

Package: PrismR (via r-universe)

October 28, 2026

Type Package

Title Pre-Modeling Statistical Audit Framework

Version 1.0.0

Description Introduces a statistical validation layer between data preprocessing and predictive modeling. Evaluates the statistical readiness of tabular datasets through data quality assessment, leakage detection, transformation analysis, and feature stability.

License MIT + file LICENSE

URL <https://arshit-dv.github.io/PrismR/>,
<https://github.com/Arshit-dv/PrismR>

BugReports <https://github.com/Arshit-dv/PrismR/issues>

Encoding UTF-8

Imports graphics, stats, utils, e1071, ggplot2, ggrepel

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Arshit Choudhary [aut, cre, cph], Vanshika Sharma [aut, ctb],
Amrit Pal [ths]

Maintainer Arshit Choudhary <arshitchoudhary3010@gmail.com>

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

Date/Publication 2026-10-28 10:50:17 UTC

RemoteUrl <https://github.com/cran/PrismR>

RemoteRef HEAD

RemoteSha d7c204e9428060e49a2cc6945fa694256f488526

Contents

detect_leakage	2
feature_stability	3
plot.PrismReport	4
print.PrismReport	6
prism	6
quality_score	7
recommend_transform	8
summary.PrismReport	9

Index	10
-------	----

detect_leakage	<i>Detect Potential Data Leakage</i>
----------------	--------------------------------------

Description

Identifies potential sources of data leakage prior to predictive modeling. The assessment includes identifier columns, duplicate columns, high-cardinality features, predictors identical to the target, and predictors showing an almost perfect correlation with the target.

Usage

```
detect_leakage(data, target = NULL)
```

Arguments

- data A data.frame to assess.
- target Optional name of the target variable. If supplied, predictors identical to the target or nearly perfectly correlated with the target are flagged as potential leakage.

Value

- A named list containing:
- identifier_columns** Names of columns identified as potential identifiers.
 - duplicate_columns** Names of duplicated columns.
 - high_cardinality_columns** Names of high-cardinality columns.
 - target_leakage** Names of predictors identical to the target variable.
 - correlation_leakage** Numeric predictors with near-perfect correlation to the target.
 - leakage_score** Overall leakage score ranging from 0 to 100.
 - variables** A data frame containing variable-level leakage diagnostics, including identifier, duplicate, high-cardinality, target leakage, correlation leakage, and leakage score.

Examples

```
# Supervised leakage detection
res <- detect_leakage(airquality, target = "Ozone")
res$leakage_score
res$correlation_leakage

# Unsupervised check for IDs and duplicates
res_unsupervised <- detect_leakage(iris)
res_unsupervised$identifier_columns
```

feature_stability	<i>Feature Stability and Distribution Drift Assessment</i>
-------------------	--

Description

Evaluates feature distribution drift between reference and current datasets (or across sequential time partitions of a single dataset) using the Population Stability Index (PSI) and scale-normalized Wasserstein distance across both numeric and categorical variables.

Usage

```
feature_stability(
  data,
  current = NULL,
  split_ratio = 0.5,
  partition_col = NULL,
  verbose = FALSE
)
```

Arguments

data	A data.frame used as the reference baseline dataset (or the full dataset for single-dataset drift evaluation).
current	Optional current data.frame to compare against data.
split_ratio	Numeric. Proportion of rows to allocate to the reference dataset in single-dataset mode. Defaults to 0.5 (first 50% vs. last 50%).
partition_col	Optional character. Column name to order by before partitioning in single-dataset mode (e.g., a timestamp or sequence ID).
verbose	Logical. If TRUE, prints a formatted console stability report. Defaults to FALSE.

Details

Feature drift is evaluated using standard industry thresholds:

- **Stable (PSI < 0.10):** Insignificant distributional shift.

- **Moderate Drift ($0.10 \leq \text{PSI} < 0.25$):** Noticeable shift; monitoring recommended.
- **Unstable / Severe Drift ($\text{PSI} \geq 0.25$):** Severe distributional divergence; retraining or feature review required.

For categorical variables and discrete bins, Bayesian Laplace smoothing is applied to prevent division-by-zero errors when frequencies are zero in one partition.

If only one dataset is supplied, the function performs a sequential partition-based stability assessment (comparing the first partition to the second partition of observations). For production monitoring, users should provide reference and current datasets separately via `data` and `current`.

Value

A list containing:

stability_score Overall stability score from 0 to 100.

verdict Categorical stability verdict: "Stable", "Moderate Drift", or "Unstable".

variables A data frame of feature-level diagnostics containing variable, type, psi, wasserstein, stability_score, and status.

unstable_features Character vector of features flagged with severe drift ($\text{PSI} \geq 0.25$).

moderate_features Character vector of features flagged with moderate drift ($0.10 \leq \text{PSI} < 0.25$).

stable_features Character vector of features exhibiting stable distributions ($\text{PSI} < 0.10$).

method Character string describing the assessment method used ("two_sample_comparison" or "sequential_partition").

Examples

```
# Single-dataset sequential drift evaluation
res <- feature_stability(airquality)
res$stability_score
res$verdict

# Comparing reference baseline vs current dataset
ref <- data.frame(val = rnorm(100, 0, 1))
cur <- data.frame(val = rnorm(100, 2, 1))
res2 <- feature_stability(ref, current = cur)
res2$verdict
```

plot.PrismReport

Plot a Prism Report

Description

Visualises the results of a Prism Analysis using one of several diagnostic plots.

Usage

```
## S3 method for class 'PrismReport'
plot(
  x,
  y = NULL,
  type = c("radial", "circular", "bubble", "radar"),
  feature = NULL,
  ...
)
```

Arguments

x	A PrismReport object returned by prism .
y	Ignored; included for compatibility with generic plot.
type	Character string specifying the plot to display. One of "radial", "circular", "bubble", or "radar".
feature	Character string specifying the variable to display when type = "radar".
...	Additional graphical arguments reserved for future extensions.

Details

Available visualisations include:

- **radial**: Radial/gauge chart summarising the overall health of each PrismR module and dataset readiness.
- **circular**: Feature Diagnostic Rose (coxcomb) displaying feature Completeness (petal length), Redundancy & Leakage Status (petal color), and Recommended Transformation (outer rim cap) around an open donut center.
- **bubble**: Statistical feature map displaying Data Completeness (100% - Missing%) versus Distribution Skewness, with leakage vector colors and transformation shapes.
- **radar**: Spider/radar chart displaying the raw statistical diagnostic profile of a selected variable (Completeness, Symmetry, Tail Normalcy, Uniqueness, Leakage Safety, and Feature Stability).

Value

A ggplot object.

See Also

[prism](#)

Examples

```
report <- prism(airquality, target = "Ozone")
plot(report, type = "radial")
plot(report, type = "circular")
plot(report, type = "bubble")
plot(report, type = "radar", feature = "Solar.R")
```

<code>print.PrismReport</code>	<i>Print a PrismReport</i>
--------------------------------	----------------------------

Description

Print a PrismReport

Usage

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

Arguments

<code>x</code>	A PrismReport object returned by prism .
<code>...</code>	Additional arguments passed to print methods.

Value

Invisibly returns the input PrismReport object.

Examples

```
report <- prism(airquality, target = "Ozone")
print(report)
```

<code>prism</code>	<i>Perform Complete Prism Statistical Validation</i>
--------------------	--

Description

Evaluates dataset readiness for predictive modeling across four foundational statistical pillars: data quality, data leakage, distribution transformations, and feature stability. Produces an overall Model Readiness score (0-100) and a gatekeeping verdict.

Usage

```
prism(data, target = NULL, current = NULL)
```

Arguments

<code>data</code>	A data.frame to assess.
<code>target</code>	Optional character string specifying the name of the target/label variable.
<code>current</code>	Optional current data.frame to evaluate distribution drift against baseline data.

Details

The four evaluation dimensions include:

- **Data Quality:** Evaluates missingness, duplicate rows, and constant features.
- **Data Leakage:** Identifies identifier keys, duplicate features, high-cardinality columns, and target correlation leakers.
- **Transformations:** Evaluates skewness and excess kurtosis to recommend variance-stabilizing transforms (Box-Cox, Yeo-Johnson, Log).
- **Feature Stability:** Assesses covariate distribution drift via Population Stability Index (PSI) and Wasserstein distance.

Value

A PrismReport S3 object containing:

- quality** List of data quality metrics returned by [quality_score](#).
- leakage** List of data leakage diagnostics returned by [detect_leakage](#).
- transformation** List of transformation recommendations returned by [recommend_transform](#).
- stability** List of stability metrics returned by [feature_stability](#).
- readiness** Overall composite Model Readiness score (0-100).
- verdict** Gatekeeping verdict: "Model Ready", "Proceed with Caution", or "Action Required".

Examples

```
report <- prism(airquality, target = "Ozone")
print(report)
summary(report)
```

quality_score	<i>Data Quality Assessment</i>
---------------	--------------------------------

Description

Computes basic data quality metrics for a dataset, including missing values, duplicate rows, constant columns, and an overall quality score.

Usage

```
quality_score(data)
```

Arguments

- data A data.frame to assess.

Value

A named list containing:

missing_percent Percentage of missing values.

duplicate_rows Number of duplicated rows.

constant_columns Names of constant columns.

n_constant_columns Number of constant columns.

quality_score Overall quality score (0-100).

variables A data frame containing variable-level quality metrics, including missing percentage, constant feature indicator, and quality score.

Examples

```
res <- quality_score(airquality)
res$quality_score
res$missing_percent
```

recommend_transform	<i>Recommend Feature Transformations</i>
---------------------	--

Description

Analyses numeric variables in a data frame and recommends suitable transformations based on skewness and excess kurtosis. The function does not modify the data; it only provides transformation recommendations to assist with preprocessing before statistical modelling or machine learning.

Usage

```
recommend_transform(data)
```

Arguments

data A data.frame containing predictor variables.

Details

Recommendations are based on:

- Distribution skewness.
- Excess kurtosis (tail heaviness).
- Whether the variable contains non-positive values.

Possible recommendations include:

- None
- Log
- Box-Cox
- Yeo-Johnson

Value

A list containing:

recommendations A data frame of features requiring transformation (excluding "None").

n_numeric Number of numeric variables analysed.

n_recommended Number of variables for which a transformation was recommended.

variables A complete data frame of all analysed numeric features and their metrics.

Examples

```
recommend_transform(iris)
```

summary.PrismReport	<i>Summarize a PrismReport</i>
---------------------	--------------------------------

Description

Summarize a PrismReport

Usage

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

Arguments

object	A PrismReport object returned by prism .
...	Additional arguments passed to summary methods.

Value

Invisibly returns the input PrismReport object.

Examples

```
report <- prism(airquality, target = "Ozone")  
summary(report)
```

Index

`detect_leakage`, [2](#), [7](#)

`feature_stability`, [3](#), [7](#)

`plot.PrismReport`, [4](#)

`print.PrismReport`, [6](#)

`prism`, [5](#), [6](#), [6](#), [9](#)

`quality_score`, [7](#), [7](#)

`recommend_transform`, [7](#), [8](#)

`summary.PrismReport`, [9](#)