Package: rolescry (via r-universe)

June 22, 2026

Title Name-Blind Variable-Role Detection by Data Signature

Version 0.1.0

Description Deterministic, name-blind detection of variable roles (group, outcome, survival time and event, paired and agreement measurements, repeated measures, scale items, subject identifier, covariate) in tabular data. Roles are assigned from each column's information-theoretic signature -- Shannon entropy, normalized mutual information, and distributional shape -- rather than from column names, so renaming columns to 'col_1', 'col_2', ... does not change the result (` `Data inspice, non nomen''). An optional, capped name-based hint and automatic header-row detection are also provided. No large language models and no external data transmission. Extracted from the 'MDStatR' biostatistics engine; see Boynukara (2026) <doi:10.5281/zenodo.20707791>.

License Apache License (== 2.0)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports stats, utils

Suggests moments, diptest, stringdist, readxl, openxlsx, haven, testthat (>= 3.0.0), knitr, rmarkdown, spelling

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/canboynukara/rolescry

BugReports https://github.com/canboynukara/rolescry/issues

Language en-US

NeedsCompilation no

Author Can Boynukara [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-6075-3923>), M. Yasir Ceyhan [ctb] (ORCID: <https://orcid.org/0009-0008-0985-9838>)

Maintainer Can Boynukara <canboynukara1@gmail.com>

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

Date/Publication 2026-06-22 18:33:02 UTC

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

RemoteRef HEAD

RemoteSha c128fb8631e28f5241bb0773b002cde369640cff

Contents

`compute_nmi` *Normalized mutual information*

Description

Computes the normalized mutual information (NMI) between two discrete variables, a name-blind, information-theoretic measure of association in $[0, 1]$. NMI is the mutual information divided by the smaller of the two marginal Shannon entropies; it is 0 for independent variables and 1 for a perfect (deterministic) association, and unlike a raw chi-squared it is comparable across variables with different numbers of levels.

Usage

```
compute_nmi(x, y = NULL)
```

Arguments

`x`	Either a two-way contingency table / matrix of counts, or a vector (factor, character, or numeric) of the first variable.
`y`	Optional. If `x` is a vector, the second variable's vector; a contingency table is formed via `table(x, y)` on complete cases. Ignored when `x` is already a table/matrix.

Value

A single numeric in $[0, 1]$. Returns 0 for degenerate input (fewer than two rows/columns, zero total, or near-zero marginal entropy).

Examples

```
set.seed(1)
g <- sample(c("A", "B", "C"), 200, replace = TRUE)
y <- ifelse(g == "A", "yes", sample(c("yes", "no"), 200, replace = TRUE))
compute_nmi(g, y)            # > 0: g carries information about y
compute_nmi(g, sample(g))    # ~0: shuffled -> independent
```

`detect_header` *Detect the header row of a raw, unparsed table*

Description

Given a raw table read with *no* header (every cell character), scores each of the first rows with a 7-signal weighted heuristic (alphabetic ratio, non-numeric ratio, uniqueness, normalized Shannon entropy, median string length, alpha-vs-next-row transition, fill completeness) and returns the most header-like row plus repaired, unique column names. Empty cells are upward-filled (merged-cell repair) and any still-empty name becomes `col_<j>`. Base + stats only; no file-format dependencies.

Usage

```
detect_header(raw, verbose = FALSE)
```

Arguments

`raw`	A data.frame or matrix of the raw sheet, read with `header = FALSE` so the header row appears as data.
`verbose`	Logical; emit the chosen row via `message()` if TRUE.

Value

A list with `header_row` (integer), `score` (numeric), `names` (repaired character vector, `length == ncol(raw)`), and `all_scores`.

See Also

`read_data()`

Examples

```
raw <- data.frame(
  V1 = c("age", "34", "51"),
  V2 = c("sex", "M", "F"),
  V3 = c("score", "8.1", "7.4"),
  stringsAsFactors = FALSE
)
detect_header(raw)$names
```

`detect_roles` *Detect variable roles by data signature, not by name*

Description

Inspects an already-loaded data frame and assigns each column (or group of columns) to statistical roles – group variable, continuous/binary outcome, survival time and event, paired and agreement measurement pairs, repeated measures, scale items, subject id, and covariates – using only the data information-theoretic signature (Shannon entropy, distributional shape, inter-column structure) and never the column names. Renaming columns to `col_1, col_2, ...` does not change the result (the name-blindness, or "turnusol", invariant).

Usage

```
detect_roles(data, name_bonus = NULL, verbose = FALSE)

## S3 method for class 'role_detection'
print(x, ...)

## S3 method for class 'role_detection'
summary(object, ...)
```

Arguments

`data`	A `data.frame` (already loaded; for header-aware loading see `read_data()`).
`name_bonus`	Optional named list mapping role keys to character vectors of case-insensitive keyword regex fragments, e.g. `list(group_var = c("treat", "arm"), outcome_binary = c("death"))`. Recognized keys: `group_var`, `outcome_continuous`, `outcome_binary`, `subject_id`, `time_variable`, `event_variable`. `NULL` (default) = pure signature detection.
`verbose`	Logical; if `TRUE`, emit per-role progress via `message()`. Default `FALSE` (silent).
`x`	A `role_detection` object.
`...`	Ignored.
`object`	A `role_detection` object.

Details

Detection is purely mathematical by default (`name_bonus = NULL`). When a keyword dictionary is supplied via `name_bonus`, column names act only as a small, capped tie-breaker (at most a +10 point nudge, i.e. <= 10 percent, applied to candidate selection for the group, outcome, subject-id and survival roles) – the reported confidence stays the mathematical signature. See `rolescry_default_name_bonus()` for a ready-made dictionary.

Value

An S3 object of class "role_detection": a list with

var_info data.frame(column, type) – value-based column typing.

roles named list; each entry has found, columns, score, max_score, pct, detected_by and a components score breakdown.

value_types named character vector of per-column value-type labels.

potential_pairs list of candidate continuous column pairs with paired and agreement scores.

n_obs, n_var dataset dimensions.

See Also

read_data(), compute_nmi(), rolescry_default_name_bonus()

Examples

```
set.seed(1)
d <- data.frame(
  arm  = rep(c(0, 1), each = 50),
  pre  = rnorm(100, 10, 2),
  post = rnorm(100, 11, 2),
  resp = rbinom(100, 1, 0.4)
)
res <- detect_roles(d)
res
res$roles$group_var$columns
```

---

read_data *Read a data file with automatic header detection*

Description

Reads a tabular file into a data.frame, detecting the header row with detect_header() (the data is first read with no header so the header row is visible as data). Delimited text (.csv/.tsv) is read with base R and always works; spreadsheet and statistical formats use optional packages and degrade gracefully with an actionable error if the package is absent.

Usage

```
read_data(
  path,
  header = NULL,
  sheet = NULL,
  na_strings = c("", "NA", "N/A", "n/a", "na", "NULL", "null", "."),
  verbose = FALSE
)
```

Arguments

path	Path to the file.
header	Optional integer giving the 1-based header row to use directly, bypassing detection. NULL (default) auto-detects.
sheet	Optional sheet name/index for Excel files.
na_strings	Character vector of tokens mapped to NA before type conversion.
verbose	Logical; emit the detected header row via message().

Details

Supported: .csv, .tsv/.tab (base); .xlsx/.xls/.xlsm (Suggests: readxl or openxlsx); .sav/.sas7bdat/.dta (Suggests: haven, header intrinsic); .rds (base, returned as stored).

Value

A data.frame with detected column names and per-column types inferred via type.convert.

See Also

detect_header(), detect_roles()

Examples

```
tmp <- tempfile(fileext = ".csv")
writeLines(c("age,sex,score", "34,M,8.1", "51,F,7.4"), tmp)
read_data(tmp)
file.remove(tmp)
```

---

rolescry_default_name_bonus

Default name-bonus keyword dictionary

Description

Returns a ready-made, ASCII-English keyword dictionary suitable for the name_bonus argument of detect_roles(). It externalizes the hard-coded keyword lists that lived inside the original MDStatR engine (group/treatment, outcome, survival-time and event, subject-id terms) into a plain, inspectable, locale-neutral list.

Usage

```
rolescry_default_name_bonus()
```

Details

Passing this turns column names into a small, capped tie-breaker only ($\leq$ 10 percent of the selection score); the mathematical signature still dominates (>= 90 percent), satisfying the name-blindness contract. Detection without it (`name_bonus = NULL`) is purely mathematical.

Value

A named list of character vectors (regex fragments), with keys `group_var`, `outcome_continuous`, `outcome_binary`, `subject_id`, `time_variable`, `event_variable`.

Examples

```
nb <- rolescry_default_name_bonus()
names(nb)
set.seed(1)
d <- data.frame(
  treatment_arm = rep(c("A", "B"), each = 60),
  biomarker     = rnorm(120),
  death         = rbinom(120, 1, 0.3)
)
detect_roles(d, name_bonus = nb)$roles$group_var$columns
```

Index