

Package: drisdiagnostics (via r-universe)

September 26, 2026

Type Package

Title Diagnostic Systems for Plant Nutrient Analysis (DRIS, MDRIS, PASS)

Version 1.0.1

Description Provides implementations of the Diagnosis and Recommendation Integrated System (DRIS), the Modified DRIS (MDRIS), and the Plant Analysis with Standardized Scores (PASS) approaches for nutrient diagnosis in crops. These methods allow quantitative evaluation of nutrient imbalances using ratio-based indices and standardized scores, supporting improved fertilizer use efficiency and crop management decisions. The DRIS method is described in Walworth, J.L. and Sumner, M.E. (1987) <[doi:10.1007/978-1-4612-4682-4_4](https://doi.org/10.1007/978-1-4612-4682-4_4)>. The MDRIS approach is detailed in Beverly, R.B. (1987) <[doi:10.1080/01904168709363672](https://doi.org/10.1080/01904168709363672)>. The PASS method combining DRIS and sufficiency ranges is presented in Baldock, J.O. and Schulte, E.E. (1996) <[doi:10.2134/agronj1996.00021962008800030015x](https://doi.org/10.2134/agronj1996.00021962008800030015x)>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Imports ggplot2, stats, rlang

Suggests readxl, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author Blesson B. Varghese [aut, cre], Mubashir Sadiq V [aut], Deepthi C [aut], Sowmiya Saravanan [aut], Aparna Mohan V [aut]

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

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

Date/Publication 2026-09-26 17:00:16 UTC
RemoteUrl <https://github.com/cran/driscdiagnostics>
RemoteRef HEAD
RemoteSha 5e4550d38ded05e4933374a3b433c675c6008131

Contents

calculate_dris_by_group	2
calculate_dris_diagnostics	4
calculate_dris_norms	6
calculate_mdriis_diagnostics	7
calculate_mdriis_norms	8
calculate_pass_analysis	9
dni_norms	11
generate_hyp_ratios_table	12
ini_norms	13
my_corn_samples	14
my_crop_data	15
plot_dris_diverging_bar	17
plot_dris_norms_wheel	18
split_populations	20
sufficiency_ranges	21
Index	24

calculate_dris_by_group
<i>Perform Grouped DRIS Diagnostics and Multi-Level Factor Filtering</i>

Description

Subsets a plant tissue dataset by a specified grouping factor (such as leaf age, soil type, crop variety, or geographic region) and runs the standard Beaufils DRIS diagnostic workflow independently for each factor level. It compiles individual diagnostic tables for each sub-group and generates a master summary table containing mean nutrient indices, the overall Nutrient Balance Index (NBI) computed directly from the group means, and a sorted nutrient requirement order to facilitate comparative agronomic evaluations across varying environmental or management conditions.

Usage

```
calculate_dris_by_group(  
  data,  
  dris_norms,  
  nutrient_cols,  
  filter_col,  
  sample_col = "Sample",
```

```

    hyp_only = TRUE
  )

```

Arguments

data	A data frame containing crop nutrient concentration columns, the grouping factor column, and the "population_category" column (if hyp_only = TRUE) as generated by split_populations().
dris_norms	A data frame of calibrated reference norms returned by calculate_dris_norms().
nutrient_cols	A character vector specifying the exact names of the nutrient columns in data to be evaluated.
filter_col	A character string representing the name of the column in data to filter and group by (e.g., "Age", "pH", or "Variety").
sample_col	A character string specifying the column name containing the sample identifiers in data. Defaults to "Sample".
hyp_only	A logical value. If TRUE (default), restricts the sub-group diagnostic evaluations strictly to high-yielding reference samples ("HYP"). If FALSE, runs the diagnostics over all samples in each group.

Value

A list containing two diagnostic components:

grouped_tables	A named list of individual compiled diagnostic tables (data frames) for each unique level of the grouping factor.
mean_summary_table	A master summary data frame containing the mean index values for each nutrient, the overall group NBI (sum of absolute values of the mean indices), and the sorted Requirement_Order for each factor level.

Examples

```

set.seed(123)
my_crop_data <- data.frame(
  Sample = paste0("S", 1:20),
  Age = rep(c("Young", "Medium", "Old"), length.out = 20),
  Yield = runif(20, 2, 5),
  N = runif(20, 2, 4),
  P = runif(20, 0.2, 0.5),
  K = runif(20, 1.5, 3),
  S = runif(20, 0.1, 0.4),
  Ca = runif(20, 0.5, 2),
  Mg = runif(20, 0.2, 0.8)
)
# Split populations
processed_data <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "percentile",

```

```

    value = 50
  )

  # Calculate DRIS norms
  dris_norms <- calculate_dris_norms(
    data = processed_data,
    nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg")
  )

  # Grouped DRIS diagnostics
  grouped_results <- calculate_dris_by_group(
    data = processed_data,
    dris_norms = dris_norms,
    nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg"),
    filter_col = "Age",
    sample_col = "Sample",
    hyp_only = TRUE
  )

```

calculate_dris_diagnostics

Calculate DRIS Functions, Individual Indices, and NBI

Description

Performs standard Diagnosis and Recommendation Integrated System (DRIS) calculations for plant tissue samples. It computes individual function values for each optimal ratio using the symmetrical Beaufils formulas, derives individual relative nutrient indices by averaging these function values with appropriate sign conventions and computes the overall Nutrient Balance Index (NBI) as the sum of absolute index values to measure general nutritional balance.

Usage

```

calculate_dris_diagnostics(
  data,
  dris_norms,
  nutrient_cols,
  sample_col = "Sample",
  hyp_only = TRUE
)

```

Arguments

<code>data</code>	A data frame containing the original plant sample dataset to be diagnosed.
<code>dris_norms</code>	A data frame of calibrated norms returned by <code>calculate_dris_norms()</code> .
<code>nutrient_cols</code>	A character vector specifying the names of the selected nutrients to evaluate.

sample_col	A character string specifying the column name containing the sample identifiers. Defaults to "Sample".
hyp_only	A logical value. If TRUE (default), restricts the diagnostic calculations strictly to samples labeled "HYP". If FALSE, runs the diagnosis over the entire dataset (both HYP and LYP).

Value

A list containing four diagnostic components:

nutrient_tables	A list of individual data frames (one for each nutrient) detailing sample ratios, calculated function values, and final indices with a summary Mean row.
formulas	A list of character strings representing the final algebraic formulas used to compile the index for each nutrient.
compiled_table	A master summary data frame containing the final relative index values and overall Nutrient Balance Index (NBI) for each sample, with an overall Mean row appended at the bottom.
raw_indices	A numeric matrix containing unrounded, raw index values for subsequent statistical modeling or graphing.

Examples

```
set.seed(123)

my_crop_data <- data.frame(
  Sample = paste0("S", 1:20),
  Yield = c(
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4,
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4
  ),
  N = runif(20, 2, 4),
  P = runif(20, 0.2, 0.5),
  K = runif(20, 1.5, 3),
  S = runif(20, 0.1, 0.4),
  Ca = runif(20, 0.5, 2),
  Mg = runif(20, 0.2, 0.8)
)

processed_data <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "percentile",
  value = 50
)

dris_norms <- calculate_dris_norms(
  data = processed_data,
  nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg")
)
```

```

)

diagnostics_results <- calculate_dris_diagnostics(
  data = processed_data,
  dris_norms = dris_norms,
  nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg"),
  sample_col = "Sample"
)
print(diagnostics_results$compiled_table)

```

calculate_dris_norms *Calculate DRIS Ratio Norms and Select Optimal Expressions*

Description

Computes descriptive statistics for both high-yielding (HYP) and low-yielding (LYP) crop sub-populations across all possible bi-variate nutrient expressions, including ratio forms (A/B, B/A) and product forms (A*B). To determine which expression is the most agronomically sensitive, the function compares the variances of the two subpopulations using a variance ratio test (Sa/Sb) and evaluates statistical significance via a standard F-test. The nutrient expression that maximizes this variance ratio is flagged as optimal, establishing the baseline diagnostic norm.

Usage

```
calculate_dris_norms(data, nutrient_cols)
```

Arguments

data A data frame containing nutrient columns and a "population_category" column.

nutrient_cols A character vector containing the names of the nutrient columns.

Value

A data frame containing calculated population statistics (means, variances, standard deviations, and coefficients of variation) for both HYP and LYP groups across all combinations, alongside variance ratios (Sa_Sb), F-test p-values, and a selection indicator ("Optimal") flagging the expression with the largest variance ratio.

Examples

```

set.seed(123)

my_crop_data <- data.frame(
  Sample = paste0("S", 1:20),
  Yield = c(
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4,
    2.0, 2.1, 2.2, 2.3, 2.4,

```

```

      4.0, 4.1, 4.2, 4.3, 4.4
    ),
    N = runif(20, 2, 4),
    P = runif(20, 0.2, 0.5),
    K = runif(20, 1.5, 3),
    S = runif(20, 0.1, 0.4),
    Ca = runif(20, 0.5, 2),
    Mg = runif(20, 0.2, 0.8)
  )

processed_data <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "percentile",
  value = 50
)

dris_norms <- calculate_dris_norms(
  data = processed_data,
  nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg")
)

print(dris_norms)

```

```
calculate_mdriis_diagnostics
```

Calculate Beverly's Modified DRIS (MDRIS) Diagnostics & Concentration Indices

Description

This function takes a tissue sample dataset and pre-calculated MDRIS norms, and computes:

1. Ratio-Based MDRIS indices (I_R) using the unified symmetrical logarithmic equation.
2. Log-Concentration Indices (J_log_r) evaluating nutrients on an absolute log scale.
3. Direct Concentration Indices (K_r) evaluating nutrients directly on raw scales.

Usage

```

calculate_mdriis_diagnostics(
  data,
  mdriis_norms,
  nutrient_cols,
  sample_col = "Sample"
)

```

Arguments

<code>data</code>	A data frame of plant tissue analysis to diagnose.
<code>mdri_norms</code>	The list returned by <code>calculate_mdri_norms()</code> .
<code>nutrient_cols</code>	A character vector containing the names of the selected nutrients.
<code>sample_col</code>	Character string representing the column name of the sample identifier.

Value

A list containing:

1. `weighted_deviations`: A data frame of calculated WDR/S values for each sample.
2. `compiled_table`: A master data frame containing MDRI indices, `J_log_r` values, `K_r` values and corresponding diagnostic rankings.

Examples

```
set.seed(123)
my_crop_data <- data.frame(
  Sample = paste0("S", 1:20),
  N = runif(20, 2, 4),
  P = runif(20, 0.2, 0.5),
  K = runif(20, 1.5, 3),
  Zn = runif(20, 0.001, 0.01),
  B = runif(20, 0.0005, 0.002)
)

mdri_norms <- calculate_mdri_norms(
  data = my_crop_data,
  nutrient_cols = c("N", "P", "K", "Zn", "B")
)

mdri_results <- calculate_mdri_diagnostics(
  data = my_crop_data,
  mdri_norms = mdri_norms,
  nutrient_cols = c("N", "P", "K", "Zn", "B"),
  sample_col = "Sample"
)
print(mdri_results$compiled_table)
```

`calculate_mdri_norms` *Calculate Beverly's Modified DRIS (MDRI) Norms*

Description

This function computes both log-concentration norms and log-ratio norms for the entire population as proposed by Beverly (1987). Unlike traditional DRIS, MDRI norms are computed for the entire population database without splitting into high/low-yielding groups, as the log-transformation stabilizes variances across different yield levels.

Usage

```
calculate_mdri_norms(data, nutrient_cols)
```

Arguments

data A data frame containing raw nutrient concentrations (dry matter percentage).
nutrient_cols A character vector containing the names of the nutrient columns.

Value

A list containing:

1. **nutrient_norms**: Data frame of individual nutrient statistics (Raw Mean, Raw SD, Log Mean, Log SD).
2. **ratio_norms**: Data frame of log-transformed ratio statistics (Expression, Log_Mean, Log_SD).

Examples

```
set.seed(123)
my_crop_data <- data.frame(
  Sample = paste0("S", 1:20),
  N = runif(20, 2, 4),
  P = runif(20, 0.2, 0.5),
  K = runif(20, 1.5, 3),
  Zn = runif(20, 0.001, 0.01),
  B = runif(20, 0.0005, 0.002)
)

mdri_norms <- calculate_mdri_norms(
  data = my_crop_data,
  nutrient_cols = c("N", "P", "K", "Zn", "B")
)
print(mdri_norms)
```

calculate_pass_analysis

Perform Plant Analysis with Standardized Scores (PASS) with Custom Norms

Description

This function implements the hybrid PASS diagnostic engine of Baldock and Schulte (1996) using data frame inputs for INI and DNI norms. It is highly robust and avoids R index-stripping issues.

Usage

```
calculate_pass_analysis(data, ini_norms, dni_norms, sample_col = "Sample")
```

Arguments

<code>data</code>	A data frame containing tissue concentration columns for the crop.
<code>ini_norms</code>	A data frame containing columns: nutrient, CL, SD, and group.
<code>dni_norms</code>	A data frame containing columns: ratio, mean, and SD.
<code>sample_col</code>	Character string representing the column name of the sample identifier.

Value

A list containing:

1. `compiled_table`: A master data frame containing calculated indices, PASS YI, and recommendations.
2. `raw_ini`: Numeric matrix of calculated raw INI values.
3. `raw_dni`: Numeric matrix of calculated raw DNI values.

Examples

```
set.seed(123)

# Tissue samples to be diagnosed
my_corn_samples <- data.frame(
  Sample = paste0("Corn_", 1:10),
  N = runif(10, 2.5, 3.5),
  P = runif(10, 0.20, 0.40),
  K = runif(10, 1.80, 2.60),
  S = runif(10, 0.15, 0.30),
  Ca = runif(10, 0.40, 0.90),
  Mg = runif(10, 0.20, 0.50)
)

# Independent Nutrient Index (INI) norms: critical level (CL), SD, and
# whether the nutrient participates in the "common" DNI calculations
ini_norms <- data.frame(
  nutrient = c("N", "P", "K", "S", "Ca", "Mg"),
  CL = c(3.00, 0.30, 2.20, 0.22, 0.65, 0.35),
  SD = c(0.25, 0.04, 0.20, 0.03, 0.10, 0.05),
  group = c("common", "common", "common",
            "rare", "rare", "rare"),
  stringsAsFactors = FALSE
)

# Dependent Nutrient Index (DNI) norms: mean and SD for each nutrient ratio
dni_norms <- data.frame(
  ratio = c("N/P", "P/N", "N/K", "K/N", "P/K", "K/P"),
  mean = c(10.00, 0.10, 1.36, 0.74, 0.136, 7.33),
  SD = c(1.20, 0.02, 0.15, 0.08, 0.020, 0.90),
  stringsAsFactors = FALSE
)

# Run the PASS analysis
```

```

pass_report <- calculate_pass_analysis(
  data = my_corn_samples,
  ini_norms = ini_norms,
  dni_norms = dni_norms,
  sample_col = "Sample"
)
print(pass_report$compiled_table)

```

dni_norms

DRIS Norms for PASS Analysis

Description

A dataset containing DRIS nutrient ratio norms used in the PASS method.

Usage

```

dni_norms

```

Format

A data frame with 11 rows and 3 columns:

ratio Nutrient ratio (e.g., N/P, K/Ca)

mean Mean value of the ratio

SD Standard deviation of the ratio

Details

These DRIS norms are taken from Baldock and Schulte (1996) and are used together with initial norms to generate PASS diagnostics.

Examples

```

data(dni_norms)
head(dni_norms)

```

```
generate_hyp_ratios_table
```

Calculate Optimal Nutrient Ratios for the High-Yielding Population (HYP)

Description

Extracts the optimal diagnostic expressions identified in the norms calibration step and calculates their exact ratio values for each sample in the high-yielding reference subpopulation (HYP). This function provides a structured reference table summarizing baseline healthy ratios, appending critical summary statistics (arithmetic means, variances, standard deviations, and coefficients of variation) as summary rows at the bottom of the table.

Usage

```
generate_hyp_ratios_table(data, dris_norms, sample_col = "Sample")
```

Arguments

<code>data</code>	A data frame containing the crop tissue and population category columns.
<code>dris_norms</code>	A data frame returned by <code>calculate_dris_norms()</code> .
<code>sample_col</code>	Character string representing the column name of the sample identifier.

Value

A character data frame where rows represent individual high-yielding samples and columns represent the calculated optimal ratio expressions, with appended summary rows labeled "Mean", "Variance", "SD", and "CV" at the bottom.

Examples

```
set.seed(123)

my_crop_data <- data.frame(
  Sample = paste0("S", 1:20),
  Yield = c(
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4,
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4
  ),
  N = runif(20, 2, 4),
  P = runif(20, 0.2, 0.5),
  K = runif(20, 1.5, 3),
  S = runif(20, 0.1, 0.4),
  Ca = runif(20, 0.5, 2),
  Mg = runif(20, 0.2, 0.8)
)
```

```

processed_data <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "percentile",
  value = 50
)

dris_norms <- calculate_dris_norms(
  data = processed_data,
  nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg")
)

hyp_ratios_results <- generate_hyp_ratios_table(
  data = processed_data,
  dris_norms = dris_norms,
  sample_col = "Sample"
)
print(hyp_ratios_results)

```

ini_norms

Initial Norms for PASS Analysis

Description

A dataset containing initial nutrient norms used in the PASS (Plant Analysis with Standardized Scores) method.

Usage

```
ini_norms
```

Format

A data frame with 12 rows and 4 columns:

nutrient Nutrient name

CL Critical level value

SD Standard deviation

group Grouping category

Details

These initial norms are taken from Baldock and Schulte (1996) and are used in combination with DRIS norms to compute PASS scores.

Examples

```
data(ini_norms)
head(ini_norms)
```

my_corn_samples	Leaf Samples for PASS Analysis
-----------------	--------------------------------

Description

A dataset containing nutrient concentrations from leaf samples used to demonstratenPASS calculation.

Usage

```
my_corn_samples
```

Format

A data frame with 3 rows and 12 columns:

- Sample** Unique identifier for each leaf sample
- N** Nitrogen concentration (%)
- P** Phosphorus concentration (%)
- K** Potassium concentration (%)
- Ca** Calcium concentration (%)
- Mg** Magnesium concentration (%)
- S** Sulfur concentration (%)
- Zn** Zinc concentration (mg/kg)
- B** Boron concentration (mg/kg)
- Cu** Copper concentration (mg/kg)
- Mn** Manganese concentration (mg/kg)
- Fe** Iron concentration (mg/kg)

Details

These leaf samples are used as example data for running PASS analyses.

Examples

```

data(my_corn_samples)
data(ini_norms)
data(dni_norms)

# Run the PASS analysis
pass_report <- calculate_pass_analysis(
  data = my_corn_samples,
  ini_norms = ini_norms,
  dni_norms = dni_norms,
  sample_col = "Sample"
)

# View the master diagnostic compiled table
print(pass_report$compiled_table)

```

my_crop_data

*Example Crop Dataset for DRIS and MDRIS***Description**

A dataset containing leaf sample information for demonstrating the Diagnosis and Recommendation Integrated System (DRIS) and Modified DRIS (MDRIS) methods.

Usage

```
my_crop_data
```

Format

A data frame with 250 rows and 15 columns:

Sample Unique identifier for each leaf sample

N Nitrogen concentration (%)

P Phosphorus concentration (%)

K Potassium concentration (%)

Ca Calcium concentration (%)

Mg Magnesium concentration (%)

S Sulfur concentration (%)

Fe Iron concentration (mg/kg)

Mn Manganese concentration (mg/kg)

Cu Copper concentration (mg/kg)

Zn Zinc concentration (mg/kg)

B Boron concentration (mg/kg)

Yield Yield corresponding to each sample

Age Age of the plant from which the leaf was taken

pH Soil pH of the sample location

Details

This dataset is provided as an example for applying DRIS and MDRIS functions in the package. It includes macronutrients (N, P, K, Ca, Mg, S), micronutrients (Fe, Mn, Cu, Zn, B), yield, plant age, and soil pH.

Examples

```
# Load the dataset
data(my_crop_data)

# Split data into HYP and LYP using different methods. Use any one of the method.
processed_data_per <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "percentile",
  value = 75
)

processed_data_abs <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "absolute",
  value = 3.2
)

processed_data <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "mean_sd",
  value = 2/3
)

# Generate DRIS norms
dris_norms <- calculate_dris_norms(
  data = processed_data,
  nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg")
)
print(dris_norms)

# Generate HYP ratios table
hyp_ratios_results <- generate_hyp_ratios_table(
  data = processed_data,
  dris_norms = dris_norms,
  sample_col = "Sample"
)
print(hyp_ratios_results, row.names = FALSE)

# Run DRIS diagnostics
diagnostics_results <- calculate_dris_diagnostics(
  data = processed_data,
  dris_norms = dris_norms,
  nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg"),
```

```

    sample_col = "Sample"
  )
  diagnostics_results

# Grouped DRIS results
grouped_results <- calculate_dris_by_group(
  data = processed_data,
  dris_norms = dris_norms,
  nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg"),
  filter_col = "Age",
  sample_col = "Sample",
  hyp_only = FALSE
)
grouped_results

# Nutrient sufficiency ranges
ranges_table <- sufficiency_ranges(
  data = processed_data,
  nutrient_cols = c("N", "P", "K", "Zn", "B")
)

# Tri-axial DRIS wheel plot
plot_dris_norms_wheel(
  dris_norms = dris_norms,
  ratios_to_plot = c("P/N", "N/K", "P/K")
)

# Calibrate MDRIS norms
mdris_norms <- calculate_mdri_norms(
  data = processed_data,
  nutrient_cols = c("N", "P", "K", "Zn", "B")
)

# Run MDRIS diagnostics
mdris_results <- calculate_mdri_diagnostics(
  data = processed_data,
  mdris_norms = mdris_norms,
  nutrient_cols = c("N", "P", "K", "Zn", "B"),
  sample_col = "Sample"
)
print(mdris_results$compiled_table, row.names = FALSE)

```

plot_dris_diverging_bar

Plot DRIS Indices as a Diverging Bar Chart

Description

Visualizes DRIS index values using a horizontal diverging bar chart. Negative index values indicate relative nutrient deficiency, whereas positive index values indicate relative nutrient excess.

Usage

```
plot_dris_diverging_bar(indices, sample_name = "Sample")
```

Arguments

indices	A named numeric vector containing DRIS index values. The names of the vector should correspond to nutrient names. Names ending in "_index" are automatically simplified for plotting.
sample_name	A character string specifying the sample name to display in the plot title. Defaults to "Sample".

Value

A ggplot object representing a horizontal diverging bar chart of DRIS indices, with nutrients classified as deficient (negative values) or excessive (positive values).

Examples

```
indices <- c(
  N_index = -15.4,
  P_index = -8.2,
  K_index = 12.6,
  Ca_index = 5.3
)

plot_dris_diverging_bar(
  indices = indices,
  sample_name = "Sample 1"
)
```

plot_dris_norms_wheel *Plot Tri-Axial DRIS Wheel Chart from Calibration Norms*

Description

Projects the calculated Z-score deviations of the low-yielding population (LYP) relative to the high-yielding population (HYP) norms on a classic concentric tri-axial balance wheel diagram. The function extracts descriptive statistics for exactly three selected ratio expressions, projects their Cartesian coordinate points at 120-degree radial intervals. The visual boundaries (representing the balanced zone at 2/3 SD radius and the imbalanced zone at 4/3 SD radius) allow agronomists to instantly diagnose nutrient balance and predict qualitative deficiency or excess dynamics.

Usage

```
plot_dris_norms_wheel(dris_norms, ratios_to_plot)
```

Arguments

dris_norms A data frame of calibrated reference norms and statistics returned by `calculate_dris_norms()`.

ratios_to_plot A character vector of exactly three ratio expressions present in `dris_norms$Expression` (e.g., `c("N/K", "Ca/K", "Ca/N")`).

Value

A ggplot object representing a fully annotated, concentric tri-axial DRIS balance wheel with directional nutrient deficiency and excess indicators.

Examples

```
set.seed(123)
my_crop_data <- data.frame(
  Sample = paste0("S", 1:20),
  Yield = c(
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4,
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4
  ),
  N = runif(20, 2, 4),
  P = runif(20, 0.2, 0.5),
  K = runif(20, 1.5, 3),
  S = runif(20, 0.1, 0.4),
  Ca = runif(20, 0.5, 2),
  Mg = runif(20, 0.2, 0.8)
)

processed_data <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "percentile",
  value = 50
)

dris_norms <- calculate_dris_norms(
  data = processed_data,
  nutrient_cols = c("N", "P", "K", "S", "Ca", "Mg")
)

plot_dris_norms_wheel(
  dris_norms = dris_norms,
  ratios_to_plot = c("P/N", "N/K", "P/K")
)
```

split_populations	<i>Split Plant Populations into High-Yielding (HYP) and Low-Yielding (LYP) Subpopulations</i>
-------------------	---

Description

Categorizes crop sample tissue records into high-yielding (HYP) and low-yielding (LYP) subpopulations based on a designated economic or yield threshold. This partitioning is a fundamental preprocessing step in DRIS allowing the calculation of baseline nutritional norms and standard deviation diagnostics exclusively from the healthy, high-yielding reference cohort.

Usage

```
split_populations(
  data,
  yield_col,
  method = c("mean_sd", "percentile", "absolute"),
  value = NULL
)
```

Arguments

data	A data frame containing plant tissue nutrient concentrations and crop yield values or economic parameter.
yield_col	Character string representing the column name of the yield variable in data.
method	Character string specifying the mathematical cutoff method. Must be one of "mean_sd" (mean plus standard deviation threshold), "percentile" (quantile-based cutoff), or "absolute" (fixed yield threshold).
value	Numeric threshold value matching the selected method. For "mean_sd", it represents a custom standard deviation multiplier added to the mean (defaults to 1.0 if NULL). For "percentile", it is the percentile ranking cutoff (e.g., 75 for the 75th percentile). For "absolute", it is the raw absolute yield value.

Value

The original data frame data with an appended factor column named "population_category" containing levels "HYP" (High-Yielding Population) and "LYP" (Low-Yielding Population).

Examples

```
set.seed(123)
my_crop_data <- data.frame(
  Sample = paste0("S", 1:20),
  Yield = c(
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4,
    2.0, 2.1, 2.2, 2.3, 2.4,
```

```

      4.0, 4.1, 4.2, 4.3, 4.4
    ),
    N = runif(20, 2, 4),
    P = runif(20, 0.2, 0.5),
    K = runif(20, 1.5, 3),
    S = runif(20, 0.1, 0.4),
    Ca = runif(20, 0.5, 2),
    Mg = runif(20, 0.2, 0.8)
  )

# Split populations using a custom percentile value
processed_data_per <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "percentile",
  value = 75
)

# Split populations using an absolute yield threshold of 3.2
processed_data_abs <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "absolute",
  value = 3.2
)

# Split populations using a custom standard deviation multiplier
processed_data <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "mean_sd",
  value = 2/3
)

```

sufficiency_ranges	<i>Create Sufficiency Ranges Table for Nutrients based on HYP</i>
--------------------	---

Description

This function calculates the sufficiency ranges for each nutrient in the High-Yielding Population (HYP) based on standard deviation thresholds (4/3 SD and 8/3 SD).

Usage

```
sufficiency_ranges(data, nutrient_cols)
```

Arguments

- data** A data frame containing the crop tissue and population category columns (after running `split_populations()`).
- nutrient_cols** A character vector containing the names of the nutrient columns.

Details

Classification rules:

- Deficient: $< \text{mean} - (8/3) * \text{SD}$
- Low: $\text{mean} - (8/3) * \text{SD}$ to $\text{mean} - (4/3) * \text{SD}$
- Optimum: $\text{mean} - (4/3) * \text{SD}$ to $\text{mean} + (4/3) * \text{SD}$
- High: $\text{mean} + (4/3) * \text{SD}$ to $\text{mean} + (8/3) * \text{SD}$
- Excessive: $> \text{mean} + (8/3) * \text{SD}$

Value

A data frame containing calculated sufficiency ranges for each nutrient with columns: Nutrient, Mean, SD, Deficient, low, optimum, high and excessive values.

Examples

```
set.seed(123)
my_crop_data <- data.frame(
  Sample = paste0("S", 1:20),
  Yield = c(
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4,
    2.0, 2.1, 2.2, 2.3, 2.4,
    4.0, 4.1, 4.2, 4.3, 4.4
  ),
  N = runif(20, 2, 4),
  P = runif(20, 0.2, 0.5),
  K = runif(20, 1.5, 3),
  Zn = runif(20, 0.001, 0.01),
  B = runif(20, 0.0005, 0.002)
)

processed_data <- split_populations(
  data = my_crop_data,
  yield_col = "Yield",
  method = "percentile",
  value = 50
)

# Generate the nutrient sufficiency ranges table
ranges_table <- sufficiency_ranges(
  data = processed_data,
  nutrient_cols = c("N", "P", "K", "Zn", "B")
)
```

sufficiency_ranges

23

```
print(ranges_table)
```

Index

* datasets

- dni_norms, [11](#)
- ini_norms, [13](#)
- my_corn_samples, [14](#)
- my_crop_data, [15](#)

- calculate_dris_by_group, [2](#)
- calculate_dris_diagnostics, [4](#)
- calculate_dris_norms, [6](#)
- calculate_mdriis_diagnostics, [7](#)
- calculate_mdriis_norms, [8](#)
- calculate_pass_analysis, [9](#)

- dni_norms, [11](#)

- generate_hyp_ratios_table, [12](#)

- ini_norms, [13](#)

- my_corn_samples, [14](#)
- my_crop_data, [15](#)

- plot_dris_diverging_bar, [17](#)
- plot_dris_norms_wheel, [18](#)

- split_populations, [20](#)
- sufficiency_ranges, [21](#)