Package: dataProfilerR (via r-universe)

June 25, 2026

Title Automated Exploratory Data Analysis and Dataset Profiling

Version 0.2.1

Description Profiles a data frame with minimal input: column type inference, missing-value analysis, distributional summary statistics (including skewness and kurtosis), normality tests, outlier detection, correlation and categorical-association analysis, date-column profiling, grouped comparisons and an overall data-quality score, alongside a set of 'ggplot2' visualisations. A single entry point, profile_data(), returns a structured S3 object holding metadata, statistics, diagnostics and plots, with print(), summary() and plot() methods, and report() renders the whole profile to a self-contained HTML file. Statistical methods include the Shapiro-Wilk normality test as implemented by Royston (1995) <doi:10.2307/2986146> and the Anderson-Darling test following Stephens (1974) <doi:10.1080/01621459.1974.10480196>, with power comparisons of these tests in Yap and Sim (2011) <doi:10.1080/00949655.2010.520163>, and the categorical association measure of Cramer (1946, ISBN:9780691080048).

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

Depends R (>= 4.1.0)

Imports ggplot2, stats, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown, nortest, spelling

VignetteBuilder knitr

URL https://github.com/mqfarooqi1/dataProfilerR, https://mqfarooqi1.github.io/dataProfilerR/

BugReports https://github.com/mqfarooqi1/dataProfilerR/issues

Config/Needs/website pkgdown

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Repository https://mqfarooqi1.r-universe.dev

Date/Publication 2026-06-25 06:14:31 UTC

RemoteUrl https://github.com/mqfarooqi1/dataProfilerR

RemoteRef HEAD

RemoteSha 7435089b8bfdb035cff01101cbba8ac2b542b339

Contents

analyze_dates *Profile date / datetime columns*

Description

For each Date/POSIXct column, reports the count, missingness, range, and the largest gap between consecutive (sorted, unique) timestamps – a quick way to spot coverage holes in a time series.

Usage

```
analyze_dates(df, types = NULL)
```

Arguments

df	A data frame.
types	Optional named character vector of column types; computed if not supplied.

Value

A data frame with one row per date column (column, n, n_missing, min, max, range_days, n_unique, max_gap_days), or NULL if there are no date columns.

Examples

```
df <- data.frame(d = as.Date("2026-01-01") + c(0, 1, 2, 10))
analyze_dates(df)
```

analyze_missing *Analyse missing values*

Description

Reports missingness per column and overall, including how many rows are fully complete. Only NA is counted as missing (blank strings are not).

Usage

```
analyze_missing(df)
```

Arguments

df	A data frame.

Value

A list with per_column (a data frame of column, n_missing, pct_missing) and overall (a list with total/missing cell counts, pct_missing, complete_rows and pct_complete_rows).

Examples

```
analyze_missing(data.frame(a = c(1, NA, 3), b = c("x", "y", NA)))
```

`categorical_association` *Categorical association (Cramer's V)*

Description

Computes Cramer's V between every pair of categorical/logical columns. V ranges from 0 (no association) to 1 (perfect association) and is the categorical analogue of a correlation matrix. It is derived from the chi-squared statistic: `V = sqrt(X^2 / (n * (k - 1)))`, where `k` is the smaller of the two factors' level counts.

Usage

```
categorical_association(df, types = NULL, max_levels = 50)
```

Arguments

`df`	A data frame.
`types`	Optional named character vector of column types (from `infer_column_types()`); computed if not supplied.
`max_levels`	Categorical columns with more than this many levels are skipped (a high-cardinality column makes the chi-squared test unreliable and the table huge). Default 50.

Value

A symmetric numeric matrix of Cramer's V with a unit diagonal, or `NULL` if fewer than two eligible categorical columns are present.

Examples

```
df <- data.frame(a = c("x", "x", "y", "y"), b = c("p", "p", "q", "q"),
                 c = c("m", "n", "m", "n"))
categorical_association(df)
```

compare_groups *Compare numeric columns across groups*

Description

Grouped profiling: split the data by a categorical column and summarise each numeric column within each group (count, mean, sd, median, missingness). This is the quickest way to see whether a metric differs by segment.

Usage

```
compare_groups(df, group, max_groups = 50)
```

Arguments

df	A data frame.
group	Name of the grouping column. Should be categorical/logical (or a low-cardinality column); a warning is issued if it has many levels.
max_groups	Maximum number of groups before erroring (guards against accidentally grouping on a near-unique column). Default 50.

Value

A list with group_sizes (a data frame of group, n) and numeric_by_group (a long data frame of group, column, n, n_missing, mean, sd, median), or NULL if there are no numeric columns to compare.

Examples

```
compare_groups(iris, "Species")
```

correlation_analysis *Correlation analysis*

Description

Correlation matrices over the numeric columns, using pairwise-complete observations.

Usage

```
correlation_analysis(df, types = NULL, method = c("pearson", "spearman"))
```

Arguments

df	A data frame.
types	Optional named character vector of column types.
method	Character vector; any of "pearson", "spearman". Default both.

Value

A named list of correlation matrices (one per requested method), or NULL if there are fewer than two numeric columns.

Examples

```
correlation_analysis(iris)
```

---

data_quality_score *Data quality score*

Description

Rolls several signals into a single 0-100 score and a letter grade. The components are completeness (share of non-missing cells), row uniqueness (penalises duplicate rows), and column variability (penalises constant, single-value columns). If an outlier_rate is supplied it adds a cleanliness component. Components are averaged with the supplied weights.

Usage

```
data_quality_score(
  df,
  missing = NULL,
  outlier_rate = NULL,
  weights = c(completeness = 0.4, uniqueness = 0.2, variability = 0.2, cleanliness = 0.2)
)
```

Arguments

df	A data frame.
missing	Optional result of analyze_missing(); computed if NULL.
outlier_rate	Optional fraction (0-1) of numeric cells flagged as outliers; if supplied, a cleanliness component is included.
weights	Optional named numeric vector of component weights. Missing components are dropped and the rest renormalised.

Value

A list with score (0-100), grade (a letter), and components (a named numeric vector of the component scores).

Examples

```
data_quality_score(iris)
```

`detect_outliers` *Detect outliers in a numeric vector*

Description

Three standard rules:

- `"iqr"`: outside `Q1 - k*IQR` / `Q3 + k*IQR` (Tukey's rule, `k = 1.5`).
- `"zscore"`: absolute z-score above `threshold` (default 3).
- `"robust"`: absolute modified z-score using the median and MAD above `threshold` (default 3.5); resistant to the outliers it is detecting.

Usage

```
detect_outliers(x, method = c("iqr", "zscore", "robust"), threshold = NULL)
```

Arguments

`x`	A numeric vector.
`method`	One of `"iqr"`, `"zscore"`, `"robust"`.
`threshold`	Cutoff for `"zscore"`/`"robust"`; the IQR multiplier for `"iqr"`. Defaults: 1.5 (iqr), 3 (zscore), 3.5 (robust).

Value

A list: `method`, `n` (non-missing count), `n_outliers`, `pct`, `is_outlier` (a logical vector aligned to `x`, `FALSE` for `NA`), and `bounds` (lower/upper, where applicable).

Examples

```
detect_outliers(c(1, 2, 3, 4, 100), method = "iqr")
```

infer_column_types *Infer a semantic type for each column*

Description

Maps each column to one of `"numeric"`, `"integer"`, `"date"`, `"logical"`, `"categorical"`, `"text"` or `"other"`. Character columns are split into `"categorical"` and `"text"` heuristically: long strings, or high-cardinality columns where most values are unique, are treated as free text; everything else is categorical.

Usage

```
infer_column_types(df, text_min_avg_chars = 50, text_unique_ratio = 0.8)
```

Arguments

`df`	A data frame.
`text_min_avg_chars`	Average character length above which a character column is considered free text. Default 50.
`text_unique_ratio`	Fraction of unique values above which a character column (with enough rows) is considered free text. Default 0.8.

Value

A named character vector of inferred types, one per column.

Examples

```
infer_column_types(data.frame(a = 1:3, b = c("x", "y", "z"),
                              d = Sys.Date() + 0:2))
```

is_data_profile *Is an object a data_profile?*

Description

Is an object a data_profile?

Usage

```
is_data_profile(x)
```

Arguments

`x`	Any object.

Value

TRUE if x has class data_profile.

Examples

```
is_data_profile(profile_data(iris))
```

---

kurtosis *Sample excess kurtosis*

Description

Moment-based kurtosis minus 3, so a normal distribution scores near 0.

Usage

```
kurtosis(x)
```

Arguments

x	A numeric vector.

Value

A single numeric value, or NA_real_ if there are fewer than four non-missing values or the variance is zero.

Examples

```
kurtosis(rnorm(100))
```

---

normality_tests *Normality tests for numeric columns*

Description

Runs the Shapiro-Wilk test on each numeric/integer column, and the Anderson-Darling test as well if the suggested **nortest** package is installed. Shapiro-Wilk requires 3 to 5000 observations; larger columns are reduced to an evenly-spaced subsample of 5000. The subsample is deterministic and does not touch the session's random-number state.

Usage

```
normality_tests(df, types = NULL, alpha = 0.05)
```

Arguments

df	A data frame.
types	Optional named character vector of column types.
alpha	Significance level for the normal verdict. Default 0.05.

Value

A data frame with one row per numeric column: column, n_used, shapiro_W, shapiro_p, ad_A and ad_p (the Anderson-Darling columns are NA if **nortest** is absent), and a logical normal. Returns NULL if there are no numeric columns.

Examples

```
normality_tests(iris)
```

---

outlier_summary *Outlier summary across numeric columns*

Description

Applies detect_outliers() to every numeric column and tabulates the result.

Usage

```
outlier_summary(df, types = NULL, method = "iqr")
```

Arguments

df	A data frame.
types	Optional named character vector of column types.
method	Outlier method passed to detect_outliers().

Value

A list with per_column (a data frame of column, n_outliers, pct) and overall_rate (fraction of numeric cells flagged, 0-1), or NULL if there are no numeric columns.

Examples

```
outlier_summary(iris)
```

`plot.data_profile` *Plot a data profile*

Description

Returns one of the figures built by `profile_data()`.

Usage

```
## S3 method for class 'data_profile'
plot(
  x,
 which = c("missing", "correlation", "association", "boxplots", "pairs", "distribution"),
  column = NULL,
  ...
)
```

Arguments

`x`	A `data_profile` object (built with `build_plots = TRUE`).
`which`	Which figure: `"missing"`, `"correlation"`, `"association"`, `"boxplots"`, `"pairs"`, or `"distribution"`.
`column`	Column name, required when `which = "distribution"`.
`...`	Ignored.

Value

A **ggplot2** object (also drawn when called at the console).

Examples

```
p <- profile_data(iris)

plot(p, which = "missing")
plot(p, which = "distribution", column = "Sepal.Length")
```

plot_association *Categorical association heatmap*

Description

Heatmap of the Cramer's V matrix from `categorical_association()`.

Usage

```
plot_association(df, max_levels = 50)
```

Arguments

`df`	A data frame.
`max_levels`	Passed to `categorical_association()`.

Value

A **ggplot2** object, or `NULL` (with a warning) if there are fewer than two eligible categorical columns.

Examples

```
plot_association(
  data.frame(a = c("x", "x", "y", "y"), b = c("p", "p", "q", "q"))
)
```

plot_boxplots *Boxplots for numeric columns*

Description

One boxplot per numeric column, faceted with free y-scales so columns on different scales are still readable. Useful as a quick outlier scan.

Usage

```
plot_boxplots(df)
```

Arguments

`df`	A data frame.

Value

A **ggplot2** object, or `NULL` (with a warning) if there are no numeric columns.

Examples

```
plot_boxplots(iris)
```

---

plot_correlation *Correlation heatmap*

Description

A heatmap of the correlation matrix over the numeric columns, annotated with the rounded coefficients.

Usage

```
plot_correlation(df, method = c("pearson", "spearman"))
```

Arguments

df	A data frame.
method	Correlation method: "pearson" or "spearman".

Value

A **ggplot2** object, or NULL (with a warning) if there are fewer than two numeric columns.

Examples

```
plot_correlation(iris)
```

---

plot_distribution *Distribution plot for a single column*

Description

Histogram with a density overlay for numeric columns; a bar chart of the most frequent levels for categorical/text/logical columns.

Usage

```
plot_distribution(df, column, bins = 30, max_levels = 20)
```

Arguments

df	A data frame.
column	Name of the column to plot.
bins	Histogram bins for numeric columns. Default 30.
max_levels	Maximum categories to show for categorical columns. Default 20.

Value

A **ggplot2** object.

Examples

```
plot_distribution(iris, "Sepal.Length")
plot_distribution(iris, "Species")
```

`plot_missing` *Missing-value heatmap*

Description

A tile plot of where `NA`s fall: columns on the x-axis, rows on the y-axis, shaded by whether each cell is missing. For wide/tall data the rows are subsampled to `max_rows` so the plot stays legible.

Usage

```
plot_missing(df, max_rows = 500)
```

Arguments

`df`	A data frame.
`max_rows`	Maximum rows to display (subsampled if exceeded). Default 500.

Value

A **ggplot2** object.

Examples

```
df <- data.frame(a = c(1, NA, 3), b = c(NA, "y", "z"))
plot_missing(df)
```

`plot_pairs` *Pairwise scatterplot matrix*

Description

A scatterplot matrix over selected numeric columns, drawn with facets. Capped at a handful of columns because the number of panels grows quadratically.

Usage

```
plot_pairs(df, columns = NULL, max_cols = 5)
```

Arguments

df	A data frame.
columns	Optional character vector of numeric columns to include. If NULL, the first max_cols numeric columns are used.
max_cols	Maximum number of columns to include. Default 5.

Value

A **ggplot2** object, or NULL (with a warning) if fewer than two numeric columns are available.

Examples

```
plot_pairs(iris, c("Sepal.Length", "Sepal.Width", "Petal.Length"))
```

---

print.data_profile *Print a concise overview of a data profile*

Description

Print a concise overview of a data profile

Usage

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

Arguments

x	A data_profile object.
...	Ignored.

Value

x, invisibly.

Examples

```
print(profile_data(iris))
```

`profile_data` *Profile a data frame*

Description

The package's single entry point. It runs type inference, missing-value analysis, summary statistics, normality tests, outlier detection, correlation analysis and a data-quality score, and (optionally) builds a set of **ggplot2** visualisations. The result is a `data_profile` S3 object with `print()`, `summary()` and `plot()` methods.

Usage

```
profile_data(
  df,
  dataset_name = NULL,
  build_plots = TRUE,
  distributions = TRUE,
  normality = TRUE,
  outlier_method = "iqr",
  cor_method = c("pearson", "spearman"),
  group_by = NULL,
  verbose = FALSE
)
```

Arguments

`df`	A data frame with at least one row and one column and unique, non-empty column names.
`dataset_name`	Optional label stored in the metadata; defaults to the deparsed name of `df`.
`build_plots`	Whether to build the **ggplot2** objects. Set `FALSE` to skip plotting on very wide data. Default `TRUE`.
`distributions`	Whether to build a per-column distribution plot (the eager, heaviest part of plotting). Set `FALSE` on wide data and use `plot_distribution()` on demand. Ignored if `build_plots = FALSE`. Default `TRUE`.
`normality`	Whether to run normality tests. Default `TRUE`.
`outlier_method`	Method passed to `outlier_summary()`: `"iqr"`, `"zscore"` or `"robust"`. Default `"iqr"`.
`cor_method`	Correlation methods: any of `"pearson"`, `"spearman"`.
`group_by`	Optional name of a categorical column. If supplied, a grouped comparison of the numeric columns is added to the diagnostics (see `compare_groups()`).
`verbose`	Print progress messages. Default `FALSE`.

Value

An object of class `data_profile`: a list with elements `metadata`, `statistics`, `diagnostics`, `plots` and `call`.

See Also

print.data_profile(), summary.data_profile(), plot.data_profile()

Examples

```
p <- profile_data(iris)
p
summary(p)

plot(p, which = "correlation")
```

---

report *Render a profile to a self-contained HTML report*

Description

Turns a data_profile into a standalone HTML file containing the metadata, quality score, statistical tables and every figure. The report is built with **rmarkdown**, so a working pandoc installation is required (R Markdown's usual dependency); report() errors clearly if pandoc is unavailable.

Usage

```
report(
  x,
  output_file = "dataProfilerR_report.html",
  title = NULL,
  quiet = TRUE
)
```

Arguments

x	A data_profile (built with build_plots = TRUE).
output_file	Path to write. A bare file name lands in the working directory. Default "dataProfilerR_report.html".
title	Report title. Defaults to the dataset name.
quiet	Passed to rmarkdown::render(). Default TRUE.

Value

The path to the written file, invisibly.

Examples

```
if (requireNamespace("rmarkdown", quietly = TRUE) &&
    rmarkdown::pandoc_available()) {
  p <- profile_data(iris)
  f <- report(p, file.path(tempdir(), "iris_report.html"))
}
```

skewness *Sample skewness*

Description

Moment-based skewness, computed as `m3 / m2^(3/2)` on the non-missing values.

Usage

```
skewness(x)
```

Arguments

`x`	A numeric vector.

Value

A single numeric value, or `NA_real_` if there are fewer than three non-missing values or the variance is zero.

Examples

```
skewness(c(1, 2, 2, 3, 10))
```

summarize_columns *Summary statistics by column type*

Description

Produces a numeric summary data frame (count, missingness, mean, sd, variance, quartiles, IQR, skewness, kurtosis) for numeric and integer columns, and a categorical summary (cardinality and most frequent level) for factor, logical, categorical and text columns.

Usage

```
summarize_columns(df, types = NULL)
```

Arguments

`df`	A data frame.
`types`	Optional named character vector of column types (as returned by `infer_column_types()`). Computed if not supplied.

Value

A list with `numeric` (a data frame, or `NULL` if no numeric columns) and `categorical` (a named list, possibly empty).

Examples

```
summarize_columns(iris)
```

`summary.data_profile` *Detailed summary of a data profile*

Description

Prints the numeric summary, the columns with the most missingness, normality verdicts, outlier counts, and the strongest correlations, and returns the same pieces invisibly as a list.

Usage

```
## S3 method for class 'data_profile'
summary(object, max_rows = 10, ...)
```

Arguments

`object`	A `data_profile` object.
`max_rows`	Maximum rows to print per table. Default 10.
`...`	Ignored.

Value

A list of the printed tables, invisibly.

Examples

```
summary(profile_data(iris))
```

Index