Package ‘aridagri’

May 7, 2026

Type Package

Title Comprehensive Statistical Tools for Agricultural Research

Version 2.0.3

Description A comprehensive suite of statistical and analytical tools for agricultural research. Includes complete analysis of variance (ANOVA) functions for all experimental designs: Completely Randomized Design (CRD), Randomized Block Design (RBD), Pooled RBD, Split Plot with all variations, Split-Split Plot, Strip Plot, Latin Square, Factorial, Augmented, and Alpha Lattice, with proper error terms and comprehensive Standard Error (SE) and Critical Difference (CD) calculations. Features multiple post-hoc tests: Least Significant Difference (LSD), Duncan Multiple Range Test (DMRT), Tukey Honestly Significant Difference (HSD), Student-Newman-Keuls (SNK), Scheffe, Bonferroni, and Dunnett, along with assumption checking and publication-ready output. Advanced methods include stability analysis using Eberhart-Russell regression, Additive Main Effects and Multiplicative Interaction (AMMI), Finlay-Wilkinson regression, Shukla stability variance, Wricke ecovalence, Coefficient of Variation (CV), and Cultivar Superiority Index as described in Eberhart and Russell (1966) <doi:10.2135/cropsci1966.0011183X000600010011x>. Thermal indices include Growing Degree Days (GDD), Heliothermal Units (HTU), Photothermal Units (PTU), and Heat Use Efficiency (HUE). Crop growth analysis covers Crop Growth Rate (CGR), Relative Growth Rate (RGR), Net Assimilation Rate (NAR), and Leaf Area Index (LAI). Also provides harvest index, yield gap analysis, economic efficiency indices (Benefit-Cost ratio), nutrient use efficiency calculations, correlation matrix, Principal Component Analysis (PCA), path analysis, and Structural Equation Modeling (SEM). Statistical methods follow Gomez and Gomez (1984, ISBN:0471870927) and Panse and Sukhatme (1985, ISBN:8170271169).

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Suggests ggplot2, dplyr, tidyr, emmeans, multcomp, corrplot, factoextra, FactoMineR, lavaan, semPlot, agricolae, car, writexl, rmarkdown, testthat (>= 3.0.0)

URL https://github.com/lalitrolaniya/aridagri

BugReports https://github.com/lalitrolaniya/aridagri/issues

NeedsCompilation no

Author Lalit Kumar Rolaniya [aut, cre] (ORCID:
<https://orcid.org/0000-0001-8908-1211>),
Ram Lal Jat [aut] (ORCID: <https://orcid.org/0009-0003-4339-0555>),
Monika Punia [aut] (ORCID: <https://orcid.org/0009-0002-0294-6767>),
Raja Ram Choudhary [aut]

Maintainer Lalit Kumar Rolaniya <lalitrolaniya@gmail.com>

Repository CRAN

Date/Publication 2026-02-24 19:20:08 UTC

Contents

`aridagri-package` *aridagri: Comprehensive Statistical Tools for Agricultural Research*

Description

The aridagri package provides comprehensive statistical and analytical tools for agricultural research. It includes complete ANOVA functions for all experimental designs, multiple post-hoc tests, stability analysis methods, thermal indices, crop growth analysis, and advanced statistical methods.

Experimental Design ANOVA Functions

- `anova_crd`: Completely Randomized Design
- `anova_rbd`: Randomized Block Design
- `anova_rbd_pooled`: Pooled RBD (Multi-Environment)
- `anova_latin`: Latin Square Design
- `anova_factorial`: Two-Factor Factorial
- `anova_factorial_3way`: Three-Factor Factorial
- `anova_spd`: Split Plot Design
- `anova_sspd`: Split-Split Plot Design
- `anova_strip`: Strip Plot Design
- `anova_augmented`: Augmented Block Design
- `anova_alpha_lattice`: Alpha Lattice Design

Post-Hoc Tests

- `perform_posthoc`: Multiple comparison tests (LSD, Duncan, Tukey, SNK, Scheffe, Bonferroni, Dunnett)
- `check_assumptions`: ANOVA assumption checking

Agronomic Analysis Functions

- `stability_analysis`: Multi-method stability analysis (Eberhart-Russell, AMMI, Finlay-Wilkinson, Shukla, Wricke, CV, Superiority)
- `thermal_indices`: GDD, HTU, PTU, Heat Use Efficiency
- `crop_growth_analysis`: CGR, RGR, NAR, LAI
- `harvest_index`: Harvest index and partitioning
- `yield_gap_analysis`: Yield gap calculations
- `economic_indices`: B:C ratio, net returns

Statistical Functions

- `correlation_analysis`: Correlation matrix with significance
- `pca_analysis`: Principal component analysis
- `path_analysis`: Path coefficient analysis
- `sem_analysis`: Structural equation modeling

Nutrient Analysis Functions

- `nue_calculate`: Nutrient use efficiency calculations
- `nutrient_response`: Response curve analysis
- `economic_analysis`: Economic viability assessment

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

See Also

Useful links:

- `https://github.com/lalitrolaniya/aridagri`
- Report bugs at `https://github.com/lalitrolaniya/aridagri/issues`

`anova_alpha_lattice` *Alpha Lattice Design ANOVA*

Description

Performs ANOVA for Alpha Lattice (Resolvable Incomplete Block) Design.

Usage

```
anova_alpha_lattice(
  data,
  response,
  genotype,
  replication,
  block,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the data
`response`	Name of response variable
`genotype`	Name of genotype column
`replication`	Name of replication column
`block`	Name of incomplete block column (nested within replication)
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA results

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
data <- expand.grid(rep = 1:2, block = 1:5, entry = 1:4)
data$genotype <- paste0("G", 1:nrow(data))
data$yield <- rnorm(nrow(data), 1200, 150)
```

`anova_augmented` *Augmented Block Design ANOVA*

Description

Performs ANOVA for Augmented Randomized Block Design where checks are replicated and test entries appear once.

Usage

```
anova_augmented(data, response, genotype, block, check_names, verbose = TRUE)
```

Arguments

`data`	Data frame containing the data
`response`	Name of response variable
`genotype`	Name of genotype/entry column
`block`	Name of block column
`check_names`	Vector of check variety names
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA results and adjusted means

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
data <- data.frame(
  block = rep(1:5, each = 7),
  genotype = c(rep(c("C1","C2","C3"), 5), paste0("T", 1:20)),
  yield = rnorm(35, 1200, 150)
)
# Note: This is simplified example
```

---

anova_crd

==
COMPREHENSIVE EXPERIMENTAL DESIGN ANALYSIS FUNCTIONS Package: aridagri Author: Lalit Kumar Rolaniya ICAR-Indian Institute of Pulses Research, Regional Centre, Bikaner
==
Completely Randomized Design (CRD) ANOVA

Description

Performs complete ANOVA for Completely Randomized Design with post-hoc tests, assumptions checking, and publication-ready output.

Usage

```
anova_crd(
  data,
  response,
  treatment,
  posthoc = "lsd",
```

```
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the experimental data
`response`	Name of the response variable (as string)
`treatment`	Name of treatment factor
`posthoc`	Post-hoc test: "lsd", "duncan", "tukey", "snk", "scheffe", or "all"
`alpha`	Significance level (default 0.05)
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA table, means, post-hoc results, and diagnostics

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- data.frame(
  treatment = rep(c("T1", "T2", "T3", "T4"), each = 5),
  yield = c(rnorm(5, 1200, 50), rnorm(5, 1350, 60),
            rnorm(5, 1100, 55), rnorm(5, 1450, 65))
)
anova_crd(data, response = "yield", treatment = "treatment", posthoc = "all")
```

`anova_factorial` *Factorial ANOVA (Two-Factor)*

Description

Performs two-factor factorial ANOVA with interaction analysis.

Usage

```
anova_factorial(
  data,
  response,
  factor1,
  factor2,
  replication = NULL,
  posthoc = "lsd",
```

```
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the data
`response`	Name of the response variable
`factor1`	Name of first factor (A)
`factor2`	Name of second factor (B)
`replication`	Name of replication factor (optional)
`posthoc`	Post-hoc test method
`alpha`	Significance level
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA results

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- expand.grid(
  rep = 1:4,
  nitrogen = c("N0", "N40", "N80", "N120"),
  phosphorus = c("P0", "P30", "P60")
)
data$yield <- rnorm(nrow(data), 1200, 150)

anova_factorial(data, response = "yield",
                factor1 = "nitrogen", factor2 = "phosphorus",
                replication = "rep")
```

`anova_factorial_3way` *Three-Factor Factorial ANOVA*

Description

Performs three-factor factorial ANOVA with all interactions.

Usage

```
anova_factorial_3way(
  data,
  response,
  factor1,
  factor2,
  factor3,
  replication = NULL,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the data
`response`	Name of the response variable
`factor1`	Name of first factor (A)
`factor2`	Name of second factor (B)
`factor3`	Name of third factor (C)
`replication`	Name of replication factor (optional)
`alpha`	Significance level
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA results

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
data <- expand.grid(rep = 1:3, A = c("A1", "A2"), B = c("B1", "B2"), C = c("C1", "C2"))
data$yield <- rnorm(nrow(data), 1200, 150)
anova_factorial_3way(data, "yield", "A", "B", "C", "rep")
```

`anova_latin` *Latin Square Design ANOVA*

Description

Performs ANOVA for Latin Square Design with row and column blocking.

Usage

```
anova_latin(
  data,
  response,
  treatment,
  row,
  column,
  posthoc = "lsd",
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the data
`response`	Name of the response variable
`treatment`	Name of treatment factor
`row`	Name of row factor
`column`	Name of column factor
`posthoc`	Post-hoc test method
`alpha`	Significance level
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA results

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
# 5x5 Latin Square
data <- data.frame(
  row = rep(1:5, each = 5),
  col = rep(1:5, 5),
  treatment = c("A","B","C","D","E", "B","C","D","E","A",
```

```
                 "C","D","E","A","B", "D","E","A","B","C",
                 "E","A","B","C","D"),
  yield = rnorm(25, 1200, 100)
)
anova_latin(data, response = "yield", treatment = "treatment",
            row = "row", column = "col")
```

---

anova_rbd *Randomized Block Design (RBD) ANOVA*

Description

Performs complete ANOVA for Randomized Block Design (RCBD) with post-hoc tests, assumptions checking, and publication-ready output.

Usage

```
anova_rbd(
  data,
  response,
  treatment,
  block,
  posthoc = "lsd",
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the experimental data
`response`	Name of the response variable
`treatment`	Name of treatment factor
`block`	Name of block/replication factor
`posthoc`	Post-hoc test: "lsd", "duncan", "tukey", "snk", "scheffe", or "all"
`alpha`	Significance level (default 0.05)
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA table, means, post-hoc results

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- data.frame(
  rep = rep(1:4, each = 5),
  treatment = rep(c("T1", "T2", "T3", "T4", "T5"), 4),
  yield = c(rnorm(5, 1200, 50), rnorm(5, 1250, 55),
            rnorm(5, 1180, 45), rnorm(5, 1270, 60))
)
anova_rbd(data, response = "yield", treatment = "treatment",
          block = "rep", posthoc = "duncan")
```

`anova_rbd_pooled` *Pooled Analysis of RBD Experiments (Multi-Environment/Year)*

Description

Performs pooled ANOVA for RBD experiments conducted across multiple environments, years, or locations. Tests homogeneity of error variances using Bartlett's test before pooling.

Usage

```
anova_rbd_pooled(
  data,
  response,
  treatment,
  environment,
  block,
  posthoc = "lsd",
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing combined data from all environments
`response`	Name of the response variable
`treatment`	Name of treatment factor
`environment`	Name of environment/year/location factor
`block`	Name of block factor (nested within environment)
`posthoc`	Post-hoc test method
`alpha`	Significance level
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing pooled ANOVA, individual ANOVAs, and interaction analysis

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
# Data from 3 years
data <- data.frame(
  year = rep(c("Y1", "Y2", "Y3"), each = 20),
  rep = rep(rep(1:4, each = 5), 3),
  treatment = rep(c("T1", "T2", "T3", "T4", "T5"), 12),
  yield = rnorm(60, 1200, 150)
)
anova_rbd_pooled(data, response = "yield", treatment = "treatment",
                 environment = "year", block = "rep")
```

`anova_spd` *== SPLIT PLOT DESIGN ANALYSIS FUNCTIONS (All Variations) Package: aridagri Author: Lalit Kumar Rolaniya ICAR-Indian Institute of Pulses Research, Regional Centre, Bikaner == Split Plot Design ANOVA (Standard)*

Description

Performs complete ANOVA for Split Plot Design with proper error terms for main plot and sub-plot factors. Includes all standard post-hoc comparisons.

Usage

```
anova_spd(
  data,
  response,
  main_plot,
  sub_plot,
  replication,
  posthoc = "lsd",
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the experimental data
`response`	Name of the response variable
`main_plot`	Name of main plot factor

`sub_plot`	Name of sub-plot factor
`replication`	Name of replication/block factor
`posthoc`	Post-hoc test method
`alpha`	Significance level
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA table, means, and post-hoc results

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- data.frame(
  rep = rep(1:3, each = 12),
  irrigation = rep(rep(c("I1", "I2", "I3"), each = 4), 3),
  variety = rep(c("V1", "V2", "V3", "V4"), 9),
  yield = rnorm(36, 1200, 150)
)
anova_spd(data, response = "yield", main_plot = "irrigation",
          sub_plot = "variety", replication = "rep")
```

`anova_spd_ab_cd` *Split Plot Design with (AB) Main and (CD) Sub*

Description

Performs ANOVA for Split Plot Design where main plot contains factorial combination of AB and sub-plot contains factorial combination of CD. Complex design for multi-factor experiments.

Usage

```
anova_spd_ab_cd(
  data,
  response,
  main_factor1,
  main_factor2,
  sub_factor1,
  sub_factor2,
  replication,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the experimental data
`response`	Name of the response variable
`main_factor1`	First factor in main plot (A)
`main_factor2`	Second factor in main plot (B)
`sub_factor1`	First factor in sub-plot (C)
`sub_factor2`	Second factor in sub-plot (D)
`replication`	Name of replication factor
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA table and means

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
# Example: Irrigation  Tillage (main), Variety  Nitrogen (sub)
data <- expand.grid(
  rep = 1:3,
  irrigation = c("I1", "I2"),
  tillage = c("CT", "ZT"),
  variety = c("V1", "V2"),
  nitrogen = c("N1", "N2", "N3")
)
data$yield <- rnorm(nrow(data), 1200, 150)

anova_spd_ab_cd(data, response = "yield",
                main_factor1 = "irrigation", main_factor2 = "tillage",
                sub_factor1 = "variety", sub_factor2 = "nitrogen",
                replication = "rep")
```

`anova_spd_ab_main` *Split Plot Design with AB in Main Plot*

Description

Performs ANOVA for Split Plot Design where main plot contains factorial combination of two factors (AB) and sub-plot contains factor C. Common in irrigation variety as main plot and nitrogen as sub-plot.

Usage

```
anova_spd_ab_main(
  data,
  response,
  main_factor1,
  main_factor2,
  sub_plot,
  replication,
  posthoc = "lsd",
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the experimental data
`response`	Name of the response variable
`main_factor1`	First factor in main plot (A)
`main_factor2`	Second factor in main plot (B)
`sub_plot`	Sub-plot factor (C)
`replication`	Name of replication factor
`posthoc`	Post-hoc test method
`alpha`	Significance level
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Details

Design structure:

- Main Plot: A B factorial
- Sub-Plot: C
- Error (a): For testing A, B, and AB
- Error (b): For testing C and all interactions with C

Value

List containing ANOVA table and means

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- data.frame(
  rep = rep(1:3, each = 24),
  irrigation = rep(rep(c("I1", "I2"), each = 12), 3),
  variety = rep(rep(c("V1", "V2", "V3"), each = 4), 6),
  nitrogen = rep(c("N0", "N1", "N2", "N3"), 18),
  yield = rnorm(72, 1200, 150)
)
anova_spd_ab_main(data, response = "yield",
                  main_factor1 = "irrigation", main_factor2 = "variety",
                  sub_plot = "nitrogen", replication = "rep")
```

`anova_spd_c_main_ab_sub` *Split Plot Design with C in Main Plot, AB in Sub-Plot*

Description

Performs ANOVA for Split Plot Design where main plot contains single factor C and sub-plot contains factorial combination of AB.

Usage

```
anova_spd_c_main_ab_sub(
  data,
  response,
  main_plot,
  sub_factor1,
  sub_factor2,
  replication,
  posthoc = "lsd",
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the experimental data
`response`	Name of the response variable
`main_plot`	Main plot factor (C)
`sub_factor1`	First factor in sub-plot (A)
`sub_factor2`	Second factor in sub-plot (B)
`replication`	Name of replication factor
`posthoc`	Post-hoc test method
`alpha`	Significance level
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA table and means

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- data.frame(
  rep = rep(1:3, each = 24),
  irrigation = rep(rep(c("I1", "I2", "I3"), each = 8), 3),
  variety = rep(rep(c("V1", "V2"), each = 4), 9),
  nitrogen = rep(c("N1", "N2", "N3", "N4"), 18),
  yield = rnorm(72, 1200, 150)
)
anova_spd_c_main_ab_sub(data, response = "yield",
                         main_plot = "irrigation",
                         sub_factor1 = "variety", sub_factor2 = "nitrogen",
                         replication = "rep")
```

`anova_spd_pooled` *Pooled Split Plot Design Analysis*

Description

Performs pooled analysis of Split Plot Design experiments conducted across multiple environments/years/locations.

Usage

```
anova_spd_pooled(
  data,
  response,
  main_plot,
  sub_plot,
  environment,
  replication,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing combined data
`response`	Name of the response variable
`main_plot`	Name of main plot factor

`sub_plot`	Name of sub-plot factor
`environment`	Name of environment factor
`replication`	Name of replication factor (nested within environment)
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing pooled ANOVA and component analyses

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- expand.grid(
  year = c("Y1", "Y2", "Y3"),
  rep = 1:3,
  irrigation = c("I1", "I2", "I3"),
  variety = c("V1", "V2", "V3", "V4")
)
data$yield <- rnorm(nrow(data), 1200, 180)

anova_spd_pooled(data, response = "yield", main_plot = "irrigation",
                 sub_plot = "variety", environment = "year", replication = "rep")
```

`anova_sspd` *Split-Split Plot Design ANOVA*

Description

Performs complete ANOVA for Split-Split Plot Design with proper error terms for main plot, sub-plot, and sub-sub-plot factors. Generates publication-ready ANOVA table with significance levels.

Usage

```
anova_sspd(
  data,
  response,
  main_plot,
  sub_plot,
  sub_sub_plot,
  replication,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the experimental data
`response`	Name of the response variable (as string)
`main_plot`	Name of main plot factor
`sub_plot`	Name of sub-plot factor
`sub_sub_plot`	Name of sub-sub-plot factor
`replication`	Name of replication/block factor
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA table, means, and significance tests

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
# Example with sample data
data <- expand.grid(rep=1:3, A=c('A1','A2'), B=c('B1','B2'), C=c('C1','C2'))
data$yield <- rnorm(24, 1200, 150)
anova_sspd(data, response='yield', main_plot='A', sub_plot='B',
          sub_sub_plot='C', replication='rep')
```

`anova_sspd_pooled` *Pooled Split-Split Plot Design ANOVA*

Description

Performs pooled ANOVA for SSPD experiments across multiple environments.

Usage

```
anova_sspd_pooled(
  data,
  response,
  main_plot,
  sub_plot,
  sub_sub_plot,
  environment,
  replication,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing combined data
`response`	Name of the response variable
`main_plot`	Name of main plot factor
`sub_plot`	Name of sub-plot factor
`sub_sub_plot`	Name of sub-sub-plot factor
`environment`	Name of environment factor
`replication`	Name of replication factor
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing pooled ANOVA results

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
data <- expand.grid(env = c("E1","E2"), rep = 1:3, A = c("A1","A2"),
                    B = c("B1","B2"), C = c("C1","C2"))
data$yield <- rnorm(nrow(data), 1200, 150)
anova_sspd_pooled(data, "yield", "A", "B", "C", "env", "rep")
```

`anova_strip` ===

ADDITIONAL EXPERIMENTAL DESIGN FUNCTIONS Package: aridagri Authors: Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

===

Strip Plot Design ANOVA

Description

Performs ANOVA for Strip Plot (Strip-Split) Design where two factors are applied in horizontal and vertical strips.

Usage

```
anova_strip(
  data,
  response,
  horizontal_factor,
  vertical_factor,
  replication,
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`data`	Data frame containing the experimental data
`response`	Name of the response variable
`horizontal_factor`	Factor applied in horizontal strips (A)
`vertical_factor`	Factor applied in vertical strips (B)
`replication`	Name of replication factor
`alpha`	Significance level
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing ANOVA results

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
data <- expand.grid(
  rep = 1:4,
  irrigation = c("I1", "I2", "I3"),
  tillage = c("CT", "MT", "ZT")
)
data$yield <- rnorm(nrow(data), 1200, 150)
anova_strip(data, response = "yield", horizontal_factor = "irrigation",
            vertical_factor = "tillage", replication = "rep")
```

`arid_plot` *Visualization Functions for aridagri*

Description

Generate publication-quality plots for aridagri analyses.

Usage

```
arid_plot(x, type = "bar", ...)
```

Arguments

`x`	An object from aridagri analysis functions
`type`	Plot type: "bar", "line", "interaction", "boxplot"
`...`	Additional arguments passed to plotting functions

Value

A ggplot2 object

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
df <- data.frame(
  yield = c(1200, 1350, 1100, 1450, 1280),
  wue = c(4.2, 4.8, 3.9, 5.1, 4.5),
  protein = c(22.1, 23.5, 21.8, 24.2, 22.9)
)
result <- correlation_analysis(df, plot = FALSE)
arid_plot(result)
```

`check_assumptions` *Check ANOVA Assumptions*

Description

Tests assumptions of ANOVA: normality and homogeneity of variances.

Usage

```
check_assumptions(model, verbose = TRUE)
```

Arguments

`model`	ANOVA model object
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing diagnostic test results

`correlation_analysis` *Correlation Analysis with Significance*

Description

Computes correlation matrix with significance levels and generates publication-ready correlation table and plot.

Usage

```
correlation_analysis(
  data,
  method = "pearson",
  plot = TRUE,
  digits = 3,
  verbose = TRUE
)
```

Arguments

`data`	Data frame with numeric variables
`method`	Correlation method: "pearson", "spearman", or "kendall"
`plot`	Logical, whether to generate correlation plot
`digits`	Number of decimal places
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List with correlation matrix and significance matrix

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- data.frame(
  yield = c(1200, 1350, 1100, 1450, 1280),
  wue = c(4.2, 4.8, 3.9, 5.1, 4.5),
  protein = c(22.1, 23.5, 21.8, 24.2, 22.9)
)
correlation_analysis(data)
```

`crop_growth_analysis` *Crop Growth Analysis (CGR, RGR, NAR, LAI)*

Description

Calculates crop growth parameters from sequential harvest data.

Usage

```
crop_growth_analysis(dry_weight, leaf_area, days, verbose = TRUE)
```

Arguments

`dry_weight`	Vector of dry matter at different stages
`leaf_area`	Vector of leaf area at different stages
`days`	Vector of days after sowing
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

Data frame with growth parameters

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
dry_weight <- c(0.5, 2.1, 8.5, 25, 45, 62, 75)
leaf_area <- c(15, 85, 350, 800, 950, 850, 600)
days <- c(15, 30, 45, 60, 75, 90, 105)
crop_growth_analysis(dry_weight, leaf_area, days)
```

`economic_analysis` *Economic Analysis for Arid Agriculture*

Description

Calculates economic indicators including Cost of Cultivation, Gross Returns, Net Returns, B:C Ratio, and profitability indices for arid farming systems.

Usage

```
economic_analysis(
  yield,
  price,
  cost_fixed,
  cost_variable,
  byproduct_yield = 0,
  byproduct_price = 0,
  verbose = TRUE
)
```

Arguments

`yield`	Crop yield (kg/ha)
`price`	Market price (Rs/kg)
`cost_fixed`	Fixed costs (Rs/ha)
`cost_variable`	Variable costs (Rs/ha)
`byproduct_yield`	Byproduct yield (kg/ha), optional
`byproduct_price`	Byproduct price (Rs/kg), optional
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

Data frame with economic analysis

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
economic_analysis(yield = 1200, price = 65, cost_fixed = 15000, cost_variable = 12000)

# With byproduct
economic_analysis(yield = 1200, price = 65, cost_fixed = 15000,
                  cost_variable = 12000, byproduct_yield = 1800, byproduct_price = 5)
```

`economic_indices` *Economic Efficiency Indices (B:C Ratio)*

Description

Calculates B:C ratio and economic efficiency metrics.

Usage

```
economic_indices(gross_return, total_cost, verbose = TRUE)
```

Arguments

`gross_return`	Gross returns (Rs/ha)
`total_cost`	Total cost (Rs/ha)
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

Data frame with economic indices

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
economic_indices(gross_return = c(75000, 82000), total_cost = c(35000, 38000))
```

`export_results` *Export Results to Publication Format*

Description

Exports aridagri analysis results to Excel format for publication.

Usage

```
export_results(x, file, format = "xlsx", digits = 3, verbose = TRUE)
```

Arguments

`x`	An object from aridagri analysis functions
`file`	Output file path
`format`	Output format: "xlsx"
`digits`	Number of decimal places
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

Invisibly returns the file path

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
df <- data.frame(
  yield = c(1200, 1350, 1100, 1450, 1280),
  wue = c(4.2, 4.8, 3.9, 5.1, 4.5)
)
result <- correlation_analysis(df, plot = FALSE)
export_results(result, tempfile(fileext = ".xlsx"))
```

`harvest_index` *Harvest Index Calculation*

Description

Calculates Harvest Index and related partitioning indices.

Usage

```
harvest_index(grain_yield, straw_yield, verbose = TRUE)
```

Arguments

`grain_yield`	Grain/economic yield
`straw_yield`	Straw/stover yield
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

Data frame with harvest indices

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
harvest_index(grain_yield = c(1200, 1350, 1100), straw_yield = c(2400, 2500, 2300))
```

`imports` *Package imports*

Description

Package imports

`nue_calculate` *Nutrient Use Efficiency Calculations*

Description

Comprehensive nutrient use efficiency calculations including Agronomic Efficiency (AE), Physiological Efficiency (PE), Apparent Recovery Efficiency (ARE), and Partial Factor Productivity (PFP). Essential for INM research in arid regions.

Usage

```
nue_calculate(
  yield_fertilized,
  yield_control,
  nutrient_applied,
  nutrient_uptake_fert = NULL,
  nutrient_uptake_ctrl = NULL,
  biomass_fert = NULL,
  biomass_ctrl = NULL,
  verbose = TRUE
)
```

Arguments

`yield_fertilized`	Yield with fertilizer application (kg/ha)
`yield_control`	Yield in control/unfertilized plot (kg/ha)
`nutrient_applied`	Amount of nutrient applied (kg/ha)

`nutrient_uptake_fert`	Nutrient uptake in fertilized plot (kg/ha), optional
`nutrient_uptake_ctrl`	Nutrient uptake in control plot (kg/ha), optional
`biomass_fert`	Total biomass in fertilized plot (kg/ha), optional
`biomass_ctrl`	Total biomass in control plot (kg/ha), optional
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Details

Efficiency calculations:

- AE (Agronomic Efficiency) = (Yield_fert - Yield_ctrl) / Nutrient_applied
- PFP (Partial Factor Productivity) = Yield_fert / Nutrient_applied
- ARE (Apparent Recovery Efficiency) = (Uptake_fert - Uptake_ctrl) / Nutrient_applied 100
- PE (Physiological Efficiency) = (Yield_fert - Yield_ctrl) / (Uptake_fert - Uptake_ctrl)

Value

Data frame with efficiency indices

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
# Basic NUE calculation
nue_calculate(yield_fertilized = 1850, yield_control = 1200, nutrient_applied = 40)

# Complete NUE with uptake data
nue_calculate(
  yield_fertilized = 1850,
  yield_control = 1200,
  nutrient_applied = 40,
  nutrient_uptake_fert = 65,
  nutrient_uptake_ctrl = 35
)
```

`nutrient_response` *Nutrient Response Curve Analysis*

Description

Fits nutrient response curves using quadratic, linear-plateau, or Mitscherlich models to determine economic optimum dose.

Usage

```
nutrient_response(
  dose,
  yield,
  model = "quadratic",
  price_output = 50,
  price_nutrient = 15,
  verbose = TRUE
)
```

Arguments

`dose`	Numeric vector of nutrient doses (kg/ha)
`yield`	Numeric vector of corresponding yields (kg/ha)
`model`	Model type: "quadratic", "linear_plateau", or "mitscherlich"
`price_output`	Price of output (Rs/kg)
`price_nutrient`	Price of nutrient (Rs/kg)
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List with model parameters and economic optimum

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
dose <- c(0, 20, 40, 60, 80, 100)
yield <- c(1100, 1350, 1520, 1610, 1650, 1660)
nutrient_response(dose, yield, model = "quadratic",
                  price_output = 60, price_nutrient = 15)
```

`path_analysis` *Path Coefficient Analysis*

Description

Performs path analysis to determine direct and indirect effects of independent variables on a dependent variable. Essential for understanding yield contributing factors.

Usage

```
path_analysis(data, dependent, independent, digits = 4, verbose = TRUE)
```

Arguments

`data`	Data frame with numeric variables
`dependent`	Name of dependent variable (e.g., "yield")
`independent`	Character vector of independent variable names
`digits`	Number of decimal places
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List with direct effects, indirect effects, and correlation breakdown

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- data.frame(
  yield = c(1200, 1350, 1100, 1450, 1280, 1380, 1220, 1400),
  pods = c(45, 52, 42, 58, 48, 54, 46, 56),
  seeds = c(8.2, 9.1, 7.8, 9.5, 8.5, 9.0, 8.3, 9.3),
  weight = c(32, 35, 30, 38, 33, 36, 31, 37)
)
path_analysis(data, dependent = "yield",
              independent = c("pods", "seeds", "weight"))
```

`pca_analysis` *Principal Component Analysis*

Description

Performs PCA with visualization suitable for agricultural research data. Includes scree plot, biplot, and variable contributions.

Usage

```
pca_analysis(data, scale = TRUE, ncp = 5, plot = TRUE, verbose = TRUE)
```

Arguments

`data`	Data frame with numeric variables
`scale`	Logical, whether to scale variables (default TRUE)
`ncp`	Number of components to retain (default 5)
`plot`	Logical, whether to generate plots
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

PCA results with eigenvalues, loadings, and scores

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
data <- data.frame(
  yield = rnorm(30, 1200, 200),
  wue = rnorm(30, 4.5, 0.5),
  protein = rnorm(30, 22, 2),
  biomass = rnorm(30, 3500, 500)
)
pca_analysis(data)
```

`perform_posthoc` *Perform Post-Hoc Tests*

Description

Performs multiple comparison tests after ANOVA.

Usage

```
perform_posthoc(
  model,
  data,
  response,
  treatment,
  mse,
  df_error,
  posthoc = "lsd",
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

`model`	ANOVA model object
`data`	Data frame
`response`	Response variable name
`treatment`	Treatment factor name
`mse`	Mean square error
`df_error`	Error degrees of freedom
`posthoc`	Test type: "lsd", "duncan", "tukey", "snk", "scheffe", or "all"
`alpha`	Significance level
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing post-hoc test results

`print.aridagri_pca` *Print Method for PCA Results*

Description

Prints a formatted summary of Principal Component Analysis (PCA) results.

Usage

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

Arguments

`x`	An object of class 'aridagri_pca' from `pca_analysis`
`...`	Additional arguments (currently unused)

Value

No return value, called for side effects. Prints the number of components retained by Kaiser criterion and cumulative variance explained to the console. The input object is returned invisibly.

`sem_analysis` *Structural Equation Modeling for Field Experiments*

Description

Performs SEM analysis for agricultural field experiments. Allows testing of hypothesized causal relationships among variables.

Usage

```
sem_analysis(data, model, plot = TRUE, verbose = TRUE)
```

Arguments

`data`	Data frame with variables
`model`	Model specification in lavaan syntax
`plot`	Logical, whether to generate path diagram
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

SEM results including fit indices and parameter estimates

Author(s)

Lalit Kumar Rolaniya, ICAR-IIPR, Bikaner

Examples

```
if (requireNamespace("lavaan", quietly = TRUE)) {
  set.seed(42)
  n <- 100
  nitrogen <- rnorm(n, 60, 10)
  phosphorus <- rnorm(n, 30, 5)
  yield <- 0.5 * nitrogen + 0.3 * phosphorus + rnorm(n, 0, 5)
  df <- data.frame(yield = yield, nitrogen = nitrogen, phosphorus = phosphorus)
  model <- 'yield ~ nitrogen + phosphorus'
  result <- sem_analysis(df, model, plot = FALSE)
}
```

`stability_analysis`

==

ADVANCED AGRONOMIC STATISTICAL METHODS Package: aridagri Authors: Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary ICAR-Indian Institute of Pulses Research, Regional Centre, Bikaner ICAR-Indian Institute of Groundnut Research, Regional Research Station, Bikaner

==

Stability Analysis for Agronomic Traits (Multiple Methods)

Description

Performs comprehensive stability analysis using multiple established methods for evaluating genotype/treatment performance across environments.

Methods included:

- Eberhart & Russell (1966): Regression approach
- AMMI: Additive Main effects and Multiplicative Interaction
- Finlay & Wilkinson (1963): Linear regression on environmental mean
- Shukla (1972): Stability variance
- Wricke (1962): Ecovalence
- Coefficient of Variation: CV-based ranking
- Superiority Index: Lin & Binns (1988)

Usage

```
stability_analysis(
  data,
  genotype,
  environment,
  replication,
  trait,
  method = "all",
  verbose = TRUE
)
```

Arguments

`data`	Data frame with genotype/treatment, environment, replication, and trait
`genotype`	Name of genotype/treatment column
`environment`	Name of environment/location/year column
`replication`	Name of replication column
`trait`	Name of trait/response variable
`method`	Method: "eberhart", "ammi", "finlay", "shukla", "wricke", "cv", "superiority", or "all"
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List containing stability parameters and rankings

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

References

Eberhart, S.A. and Russell, W.A. (1966). Crop Science, 6: 36-40.

Examples

```
data <- expand.grid(
  variety = paste0("V", 1:10),
  location = paste0("L", 1:5),
  rep = 1:3
)
data$yield <- rnorm(nrow(data), 1200, 200)

stability_analysis(data, genotype = "variety", environment = "location",
                   replication = "rep", trait = "yield", method = "all")
```

`thermal_indices` *Growing Degree Days (GDD) and Thermal Indices*

Description

Calculates Growing Degree Days, Helio-thermal Units, Photo-thermal Units, and Heat Use Efficiency from temperature data.

Usage

```
thermal_indices(
  tmax,
  tmin,
  base_temp = 10,
  sunshine_hours = NULL,
  day_length = NULL,
  crop_yield = NULL,
  biomass = NULL,
  verbose = TRUE
)
```

Arguments

`tmax`	Vector of daily maximum temperatures (C)
`tmin`	Vector of daily minimum temperatures (C)
`base_temp`	Base temperature (C)
`sunshine_hours`	Vector of daily sunshine hours (optional)
`day_length`	Vector of day length in hours (optional)
`crop_yield`	Crop yield for HUE calculation (optional)
`biomass`	Biomass for HUE calculation (optional)
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

List with thermal indices

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
tmax <- runif(90, 30, 42)
tmin <- runif(90, 18, 28)
thermal_indices(tmax, tmin, base_temp = 10)
```

`yield_gap_analysis` *Yield Gap Analysis*

Description

Calculates yield gaps comparing actual with potential yields.

Usage

```
yield_gap_analysis(
  actual_yield,
  potential_yield,
  attainable_yield = NULL,
  verbose = TRUE
)
```

Arguments

`actual_yield`	Actual yield (kg/ha)
`potential_yield`	Potential yield (kg/ha)
`attainable_yield`	Attainable yield (kg/ha, optional)
`verbose`	Logical. If TRUE (default), prints formatted output to console.

Value

Data frame with yield gap analysis

Author(s)

Lalit Kumar Rolaniya, Ram Lal Jat, Monika Punia, Raja Ram Choudhary

Examples

```
yield_gap_analysis(actual_yield = c(800, 950, 720), potential_yield = 1500)
```

Index