Package 'AgriDataTools'

August 8, 2026

Type Package

Title Automated Statistical Analysis and Tools for Agricultural Research

Version 0.1.2

Description A comprehensive suite of statistical tools tailored for agricultural and plant breeding research. Provides automated pipelines for analysis of variance and covariance under randomized complete block designs and completely randomized designs, descriptive summary statistics, and post-hoc multiple range tests including Least Significant Difference, Tukey, and Scheffe based on Steel et al. (1997) <isbn:978-0070610286>. Quantitative genetic parameters including genotypic, phenotypic, and environmental variance components and broad-sense heritability follow Burton and Devane (1953) <doi:10.2134/agronj1953.00021962004500100005x>. Genetic advance and genetic advance as percentage of mean estimation follow Johnson et al. (1955) <doi:10.2134/agronj1955.00021962004700070009x>. Genotypic, phenotypic, and environmental correlations follow Miller et al. (1958) <doi:10.2134/agronj1958.00021962005000100020x>. Genotypic and phenotypic path coefficient analysis direct and indirect effects decomposition follows Dewey and Lu (1959) <doi:10.2134/agronj1959.00021962005100090002x>. Principal component analysis follows Jolliffe (2002) <isbn:978-0387954424> and hierarchical clustering follows Sneath and Sokal (1973) <isbn:978-0716706977>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

Imports ggplot2, reshape2, factoextra, dendextend, circlize, stats, graphics, dplyr, utils

Config/roxygen2/version 8.0.0

VignetteBuilder knitr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

NeedsCompilation no

Author Faheem Khan [aut, cre] (ORCID: <https://orcid.org/0009-0002-8963-7157>)

Maintainer Faheem Khan <2022ag94@uaf.edu.pk>

Repository CRAN

Date/Publication 2026-08-08 11:20:06 UTC

Contents

`analyze_clustering` *Hierarchical Cluster Analysis and Phenotypic Diversity Engine*

Description

The `analyze_clustering` function executes an agglomerative hierarchical clustering routine over multi-trait breeding datasets. It automatically computes cluster assignments, genotype groupings, trait cluster means, intra-cluster average distances, and inter-cluster centroid distances.

Usage

```
analyze_clustering(
  data,
  traits,
  k = 4,
  linkage_method = "ward.D2",
  reporting_level = 1
)
```

Arguments

data	A data.frame containing phenotypic records with a Genotype column or purely numeric traits.
traits	A character vector specifying the quantitative traits to be integrated into the cluster matrix.
k	An integer specifying the target number of clusters to partition the tree. Defaults to 4.
linkage_method	A character string specifying the target agglomerative clustering algorithm (e.g., "ward.D2", "complete", "average"). Defaults to "ward.D2".
reporting_level	An integer flag defining console output verbosity: 0 for silent execution and 1 for detailed summary output to the console. Defaults to 1.

Value

Invisibly returns a named list of class "list" containing 9 detailed computational components:

dist_matrix	A spatial dist object representing calculated multidimensional Euclidean distances between genotypes based on standardized phenotypic scores.
hc_object	The raw hierarchical clustering output object of class hclust.
cophenetic_corr	A numeric value indicating the cophenetic correlation coefficient, validating tree fit accuracy.
cluster_assignment	A data.frame mapping each genotype/line identifier to its designated cluster label.
cluster_summary	A named list categorizing genotypes into vector groups corresponding to their assigned clusters.
cluster_means	A data.frame summarizing original trait mean values across each cluster group.
intra_cluster_dist	A named numeric vector of average within-cluster Euclidean spatial distances for each cluster.
inter_cluster_dist	A symmetric matrix representing Euclidean distances between cluster centroids in standardized space.
genotype_means	A data.frame of line-wise aggregated trait averages used as input for spatial scaling.

If reporting_level >= 1, comprehensive cluster summary tables and distance matrices are printed to the console prior to returning the list.

Examples

```
# Load benchmark breeding dataset
data(gv_data, package = "AgriDataTools")
```

```
# Specify trait columns matching gv_data structure
my_traits <- c("PH", "SL", "PL", "NOT", "NOSS", "TGW", "GYPM")

# Run cluster engine (Modify k as needed, e.g., 3, 4, 5, or 6)
cluster_results <- analyze_clustering(
  data = gv_data,
  traits = my_traits,
  k = 4,
  linkage_method = "ward.D2",
  reporting_level = 1
)
```

`analyze_pca` *Principal Component Analysis for Agronomic Traits*

Description

Performs Principal Component Analysis (PCA) on targeted quantitative agronomic parameters. Supports dynamic trait mapping to convert trait abbreviations into full descriptive names, and computes modern multivariate metrics including Kaiser-Guttman retention rules, eigenvector loadings, and percentage trait contributions.

Usage

```
analyze_pca(
  data,
  traits,
  scale = TRUE,
  reporting_level = 2,
  trait_lookup = NULL
)
```

Arguments

`data`	A data frame containing genotype information and trait columns.
`traits`	A character vector specifying the exact trait column names to include.
`scale`	Logical. If TRUE (default), variables are standardized to unit variance.
`reporting_level`	Integer. Control output verbosity: 0 (silent), 1 (summary), or 2 (exhaustive full reporting). Defaults to 2.
`trait_lookup`	An optional named character vector for mapping trait abbreviations to full descriptive labels (e.g., c("PH" = "Plant Height")).

Value

A structured named `list` containing 6 multivariate components:

`pca_object`	The raw `prcomp` output object.
`eigenvalues`	Data frame of eigenvalues, variance percentages, cumulative variance, and Kaiser retention decision.
`loadings`	Data frame of eigenvector loadings matrix.
`contributions`	Data frame of percentage contributions of each trait across components.
`cos2`	Data frame representing quality of representation (Cos^2) for each trait.
`scores`	Data frame of principal component scores assigned to individual genotypes.

Examples

```
library(AgriDataTools)
data("gv_data", package = "AgriDataTools")

# Define your custom trait mapping before running (Edit names as needed)
custom_traits_map <- c(
   "PH"   = "Plant Height",
   "SL"   = "Spike Length",
   "PL"   = "Peduncle Length",
   "NOT"  = "Number of Tillers",
   "NOSS" = "Number of Spikelets per Spike",
   "TGW"  = "Thousand Grain Weight",
   "GYPM" = "Grain Yield per Meter"
)

# Run Modern PCA with full trait names
pca_results <- analyze_pca(
   data = gv_data,
   traits = names(custom_traits_map),
   trait_lookup = custom_traits_map
)
```

`anova_crd` *Comprehensive Analysis of Variance (ANOVA) Engine for Completely Randomized Design (CRD)*

Description

The ‘anova_crd‘ function executes a complete, high-precision linear model analysis for agricultural, laboratory, or greenhouse trials laid out under a Completely Randomized Design (CRD). It computes partition sums of squares, hypothesis testing statistics, treatment variances, significance flags, and the Coefficient of Variation (CV

Usage

```
anova_crd(data, trait, reporting_level = 2)
```

Arguments

`data`	A verified `data.frame` containing the columns `Genotype` and the target phenotypic trait response.
`trait`	A single character string specifying the exact column name of the numeric trait to analyze.
`reporting_level`	An integer flag defining console trace settings: `0` for silent execution, `1` for printing basic ANOVA summary tables, and `2` for comprehensive diagnostic trace logs. Defaults to `2`.

Details

In laboratory experiments, growth chamber studies, or field trials with completely homogeneous environments, blocking is unnecessary. This function utilizes standard least-squares projection to build the classic orthogonal CRD ANOVA matrix, modeling the response vector as a function of treatment effects without blocking constraints:

$$Y_{ij} = \mu + T_i + \varepsilon_{ij}$$

Where T_i represents the treatment/genotype effect, and ε_{ij} is the residual experimental error. The function handles both balanced and unbalanced data structures perfectly, ensuring proper adjustments to degrees of freedom if replication numbers vary across lines.

Value

Invisibly returns a structured named `list` of class `"list"` containing 4 computational components:

`anova_table`	A `data.frame` acting as the standard ANOVA source matrix table for CRD, containing degrees of freedom, sums of squares, mean squares, F-statistics, and p-values.
`cv_percentage`	A numeric scalar representing the computed Coefficient of Variation percentage (CV%).
`mean_square_error`	A numeric scalar representing the isolated Residual Error Mean Square (EMS), ready for downstream evaluation.
`grand_mean`	A numeric scalar representing the overall arithmetic mean value of the evaluated phenotypic trait.

If `reporting_level` >= `1`, the compiled ANOVA table along with the grand mean and CV% are printed directly to the console before returning the list.

See Also

anova_rcbd

Examples

```
# Execute complete CRD partition on your actual target trait (GYPM)
crd_results <- anova_crd(data = gv_data, trait = "GYPM")
```

`anova_rcbd` *Comprehensive Analysis of Variance (ANOVA) Engine for Randomized Complete Block Design (RCBD)*

Description

The 'anova_rcbd' function executes a complete, high-precision linear model analysis for agricultural trials laid out under an RCBD framework. It computes partition sums of squares, hypothesis testing statistics, significance flags, and the Coefficient of Variation (CV

Usage

```
anova_rcbd(data, trait, reporting_level = 2)
```

Arguments

`data`	A verified `data.frame` containing the columns `Genotype`, `Replication`, and the target phenotypic trait response.
`trait`	A single character string specifying the exact column name of the numeric trait to analyze.
`reporting_level`	An integer vector flag defining console trace settings: `0` for silent, `1` for basic summary tables, and `2` for comprehensive descriptive metrics. Defaults to `2`.

Details

In plant breeding and agronomy trials, isolating block variance from the true experimental error is vital to properly evaluate lines, cultivars, or treatments. This function uses standard least-squares projection to build the classic orthogonal ANOVA matrix:

$$Y_{ij} = \mu + G_i + R_j + e_{ij}$$

Where G_i represents the genotype effect, R_j is the replication block effect, and e_{ij} is the residual experimental error.

Value

A structured named `list` containing 4 computational components:

`anova_table`	A `data.frame` acting as the standard ANOVA source matrix table containing Df, SS, MS, F_value, and p_value.
`cv_percentage`	The computed Coefficient of Variation percentage scalar ($CV\%$).

`mean_square_error`
: The isolated Residual Error Mean Square (EMS), ready for genetic parameter engines.

`grand_mean`
: The general mean arithmetic value of the evaluated trait.

See Also

`compute_lsd`

Examples

```
# Execute complete RCBD partition on gv_data asset for Plant Height (PH)
rcbd_results <- anova_rcbd(data = gv_data, trait = "PH")
```

---

compute_ancova *Analysis of Covariance (ANCOVA) Engine for Plant Breeding Trials*

Description

The ‘compute_ancova‘ function performs an exhaustive Analysis of Covariance across agricultural experimental records. It adjusts the treatment (genotypic) means for differences in a concomitant variable (covariate) to improve experimental precision and error control.

Usage

```
compute_ancova(data, response_trait, covariate_trait, reporting_level = 2)
```

Arguments

`data`
: A verified `data.frame` containing the experimental trial records.

`response_trait`
: A character string specifying the dependent phenotypic trait column.

`covariate_trait`
: A character string specifying the auxiliary covariate column.

`reporting_level`
: An integer vector flag defining console trace settings: `0` for silent execution, `1` for printing the variance partitioning table, and `2` for exhaustive diagnostic tracking logs. Defaults to `2`.

Details

In agricultural experiments, variation in a dependent phenotypic trait can sometimes be partially controlled by measuring a secondary auxiliary property (covariate, e.g., initial stand count or flowering duration). This engine fits a linear model incorporating both the categorical genotypic design structure and the continuous covariate vector, extracting the adjusted sums of squares, F-statistics, and adjusted marginal means.

Value

Invisibly returns a structured named `list` of class `"list"` containing 5 computational components:

`response_trait`	A character string indicating the target phenotypic response variable evaluated.
`covariate_trait`	A character string indicating the concomitant covariate variable utilized for model adjustment.
`model`	The underlying fitted linear model object of class `aov`.
`ancova_table`	A summary list structure of class `"summary.aov"` containing the Analysis of Covariance table with degrees of freedom, sums of squares, mean squares, F-values, and p-values.
`adjusted_means`	A `data.frame` containing the covariate-adjusted genotypic/cultivar means (least-squares means) calculated at the mean value of the covariate.

If `reporting_level >= 1`, the compiled ANCOVA table is printed directly to the console before returning the output object.

See Also

`validate_agri_data`, `compute_lsd`

Examples

```
data(gv_data, package = "AgriDataTools")

# Perform ANCOVA using Plant Height (PH) as response and Spike Length (SL) as covariate
ancova_res <- compute_ancova(
  data = gv_data,
  response_trait = "PH",
  covariate_trait = "SL",
  reporting_level = 2
)
print(ancova_res$ancova_table)
```

`compute_correlation` *Multi-Level Genetic, Phenotypic, and Environmental Correlation Engine*

Description

The `compute_correlation` function calculates genotypic (r_g), phenotypic (r_p), and environmental (r_e) correlation coefficient matrices across quantitative traits using analysis of variance (ANOVA) and covariance (ANCOVA) partitions, with integrated significance flags.

Usage

```
compute_correlation(data, traits = NULL, reporting_level = 1)
```

Arguments

`data`	A `data.frame` containing experimental phenotypic records with `Genotype` and `Replication` (or `Rep`) factors.
`traits`	A character vector specifying numeric trait columns to evaluate. Defaults to `NULL` for automatic detection.
`reporting_level`	An integer flag defining console trace settings: `0` for silent execution and `1` for displaying summary matrices with significance codes. Defaults to `1`.

Details

The engine partitions variance and covariance components using mean squares (MS) and mean cross-products (MCP):

$$r_g = \frac{Cov_g}{\sqrt{\sigma_{g1}^2 \cdot \sigma_{g2}^2}}$$

$$r_p = \frac{Cov_p}{\sqrt{\sigma_{p1}^2 \cdot \sigma_{p2}^2}}$$

$$r_e = \frac{Cov_e}{\sqrt{\sigma_{e1}^2 \cdot \sigma_{e2}^2}}$$

Value

Invisibly returns a structured named `list` of class `"list"` containing 9 correlation and significance matrices:

`genotypic_correlation`
: A `data.frame` matrix of genotypic correlation coefficients (r_g) between evaluated traits.

`phenotypic_correlation`
: A `data.frame` matrix of phenotypic correlation coefficients (r_p) between evaluated traits.

`environmental_correlation`
: A `data.frame` matrix of environmental correlation coefficients (r_e) between evaluated traits.

`genotypic_significance`
: A `data.frame` matrix of genotypic correlations formatted with significance stars (`***`, `**`, `*`, or `ns`).

`phenotypic_significance`
: A `data.frame` matrix of phenotypic correlations formatted with significance stars.

`environmental_significance`
: A `data.frame` matrix of environmental correlations formatted with significance stars.

`genotypic_p_values`
: A `data.frame` matrix containing raw calculated two-tailed p-values for genotypic correlations.

phenotypic_p_values
: A data.frame matrix containing raw calculated two-tailed p-values for phenotypic correlations.

environmental_p_values
: A data.frame matrix containing raw calculated two-tailed p-values for environmental correlations.

If reporting_level >= 1, formatted correlation matrices with significance flags are printed directly to the console before returning the output object.

Examples

```
# Load benchmark breeding dataset
data(gv_data, package = "AgriDataTools")

# Define trait columns
my_traits <- c("PH", "SL", "PL", "NOT", "NOSS", "TGW", "GYPM")

# Run multi-level correlation engine
corr_results <- compute_correlation(
  data = gv_data,
  traits = my_traits,
  reporting_level = 1
)
```

---

compute_lsd *Fisher's Least Significant Difference (LSD) Post-Hoc Mean Comparison Engine*

Description

The 'compute_lsd' function executes a rigorous, high-precision pairwise post-hoc mean separation analysis using Fisher's Least Significant Difference protocol. It isolates critical differences, evaluates pairwise significance metrics, and outputs comprehensive ranking tables with group letters.

Usage

```
compute_lsd(
  data,
  trait,
  anova_results,
  total_replications,
  alpha = 0.05,
  reporting_level = 2
)
```

Arguments

data	A verified data.frame containing the columns Genotype and the phenotypic trait under investigation.
trait	A single character string specifying the column name of the target trait.
anova_results	A structured list derived from upstream ANOVA layouts containing an anova_table.
total_replications	An integer specifying the absolute number of replication blocks (r).
alpha	A numeric value defining the Type-I error rate probability threshold. Defaults to 0.05.
reporting_level	An integer vector flag defining console trace settings: 0 for silent execution, 1 for summary, and 2 for exhaustive tracking. Defaults to 2.

Value

A structured named list of class "list" containing 4 computational components:

lsd_value	A numeric scalar representing the absolute calculated value of Fisher's Least Significant Difference at the specified alpha level.
sed	A numeric scalar representing the isolated Standard Error of Difference (SED) between two treatment means.
comparison_matrix	A data.frame or NULL layout reserved for detailed pairwise lines differences.
ranked_means	A data.frame containing lines/cultivars sorted by mean performance alongside their assigned statistical significance group letters (LSD_Letters).

Examples

```
# Load integrated data matrices
data(gv_data, package = "AgriDataTools")

# Total replications matching the experimental design blocks
reps <- length(unique(gv_data$Replication))

# Generating upstream ANOVA structure from gv_data for the target trait (PH)
model_fit <- aov(PH ~ Genotype + Replication, data = gv_data)
anova_summary <- summary(model_fit)[[1]]

mock_anova <- list(
  anova_table = data.frame(
    Source = c("Genotype", "Replication", "Error"),
    Df = anova_summary$Df,
    MS = anova_summary$`Mean Sq`,
    stringsAsFactors = FALSE
  )
)

# Running post-hoc separation sweeps with clean letters group layout
lsd_output <- compute_lsd(
```

```
  data = gv_data,
  trait = "PH",
  anova_results = mock_anova,
  total_replications = reps,
  alpha = 0.05
)
print(lsd_output$ranked_means)
```

`compute_path_analysis` *High-Precision Multi-Trait Cause-and-Effect Path Coefficient Analysis Engine*

Description

The 'compute_path_analysis' function executes phenotypic and genotypic path coefficient analysis based on the classic standard methodology of Dewey and Lu (1959). It partitions correlation coefficients between causal developmental traits and a target response variable into direct influence coefficients and indirect pathways acting through interconnected traits.

Usage

```
compute_path_analysis(
  correlation_payload,
  response_trait,
  predictor_traits = NULL,
  reporting_level = 2
)
```

Arguments

`correlation_payload`
: A structured `list` generated by `compute_correlation` containing `genotypic_correlation` and/or `phenotypic_correlation` matrices, or a single matrix.

`response_trait`
: A single character string specifying the target dependent trait column name.

`predictor_traits`
: A character vector identifying the causal predictor traits. If `NULL`, all remaining numeric traits except non-agronomic design factors and `response_trait` are automatically selected.

`reporting_level`
: An integer flag defining console output verbosity: `0` for silent execution, `1` for path decomposition summary, and `2` for detailed separate component tables. Defaults to `2`.

Details

Path coefficient analysis provides a matrix-based decomposition of direct and indirect components. For a target response variable, let R denote the correlation matrix among causal predictor traits, and r denote the vector of correlations between predictors and the response variable. Standardized direct path coefficients (β) are calculated via matrix inversion:

$$\beta = R^{-1}r$$

Indirect effects are cross-products between inter-trait correlations and direct path coefficients. The unexplained residual effect (R_X) is derived as:

$$R_X = \sqrt{1 - \sum(\beta_i \times r_i)}$$

Value

A structured named `list` containing path partition analyses for available correlation levels (`Genotypic` and/or `Phenotypic`):

`direct_effects` Numeric vector storing standardized direct path coefficients (β).

`indirect_effects_matrix`
: Data frame capturing inter-trait indirect path components alongside total correlation.

`total_correlation_vector`
: Original correlation alignment vector with the target response trait.

`residual_effect`
: Unexplained model residual variation (R_X).

`r_squared` Total variance explained by causal predictors (R^2).

`predictors` Character vector of causal predictor traits included in the pathway model.

See Also

compute_correlation, compute_summary_stats

Examples

```
# Compute multi-level correlation matrices
corr_payload <- compute_correlation(data = gv_data, reporting_level = 0)

# Define all 6 causal predictor traits driving Grain Yield per Meter
all_predictors <- c("PH", "SL", "PL", "NOT", "NOSS", "TGW")

# Run path coefficient analysis across all causal traits with full report
path_out <- compute_path_analysis(
  correlation_payload = corr_payload,
  response_trait = "GYPM",
  predictor_traits = all_predictors,
  reporting_level = 2
)
```

compute_scheffe *Scheffe's Post-Hoc Mean Comparison and Contrast Evaluation Engine*

Description

The 'compute_scheffe' function executes a rigorous, high-precision post-hoc mean separation analysis using Scheffe's method.

Usage

```
compute_scheffe(
  data,
  trait,
  anova_results,
  total_replications,
  alpha = 0.05,
  reporting_level = 2
)
```

Arguments

data	A verified data.frame containing the columns Genotype and the phenotypic trait under evaluation.
trait	A single character string specifying the column name of the target trait.
anova_results	A structured list derived from upstream ANOVA layouts.
total_replications	An integer specifying the absolute number of replication blocks (r).
alpha	A numeric value defining the Type-I error rate ceiling threshold. Defaults to 0.05.
reporting_level	An integer vector flag defining console trace settings. Defaults to 2.

Value

A structured named list of class "list" containing 4 computational components:

scheffe_critical_value	The absolute calculated scalar threshold value of Scheffe's adjustment.
sed	The isolated Standard Error of Difference (SED).
comparison_matrix	A detailed data frame or NULL layout reserved for pairwise lines differences.
ranked_means	A structured data frame containing sorted treatment means and significance group letters (Scheffe_Letters).

Author(s)

Faheem Khan (<2022ag94@uaf.edu.pk>)

Examples

```
data(gv_data, package = "AgriDataTools")
reps <- length(unique(gv_data$Replication))

model_fit <- aov(PH ~ Genotype + Replication, data = gv_data)
anova_summary <- summary(model_fit)[[1]]

mock_anova <- list(
  anova_table = data.frame(
    Source = c("Genotype", "Replication", "Error"),
    Df = anova_summary$Df,
    MS = anova_summary$`Mean Sq`,
    stringsAsFactors = FALSE
  )
)

scheffe_output <- compute_scheffe(
  data = gv_data,
  trait = "PH",
  anova_results = mock_anova,
  total_replications = reps
)
print(scheffe_output$ranked_means)
```

---

compute_summary_stats *Comprehensive Descriptive and Summary Statistics Engine for Phenotypic Traits*

Description

The ‘compute_summary_stats‘ function performs an exhaustive descriptive statistical sweep across multiple numeric traits in an agricultural dataset. It calculates central tendency, dispersion, and distribution shape metrics (skewness and kurtosis) for line screening.

Usage

```
compute_summary_stats(data, traits = NULL, reporting_level = 2)
```

Arguments

data	A verified data.frame containing the experimental trial records.
traits	A character vector specifying the exact column names to analyze. If NULL, the system automatically discovers and evaluates all numeric columns. Defaults to NULL.

`reporting_level`
: An integer vector flag defining console trace settings: `0` for silent, `1` for descriptive summary grids, and `2` for intensive diagnostic tracking. Defaults to `2`.

Details

Before executing hypothesis testing models like ANOVA, establishing dataset distribution profiles is critical. This engine parses target numerical vectors to extract metrics: Standard Error of the Mean is calculated as $SE = \frac{SD}{\sqrt{n}}$, Skewness measures distribution asymmetry, and Kurtosis indicates tail weight relative to a normal curve. It dynamically filters out environmental factors like 'Genotype' or 'Replication' and targets purely phenotypic observations.

Value

A detailed structured `data.frame` where rows represent traits and columns contain calculated metrics.

See Also

`validate_agri_data`

Examples

```
if (interactive()) {
   # Generate standard summary profiles across all phenotypic traits
   descriptive_grid <- compute_summary_stats(data = gv_data)
   print(descriptive_grid)
}
```

---

compute_tukey *Tukey's Honestly Significant Difference (HSD) Post-Hoc Mean Comparison Engine*

Description

The 'compute_tukey' function executes a high-precision pairwise post-hoc mean separation analysis using Tukey's Honestly Significant Difference framework.

Usage

```
compute_tukey(
  data,
  trait,
  anova_results,
  total_replications,
  alpha = 0.05,
  reporting_level = 2
)
```

Arguments

`data`	A verified `data.frame` containing the columns `Genotype` and the phenotypic trait under evaluation.
`trait`	A single character string specifying the column name of the target trait.
`anova_results`	A structured `list` derived from upstream ANOVA layouts.
`total_replications`	An integer specifying the absolute number of replication blocks (r).
`alpha`	A numeric value defining the adjusted family-wise error rate threshold. Defaults to `0.05`.
`reporting_level`	An integer vector flag defining console trace settings. Defaults to `2`.

Value

A structured named `list` of class `"list"` containing 4 computational components:

`tukey_value`	The absolute calculated scalar value of Tukey's Honestly Significant Difference threshold.
`se_mean`	The isolated Standard Error of the treatment mean scalar ($SE_{\bar{y}}$).
`comparison_matrix`	A detailed data frame or `NULL` layout reserved for pairwise lines differences.
`ranked_means`	A structured data frame containing sorted treatment means and significance group letters (`Tukey_Letters`).

Author(s)

Faheem Khan (<2022ag94@uaf.edu.pk>)

Examples

```
data(gv_data, package = "AgriDataTools")
reps <- length(unique(gv_data$Replication))

model_fit <- aov(PH ~ Genotype + Replication, data = gv_data)
anova_summary <- summary(model_fit)[[1]]

mock_anova <- list(
  anova_table = data.frame(
    Source = c("Genotype", "Replication", "Error"),
    Df = anova_summary$Df,
    MS = anova_summary$`Mean Sq`,
    stringsAsFactors = FALSE
  )
)

tukey_output <- compute_tukey(
  data = gv_data,
  trait = "PH",
```

```
  anova_results = mock_anova,
  total_replications = reps
)
print(tukey_output$ranked_means)
```

`estimate_variability` *Genetic Variability and Quantitative Inheritance Parameters Estimation Engine*

Description

The 'estimate_variability' function calculates comprehensive biometric genetic profiles from replication-based agricultural trial datasets. It partitions phenotypic variance into Genotypic Variance (Vg), Phenotypic Variance (Vp), and Environmental Variance (Ve), and computes critical breeding metrics including Genotypic Coefficient of Variation (GCV), Phenotypic Coefficient of Variation (PCV), Broad-Sense Heritability (H2), Genetic Advance (GA), and Genetic Advance as

Usage

```
estimate_variability(anova_results, total_replications, reporting_level = 2)
```

Arguments

`anova_results`	A structured `list` returned by either the `anova_rcbd` or `anova_crd` analysis pipelines within this package.
`total_replications`	An integer specifying the total number of replications/blocks used in the experimental trial layout.
`reporting_level`	An integer flag defining console output details: `0` for silent, `1` for summary parameter tables, and `2` for comprehensive descriptive logs. Defaults to `2`.

Details

In quantitative genetics, phenotypic variance must be dissected into its components to determine the role of genetic factors versus environmental noise. This function extracts the Error Mean Square (EMS) and Genotypic Mean Square (GMS) directly from completed ANOVA matrices:

$$V_g = \frac{GMS - EMS}{r}$$

$$V_p = V_g + EMS$$

$$H^2 = \frac{V_g}{V_p}$$

Where r represents the total absolute replication or block count.

If high environmental variations cause the computed Genotypic Variance to become negative, the system automatically applies a mathematical lower boundary floor at `0.0001` to preserve downstream pipeline integrity and issues a detailed structural warning message.

Value

A structured named `list` containing calculated genetic variability components:

`genotypic_variance`
: Estimated genotypic variance component (V_g).

`phenotypic_variance`
: Total phenotypic variance component (V_p).

`environmental_variance`
: Environmental variance/Error Mean Square (V_e).

`gcv`
: Genotypic Coefficient of Variation percentage (GCV%).

`pcv`
: Phenotypic Coefficient of Variation percentage (PCV%).

`heritability_percentage`
: Broad-Sense Heritability percentage ($H^2\%$).

`genetic_advance`
: Expected Genetic Advance (GA) at 5% selection intensity ($k = 2.06$).

`gam_percentage`
: Genetic Advance as a percentage of the Grand Mean (GAM%).

See Also

anova_rcbd, anova_crd

Examples

```
if (interactive()) {
   # Execute complete genetic variability partitioning
   rcbd_out <- anova_rcbd(data = gv_data, trait = "PH", reporting_level = 0)
   var_metrics <- estimate_variability(anova_results = rcbd_out, total_replications = 3)
   print(var_metrics$heritability_percentage)
}
```

---

gv_data *Genotypic Variability and Agricultural Research Dataset*

Description

Evaluated performance parameters across multiple line and cultivar iterations under a randomized complete block layout.

Usage

```
data(gv_data)
```

Format

A `data.frame` containing phenotypic records with wheat morphological traits:

Genotype Factor or character vector identifying evaluated breeding germplasm lines or cultivars.

Replication Factor or integer vector indicating experimental replications or blocks within the layout.

PH Plant Height measured in centimeters (cm).

PL Peduncle Length measured in centimeters (cm).

SL Spike Length measured in centimeters (cm).

NOT Number of tillers per plant.

NOSS Number of spikelets per spike.

TGW Thousand grain weight measured in grams (g).

GYPM Grain yield per meter measured in grams (g).

Source

Experimental plant breeding field trial records.

Examples

```
data(gv_data, package = "AgriDataTools")
head(gv_data)
```

---

plot_agri_graphics *Advanced High-Precision Publication-Ready Graphics Suite*

Description

The ‘plot_agri_graphics‘ function serves as the unified visualization hub for the AgriDataTools package. It handles basic statistical diagnostics (residuals, correlations) alongside modern publication-grade graphical representations for mean performance, PCA space, hierarchical dendrograms, and path analysis direct effects.

Usage

```
plot_agri_graphics(
  type,
  payload,
  trait_name = "Target Character Matrix",
  reporting_level = 2,
  num_clusters = 4
)
```

Arguments

type	A single character string specifying the target chart module: "residual", "correlation", "mean", "pca", "cluster", or "path".
payload	A structured analysis list derived from computational engines (e.g., compute_lsd, analyze_pca, analyze_clustering, compute_path_analysis).
trait_name	A character string defining the target phenotypic trait title label. Used primarily in "mean" and "path" layouts.
reporting_level	An integer vector flag defining console trace settings: 0 for silent, 1 for structural updates, and 2 for exhaustive analytical tracing. Defaults to 2.
num_clusters	An integer specifying the number of cluster groups to color in the circular dendrogram module. Defaults to 4.

Details

Visualizing high-dimensional screening metrics across diverse lines or cultivars requires balancing diagnostic model validation checks with advanced multivariate aesthetics. This engine supports base diagnostic rendering as well as optimized ggplot2 geometries featuring dynamic color palettes, non-overlapping labels, and geometric layout vector mapping fields.

Value

Invisibly returns a logical scalar TRUE upon successful execution. This function is primarily invoked for its side effect of rendering publication-grade graphical plots (e.g., residual diagnostic plots, correlation heatmaps, mean performance barcharts, PCA biplots, circular dendrograms, or path analysis plots) to the active graphics device.

Examples

```
data(gv_data, package = "AgriDataTools")
traits <- c("PH", "SL", "PL", "NOT", "NOSS", "TGW", "GYPM")

# Define custom mapping or number of clusters beforehand
k_groups <- 4

# 1. Mean performance: Genotypic performance with LSD
reps <- length(unique(gv_data$Replication))
fit <- aov(PH ~ Genotype + Replication, data = gv_data)
m_anova <- list(anova_table = data.frame(
   Source = c("Genotype", "Replication", "Error"),
   Df = summary(fit)[[1]]$Df,
   MS = summary(fit)[[1]][[3]]
))
lsd_res <- compute_lsd(gv_data, "PH", m_anova, reps)
plot_agri_graphics(type = "mean", payload = lsd_res,
                   trait_name = "Plant Height")

# 2. PCA: Multivariate variation
custom_traits_map <- c(
```

```
     "PH"    = "Plant Height",
     "SL"    = "Spike Length",
     "PL"    = "Peduncle Length",
     "NOT"   = "Number of Tillers",
     "NOSS"  = "Number of Spikelets per Spike",
     "TGW"   = "Thousand Grain Weight",
     "GYPM"  = "Grain Yield per Meter"
)
pca_res <- analyze_pca(data = gv_data, traits = traits, trait_lookup = custom_traits_map)
plot_agri_graphics(type = "pca", payload = pca_res,
                   trait_name = "PCA Plot")

# 3. Clustering: Dendrogram with flexible cluster parameter option
cl_res <- analyze_clustering(data = gv_data, traits = traits, k = k_groups)
plot_agri_graphics(type = "cluster", payload = cl_res,
                   trait_name = "Clustering", num_clusters = k_groups)

# 4. Residuals: Diagnostic plots
fit <- lm(PH ~ Genotype, data = gv_data)
res_pl <- list(residuals = residuals(fit),
               fitted_values = fitted(fit))
plot_agri_graphics(type = "residual", payload = res_pl,
                   trait_name = "Residuals")

# 5. Correlations: Phenotypic matrix
cor_m <- cor(gv_data[, traits], use = "pairwise.complete.obs")
plot_agri_graphics(type = "correlation",
                   payload = list(correlation_matrix = cor_m),
                   trait_name = "Correlation")

# 6. Path Analysis: Direct Effects Plot for Grain Yield per Meter (GYPM)
corr_res <- compute_correlation(data = gv_data, traits = traits, reporting_level = 0)
all_predictors <- c("PH", "SL", "PL", "NOT", "NOSS", "TGW")
path_res <- compute_path_analysis(
   correlation_payload = corr_res,
   response_trait = "GYPM",
   predictor_traits = all_predictors,
   reporting_level = 0
)
plot_agri_graphics(type = "path", payload = path_res, trait_name = "Grain Yield per Meter (GYPM)")
```

`validate_agri_data` *Rigid Multi-Layered Structural Data Validation and Matrix Integrity Engine*

Description

The ‘validate_agri_data‘ function serves as the primary data defense matrix of the `AgriDataTools` package. It is engineered to perform exhaustive, multi-dimensional quality control, type alignment

checks, and semantic structural audits on user-provided datasets before they are passed to sensitive breeding and quantitative genetic workflows.

Usage

```
validate_agri_data(data, strict_mode = TRUE, reporting_level = 2)
```

Arguments

`data`	A non-null `data.frame` containing the experimental field trial records.
`strict_mode`	A logical scalar. If `TRUE`, the validation engine demands an absolute structural match with the core dataset archetype: exactly 120 rows, 40 distinct genotypes, 3 replications, and all 7 mandatory phenotypic traits (`PH`, `SL`, `PL`, `NOT`, `NOSS`, `TGW`, `GYPM`). Defaults to `TRUE`.
`reporting_level`	An integer mapping scale indicating console log verbosity: `0` for dead silent, `1` for critical system steps, and `2` for exhaustive vector diagnostics. Defaults to `2`.

Details

In biometric computing, agricultural research, and plant breeding quantitative data analysis, downstream models like ANOVA, Heritability estimations, and Path analysis are highly vulnerable to layout irregularities. Silent formatting issues inside spreadsheets can bias variance components or trigger generic engine failures.

To solve this, ‘validate_agri_data‘ runs an automated, multi-tiered defensive pipeline:

1. **Object and Class Matrix Verification:** Ensures the input object is a true dataframe.
2. **Dimension Capability Evaluation:** Checks for absolute row/column structural thresholds.
3. **Strict Column Header Matching:** Verifies the existence of structural keys and core traits.
4. **Categorical Factor Integrity Audits:** Verifies grouping layouts for Genotypes and Replications.
5. **Quantitative Type Compliance Testing:** Validates phenotypic trait vectors for numeric compliance.
6. **Biological Range and Bound Inspections:** Screens for mathematical anomalies like negative values.
7. **Missing Value (NA) Variance Profiling:** Analyzes missing data distribution across blocks.

Value

A logical scalar `TRUE` if the dataset completely satisfies the operational constraints of the biometrical pipeline. If any structural non-compliance is detected, it throws a highly structured, informative exception detailing the exact coordinates of the failure.

References

- Cochran, W.G. and Cox, G.M. (1957). Experimental Designs. 2nd Edition, John Wiley & Sons, New York.
- Falconer, D.S. and Mackay, T.F.C. (1996). Introduction to Quantitative Genetics. 4th Edition, Longman, Essex.

Examples

```
# Assuming package environments are active and datasets are loaded via data(gv_data)
if (interactive()) {
    # Trigger standard validation sweep
    validation_status <- validate_agri_data(data = gv_data, strict_mode = TRUE)
    message("Is dataset operational? ", validation_status)
}
```

Index