = "_my_qcoder_project")
unlink("./_my_qcoder_project", recursive=TRUE)
```

read_data *This launches the data-reader Shiny app*

Description

This launches the data-reader Shiny app

Usage

```
read_data()
```

Examples

```
 ## Not run:
 read_data()

## End(Not run)
```

`read_documents_data` *Create a data frame of documents*

Description

Create a data frame of documents

Usage

```
read_documents_data(
  project_name,
  data_path = "documents/",
  df_path = "data_frames",
  data_frame_name = "qcoder_documents",
  project_path = ""
)
```

Arguments

`project_name`	Name of the Qcoder project
`data_path`	path to a folder contain text files to be analyzed.
`df_path`	Full path to the docs data frame.
`data_frame_name`	The name of the RDS file that the data frame will be stored in.
`project_path`	Full path to the project folder.

Examples

```
 ## Not run:
read_documents_data("_my_qcoder_project")

## End(Not run)
```

`read_unit_data` *Create a data frame of units from csv file Use this is you have a spreadsheet of units already created.*

Description

Create a data frame of units from csv file Use this is you have a spreadsheet of units already created.

Usage

```
read_unit_data(
  data_path = "units/units.csv",
  data_frame_name = "qcoder_units",
  project_name,
  project_path = "",
  df_path = "data_frames"
)
```

Arguments

`data_path`	path to a file containing unit data in csv.
`data_frame_name`	The name of the RDS file that the data frame will be stored in.
`project_name`	Name of project if available
`project_path`	Full path to the project folder.
`df_path`	Full path to the units data frame.

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project", sample = TRUE)
read_unit_data(project_name = "_my_qcoder_project")
unlink("./_my_qcoder_project", recursive=TRUE)
```

`read_unit_document_map_data` *Create a data frame of unit to document links from csv file Use this is you have a spreadsheet already created.*

Description

Create a data frame of unit to document links from csv file Use this is you have a spreadsheet already created.

Usage

```
read_unit_document_map_data(
  project_name,
  data_path = "units/unit_document_map.csv",
  data_frame_name = "qcoder_unit_document_map",
  project_path = "",
  df_path = "data_frames"
)
```

Arguments

project_name	Name of project
data_path	path to a file containing unit document map data in csv.
data_frame_name	The name of the RDS file that the data frame will be stored in.
project_path	Full path to the project folder
df_path	Full path to the documents data frame.

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project", sample = TRUE)
project_name = "_my_qcoder_project"
read_unit_document_map_data( project_name = "_my_qcoder_project")
unlink("./_my_qcoder_project", recursive=TRUE)
```

---

txt2html *Format text as HTML Minimal conversion of a text to html*

Description

Format text as HTML Minimal conversion of a text to html

Usage

```
txt2html(text)
```

Arguments

text	text to be converted

Examples

```
txt2html("The quick brown (QCODE)fox(/QCODE){#animal} jumped over ")
```

update_links *Update document to unit links Saves or updates the links between observation units and documents*

Description

Update document to unit links Saves or updates the links between observation units and documents

Usage

```
update_links(
  checked = "",
  docs_df_path = "",
  this_doc_path = "",
  units_docs_path = ""
)
```

Arguments

checked	vector of new or updated links
docs_df_path	full path to document dataset
this_doc_path	value of doc_path for the document
units_docs_path	full path of the data frame of unit to docs links

Examples

```
unlink("./_my_qcoder_project", recursive=TRUE)
```

validate_project *Check for a valid qcoder project*

Description

Check for a valid qcoder project

Usage

```
validate_project(path_to_test)
```

Arguments

path_to_test	Path to possible project folder

Value

NULL for valid project, Error otherwise.

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project")
validate_project("_my_qcoder_project")
unlink("./_my_qcoder_project", recursive=TRUE)
```

`validate_project_files` *Check for required imported data frames.*

Description

Check for required imported data frames.

Usage

```
validate_project_files(path_to_test)
```

Arguments

`path_to_test`	Path to possible project folder

Value

NULL for valid project, Error otherwise.

Examples

```
create_qcoder_project(project_name = "_my_qcoder_project", sample = TRUE)
import_project_data("_my_qcoder_project")
validate_project_files("_my_qcoder_project")
unlink("./_my_qcoder_project", recursive=TRUE)
```

Index

add_code, 2
add_codes_to_selection, 3
add_discovered_code, 3
add_new_documents, 4
add_unit, 4

build_paths, 5

create_empty_code_file, 6
create_empty_docs_file, 6
create_empty_unit_doc_file, 7
create_empty_units_file, 7
create_qcoder_project, 8

do_update_document, 8

error_check, 9

get_codes, 10

import_project_data, 10

parse_one_document, 11, *12*
parse_qcodes, 11
parse_splititem, 12

qcode, 13
qcode_custom, 13

read_code_data, 14
read_data, 14
read_documents_data, 15
read_unit_data, 16
read_unit_document_map_data, 16

txt2html, 17

update_links, 18

validate_project, 18
validate_project_files, 19