Package: agriDQ (via r-universe)

May 21, 2026

Title Data Quality Checks and Statistical Assumption Testing for Agricultural Experiments

Version 0.1.3

Description Provides a comprehensive pipeline for data quality checks and statistical assumption diagnostics in agricultural experimental data. Functions cover outlier detection using Interquartile Range (IQR) fence, Z-score, modified Z-score (Hampel identifier), Grubbs test and Dixon Q-test with consensus flagging; missing data pattern analysis and mechanism classification (Missing Completely At Random/Missing At Random/Missing Not At Random (MCAR/MAR/MNAR)) via Little's test; normality testing using Shapiro-Wilk, Anderson-Darling, Kolmogorov-Smirnov, Lilliefors, Pearson chi-square and Jarque-Bera tests; homogeneity of variance via Bartlett, Levene and Fligner-Killeen tests; independence of errors via Durbin-Watson, Breusch-Godfrey and Wald-Wolfowitz runs tests; experimental design validation for Completely Randomised Design (CRD), Randomised Complete Block Design (RCBD), Latin Square Design (LSD) and factorial designs; qualitative variable consistency checks; and automated HyperText Markup Language (HTML) report generation. Designed to align with Findable, Accessible, Interoperable and Reusable (FAIR) data principles. Methods follow Gomez and Gomez (1984, ISBN:978-0471870920) and Montgomery (2017, ISBN:978-1119492443).

License GPL (>= 3)

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports stats, graphics, grDevices, utils, nortest, car, lmtest, tseries, stringdist

Suggests testthat (>= 3.0.0), covr, MASS

Config/testthat/edition 3

NeedsCompilation no

Author Sadikul Islam [aut, cre] (ORCID:
<https://orcid.org/0000-0003-2924-7122>)

Maintainer Sadikul Islam <sadikul.islamiasri@gmail.com>

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

Date/Publication 2026-04-21 19:42:01 UTC

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

RemoteRef HEAD

RemoteSha eb751ad5c66b0c82091fbd929c56647af37e0290

Contents

`agri_trial` *Simulated wheat variety trial dataset (RCBD)*

Description

A simulated Randomised Complete Block Design (RCBD) dataset for a wheat variety trial with 4 treatments and 5 blocks (20 plots total). The dataset contains one intentional high outlier (plot P03, yield = 8.9 t/ha) and one missing value (plot P17) for demonstration of the **agriDQ** quality-check functions.

Usage

```
agri_trial
```

Format

A data frame with 20 rows and 7 variables:

`plot_id` Character. Unique plot identifier (P01–P20).

`block` Factor. Block identifier (B1–B5).

`treatment` Factor. Treatment/variety label (T1–T4).

`variety` Character. Wheat variety name corresponding to each treatment (HD2967, GW322, PBW343, WH1105).

`yield` Numeric. Grain yield in tonnes per hectare (t/ha). Contains one outlier (~8.9 t/ha) and one `NA`.

`plant_height` Numeric. Mean plant height in cm.

`tillers` Numeric. Mean effective tiller count per plant.

Details

Data were generated with `set.seed(2025)` using an additive RCBD model:

$$y_{ij} = \mu + \tau_i + \beta_j + \varepsilon_{ij}$$

where $\mu = 4.2$ t/ha (grand mean), treatment effects are T1 = 0, T2 = +0.4, T3 = +0.8, T4 = -0.2 t/ha, block effects are $N(0, 0.3^2)$, and errors are $N(0, 0.4^2)$. Two observations were manually perturbed: plot P03 set to 8.9 t/ha (high outlier) and plot P17 set to `NA` (missing plot).

Source

Simulated data generated for package demonstration purposes.

Examples

```
data(agri_trial)
str(agri_trial)
summary(agri_trial)
```

`check_design` *Validate experimental design structure and balance*

Description

Checks the structural integrity of agricultural experimental data against a declared experimental design. Verifies treatment completeness, replication balance, block structure, missing treatment combinations, degrees of freedom for error, and minimum sample size.

Usage

```
check_design(
  df,
  treatment = NULL,
  block = NULL,
  response = NULL,
  design = c("RCBD", "CRD", "LSD", "factorial"),
  factors = NULL,
  expected_reps = NULL,
  alpha = 0.05
)
```

Arguments

df	A data frame containing the experimental data.
treatment	Character. Name of the treatment factor column.
block	Character or NULL. Name of the block/replicate column (required for RCBD and LSD).
response	Character. Name of the numeric response column.
design	Character. One of "CRD", "RCBD", "LSD", "factorial". Default "RCBD".
factors	Character vector. Additional factor column names for factorial designs.
expected_reps	Integer or NULL. Expected replications per treatment. If NULL, inferred from data.
alpha	Numeric. Significance level. Default 0.05.

Details

Checks performed:

1. Response variable is numeric.
2. Missing values in response column.
3. Replication balance (equal n per treatment).
4. Expected replications match (if expected_reps supplied).
5. RCBD: each treatment appears exactly once per block.
6. Error degrees of freedom ≥ 10 (Gomez & Gomez, 1984).
7. Factorial: all factor-level combinations present.
8. Minimum sample size guideline.

Value

An object of class "agriDQ_design" with per-check results, treatment levels, and a pass/warn/fail summary.

References

Gomez, K.A. and Gomez, A.A. (1984). *Statistical Procedures for Agricultural Research*, 2nd ed. Wiley, ISBN:978-0471870920. pp. 8–55.

Examples

```
df <- expand.grid(
  treatment = paste0("T", 1:4),
  block     = paste0("B", 1:3),
  KEEP.OUT.ATTRS = FALSE,
  stringsAsFactors = FALSE
)
df$yield <- rnorm(nrow(df), 4.5, 0.5)
result <- check_design(df, treatment = "treatment",
                       block = "block", response = "yield",
                       design = "RCBD")
print(result)
```

---

check_homogeneity *Test homogeneity of variance across treatment groups*

Description

Tests the equal-variance assumption required for ANOVA using three complementary tests: Bartlett, Levene (Brown-Forsythe), and Fligner-Killeen. Reports a consensus and a practical variance ratio.

Usage

```
check_homogeneity(x, group, alpha = 0.05)
```

Arguments

x	Numeric vector of the response variable.
group	Factor or character vector of group labels.
alpha	Numeric. Significance level. Default 0.05.

Details

Test choice:

Bartlett Most powerful when data are truly normal; sensitive to departures from normality.

Levene (Brown-Forsythe) Robust to non-normality; uses group medians rather than means. Recommended for most agricultural data where mild skewness is common.

Fligner-Killeen Fully nonparametric; most robust option for clearly non-normal data.

The variance ratio (max/min across groups) is also reported. A ratio exceeding 3 is a practical warning for ANOVA robustness (Montgomery, 2017).

Value

An object of class "agriDQ_homogeneity" containing results (list of agriDQ_result), var_by_group, var_ratio, consensus, and n.

References

Levene, H. (1960). Robust tests for equality of variances. In *Contributions to Probability and Statistics*, ed. I. Olkin, pp. 278–292. Stanford University Press.

Montgomery, D.C. (2017). *Design and Analysis of Experiments*, 9th ed. Wiley, ISBN:978-1119492443.

Examples

```
set.seed(3)
yield <- c(rnorm(10, 4, 0.5), rnorm(10, 4, 1.5), rnorm(10, 4, 0.8))
trt   <- rep(c("T1", "T2", "T3"), each = 10)
result <- check_homogeneity(yield, trt)
print(result)
```

`check_independence` *Test independence of residuals / errors*

Description

Tests whether residuals from a fitted model (or a raw sequential vector) are independent — a core assumption for ANOVA and regression in agricultural field trials. Applies three complementary tests.

Usage

```
check_independence(residuals, alpha = 0.05, plot = TRUE)
```

Arguments

`residuals`	Numeric vector of model residuals or raw sequential observations.
`alpha`	Numeric. Significance level. Default `0.05`.
`plot`	Logical. Produce residuals-vs-order and ACF plots. Default `TRUE`.

Details

Tests applied:

Durbin-Watson Tests for lag-1 autocorrelation. DW ≈ 2 indicates no autocorrelation; DW < 1.5 suggests positive autocorrelation (common in field trials with spatial trends).

Breusch-Godfrey Tests for higher-order serial correlation (lags 1 and 2).

Wald-Wolfowitz runs test Nonparametric test for randomness of the residual sequence.

Pass all three residuals from `residuals(fit)` after fitting an ANOVA or regression model, with observations in field-plot order.

Value

An object of class "agriDQ_independence" containing results (list of agriDQ_result), consensus, and n.

References

Durbin, J. and Watson, G.S. (1950). Testing for serial correlation in least squares regression. *Biometrika*, **37**(3/4), 409–428. doi:10.1093/biomet/37.34.409

Examples

```
set.seed(5)
fit <- lm(rnorm(30) ~ rep(1:3, 10))
result <- check_independence(residuals(fit), plot = FALSE)
print(result)
```

---

check_missing *Analyse missing data patterns and classify missingness mechanism*

Description

Provides comprehensive missing data analysis: per-column and per-row missingness rates, pattern matrix, Little's MCAR test, and an inferred missingness mechanism with imputation recommendation.

Usage

```
check_missing(df, alpha = 0.05, plot = TRUE)
```

Arguments

df	A data frame (numeric and/or factor/character columns).
alpha	Numeric. Significance level for Little's MCAR test. Default 0.05.
plot	Logical. Produce a missingness pattern heatmap. Default TRUE.

Details

Missingness mechanisms:

MCAR Missing Completely At Random — independent of observed and unobserved values. Complete-case analysis is valid.

MAR Missing At Random — depends only on observed values. Multiple imputation is appropriate.

MNAR Missing Not At Random — depends on the missing value itself. Requires sensitivity analysis.

Little's (1988) MCAR test is applied to numeric columns. A significant chi-squared statistic rejects MCAR, suggesting MAR or MNAR.

Value

An object of class "agriDQ_missing" containing:

col_summary Per-column missing count and percentage.

row_summary Per-row missing count.

pattern_matrix Binary matrix (1 = missing).

little_test Named list: statistic, df, p_value.

mechanism Character: "MCAR", "MAR", or "undetermined".

recommendation Character: suggested next step.

References

Little, R.J.A. (1988). A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, **83**(404), 1198–1202. doi:10.1080/01621459.1988.10478722

Examples

```
set.seed(1)
df <- data.frame(
  yield    = c(rnorm(18, 4.5), NA, NA),
  height   = c(NA, rnorm(19, 80)),
  treatment = rep(c("T1", "T2"), 10)
)
result <- check_missing(df, plot = FALSE)
print(result)
```

---

check_normality *Comprehensive normality testing for agricultural experimental data*

Description

Applies a battery of normality tests selected by sample size, together with skewness, excess kurtosis, and a Q-Q plot. Returns a consensus recommendation for ANOVA/regression suitability.

Usage

```
check_normality(
  x,
  alpha = 0.05,
  tests = c("shapiro", "anderson", "ks", "lilliefors", "pearson", "jarque"),
  plot = TRUE,
  varname = "variable"
)
```

Arguments

x	Numeric vector of observations.
alpha	Numeric. Significance level. Default 0.05.
tests	Character vector of tests to apply. Any subset of "shapiro", "anderson", "ks", "lilliefors", "pearson", "jarque". Defaults to all.
plot	Logical. Produce Q-Q and histogram plots. Default TRUE.
varname	Character. Label for plot titles and output.

Details

Test selection guidance for agricultural data:

- $n < 50$: Shapiro-Wilk is most powerful (Razali & Wah, 2011).
- $50 \leq n < 200$: Anderson-Darling is preferred.
- $n \geq 200$: Lilliefors or Kolmogorov-Smirnov.
- Jarque-Bera assesses skewness and kurtosis directly.

Consensus is "pass" when the majority of applicable tests do not reject normality.

Value

An object of class "agriDQ_normality" with:

varname Variable label.

n Sample size (non-missing).

descriptives List: mean, median, SD, CV, skewness, excess kurtosis, min, max.

results Named list of agriDQ_result objects.

consensus Character: "pass", "warning", or "fail".

consensus_msg Character: actionable recommendation.

References

Razali, N.M. and Wah, Y.B. (2011). Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analytics*, **2**(1), 21–33.

Examples

```
yield <- rnorm(30, mean = 4.2, sd = 0.6)
result <- check_normality(yield, varname = "Wheat yield (t/ha)",
                          plot = FALSE)
print(result)
```

`check_outliers` *Univariate outlier detection for agricultural experimental data*

Description

Applies five complementary outlier detection methods and combines them into a consensus flag. A consensus flag is raised when **at least two** methods independently flag the same observation, which substantially reduces false positives compared to any single method.

Usage

```
check_outliers(
  x,
  method = c("iqr", "zscore", "hampel", "grubbs", "dixon"),
  alpha = 0.05,
  iqr_k = 1.5,
  z_threshold = 3,
  hampel_k = 3.5,
  labels = NULL
)
```

Arguments

`x`	Numeric vector of observations (e.g., yield, plant height).
`method`	Character vector. One or more of `"iqr"`, `"zscore"`, `"hampel"`, `"grubbs"`, `"dixon"`. Default uses all five.
`alpha`	Numeric. Significance level for formal tests. Default `0.05`.
`iqr_k`	Numeric. IQR multiplier for the fence method. Default `1.5`; use `3` for extreme-outliers-only detection.
`z_threshold`	Numeric. Z-score threshold. Default `3`.
`hampel_k`	Numeric. Hampel identifier threshold in MAD units. Default `3.5`.
`labels`	Optional character vector of observation labels (e.g., plot IDs) of the same length as `x`.

Details

Methods applied:

- **IQR fence** — flags values outside $[Q_1 - k \cdot IQR,\ Q_3 + k \cdot IQR]$.
- **Z-score** — flags $|z| >$ threshold where $z = (x_i - \bar{x})/s$.
- **Hampel identifier (modified Z-score)** — robust to masking. Uses $M_i = 0.6745(x_i - \tilde{x})/MAD$. Recommended for small agricultural trial datasets where classical Z-score is distorted by the very outliers being sought.
- **Grubbs test** — formal test for a single extreme outlier under normality (Grubbs, 1950). Iterates if an outlier is found.
- **Dixon Q-test** — suitable for small samples ($n \leq 30$) (Dixon, 1950).

Value

An object of class "agriDQ_outlier" — a list containing:

`flags` Data frame with flag status from each method and a `consensus` column.

`summary` Named integer vector: outlier count per method.

`n_flagged` Integer: observations flagged by consensus.

`n_total` Integer: total observations.

`n_valid` Integer: non-missing observations.

References

Grubbs, F.E. (1950). Sample criteria for testing outlying observations. *Annals of Mathematical Statistics*, **21**(1), 27–58. doi:10.1214/aoms/1177729885

Dixon, W.J. (1950). Analysis of extreme values. *Annals of Mathematical Statistics*, **21**(4), 488–506. doi:10.1214/aoms/1177729747

Examples

```
set.seed(42)
yield <- c(rnorm(20, mean = 4.5, sd = 0.5), 9.8, 0.2)
result <- check_outliers(yield, method = c("iqr", "zscore", "hampel"))
print(result)
```

---

check_outliers_mv *Multivariate outlier detection using Mahalanobis distance*

Description

Detects multivariate outliers using the squared Mahalanobis distance with a chi-squared critical value. Useful for observations that are not extreme on any single variable but are unusual in combination (e.g., very high yield paired with very low plant height).

Usage

```
check_outliers_mv(df, alpha = 0.05, robust = FALSE)
```

Arguments

`df`	A numeric data frame or matrix. Rows are observations, columns are variables.
`alpha`	Numeric. Significance level for the chi-squared critical value. Default `0.05`.
`robust`	Logical. If `TRUE`, use robust covariance estimation via the minimum covariance determinant (MCD) from the **MASS** package if available. Default `FALSE`.

Value

An object of class "agriDQ_mout" containing Mahalanobis distances (distances), critical value (critical), logical flag vector (flags), count of flagged observations (n_flagged), and a summary.

Examples

```
set.seed(7)
df <- data.frame(
  yield    = c(rnorm(20, 4.5, 0.5), 9.0),
  plant_ht = c(rnorm(20, 80,   5),  30.0)
)
result <- check_outliers_mv(df)
print(result)
```

---

check_qualitative *Check quality of categorical / qualitative variables*

Description

Detects common data quality issues in categorical variables: inconsistent capitalisation, whitespace errors, near-duplicate labels (fuzzy matching), unexpected factor levels, and rare categories.

Usage

```
check_qualitative(
  df,
  cols = NULL,
  expected_levels = NULL,
  fuzzy_threshold = 2L,
  rare_threshold = 0.02
)
```

Arguments

df	A data frame.
cols	Character vector of columns to check. If NULL (default), all character and factor columns are checked.
expected_levels	Named list mapping column names to character vectors of valid levels. E.g. list(season = c("Kharif", "Rabi")).
fuzzy_threshold	Integer. Levenshtein distance threshold for near-duplicate detection. Applied only when minimum label length exceeds 3 characters (to avoid false positives on short codes). Default 2.
rare_threshold	Numeric. Proportion below which a category is flagged as rare. Applied only when $n \geq 20$. Default 0.02.

Details

Issues detected per column:

1. **Missing values** — count and percentage.
2. **Case inconsistency** — e.g., `"Kharif"` vs `"kharif"` vs `"KHARIF"`.
3. **Whitespace** — leading/trailing spaces or double spaces.
4. **Near-duplicates** — label pairs within `fuzzy_threshold` Levenshtein distance (long labels only).
5. **Unexpected levels** — values not in `expected_levels`.
6. **Rare categories** — frequency below `rare_threshold` (large samples only).

Value

An object of class `"agriDQ_qualitative"` with per-column results (`col_results`), a consolidated issue table (`issue_table`), and `n_issues`.

Examples

```
df <- data.frame(
  treatment = c("T1", "T1", "t1", "T2", "T2"),
  season    = c("Kharif", "Kharif", "kharif", "Rabi", "Rabi"),
  stringsAsFactors = FALSE
)
result <- check_qualitative(df,
  expected_levels = list(season = c("Kharif", "Rabi")))
print(result)
```

`classify_missing` *Classify missingness mechanism per variable using logistic regression*

Description

For each variable with missing values, fits a logistic regression of the missingness indicator on all other observed variables to assess whether the MAR assumption is plausible.

Usage

```
classify_missing(df, alpha = 0.05)
```

Arguments

`df`	A data frame.
`alpha`	Numeric. Significance level. Default `0.05`.

Value

A data frame with columns `variable`, `pct_missing`, `lr_pvalue`, and `mechanism`.

Examples

```
set.seed(2)
df <- data.frame(
  yield = c(NA, rnorm(9, 4.5, 0.5)),
  trt   = rep(c("T1", "T2"), 5)
)
classify_missing(df)
```

`generate_dq_report` *Generate an automated HTML data quality report*

Description

Produces a self-contained HTML report from a `run_dq_pipeline` result. The report includes a colour-coded scorecard (green / amber / red), a detailed results table, and an interpretation guide.

Usage

```
generate_dq_report(
  pipeline,
  output_file,
  title = "agriDQ Data Quality Report",
  author = "agriDQ"
)
```

Arguments

`pipeline`	An object of class `"agriDQ_pipeline"` from `run_dq_pipeline`.
`output_file`	Character. Path for the HTML output file (e.g. `tempfile(fileext = ".html")`). No default; the caller must supply a path. Use `tempdir()` in examples and tests.
`title`	Character. Report title.
`author`	Character. Author name for the report header.

Value

Invisibly returns the path to the generated HTML file.

Examples

```
data(agri_trial)

pl  <- run_dq_pipeline(agri_trial, response = "yield",
                       treatment = "treatment", block = "block",
                       plot = FALSE)
tmp <- tempfile(fileext = ".html")
generate_dq_report(pl, output_file = tmp, author = "Researcher")
```

---

print.agriDQ_result *Print an agriDQ_result object*

Description

Print an agriDQ_result object

Usage

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

Arguments

x	An object of class "agriDQ_result".
...	Ignored.

Value

Invisibly returns x.

---

run_dq_pipeline *Run the complete data quality pipeline*

Description

Runs all six data quality modules in sequence on a numeric response variable within an agricultural experimental data frame and returns a unified result with a master summary table.

Usage

```
run_dq_pipeline(
  df,
  response = NULL,
  treatment = NULL,
  block = NULL,
  design = "RCBD",
  alpha = 0.05,
  plot = TRUE,
  outlier_method = c("iqr", "zscore", "hampel")
)
```

Arguments

df	A data frame.
response	Character. Name of the numeric response variable.
treatment	Character or NULL. Treatment factor column name.
block	Character or NULL. Block/replicate column name.
design	Character. Experimental design type passed to check_design. Default "RCBD".
alpha	Numeric. Significance level. Default 0.05.
plot	Logical. Produce diagnostic plots from sub-modules. Default TRUE.
outlier_method	Character vector. Methods for check_outliers. Default is IQR, Z-score, and Hampel.

Value

An object of class "agriDQ_pipeline" containing:

steps Named list of sub-module results.

summary Data frame: module, test, statistic, p-value, status.

response, treatment, block, design Input parameters.

n, alpha, timestamp Metadata.

See Also

generate_dq_report

Examples

```
data(agri_trial)

result <- run_dq_pipeline(agri_trial,
  response  = "yield",
  treatment = "treatment",
  block     = "block",
  design    = "RCBD",
  plot      = FALSE)
```

```
print(result)
```

`standardise_labels` *Standardise categorical labels in a data frame*

Description

Applies automatic label standardisation: trims whitespace, collapses multiple spaces, and optionally converts case or applies a lookup-table replacement.

Usage

```
standardise_labels(
  df,
  cols = NULL,
  case = c("none", "lower", "upper", "title"),
  lookup = NULL
)
```

Arguments

`df`	A data frame.
`cols`	Character vector of column names to standardise. Defaults to all character/factor columns.
`case`	Character. One of `"none"`, `"lower"`, `"upper"`, `"title"`. Default `"none"`.
`lookup`	Named list of replacement maps, e.g. `list(season = c("kharif" = "Kharif", "rabi" = "Rabi"))`.

Value

A data frame with standardised labels.

Examples

```
df <- data.frame(trt = c(" T1 ", "T1", "t1", "T2"),
                 stringsAsFactors = FALSE)
standardise_labels(df, case = "upper")
```

Index