Package: cndplantanalysis (via r-universe)

October 21, 2026

Type Package

Title Compositional Nutrient Diagnosis for Plant Tissue Analysis

Version 1.0.2

Description Provides an implementation of Compositional Nutrient Diagnosis (CND) for plant tissue tests. Computes scale invariant simplex filling values(Rd), row-centered log-ratios (CLR), iterative database partitioning via Cate-Nelson procedures, cubic polynomial regression of cumulative variance ratio functions to determine inflection point yield cutoffs, critical CND nutrient imbalance thresholds (r^2) using the chi-square distribution, diagnostic standardized indices (Ix) and automated Cate Nelson scatter partitioning. Includes linear, non-linear, cumulative variance, chi-square and quadrant plotting functions.Methodology and theoretical frameworks are grounded in Parent and Dafir (1992) <doi:10.21273/JASHS.117.2.239> and Khiari et al. (2001) <doi:10.2134/agronj2001.934802x>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Imports ggplot2, graphics, utils, stats

Suggests readxl, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author Blesson B. Varghese [aut, cre], Rohith A K [aut], Anjali S. [aut], Aliya Shirin K S [aut], Hilal Mohamed [aut]

Maintainer Blesson B. Varghese <blessonvarghese1234@gmail.com>

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

Date/Publication 2026-10-21 15:40:13 UTC

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

RemoteRef HEAD

RemoteSha 0dcffe2fd0ccefed1876cab26745d40b8671282e

Contents

calc_cate_nelson_empirical 2
cut_off_table 3
diagnose_specimen 5
iteration_for_hyp 7
log_centered_table 9
plant_leaf_data 10
plot_cate_nelson 11
plot_cnd_chisq 12
plot_cnd_yield_linear 13
plot_cnd_yield_nlr 15
plot_cumulative_variance 18
simplex_table 19
validation_data 20

Index **22**

`calc_cate_nelson_empirical`

Empirical Cate Nelson Threshold Calculation via ANOVA Grid Search

---

Description

Evaluates candidate split points across bivariate diagnostic data (CND r^2 vs. Crop Yield) to automatically determine critical thresholds maximizing the ANOVA Sum of Squares Between groups (SS_{between}) (Nelson & Anderson, 1977; Khiari et al., 2001).

Usage

```
calc_cate_nelson_empirical(diag_data, yield_col = "Yield")
```

Arguments

`diag_data`	Data frame output from `diagnose_specimen` containing `"r2"` and a yield column.
`yield_col`	Character string specifying the yield column name. Default is `"Yield"`.

Value

A named list containing:

`crit_r2`	Empirically optimized critical CND r^2 threshold.
`crit_yield`	Empirically optimized critical yield cutoff.

Examples

```
{
data(plant_leaf_data)

simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

lc_res <- log_centered_table(
  simplex_data = simplex_res,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
iter_res <- iteration_for_hyp(
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)
cnd_out <- cut_off_table(
  iter_data = iter_res,
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)

data(validation_data)
diag_results <- diagnose_specimen(
  new_data = validation_data,
  norms_table = cnd_out$norms_table,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

auto_thresh <- calc_cate_nelson_empirical(diag_results, yield_col = "Yield")
print(auto_thresh)
}
```

`cut_off_table` *Fit Cubic Models, Determine Yield Cutoff, and Extract CND High-Yield Norms*

Description

Fits a cubic polynomial model $F_i^C(V_X) = aY^3 + bY^2 + cY + d$ for each component, calculates the inflection point yield cutoff ($-b/3a$), determines the maximum yield target, computes the low-yielding population percentage, and returns high-yield (`hyp`) and low-yield (`lyp`) CND norms (Khiari et al., 2001).

Usage

```
cut_off_table(iter_data, lc_data, yield_col, nutrient_cols, digits = 2)
```

Arguments

`iter_data`	Data frame produced by `iteration_for_hyp`.
`lc_data`	Data frame produced by `log_centered_table`.
`yield_col`	Character string specifying the crop yield column name.
`nutrient_cols`	Character vector specifying selected nutrient names.
`digits`	Numeric integer specifying decimal rounding precision. Default is `2`.

Details

Setting the second derivative of the cubic function to zero yields the inflection point:

$$\frac{\partial^2 F_i^C(V_X)}{\partial Y^2} = 6aY + 2b = 0 \implies Y_{\text{cutoff}} = -\frac{b}{3a}$$

The highest inflection point across all components is retained as the minimum yield target. The mean (V_X^*) and standard deviation (SD_X^*) of row-centered log ratios within the high-yield subpopulation ($Y \geq Y_{\text{cutoff}}$) form the baseline CND norms.

Value

A named list containing two data frames:

`cut_off_table`	Polynomial coefficients (a, b, c, d) and calculated inflection points for each component.
`norms_table`	CND reference norms containing `hyp mean`, `hyp sd`, `lyp mean`, and `lyp sd`.

Examples

```
{

data(plant_leaf_data)

simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
```

```
lc_res <- log_centered_table(
  simplex_data = simplex_res,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
iter_res <- iteration_for_hyp(
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)
cnd_out <- cut_off_table(
  iter_data = iter_res,
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)
print(cnd_out$cut_off_table)
print(cnd_out$ow_yield_percentage)
print(cnd_out$norms_table)
}
```

`diagnose_specimen` *Diagnose Plant Tissue Specimens using CND Norms*

Description

Standardizes new plant tissue samples against high-yield CND reference norms (V_X^* and SD_X^*), computing individual CND nutrient indices (I_X) and the global nutrient imbalance index (r^2).

Usage

```
diagnose_specimen(
  new_data,
  norms_table,
  sample_col,
  nutrient_cols,
  yield_col = NULL,
  digits = 2
)
```

Arguments

`new_data`	Data frame containing new specimen concentrations and sample identifiers.
`norms_table`	Data frame output from `cut_off_table` containing high-yield norms (`hyp mean` and `hyp sd`).

sample_col	Character string specifying the sample identifier column name.
nutrient_cols	Character vector specifying selected nutrient names.
yield_col	Optional character string specifying the yield column name if present in new_data. Default is NULL.
digits	Numeric integer for decimal rounding precision. Default is 2.

Details

Individual CND indices are calculated as standardized score deviations from high-yield norms:

$$I_X = \frac{V_X - V_X^*}{SD_X^*}$$

The global nutrient imbalance index (r^2) sums the squared indices across all $d + 1$ components:

$$r^2 = \sum_{i=1}^{d+1} I_{X_i}^2$$

Value

A data frame containing:

sample_col	Sample identifiers.
yield_col	Crop yield values (if provided).
nutrient_cols	Nutrient concentrations.
Rd	Residual filling value R_d.
G	Geometric mean G.
Vx	Row-centered log ratios.
Ix	Individual CND nutrient indices.
r2	Global nutrient imbalance index r^2.

Examples

```
{
data(plant_leaf_data)
simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

lc_res <- log_centered_table(
  simplex_data = simplex_res,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
```

```
iter_res <- iteration_for_hyp(
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)
cnd_out <- cut_off_table(
  iter_data = iter_res,
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)

data(validation_data)

diag_results <- diagnose_specimen(
  new_data = validation_data,
  norms_table = cnd_out$norms_table,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
print(diag_results)
}
```

`iteration_for_hyp` *Iterative Partitioning and Variance Ratio Calculation for CND*

Description

Iteratively partitions a yield-ordered dataset into two subpopulations (n_1 high-yielding and n_2 low-yielding specimens) across $n - 3$ steps, computing the variance ratio function $f_i(V_X)$ and the cumulative variance ratio function $F_i^C(V_X)$ for each component (Khiari et al., 2001).

Usage

```
iteration_for_hyp(lc_data, yield_col, nutrient_cols, digits = 2)
```

Arguments

`lc_data`	Data frame output from `log_centered_table`.
`yield_col`	Character string specifying the crop yield column name.
`nutrient_cols`	Character vector specifying selected nutrient names.
`digits`	Numeric integer specifying decimal rounding precision. Default is 2.

Details

At each iteration i (from $n_1 = 2$ to $n_1 = n - 2$), the ratio function $f_i(V_X)$ is evaluated as:

$$f_i(V_X) = \frac{\text{Var}(V_X \text{ of } n_2 \text{ low-yield observations})}{\text{Var}(V_X \text{ of } n_1 \text{ high-yield observations})}$$

The cumulative variance ratio function $F_i^C(V_X)$ scales the accumulated variance ratios on a zero to hundred percentage scale:

$$F_i^C(V_X) = \frac{\sum_{j=1}^{i} f_j(V_X)}{\sum_{j=1}^{n-3} f_j(V_X)} \times 100$$

Value

A data frame containing:

`iteration`	Iteration index (1 to $n - 3$).
`n1`	Number of observations in the high-yield subpopulation.
`n2`	Number of observations in the low-yield subpopulation.
`fi(VX)`	Variance ratio functions for each nutrient and filling value R_d.
`Fi(VX)`	Cumulative variance ratio functions ($F_i^C(V_X)$) expressed as percentages.

Examples

```
{
data(plant_leaf_data)
simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

lc_res <- log_centered_table(
  simplex_data = simplex_res,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
iter_res <- iteration_for_hyp(
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)
}
```

log_centered_table *Generate Log-Centered Ratio Table for CND*

Description

Computes row-centered log ratios (V_X) for all $d+1$ components in the simplex S^d by taking the natural logarithm of each component proportion divided by the geometric mean G (Aitchison, 1986; Parent & Dafir, 1992; Khiari et al., 2001).

Usage

```
log_centered_table(simplex_data, sample_col, yield_col, nutrient_cols)
```

Arguments

`simplex_data`	A data frame produced by `simplex_table` containing sample identifiers, yield, nutrient concentrations, filling value R_d, and geometric mean G.
`sample_col`	Character string specifying the sample identifier column name.
`yield_col`	Character string specifying the crop yield column name.
`nutrient_cols`	Character vector specifying column names for selected nutrients.

Details

Row-centered log ratios transform constrained compositional tissue proportions into scale-invariant, unconstrained variables:

$$V_X = \ln\left(\frac{X}{G}\right)$$

for each nutrient X and the residual filling value R_d. A fundamental property of CND row-centered log ratios is that their sum across all $d+1$ components equals zero:

$$\sum_{i=1}^{d+1} V_{X_i} = 0$$

Value

A data frame containing:

`sample_col`	Sample identifiers.
`yield_col`	Crop yield values.
`Vx`	Row-centered log ratio columns for each nutrient (e.g., `VN`, `VP`) and R_d (e.g., `VR2`).

Examples

```
{
data(plant_leaf_data)
simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

lc_res <- log_centered_table(
  simplex_data = simplex_res,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
}
```

---

plant_leaf_data *Example Crop Dataset for CND*

Description

A dataset containing leaf sample nutrient information for demonstrating CND methodology. All the nutrients used for analysis should in percentage (%).

Usage

```
plant_leaf_data
```

Format

An object of class `data.frame` with 100 rows and 7 columns.

Details

@format A data frame with 100 rows and 7 columns:

Sample Unique identifier for each leaf sample

Yield Yield corresponding to each sample

N Nitrogen concentration (%)

P Phosphorus concentration (%)

Ca Calcium concentration (%)

Mg Magnesium concentration (%)

S Sulfur concentration (%)

@details This dataset is provided as an example for applying CND functions in the package. It includes macronutrients (N, P, Ca, Mg, S), Sample and yield columns.

@examples

Load the dataset

data(plant_leaf_data)

#Sorts dataset by yield, computes the residual filling value simplex_res <- simplex_table(data = plant_leaf_data, sample_col = "Sample", yield_col = "Yield", nutrient_cols = c("N", "P","Ca","Mg","S")) print(simplex_res)

#Computes scale-invariant row-centered log ratios lc_res <- log_centered_table(simplex_data = simplex_res, sample_col = "Sample", yield_col = "Yield", nutrient_cols = c("N", "P","Ca","Mg","S")) print(lc_res)

#Iteratively partitions the database iter_res <- iteration_for_hyp(lc_data = lc_res, yield_col = "Yield", nutrient_cols = c("N", "P","Ca","Mg","S"), digits = 2)

print(head(iter_res))

#Fits cubic models , calculates inflection point yield cutoffs and extracts high-yield norms. cnd_out <- cut_off_table(iter_data = iter_res, lc_data = lc_res, yield_col = "Yield", nutrient_cols = c("N", "P","Ca","Mg","S"), digits = 2)

print(cnd_out$cut_off_table) print(cnd_out$norms_table)

#Plots upper-tail cumulative chi-square distribution curve and marks theoretical critical threshold. plot_cnd_chisq(nutrient_cols = c("N", "P","Ca","Mg","S"), low_yield_pct = 78, max_r2 = 16)

#Generates a connected scatter plot cumulative function vs. Yield for an #individual nutrient or all nutrients simultaneously. p_cum <- plot_cumulative_variance(iter_res, lc_res, nutrient = "all") print(p_cum)

---

plot_cate_nelson *Cate Nelson Quadrant Scatter Plot for CND Imbalance Index vs. Yield*

Description

Generates a publication-ready `ggplot2` scatter diagram partitioning crop yield against the global CND nutrient imbalance index (r^2) using Cate Nelson ANOVA quadrant lines (Khiari et al., 2001). Automatically calculates critical split points if not provided.

Usage

```
plot_cate_nelson(
  diag_data,
  crit_r2 = NULL,
  crit_yield = NULL,
  yield_col = "Yield"
)
```

Arguments

`diag_data`	Data frame output from `diagnose_specimen`.
`crit_r2`	Numeric critical CND r^2 threshold. If `NULL`, auto-calculated via ANOVA.
`crit_yield`	Numeric critical yield cutoff. If `NULL`, auto-calculated via ANOVA.
`yield_col`	Character string specifying the crop yield column name. Default is `"Yield"`.

Value

A ggplot object representing the Cate Nelson quadrant scatter diagram.

Examples

```
data(plant_leaf_data)

simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

lc_res <- log_centered_table(
  simplex_data = simplex_res,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
iter_res <- iteration_for_hyp(
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)
cnd_out <- cut_off_table(
  iter_data = iter_res,
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)

data(validation_data)
diag_results <- diagnose_specimen(
  new_data = validation_data,
  norms_table = cnd_out$norms_table,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

p_cn <- plot_cate_nelson(diag_results, crit_r2 = 1.4, crit_yield = 6.40)
```

---

plot_cnd_chisq — *Plot Cumulative Chi-Square Distribution Curve and Mark Theoretical Critical CND r^2*

Description

Plots the theoretical upper-tail cumulative chi-square distribution curve for $d+1$ degrees of freedom and marks the theoretical critical CND nutrient imbalance threshold (r^2) corresponding to the low-yield subpopulation percentage (Parent et al., 1994; Khiari et al., 2001).

Usage

```
plot_cnd_chisq(nutrient_cols, low_yield_pct, max_r2 = 16, digits = 2)
```

Arguments

`nutrient_cols`	Character vector of selected nutrient names.
`low_yield_pct`	Numeric percentage of low-yielding population below cutoff (e.g., `67.5`).
`max_r2`	Numeric maximum value for the x-axis plot range. Default is `16`.
`digits`	Numeric integer for decimal rounding precision. Default is `2`.

Details

Because CND indices (I_X) are independent unit-normal variables, their sum of squares follows a chi-square distribution with $d+1$ degrees of freedom:

$$r^2 = \sum_{i=1}^{d+1} I_{X_i}^2 \sim \chi^2(d+1)$$

Mapping the proportion of low-yielding observations as an upper-tail cumulative probability yields the theoretical critical threshold $r^2 = \text{qchisq}(p, df = d+1, \text{lower.tail} = \text{FALSE})$.

Value

Generates a plot and invisibly returns a list with `df`, `low_yield_pct`, and `critical_r2`.

Examples

```
plot_cnd_chisq(nutrient_cols = c("N", "P","Ca","Mg","S"), low_yield_pct = 78, max_r2 = 16)
```

`plot_cnd_yield_linear` *Linear Regression Plot for Crop Yield vs. CND Imbalance Index (r^2)*

Description

Fits a simple linear regression model relating crop yield to the global CND nutrient imbalance index (r^2) based on output from `diagnose_specimen` (Khiari et al., 2001).

Usage

```
plot_cnd_yield_linear(
  x = NULL,
  y = NULL,
  data = NULL,
  r2_col = "r2",
  yield_col = "Yield",
  critical_r2 = 1.4,
  x_label = "CND r^2",
  y_label = "Yield",
  plot_title = "Linear Yield Trend with CND Imbalance Index"
)
```

Arguments

x	Optional numeric vector of CND r^2 imbalance index values.
y	Optional numeric vector of crop yield values.
data	Optional data frame containing CND r^2 and yield columns (e.g. output from diagnose_specimen).
r2_col	Character string specifying the column name for r^2 when data is supplied. Default is "r2".
yield_col	Character string specifying the column name for yield when data is supplied. Default is "Yield".
critical_r2	Optional numeric threshold value at which to extrapolate and annotate yield. Default is 1.4.
x_label	Character string for the x-axis label. Default is "CND r^2".
y_label	Character string for the y-axis label. Default is "Yield".
plot_title	Character string for the plot title. Default is "Linear Yield Trend with CND Imbalance Index".

Value

A list containing:

model	The fitted lm linear regression model object.
predicted_yield	Numeric extrapolated yield value at critical_r2 (or NULL if critical_r2 is omitted).
plot	The generated ggplot object.

Examples

```
data(plant_leaf_data)
simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
```

```
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

lc_res <- log_centered_table(
  simplex_data = simplex_res,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
iter_res <- iteration_for_hyp(
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)
cnd_out <- cut_off_table(
  iter_data = iter_res,
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)

data(validation_data)
diag_results <- diagnose_specimen(
  new_data = validation_data,
  norms_table = cnd_out$norms_table,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

linear_result <- plot_cnd_yield_linear(
  data = diag_results,
  r2_col = "r2",
  yield_col = "Yield",
  critical_r2 = 1.4,
  y_label = "Yield",
  plot_title = "Linear Yield Trend with CND Imbalance Index"
)
linear_result$predicted_yield
summary(linear_result$model)
print(linear_result$plot)
```

`plot_cnd_yield_nlr` *Non-Linear Polynomial Regression Plot for Crop Yield vs. CND Imbalance Index*

Description

Fits a non-linear polynomial regression model relating crop yield to the global CND nutrient imbalance index (r^2) based on output from diagnose_specimen or direct vector inputs. Predicts and visualizes the extrapolated yield at a specified critical r^2 threshold with arrow and point annotations.

Usage

```
plot_cnd_yield_nlr(
  x = NULL,
  y = NULL,
  data = NULL,
  r2_col = "r2",
  yield_col = "yield",
  critical_r2 = 9.65,
  x_label = "CND r^2",
  y_label = "Yield",
  plot_title = "Non-Linear Regression: Yield vs. Imbalance Index",
  poly_degree = 2
)
```

Arguments

x	Optional numeric vector of CND r^2 imbalance index values.
y	Optional numeric vector of crop yield values.
data	Optional data frame containing CND r^2 and yield columns (e.g. output from diagnose_specimen).
r2_col	Character string specifying the column name for r^2 when data is supplied. Default is "r2".
yield_col	Character string specifying the column name for yield when data is supplied. Default is "yield".
critical_r2	Numeric threshold value at which to extrapolate and annotate yield. Default is 9.65.
x_label	Character string for the x-axis label. Default is "CND r^2".
y_label	Character string for the y-axis label. Default is "Yield".
plot_title	Character string for the plot title. Default is "Non-Linear Regression: Yield vs. Imbalance Index".
poly_degree	Integer specifying the polynomial degree for regression fit. Default is 2.

Value

A list containing:

model	The fitted lm polynomial regression model object.
predicted_yield	Numeric extrapolated yield value at critical_r2.
plot	The generated ggplot object.

Examples

```
data(plant_leaf_data)
simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)

lc_res <- log_centered_table(
  simplex_data = simplex_res,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
iter_res <- iteration_for_hyp(
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)
cnd_out <- cut_off_table(
  iter_data = iter_res,
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"),
  digits = 2
)

data(validation_data)
diag_results <- diagnose_specimen(
  new_data = validation_data,
  norms_table = cnd_out$norms_table,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
# Example using diagnose_specimen output data frame
nlr_result<- plot_cnd_yield_nlr(
  data = diag_results,
  r2_col = "r2",
  yield_col = "Yield",
  critical_r2 = 3.23,
  poly_degree = 2
)

nlr_result$predicted_yield
summary(nlr_result$model)
print(nlr_result$plot)

# Example using direct vector inputs
nlr_out2 <- plot_cnd_yield_nlr(
```

```
  x = diag_results$r2,
  y = diag_results$Yield,
  critical_r2 = 3.23
)
```

---

plot_cumulative_variance

Connected Scatter Plot for Cumulative Variance Ratio Function Fi(Vx) vs. Yield

---

Description

Generates a connected scatter plot (line + points) using `ggplot2` displaying the cumulative variance ratio function $F_i^C(V_X)$ on the y-axis versus crop yield on the x-axis for selected nutrients or all nutrients simultaneously (Khiari et al., 2001).

Usage

```
plot_cumulative_variance(
  iter_data,
  lc_data,
  nutrient = "all",
  yield_col = "Yield"
)
```

Arguments

`iter_data`	Data frame output from `iteration_for_hyp`.
`lc_data`	Data frame output from `log_centered_table`.
`nutrient`	Character string specifying the nutrient to plot (e.g., `"N"`, `"P"`, `"Residual"`, or `"all"`). Default is `"all"`.
`yield_col`	Character string specifying the crop yield column name. Default is `"Yield"`.

Value

A `ggplot` object representing the connected scatter plot.

Examples

```
data(plant_leaf_data)

simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
```

```
lc_res <- log_centered_table(
  simplex_data = simplex_res,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S")
)
iter_res <- iteration_for_hyp(
  lc_data = lc_res,
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"))

p_cum <- plot_cumulative_variance(iter_res, lc_res, nutrient = "all")
```

`simplex_table` *Generate Simplex Table for Compositional Nutrient Diagnosis (CND)*

Description

Rearranges a plant tissue database by crop yield in descending order, calculates the scale-invariant filling value (R_d) for the d-dimensional simplex (S^d), and computes the geometric mean (G) across all $d + 1$ components (Parent & Dafir, 1992; Khiari et al., 2001).

Usage

```
simplex_table(data, sample_col, yield_col, nutrient_cols)
```

Arguments

`data`	A data frame containing sample identifiers, crop yield values, and nutrient concentration percentages.
`sample_col`	Character string specifying the column name containing sample identifiers.
`yield_col`	Character string specifying the column name containing crop yield values.
`nutrient_cols`	Character vector specifying column names for the selected nutrients (e.g., `c("N", "P", "K")`).

Details

In Compositional Nutrient Diagnosis (CND), plant tissue composition forms a d-dimensional simplex (S^d) composed of $d + 1$ nutrient proportions including d nutrients and a residual filling value (R_d):

$$R_d = 100 - \sum_{i=1}^{d} \text{Nutrient}_i$$

The geometric mean (G) of all $d + 1$ components is computed as:

$$G = \left(\left(\prod_{i=1}^{d} \text{Nutrient}_i \right) \times R_d \right)^{\frac{1}{d+1}}$$

Value

A data frame arranged in descending order of crop yield containing:

`sample_col`	Original sample identifiers.
`yield_col`	Crop yield values.
`nutrient_cols`	Selected nutrient concentration percentages.
`Rd`	Residual filling value (R_d, where suffix d matches the number of selected nutrients).
`G`	Geometric mean across all $d + 1$ components for each specimen.

Examples

```
{
data(plant_leaf_data)
simplex_res <- simplex_table(
  data = plant_leaf_data,
  sample_col = "Sample",
  yield_col = "Yield",
  nutrient_cols = c("N", "P","Ca","Mg","S"))
}
```

`validation_data` *Example Crop Dataset for CND norms validation*

Description

A dataset containing leaf sample nutrient information for demonstrating CND norms validation methodology. All the nutrients used for analysis should in percentage (%).

Usage

```
validation_data
```

Format

An object of class `data.frame` with 20 rows and 7 columns.

Details

@format A data frame with 20 rows and 7 columns:

Sample Unique identifier for each leaf sample

Yield Yield corresponding to each sample

N Nitrogen concentration (%)

P Phosphorus concentration (%)

Ca Calcium concentration (%)

Mg Magnesium concentration (%)

S Sulfur concentration (%)

@details This dataset is provided as an example dataset for standardizing its row-centered log ratio against the high-yield norms to obtain the CND index and to compute global nutrient imbalance index. The dataset includes following columns:Sample,Yield and macronutrients (N, P, Ca, Mg, S).

@examples

Load the dataset

data(validation_data)

#Standardizes new crop samples against norms to calculate individual CND nutrient indices and global imbalance index. diag_results <- diagnose_specimen(new_data = validation_data, norms_table = cnd_out$norms_table, sample_col = "Sample", yield_col = "Yield", nutrient_cols = c("N", "P","Ca","Mg","S")) print(diag_results)

#Evaluates candidate split pairs via ANOVA grid search to maximize between sum of squares for automated Cate–Nelson thresholding. auto_thresh <- calc_cate_nelson_empirical(diag_results, yield_col = "Yield") auto_thresh

#Generates a quadrant scatter plot (Yield vs. r^2) with automated or specified Cate–Nelson threshold lines. plot_cate_nelson(diag_results)

#If there are specified values then use function in following way p_cn <- plot_cate_nelson(diag_results, crit_r2 = 1.4, crit_yield = 6.40) print(p_cn)

#Fits a non-linear polynomial regression (`ggplot2`) relating crop yield to CND indices. regplot<- plot_cnd_yield_nlr(data = diag_results, r2_col = "r2", # Name of your r^2 column yield_col = "Yield", # Name of your yield column critical_r2 = 3.23, # Custom critical value y_label = "Yield", # Y-axis label plot_title = "Yield vs CND Imbalance Index") regplot

#Fits a linear regression model (`ggplot2`) relating crop yield to CND indices. plot <- plot_cnd_yield_linear(data = diag_results, r2_col = "r2", # Column name for r^2 yield_col = "Yield", # Column name for yield critical_r2 = 3.23, # Critical value y_label = "Yield vs CND Imbalane Index")

Index

∗ **datasets**
    plant_leaf_data, 10
    validation_data, 20

calc_cate_nelson_empirical, 2
cut_off_table, 3, *5*

diagnose_specimen, *2*, 5, *11*, *13, 14*, *16*

ggplot, *12*, *14*, *16*, *18*

iteration_for_hyp, *4*, 7, *18*

log_centered_table, *4*, *7*, 9, *18*

plant_leaf_data, 10
plot_cate_nelson, 11
plot_cnd_chisq, 12
plot_cnd_yield_linear, 13
plot_cnd_yield_nlr, 15
plot_cumulative_variance, 18

simplex_table, *9*, 19

validation_data, 20