Package 'QuickExplore'

April 21, 2026

Type Package

Title Interactive Dataset Explorer for 'R' and 'SAS' and Other Data Formats

Version 0.1.0

Description A 'Shiny' application that provides nice interface for browsing, exploring, summarising, and converting datasets stored in 'SAS' (.sas7bdat, .xpt), CSV (.csv), and 'R' (.rds) formats. Users can register multiple directory-based libraries, interactively filter data using 'dplyr' expressions, inspect per-variable statistics, and export datasets to Excel, JSON, CSV, 'R' data, or 'SAS' transport formats.

License MIT + file LICENSE

URL https://github.com/ramsas88/QuickExplore

BugReports https://github.com/ramsas88/QuickExplore/issues

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports shiny (>= 1.7.0), bslib (>= 0.5.0), DT (>= 0.28), dplyr (>= 1.1.0), haven (>= 2.5.0), readr (>= 2.1.0), jsonlite (>= 1.8.0), writexl (>= 1.4.0), rlang (>= 1.1.0), digest (>= 0.6.0), tools, stats, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown, withr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Ram Gaduputi [aut, cre], Jagadish Katam [aut]

Maintainer Ram Gaduputi <ramsas88@gmail.com>

Repository CRAN

Date/Publication 2026-04-21 20:12:20 UTC

Contents

`code_generator_ui` *Code Generator Module - UI*

Description

Renders a panel displaying auto-generated R code that reproduces the current QuickExplore session (load dataset -> filter/select -> summarise -> export). Users can copy the code to the clipboard or download it as a `.R` script.

Usage

```
code_generator_ui(id)
```

Arguments

`id`	Character string. The Shiny module namespace identifier.

Value

A `shiny::tagList()` with the code generator UI.

See Also

`code_generator_server()`

compute_categorical_summary *Compute frequency statistics for categorical variables*

Description

Returns value frequencies and percentages for each non-numeric variable in `vars`, optionally grouped by a second variable.

Usage

```
compute_categorical_summary(df, vars, group_var = NULL)
```

Arguments

`df`	A `data.frame` or tibble.
`vars`	Character vector of variable names to summarise.
`group_var`	Optional character string naming a grouping variable. Pass `NULL` (default) for no grouping.

Value

A `data.frame` with columns for the grouping variable (if any), the value, its frequency count, percentage, and the variable name. Returns `NULL` if there are no categorical variables in `vars`.

Examples

```
df <- data.frame(sex = c("M","F","M","F","M"), trt = c("A","A","B","B","A"))
compute_categorical_summary(df, c("sex", "trt"))
```

compute_crosstab *Compute a cross-tabulation of two categorical variables*

Description

Produces a wide-format contingency table of `row_var` (rows) by `col_var` (columns), including row and column totals. When a `strat_var` is supplied the table is computed separately for each level of the stratification variable and the results are stacked with a leading `Stratum` column.

Usage

```
compute_crosstab(df, row_var, col_var, strat_var = NULL)
```

Arguments

df	A data.frame or tibble.
row_var	Character string. Name of the row variable (e.g. "SEX").
col_var	Character string. Name of the column variable (e.g. "RACE").
strat_var	Character string or NULL. Optional stratification variable (e.g. "TRT01P"). Pass NULL or "" for an unstratified table.

Details

Missing values in any of the three variables are displayed as "(Missing)" rather than being silently dropped, so analysts can spot incomplete records.

Value

A data.frame in wide format:

- Column 1 (or 2 if stratified): row_var levels plus a "Total" row.
- Middle columns: one column per col_var level.
- Last column: Total (row sums).
- If strat_var is given, a leading Stratum column identifies each stratum. A grand-total block across all strata is **not** appended automatically — compute the unstratified table for that.

Examples

```
df <- data.frame(
  SEX  = c("M","F","M","F","M","F"),
  RACE = c("White","White","Black","Asian","Black","White"),
  TRT  = c("A","A","B","B","A","B")
)
compute_crosstab(df, "SEX", "RACE")
compute_crosstab(df, "SEX", "RACE", strat_var = "TRT")
```

---

compute_numeric_summary

Compute summary statistics for numeric variables

Description

Returns a tidy data frame with N, mean, median, standard deviation, minimum, and maximum for each numeric variable in vars.

Usage

```
compute_numeric_summary(df, vars, group_var = NULL)
```

Arguments

df	A data.frame or tibble.
vars	Character vector of variable names to summarise.
group_var	Optional character string naming a grouping variable. Pass NULL (default) for no grouping.

Value

A data.frame (one row per variable, or per variable × group level) or NULL if there are no numeric variables in vars.

Examples

```
df <- data.frame(x = rnorm(100), y = runif(100), g = rep(c("A", "B"), 50))
compute_numeric_summary(df, c("x", "y"))
compute_numeric_summary(df, c("x", "y"), group_var = "g")
```

---

converter_server *Dataset Converter Module – Server*

Description

Handles the dataset-conversion download for all supported output formats: .rds, .xlsx, .csv, .json, and SAS transport .xpt.

Usage

```
converter_server(id, loaded_data, selected_dataset)
```

Arguments

id	Character string. The Shiny module namespace identifier.
loaded_data	A shiny::reactiveVal() containing the current data.frame.
selected_dataset	A shiny::reactiveVal() with the file path of the active dataset.

Value

A named list with three elements:

output_format A shiny::reactive() returning the selected output format string.

csv_delim A shiny::reactive() returning the CSV delimiter character.

json_pretty A shiny::reactive() returning TRUE to pretty-print JSON output.

See Also

converter_ui()

converter_ui *Dataset Converter Module – UI*

Description

Renders a two-column card layout: a conversion form on the left and a format-reference table plus output preview on the right.

Usage

```
converter_ui(id)
```

Arguments

id	Character string. The Shiny module namespace identifier.

Value

A `shiny::tagList()` with the converter UI.

See Also

`converter_server()`

dataset_browser_server *Dataset Browser Module – Server*

Description

Handles library registration, dataset listing, and dataset loading for the sidebar browser panel.

Usage

```
dataset_browser_server(id, selected_dataset, loaded_data)
```

Arguments

id	Character string. The Shiny module namespace identifier.
selected_dataset	A `shiny::reactiveVal()` that stores the full file path of the currently selected dataset.
loaded_data	A `shiny::reactiveVal()` that stores the loaded `data.frame`.

Value

A list of reactive values: `libraries` (named list of library-path pairs) and `selected_library` (the currently active library name).

See Also

`dataset_browser_ui()`

---

dataset_browser_ui *Dataset Browser Module – UI*

Description

Renders the sidebar panel that lets users add/remove directory-based libraries and select a dataset to load.

Usage

```
dataset_browser_ui(id)
```

Arguments

`id`	Character string. The Shiny module namespace identifier.

Value

A `shiny::tagList()` containing the sidebar UI elements.

See Also

`dataset_browser_server()`

---

data_viewer_server *Data Viewer Module – Server*

Description

Handles data display, filtering, variable inspection, and download for the Data Viewer tab.

Usage

```
data_viewer_server(id, loaded_data, selected_dataset)
```

Arguments

id	Character string. The Shiny module namespace identifier.
loaded_data	A shiny::reactiveVal() containing the current data.frame.
selected_dataset	A shiny::reactiveVal() with the file path of the active dataset.

Value

A named list with three elements:

filtered_data A shiny::reactiveVal() with the current filtered data.frame.

filter_expr A shiny::reactive() returning the raw filter expression string.

selected_vars A shiny::reactive() returning the selected variable names.

See Also

data_viewer_ui()

---

data_viewer_ui *Data Viewer Module – UI*

Description

Creates a tabbed panel with three sub-tabs: an interactive data table (Data Viewer), a filter/subset interface (Explore Data), and a variable metadata explorer (Variables).

Usage

```
data_viewer_ui(id)
```

Arguments

id	Character string. The Shiny module namespace identifier.

Value

A shiny::tagList() with the viewer UI.

See Also

data_viewer_server()

format_file_size *Format a file size in bytes as a human-readable string*

Description

Format a file size in bytes as a human-readable string

Usage

```
format_file_size(size)
```

Arguments

size	Numeric. File size in bytes.

Value

A character string such as "1.4 MB" or "340 KB".

Examples

```
format_file_size(1048576)   # "1 MB"
format_file_size(512)       # "512 B"
```

get_dataset_metadata *Get metadata for a loaded dataset*

Description

Returns file-level metadata including the number of rows and columns, file size, and timestamps.

Usage

```
get_dataset_metadata(df, filepath)
```

Arguments

df	A data.frame or tibble (the loaded data).
filepath	Character string. Path to the source file.

Value

A named list with elements: filename, filepath, format, n_rows, n_cols, file_size, modified, and created.

Examples

```
df <- read_dataset("/data/demog.csv")
meta <- get_dataset_metadata(df, "/data/demog.csv")
meta$n_rows
```

`get_variable_info` *Extract variable-level metadata from a dataset*

Description

Returns a data frame describing each variable: its type, SAS label, SAS format, missing value counts, and number of unique values.

Usage

```
get_variable_info(df)
```

Arguments

`df`	A `data.frame` or tibble.

Value

A `data.frame` with columns `Variable`, `Type`, `Label`, `Format`, `Missing_Count`, `Missing_Pct`, and `N_Unique`.

Examples

```
df <- data.frame(x = 1:5, y = letters[1:5])
get_variable_info(df)
```

`list_datasets` *List supported dataset files in a directory*

Description

Scans a directory for files with extensions `.sas7bdat`, `.xpt`, `.csv`, or `.rds` (case-insensitive) and returns a summary data frame.

Usage

```
list_datasets(dirpath)
```

Arguments

dirpath	Character string. Path to the directory to scan.

Value

A data.frame with columns Name, Format, Size, Modified, and Path. Returns an empty data frame if no supported files are found.

Examples

```
datasets <- list_datasets("/data/mylib")
```

---

read_dataset *Read a dataset based on its file extension*

Description

Dispatches to the appropriate reader based on the file extension. Supported formats: .sas7bdat, .xpt, .csv, .rds.

Usage

```
read_dataset(filepath)
```

Arguments

filepath	Character string. Full path to the dataset file.

Details

For SAS formats (.sas7bdat, .xpt), blank strings are automatically converted to NA after loading. This matches SAS behaviour where a blank character value is treated as a system-missing value, not as a valid empty string.

Value

A data.frame (or tibble) with the dataset contents. For SAS formats, all-whitespace character values are coerced to NA_character_.

Examples

```
df <- read_dataset("/data/mylib/demog.sas7bdat")
df <- read_dataset("/data/exports/study.csv")
```

run_app *Launch the Dataset Explorer Shiny Application*

Description

Opens the interactive Dataset Explorer in your default web browser (or the RStudio Viewer pane when called from within RStudio). The application provides a SAS Studio-style interface for browsing libraries, exploring datasets, computing summary statistics, and converting between data formats.

Usage

```
run_app(...)
```

Arguments

`...`	Additional arguments passed to `shiny::runApp()`, such as `port`, `host`, or `launch.browser`.

Value

Called for its side effect of launching a Shiny application. Returns `NULL` invisibly.

Examples

```
# Launch with default settings
run_app()

# Launch on a specific port without opening a browser
run_app(port = 4321, launch.browser = FALSE)
```

summary_panel_server *Summary Panel Module – Server*

Description

Computes and renders descriptive statistics for numeric and categorical variables, plus a missing-value summary table.

Usage

```
summary_panel_server(id, loaded_data)
```

Arguments

id	Character string. The Shiny module namespace identifier.
loaded_data	A shiny::reactiveVal() containing the current data.frame.

Value

A named list with two elements:

summary_vars A shiny::reactive() returning the selected variable names.

group_var A shiny::reactive() returning the grouping variable name ("" = none).

See Also

summary_panel_ui()

---

summary_panel_ui *Summary Panel Module – UI*

Description

Renders the summary statistics panel with dataset overview cards plus tables for numeric, categorical, and missing-value statistics.

Usage

```
summary_panel_ui(id)
```

Arguments

id	Character string. The Shiny module namespace identifier.

Value

A shiny::tagList() with the summary UI elements.

See Also

summary_panel_server()

Index