Package: agriReg (via r-universe)

June 1, 2026

Title Linear and Nonlinear Regression for Agricultural Data

Version 0.1.0

Description Fit, compare, and visualise linear and nonlinear regression models tailored to field-trial and dose-response agricultural data. Provides S3 classes for mixed-effects models (via 'lme4'), nonlinear growth curves (logistic, 'Gompertz', asymptotic, linear-plateau, quadratic), and four/five-parameter log-logistic dose-response models (via 'drc'). Includes automated starting-value heuristics, goodness-of-fit statistics, residual diagnostics, and 'ggplot2'-based visualisation. Methods are based on Bates and Watts (1988, ISBN:9780471816430), Ritz and others (2015) <doi:10.1371/journal.pone.0146021>, and Bates and others (2015) <doi:10.18637/jss.v067.i01>.

Depends R (>= 4.1.0)

Imports lme4 (>= 1.1-35), drc (>= 3.0-1), ggplot2 (>= 3.4.0), patchwork (>= 1.1.0), stats, utils

Suggests broom (>= 1.0.0), broom.mixed (>= 0.2.9), emmeans (>= 1.8.0), nlme (>= 3.1-0), methods, testthat (>= 3.0.0), knitr, rmarkdown, lmerTest, car, multcomp

License MIT + file LICENSE

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

VignetteBuilder knitr

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 2026-03-31 15:40:35 UTC
RemoteUrl https://github.com/cran/agriReg
RemoteRef HEAD
RemoteSha d86450b33c737e87ed7e4c3176711e39aeb13c51

Contents

`agriReg-package` *agriReg: Linear and Nonlinear Regression for Agricultural Data*

Description

`agriReg` provides a unified interface for fitting, comparing, and visualising regression models commonly used in agricultural research:

- **Linear & mixed-effects models** via `fit_linear()` (wraps `lm` / `lme4::lmer`)
- **Nonlinear growth curves** via `fit_nonlinear()` — logistic, Gompertz, asymptotic, linear-plateau, quadratic
- **Dose-response models** via `fit_dose_response()` — 4/5-parameter log-logistic and Weibull (wraps `drc`)
- **Model comparison** via `compare_models()` — AIC, BIC, RMSE side by side
- **Diagnostics** via `gof_stats()`, `residual_check()`, and S3 `plot()` methods

Typical workflow

```
library(agriReg)

# 1. Clean field trial data
trial <- clean_agri_data(my_data, yield_col = "yield_tha")

# 2. Linear mixed model (treatment + block structure)
lm_fit <- fit_linear(trial,
                     formula = "yield_tha ~ nitrogen + phosphorus",
                     random  = "(1|block)")
summary(lm_fit)
plot(lm_fit, type = "residuals")

# 3. Nonlinear growth curve
nl_fit <- fit_nonlinear(trial, x_col = "days", y_col = "biomass",
                        model = "logistic")
plot(nl_fit)

# 4. Dose-response
dr_fit <- fit_dose_response(herb_data,
                            dose_col = "dose_g_ha",
                            resp_col = "weed_control_pct")
ed_estimates(dr_fit, respLev = c(50, 90))

# 5. Compare models
compare_models(linear = lm_fit, nonlinear = nl_fit)
```

Author(s)

Maintainer: Sadikul Islam <sadikul.islamiasri@gmail.com> (ORCID)

---

clean_agri_data *Clean and validate agricultural field-trial data*

Description

Removes rows with missing yield values, coerces the yield column to numeric, optionally flags IQR-based outliers, and returns an `agriData` object with a concise summary.

Usage

```
clean_agri_data(
  df,
  yield_col = "yield",
  flag_outliers = TRUE,
  na_action = c("remove", "warn")
)
```

Arguments

df	A data.frame containing field trial records.
yield_col	Character. Name of the column that holds the primary response variable (e.g. "yield", "biomass"). Default "yield".
flag_outliers	Logical. When TRUE (default), a logical column .outlier is appended: TRUE marks observations that fall more than 1.5 x IQR below Q1 or above Q3.
na_action	One of "remove" (default) or "warn". "remove" drops rows with NA in yield_col; "warn" keeps them but issues a warning.

Value

A data.frame with additional class "agriData". Attributes n_removed (rows dropped) and n_flagged (outliers flagged) are attached.

Examples

```
df <- data.frame(
  yield = c(3.1, 4.2, NA, 8.9, 3.8, 100),
  block = c("A", "A", "B", "B", "C", "C")
)
clean_agri_data(df, yield_col = "yield")
```

---

compare_models *Compare multiple agriReg models side by side*

Description

Computes AIC, BIC, RMSE, MAE, and R^2 for a collection of agriLM, agriNLS, or agriDRC objects and returns them in a tidy data frame ranked by AIC.

Usage

```
compare_models(...)
```

Arguments

...	Named model objects. Names are used as row identifiers. If unnamed, models are labelled model1, model2, ...

Value

A data.frame with columns model, engine, n_par, AIC, BIC, RMSE, MAE, R2, and delta_AIC (difference from best).

Examples

```
dat <- data.frame(
  nitrogen = rep(c(0, 40, 80, 120, 160), each = 8),
  yield    = c(2.1, 2.8, 3.6, 4.1, 4.3,
               2.0, 2.9, 3.5, 4.2, 4.4,
               2.2, 2.7, 3.7, 4.0, 4.5,
               2.1, 3.0, 3.4, 4.3, 4.2,
               2.3, 2.8, 3.6, 4.1, 4.3,
               2.0, 2.9, 3.5, 4.2, 4.4,
               2.2, 2.7, 3.7, 4.0, 4.5,
               2.1, 3.0, 3.4, 4.3, 4.2)
)
m1 <- fit_linear(dat, "yield ~ nitrogen")
m2 <- fit_nonlinear(dat, "nitrogen", "yield", "quadratic")
compare_models(linear = m1, quadratic = m2)
```

`ed_estimates` *Compute effective dose (ED) estimates from a dose-response model*

Description

Returns the dose required to achieve a specified percentage of the maximum response, with confidence intervals via the delta method.

Usage

```
ed_estimates(
  drc_fit,
  respLev = c(10, 50, 90),
  interval = "delta",
  level = 0.95
)
```

Arguments

`drc_fit`	An `agriDRC` object.
`respLev`	Numeric vector of response levels (default `c(10, 50, 90)`). E.g. `50` = ED50 (dose for 50% of maximum effect).
`interval`	Type of confidence interval: `"delta"` (default) or `"none"`.
`level`	Confidence level (default `0.95`).

Value

A matrix of ED estimates and confidence bounds (printed and returned invisibly).

`fit_dose_response` *Fit a log-logistic or Weibull dose-response model*

Description

A wrapper around `drc::drm()` providing the four model families most commonly used in herbicide/pesticide and nutrient-response studies. Returns an `agriDRC` object with `print`, `summary`, and `plot` methods.

Usage

```
fit_dose_response(
  data,
  dose_col,
  resp_col,
  fct = c("LL.4", "LL.5", "W1.4", "W2.4"),
  curveid = NULL,
  ...
)
```

Arguments

`data`	A data frame containing dose and response columns.
`dose_col`	Character. Name of the dose/concentration column.
`resp_col`	Character. Name of the response column (e.g. percentage inhibition, weed biomass reduction, germination rate).
`fct`	Character. Model function:
	`"LL.4"` 4-parameter log-logistic (default). Parameters: b, c, d, e.
	`"LL.5"` 5-parameter log-logistic (asymmetry parameter f added).
	`"W1.4"` 4-parameter Weibull type-1.
	`"W2.4"` 4-parameter Weibull type-2.
`curveid`	Optional column name for grouping curves (e.g. `"species"` or `"treatment"`). Passed directly to `drc::drm()`.
`...`	Additional arguments forwarded to `drc::drm()`.

Value

An object of class `"agriDRC"` containing:

fit The underlying `drc` object.

fct The model function name.

dose_col, resp_col Column names used.

data The data used.

call The matched call.

Examples

```
herbicide_trial <- load_example_data("herbicide_trial")
dr <- fit_dose_response(herbicide_trial,
                        dose_col = "dose_g_ha",
                        resp_col = "weed_control_pct")
summary(dr)
ed_estimates(dr, respLev = c(10, 50, 90))
plot(dr)
```

---

fit_linear *Fit a linear or mixed-effects model for agricultural trials*

Description

A wrapper around `stats::lm()` and `lme4::lmer()` that returns an `agriLM` object with consistent `print`, `summary`, `plot`, `coef`, `fitted`, `residuals`, and `predict` S3 methods.

Usage

```
fit_linear(data, formula, random = NULL, weights = NULL, ...)
```

Arguments

`data`	A data frame or `agriData` object.
`formula`	A model formula as a character string or formula object. Example: `"yield ~ treatment + block"`.
`random`	Optional character string specifying the random-effects term in `lme4` syntax, e.g. `"(1|block)"` or `"(1|block) + (1|location)"`. When supplied `lme4::lmer()` is used; otherwise `stats::lm()`.
`weights`	Optional numeric vector of observation weights passed to `lm()` or `lmer()`.
`...`	Additional arguments forwarded to `lm()` or `lmer()`.

Value

An object of class `"agriLM"` (a named list) containing:

fit The underlying `lm` or `lmerMod` object.

engine Character: `"lm"` or `"lmer"`.

formula The fixed-effects formula.

random The random-effects term (or `NULL`).

data The data used for fitting.

call The matched call.

Examples

```
wheat_trial <- load_example_data("wheat_trial")
# Ordinary least squares
m1 <- fit_linear(wheat_trial, "yield ~ nitrogen + phosphorus")

# Mixed model with block as random effect
m2 <- fit_linear(wheat_trial,
                 "yield ~ nitrogen + phosphorus",
                 random = "(1|block)")
summary(m2)
plot(m2, type = "residuals")
```

`fit_nonlinear` *Fit a nonlinear growth or response curve to agricultural data*

Description

Provides a unified interface for five model families commonly used in agronomic research. Automatic starting-value heuristics are applied when `start` is not supplied, making the function usable without prior knowledge of parameter ranges.

Usage

```
fit_nonlinear(
  data,
  x_col,
  y_col,
  model = c("logistic", "gompertz", "asymptotic", "linear_plateau", "quadratic"),
  start = NULL,
  control = nls.control(maxiter = 500, tol = 1e-06)
)
```

Arguments

`data`	A data frame (or `agriData`) with at least `x_col` and `y_col`.
`x_col`	Character. Name of the independent variable column (e.g. `"days"`, `"dose"`, `"nitrogen_kg_ha"`).
`y_col`	Character. Name of the response column (e.g. `"yield"`, `"biomass"`, `"plant_height"`).
`model`	One of: `"logistic"` 3-parameter logistic: $y = Asym/(1 + exp((xmid - x)/scal))$ `"gompertz"` 3-parameter Gompertz: $y = Asym \cdot \exp(-\exp(b2(b3 - x)))$ `"asymptotic"` Asymptotic / Mitscherlich: $y = Asym(1 - \exp(-rate \cdot x))$ `"linear_plateau"` Linear-plateau (broken-stick): $y = a + b \cdot \min(x, cp)$ `"quadratic"` Quadratic polynomial: $y = a + b \cdot x + c \cdot x^2$

start	Optional named list of starting parameter values. When NULL (default), heuristic starting values are derived from the data.
control	A list passed to stats::nls.control(). Increase maxiter if convergence fails.

Value

An object of class "agriNLS", a named list containing:

fit The nls object.

model Character: the model name.

x_col, y_col Column names used.

data The data used for fitting.

call The matched call.

Examples

```
maize_growth <- load_example_data("maize_growth")
nl <- fit_nonlinear(maize_growth, x_col = "days", y_col = "biomass_g",
                    model = "logistic")
summary(nl)
plot(nl)
```

---

fit_polynomial *Fit a sequence of polynomial linear models and pick the best*

Description

Convenience function for testing linear, quadratic, and cubic fits of a single continuous predictor and selecting the best by AIC.

Usage

```
fit_polynomial(data, x_col, y_col, max_degree = 3)
```

Arguments

data	A data frame.
x_col	Predictor column name.
y_col	Response column name.
max_degree	Maximum polynomial degree to test (default 3).

Value

An agriLM object for the best-fitting model. A comparison table is printed as a side-effect.

Examples

```
wheat_trial <- load_example_data("wheat_trial")
best <- fit_polynomial(wheat_trial, x_col = "nitrogen", y_col = "yield")
plot(best)
```

`gof_stats` *Compute goodness-of-fit statistics for a model*

Description

Extracts R^2, adjusted-R^2, RMSE, MAE, AIC, and BIC from an `agriLM`, `agriNLS`, or `agriDRC` object (or any object with `fitted()`, `residuals()`, and `AIC()` methods).

Usage

```
gof_stats(model)
```

Arguments

`model`	An `agriLM`, `agriNLS`, `agriDRC`, or compatible model object.

Value

A named list with elements `R2`, `adj_R2`, `RMSE`, `MAE`, `AIC`, and `BIC`.

`herbicide_trial` *Simulated herbicide dose-response dataset*

Description

Weed control percentage measured across seven herbicide doses for two weed species.

Format

A data frame with 84 rows and 4 variables:

species Weed species (`Amaranth`, `Ryegrass`).

dose_g_ha Herbicide dose (g active ingredient per hectare).

weed_control_pct Weed control percentage (0–100).

rep Replicate number (1–6).

Source

Simulated for package demonstration.

load_example_data *Load a bundled agriReg example dataset*

Description

Reads one of the three example datasets shipped with the package from `inst/extdata`. Under normal installation the datasets are available directly via `data(wheat_trial)` etc; this function is a fallback that works even when `.rda` files have not been compiled.

Usage

```
load_example_data(name = c("wheat_trial", "maize_growth", "herbicide_trial"))
```

Arguments

`name`	One of `"wheat_trial"`, `"maize_growth"`, or `"herbicide_trial"`.

Value

A `data.frame`.

Examples

```
wheat <- load_example_data("wheat_trial")
head(wheat)
```

maize_growth *Simulated maize growth time-series*

Description

Daily above-ground biomass measurements from 10 maize plants tracked from emergence to physiological maturity under well-watered conditions.

Format

A data frame with 200 rows and 4 variables:

plant_id Plant identifier (1–10).

days Days after emergence.

biomass_g Dry biomass (g/plant).

treatment Water treatment (`WW` = well-watered, `DS` = drought-stressed).

Source

Simulated for package demonstration.

model_summary *Comprehensive model summary with fit statistics*

Description

Prints a formatted block with coefficient table, goodness-of-fit statistics, and (for agriLM) an ANOVA table.

Usage

```
model_summary(model, anova = TRUE)
```

Arguments

model	An agriLM or agriNLS object.
anova	Logical. Include ANOVA table for agriLM models (default TRUE).

Value

Invisibly returns the model object (called for its side effect of printing a formatted summary block containing coefficients, goodness-of-fit statistics, and, for agriLM models fitted with engine = "lm", an ANOVA table).

optimum_dose *Find the optimal x value (e.g. economic optimum dose)*

Description

For quadratic and linear-plateau models, computes the x value that maximises the predicted response.

Usage

```
optimum_dose(nls_fit)
```

Arguments

nls_fit	An agriNLS object fitted with model = "quadratic" or "linear_plateau".

Value

A named numeric vector with x_opt (optimum x) and y_max (predicted maximum y).

outlier_flag *Flag outliers using multiple methods*

Description

A standalone outlier-detection function supporting IQR, Z-score, and modified Z-score (Iglewicz-Hoaglin) methods.

Usage

```
outlier_flag(df, col, method = c("iqr", "zscore", "modz"), threshold = NULL)
```

Arguments

df	A data frame.
col	Column name to test.
method	One of "iqr" (default), "zscore", "modz".
threshold	Numeric threshold: for "iqr" the IQR multiplier (default 1.5); for "zscore" the SD multiplier (default 3); for "modz" the modified Z-score cutoff (default 3.5).

Value

df with a logical column <col>_outlier.

plot.agriDRC *Plot a fitted dose-response curve*

Description

Plot a fitted dose-response curve

Usage

```
## S3 method for class 'agriDRC'
plot(x, log_dose = TRUE, n_points = 200, ...)
```

Arguments

x	An agriDRC object.
log_dose	Logical. Plot dose on a log10 scale (default TRUE).
n_points	Integer. Points on the smooth curve (default 200).
...	Ignored.

Value

A ggplot object (printed invisibly).

plot.agriLM *Plot diagnostics for an agriLM model*

Description

Plot diagnostics for an agriLM model

Usage

```
## S3 method for class 'agriLM'
plot(x, type = c("fit", "residuals", "qq", "scale"), ...)
```

Arguments

x	An agriLM object.
type	One of "fit" (observed vs fitted), "residuals" (residuals vs fitted), "qq" (normal Q-Q), or "scale" (sqrt(|residuals|) vs fitted). Default "fit".
...	Ignored.

Value

A ggplot object (printed invisibly).

plot.agriNLS *Plot a fitted nonlinear curve over observed data*

Description

Plot a fitted nonlinear curve over observed data

Usage

```
## S3 method for class 'agriNLS'
plot(x, n_points = 300, show_residuals = FALSE, ...)
```

Arguments

x	An agriNLS object.
n_points	Integer. Number of points on the fitted curve (default 300).
show_residuals	Logical. When TRUE, adds vertical line segments from observed points to the curve. Default FALSE.
...	Ignored.

Value

A ggplot object (printed invisibly).

plot_fit *A convenience wrapper for ggplot2 visualisation of model fit*

Description

Dispatches to `plot.agriLM`, `plot.agriNLS`, or `plot.agriDRC` depending on model class. Useful in pipeline contexts.

Usage

```
plot_fit(model, ...)
```

Arguments

`model`	An `agriLM`, `agriNLS`, or `agriDRC` object.
`...`	Arguments passed to the underlying plot method.

Value

A `ggplot` object, returned invisibly. Called primarily for the side effect of printing the plot. The exact structure of the returned object depends on the class of `model`: `plot.agriLM`, `plot.agriNLS`, or `plot.agriDRC` is dispatched accordingly.

residual_check *Residual diagnostic plot panel*

Description

Generates a 2×2 panel of residual diagnostics: residuals vs fitted, normal Q-Q, scale-location, and a histogram of residuals.

Usage

```
residual_check(model, ncol = 2)
```

Arguments

`model`	An `agriLM` or `agriNLS` object.
`ncol`	Integer. Number of plot columns (default 2).

Value

Invisibly returns the list of four `ggplot` objects.

wheat_trial *Simulated wheat nitrogen-response trial*

Description

A simulated dataset from a randomised complete block design (RCBD) testing four nitrogen rates across three blocks and two phosphorus levels.

Format

A data frame with 72 rows and 6 variables:

block Block identifier (`A`, `B`, `C`).

nitrogen Nitrogen application rate (kg/ha).

phosphorus Phosphorus level (`low`, `high`).

variety Crop variety (`V1`, `V2`).

yield Grain yield (t/ha).

biomass Above-ground dry biomass (t/ha).

Source

Simulated for package demonstration.

yield_normalize *Normalise yield values within groups*

Description

Z-score or min-max normalises a numeric column, optionally within groups (e.g. per trial site or year).

Usage

```
yield_normalize(
  df,
  yield_col = "yield",
  method = c("zscore", "minmax"),
  group_by = NULL
)
```

Arguments

`df`	A data frame.
`yield_col`	Character. Column to normalise.
`method`	`"zscore"` (default) or `"minmax"`.
`group_by`	Character vector of grouping column names, or `NULL` for global normalisation.

Value

df with a new column <yield_col>_norm.

Examples

```
df <- data.frame(yield = c(3, 4, 5, 6), site = c("A","A","B","B"))
yield_normalize(df, group_by = "site")
```

Index