Package: EasyStat (via r-universe)

May 25, 2026

Type Package

Title Automated Statistical Analysis, Visualization and Multi-Format Narrative Reporting

Version 2.0.0

Date 2026-04-28

Description Provides automated statistical analysis, rich visualization, and multi-format narrative reporting through a unified pipeline. Descriptive statistics are available via easy_describe() and easy_group_summary(). Inferential tests with plain-language narratives are provided by easy_regression(), easy_logistic_regression(), easy_ttest(), easy_anova(), easy_chisq(), easy_ztest(), easy_ftest(), easy_correlation(), easy_wilcox(), and easy_kruskal(). Publication-ready 'ggplot2' visualizations are produced by easy_histogram(), easy_boxplot(), easy_scatter(), easy_barplot(), easy_qqplot(), easy_density(), easy_correlation_heatmap(), easy_regression_diagnostics(), and easy_odds_ratio_plot(). The core Narrative Generator Module applies conditional logic to extracted p-values, effect sizes, and model-fit metrics to produce statistically sound, human-readable explanations automatically. Results render in the 'RStudio' Viewer (HTML), the console (ASCII), or export directly to Microsoft Word via 'flextable' and 'officer'.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.1.0)

RoxygenNote 7.3.3

Imports broom (>= 1.0.0), glue (>= 1.6.0), ggplot2 (>= 3.4.0), knitr (>= 1.40), kableExtra (>= 1.3.4), htmltools (>= 0.5.0), flextable (>= 0.9.0), officer (>= 0.6.0), stats, utils, tools

Suggests testthat (>= 3.0.0), scales (>= 1.2.0), rmarkdown (>= 2.0)

VignetteBuilder knitr

URL https://github.com/itsmdivakaran/Easystat,
https://itsmdivakaran.github.io/Easystat/

BugReports https://github.com/itsmdivakaran/Easystat/issues

Config/pak/sysreqs
libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev

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

Date/Publication 2026-04-30 08:26:18 UTC

RemoteUrl https://github.com/itsmdivakaran/easystat

RemoteRef HEAD

RemoteSha 3cc2cb2b3973d3ea36585d7d82ef42604a27604d

Contents

EasyStat-package *EasyStat: Automated Statistical Analysis, Visualization and Multi-Format Narrative Reporting*

Description

EasyStat provides a unified pipeline for automated statistical analysis and plain-language narrative reporting in R. The four-step pipeline — Core Statistical Engine, Metric Extractor (broom), Narrative Generator Module, and Unified Result Object — transforms raw data into publication-ready output with a single function call.

Descriptive statistics: easy_describe, easy_group_summary

Inferential tests: easy_regression, easy_ttest, easy_anova, easy_chisq, easy_ztest, easy_ftest, easy_correlation

Visualizations: easy_histogram, easy_boxplot, easy_scatter, easy_barplot, easy_qqplot, easy_density, easy_correlation_heatmap, easy_autoplot

Reporting: export_to_word, theme_easystat

Authors

Mahesh Divakaran (Amity School of Applied Sciences, Amity University Lucknow), Gunjan Singh (Amity School of Applied Sciences, Amity University Lucknow), Jayadevan Shreedharan (Gulf Medical University)

Author(s)

Maintainer: Mahesh Divakaran <imaheshdivakaran@gmail.com> (Amity School of Applied Sciences, Amity University Lucknow)

Authors:

- Gunjan Singh (Amity School of Applied Sciences, Amity University Lucknow)
- Jayadevan Shreedharan (Gulf Medical University)

See Also

Useful links:

- https://github.com/itsmdivakaran/Easystat
- https://itsmdivakaran.github.io/Easystat/
- Report bugs at https://github.com/itsmdivakaran/Easystat/issues

easy_anova *Run a One-Way ANOVA with Automated Narrative Reporting*

Description

Executes a one-way ANOVA using `stats::aov()`, extracts key metrics via `broom`, computes eta-squared as an effect-size measure, and generates a plain-language narrative via the Narrative Generator Module.

Usage

```
easy_anova(formula, data, alpha = 0.05)
```

Arguments

`formula`	A `formula` of the form `outcome ~ group_factor`, or a character string. Passed directly to `stats::aov()`.
`data`	A data frame containing the variables in `formula`.
`alpha`	Significance threshold for the narrative. Default `0.05`.

Value

An object of class `"easystat_result"` with:

`test_type` Character: `"anova"`

`formula_str` Character string of the formula used

`raw_model` The raw `aov` object

`coefficients_table` ANOVA table (SS, df, MS, F, p)

`model_fit_table` Summary metrics (F-statistic, eta-squared, p-value)

`explanation` Plain-language narrative string

Examples

```
result <- easy_anova(Sepal.Length ~ Species, data = iris)
print(result)
```

easy_autoplot *Automatically Plot an EasyStat Result*

Description

Chooses the most appropriate plot type based on the `test_type` of an `easystat_result` object and renders it.

Usage

```
easy_autoplot(result, data = NULL, ...)
```

Arguments

`result`	An `"easystat_result"` object.
`data`	The original data frame (required for some plot types).
`...`	Additional arguments passed to the underlying plot function.

Value

An `"easystat_result"` plot object, invisibly.

easy_barplot *Annotated Bar Chart*

Description

Creates a frequency bar chart for categorical variables, or a mean-and-error bar chart for numeric outcomes grouped by a factor.

Usage

```
easy_barplot(
  x,
  data,
  group_by = NULL,
  stat = c("count", "mean"),
  fill_palette = NULL,
  title = NULL
)
```

Arguments

x	Column name of the variable to plot.
data	A data frame.
group_by	Optional grouping column for grouped frequency bars.
stat	"count" (default) for frequency, or "mean" for mean ± SE bars.
fill_palette	Color palette vector. Default EasyStat palette.
title	Custom plot title.

Value

An "easystat_result" object with plot_object.

---

easy_boxplot *Grouped Boxplot with Outlier Detection*

Description

Produces a boxplot for one variable, optionally grouped by a factor. Adds a jittered dot overlay, labels each group's median, and highlights outliers.

Usage

```
easy_boxplot(formula, data, fill_palette = NULL, notch = FALSE, title = NULL)
```

Arguments

formula	A formula: outcome ~ group for grouped, or a bare column name / numeric vector for a single-variable boxplot.
data	A data frame.
fill_palette	Character vector of fill colors. Default EasyStat palette.
notch	Logical; draw notched boxes? Default FALSE.
title	Custom plot title.

Value

An "easystat_result" object with plot_object.

`easy_chisq` *Chi-Square Tests with Automated Narrative Reporting*

Description

Runs either a **chi-square test of independence** (two categorical variables) or a **goodness-of-fit test** (one variable vs. expected proportions), extracts Cramér's V as the effect-size measure, and generates a plain-language narrative via the Narrative Generator Module.

Usage

```
easy_chisq(x, y = NULL, data = NULL, p = NULL, correct = TRUE, alpha = 0.05)
```

Arguments

`x`	A factor/character vector, OR a contingency table (matrix), OR a formula `~ var1 + var2` for independence, or `~ var1` for GOF.
`y`	A factor/character vector (second categorical variable) for the independence test. Ignored when `x` is a table or formula.
`data`	A data frame. Required when `x` is a formula.
`p`	Numeric vector of expected probabilities for the GOF test. If `NULL` (default), equal probabilities are assumed.
`correct`	Logical; apply Yates' continuity correction? Default `TRUE`.
`alpha`	Significance threshold for narrative. Default `0.05`.

Value

An `"easystat_result"` object with:

`coefficients_table` Observed vs. expected frequency table

`model_fit_table` Chi-square statistic, df, p-value, Cramér's V

`explanation` Plain-language narrative

Examples

```
# Independence test
result <- easy_chisq(~ cyl + am, data = mtcars)
print(result)

# Goodness-of-fit
result <- easy_chisq(~ cyl, data = mtcars)
print(result)
```

easy_correlation *Correlation Analysis with Automated Narrative Reporting*

Description

Computes bivariate or pairwise correlations (Pearson, Spearman, or Kendall) with significance tests and confidence intervals. For two variables a full narrative is generated; for multiple variables a correlation matrix is returned with a summary digest.

Usage

```
easy_correlation(
  x,
  y = NULL,
  data = NULL,
  vars = NULL,
  method = "pearson",
  conf_level = 0.95,
  alpha = 0.05
)
```

Arguments

x	A numeric vector, a data frame, OR a formula ~ x + y.
y	A numeric vector (paired with x). Ignored when x is a formula or data frame.
data	A data frame. Required when x is a formula.
vars	Character vector of column names when x is a data frame and pairwise analysis is desired. Default NULL = all numeric cols.
method	Correlation method: "pearson" (default), "spearman", or "kendall".
conf_level	Confidence level for Pearson CI. Default 0.95.
alpha	Significance threshold for narrative. Default 0.05.

Value

An "easystat_result" object.

Examples

```
result <- easy_correlation(~ mpg + wt, data = mtcars)
print(result)

result <- easy_correlation(mtcars, vars = c("mpg", "hp", "wt", "disp"))
print(result)
```

easy_correlation_heatmap *Correlation Matrix Heatmap*

Description

Computes pairwise correlations and displays them as a color-coded heatmap, annotating each cell with the correlation coefficient and a significance star.

Usage

```
easy_correlation_heatmap(data, vars = NULL, method = "pearson", title = NULL)
```

Arguments

data	A data frame.
vars	Character vector of numeric column names. Default all numerics.
method	Correlation method: "pearson" (default), "spearman".
title	Custom plot title.

Value

An "easystat_result" object with plot_object.

easy_density *Kernel Density Plot with Optional Group Overlay*

Description

Draws a smooth kernel density estimate for a numeric variable. If a grouping variable is provided, separate overlapping density curves are drawn per group.

Usage

```
easy_density(x, data = NULL, group_by = NULL, fill_alpha = 0.35, title = NULL)
```

Arguments

x	Column name or numeric vector.
data	A data frame.
group_by	Optional grouping column for multi-group densities.
fill_alpha	Alpha for filled area. Default 0.35.
title	Custom plot title.

Value

An "easystat_result" object with plot_object.

---

easy_describe *Comprehensive Descriptive Statistics with Narrative*

Description

Computes a rich set of descriptive statistics for one or more numeric variables, including measures of central tendency, dispersion, shape (skewness, kurtosis), and normality (Shapiro-Wilk), together with an automatic plain-language narrative interpretation.

Usage

```
easy_describe(data, vars = NULL, digits = 4, conf_level = 0.95)
```

Arguments

data	A numeric vector **or** a data frame.
vars	Character vector of column names to describe when data is a data frame. If NULL (default), all numeric columns are used.
digits	Number of decimal places in the summary table. Default 4.
conf_level	Confidence level for the mean CI. Default 0.95.

Value

An "easystat_result" object with:

coefficients_table Wide-format summary statistics table

model_fit_table Shape and normality digest

explanation Plain-language narrative (one per variable)

Examples

```
result <- easy_describe(mtcars, vars = c("mpg", "hp", "wt"))
print(result)
```

`easy_ftest` *F-Test for Equality of Variances with Automated Narrative Reporting*

Description

Performs an F-test to compare the variances of two independent groups using `stats::var.test()`, extracts the F-statistic, degrees of freedom, p-value, variance ratio, and confidence interval, and generates a plain-language narrative that includes a practical recommendation for downstream t-test selection (equal vs. unequal variances).

Usage

```
easy_ftest(
  x,
  y = NULL,
  data = NULL,
  ratio = 1,
  alternative = "two.sided",
  conf_level = 0.95,
  alpha = 0.05
)
```

Arguments

`x`	A numeric vector (Group 1), OR a formula `outcome ~ group`.
`y`	A numeric vector (Group 2). Ignored when `x` is a formula.
`data`	A data frame. Required when `x` is a formula.
`ratio`	Hypothesized ratio of variances under H0. Default `1`.
`alternative`	`"two.sided"` (default), `"less"`, or `"greater"`.
`conf_level`	Confidence level for the variance ratio CI. Default `0.95`.
`alpha`	Significance threshold for narrative. Default `0.05`.

Value

An `"easystat_result"` object.

Examples

```
result <- easy_ftest(mpg ~ am, data = mtcars)
print(result)
```

easy_group_summary *Group-Wise Summary Statistics with Narrative*

Description

Computes descriptive statistics for a numeric outcome variable stratified by a grouping factor, providing both a comparison table and a narrative highlighting which group has the highest/lowest mean and variability.

Usage

```
easy_group_summary(formula, data, digits = 4)
```

Arguments

formula	A formula of the form outcome ~ group.
data	A data frame containing the variables.
digits	Number of decimal places. Default 4.

Value

An "easystat_result" object.

Examples

```
result <- easy_group_summary(mpg ~ cyl, data = mtcars)
print(result)
```

easy_histogram *Annotated Histogram with Normal Curve Overlay*

Description

Draws a histogram of a numeric variable, overlays a fitted normal density curve, and annotates the plot with the mean, median, and standard deviation. Normality is assessed via the Shapiro-Wilk test, and the result is displayed in the subtitle.

Usage

```
easy_histogram(
  x,
  data = NULL,
  bins = NULL,
  fill_color = NULL,
  show_normal = TRUE,
  title = NULL
)
```

Arguments

x	Character column name OR a numeric vector.
data	A data frame (required when x is a column name).
bins	Number of histogram bins. Default NULL (auto).
fill_color	Bar fill color. Default EasyStat primary blue.
show_normal	Logical; overlay normal curve? Default TRUE.
title	Custom plot title. Default auto-generated.

Value

An "easystat_result" object with plot_object.

---

easy_kruskal *Kruskal-Wallis Test with Automated Narrative Reporting*

Description

Runs a Kruskal-Wallis rank-sum test for comparing three or more groups.

Usage

```
easy_kruskal(formula, data, alpha = 0.05)
```

Arguments

formula	A formula of the form outcome ~ group.
data	A data frame.
alpha	Significance threshold for narrative. Default 0.05.

Value

An "easystat_result" object.

Examples

```
result <- easy_kruskal(Sepal.Length ~ Species, data = iris)
print(result)
```

easy_logistic_regression *Run a Logistic Regression with Automated Narrative Reporting*

Description

Executes a binary logistic regression using `stats::glm()` with `family = binomial`, extracts coefficients, odds ratios, approximate confidence intervals, and model-fit metrics, then generates a plain-language narrative via the Narrative Generator Module.

Usage

```
easy_logistic_regression(formula, data, alpha = 0.05, conf_level = 0.95)
```

Arguments

`formula`	A `formula` object or a character string formula (e.g., `"am ~ mpg + wt"`). The outcome must be binary.
`data`	A data frame containing the variables referenced in `formula`.
`alpha`	Significance threshold used in narrative generation. Default `0.05`.
`conf_level`	Confidence level for odds-ratio intervals. Default `0.95`.

Value

An object of class `"easystat_result"` with coefficient, odds-ratio, model-fit, raw `glm`, and narrative components.

Examples

```
result <- easy_logistic_regression(am ~ mpg + wt, data = mtcars)
print(result)
```

easy_odds_ratio_plot *Odds Ratio Plot for Logistic Regression*

Description

Creates a coefficient figure showing odds ratios and confidence intervals for `easy_logistic_regression` results.

Usage

```
easy_odds_ratio_plot(result)
```

Arguments

result	An "easystat_result" from easy_logistic_regression().

Value

An "easystat_result" object with plot_object.

---

easy_qqplot *Normal Q-Q Plot with Shapiro-Wilk Annotation*

Description

Plots sample quantiles against theoretical normal quantiles and annotates the Shapiro-Wilk p-value. Deviations from the diagonal indicate non-normality.

Usage

```
easy_qqplot(x, data = NULL, title = NULL)
```

Arguments

x	Column name or numeric vector.
data	A data frame (required when x is a column name).
title	Custom plot title.

Value

An "easystat_result" object with plot_object.

---

easy_regression *Run a Linear Regression with Automated Narrative Reporting*

Description

Executes a standard OLS linear regression using stats::lm(), extracts key metrics via the broom package, and automatically generates a plain-language narrative explanation via the Narrative Generator Module.

Usage

```
easy_regression(formula, data, alpha = 0.05)
```

Arguments

formula	A formula object or a character string formula (e.g., "mpg ~ wt + hp"). Passed directly to stats::lm().
data	A data frame containing the variables referenced in formula.
alpha	Significance threshold used in narrative generation. Default 0.05.

Value

An object of class "easystat_result" (an R list) with:

test_type Character: "regression"

formula_str Character string of the formula used

raw_model The raw lm object for advanced use

coefficients_table Tidy data frame of coefficients, SEs, t-stats, p-values

model_fit_table Data frame with R^2, Adjusted R^2, F-statistic, p-value

explanation Plain-language narrative string

Examples

```
result <- easy_regression(mpg ~ wt + hp, data = mtcars)
print(result)
```

---

easy_regression_diagnostics

Regression Diagnostic Plot

Description

Creates a fitted-vs-residuals diagnostic figure for linear regression results returned by easy_regression.

Usage

```
easy_regression_diagnostics(result)
```

Arguments

result	An "easystat_result" from easy_regression().

Value

An "easystat_result" object with plot_object.

`easy_scatter` *Scatter Plot with Regression Line and Correlation Annotation*

Description

Draws a scatter plot for two numeric variables, overlays a linear regression line with confidence band, and annotates the Pearson r and p-value.

Usage

```
easy_scatter(
  formula,
  data,
  color_by = NULL,
  smooth = TRUE,
  ellipse = TRUE,
  title = NULL
)
```

Arguments

`formula`	A formula: `y ~ x`.
`data`	A data frame.
`color_by`	Optional column name to color points by a third variable.
`smooth`	Logical; show regression line? Default `TRUE`.
`ellipse`	Logical; draw a 95% data ellipse? Default `TRUE`.
`title`	Custom plot title.

Value

An `"easystat_result"` object with `plot_object`.

`easy_ttest` *Run an Independent-Samples t-Test with Automated Narrative Reporting*

Description

Executes a two-sample (or one-sample) t-test using `stats::t.test()`, extracts key metrics via `broom`, and generates a plain-language narrative via the Narrative Generator Module.

Usage

```
easy_ttest(
  x,
  y = NULL,
  data = NULL,
  mu = 0,
  var.equal = FALSE,
  conf.level = 0.95,
  alpha = 0.05
)
```

Arguments

x	A numeric vector, OR a formula of the form outcome ~ group when data is provided.
y	A numeric vector (second group) when x is not a formula. Ignored when x is a formula.
data	A data frame. Required when x is a formula.
mu	Null hypothesis value for the mean (one-sample test). Default 0.
var.equal	Logical; assume equal variances? Default FALSE (Welch).
conf.level	Confidence level. Default 0.95.
alpha	Significance threshold for narrative. Default 0.05.

Value

An object of class "easystat_result" with:

test_type Character: "ttest"

formula_str Description of the comparison

raw_model The raw htest object

coefficients_table Group means and confidence interval

model_fit_table t-statistic, df, and p-value

explanation Plain-language narrative string

Examples

```
result <- easy_ttest(mpg ~ am, data = mtcars)
print(result)
```

easy_wilcox *Wilcoxon Tests with Automated Narrative Reporting*

Description

Runs a one-sample, paired, or two-sample Wilcoxon test using `stats::wilcox.test()` and returns an `"easystat_result"` object.

Usage

```
easy_wilcox(
  x,
  y = NULL,
  data = NULL,
  mu = 0,
  paired = FALSE,
  alternative = "two.sided",
  conf_level = 0.95,
  alpha = 0.05
)
```

Arguments

`x`	Numeric vector, or a formula of the form `outcome ~ group`.
`y`	Optional numeric vector for paired or two-sample tests.
`data`	Data frame used when `x` is a formula.
`mu`	Null hypothesized location or location shift. Default `0`.
`paired`	Logical. Use paired test? Default `FALSE`.
`alternative`	`"two.sided"`, `"less"`, or `"greater"`.
`conf_level`	Confidence level. Default `0.95`.
`alpha`	Significance threshold for narrative. Default `0.05`.

Value

An `"easystat_result"` object.

Examples

```
result <- easy_wilcox(mpg ~ am, data = mtcars)
print(result)
```

`easy_ztest` *One-Sample and Two-Sample Z-Tests with Automated Narrative Reporting*

Description

Performs a z-test using the normal distribution. When the population standard deviation (`sigma`) is not provided, the sample SD is used (valid for large samples, $n \geq 30$, by the Central Limit Theorem). Key metrics — z-statistic, p-value, confidence interval, and Cohen's d — are extracted and fed to the Narrative Generator Module.

Usage

```
easy_ztest(
  x,
  y = NULL,
  data = NULL,
  mu = 0,
  sigma = NULL,
  sigma2 = NULL,
  alternative = "two.sided",
  conf_level = 0.95,
  alpha = 0.05
)
```

Arguments

`x`	A numeric vector (Group 1), OR a formula `outcome ~ group` for a two-sample test when `data` is provided.
`y`	A numeric vector (Group 2) for a two-sample test. Ignored when `x` is a formula.
`data`	A data frame. Required when `x` is a formula.
`mu`	Hypothesized population mean (one-sample) or mean difference (two-sample). Default `0`.
`sigma`	Known population SD for Group 1 (or the single group). If `NULL` (default), the sample SD is used.
`sigma2`	Known population SD for Group 2. If `NULL`, uses the sample SD of Group 2.
`alternative`	`"two.sided"` (default), `"less"`, or `"greater"`.
`conf_level`	Confidence level. Default `0.95`.
`alpha`	Significance threshold for narrative. Default `0.05`.

Value

An `"easystat_result"` object.

Examples

```
# One-sample z-test (large n, CLT)
result <- easy_ztest(mtcars$mpg, mu = 20)
print(result)

# Two-sample z-test via formula
result <- easy_ztest(mpg ~ am, data = mtcars)
print(result)
```

---

export_to_word *Export an EasyStat Result to a Formatted Microsoft Word Document*

Description

Takes a unified `easystat_result` object and writes a fully formatted `.docx` report using the `flextable` and `officer` packages. The report contains a title page header, the plain-language narrative, both statistical tables rendered as professional `flextable` objects, and a footer with metadata. All of this is produced in a single function call.

Usage

```
export_to_word(result, file, title = NULL, author = "EasyStat")
```

Arguments

`result`	An object of class `"easystat_result"` as returned by `easy_regression`, `easy_logistic_regression`, `easy_ttest`, or `easy_anova`.
`file`	Character string. Path to the output `.docx` file. Must be provided by the caller; for example, use `tempfile(fileext = ".docx")` for a temporary file.
`title`	Character string. Report title printed at the top of the document. If `NULL` (default) a title is auto-generated from the test type.
`author`	Character string. Author name(s) for the report header. Default `"EasyStat"`.

Value

The `file` path invisibly (a length-one character vector). The `.docx` file is written to the path supplied via `file` as a side-effect.

Examples

```
result <- easy_regression(mpg ~ wt + hp, data = mtcars)
export_to_word(result, file = tempfile(fileext = ".docx"),
               author = "Mr. Mahesh Divakaran")
```

print.easystat_result *Print an EasyStat Result Object*

Description

Automatically renders an `easystat_result` object. In an interactive RStudio session the HTML Viewer is used; otherwise clean ASCII tables and the plain-language narrative are written to the console.

Usage

```
## S3 method for class 'easystat_result'
print(x, viewer = NULL, ...)
```

Arguments

`x`	An object of class `"easystat_result"`.
`viewer`	Logical. Force HTML Viewer output (`TRUE`) or console output (`FALSE`). Default `NULL` auto-detects.
`...`	Currently ignored.

Value

`x` invisibly.

summary.easystat_result *Summarize an EasyStat Result Object*

Description

Summarize an EasyStat Result Object

Usage

```
## S3 method for class 'easystat_result'
summary(object, ...)
```

Arguments

`object`	An `"easystat_result"` object.
`...`	Passed to `print.easystat_result`.

Value

Called for its side effects (printing to the console or RStudio Viewer). Returns `object` invisibly via `print.easystat_result`.

`theme_easystat` *Apply EasyStat ggplot2 Theme*

Description

Adds a clean, professional EasyStat visual theme to any ggplot2 object.

Usage

```
theme_easystat(base_size = 12, legend_position = "right")
```

Arguments

`base_size`	Base font size. Default `12`.
`legend_position`	Where to place the legend. Default `"right"`.

Value

A `ggplot2::theme` object.

Index