

Package: mod2rm (via r-universe)

October 15, 2024

Type Package

Title Moderation Analysis for Two-Instance Repeated Measures Designs

Description Multiple moderation analysis for two-instance repeated measures designs, with up to three simultaneous moderators (dichotomous and/or continuous) with additive or multiplicative relationship. Includes analyses of simple slopes and conditional effects at (automatically determined or manually set) values of the moderator(s), as well as an implementation of the Johnson-Neyman procedure for determining regions of significance in single moderator models. Based on Montoya, A. K. (2018) "Moderation analysis in two-instance repeated measures designs: Probing methods and multiple moderator models" <[doi:10.3758/s13428-018-1088-6](https://doi.org/10.3758/s13428-018-1088-6)> .

Version 0.2.1

Date 2022-06-29

Encoding UTF-8

RoxygenNote 7.2.0

License GPL (>= 2)

Imports stats, utils, methods, ggplot2, scales

NeedsCompilation no

Maintainer Matthias Forstmann <matthias.forstmann@uzh.ch>

Repository CRAN

Author Matthias Forstmann [aut, cre]

Date/Publication 2022-06-30 07:20:02 UTC

Contents

mod2rm	2
summary.mod2rm	4

Index	6
--------------	----------

Description

Multiple moderation analysis for two-instance repeated measures designs, including analyses of simple slopes and conditional effects at values of the moderator(s).

Currently supports both single- and multi-moderator models, with up to three simultaneous moderators (continuous and/or binary). Multi-moderator models support both additive (method = 1) and multiplicative (method = 2) moderation.

Also supports the Johnson-Neyman procedure for determining regions of significance in single moderator models (jn = T). Plots of the JN region can be obtained from the summary function (plotjn = T).

Moderator values at which to test for conditional effects are determined automatically (at -1, 0, and +1 SD of the mean if the moderator is continuous, and at both values of the moderator if it is binary), but any number of test values can also be set manually for each moderator.

Method and output based on: Montoya, A. K. (2018). Moderation analysis in two-instance repeated measures designs: Probing methods and multiple moderator models. *Behavior Research Methods*, 51(1), 61-82.

Usage

```
mod2rm(
  data,
  Y1,
  Y2,
  MOD1,
  MOD2 = NULL,
  MOD3 = NULL,
  MOD1val = NULL,
  MOD2val = NULL,
  MOD3val = NULL,
  method = 1,
  standardize = FALSE,
  jn = FALSE
)
```

Arguments

data	A data frame
Y1	Name of the first outcome variable

Y2	Name of the second outcome variable
MOD1	Name of moderator1 variable
MOD2	Name of moderator2 variable (optional)
MOD3	Name of moderator3 variable (optional)
MOD1val	A vector containing values of moderator1 at which to test for conditional effects (even when variables have been standardized!)(optional)
MOD2val	A vector containing values of moderator2 at which to test for conditional effects (even when variables have been standardized!)(optional)
MOD3val	A vector containing values of moderator3 at which to test for conditional effects (even when variables have been standardized!)(optional)
method	Method for dealing with two or more moderators (1 = additive, 2 = multiplicative) (default: additive)
standardize	boolean variable indicating whether all predictor variables (moderators) should be standardized prior to the analyses (default: FALSE)
jn	boolean variable indicating whether the Johnson-Neyman procedure should be calculated (only available for single moderator models)

Value

total	A list of class "mod2rm" containing:
info	A named number vector containing values for the number of moderators in the model (num_mods), the number of binary moderators (num_binary_mods), the sample size (sample_size), the method of moderation (method; 1 = additive, 2 = multiplicative), and whether the Johnson-Neyman procedure was run (jn)
var_names	A named character vector containing the name of the original dataframe (dataframe), the two outcome variables (y1,y2), and up to three moderators (mod1,mod2,mod3)
res_mod	A list including the results of a simple regression, regressing the difference between y1 and y2 on the moderator
res_simple_y1	A list including the results of a simple regression, regressing the y1 on the moderator
res_simple_y2	A list including the results of a simple regression, regressing the y2 on the moderator
res_cond_eff	A list including the results of an analysis of conditional effects at different levels of the moderator(s)
res_y1y2_diff	A list including the results of a repeated measures t-test for y1 and y2
res_jn_area	A list containing information on the Johnson-Neyman procedure, including the number of significance points identified within the data range (num_jn), the moderator values of these points, as well as the proportion of the sample scoring higher than these values (jn_values), and information on whether the JN region is significant or non-significant (center_significant; used for plotting.)
res_jn_cond_eff	A list containing additional conditional effects at levels of the moderator around the JN region. Values span the entire data range in 20 steps.

Examples

```
# Generate a dataset with a Johnson-Neyman (non-)significance region within the response range:

repeat{
  df = data.frame(out1 = runif(n = 100, min = 1, max = 9),
                 out2 = runif(n = 100, min = 1, max = 9),
                 w1 = runif(n = 100, min = 1, max = 9),
                 w2 = runif(n = 100, min = 1, max = 9),
                 w3 = runif(n = 100, min = 1, max = 9))
  res = mod2rm(df, out1, out2, w1, jn = TRUE)
  if(res$res_jn_area["num_jn"] == 2 & res$res_jn_area["center_significant"] == FALSE)
    break
}

# Show summary including plot
summary.mod2rm(res, plotjn = TRUE, plotstyle = "simple")

# Multiple regression (3 moderators, additive)
res1 = mod2rm(df, out1, out2, w1, w2, w3, method = 1)
summary.mod2rm(res1)

# Multiple regression (2 moderators, multiplicative, manually defined conditional effects)
res2 = mod2rm(df, out1, out2, w1, w2, MOD1val = c(2,3,4), MOD2val = c(4,5), method = 2)
summary.mod2rm(res2)
```

summary.mod2rm

Print and summary function for objects of class "mod2rm"

Description

Prints a summary of a list object of class "mod2rm", and (if requested) plot the results of the Johnson-Neyman procedure.

Usage

```
## S3 method for class 'mod2rm'
summary(object, ...)
```

Arguments

object	An object of class "mod2rm"
...	Additional parameters. "plotjn = TRUE" produces a ggplot for the Johnson-Neyman procedure, "plotstyle" can be set to "simple" or "points" (including data points in the plot)

Details

This function produces a summary for the results of an object of the type `mod2rm`, and can further be used to plot a graph showing the results of the JN procedure, if it is included in the `mod2rm` object. # Results include number and name(s) of moderator(s), sample size, results of a paired t-test between both dependent variables, the results of the moderation analysis, conditional effects of the moderator on each of the dependent variables, conditional effects at values of the moderator, and the results of the Johnson-Neyman procedure (including critical values, proportion of the sample above/below these values, and conditional effects around the significance regions).

Value

Prints summary of the object, then returns NULL or (when requested) a `ggplot2` object for the Johnson-Neyman plot

Examples

```
# Generate a dataset with a Johnson-Neyman (non-)significance region within the response range:
repeat{
  df = data.frame(out1 = runif(n = 100, min = 1, max = 9),
                 out2 = runif(n = 100, min = 1, max = 9),
                 w1 = runif(n = 100, min = 1, max = 9),
                 w2 = runif(n = 100, min = 1, max = 9),
                 w3 = runif(n = 100, min = 1, max = 9))
  res = mod2rm(df, out1, out2, w1, jn = TRUE)
  if(res$res_jn_area["num_jn"] == 2 & res$res_jn_area["center_significant"] == FALSE)
    break
}

# Show summary including plot
summary.mod2rm(res, plotjn = TRUE, plotstyle = "simple")

# Multiple regression (3 moderators, additive)
res1 = mod2rm(df, out1, out2, w1, w2, w3, method = 1)
summary.mod2rm(res1)

# Multiple regression (2 moderators, multiplicative, manually defined conditional effects)
res2 = mod2rm(df, out1, out2, w1, w2, MOD1val = c(2,3,4), MOD2val = c(4,5), method = 2)
summary.mod2rm(res2)
```

Index

`mod2rm`, [2](#)

`summary.mod2rm`, [4](#)