Package ‘robcat’

May 1, 2025

Type Package

Title Robust Categorical Data Analysis

Version 0.1.0

Date 2025-04-25

Maintainer Max Welz <max.welz@uzh.ch>

Description Robust categorical data analysis based on the theory of C-estimation developed in Welz (2024) <doi:10.48550/arXiv.2403.11954>. For now, the package only implements robust estimation of polychoric correlation as proposed in Welz, Mair and Alfons (2024) <doi:10.48550/arXiv.2407.18835> with methods for printing and plotting. We will implement further models in future releases. In addition, the package is still experimental, so input arguments and class structure may change in future releases.

License GPL (>= 2)

Encoding UTF-8

Depends ggplot2

Imports Rcpp (>= 1.0.10), stats, mvtnorm, stringr, parallel

Suggests testthat (>= 3.0.0)

LinkingTo Rcpp

RoxygenNote 7.3.2

NeedsCompilation yes

Author Max Welz [aut, cre] (ORCID: <https://orcid.org/0000-0003-2945-1860>),
Andreas Alfons [aut] (ORCID: <https://orcid.org/0000-0002-2513-3788>),
Patrick Mair [aut] (ORCID: <https://orcid.org/0000-0003-0100-6511>)

Repository CRAN

Date/Publication 2025-05-01 10:40:02 UTC

Contents

initialize_param 2
plot.robpolycor 3
polycor 3

polycormat 5
polycormat_mle 6
polycor_mle 7
print.robpolycor 9
vcov.robpolycor 9

Index **11**

initialize_param *Neutral initialization of starting values*

Description

Initializes starting values for numerical optimization in a neutral way. The optimization problem itself is convex, so the initialization should not matter much.

Usage

```
initialize_param(x, y)
```

Arguments

x	Vector of integer-valued responses to first rating variable, or contingency table (a `table` object).
y	Vector of integer-valued responses to second rating variable; only required if `x` is not a contingency table.

Value

A vector of initial values for the polychoric correlation coefficient, the X-threshold parameters, and the Y-threshold parameters

Examples

```
## example data
set.seed(123)
x <- sample(c(1,2,3), size = 100, replace = TRUE)
y <- sample(c(1,2,3), size = 100, replace = TRUE)
initialize_param(x, y)
```

`plot.robpolycor` *Plot method for classes* `"robpolycor"` *and* `"polycor"`*.*

Description

Plot method for classes `"robpolycor"` and `"polycor"`.

Usage

```
## S3 method for class 'robpolycor'
plot(x, cutoff = 3, ...)
```

Arguments

`x`	Object of class `"robpolycor"` or `"polycor"`.
`cutoff`	Cutoff beyond which the color scale for Pearson residuals is truncated.
`...`	Additional parameters to be passed down.

Value

An object of class `"ggplot"`.

Examples

```
## example data
set.seed(123)
x <- sample(c(1,2,3), size = 100, replace = TRUE)
y <- sample(c(1,2,3), size = 100, replace = TRUE)

fit <- polycor(x,y)
plot(fit)
```

`polycor` *Robust estimation of polychoric correlation*

Description

Implements to robust estimator of Welz, Mair and Alfons (2024, doi:10.48550/arXiv.2407.18835) for the polychoric correlation model, based on the general theory of C-estimation proposed by Welz (2024, doi:10.48550/arXiv.2403.11954).

Usage

```
polycor(
  x,
  y = NULL,
  c = 0.6,
  variance = TRUE,
  constrained = TRUE,
  method = ifelse(constrained, "Nelder-Mead", "L-BFGS-B"),
  maxcor = 0.999,
  tol_thresholds = 0.01,
  init = initialize_param(x, y)
)
```

Arguments

`x` Vector of integer-valued responses to first item, or contingency table (a "`table`" object).

`y` Vector of integer-valued responses to second item; only required if `x` is not a contingency table.

`c` Tuning constant that governs robustness; must be in `[0, Inf]`. Defaults to 0.6.

`variance` Shall an estimated asymptotic covariance matrix be returned? Default is `TRUE`.

`constrained` Shall strict monotonicity of thresholds be explicitly enforced by linear constraints? Default is `TRUE`.

`method` Numerical optimization method.

`maxcor` Maximum absolute correlation (to ensure numerical stability). Default is 0.999.

`tol_thresholds` Minimum distance between consecutive thresholds (to enforce strict monotonicity); only relevant if `constrained = TRUE`. Default is 0.01.

`init` Initialization of numerical optimization. Default is neutral.

Value

An object of class `"robpolycor"`, which is a list with the following components.

`theahat` A vector of estimates for the polychoric correlation coefficient (`rho`) as well as thresholds for `x` (named `a1,a2,...,a_{Kx-1}`) and `y` (named `b1,b2,...,b_{Ky-1}`).

`stderr` A vector of standard errors for each estimate in `theahat`.

`vcov` Estimated asymptotic covariance matrix of `theahat`. The matrix Σ in the paper (asymptotic covariance matrix of $\sqrt{N}\hat{\theta}$) can be obtained via `vcov * N`, where `N` is the sample size.

`chisq,pval,df` Currently `NULL`, will in a future release be the test statistic, p-value, and degrees of freedom of a test for bivariate normality.

`objective` Value of minimized loss function.

`optim` Object of class `optim`.

Examples

```
## example data
set.seed(123)
x <- sample(c(1,2,3), size = 100, replace = TRUE)
y <- sample(c(1,2,3), size = 100, replace = TRUE)

polycor(x,y)     # robust
polycor_mle(x,y) # non-robust MLE
```

---

polycormat *Robust estimation of polychoric correlation matrix*

Description

A useful wrapper of `polycor` to robustly estimate a polychoric correlation matrix by calculating all unique pairwise polychoric correlation coefficients.

Usage

```
polycormat(
  data,
  c = 0.6,
  parallel = FALSE,
  num_cores = 1L,
  return_polycor = TRUE,
  variance = TRUE,
  constrained = TRUE,
  method = ifelse(constrained, "Nelder-Mead", "L-BFGS-B"),
  maxcor = 0.999,
  tol_thresholds = 0.01
)
```

Arguments

`data` Data matrix or `data.frame` of integer-valued responses, individual respondents are in rows and responses to the items in the columns.

`c` tuning constant that governs robustness; takes values in `[0, Inf]`. Defaults to 0.6.

`parallel` Logical. Shall parallelization be used? Default is `FALSE`.

`num_cores` Number of cores to be used, only relevant if `parallel` = `TRUE`. Defaults to the number of system cores.

`return_polycor` Logical. Shall the individual `"polycor"` objects for each item pair estimate be returned? Deafult is `TRUE`.

`variance` Shall an estimated asymptotic covariance matrix be returned? Default is `TRUE`.

`constrained` Shall strict monotonicity of thresholds be explicitly enforced by linear constraints?

`method` Numerical optimization method.

`maxcor` Maximum absolute correlation (to ensure numerical stability).

`tol_thresholds` Minimum distance between consecutive thresholds (to enforce strict monotonicity); only relevant if `constrained = TRUE`.

Value

If `return_polycor = TRUE`, returns a list with a polychoric correlation matrix and list of `"polycor"` objects. If `return_polycor = FALSE`, then only a correlation matrix is returned.

Examples

```
## example data
set.seed(123)
data <- matrix(sample(c(1,2,3), size = 3*100, replace = TRUE), nrow = 100)
polycormat(data)     # robust
polycormat_mle(data) # non-robust MLE
```

polycormat_mle *Maximum likelihood estimation of polychoric correlation matrix*

Description

A useful wrapper of `polycor_mle` to estimate a polychoric correlation matrix via maximum likelihood by calculating all unique pairwise polychoric correlation coefficients.

Usage

```
polycormat_mle(
  data,
  parallel = FALSE,
  num_cores = 1L,
  return_polycor = TRUE,
  variance = TRUE,
  constrained = TRUE,
  method = ifelse(constrained, "Nelder-Mead", "L-BFGS-B"),
  maxcor = 0.999,
  tol_thresholds = 0.01
)
```

Arguments

`data` Data matrix or `data.frame` of integer-valued responses, individual respondents are in rows and responses to the items in the columns.

`parallel` Logical. Shall parallelization be used? Default is `FALSE`.

`num_cores` Number of cores to be used, only relevant if `parallel = TRUE`. Defaults to the number of system cores.

`return_polycor` Logical. Shall the individual `"polycor"` objects for each item pair estimate be returned? Deafult is `TRUE`.

`variance` Shall an estimated asymptotic covariance matrix be returned? Default is `TRUE`.

`constrained` Shall strict monotonicity of thresholds be explicitly enforced by linear constraints?

`method` Numerical optimization method.

`maxcor` Maximum absolute correlation (to ensure numerical stability).

`tol_thresholds` Minimum distance between consecutive thresholds (to enforce strict monotonicity); only relevant if `constrained = TRUE`.

Value

If `return_polycor = TRUE`, returns a list with a polychoric correlation matrix and list of `"polycor"` objects. If `return_polycor = FALSE`, then only a correlation matrix is returned.

Examples

```
## example data
set.seed(123)
data <- matrix(sample(c(1,2,3), size = 3*100, replace = TRUE), nrow = 100)
polycormat(data)     # robust
polycormat_mle(data) # non-robust MLE
```

`polycor_mle` *Maximum likelihood estimation of polychoric correlation coefficient*

Description

Implements the maximum likelihood estimator of Olsson (1979, Psychometrika, doi:10.1007/BF02296207) for the polychoric correlation model.

Usage

```
polycor_mle(
  x,
  y = NULL,
  variance = TRUE,
  constrained = TRUE,
  twostep = FALSE,
  method = ifelse(constrained, "Nelder-Mead", "L-BFGS-B"),
  maxcor = 0.999,
  tol_thresholds = 0.01,
  init = initialize_param(x, y)
)
```

Arguments

`x` Vector of integer-valued responses to first item, or contingency table (a "`table`" object).

`y` Vector of integer-valued responses to second item; only required if `x` is not a contingency table.

`variance` Shall an estimated asymptotic covariance matrix be returned? Default is `TRUE`.

`constrained` shall strict monotonicity of thresholds be explicitly enforced by linear constraints? Only relevant if `twostep = FALSE`. Default is `TRUE`.

`twostep` Shall two-step estimation of Olsson (1979) <doi:10.1007/BF02296207> be performed? Default is `FALSE`.

`method` Numerical optimization method; default is Nelder-Mead.

`maxcor` Maximum absolute correlation (to ensure numerical stability). Deafult is 0.999.

`tol_thresholds` Minimum distance between consecutive thresholds (to enforce strict monotonicity); only relevant if `constrained = TRUE`. Default is 0.01.

`init` Initialization of numerical optimization. Default is neutral. If `twostep = TRUE`, only the first element (the correlation coefficient) will be used.

Value

An object of class `"robpolycor"`. See `polycor()` for details.

Examples

```
## example data
set.seed(123)
x <- sample(c(1,2,3), size = 100, replace = TRUE)
y <- sample(c(1,2,3), size = 100, replace = TRUE)

polycor(x,y)     # robust
polycor_mle(x,y) # non-robust MLE
```

`print.robpolycor` *Print method for classes* `"robpolycor"` *and* `"polycor"`.

Description

Print method for classes `"robpolycor"` and `"polycor"`.

Usage

```
## S3 method for class 'robpolycor'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

`x`	Object of class `"robpolycor"` or `"polycor"`.
`digits`	Number of digits to be printed.
`...`	Additional parameters to be passed down.

Value

A print to the console.

Examples

```
set.seed(123)
x <- sample(c(1,2,3), size = 100, replace = TRUE)
y <- sample(c(1,2,3), size = 100, replace = TRUE)
fit <- polycor(x,y)

print(fit)
fit # equivalent
```

`vcov.robpolycor` *Obtain estimated asymptotic variance-covariance matrix*

Description

Method for classes `"robpolycor"` and `"polycor"`. Returns the estimated asymptotic variance-covariance matrix of a point estimate `theahat`. The matrix Σ in the paper (asymptotic variance-covariance matrix of $\sqrt{N}\hat{\theta}$) can be obtained via multiplying the returned matrix by the sample size.

Usage

```
## S3 method for class 'robpolycor'
vcov(object, ...)
```

Arguments

`object` Object of class "`robpolycor`" or "`polycor`".

`...` Additional parameters to be passed down.

Value

A numeric matrix, being the estimated asymptotic covariance matrix for the model parameters

Examples

```
set.seed(123)
x <- sample(c(1,2,3), size = 100, replace = TRUE)
y <- sample(c(1,2,3), size = 100, replace = TRUE)
fit <- polycor(x,y)

vcov(fit)
```

Index

data.frame, *5*, *7*

ggplot, *3*

initialize_param, 2

plot.robpolycor, 3
polycor, 3, *5–8*
polycor_mle, *6*, 7
polycormat, 5
polycormat_mle, 6
print.robpolycor, 9

table, *4*, *8*

vcov.robpolycor, 9