Package 'novelqualcodes'

May 9, 2026

Type Package

Title Visualise the Path to a Stopping Point in Qualitative Interviews Based on Novel Codes

Version 0.13.5

URL https://github.com/DesiQuintans/novelqualcodes

BugReports https://github.com/DesiQuintans/novelqualcodes/issues

Maintainer Desi Quintans <science@desiquintans.com>

Description In semi-structured interviews that use the 'framework' method, it is not always clear how refinements to interview questions affect the decision of when to stop interviews. The trend of 'novel' and 'duplicate' interview codes (novel codes are information that other interviewees have not previously mentioned) provides insight into the richness of qualitative information. This package provides tools to visualise when refinements occur and how that affects the trends of novel and duplicate codes. These visualisations, when used progressively as new interviews are finished, can help the researcher to decide on a stopping point for their interviews. For context, see Wong et al., (2023) <doi:10.1177/16094069231220773>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.4.0)

Imports readxl, naturalsort, ggplot2, ggpattern, utils

RoxygenNote 7.3.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Kam Wong [aut, cph] (ORCID: <https://orcid.org/0000-0003-1898-7678>), Desi Quintans [cre, aut, cph] (ORCID: <https://orcid.org/0000-0003-3356-0293>)

Repository CRAN

Date/Publication 2025-10-30 23:50:09 UTC

Contents

`create_field_notes_template` *Create a template for refinement field notes*

Description

It is good practice to record when refinements were made, what was done, and the reasons behind them. These "field notes" are used by this package to annotate plots created by `plot_novelty()` and `plot_richness()`, and they're also requested by peer reviewers as part of the publication process.

This template contains 5 columns (only the first one, `interview_num`, is currently used by the package, but this is subject to change):

1. `interview_num` is the upcoming interview number where these refinements will take effect.
2. `refinement_type` is free text describing the kind of refinement, e.g. "add" or "rephrase" or "remove".
3. `refinement` is the actual text that has been changed.
4. `reason` describes the rationale behind the refinement.
5. `other` is for additional information you may want to include.

Usage

```
create_field_notes_template(path = stop("A save path must be specified."))

create_fieldnotes_template(path = stop("A save path must be specified."))
```

Arguments

`path`	(Character) The path where the field notes template should be created.

Value

Invisibly returns the path to the template (Character). In an interactive session, also opens `path` in the system's file viewer.

Examples

```
# Create the template in a temporary directory.
create_field_notes_template(path = tempdir())
```

`import_coding_matrices` *Read exported NVivo Coding Matrices from a folder*

Description

Coding matrices are built inside NVivo's *Matrix Coding Query* tool, with codes as rows and one participant ("case") as column. These files should be exported as Excel spreadsheets (XLS or XLSX format), which is the default for NVivo. There must only be one participant per file.

Filenames **must** reflect the chronological order of interviews when they are sorted. You can do this by naming them in sequence like *"Interview 07 PID 2345"*, or by including a YMD HM timestamp like *"2023-06-17 1345"*. Sorting is number-aware and only uses the filename itself (i.e. file path is ignored during sorting).

Usage

```
import_coding_matrices(path, recursive = FALSE)
```

Arguments

`path`	(Character) Path to a folder that contains coding matrices exported from NVivo (*Explore* then *Matrix Coding Query* then *Export Coding Matrix*). All files with *.XLS* or *.XLSX* extensions will be imported.
`recursive`	(Logical) If `TRUE`, also imports files inside subfolders of `path`.

Value

A list of dataframes.

See Also

`score_codes()`, `import_field_notes()`

Examples

```
# A folder of example coding matrices included with the package
path_to_matrices <- system.file("insect_study/matrices/", package = "novelqualcodes")
print(path_to_matrices)

# A list of files in that folder
list.files(path_to_matrices)
```

```
# Import them all at once
my_matrices <- import_coding_matrices(path_to_matrices)

# Look inside the result; each entry of 'my_matrices' is an interview, listed
# in chronological order.
print(my_matrices)
```

`import_field_notes` *Import field notes from an Excel spreadsheet*

Description

'Field notes' in this context is a spreadsheet that records the refinements that a researcher makes throughout their interview process. This package is opinionated about what these field notes should look like: use `create_field_notes_template()` to get a template for what the package accepts.

Usage

```
import_field_notes(path, ...)

import_fieldnotes(path, ...)
```

Arguments

`path`	(Character) The full path (including filename) to the Excel spreadsheet.
`...`	Other named arguments that will be passed to `readxl::read_excel()`.

Value

A named list of class `field_notes`.

Examples

```
# An example field notes spreadsheet included with the package.
path_to_notes <- system.file("insect_study/records/refinements.xlsx", package = "novelqualcodes")
print(path_to_notes)

# Importing the spreadsheet
my_refinements <- import_field_notes(path_to_notes)

# Looking at its contents
str(my_refinements, max.level = 1)
print(my_refinements$df)
```

plot_novelty *Plot novelty of interviews over time*

Description

Novel codes are information that has not been previously mentioned by other interviewees. The trend of 'novel' interview codes provides insight into the richness of qualitative information.

This plot shows the trend of novel code generation; in the most basic way, the higher the number, the richer the information that has been generated in the study. By showing novel codes in context with any refinements to the questions, it also shows how that trend may have been affected by those refinements, and whether novel code generation is plateauing.

This chart alone should not be used to decide on a stopping point because it does not show the richness of individual interviews; some interviews are richer than others, therefore consider also using `plot_richness()` to look at the richness of each interview in terms of novel and duplicate codes.

Usage

```
plot_novelty(
  score_df,
  refinements = integer(0),
  col = list(stroke = "black", fill_ref = "black", fill = "grey80")
)
```

Arguments

`score_df`	(Dataframe) A dataframe of scored codes, as generated by `score_codes()`.
`refinements`	Either a list object generated by `import_field_notes()`, or an Integer vector that lists when (in terms of interview sequence) refinements were made to the interview questions. For example, `c(10, 15)` means that interview questions were revised twice: First **before** the 10th interview, and then again **before** the 15th interview.
`col`	(List) A List containing named Character vectors. Accepted names are: • `stroke` is the colour of point outlines as well as the line linking points together. • `fill_ref` is the colour of points after a refinement. • `fill` is the fill colour of points were no refinements were made.

Value

A ggplot object.

See Also

`score_codes()`, `import_field_notes()`, `plot_richness()`, `save_last_plot()`

Examples

```
# Field notes and coding matrices included with the package
path_to_notes   <- system.file("insect_study/records/refinements.xlsx", package = "novelqualcodes")
path_to_matrices <- system.file("insect_study/matrices/", package = "novelqualcodes")

# Import the data
my_refinements <- import_field_notes(path_to_notes)
my_matrices    <- import_coding_matrices(path_to_matrices)

# Score novel and duplicate codes
my_scores <- score_codes(my_matrices)

# Generate a plot with no refinements
plot_novelty(score_df = my_scores)

# Generate a plot using scored codes and imported refinements
plot_novelty(score_df = my_scores, refinements = my_refinements)

# Generate a plot using scored codes and a vector of refinement times
plot_novelty(score_df = my_scores, refinements = c(4, 8, 10))

# Add colours!

plot_novelty(
  score_df = my_scores,
  refinements = c(4, 8, 10),
  col = list(stroke    = "lightgreen",
             fill_ref  = "red",
             fill      = "blue")
)
```

`plot_richness` *Plot richness of interview codes over time*

Description

The full definition of novel and duplicate codes is in `score_codes()`. Briefly, 'novel' codes are topics/ideas/concepts that were not mentioned in previous interviews, whereas 'duplicate' codes are topics that other interviews have discussed previously.

Some interviews will touch on many different topics and generate many different codes, whereas other interviews will be brief or limited. We call this 'richness'. This plot complements `plot_novelty()` by visualising the richness of each interview in terms of novel and duplicate codes, in context with any refinements to interview questions that were made (marked by stars underneath each bar). By examining this plot together with their field notes, researchers can get insight into the effects of their refinements and the richness of the data.

Usage

```
plot_richness(
  score_df,
  refinements = integer(0),
 col = list(stroke_novel = "black", stroke_duplicate = "gray80", fill_novel = "black",
    fill_duplicate = "gray90")
)
```

Arguments

score_df	(Dataframe) A dataframe of scored codes, as generated by score_codes().
refinements	Either a list object generated by import_field_notes(), or an Integer vector that lists when (in terms of interview sequence) refinements were made to the interview questions. For example, c(10, 15) means that interview questions were revised twice: First **before** the 10th interview, and then again **before** the 15th interview.
col	(List) A List containing named Character vectors. Accepted names are: • stroke_novel and stroke_duplicate control line colours for novel and duplicate codes. • fill_novel and fill_duplicate control fill colours for novel and duplicate codes.

Value

A ggplot object.

See Also

score_codes(), import_field_notes(), plot_novelty(), save_last_plot()

Examples

```
# Field notes and coding matrices included with the package
path_to_notes   <- system.file("insect_study/records/refinements.xlsx", package = "novelqualcodes")
path_to_matrices <- system.file("insect_study/matrices/", package = "novelqualcodes")

# Import the data
my_refinements <- import_field_notes(path_to_notes)
my_matrices    <- import_coding_matrices(path_to_matrices)

# Score novel and duplicate codes
my_scores <- score_codes(my_matrices)

# Generate a plot with no refinements
plot_richness(score_df = my_scores)

# Generate a plot using scored codes and imported refinements
plot_richness(score_df = my_scores, refinements = my_refinements)
```

```
# Generate a plot using scored codes and a vector of refinement times
plot_richness(score_df = my_scores, refinements = c(4, 8, 10))

# Add colours!
plot_richness(
    score_df = my_scores,
    refinements = c(4, 8, 10),
    col = list(stroke_novel     = "lightblue",
               stroke_duplicate = "green",
               fill_novel       = "blue",
               fill_duplicate   = "orange")
)
```

`reshape_for_plots` *Reshape score matrix for ggplot2.*

Description

Manually reshapes the data into Long format and adds a refinement group factor.

Usage

```
reshape_for_plots(score_df, refinements = integer(0))
```

Arguments

`score_df`	(Dataframe) A scored code matrix as generated by `score_codes()`.
`refinements`	Either a list object generated by `import_field_notes()`, or an Integer vector that lists when (in terms of interview sequence) refinements were made to the interview questions. For example, `c(10, 15)` means that interview questions were revised twice: First **before** the 10th interview, and then again **before** the 15th interview.

Value

A dataframe.

save_last_plot *Save the most recent plot to a file*

Description

Save the most recent plot to a file

Usage

```
save_last_plot(
  filename = stop("A save path and filename must be specified."),
  size = "4 x 3 in",
  dpi = 300,
  ...
)
```

Arguments

filename	(Character) The path and filename of the file to create.
size	(Character) The output size of the file, in the form "width x height unit. For example: - "5 x 7 in" - "12 x 8 cm" - "300 x 150 mm" - "1920 x 1080 px"
dpi	(Integer) The resolution (dots per inch) of the output file.
...	Other arguments passed to ggplot2::ggsave().

Value

This function returns nothing, but has the side-effect of writing a file to filename.

See Also

plot_novelty(), plot_richness()

Examples

```
# Coding matrices included with the package
path_to_matrices <- system.file("insect_study/matrices/", package = "novelqualcodes")

# Import the data
my_matrices    <- import_coding_matrices(path_to_matrices)

# Score novel and duplicate codes
my_scores <- score_codes(my_matrices)

# Generate a plot with no refinements
plot_richness(score_df = my_scores)

# Save it to a temporary directory
save_last_plot(file.path(tempdir(), "test_plot.png"), size = "4 x 3 in")
```

```
# Open the temporary directory (if session is running interactively)
if (interactive()) { utils::browseURL(tempdir()) }
```

`score_codes` *Score novel and duplicates codes across interviews*

Description

'Novel' and 'duplicate' codes are scored once per interview; the number of times they are spoken in an interview does not matter.

The definition of whether a code is *novel* or *duplicated* is entirely chronological:

- A **novel code** is a topic/idea/concept that, for example, is mentioned in Interview 17, but was not mentioned in Interviews 1 through 16.
- A **duplicate code** is one that has been talked about in other interviews previously.

The cumulative sum of novel codes is used to visualise a stopping point for qualitative interviews.

Usage

```
score_codes(interviews)
```

Arguments

`interviews`	(List) A list of dataframes, as generated by `import_coding_matrices()`.

Value

A dataframe, with one row per interview and these columns:

1. `itvw_seq`, the chronological order of interviews.
2. `n_codes`, the number of unique codes mentioned in this interview.
3. `n_duplicate`, how many of those codes are duplicates mentioned in previous interviews).
4. `n_novel`, how many of those codes are novel (mentioned for the first time in this interview).
5. `prop_duplicate`, the proportion of this interview's codes that are duplicates.
6. `prop_novel`, the proportion of this interview's codes that are novel.
7. `cumsum_novel`, the cumulative sum of novel codes over time (i.e. across interviews).

See Also

`plot_novelty()`, `plot_richness()`

Examples

```
# A folder of example coding matrices included with the package
path_to_matrices <- system.file("insect_study/matrices/", package = "novelqualcodes")
print(path_to_matrices)

# A list of files in that folder
list.files(path_to_matrices)

# Import them all at once
my_matrices <- import_coding_matrices(path_to_matrices)

# Score them for novel and duplicate codes
my_scores <- score_codes(my_matrices)

# Look inside the result; novel and duplicate codes are scored across
# all interviews.
print(my_scores)
```

Index