Package 'unusualprofile'

February 15, 2024

Type Package

Title Calculates Conditional Mahalanobis Distances

Version 0.1.4

Description Calculates a Mahalanobis distance for every row of a set of outcome variables (Mahalanobis, 1936 <doi:10.1007/s13171-019-00164-5>). The conditional Mahalanobis distance is calculated using a conditional covariance matrix (i.e., a covariance matrix of the outcome variables after controlling for a set of predictors). Plotting the output of the cond_maha() function can help identify which elements of a profile are unusual after controlling for the predictors.

License GPL (>= 3)

URL https://github.com/wjschne/unusualprofile, https://wjschne.github.io/unusualprofile/

BugReports https://github.com/wjschne/unusualprofile/issues

Depends R (>= 3.1)

Imports dplyr, ggnormalviolin, ggplot2, magrittr, purrr, rlang, stats, tibble, tidyr

Suggests bookdown, covr, extrafont, forcats, glue, kableExtra, knitr, lavaan, lifecycle, mvtnorm, patchwork, ragg, rmarkdown, roxygen2, scales, simstandard (>= 0.6.3), stringr, sysfonts, testthat

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData TRUE

RoxygenNote 7.3.1

NeedsCompilation no

Author W. Joel Schneider [aut, cre] (<https://orcid.org/0000-0002-8393-5316>), Feng Ji [aut]

Maintainer W. Joel Schneider <w.joel.schneider@gmail.com>

Repository CRAN

Date/Publication 2024-02-14 23:20:03 UTC

R topics documented:

`cond_maha` *Calculate the conditional Mahalanobis distance for any variables.*

Description

Calculate the conditional Mahalanobis distance for any variables.

Usage

```
cond_maha(
  data,
  R,
  v_dep,
  v_ind = NULL,
  v_ind_composites = NULL,
  mu = 0,
  sigma = 1,
  use_sample_stats = FALSE,
  label = NA
)
```

Arguments

`data`	Data.frame with the independent and dependent variables. Unless mu and sigma are specified, data are assumed to be z-scores.
`R`	Correlation among all variables.
`v_dep`	Vector of names of the dependent variables in your profile.
`v_ind`	Vector of names of independent variables you would like to control for.

v_ind_composites
: Vector of names of independent variables that are composites of dependent variables

mu
: A vector of means. A single value means that all variables have the same mean.

sigma
: A vector of standard deviations. A single value means that all variables have the same standard deviation

use_sample_stats
: If TRUE, estimate R, mu, and sigma from data. Only complete cases are used (i.e., no missing values in v_dep, v_ind, v_ind_composites).

label
: optional tag for labeling output

Value

a list with the conditional Mahalanobis distance

- `dCM` = Conditional Mahalanobis distance
- `dCM_df` = Degrees of freedom for the conditional Mahalanobis distance
- `dCM_p` = A proportion that indicates how unusual this profile is compared to profiles with the same independent variable values. For example, if `dCM_p` = 0.88, this profile is more unusual than 88 percent of profiles after controlling for the independent variables.
- `dM_dep` = Mahalanobis distance of just the dependent variables
- `dM_dep_df` = Degrees of freedom for the Mahalanobis distance of the dependent variables
- `dM_dep_p` = Proportion associated with the Mahalanobis distance of the dependent variables
- `dM_ind` = Mahalanobis distance of just the independent variables
- `dM_ind_df` = Degrees of freedom for the Mahalanobis distance of the independent variables
- `dM_ind_p` = Proportion associated with the Mahalanobis distance of the independent variables
- `v_dep` = Dependent variable names
- `v_ind` = Independent variable names
- `v_ind_singular` = Independent variables that can be perfectly predicted from the dependent variables (e.g., composite scores)
- `v_ind_nonsingular` = Independent variables that are not perfectly predicted from the dependent variables
- `data` = data used in the calculations
- `d_ind` = independent variable data
- `d_inp_p` = Assuming normality, cumulative distribution function of the independent variables
- `d_dep` = dependent variable data
- `d_dep_predicted` = predicted values of the dependent variables
- `d_dep_deviations` = `d_dep - d_dep_predicted` (i.e., residuals of the dependent variables)
- `d_dep_residuals_z` = standardized residuals of the dependent variables
- `d_dep_cp` = conditional proportions associated with standardized residuals
- `d_dep_p` = Assuming normality, cumulative distribution function of the dependent variables

- R2 = Proportion of variance in each dependent variable explained by the independent variables
- zSEE = Standardized standard error of the estimate for each dependent variable
- SEE = Standard error of the estimate for each dependent variable
- ConditionalCovariance = Covariance matrix of the dependent variables after controlling for the independent variables
- distance_reduction = 1 - (dCM / dM_dep) (Degree to which the independent variables decrease the Mahalanobis distance of the dependent variables. Negative reductions mean that the profile is more unusual after controlling for the independent variables. Returns 0 if dM_dep is 0.)
- variability_reduction = 1 - sum((X_dep - predicted_dep) ^ 2) / sum((X_dep - mu_dep) ^ 2) (Degree to which the independent variables decrease the variability the dependent variables (X_dep). Negative reductions mean that the profile is more variable after controlling for the independent variables. Returns 0 if X_dep == mu_dep)
- mu = Variable means
- sigma = Variable standard deviations
- d_person = Data frame consisting of Mahalanobis distance data for each person
- d_variable = Data frame consisting of variable characteristics
- label = label slot

Examples

```
library(unusualprofile)
library(simstandard)

m <- "
Gc =~ 0.85 * Gc1 + 0.68 * Gc2 + 0.8 * Gc3
Gf =~ 0.8 * Gf1 + 0.9 * Gf2 + 0.8 * Gf3
Gs =~ 0.7 * Gs1 + 0.8 * Gs2 + 0.8 * Gs3
Read =~ 0.66 * Read1 + 0.85 * Read2 + 0.91 * Read3
Math =~ 0.4 * Math1 + 0.9 * Math2 + 0.7 * Math3
Gc ~ 0.6 * Gf + 0.1 * Gs
Gf ~ 0.5 * Gs
Read ~ 0.4 * Gc + 0.1 * Gf
Math ~ 0.2 * Gc + 0.3 * Gf + 0.1 * Gs"
# Generate 10 cases
d_demo <- simstandard::sim_standardized(m = m, n = 10)

# Get model-implied correlation matrix
R_all <- simstandard::sim_standardized_matrices(m)$Correlations$R_all

cond_maha(data = d_demo,
          R = R_all,
          v_dep = c("Math", "Read"),
          v_ind = c("Gf", "Gs", "Gc"))
```

d_example — *An example data.frame*

Description

A dataset with 1 row of data for a single case.

Usage

```
d_example
```

Format

A data frame with 1 row and 8 variables:

X_1 A predictor variable

X_2 A predictor variable

X_3 A predictor variable

Y_1 An outcome variable

Y_2 An outcome variable

Y_3 An outcome variable

X A latent predictor variable

Y A latent outcome variable

plot.cond_maha — *Plot the variables from the results of the cond_maha function.*

Description

Plot the variables from the results of the cond_maha function.

Usage

```
## S3 method for class 'cond_maha'
plot(
  x,
  ...,
  p_tail = 0,
  family = "sans",
  score_digits = ifelse(min(x$sigma) >= 10, 0, 2)
)
```

Arguments

`x`	The results of the cond_maha function.
`...`	Arguments passed to print function
`p_tail`	The proportion of the tail to shade
`family`	Font family.
`score_digits`	Number of digits to round scores.

Value

A ggplot2-object

`plot.maha` *Plot objects of the maha class (i.e, the results of the cond_maha function using dependent variables only).*

Description

Plot objects of the maha class (i.e, the results of the cond_maha function using dependent variables only).

Usage

```
## S3 method for class 'maha'
plot(
  x,
  ...,
  p_tail = 0,
  family = "sans",
  score_digits = ifelse(min(x$sigma) >= 10, 0, 2)
)
```

Arguments

`x`	The results of the cond_maha function.
`...`	Arguments passed to print function
`p_tail`	Proportion in violin tail (defaults to 0).
`family`	Font family.
`score_digits`	Number of digits to round scores.

Value

A ggplot2-object

`proportion2percentile` *Rounds proportions to significant digits both near 0 and 1, then converts to percentiles*

Description

Rounds proportions to significant digits both near 0 and 1, then converts to percentiles

Usage

```
proportion2percentile(
  p,
  digits = 2,
  remove_leading_zero = TRUE,
  add_percent_character = FALSE
)
```

Arguments

`p`	probability
`digits`	rounding digits. Defaults to 2
`remove_leading_zero`	Remove leading zero for small percentiles, Defaults to TRUE
`add_percent_character`	Append percent character. Defaults to FALSE

Value

character vector

Examples

```
proportion2percentile(0.01111)
```

`proportion_round` *Rounds proportions to significant digits both near 0 and 1*

Description

Rounds proportions to significant digits both near 0 and 1

Usage

```
proportion_round(p, digits = 2)
```

Arguments

`p`	probability
`digits`	rounding digits

Value

numeric vector

Examples

```
proportion_round(0.01111)
```

`R_example` *An example correlation matrix*

Description

A correlation matrix used for demonstration purposes It is the model-implied correlation matrix for this structural model: `X =~ 0.7 * X_1 + 0.5 * X_2 + 0.8 * X_3 Y =~ 0.8 * Y_1 + 0.7 * Y_2 + 0.9 * Y_3 Y ~ 0.6 * X`

Usage

```
R_example
```

Format

A matrix with 8 rows and 8 columns:

X_1 A predictor variable

X_2 A predictor variable

X_3 A predictor variable

Y_1 An outcome variable

Y_2 An outcome variable

Y_3 An outcome variable

X A latent predictor variable

Y A latent outcome variable

Index