Package: leafareaR (via r-universe)

July 7, 2026

Type Package

Title Leaf Area Modeling, Evaluation, and Prediction

Version 0.0.1

Description Tools for leaf area estimation based on leaf length, leaf width, and observed leaf area. The package supports data validation, predictor generation, descriptive statistics, exploratory graphics, scatterplot matrices, linear models, nonlinear models, mixed models, model evaluation, ranking, equation generation, prediction, export of results and plots, and an interactive 'shiny' application. Methods implemented in the package are aligned with non-destructive allometric workflows described by Ribeiro et al. (2024) <doi:10.1016/j.sajb.2024.07.006>, Ribeiro et al. (2023) <doi:10.1590/1807-1929/agriambi.v27n3p209-215>, and Ribeiro et al. (2025) <doi:10.1590/0103-8478cr20230550>.

License GPL (>= 3)

URL `https://github.com/agrobioestat/leafareaR`

BugReports `https://github.com/agrobioestat/leafareaR/issues`

Encoding UTF-8

LazyData true

Depends R (>= 4.4.0)

Imports graphics, ggplot2, lme4, lmerTest, MuMIn, reformulas, rlang, shiny, stats, utils

Suggests knitr, readxl, rmarkdown, testthat (>= 3.0.0), writexl

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev

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

Date/Publication 2026-06-24 18:14:29 UTC

RemoteUrl https://github.com/agrobioestat/leafarear

RemoteRef HEAD

RemoteSha ec205ffcdb13eca0bee11b85a5802b83401db367

Contents

`la_abs_bias_metric` *Calculate absolute bias*

Description

Calculate absolute bias

Usage

```
la_abs_bias_metric(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_abs_bias_metric(obs, pred)
```

`la_add_equation_to_results` *Add equation information to a results table*

Description

Adds `equation`, `coefficients_text`, and `random_effect` columns to the results or summary table returned by a model-fitting function.

Usage

```
la_add_equation_to_results(fit_object, digits = 4)
```

Arguments

`fit_object`	A fit object containing `models` and either `results` or `summary`.
`digits`	Number of decimal places used in displayed equations.

Value

A data.frame.

`la_bias` *Calculate prediction bias*

Description

Calculate prediction bias

Usage

```
la_bias(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_bias(obs, pred)
```

`la_build_equation` *Build a readable equation from a fitted model*

Description

Returns a readable equation starting with `LA =`.

Usage

```
la_build_equation(model, digits = 4)
```

Arguments

`model`	A fitted model object.
`digits`	Number of decimal places used in the displayed equation.

Value

A character string.

`la_ccc` *Calculate Lin's concordance correlation coefficient*

Description

Calculate Lin's concordance correlation coefficient

Usage

```
la_ccc(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_ccc(obs, pred)
```

`la_create_derived` *Create derived leaf parameters*

Description

Generates derived variables from leaf length (`L`) and leaf width (`W`) to support leaf area (`LA`) modeling.

Usage

```
la_create_derived(data, variables = "all", keep_original = TRUE)
```

Arguments

`data`	A data.frame containing at least `L` and `W`.
`variables`	Character vector with the derived variables to create. Use `"all"` to create all available derived variables.
`keep_original`	Logical. If `TRUE`, keeps original columns in the output.

Value

A data.frame with derived variables added.

Examples

```
data(leafarea_sample)
head(la_create_derived(leafarea_sample, variables = c("LW", "L2", "W2")))
```

`la_d` *Calculate Willmott's index of agreement*

Description

Calculate Willmott's index of agreement

Usage

```
la_d(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_d(obs, pred)
```

`la_descriptive_default` *Summarize the default leaf area variables*

Description

Returns descriptive statistics for the default variables available in a `leafareaR` dataset, using any combination of `L`, `W`, and `LA` found in the supplied data.frame.

Usage

```
la_descriptive_default(data, na.rm = TRUE, digits = 4)
```

Arguments

`data`	A data.frame containing at least one of `L`, `W`, or `LA`.
`na.rm`	Logical. If `TRUE`, missing values are removed.
`digits`	Number of decimal places used to round the output.

Value

A data.frame with descriptive statistics for the available default variables.

Examples

```
data(leafarea_sample)
la_descriptive_default(leafarea_sample)
```

`la_descriptive_stats` *Calculate descriptive statistics for selected variables*

Description

Computes a descriptive summary for selected numeric variables in a leaf area dataset. The output is intended to support data exploration before model fitting and can be applied to the original measurements as well as derived predictors.

Usage

```
la_descriptive_stats(data, variables = NULL, na.rm = TRUE, digits = 4)
```

Arguments

`data`	A data.frame containing numeric variables.
`variables`	Character vector with variable names to summarize. If `NULL`, all numeric variables in `data` are used.
`na.rm`	Logical. If `TRUE`, missing values are removed before computing statistics.
`digits`	Number of decimal places used to round the output.

Value

A data.frame with descriptive statistics for each selected variable.

Examples

```
data(leafarea_sample)
la_descriptive_stats(leafarea_sample, variables = c("L", "W", "LA"))
```

`la_evaluate_linear_models` *Evaluate all linear models from a* `la_fit_linear_models()` *object*

Description

Evaluate all linear models from a `la_fit_linear_models()` object

Usage

```
la_evaluate_linear_models(fit_object, digits = 4)
```

Arguments

`fit_object`	Object returned by `la_fit_linear_models()`.
`digits`	Number of decimal places for rounding.

Value

A data.frame with metrics for all candidate linear models.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
met <- la_evaluate_linear_models(fit)
met[, c("model_id", "RMSE", "R2")]
```

`la_evaluate_mixed_models` *Evaluate all mixed models from a* `la_fit_mixed_models()` *object*

Description

Evaluate all mixed models from a `la_fit_mixed_models()` object

Usage

```
la_evaluate_mixed_models(fit_object, digits = 4)
```

Arguments

`fit_object`	Object returned by `la_fit_mixed_models()`.
`digits`	Number of decimal places for rounding.

Value

A data.frame with metrics for all candidate mixed models.

Examples

```
data(leafarea_sample)
fit <- la_fit_mixed_models(leafarea_sample, group_var = "species")
la_evaluate_mixed_models(fit)[, c("model_id", "RMSE", "R2")]
```

`la_evaluate_model` *Evaluate a single fitted model*

Description

Evaluate a single fitted model

Usage

```
la_evaluate_model(
  model,
  data = NULL,
  response = "LA",
  model_id = NA_character_,
  model_type = NA_character_,
  digits = 4
)
```

Arguments

`model`	A fitted model object (`lm`, `nls`, or `merMod`).
`data`	Optional data.frame used to fit the model. Required for `nls` and mixed models.
`response`	Character string with the response variable name.
`model_id`	Optional model identifier.
`model_type`	Optional model type label.
`digits`	Number of decimal places for rounding.

Value

A one-row data.frame.

Examples

```
data(leafarea_sample)
dat <- la_create_derived(leafarea_sample, variables = "LW")
m <- lm(LA ~ LW, data = dat)
la_evaluate_model(m, model_id = "lm_LW", model_type = "linear")
```

`la_evaluate_nonlinear_models`
Evaluate all nonlinear models from a `la_fit_nonlinear_models()` *object*

Description

Evaluate all nonlinear models from a `la_fit_nonlinear_models()` object

Usage

```
la_evaluate_nonlinear_models(fit_object, digits = 4)
```

Arguments

`fit_object`	Object returned by `la_fit_nonlinear_models()`.
`digits`	Number of decimal places for rounding.

Value

A data.frame with metrics for all candidate nonlinear models.

Examples

```
data(leafarea_sample)
fit <- la_fit_nonlinear_models(leafarea_sample,
                               models = c("power_LW", "exponential_LW"))
la_evaluate_nonlinear_models(fit)[, c("model_id", "RMSE", "R2")]
```

`la_extract_coefficients`
Extract model coefficients

Description

Returns a coefficient table from a fitted linear, nonlinear, or mixed model.

Usage

```
la_extract_coefficients(model)
```

Arguments

`model`	A fitted model object.

Value

A data.frame with coefficient names and estimates.

`la_feature_display_names` *Display labels for leaf variables*

Description

Returns user-friendly labels as either a named character vector or a plain character vector.

Usage

```
la_feature_display_names(named = FALSE)
```

Arguments

`named`	Logical. If `TRUE`, returns a named vector where names are the internal variable names and values are display labels.

Value

A character vector.

`la_feature_labels` *Display labels for leaf variables*

Description

Returns a table of internal variable names and user-friendly labels used throughout the package interface.

Usage

```
la_feature_labels()
```

Value

A data.frame with internal names and display labels.

`la_fit_linear_models` *Fit candidate linear models for leaf area estimation*

Description

Fits a default or user-supplied set of linear models using `L`, `W`, `LA`, and derived variables created automatically when needed.

Usage

```
la_fit_linear_models(
  data,
  formulas = NULL,
  include_no_intercept = TRUE,
  include_multiple = TRUE,
  include_polynomial = TRUE
)
```

Arguments

`data`	A data.frame containing at least `L`, `W`, and `LA`.
`formulas`	Optional named list of formulas. If `NULL`, the default formulas from `la_linear_formulas()` are used.
`include_no_intercept`	Logical. Only used when `formulas = NULL`.
`include_multiple`	Logical. Only used when `formulas = NULL`.
`include_polynomial`	Logical. Only used when `formulas = NULL`.

Value

A list with fitted models, formulas, modeling data, and a summary table enriched with equation information.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
names(fit$models)
```

`la_fit_mixed_models` *Fit candidate linear mixed-effects models for leaf area estimation*

Description

Fits a default or user-supplied set of linear mixed-effects models using `L`, `W`, `LA`, derived variables created automatically when needed, and a user-supplied grouping variable.

Usage

```
la_fit_mixed_models(
  data,
  group_var,
  formulas = NULL,
  random_slope = FALSE,
  include_multiple = TRUE,
  include_polynomial = TRUE,
  REML = FALSE,
  control = NULL
)
```

Arguments

`data`	A data.frame containing at least `L`, `W`, `LA`, and the grouping variable.
`group_var`	Character string with the grouping variable name.
`formulas`	Optional named list of formulas. If `NULL`, default formulas from `la_mixed_formulas()` are used.
`random_slope`	Logical. Only used when `formulas = NULL`.
`include_multiple`	Logical. Only used when `formulas = NULL`.
`include_polynomial`	Logical. Only used when `formulas = NULL`.
`REML`	Logical passed to `lmer()`.
`control`	Optional control object passed to `lmer()`.

Value

A list with fitted models, formulas, modeling data, and a summary table enriched with equation information.

Examples

```
data(leafarea_sample)
fit <- la_fit_mixed_models(leafarea_sample, group_var = "species")
names(fit$models)
```

`la_fit_nonlinear_models`

Fit multiple nonlinear models

Description

Fits built-in or user-supplied nonlinear model specifications.

Usage

```
la_fit_nonlinear_models(
  data,
  models = NULL,
  specs = NULL,
  control = stats::nls.control(maxiter = 200, warnOnly = TRUE)
)
```

Arguments

`data`	A data.frame containing at least `L`, `W`, and `LA`.
`models`	Optional character vector of model IDs. Used only when `specs` is `NULL`.
`specs`	Optional named list of nonlinear model specifications.
`control`	Control list passed to `stats::nls()`.

Value

A list containing fitted models, specifications, data, and summary.

Examples

```
data(leafarea_sample)
fit <- la_fit_nonlinear_models(leafarea_sample, models = c("power_LW", "exponential_LW"))
names(fit$models)
fit$summary[, c("model_id", "converged")]
```

`la_input_overview` *Summarize a validated leaf area dataset*

Description

Produces a concise overview of a validated dataset, including its dimensions, variable names, and summary statistics for `L`, `W`, and `LA`. This function is useful as a quick check before creating derived variables or fitting candidate models.

Usage

```
la_input_overview(data)
```

Arguments

`data`	A validated data.frame containing `L`, `W`, and `LA`.

Value

A list with the number of rows and columns, variable names, and a summary of the main measurement variables.

Examples

```
data(leafarea_sample)
validated_data <- la_validate_input(leafarea_sample)
overview <- la_input_overview(validated_data)
overview$n_rows
overview$summary
```

`la_linear_fitted_values` *Extract fitted values from linear model results*

Description

Extracts observed and fitted values for one selected linear model from the object returned by `la_fit_linear_models()`.

Usage

```
la_linear_fitted_values(fit_object, model_id)
```

Arguments

`fit_object`	Object returned by `la_fit_linear_models()`.
`model_id`	Character string with the model identifier.

Value

A data.frame with observed, fitted, and residual values.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
head(la_linear_fitted_values(fit, "lm_LW"))
```

`la_linear_formulas` *List default linear model formulas*

Description

Returns the default set of candidate linear formulas used by `la_fit_linear_models()`.

Usage

```
la_linear_formulas(
  include_no_intercept = TRUE,
  include_multiple = TRUE,
  include_polynomial = TRUE
)
```

Arguments

`include_no_intercept`	Logical. If `TRUE`, includes the no-intercept model `LA ~ 0 + LW`.
`include_multiple`	Logical. If `TRUE`, includes multiple linear models.
`include_polynomial`	Logical. If `TRUE`, includes quadratic and cubic linear models based on derived variables.

Value

A named list of formulas.

Examples

```
names(la_linear_formulas())
```

`la_list_derived` *List available derived variables*

Description

Returns the default derived variables available in `leafareaR`.

Usage

```
la_list_derived()
```

Value

A character vector.

Examples

```
la_list_derived()
```

`la_mae` *Calculate mean absolute error*

Description

Calculate mean absolute error

Usage

```
la_mae(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_mae(obs, pred)
```

`la_mape` *Calculate mean absolute percentage error*

Description

Calculate mean absolute percentage error

Usage

```
la_mape(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value in percentage.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_mape(obs, pred)
```

`la_matrixplot` *Create a matrix plot for selected variables*

Description

Generates a scatterplot matrix for selected numeric variables.

Usage

```
la_matrixplot(
  data,
  variables = NULL,
  hist = TRUE,
  pch = 19,
  cex = 0.6,
  col = "darkgreen",
  main = NULL
)
```

Arguments

`data`	A data.frame containing numeric variables.
`variables`	Character vector with variable names to include. If NULL, uses available default variables among L, W, LA.
`hist`	Logical; if TRUE, draws histograms on the diagonal.
`pch`	Plotting character for points.
`cex`	Point size.
`col`	Point color.
`main`	Optional main title.

Value

Invisibly returns the selected data used in the matrix plot.

Examples

```
data(leafarea_sample)

# To display the plot in an interactive R session, run:
# la_matrixplot(leafarea_sample, variables = c("L", "W", "LA"))

# CRAN-safe example:
grDevices::pdf(NULL)
la_matrixplot(leafarea_sample, variables = c("L", "W", "LA"))
grDevices::dev.off()
```

`la_matrixplot_default` *Create a default matrix plot for leaf variables*

Description

Generates a matrix plot using available default variables among L, W, and LA.

Usage

```
la_matrixplot_default(
  data,
  hist = TRUE,
  pch = 19,
  cex = 0.6,
  col = "darkgreen",
  main = "Matrix plot of leaf variables"
)
```

Arguments

`data`	A data.frame containing leaf measurements.
`hist`	Logical; if TRUE, draws histograms on the diagonal.
`pch`	Plotting character for points.
`cex`	Point size.
`col`	Point color.
`main`	Optional main title.

Value

Invisibly returns the selected data used in the matrix plot.

Examples

```
data(leafarea_sample)

# To display the plot in an interactive R session, run:
# la_matrixplot_default(leafarea_sample)

# CRAN-safe example:
grDevices::pdf(NULL)
la_matrixplot_default(leafarea_sample)
grDevices::dev.off()
```

`la_metric_table` *Calculate a standard metric table from observed and predicted values*

Description

Calculate a standard metric table from observed and predicted values

Usage

```
la_metric_table(
  observed,
  predicted,
  n_parameters = NA_integer_,
  model_object = NULL,
  digits = 4
)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.
`n_parameters`	Optional number of estimated parameters.
`model_object`	Optional fitted model object used to extract AIC and BIC.
`digits`	Number of decimal places for rounding.

Value

A one-row data.frame.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_metric_table(obs, pred)
```

`la_mixed_coefficients` *Extract coefficients from a mixed model*

Description

Extract coefficients from a mixed model

Usage

```
la_mixed_coefficients(fit_object, model_id)
```

Arguments

`fit_object`	Object returned by `la_fit_mixed_models()`.
`model_id`	Character string with the model identifier.

Value

A data.frame with fixed-effect coefficients.

Examples

```
data(leafarea_sample)
fit <- la_fit_mixed_models(leafarea_sample, group_var = "species")
la_mixed_coefficients(fit, names(fit$models)[1])
```

`la_mixed_fitted_values` *Extract fitted values from mixed-model results*

Description

Extract fitted values from mixed-model results

Usage

```
la_mixed_fitted_values(fit_object, model_id)
```

Arguments

`fit_object`	Object returned by `la_fit_mixed_models()`.
`model_id`	Character string with the model identifier.

Value

A data.frame with observed, fitted, residual, and group values.

Examples

```
data(leafarea_sample)
fit <- la_fit_mixed_models(leafarea_sample, group_var = "species")
head(la_mixed_fitted_values(fit, names(fit$models)[1]))
```

`la_mixed_formulas` *List default mixed-model formulas*

Description

Returns the default set of candidate mixed-model formulas used by `la_fit_mixed_models()`.

Usage

```
la_mixed_formulas(
  group_var,
  random_slope = FALSE,
  include_multiple = TRUE,
  include_polynomial = TRUE
)
```

Arguments

`group_var`	Character string with the grouping variable name.
`random_slope`	Logical. If `TRUE`, includes selected random-slope formulations.
`include_multiple`	Logical. If `TRUE`, includes multiple fixed-effect mixed models.
`include_polynomial`	Logical. If `TRUE`, includes quadratic and cubic mixed models based on derived variables.

Value

A named list of formulas.

Examples

```
names(la_mixed_formulas("species"))
```

`la_mse` *Calculate mean squared error*

Description

Calculate mean squared error

Usage

```
la_mse(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_mse(obs, pred)
```

`la_nonlinear_coefficients` *Return coefficients from a selected nonlinear model*

Description

Return coefficients from a selected nonlinear model

Usage

```
la_nonlinear_coefficients(fit_object, model_id)
```

Arguments

`fit_object`	Object returned by `la_fit_nonlinear_models()`.
`model_id`	Character string with the model identifier.

Value

A data.frame with parameter estimates.

Examples

```
data(leafarea_sample)
fit <- la_fit_nonlinear_models(leafarea_sample, models = c("power_LW"))
la_nonlinear_coefficients(fit, "power_LW")
```

`la_nonlinear_fitted_values` *Extract observed, fitted values and residuals for a selected nonlinear model*

Description

Extract observed, fitted values and residuals for a selected nonlinear model

Usage

```
la_nonlinear_fitted_values(fit_object, model_id)
```

Arguments

`fit_object`	Object returned by `la_fit_nonlinear_models()`.
`model_id`	Character string with the model identifier.

Value

A data.frame with observed, fitted, and residual values.

Examples

```
data(leafarea_sample)
fit <- la_fit_nonlinear_models(leafarea_sample, models = c("power_LW"))
head(la_nonlinear_fitted_values(fit, "power_LW"))
```

`la_nonlinear_specs` *Default nonlinear model specifications*

Description

Returns the default built-in nonlinear candidate specifications.

Usage

```
la_nonlinear_specs()
```

Value

A named list of nonlinear model specifications.

Examples

```
names(la_nonlinear_specs())
```

`la_nse` *Calculate Nash-Sutcliffe efficiency*

Description

Calculate Nash-Sutcliffe efficiency

Usage

```
la_nse(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_nse(obs, pred)
```

`la_plot_observed_predicted` *Observed versus predicted leaf area plot*

Description

Creates a plot comparing observed and predicted leaf area values.

Usage

```
la_plot_observed_predicted(
  observed,
  predicted,
  model_name = "Selected model",
  point_size = 2.2,
  alpha = 0.75
)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.
`model_name`	Character. Label used in the plot title.
`point_size`	Numeric. Point size.
`alpha`	Numeric. Point transparency.

Value

A `ggplot` object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  data(leafarea_sample)
  fit <- la_fit_linear_models(leafarea_sample)
  vals <- la_linear_fitted_values(fit, model_id = "lm_LW")
  p <- la_plot_observed_predicted(vals$observed, vals$fitted, model_name = "lm_LW")
  print(p)
}
```

`la_plot_residual_histogram`

Histogram of residuals

Description

Creates a histogram of residuals from observed and predicted values.

Usage

```
la_plot_residual_histogram(
  observed,
  predicted,
  bins = 30,
  model_name = "Selected model"
)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.
`bins`	Number of histogram bins.
`model_name`	Character. Label used in the plot title.

Value

A `ggplot` object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  data(leafarea_sample)
  fit <- la_fit_linear_models(leafarea_sample)
  vals <- la_linear_fitted_values(fit, model_id = "lm_LW")
  p <- la_plot_residual_histogram(vals$observed, vals$fitted, model_name = "lm_LW")
  print(p)
}
```

`la_plot_residual_qq` *QQ plot of residuals*

Description

Creates a QQ plot of residuals from observed and predicted values.

Usage

```
la_plot_residual_qq(observed, predicted, model_name = "Selected model")
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.
`model_name`	Character. Label used in the plot title.

Value

A `ggplot` object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  data(leafarea_sample)
  fit <- la_fit_linear_models(leafarea_sample)
  vals <- la_linear_fitted_values(fit, model_id = "lm_LW")
  p <- la_plot_residual_qq(vals$observed, vals$fitted, model_name = "lm_LW")
  print(p)
}
```

`la_plot_residuals` *Residuals versus fitted values plot*

Description

Creates a residual diagnostic plot from observed and predicted values.

Usage

```
la_plot_residuals(
  observed,
  predicted,
  model_name = "Selected model",
  point_size = 2.2,
  alpha = 0.75
)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.
`model_name`	Character. Label used in the plot title.
`point_size`	Numeric. Point size.
`alpha`	Numeric. Point transparency.

Value

A `ggplot` object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  data(leafarea_sample)
  fit <- la_fit_linear_models(leafarea_sample)
  vals <- la_linear_fitted_values(fit, model_id = "lm_LW")
  p <- la_plot_residuals(vals$observed, vals$fitted, model_name = "lm_LW")
  print(p)
}
```

`la_plot_scatter` *Scatter plot between two selected variables*

Description

Creates a scatter plot for two selected variables in a leaf area dataset.

Usage

```
la_plot_scatter(
  data,
  x,
  y = "LA",
  color_var = NULL,
  add_smooth = TRUE,
  point_size = 2.2,
  alpha = 0.75
)
```

Arguments

`data`	A data.frame containing the selected variables.
`x`	Character. Name of the x-axis variable.
`y`	Character. Name of the y-axis variable. Default is `"LA"`.
`color_var`	Optional character. Grouping variable used for point color.
`add_smooth`	Logical. If `TRUE`, adds a linear trend line.
`point_size`	Numeric. Point size.
`alpha`	Numeric. Point transparency.

Value

A `ggplot` object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  data(leafarea_sample)
  p <- la_plot_scatter(leafarea_sample, x = "L", y = "LA")
  print(p)
}
```

`la_plot_scatter_set` *Scatter plots for multiple selected predictors against leaf area*

Description

Creates a list of scatter plots using the selected predictors against the response variable.

Usage

```
la_plot_scatter_set(
  data,
  predictors = c("L", "W", "LW"),
  response = "LA",
  add_smooth = TRUE,
  point_size = 2.2,
  alpha = 0.75
)
```

Arguments

`data`	A data.frame containing the selected variables.
`predictors`	Character vector of predictor names.
`response`	Character. Name of the response variable.
`add_smooth`	Logical. If `TRUE`, adds a linear trend line.
`point_size`	Numeric. Point size.
`alpha`	Numeric. Point transparency.

Value

A named list of `ggplot` objects.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  data(leafarea_sample)
  dat <- la_create_derived(leafarea_sample, variables = c("LW"))
  plots <- la_plot_scatter_set(dat, predictors = c("L", "W", "LW"))
  print(plots[[1]])
}
```

`la_predict_from_results` *Predict using one selected model from a fit object*

Description

Predict using one selected model from a fit object

Usage

```
la_predict_from_results(
  fit_object,
  model_id = 1,
  newdata,
  allow_new_levels = TRUE,
  re_form = NULL
)
```

Arguments

`fit_object`	A fitted-model result object containing `models`.
`model_id`	Model position or model name.
`newdata`	A data.frame for prediction.
`allow_new_levels`	Logical used for mixed models.
`re_form`	Optional random-effects structure used for mixed models.

Value

A data.frame.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
newdata_ex <- leafarea_sample[1:5, c("L", "W", "LA")]
la_predict_from_results(fit, model_id = "lm_LW", newdata = newdata_ex)
```

`la_predict_linear_model` *Predict from a linear model*

Description

Predict from a linear model

Usage

```
la_predict_linear_model(model, newdata)
```

Arguments

`model`	An object of class `lm`.
`newdata`	A data.frame for prediction.

Value

A data.frame.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
newdata_ex <- leafarea_sample[1:5, c("L", "W", "LA")]
la_predict_linear_model(fit$models[["lm_LW"]], newdata_ex)
```

`la_predict_mixed_model` *Predict from a mixed model*

Description

Predict from a mixed model

Usage

```
la_predict_mixed_model(model, newdata, allow_new_levels = TRUE, re_form = NULL)
```

Arguments

`model`	An object of class `lmerMod`.
`newdata`	A data.frame for prediction.
`allow_new_levels`	Logical used for mixed models.
`re_form`	Optional random-effects structure used for mixed models.

Value

A data.frame.

Examples

```
data(leafarea_sample)
fit <- la_fit_mixed_models(leafarea_sample, group_var = "species")
newdata_ex <- leafarea_sample[1:5, c("L", "W", "LA", "species")]
la_predict_mixed_model(fit$models[[names(fit$models)[1]]], newdata_ex)
```

`la_predict_model` *Predict from a fitted model*

Description

Generic dispatcher for prediction from linear, nonlinear, or mixed models.

Usage

```
la_predict_model(
  model,
  newdata,
  model_type = c("auto", "linear", "nonlinear", "mixed"),
  allow_new_levels = TRUE,
  re_form = NULL
)
```

Arguments

`model`	A fitted model object.
`newdata`	A data.frame for prediction.
`model_type`	One of `"auto"`, `"linear"`, `"nonlinear"`, or `"mixed"`.
`allow_new_levels`	Logical used for mixed models.
`re_form`	Optional random-effects structure used for mixed models.

Value

A data.frame containing the prediction columns.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
newdata_ex <- leafarea_sample[1:5, c("L", "W", "LA")]
la_predict_model(fit$models[["lm_LW"]], newdata_ex)
```

`la_predict_nonlinear_model` *Predict from a nonlinear model*

Description

Predict from a nonlinear model

Usage

```
la_predict_nonlinear_model(model, newdata)
```

Arguments

`model`	An object of class `nls`.
`newdata`	A data.frame for prediction.

Value

A data.frame.

Examples

```
data(leafarea_sample)
fit <- la_fit_nonlinear_models(leafarea_sample, models = c("power_LW"))
newdata_ex <- leafarea_sample[1:5, c("L", "W", "LA")]
la_predict_nonlinear_model(fit$models[["power_LW"]], newdata_ex)
```

`la_predict_top_ranked` *Predict from the top-ranked model*

Description

Predict from the top-ranked model

Usage

```
la_predict_top_ranked(
  ranked_table,
  fit_object,
  rank_position = 1,
  newdata,
  allow_new_levels = TRUE,
  re_form = NULL
)
```

Arguments

`ranked_table`	A ranked data.frame containing `model_id`.
`fit_object`	A fitted-model result object containing `models`.
`rank_position`	Row position within `ranked_table`.
`newdata`	A data.frame for prediction.
`allow_new_levels`	Logical used for mixed models.
`re_form`	Optional random-effects structure used for mixed models.

Value

A data.frame.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
met <- la_evaluate_linear_models(fit)
ranked <- la_rank_models(met)
newdata_ex <- leafarea_sample[1:5, c("L", "W", "LA")]
la_predict_top_ranked(
  ranked,
  fit,
  rank_position = 1,
  newdata = newdata_ex
)
```

`la_r` *Calculate Pearson correlation coefficient*

Description

Calculate Pearson correlation coefficient

Usage

```
la_r(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_r(obs, pred)
```

`la_r_squared` *Calculate coefficient of determination*

Description

Calculate coefficient of determination

Usage

```
la_r_squared(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_r_squared(obs, pred)
```

`la_rank_models` *Rank models using a simple metric priority rule*

Description

Orders models according to a sequence of evaluation criteria.

Usage

```
la_rank_models(
  metrics_table,
  sort_by = c("RMSE", "MAE", "CCC", "R2", "ABS_BIAS"),
  ascending = c(TRUE, TRUE, FALSE, FALSE, TRUE)
)
```

Arguments

`metrics_table`	A data.frame containing model metrics.
`sort_by`	Character vector with metric names used for ordering.
`ascending`	Logical vector indicating whether each metric should be sorted in ascending order.

Value

A ranked data.frame.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
met <- la_evaluate_linear_models(fit)
la_rank_models(met)
```

`la_rank_models_by_metrics`

Rank models by average metric positions

Description

Computes metric-wise ranks and aggregates them using the mean rank.

Usage

```
la_rank_models_by_metrics(metrics_table)
```

Arguments

`metrics_table`	A data.frame containing model metrics.

Value

A ranked data.frame.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
met <- la_evaluate_linear_models(fit)
la_rank_models_by_metrics(met)
```

`la_rank_models_weighted`

Rank models using a weighted score

Description

Computes a weighted score from selected metrics after min-max scaling.

Usage

```
la_rank_models_weighted(
  metrics_table,
 weights = list(RMSE = 0.3, MAE = 0.2, CCC = 0.2, R2 = 0.15, ABS_BIAS = 0.1, d = 0.05)
)
```

Arguments

`metrics_table`	A data.frame containing model metrics.
`weights`	Named list of metric weights.

Value

A ranked data.frame.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
met <- la_evaluate_linear_models(fit)
la_rank_models_weighted(met)
```

`la_rmse` *Calculate root mean squared error*

Description

Calculate root mean squared error

Usage

```
la_rmse(observed, predicted)
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.

Value

A numeric value.

Examples

```
obs <- c(10, 12, 15, 18)
pred <- c(9.8, 12.1, 14.7, 18.4)
la_rmse(obs, pred)
```

`la_top_models` *Select the top models from a ranking table*

Description

Returns the first `n` rows from a ranked table.

Usage

```
la_top_models(
  ranking_table,
  n = 5,
  rank_column = c("rank_simple", "rank_weighted", "rank_mean")
)
```

Arguments

`ranking_table`	A ranked data.frame.
`n`	Number of rows to return.
`rank_column`	Column used to order the rows.

Value

A data.frame.

Examples

```
data(leafarea_sample)
fit <- la_fit_linear_models(leafarea_sample)
met <- la_evaluate_linear_models(fit)
ranked <- la_rank_models(met)
la_top_models(ranked, n = 3)
```

`la_validate_input` *Validate and standardize input data for leaf area analysis*

Description

Checks that the selected leaf length, leaf width, and observed leaf area columns are present and numeric, applies optional cleaning rules, and standardizes their names to `L`, `W`, and `LA` for downstream use in `leafareaR`.

Usage

```
la_validate_input(
  data,
  l_col = "L",
  w_col = "W",
  la_col = "LA",
  remove_na = TRUE,
  remove_nonpositive = TRUE,
  standardize_names = TRUE,
  keep_all_columns = FALSE
)
```

Arguments

`data`	A data.frame containing at least the columns `L`, `W`, and `LA`, or equivalent columns selected through `l_col`, `w_col`, and `la_col`.
`l_col`	Character. Name of the column containing leaf length.
`w_col`	Character. Name of the column containing leaf width.
`la_col`	Character. Name of the column containing observed leaf area.
`remove_na`	Logical. If `TRUE`, rows with missing values in the selected columns are removed.
`remove_nonpositive`	Logical. If `TRUE`, rows with values less than or equal to zero in the selected columns are removed.
`standardize_names`	Logical. If `TRUE`, the selected columns are renamed internally to `L`, `W`, and `LA` in the returned object.
`keep_all_columns`	Logical. If `TRUE`, keeps all original columns in the returned data.frame while standardizing the selected measurement columns.

Value

A validated data.frame ready for descriptive analysis, model fitting, prediction, and visualization in the `leafareaR` workflow.

Examples

```
data(leafarea_sample)
validated_data <- la_validate_input(leafarea_sample)
head(validated_data)

validated_with_groups <- la_validate_input(
  data = leafarea_sample,
  keep_all_columns = TRUE
)
head(validated_with_groups)
```

`leafarea_sample` *Example dataset for leaf area modeling*

Description

A sample dataset included in `leafareaR` for testing data validation, descriptive statistics, derived variables, plotting, linear models, nonlinear models, mixed models, ranking, and prediction.

Usage

```
data(leafarea_sample)
```

Format

A data.frame with 9999 rows and 6 variables:

L Leaf length.

W Leaf width.

LA Observed leaf area.

species Species identifier.

block Block identifier.

genotype Genotype identifier.

Examples

```
data(leafarea_sample)
head(leafarea_sample)
```

`run_leafareaR_app` *Launch the built-in Shiny application*

Description

Opens the interactive `leafareaR` Shiny app, which provides a graphical interface for loading example data or uploading user data, exploratory analysis, model fitting, evaluation, ranking, and prediction.

Usage

```
run_leafareaR_app(...)
```

Arguments

`...`	Additional arguments passed to `shiny::runApp()`.

Value

Launches the application.

Examples

```
app_dir <- system.file("shiny", "leafareaR-app", package = "leafareaR")
dir.exists(app_dir)
if (interactive()) {
  run_leafareaR_app()
}
```

Index