Package: statease (via r-universe)

July 3, 2026

Title Simplified Statistical Analysis with Plain-English Interpretation

Version 1.3.0

Description A toolkit for common statistical analyses including descriptive statistics, Student's t-tests (one-sample, independent, and paired), one-way and two-way Analysis of Variance (ANOVA), Multivariate Analysis of Variance (MANOVA), chi-square tests, Fisher's Exact Test, McNemar's Test, correlation analysis, simple and multiple linear regression, logistic regression, Friedman Test, and non-parametric tests (Mann-Whitney U, Wilcoxon Signed Rank, and Kruskal-Wallis). Additional tools include statistical power analysis and automated assumption checking. Each function automatically interprets results in plain English, reporting effect sizes, confidence intervals, and p-value interpretations. Post-hoc tests are automatically applied following significant results. A master function automatically detects the appropriate test based on the structure of the input data. Methods are based on Cohen, J. (1988) <doi:10.4324/9780203771587>, Tukey, J. W. (1949) <doi:10.2307/3001913>, and Shapiro and Wilk (1965) <doi:10.2307/2333709>.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/DevWebWacky/statease, https://devwebwacky.github.io/statease/

BugReports https://github.com/DevWebWacky/statease/issues

Config/roxygen2/version 8.0.0

Suggests DT, shinydashboard, shinyjs, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Imports car, pwr, shiny,

Config/testthat/edition 3

NeedsCompilation no

Author Uwakmfon Paul [aut, cre, cph]

Maintainer Uwakmfon Paul <uwakmfon31@gmail.com>

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

Date/Publication 2026-07-02 07:00:10 UTC

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

RemoteRef HEAD

RemoteSha 06b2ccbdaccd2676c04e278e0f4fbef7dce0124c

Contents

`analyze` *Master Analysis Function - Auto-detects and runs the right test*

Description

Master Analysis Function - Auto-detects and runs the right test

Usage

```
analyze(
  x = NULL,
  y = NULL,
  data = NULL,
  formula = NULL,
  mu = 0,
  paired = FALSE,
  nonparam = FALSE,
  conf.level = 0.95,
  var_name = "Variable",
  var1_name = "Variable 1",
  var2_name = "Variable 2",
  method = "pearson",
  test_type = NULL,
  effect_size = NULL,
  power = 0.8,
  n_groups = 2,
  n_predictors = 1,
  check = FALSE
)
```

Arguments

x	A numeric vector (required always)
y	A numeric vector, factor, or character group variable (optional)
data	A data frame (required if using a formula)
formula	A formula of the form outcome ~ predictor or outcome ~ group1 * group2 or cbind(y1, y2) ~ group (optional)
mu	Hypothesised mean for one-sample t-test. Default 0.
paired	Logical. TRUE for paired t-test. Default FALSE.
nonparam	Logical. TRUE to use non-parametric tests. Default FALSE.
conf.level	Confidence level. Default 0.95.
var_name	Optional label for the report.
var1_name	Optional name for first variable in correlation.
var2_name	Optional name for second variable in correlation.
method	Correlation method: "pearson", "spearman", or "kendall". Default "pearson".
test_type	For power analysis: one of "ttest.one", "ttest.two", "ttest.paired", "anova", "correlation", "chisq", "regression".
effect_size	For power analysis: the expected effect size.
power	For power analysis: desired power level. Default 0.80.
n_groups	For power analysis ANOVA: number of groups. Default 2.
n_predictors	For power analysis regression: number of predictors. Default 1.
check	Logical. TRUE to run assumption checks before analysis. Default FALSE.

Value

A printed analysis report from the appropriate test

Examples

```
# Descriptive only
analyze(x = c(23, 45, 12, 67, 34))

# Auto t-test
analyze(x = c(23,45,12,67,34), y = c(19,38,22,51,29))

# Auto ANOVA
df <- data.frame(
  score = c(23,45,12,67,34,89,56,43,78,90,11,34),
  group = rep(c("A","B","C"), each = 4)
)
analyze(formula = score ~ group, data = df)

# Power analysis
analyze(test_type = "ttest.two", effect_size = 0.5)

# Check assumptions
analyze(x = c(23,45,12,67,34), y = c(19,38,22,51,29),
        check = TRUE)
```

---

anova_interpret *One-Way ANOVA with Post-Hoc Tukey and Plain-English Interpretation*

Description

One-Way ANOVA with Post-Hoc Tukey and Plain-English Interpretation

Usage

```
anova_interpret(formula, data, conf.level = 0.95)
```

Arguments

formula	A formula of the form outcome ~ group
data	A data frame containing the variables
conf.level	Confidence level. Default 0.95

Value

An object of class `statease_anova` containing ANOVA results, effect size, and post-hoc comparisons. Use `print()` to display the formatted report.

Examples

```
df <- data.frame(
  score = c(23,45,12,67,34,89,56,43,78,90,11,34),
  group = rep(c("A","B","C"), each = 4)
)
result <- anova_interpret(score ~ group, data = df)
print(result)
```

`anova2_interpret` *Two-Way ANOVA with Plain-English Interpretation*

Description

Uses Type-2 SS by default (safe for unbalanced designs). Automatically switches to Type-3 SS when an interaction term is detected and sets the correct contrasts. Users are warned when interpreting main effects in the presence of a significant interaction.

Usage

```
anova2_interpret(formula, data, type = 2, conf.level = 0.95)
```

Arguments

`formula`	A formula of the form outcome ~ group1 * group2
`data`	A data frame containing the variables
`type`	ANOVA type: 2 or 3. Default is 2. Type 3 is automatically used when an interaction term is detected in the formula.
`conf.level`	Confidence level. Default 0.95.

Value

An object of class `statease_anova2` containing two-way ANOVA results and interpretation. Use `print()` to display the formatted report.

Examples

```
df <- data.frame(
  score  = c(23,45,12,67,34,89,56,43,78,90,11,34),
  method = rep(c("Online","Traditional"), each = 6),
  gender = rep(c("Male","Female"), times = 6)
)
result <- anova2_interpret(score ~ method * gender, data = df)
print(result)
```

check_assumptions *Check Statistical Assumptions Before Running a Test*

Description

Check Statistical Assumptions Before Running a Test

Usage

```
check_assumptions(test, x = NULL, y = NULL, data = NULL, formula = NULL)
```

Arguments

test	The test you plan to run. One of "ttest", "anova", "anova2", "correlation", "regression".
x	A numeric vector (required for most tests)
y	A numeric vector, factor, or character group variable (optional depending on test)
data	A data frame (required for anova, anova2, regression)
formula	A formula (required for anova, anova2, regression)

Value

An object of class `statease_assumptions` containing assumption check results. Use `print()` to display the formatted report.

Examples

```
x <- c(23, 45, 12, 67, 34, 89, 56, 43, 78, 90)
y <- c(19, 38, 22, 51, 29, 74, 44, 38, 65, 80)
result <- check_assumptions("ttest", x = x, y = y)
print(result)
```

chisq_interpret *Chi-Square Test with Plain-English Interpretation*

Description

Chi-Square Test with Plain-English Interpretation

Usage

```
chisq_interpret(x, y, correct = TRUE, conf.level = 0.95)
```

Arguments

x	A factor or character vector (first categorical variable)
y	A factor or character vector (second categorical variable)
correct	Logical. Apply Yates continuity correction. Default TRUE.
conf.level	Confidence level. Default 0.95.

Value

An object of class statease_chisq containing test results and interpretation. Use print() to display the formatted report.

Examples

```
x <- c("Yes","No","Yes","Yes","No","Yes","No","No","Yes","Yes")
y <- c("Male","Female","Male","Female","Male","Female","Male","Female","Male","Female")
result <- chisq_interpret(x, y)
print(result)
```

---

cor_interpret *Correlation Analysis with Plain-English Interpretation*

Description

Correlation Analysis with Plain-English Interpretation

Usage

```
cor_interpret(
  x,
  y,
  method = "pearson",
  conf.level = 0.95,
  var1_name = "Variable 1",
  var2_name = "Variable 2"
)
```

Arguments

x	A numeric vector
y	A numeric vector
method	Correlation method: "pearson", "spearman", or "kendall". Default "pearson".
conf.level	Confidence level. Default 0.95.
var1_name	Optional name for first variable. Default "Variable 1"
var2_name	Optional name for second variable. Default "Variable 2"

Value

An object of class `statease_cor` containing correlation results and interpretation. Use `print()` to display the formatted report.

Examples

```
x <- c(23, 45, 12, 67, 34, 89, 56, 43, 78, 90)
y <- c(19, 42, 15, 70, 30, 85, 52, 48, 80, 88)
result <- cor_interpret(x, y)
print(result)
```

`describe` *Descriptive Statistics with Interpretation*

Description

Descriptive Statistics with Interpretation

Usage

```
describe(x, var_name = "Variable")
```

Arguments

`x`	A numeric vector
`var_name`	Optional name for the variable (used in the report)

Value

An object of class `statease_describe` containing descriptive statistics and interpretation. Use `print()` to display the formatted report.

Examples

```
result <- describe(c(23, 45, 12, 67, 34, 89, 56))
print(result)
```

`fisher_interpret` *Fisher's Exact Test with Plain-English Interpretation*

Description

Fisher's Exact Test with Plain-English Interpretation

Usage

```
fisher_interpret(x, y, conf.level = 0.95, simulate.p.value = FALSE)
```

Arguments

`x`	A factor or character vector (first categorical variable)
`y`	A factor or character vector (second categorical variable)
`conf.level`	Confidence level. Default 0.95.
`simulate.p.value`	Logical. Whether to use simulation to compute p-values for larger tables. Default FALSE.

Value

An object of class `statease_fisher` containing test results and interpretation. Use `print()` to display the formatted report.

Examples

```
x <- c("Yes","No","Yes","Yes","No","Yes","No","No","Yes","Yes")
y <- c("Male","Female","Male","Female","Male",
       "Female","Male","Female","Male","Female")
result <- fisher_interpret(x, y)
print(result)
```

`friedman_interpret` *Friedman Test with Plain-English Interpretation*

Description

Friedman Test with Plain-English Interpretation

Usage

```
friedman_interpret(formula, data, conf.level = 0.95)
```

Arguments

formula	A formula of the form outcome ~ time | subject
data	A data frame containing the variables
conf.level	Confidence level. Default 0.95.

Value

An object of class statease_friedman containing test results and interpretation. Use print() to display the formatted report.

Examples

```
df <- data.frame(
  score   = c(23,45,12,67,34,89,56,43,78,90,11,34),
  time    = rep(c("T1","T2","T3"), each = 4),
  subject = rep(1:4, times = 3)
)
result <- friedman_interpret(score ~ time | subject, data = df)
print(result)
```

---

interpret_p *Standalone P-Value Interpreter*

Description

Standalone P-Value Interpreter

Usage

```
interpret_p(p, alpha = 0.05, context = NULL)
```

Arguments

p	A numeric p-value between 0 and 1
alpha	Significance level. Default 0.05
context	Optional string describing the test context

Value

An object of class statease_pvalue containing the p-value interpretation. Use print() to display the report.

Examples

```
result <- interpret_p(0.03)
print(result)

result2 <- interpret_p(0.12, alpha = 0.05, context = "treatment vs control")
print(result2)
```

kruskal_interpret *Kruskal-Wallis Test with Plain-English Interpretation*

Description

Kruskal-Wallis Test with Plain-English Interpretation

Usage

```
kruskal_interpret(formula, data, conf.level = 0.95)
```

Arguments

formula	A formula of the form outcome ~ group
data	A data frame containing the variables
conf.level	Confidence level. Default 0.95.

Value

An object of class `statease_kruskal` containing test results and interpretation. Use `print()` to display the formatted report.

Examples

```
df <- data.frame(
  score = c(23,45,12,67,34,89,56,43,78,90,11,34),
  group = rep(c("A","B","C"), each = 4)
)
result <- kruskal_interpret(score ~ group, data = df)
print(result)
```

logistic_interpret *Logistic Regression with Plain-English Interpretation*

Description

Logistic Regression with Plain-English Interpretation

Usage

```
logistic_interpret(formula, data, conf.level = 0.95)
```

Arguments

formula	A formula of the form outcome ~ predictor1 + predictor2 + ...
data	A data frame containing the variables
conf.level	Confidence level. Default 0.95.

Value

An object of class `statease_logistic` containing logistic regression results and interpretation. Use `print()` to display the formatted report.

Examples

```
df <- data.frame(
  passed      = c(1,1,0,1,0,1,1,0,1,1,0,0,1,1,0),
  study_hours = c(9,8,3,7,2,9,8,3,7,6,2,1,8,7,3),
  attendance  = c(90,85,50,80,45,95,88,55,78,70,40,35,92,83,52)
)
result <- logistic_interpret(passed ~ study_hours + attendance, data = df)
print(result)
```

---

mannwhitney_interpret *Mann-Whitney U Test with Plain-English Interpretation*

Description

Mann-Whitney U Test with Plain-English Interpretation

Usage

```
mannwhitney_interpret(x, y, conf.level = 0.95, var_name = "Variable")
```

Arguments

`x`	A numeric vector (group 1)
`y`	A numeric vector (group 2)
`conf.level`	Confidence level. Default 0.95.
`var_name`	Optional label for the report. Default "Variable"

Value

An object of class `statease_mannwhitney` containing test results and interpretation. Use `print()` to display the formatted report.

Examples

```
x <- c(23, 45, 12, 67, 34, 89, 56)
y <- c(19, 38, 22, 51, 29, 74, 44)
result <- mannwhitney_interpret(x, y)
print(result)
```

`manova_interpret` *MANOVA with Plain-English Interpretation*

Description

MANOVA with Plain-English Interpretation

Usage

```
manova_interpret(formula, data, conf.level = 0.95)
```

Arguments

`formula`	A formula of the form cbind(outcome1, outcome2, ...) ~ group
`data`	A data frame containing the variables
`conf.level`	Confidence level. Default 0.95.

Value

An object of class `statease_manova` containing MANOVA results and interpretation. Use `print()` to display the formatted report.

Examples

```
df <- data.frame(
  math    = c(23,45,12,67,34,89,56,43,78,90,11,34),
  english = c(34,56,23,78,45,90,67,54,89,95,22,45),
  group   = rep(c("A","B","C"), each = 4)
)
result <- manova_interpret(cbind(math, english) ~ group, data = df)
print(result)
```

`mcnemar_interpret` *McNemar's Test with Plain-English Interpretation*

Description

McNemar's Test with Plain-English Interpretation

Usage

```
mcnemar_interpret(x, y, conf.level = 0.95)
```

Arguments

x	A factor or character vector (first measurement)
y	A factor or character vector (second measurement)
conf.level	Confidence level. Default 0.95.

Value

An object of class statease_mcnemar containing test results and interpretation. Use print() to display the formatted report.

Examples

```
x <- c("Yes","No","Yes","Yes","No","Yes","No","No","Yes","Yes")
y <- c("No","No","Yes","Yes","No","Yes","Yes","No","Yes","No")
result <- mcnemar_interpret(x, y)
print(result)
```

---

mlr_interpret *Multiple Linear Regression with Plain-English Interpretation*

Description

Multiple Linear Regression with Plain-English Interpretation

Usage

```
mlr_interpret(formula, data, conf.level = 0.95)
```

Arguments

formula	A formula of the form outcome ~ predictor1 + predictor2 + ...
data	A data frame containing the variables
conf.level	Confidence level. Default 0.95.

Value

An object of class statease_mlr containing multiple regression results and interpretation. Use print() to display the formatted report.

Examples

```
df <- data.frame(
  exam_score  = c(23,45,12,67,34,89,56,43,78,90),
  study_hours = c(2,5,1,7,3,9,6,4,8,10),
  attendance  = c(60,80,50,90,70,95,85,75,88,92)
)
result <- mlr_interpret(exam_score ~ study_hours + attendance, data = df)
print(result)
```

power_interpret *Power Analysis with Plain English Interpretation*

Description

Power Analysis with Plain English Interpretation

Usage

```
power_interpret(
  test,
  effect_size,
  n = NULL,
  alpha = 0.05,
  power = 0.8,
  n_groups = 2,
  n_predictors = 1
)
```

Arguments

test	The statistical test. One of "ttest.one", "ttest.two", "ttest.paired", "anova", "correlation", "chisq", "regression".
effect_size	The expected effect size. Use Cohen's conventions: small = 0.2, medium = 0.5, large = 0.8 for t-tests; small = 0.10, medium = 0.25, large = 0.40 for ANOVA; small = 0.10, medium = 0.30, large = 0.50 for correlation.
n	Sample size per group. If provided, calculates achieved power. If NULL, calculates required sample size.
alpha	Significance level. Default 0.05.
power	Desired power level. Default 0.80.
n_groups	Number of groups (for ANOVA only). Default 2.
n_predictors	Number of predictors (for regression only). Default 1.

Value

An object of class `statease_power` containing power analysis results and interpretation. Use `print()` to display the formatted report.

Examples

```
# Calculate required sample size for independent t-test
result <- power_interpret("ttest.two", effect_size = 0.5)
print(result)

# Calculate achieved power for given sample size
result2 <- power_interpret("ttest.two", effect_size = 0.5, n = 30)
print(result2)
```

reg_interpret *Simple Linear Regression with Plain-English Interpretation*

Description

Simple Linear Regression with Plain-English Interpretation

Usage

```
reg_interpret(formula, data, conf.level = 0.95)
```

Arguments

formula	A formula of the form outcome ~ predictor
data	A data frame containing the variables
conf.level	Confidence level. Default 0.95.

Value

An object of class `statease_reg` containing regression results and interpretation. Use `print()` to display the formatted report.

Examples

```
df <- data.frame(
  exam_score = c(23,45,12,67,34,89,56,43,78,90),
  study_hours = c(2,5,1,7,3,9,6,4,8,10)
)
result <- reg_interpret(exam_score ~ study_hours, data = df)
print(result)
```

run_app *Launch the statease Shiny App*

Description

Launches an interactive Shiny application for running statistical analyses without writing any code.

Usage

```
run_app(...)
```

Arguments

...	Additional arguments passed to shiny::runApp()

Value

Launches the Shiny app in your browser

Examples

```
if(interactive()){
  run_app()
}
```

`statease` *statease: Simplified Statistical Analysis with Plain-English Interpretation*

Description

statease provides a suite of functions for performing common statistical analyses and automatically interpreting the results in plain English. It is designed for students, researchers, and educators who want fast, readable statistical output without sacrificing rigour.

Main Functions

`analyze` Master function — auto-detects and runs the right test

`describe` Descriptive statistics with interpretation

`ttest_interpret` T-tests (one-sample, independent, paired) with Cohen's d

`anova_interpret` One-way ANOVA with Tukey post-hoc and eta squared

`interpret_p` Standalone p-value interpreter

Typical Workflow

The simplest way to use statease is through the master `analyze()` function, which automatically detects what test to run based on your input:

```
# Descriptive statistics
analyze(x = my_vector, var_name = "My Variable")

# Independent samples t-test
analyze(x = group1, y = group2, var_name = "Scores")

# One-way ANOVA
analyze(formula = score ~ group, data = my_df)

# Interpret a p-value
interpret_p(0.03, context = "treatment vs control")
```

Author(s)

Uwakmfon Usen Paul

See Also

Useful links:

- https://github.com/DevWebWacky/statease
- https://devwebwacky.github.io/statease/
- Report bugs at https://github.com/DevWebWacky/statease/issues

---

`ttest_interpret` *T-Test with Plain-English Interpretation*

Description

T-Test with Plain-English Interpretation

Usage

```
ttest_interpret(
  x,
  y = NULL,
  mu = 0,
  paired = FALSE,
  conf.level = 0.95,
  var_name = "Variable"
)
```

Arguments

`x`	A numeric vector (group 1, or the only group for one-sample)
`y`	A numeric vector (group 2, for independent samples). Default NULL.
`mu`	Hypothesised mean for one-sample t-test. Default 0.
`paired`	Logical. TRUE for paired t-test. Default FALSE.
`conf.level`	Confidence level. Default 0.95.
`var_name`	Optional label for the report. Default "Variable"

Value

An object of class `statease_ttest` containing test results and interpretation. Use `print()` to display the formatted report.

Examples

```
result <- ttest_interpret(c(23,45,12,67,34), c(19,38,22,51,29))
print(result)
```

`wilcoxon_interpret` *Wilcoxon Signed Rank Test with Plain-English Interpretation*

Description

Wilcoxon Signed Rank Test with Plain-English Interpretation

Usage

```
wilcoxon_interpret(x, y, conf.level = 0.95, var_name = "Variable")
```

Arguments

`x`	A numeric vector (first measurement)
`y`	A numeric vector (second measurement)
`conf.level`	Confidence level. Default 0.95.
`var_name`	Optional label for the report. Default "Variable"

Value

An object of class `statease_wilcoxon` containing test results and interpretation. Use `print()` to display the formatted report.

Examples

```
x <- c(23, 45, 12, 67, 34, 89, 56)
y <- c(19, 38, 22, 51, 29, 74, 44)
result <- wilcoxon_interpret(x, y)
print(result)
```

Index