

Package: AgriDataTools (via r-universe)

October 5, 2026

Type Package

Title Automated Statistical Analysis and Tools for Agricultural Research

Version 0.2.1

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, ggrepel, stats, graphics, dplyr, utils

Config/roxygen2/version 8.0.0

VignetteBuilder knitr
Suggests knitr, rmarkdown, testthat (>= 3.0.0)
URL <https://faheemkhan15326.github.io/AgriDataTools/>
BugReports <https://github.com/faheemkhan15326/AgriDataTools/issues>
Config/pak/sysreqs cmake libfreetype6-dev make libicu-dev libpng-dev libuv1-dev zlib1g-dev
Repository <https://faheemkhan15326.r-universe.dev>
Date/Publication 2026-09-05 08:16:53 UTC
RemoteUrl <https://github.com/faheemkhan15326/agridatatools>
RemoteRef HEAD
RemoteSha 0f3b955ec48611d46e432de967e4d5b4ed669427

Contents

analyze_clustering	2
analyze_pca	4
anova_crd	6
anova_rcbd	7
compute_ancova	8
compute_correlation	10
compute_lsd	12
compute_path_analysis	13
compute_scheffe	15
compute_summary_stats	16
compute_tukey	17
estimate_variability	19
gv_data	20
plot_agri_graphics	21
validate_agri_data	24
Index	26

analyze_clustering	<i>Hierarchical Cluster Analysis and Phenotypic Diversity Engine</i>
--------------------	--

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,
  genotype_col = NULL,
  k = 4,
  linkage_method = "ward.D2",
  reporting_level = 1
)
```

Arguments

<code>data</code>	A <code>data.frame</code> containing phenotypic records with genotype identifiers and numeric traits.
<code>traits</code>	A character vector specifying the quantitative traits to be integrated into the cluster matrix.
<code>genotype_col</code>	A character string specifying the column name for genotypes. If <code>NULL</code> , automatic column detection is performed. Defaults to <code>NULL</code> .
<code>k</code>	An integer specifying the target number of clusters to partition the tree. Defaults to 4.
<code>linkage_method</code>	A character string specifying the target agglomerative clustering algorithm (e.g., "ward.D2", "complete", "average"). Defaults to "ward.D2".
<code>reporting_level</code>	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:

<code>dist_matrix</code>	A spatial <code>dist</code> object representing calculated multidimensional Euclidean distances between genotypes based on standardized phenotypic scores.
<code>hc_object</code>	The raw hierarchical clustering output object of class <code>hclust</code> .
<code>cophenetic_corr</code>	A numeric value indicating the cophenetic correlation coefficient, validating tree fit accuracy.
<code>cluster_assignment</code>	A <code>data.frame</code> mapping each genotype/line identifier to its designated cluster label.
<code>cluster_summary</code>	A named list categorizing genotypes into vector groups corresponding to their assigned clusters.
<code>cluster_means</code>	A <code>data.frame</code> summarizing original trait mean values across each cluster group.
<code>intra_cluster_dist</code>	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 your own dataset
data(gv_data, package = "AgriDataTools")

# Specify trait columns matching your dataset structure
traits <- c("PH", "SL", "PL", "NOT", "NOSS", "TGW", "GYPM")

# Run cluster engine (Modify k as needed, e.g., 3, 4, 5, or 12)
cluster_results <- analyze_clustering(
  data = gv_data,
  traits = traits,
  genotype_col = "Genotype",
  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, dynamic genotype column recognition, automatic replication aggregation to genotypic means, and computes modern multivariate metrics including Kaiser-Guttman retention rules, eigenvector loadings, and percentage trait contributions.

Usage

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

Arguments

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

Value

A structured named list containing 6 multivariate components:

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

Examples

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

# Define 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),
  genotype_col = "Genotype",
  trait_lookup = custom_traits_map
)
```

anova_crd	<i>Comprehensive Analysis of Variance (ANOVA) Engine for Completely Randomized Design (CRD)</i>
-----------	---

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, genotype_col = NULL, reporting_level = 1)
```

Arguments

<code>data</code>	A verified <code>data.frame</code> containing genotype/treatment identifiers and phenotypic responses.
<code>trait</code>	A single character string specifying the exact column name of the numeric trait to analyze.
<code>genotype_col</code>	An optional character string specifying the column name for genotypes. If <code>NULL</code> , automatic column detection is performed. Defaults to <code>NULL</code> .
<code>reporting_level</code>	An integer flag defining console trace settings: 0 for silent execution and 1 for printing formatted ANOVA summary tables. Defaults to 1.

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 containing 4 computational components:

- **anova_table**: A `data.frame` acting as the standard ANOVA source matrix table for CRD.
- **cv_percentage**: A numeric scalar representing the computed Coefficient of Variation percentage.
- **mean_square_error**: A numeric scalar representing the isolated Residual Error Mean Square.
- **grand_mean**: A numeric scalar representing the overall arithmetic mean value.

See Also

[anova_rcbd](#)

Examples

```
# Load your own dataset
data(gv_data, package = "AgriDataTools")
# Execute complete CRD partition on target trait Plant Height(PH)
crd_results <- anova_crd(data = gv_data, trait = "PH", genotype_col = "Genotype")
```

anova_rcbd	<i>Comprehensive Analysis of Variance (ANOVA) Engine for Randomized Complete Block Design (RCBD)</i>
------------	--

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,
  genotype_col = NULL,
  rep_col = NULL,
  reporting_level = 1
)
```

Arguments

data	A verified data.frame containing genotype, replication block, and phenotypic response vectors.
trait	A single character string specifying the exact column name of the numeric trait to analyze.
genotype_col	An optional character string specifying the column name for genotypes. If NULL, automatic column detection is performed. Defaults to NULL.
rep_col	An optional character string specifying the column name for replications/blocks. If NULL, automatic column detection is performed. Defaults to NULL.
reporting_level	An integer flag defining console trace settings: 0 for silent execution, and 1 for printing formatted summary tables. Defaults to 1.

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

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 containing Df, SS, MS, F_value, p_value, and significance flags.
- **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

[anova_crd](#)

Examples

```
# Load your own dataset
data(gv_data, package = "AgriDataTools")
# Execute complete RCBD partition on target trait Plant Height(PH)
rcbd_results <- anova_rcbd(data = gv_data, trait = "PH")
```

compute_ancova	<i>Compact Analysis of Covariance (ANCOVA) Table Engine for Plant Breeding</i>
----------------	--

Description

The compute_ancova function evaluates an Analysis of Covariance (ANCOVA) and compiles a streamlined, standard biometrical table featuring Degrees of Freedom (Df), Sum of Products (SP_XY), Mean Products (MP_XY), exact Biometrical F-values, p-values, significance flags, and rigorous Covariance components (Cov_e, Cov_g, Cov_p).

Usage

```
compute_ancova(
  data,
  response_trait,
  covariate_trait,
  genotype_col = NULL,
  rep_col = NULL,
  reporting_level = 1
)
```

Arguments

data A verified data.frame containing experimental trial records.

response_trait A character string specifying the dependent phenotypic response variable (e.g., "GYPM").

covariate_trait A character string specifying the auxiliary covariate variable (e.g., "PH").

genotype_col Optional character string specifying the genotype column. Defaults to NULL for auto-detection.

rep_col Optional character string specifying the replication column. Defaults to NULL for auto-detection.

reporting_level An integer flag: 0 for silent execution, 1 for printing the ANCOVA table. Defaults to 1.

Details

The mathematical partitioning under RCBD framework is computed using joint reduction methods:

$$SP_{Total} = SP_{Replications} + SP_{Genotypes} + SP_{Residual}$$

$$MP_{XY} = \frac{SP_{XY}}{Df}$$

$$Cov_e = MP_{Error}$$

$$Cov_g = \frac{MP_{Genotypes} - MP_{Error}}{r}$$

$$Cov_p = Cov_g + \frac{Cov_e}{r}$$

Value

Invisibly returns a structured list containing:

ancova_table A compact data.frame containing the exact ANCOVA source table.

covariance_components A named numeric vector containing isolated Cov_e, Cov_g, and Cov_p values.

adjusted_means A data.frame containing covariate-adjusted genotypic least-squares means.

Examples

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

# Run Analysis of Covariance between Grain Yield per Meter and Plant Height
ancova_results <- compute_ancova(
  data = gv_data,
  response_trait = "GYPM",
  covariate_trait = "PH"
)
```

compute_correlation	<i>Multi-Level Genetic, Phenotypic, and Environmental Correlation Engine</i>
---------------------	--

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,
  genotype_col = NULL,
  rep_col = NULL,
  reporting_level = 1
)
```

Arguments

<code>data</code>	A data.frame containing experimental phenotypic records with Genotype and Replication (or Rep) factors.
<code>traits</code>	A character vector specifying numeric trait columns to evaluate. Defaults to NULL for automatic detection.
<code>genotype_col</code>	Optional character string specifying the genotype column. Defaults to NULL for auto-detection.
<code>rep_col</code>	Optional character string specifying the replication column. Defaults to NULL for auto-detection.
<code>reporting_level</code>	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 containing 9 correlation and significance matrices:

genotypic_correlation

A data.frame matrix of genotypic correlation coefficients (r_g).

phenotypic_correlation

A data.frame matrix of phenotypic correlation coefficients (r_p).

environmental_correlation

A data.frame matrix of environmental correlation coefficients (r_e).

genotypic_significance

A data.frame matrix of genotypic correlations formatted with significance stars.

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 p-values for genotypic correlations.

phenotypic_p_values

A data.frame matrix containing raw calculated p-values for phenotypic correlations.

environmental_p_values

A data.frame matrix containing raw calculated p-values for environmental correlations.

Examples

```
# Load your own dataset
data(gv_data, package = "AgriDataTools")

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

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

compute_lsd	<i>Fisher’s Least Significant Difference (LSD) Post-Hoc Mean Comparison Engine</i>
-------------	--

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,
  geno_col = NULL,
  alpha = 0.05,
  reporting_level = 2
)
```

Arguments

data	A verified data.frame containing treatment lines/cultivars and the target phenotypic trait.
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 (<i>r</i>).
geno_col	A character string specifying the treatment/genotype column name. Defaults to auto-detecting "Genotype".
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 layout containing detailed pairwise line differences, standard errors, and significance markers.
ranked_means	A data.frame containing lines/cultivars sorted by mean performance alongside their assigned statistical significance group letters (LSD_Letters).

Examples

```
data(gv_data, package = "AgriDataTools")
reps <- length(unique(gv_data$Replication))
rcbd_results <- anova_rcbd(data = gv_data, trait = "PH", reporting_level = 0)

lsd_output <- compute_lsd(
  data = gv_data,
  trait = "PH",
  anova_results = rcbd_results,
  total_replications = reps,
  alpha = 0.05
)
```

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](#)

Examples

```

data(gv_data, package = "AgriDataTools")
corr_results <- compute_correlation(data = gv_data, reporting_level = 0)
all_predictors <- c("PH", "SL", "PL", "NOT", "NOSS", "TGW")
path_results <- compute_path_analysis(
  correlation_payload = corr_results,
  response_trait = "GYPM",
  predictor_traits = all_predictors,
  reporting_level = 2
)

```

compute_scheffe	<i>Scheffe's Post-Hoc Mean Comparison and Contrast Evaluation Engine</i>
-----------------	--

Description

The compute_scheffe function executes a rigorous, high-precision post-hoc mean separation analysis using Scheffe's method. It evaluates critical differences, pairwise contrasts, and outputs comprehensive ranking tables with statistical group letters.

Usage

```

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

```

Arguments

data	A verified data.frame containing treatment lines/cultivars and the target phenotypic trait.
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).
geno_col	A character string specifying the treatment/genotype column name. Defaults to auto-detecting "Genotype".
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: 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:

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 containing pairwise differences, F-ratios, p-values, and significance status.
ranked_means	A structured data frame containing sorted treatment means and significance group letters (Scheffe_Letters).

Examples

```
data(gv_data, package = "AgriDataTools")
reps <- length(unique(gv_data$Replication))
rcbd_results <- anova_rcbd(data = gv_data, trait = "PH", reporting_level = 0)

scheffe_output <- compute_scheffe(
  data = gv_data,
  trait = "PH",
  anova_results = rcbd_results,
  total_replications = reps
)
```

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 = 1)
```

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. Defaults to 1.

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

```
# Generate standard summary profiles across all phenotypic traits
descriptive_grid <- compute_summary_stats(data = gv_data)
```

compute_tukey	<i>Tukey's Honestly Significant Difference (HSD) Post-Hoc Mean Comparison Engine</i>
---------------	--

Description

The compute_tukey function executes a high-precision pairwise post-hoc mean separation analysis using Tukey's Honestly Significant Difference framework. It calculates family-wise error control thresholds, pairwise differences, and outputs comprehensive ranking tables with group letters.

Usage

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

Arguments

<code>data</code>	A verified <code>data.frame</code> containing treatment lines/cultivars and the target phenotypic trait.
<code>trait</code>	A single character string specifying the column name of the target trait.
<code>anova_results</code>	A structured list derived from upstream ANOVA layouts containing an <code>anova_table</code> .
<code>total_replications</code>	An integer specifying the absolute number of replication blocks (r).
<code>geno_col</code>	A character string specifying the treatment/genotype column name. Defaults to auto-detecting "Genotype".
<code>alpha</code>	A numeric value defining the adjusted family-wise error rate threshold. Defaults to 0.05.
<code>reporting_level</code>	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:

<code>tukey_value</code>	The absolute calculated scalar value of Tukey's Honestly Significant Difference threshold.
<code>se_mean</code>	The isolated Standard Error of the treatment mean scalar ($SE_{\bar{y}}$).
<code>comparison_matrix</code>	A detailed data frame containing pairwise differences, q-statistics, p-values, and significance markers.
<code>ranked_means</code>	A structured data frame containing sorted treatment means and significance group letters (Tukey_Letters).

Examples

```
data(gv_data, package = "AgriDataTools")
reps <- length(unique(gv_data$Replication))
rcbd_results <- anova_rcbd(data = gv_data, trait = "PH", reporting_level = 0)

tukey_output <- compute_tukey(
  data = gv_data,
  trait = "PH",
  anova_results = rcbd_results,
  total_replications = reps
)
```

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 (V_g), Phenotypic Variance (V_p), and Environmental Variance (V_e), and computes critical breeding metrics including Genotypic Coefficient of Variation (GCV), Phenotypic Coefficient of Variation (PCV), Broad-Sense Heritability (H^2), Genetic Advance (GA), and Genetic Advance as

Usage

```
estimate_variability(anova_results, total_replications, reporting_level = 1)
```

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 table. Defaults to 1.

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 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

```
data(gv_data, package = "AgriDataTools")
reps <- length(unique(gv_data$Replication))
rcbd_results <- anova_rcbd(data = gv_data, trait = "PH", reporting_level = 0)
var_metrics <- estimate_variability(anova_results = rcbd_results, total_replications = reps)
```

gv_data	<i>Genotypic Variability and Agricultural Research Dataset</i>
---------	--

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

Original experimentSal data from a wheat field trial conducted under Randomized Complete Block Design (RCBD) involving multiple genotypes and phenotypic traits.

Examples

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

plot_agri_graphics	<i>Advanced High-Precision Publication-Ready Graphics Suite</i>
--------------------	---

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 = 0,
  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 0 (Clean Console).
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
if (interactive()) {
  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_output <- compute_lsd(gv_data, "PH", m_anova, reps, reporting_level = 0)
  plot_agri_graphics(type = "mean", payload = lsd_output,
    trait_name = "Plant Height")
}
```

```

# 2. PCA: Multivariate variation with unique actual genotype names
if (interactive()) {
  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_results <- analyze_pca(data = gv_data, traits = traits,
                             trait_lookup = custom_traits_map,
                             reporting_level = 0)
  plot_agri_graphics(type = "pca", payload = pca_results,
                     trait_name = "PCA Plot")
}

# 3. Clustering: Dendrogram with flexible cluster parameter option
if (interactive()) {
  cluster_results <- analyze_clustering(data = gv_data, traits = traits,
                                         k = k_groups, reporting_level = 0)
  plot_agri_graphics(type = "cluster", payload = cluster_results,
                     trait_name = "Clustering", num_clusters = k_groups)
}

# 4. Residuals: Diagnostic plots
if (interactive()) {
  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
if (interactive()) {
  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)
if (interactive()) {
  corr_results <- compute_correlation(data = gv_data, traits = traits, reporting_level = 0)
  all_predictors <- c("PH", "SL", "PL", "NOT", "NOSS", "TGW")
  path_results <- compute_path_analysis(
    correlation_payload = corr_results,
    response_trait = "GYPM",
    predictor_traits = all_predictors,
    reporting_level = 0
  )
}

```

```
plot_agri_graphics(type = "path", payload = path_results,
  trait_name = "Grain Yield per Meter (GYPM)")
}
```

validate_agri_data	<i>Rigid Multi-Layered Structural Data Validation and Matrix Integrity Engine</i>
--------------------	---

Description

The `validate_agri_data` function serves as the primary data defense matrix of the `AgriDataTools` package. It performs automated multi-dimensional quality control, type alignment checks, and semantic structural audits on agricultural research datasets before passing them to downstream quantitative genetic workflows.

Usage

```
validate_agri_data(data, reporting_level = 1)
```

Arguments

data	A non-null <code>data.frame</code> containing experimental field trial records.
reporting_level	An integer mapping scale for console trace output: 0 for silent, 1 for clean summary status report. Defaults to 1.

Details

In quantitative genetics, downstream models like ANOVA, Heritability estimations, and Path analysis are highly vulnerable to layout irregularities. Silent formatting issues can bias variance components or trigger execution failures.

`validate_agri_data` runs a dynamic defensive pipeline:

1. **Object Matrix Verification:** Ensures input is a valid `data.frame` with observations.
2. **Dynamic Header Discovery:** Auto-detects Genotype, Replication, and Trait columns without rigid hardcoding.
3. **Factor Balance Audits:** Verifies minimum levels for lines and blocks.
4. **Numeric Type & Range Compliance:** Validates numeric integrity and flags biological anomalies (e.g., negative values).
5. **Orthogonality Sweep:** Evaluates design layout cell frequencies for balance.

Value

A logical scalar `TRUE` if the dataset completely satisfies the operational constraints of the biometrical pipeline. Throws informative errors if critical structural faults are detected.

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

```
# Trigger dynamic validation sweep  
validation_status <- validate_agri_data(data = gv_data)
```

Index

* datasets

- gv_data, [20](#)
- analyze_clustering, [2](#)
- analyze_pca, [4](#)
- anova_crd, [6](#), [8](#), [20](#)
- anova_rcbd, [7](#), [7](#), [20](#)
- compute_ancova, [8](#)
- compute_correlation, [10](#), [14](#)
- compute_lsd, [12](#)
- compute_path_analysis, [13](#)
- compute_scheffe, [15](#)
- compute_summary_stats, [16](#)
- compute_tukey, [17](#)
- estimate_variability, [19](#)
- gv_data, [20](#)
- hclust, [3](#)
- plot_agri_graphics, [21](#)
- validate_agri_data, [17](#), [24](#)