Package: cataScience (via r-universe)

November 23, 2026

Title Interactive Training App for Data Science and AI Skills

Version 3.0.0

Description An interactive 'shiny' training companion for people who are new to data, developed for World Health Organization data trainings. Bundles hands-on modules for importing data, missing values, outliers, text cleaning, merging, visualization, and basic statistics, plus a set of AI-skills pages (prompting levels, four practical tasks, AI safety rules, and a methodology case study) and an interactive quiz with per-session topic filters. Includes reusable data-quality, plotting and reporting functions. Launch the cat classroom with run_cata().

License MIT + file LICENSE

Language en-US

URL https://github.com/shanlong-who/cataScience, https://shanlong-who.github.io/cataScience/

BugReports https://github.com/shanlong-who/cataScience/issues

Encoding UTF-8

Depends R (>= 4.1.0)

Imports base64enc, broom, bslib (>= 0.6.0), cowplot, dplyr (>= 1.1.0), DT, forcats, ggplot2, htmltools, markdown, mice, naniar, plotly, purrr, readr, readxl, rlang, scales, shiny (>= 1.8.1), skimr, stringr, tibble, tidyr, zip

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

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Shanlong Ding [aut, cre] (ORCID: <https://orcid.org/0000-0001-9048-6670>)

Maintainer Shanlong Ding <dings@who.int>

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

Date/Publication 2026-11-23 14:58:03 UTC

RemoteUrl https://github.com/cran/cataScience

RemoteRef HEAD

RemoteSha 7927f35dc8461b6bbd5ecf37068d47eda99619f7

Contents

cata_check_keys . . . 2
cata_count_status . . . 3
cata_example . . . 4
cata_export . . . 4
cata_flag_outliers . . . 6
cata_join . . . 6
cata_mice_demo . . . 7
cata_missing . . . 8
cata_plot . . . 8
cata_profile . . . 10
cata_proportion . . . 10
cata_read . . . 11
cata_report . . . 12
run_cata . . . 13
theme_cata . . . 14

Index **15**

`cata_check_keys` *Check a proposed record key*

Description

Check a proposed record key

Usage

```
cata_check_keys(data, key)
```

Arguments

`data`	A data frame.
`key`	Character vector of key column names.

Value

A list with a one-row `summary`, per-row `flags`, and all `review_rows`. Every member of a repeated-key or identical-row group is flagged. Repeated IDs are not automatically deleted.

Examples

```
cata_check_keys(data.frame(id = c("A", "A", NA), value = 1:3), "id")
```

---

cata_count_status *Summarise a reported binary status with unknowns kept visible*

Description

Summarise a reported binary status with unknowns kept visible

Usage

```
cata_count_status(data, status, yes = "Yes", no = "No", by = NULL)
```

Arguments

`data`	A data frame.
`status`	Status column name.
`yes, no`	Distinct character labels for success and failure.
`by`	Optional grouping column name.

Value

A tibble with records, yes, no, unknown, blank, known denominator and percent. Unknown/blank statuses are excluded from the denominator and reported separately. The measure is reported status, not verified service/result availability.

Examples

```
cata_count_status(data.frame(test = c("Yes", "No", "Unknown", NA)), "test")
```

cata_example *Load a documented example*

Description

Load a documented example

Usage

```
cata_example(name = NULL)
```

Arguments

name	Example name. NULL returns the catalogue.

Value

A tibble with a cata_source attribute, or the catalogue.

Examples

```
cata_example()
cats <- cata_example("cats")
```

cata_export *Export a reproducible communication project*

Description

Writes a new directory containing input and selected-data snapshots, an R script, report and dashboard HTML, Quarto sources, and a local Shiny dashboard. The R script reconstructs selected data using the supplied preparation steps, then recomputes summaries and the chart. It uses an input RDS snapshot to preserve identifiers, factors, dates and column types.

Usage

```
cata_export(
  data,
  directory,
  input = data,
  steps = character(),
  join_data = NULL,
  plot_args = NULL,
  summary_spec = NULL,
  chart_data = c("records", "summary"),
  audit = NULL,
```

```
  title = "Data analysis report",
  question = "",
  methods = "",
  finding = "",
  limitations = "",
  source = "",
  unit = "",
  denominator = "",
  width = 8,
  height = 5
)
```

Arguments

`data`	Selected analysis data.
`directory`	A new output directory; existing directories are protected.
`input`	Original input snapshot, before the listed preparation steps.
`steps`	Character vector of R preparation statements, each operating on `analysis.data`. These are written to the exported script, not executed by this function. Only export statements you have reviewed.
`join_data`	Optional reference-table snapshot used by preparation steps.
`plot_args`	Named arguments for `cata_plot()`, excluding data.
`summary_spec`	Optional arguments for `cata_count_status()`, excluding data. The status summary is recomputed when the project is run.
`chart_data`	Use selected records or the status summary for the chart.
`audit`	Optional preparation log.
`title`, `question`, `methods`, `finding`, `limitations`, `source`, `unit`, `denominator`	Author-supplied report text.
`width`, `height`	Figure dimensions in inches, retained in the project.

Value

The new directory, invisibly. Source files are generated even when Quarto is not installed. HTML exports need neither Quarto nor a server.

Examples

```
out <- tempfile("cata-project-")
cata_export(data.frame(value = 1:5), out, title = "Example report")
```

cata_flag_outliers *Flag values for review without changing them*

Description

An IQR flag is a screening result, not evidence of an erroneous value. Use explicit `limits` only when a documented rule supports them.

Usage

```
cata_flag_outliers(x, k = 1.5, limits = NULL)
```

Arguments

`x`	A numeric vector.
`k`	Non-negative multiplier for the IQR rule.
`limits`	Optional numeric lower and upper limits (inclusive).

Value

A tibble retaining every row and value, with bounds, a review flag, and a reason. Missing values receive an unknown outlier flag.

Examples

```
cata_flag_outliers(c(1:9, 100, NA))
```

cata_join *Join tables and return an audit*

Description

Join tables and return an audit

Usage

```
cata_join(
  x,
  y,
  by,
  type = c("left", "inner", "full", "right"),
  relationship = "many-to-one",
  na_matches = "never",
  suffix = c(".x", ".y")
)
```

Arguments

x, y	Data frames.
by	Character join keys, optionally named to match different names.
type	Join type.
relationship	Expected key relationship; checked by dplyr.
na_matches	Whether missing keys can match. Defaults to "never".
suffix	Suffixes for overlapping non-key columns.

Value

A list containing data, audit, both key checks, and unmatched input rows. Input tables are unchanged.

Examples

```
cata_join(data.frame(id = c(1, 2, NA)), data.frame(id = 1:2, label = c("A", "B")), by = "id")
```

---

cata_mice_demo *Explore multiple imputation using synthetic cats*

Description

A fixed teaching simulation with missing weights. Fits the same linear model to each completed dataset and pools model estimates, not datasets. It does not impute uploaded records. Diagnostic plots do not establish that the missingness assumptions hold in a real study.

Usage

```
cata_mice_demo(m = 5, maxit = 5, seed = 2026)
```

Arguments

m	Number of imputations, from 2 to 20.
maxit	Number of iterations, from 2 to 20.
seed	Integer seed for reproducibility. The caller's random-number state is restored on exit.

Value

A list with complete and incomplete synthetic data, the mids object, completed datasets, pooled model estimates and a comparison of the complete-data, complete-case, mean-imputed and pooled slopes.

Examples

```
demo <- cata_mice_demo(m = 3, maxit = 3)
demo$comparison
```

cata_missing *Count missing values and patterns*

Description

Count missing values and patterns

Usage

```
cata_missing(data, by = c("variable", "record", "pattern"))
```

Arguments

data	A data frame.
by	Summary level: variables, records, or missingness patterns.

Value

A tibble. Patterns use `1` for missing and `0` for observed, in input column order; the `variables` attribute records that order.

Examples

```
cata_missing(data.frame(a = c(1, NA), b = c(NA, NA)))
```

cata_plot *Create a chart with explicit variable names*

Description

Returns an editable ggplot object. Bar charts count rows when y is NULL; supplying y uses already summarised values and requires unique chart keys. Lines also require one value per x/group/facet. Charts omit rows missing a required measurement and report that count in the caption. Missing categories in comparison charts are labelled explicitly.

Usage

```
cata_plot(
  data,
  type = "bar",
  x,
  y = NULL,
  group = NULL,
  facet = NULL,
  value = NULL,
  yend = NULL,
```

```
  lower = NULL,
  upper = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  xlab = NULL,
  ylab = NULL,
  sort = FALSE,
  percent = FALSE,
  bins = 10,
  style = "cata",
  log_scale = FALSE,
  reference = 1
)
```

Arguments

`data`	A data frame.
`type`	Chart type: bar, histogram, density, line, scatter, box, dot, heatmap, dumbbell, forest, or cascade.
`x`, `y`	Names of axis variables. `y` is optional only for a count bar, histogram, or density plot. For a forest plot, `x` contains term labels.
`group`, `facet`	Optional grouping and faceting variable names.
`value`	Numeric fill variable for a heatmap.
`yend`	Second numeric value for a dumbbell chart.
`lower`, `upper`	Numeric confidence-limit columns for a forest plot.
`title`, `subtitle`, `caption`	Plot text.
`xlab`, `ylab`	Optional axis labels, including units where appropriate.
`sort`	Sort a single-series bar/dot/dumbbell by its value.
`percent`	Whether values are percentages on a 0 to 100 scale.
`bins`	Number of histogram bins.
`style`	Built-in `"cata"` or optional `"dsi"` theme.
`log_scale`	Use a logarithmic estimate axis for a forest plot.
`reference`	Reference value for a forest plot (1 for an odds ratio).

Value

A ggplot object with a `cata_rows` attribute describing input, plotted, and omitted rows.

Examples

```
d <- data.frame(district = c("A", "B"), percent = c(75, 68))
cata_plot(d, "dot", "district", "percent", percent = TRUE)
```

cata_profile *Summarise the structure and quality of a table*

Description

Summarise the structure and quality of a table

Usage

```
cata_profile(data)
```

Arguments

data	A data frame.

Value

One row per variable, with storage type, record count, observed unique values, missingness, and numeric summaries. No values are changed.

Examples

```
cata_profile(data.frame(id = c("001", "002"), value = c(3, NA)))
```

cata_proportion *Calculate proportions with explicit denominators*

Description

Calculate proportions with explicit denominators

Usage

```
cata_proportion(numerator, denominator, scale = 100)
```

Arguments

numerator, denominator	Numeric counts of equal length, or one scalar.
scale	Multiplier, usually 100 for percentages.

Value

A tibble with the input counts, `value`, and `status`. Invalid counts and zero denominators produce NA, with the reason retained.

Examples

```
cata_proportion(c(200, 233, 0), c(265, 315, 0))
```

`cata_read` *Read a training dataset*

Description

Read a CSV or Excel table without modifying the source file. Specify column types when identifiers contain leading zeroes. Excel notes and footers can be excluded with `range`.

Usage

```
cata_read(
  path,
  format = tools::file_ext(path),
  sheet = 1,
  range = NULL,
  col_types = NULL,
  na = c("", "NA")
)
```

Arguments

`path`	Path to an existing file.
`format`	File extension (`csv`, `xls`, or `xlsx`). Useful for uploaded temporary files whose names do not contain the original extension.
`sheet`	Excel sheet name or number.
`range`	Optional Excel cell range.
`col_types`	Column specification passed to the selected reader.
`na`	Character values to interpret as missing.

Value

A tibble. CSV parsing problems remain available through `readr::problems()`.

Examples

```
cats <- cata_example("cats")
cata_profile(cats)
```

`cata_report` *Write a self-contained report or dashboard*

Description

Creates a local HTML file with no online resources. Text is escaped; narrative findings and limitations come from the author. The dashboard includes local table search. For editable and reproducible source files, use `cata_export()`.

Usage

```
cata_report(
  data,
  file,
  title = "Data analysis report",
  question = "",
  methods = "",
  finding = "",
  limitations = "",
  source = "",
  unit = "",
  denominator = "",
  plot = NULL,
  tables = list(),
  audit = NULL,
  format = c("report", "dashboard"),
  width = 8,
  height = 5,
  overwrite = FALSE
)
```

Arguments

`data`	Data frame describing the selected analysis records.
`file`	Output HTML path. Existing files are protected by default.
`title`, `question`, `methods`, `finding`, `limitations`, `source`, `unit`, `denominator`	Author-supplied report text. Missing text is displayed as "Not provided".
`plot`	Optional ggplot figure.
`tables`	Named list of data frames to include before the data profile.
`audit`	Optional data frame of preparation/selection steps.
`format`	Report or dashboard layout.
`width`, `height`	Figure dimensions in inches.
`overwrite`	Explicitly allow replacing an existing output file.

Value

The output path, invisibly.

Examples

```
f <- tempfile(fileext = ".html")
cata_report(data.frame(value = 1:3), f, title = "Teaching example")
```

`run_cata` *Launch the "A Journey of Data Science" training app*

Description

Starts the interactive training app bundled with the package: import, clean, visualize, and understand data, then learn to work with AI assistants safely. The app opens in your default browser.

Usage

```
run_cata(..., language = c("auto", "en", "zh"))
```

Arguments

`...`	Passed on to `shiny::runApp()`, for example `port` or `launch.browser`.
`language`	Interface language: `"zh"` (Simplified Chinese), `"en"` (English), or `"auto"` (saved browser preference, then browser language). Participants can switch languages in the navigation bar during a session.

Value

No return value; called for the side effect of launching the app. The function blocks the R session while the app is running.

Examples

```
# The app is bundled with the package and launched from its own directory.
app_dir <- system.file("app", package = "cataScience")
file.exists(file.path(app_dir, "app.R"))

# Starting the app needs an interactive session, since it blocks R until
# the browser window is closed.
if (interactive()) {
  run_cata()
}
```

`theme_cata` *A clear theme for training figures*

Description

A clear theme for training figures

Usage

```
theme_cata(base_size = 13, style = c("cata", "dsi"))
```

Arguments

`base_size`	Base text size in points.
`style`	Use the built-in theme or the optional DSIR theme.

Value

A ggplot2 theme.

Examples

```
theme_cata()
```

Index

cata_check_keys, 2
cata_count_status, 3
cata_count_status(), *5*
cata_example, 4
cata_export, 4
cata_export(), *12*
cata_flag_outliers, 6
cata_join, 6
cata_mice_demo, 7
cata_missing, 8
cata_plot, 8
cata_plot(), *5*
cata_profile, 10
cata_proportion, 10
cata_read, 11
cata_report, 12

readr::problems(), *11*
run_cata, 13

shiny::runApp(), *13*

theme_cata, 14