Package 'dqcheckr'

May 20, 2026

Type Package

Title Automated Data Quality Checks for Recurring Dataset Deliveries

Version 0.1.2

Date 2026-05-16

Description Automates quality verification of recurring external dataset deliveries. For each new file arrival, it runs single-snapshot quality checks, compares the file to the previous delivery, writes a self-contained 'HTML' report, and records summary statistics in a local 'SQLite' database for long-term trend tracking. Supports 'CSV' and fixed-width formats. Custom organisation-specific checks can be supplied as plain R files.

License MIT + file LICENSE

URL https://github.com/mickmioduszewski/dqcheckr

BugReports https://github.com/mickmioduszewski/dqcheckr/issues

Encoding UTF-8

Language en-GB

Depends R (>= 4.2)

Imports readr, DBI, RSQLite, rmarkdown, knitr, kableExtra, ggplot2, gridExtra, dplyr, tidyr, yaml, rlang

Suggests testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Mick Mioduszewski [aut, cre]

Maintainer Mick Mioduszewski <mick@mioduszewski.net>

Repository CRAN

Date/Publication 2026-05-20 08:00:07 UTC

Contents

`check_missing_rate` *QC-01: Check missing rate per column*

Description

Returns a `dq_result` per column flagging columns whose proportion of missing or empty values exceeds `max_missing_rate`.

Usage

```
check_missing_rate(df, config)
```

Arguments

`df`	A data frame with all columns as character vectors.
`config`	Named list as returned by `load_config`.

Value

A list of `dq_result` objects, one per column.

Examples

```
cfg_dir <- system.file("demonstrations/config", package = "dqcheckr")
cfg  <- load_config("starwars_csv", config_dir = cfg_dir)
path <- system.file("demonstrations/data/starwars.csv", package = "dqcheckr")
df   <- read_dataset(path, cfg)
check_missing_rate(df, cfg)
```

detect_files *Detect current and previous dataset files*

Description

Resolves the current and previous file paths from the configuration. If `current_file` is set explicitly, it is used directly. Otherwise the two most recently modified files in `folder` are used.

Usage

```
detect_files(config)
```

Arguments

`config`	Named list. Merged configuration as returned by `load_config`.

Value

A named list with elements `current` (character path) and `previous` (character path or `NULL`).

Examples

```
cfg_dir <- system.file("demonstrations/config", package = "dqcheckr")
cfg <- load_config("starwars_csv", config_dir = cfg_dir)
cfg$current_file <- system.file("demonstrations/data/starwars.csv",
                                package = "dqcheckr")
files <- detect_files(cfg)
files$current
```

dq_result *Construct a data quality result object*

Description

Creates the atomic result unit returned by every check function.

Usage

```
dq_result(
  check_id,
  check_name,
  column = NA_character_,
  status,
  observed,
  threshold = NA_character_,
  message
)
```

Arguments

check_id	Character. Short identifier for the check (e.g. "QC-01").
check_name	Character. Human-readable name of the check.
column	Character. Column the check applies to, or NA_character_ for row-level or file-level checks.
status	Character. One of "PASS", "WARN", "FAIL", or "INFO".
observed	Character. What was observed (e.g. "5.2% missing").
threshold	Character. The configured threshold, or NA_character_ if not applicable.
message	Character. Human-readable description of the result.

Value

A named list with seven elements: check_id, check_name, column, status, observed, threshold, message.

Examples

```
dq_result("QC-01", "Missing rate", column = "age",
          status = "PASS", observed = "0% missing",
          message = "No missing values.")
```

---

infer_col_type *Infer the logical type of a character column*

Description

Classifies a character vector as "date", "numeric", "character", or "unknown" by applying rules in priority order.

Usage

```
infer_col_type(x, threshold = 0.9)
```

Arguments

x	Character vector to classify (as read from a CSV or FWF file).
threshold	Numeric. Minimum proportion of non-empty values that must parse as numeric for the column to be classified as "numeric". Defaults to 0.90. Configurable via type_inference_threshold in rule_overrides.

Value

A single character string: "date", "numeric", "character", or "unknown".

Examples

```
infer_col_type(c("2024-01-01", "2024-06-15"))   # "date"
infer_col_type(c("1.5", "2.0", "3.1"))          # "numeric"
infer_col_type(c("high", "low", "medium"))       # "character"
infer_col_type(c(NA, "", NA))                    # "unknown"
infer_col_type(c(rep("1", 17), "a", "b", "c"), threshold = 0.80)  # "numeric"
```

`load_config` *Load and merge dataset configuration*

Description

Reads the global `dqcheckr.yml` and the dataset-specific YAML, merging `rule_overrides` from the dataset config on top of `default_rules` from the global config.

Usage

```
load_config(dataset_name, config_dir)
```

Arguments

`dataset_name`	Character. Dataset name; must match `<dataset_name>.yml` in `config_dir`.
`config_dir`	Character. Path to the directory containing both YAML files.

Value

A named list representing the merged configuration.

Examples

```
cfg_dir <- system.file("demonstrations/config", package = "dqcheckr")
cfg <- load_config("starwars_csv", config_dir = cfg_dir)
cfg$format
```

overall_status *Compute the worst status across a list of dq_result objects*

Description

Returns the single worst status in precedence order: "FAIL" > "WARN" > "PASS" > "INFO".

Usage

```
overall_status(results)
```

Arguments

results	A list of dq_result objects.

Value

A single character string: "FAIL", "WARN", "PASS", or "INFO".

Examples

```
r1 <- dq_result("QC-01", "test", status = "PASS", observed = "ok", message = "ok")
r2 <- dq_result("QC-02", "test", status = "WARN", observed = "ok", message = "ok")
overall_status(list(r1, r2))  # "WARN"
```

read_dataset *Read a dataset file into a data frame*

Description

Reads a CSV or fixed-width file, coercing all columns to character and trimming whitespace. Encoding and delimiter are taken from config.

Usage

```
read_dataset(path, config)
```

Arguments

path	Character. Path to the file to read.
config	Named list. Merged configuration as returned by load_config. Must include format ("csv" or "fwf"). For FWF files, fwf_widths is required and fwf_col_names and fwf_skip are optional.

Value

A data frame with all columns as character vectors.

Examples

```
cfg_dir <- system.file("demonstrations/config", package = "dqcheckr")
cfg  <- load_config("starwars_csv", config_dir = cfg_dir)
path <- system.file("demonstrations/data/starwars.csv", package = "dqcheckr")
df   <- read_dataset(path, cfg)
```

---

read_recent_snapshots *Read recent snapshot history from the SQLite database*

Description

Retrieves the `n` most recent run records for a given dataset from the snapshot database, ordered newest-first.

Usage

```
read_recent_snapshots(db_path, dataset_name, n = 10)
```

Arguments

`db_path`	Character. Path to the SQLite database file.
`dataset_name`	Character. Dataset name to filter on.
`n`	Integer. Maximum number of records to return. Defaults to 10.

Value

A data frame with one row per run and columns including `id`, `run_timestamp`, `file_name`, `row_count`, `overall_status`, `check_pass_count`, `check_warn_count`, `check_fail_count`. Returns an empty data frame if the database does not exist or contains no records for the dataset.

Examples

```
history <- read_recent_snapshots(tempfile(fileext = ".sqlite"), "starwars_csv")
```

run_comparison_checks *Run all version comparison checks between two dataset snapshots*

Description

Runs CP-01 to CP-08 comparing a current delivery against the previous one.

Usage

```
run_comparison_checks(df_current, df_previous, config)
```

Arguments

df_current	A data frame. The current delivery.
df_previous	A data frame. The previous delivery.
config	Named list. Merged configuration as returned by load_config.

Value

A list of dq_result objects. The list carries attributes new_cols and dropped_cols (character vectors) for use by the snapshot writer.

Examples

```
cfg_dir   <- system.file("demonstrations/config", package = "dqcheckr")
cfg       <- load_config("starwars_csv", config_dir = cfg_dir)
curr_path <- system.file("demonstrations/data2/starwars_v2.csv", package = "dqcheckr")
prev_path <- system.file("demonstrations/data2/starwars_v1.csv", package = "dqcheckr")
curr      <- read_dataset(curr_path, cfg)
prev      <- read_dataset(prev_path, cfg)
results   <- run_comparison_checks(curr, prev, cfg)
```

run_custom_checks *Run organisation-specific custom checks*

Description

Sources the R file specified by config$custom_checks_file, which must define a function custom_checks(df) returning a list of dq_result objects. Returns an empty list if custom_checks_file is not set in the config.

Usage

```
run_custom_checks(df, config)
```

Arguments

df	A data frame. The current delivery.
config	Named list. Merged configuration as returned by load_config.

Value

A list of dq_result objects (may be empty).

Examples

```
cfg_dir <- system.file("demonstrations/config", package = "dqcheckr")
cfg     <- load_config("starwars_csv", config_dir = cfg_dir)
path    <- system.file("demonstrations/data/starwars.csv", package = "dqcheckr")
df      <- read_dataset(path, cfg)
results <- run_custom_checks(df, cfg)
```

---

run_dq_check *Run a full data quality check pipeline*

Description

Orchestrates the complete dqcheckr pipeline: loads configuration, detects files, runs QC and comparison checks, writes a snapshot to SQLite, and renders an HTML report.

Usage

```
run_dq_check(dataset_name, config_dir = ".", open_report = TRUE)
```

Arguments

dataset_name	Character. Name of the dataset; must match a YAML config file <dataset_name>.yml in config_dir.
config_dir	Character. Path to the directory containing dqcheckr.yml and the dataset YAML file. Defaults to ".".
open_report	Logical. Whether to open the HTML report in the browser after rendering (only takes effect in interactive sessions).

Value

Invisibly, a named list with:

status Overall status string: "PASS", "WARN", "FAIL", or "INFO".

report_path Absolute path to the rendered HTML report.

snapshot_id Integer row ID of the snapshot written to SQLite, or NULL if the write failed.

Examples

```
tmp <- gsub("\\\\", "/", tempdir())
dat <- system.file("demonstrations/data/starwars.csv", package = "dqcheckr")
writeLines(c(
  paste0('snapshot_db: "',       tmp, '/snap.sqlite"'),
  paste0('report_output_dir: "', tmp, '"'),
  'default_rules:',
  '  max_missing_rate: 0.60',
  '  min_row_count: 80'
), file.path(tmp, "dqcheckr.yml"))
writeLines(c(
  'dataset_name: "starwars_csv"',
  paste0('current_file: "', dat, '"'),
  'format: csv',
  'encoding: "UTF-8"',
  'delimiter: ","'
), file.path(tmp, "starwars_csv.yml"))
result <- run_dq_check("starwars_csv", config_dir = tmp, open_report = FALSE)
result$status
```

---

run_qc_checks *Run all generic quality checks on a dataset*

Description

Runs the full QC check suite (QC-01 to QC-14, SC-01, SC-02) against a single data frame snapshot.

Usage

```
run_qc_checks(df, config)
```

Arguments

df	A data frame with all columns as character vectors (as returned by read_dataset).
config	Named list. Merged configuration as returned by load_config.

Value

A list of dq_result objects.

Examples

```
cfg_dir <- system.file("demonstrations/config", package = "dqcheckr")
cfg     <- load_config("starwars_csv", config_dir = cfg_dir)
path    <- system.file("demonstrations/data/starwars.csv", package = "dqcheckr")
df      <- read_dataset(path, cfg)
results <- run_qc_checks(df, cfg)
```

Index