Package 'ModLR'

May 29, 2026

Type Package

Title Information-Theoretic Approach for Moderation Analysis

Version 0.1.29

Description Provides a robust implementation of information-theoretic moderation analysis using multi-model inference based on Akaike's Information Criterion (AIC) and its small-sample corrected form (Corrected AIC).
The package enables researchers to compare competing model specifications and helps distinguish true interaction effects from nonlinear relationships that may produce spurious moderation. The methods build on Daryanto (2019) <doi:10.1016/j.jbusres.2019.06.012>.

License MIT + file LICENSE

Encoding UTF-8

Config/roxygen2/version 8.0.0

URL https://github.com/ahdar1/ModLR

Imports stats, ggplot2, broom, lmtest, sandwich, rlang

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Ahmad Daryanto [aut, cre]

Maintainer Ahmad Daryanto <ahdar_2000@yahoo.com>

Repository CRAN

Date/Publication 2026-05-29 10:10:08 UTC

Contents

compare_models *Compare Moderation Models*

Description

Compares candidate moderation models using information criteria (AIC/AICc).

Usage

```
compare_models(object, models = NULL, corrected = TRUE)
```

Arguments

object	A "modlr" object
models	Optional numeric vector specifying which models to compare. If NULL (default), all candidate models are evaluated.
corrected	Logical; whether to use AICc

Value

A data frame with model comparison results

Examples

```
set.seed(123)

n <- 100
x <- rnorm(n)
z <- 0.5 * x + sqrt(1 - 0.5^2) * rnorm(n)

y <- 0.3 * x + 0.3 * z + 0.8 * x * z + rnorm(n)

data <- data.frame(x, z, y)

result <- moderated_regression(data, iv = "x", moderator = "z", dv = "y")

compare_models(result)
compare_models(result, models = c(1, 2))
```

johnson_neyman *Johnson-Neyman Analysis*

Description

Computes regions of significance for an interaction effect.

Usage

```
johnson_neyman(object, alpha = 0.05, robust = NULL)
```

Arguments

object	A fitted model (modlr object)
alpha	Significance level
robust	Logical; use HC3 robust standard errors

Value

A data.frame of Johnson-Neyman results

Examples

```
set.seed(123)

n <- 100
x <- rnorm(n)
z <- 0.5 * x + sqrt(1 - 0.5^2) * rnorm(n)
y <- 0.3 * x + 0.3 * z + 0.8 * x * z + rnorm(n)

data <- data.frame(x, z, y)

result <- moderated_regression(data, iv = "x", moderator = "z", dv = "y")

johnson_neyman(result)
johnson_neyman(result, robust = TRUE)
```

moderated_regression *Moderated Regression Model*

Description

Fits a moderated regression model with optional extensions.

Usage

```
moderated_regression(
  data,
  iv,
  moderator,
  dv,
  covariates = NULL,
  center = TRUE,
  quadratic = FALSE,
  robust_se = FALSE
)
```

Arguments

data	A data frame
iv	Independent variable
moderator	Moderator variable
dv	Dependent variable
covariates	Optional character vector of covariate names. Defaults to NULL (no covariates).
center	Logical; whether to center variables
quadratic	Logical; include quadratic terms
robust_se	Logical; use HC3 robust standard errors

Value

A fitted model object

Examples

```
set.seed(123)

n <- 100
x <- rnorm(n)
w1 <- rnorm(n)
w2 <- rnorm(n)
z <- 0.5 * x + sqrt(1 - 0.5^2) * rnorm(n)

y <- 0.3 * x + 0.3 * z + 0.8 * x * z + rnorm(n)

data <- data.frame(w1, w2, x, z, y)

result <- moderated_regression(
 data,
 iv = "x",
 moderator = "z",
 dv = "y",
 covariates = c("w1", "w2")
 )
```

plot_moderation *Plot Moderation Effect*

Description

Produces a plot of the moderation effect.

Usage

```
plot_moderation(object)
```

Arguments

object	A fitted model

Value

A ggplot object

Examples

```
set.seed(123)

n <- 100
x <- rnorm(n)
z <- 0.5 * x + sqrt(1 - 0.5^2) * rnorm(n)

y <- 0.3 * x + 0.3 * z + 0.8 * x * z + rnorm(n)

data <- data.frame(x, z, y)

result <- moderated_regression(data, iv = "x", moderator = "z", dv = "y")

plot_moderation(result)
```

`simple_slopes` *Simple Slopes Analysis*

Description

Computes simple slopes for moderation analysis.

Usage

```
simple_slopes(object, values = NULL)
```

Arguments

`object`	A fitted model
`values`	Moderator values at which to compute slopes

Value

A data frame of slopes

Examples

```
set.seed(123)

n <- 100
x <- rnorm(n)
z <- 0.5 * x + sqrt(1 - 0.5^2) * rnorm(n)

y <- 0.3 * x + 0.3 * z + 0.8 * x * z + rnorm(n)

data <- data.frame(x, z, y)

result <- moderated_regression(data, iv = "x", moderator = "z", dv = "y")

simple_slopes(result)
```

Index