Package: robcat (via r-universe)

May 11, 2026

Type Package

Title Robust Categorical Data Analysis

Version 0.2

Date 2026-02-10

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 (2026) <doi:10.1017/psy.2025.10066> and robust estimation of polyserial correlation (Welz, 2026 <doi:10.1017/psy.2026.10091>) 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, Matrix, numDeriv, pracma, utils

Suggests testthat (>= 3.0.0)

LinkingTo Rcpp

RoxygenNote 7.3.2

NeedsCompilation yes

Config/pak/sysreqs libicu-dev

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

Date/Publication 2026-02-10 10:53:08 UTC

RemoteUrl https://github.com/mwelz/robcat

RemoteRef HEAD

RemoteSha 77ee32cc47dfc94a7d894b9d21065d6c690d0284

Contents

`initialize_param` *Neutral initialization of starting values for polychoric correlation*

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 = "ifneeded",
  method = NULL,
  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? This can be a logical (TRUE or FALSE), or "ifneeded" to first try unconstrained optimization and in case of an error perform constrained optimization. Default is "ifneeded".
method	Numerical optimization method, see optim() and constrOptim(). Default is to use "L-BFGS-B" in case of unconstrained optimization and "Nelder-Mead" in case of constrained optimization.
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 in case of constrained optimization. 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.

thetahat 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 thetahat.

vcov Estimated asymptotic covariance matrix of thetahat. 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
```

---

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 = "ifneeded",
  twostep = FALSE,
  method = NULL,
  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? This can be a logical (TRUE or FALSE), or "ifneeded" to first try unconstrained optimization and in case of an error perform constrained optimization. Default is "ifneeded".
twostep	Shall two-step estimation of Olsson (1979) <doi:10.1007/BF02296207> be performed? Default is FALSE.
method	Numerical optimization method, see optim() and constrOptim(). Default is to use "L-BFGS-B" in case of unconstrained optimization and "Nelder-Mead" in case of constrained optimization.
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 in case of constrained optimization. Default is 0.01.
init	Initialization of numerical optimization. Default is neutral.

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
```

---

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 = "ifneeded",
  method = NULL,
  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; must be 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? Default 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? This can be a logical (TRUE or FALSE), or "ifneeded" to first try unconstrained optimization and in case of an error perform constrained optimization. Default is "ifneeded".
method	Numerical optimization method, see optim() and constrOptim(). Default is to use "L-BFGS-B" in case of unconstrained optimization and "Nelder-Mead" in case of constrained optimization.
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 in case of constrained optimization. Default is 0.01.

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 = "ifneeded",
  method = NULL,
  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? Default 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? This can be a logical (`TRUE` or `FALSE`), or `"ifneeded"` to first try unconstrained optimization and in case of an error perform constrained optimization. Default is `"ifneeded"`.
`method`	Numerical optimization method, see `optim()` and `constrOptim()`. Default is to use `"L-BFGS-B"` in case of unconstrained optimization and `"Nelder-Mead"` in case of constrained optimization.
`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 in case of constrained optimization. Default is 0.01.

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
```

`polyserial` *Robust estimation of polyserial correlation*

Description

Implements the robust estimator of Welz (2025, doi:10.48550/arXiv.2510.15632) for the polyserial correlation model.

Usage

```
polyserial(
  x,
  y,
  alpha = 0.5,
  num_y = max(y),
  constrained = "ifneeded",
  method = NULL,
  variance = TRUE,
  weights = TRUE,
  init = polyserial_initialize_param(x = x, num_y = num_y, robust = TRUE),
  maxcor = 0.999,
  tol_thresholds = 0.01,
  tol_sigma2 = 0.01
)
```

Arguments

`x`	Vector of numeric values.
`y`	Vector of integer-valued ordinal values.
`alpha`	Tuning constant that governs robustness-efficiency tradeoff; must be in `[0, Inf]`. Defaults to 0.5.
`num_y`	Number of response categories in y; defaults to max(y)

constrained	Shall parameter restructions be enforced by linear constraints? This can be a logical (TRUE or FALSE), or "ifneeded" to first try unconstrained optimization and in case of an error perform constrained optimization. Default is "ifneeded".
method	Numerical optimization method, see optim() and constrOptim(). Default is to use "BFGS" in case of unconstrained optimization and "Nelder-Mead" in case of constrained optimization.
variance	Shall an estimated asymptotic covariance matrix be returned? Default is TRUE.
weights	Shall weights be returned? Default is TRUE.
init	Initialization of numerical optimization. Default is neutral.
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 in case of constrained optimization. Default is 0.01.
tol_sigma2	Minimum value of sigma2 parameter (population variance of X); only relevant in case of constrained optimization. Default is 0.01.

Value

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

thetahat A vector of estimates for the polyserial correlation coefficient (rho), population mean of X (mu), population variance of Y (sigma2), as well as thresholds for y (named tau1, tau2, ..., tau_{r-1}).

stderr A vector of standard errors for each estimate in thetahat.

vcov Estimated asymptotic covariance matrix of thetahat. 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.

pointpolyserial Estimated polyserial correlation coefficient, calculated with provided scoring of Y

weights List of rescaled and raw outlyingness weights for each observation as well as maximum possible raw weight that was used for rescaling (sup).

objective Value of minimized loss function.

optim Object of class optim.

inputs List of provided inputs.

Examples

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

polyserial(x,y)
```

polyserial_efficiency *Efficiency of minimum DPD estimators of polyserial model*

Description

Calculate population asymptotic variance-covariance matrix associated with a parameter vector `theta`, assuming that the polyserial model is correctly specified and that `theta` is the true model parameter. May take a few moments to compute because a relatively large number of integrals need to be numerically solved.

Usage

```
polyserial_efficiency(theta, alpha)
```

Arguments

`theta`	Parameter vector of polyserial model; assumed to be the true one. First element is polyserial correlatio coefficient, second is the population mean of X, third is population variance of X, and the remaining elements are the thresholds associated with the ordnal Y (must be in increasing order)
`alpha`	Tuning constant governing robustness-efficiency tradeoff. Set to 0 for maximum likelihood.

Value

A numeric matrix that is the population asymptotic variance-covariance matrix associated with a parameter vector `theta` and tuning constant `alpha`.

Examples

```
theta <- c(rho = 0, mu = 0, sigma2 = 1, tau1 = 0) # true parameter vector
polyserial_efficiency(theta, alpha = 0.5)
```

polyserial_initialize_param *Neutral initialization of starting values for polyserial correlation*

Description

Initializes starting values for numerical optimization in a neutral way.

Usage

```
polyserial_initialize_param(x, num_y, robust = FALSE)
```

Arguments

x	Vector of numeric values
num_y	Number of response options of ordinal variable
robust	Should values of mu and sigma2 be initialized in robust way (that is, median and squared MAD)? If FALSE (default), then nonrobust sample mean and variance are used.

Value

A vector of initial values for the polyserial correlation coefficient, mu, sigma2, and Y-threshold parameters

Examples

```
## example data
set.seed(123)
x <- rnorm(100)
polyserial_initialize_param(x = x, num_y = 3)
```

---

polyserial_mle *Maximum likelihood estimation of polyserial correlation*

Description

Implements maximum likelihood estimation of the polyserial correlation model.

Usage

```
polyserial_mle(
  x,
  y,
  num_y = max(y),
  constrained = "ifneeded",
  method = NULL,
  variance = TRUE,
  init = polyserial_initialize_param(x = x, num_y = num_y, robust = FALSE),
  maxcor = 0.999,
  tol_thresholds = 0.01,
  tol_sigma2 = 0.01
)
```

Arguments

x	Vector of numeric values.
y	Vector of integer-valued ordinal values.
num_y	Number of response categories in y; defaults to max(y)
constrained	Shall parameter restructions be enforced by linear constraints? This can be a logical (TRUE or FALSE), or "ifneeded" to first try unconstrained optimization and in case of an error perform constrained optimization. Default is "ifneeded".
method	Numerical optimization method, see optim() and constrOptim(). Default is to use "BFGS" in case of unconstrained optimization and "Nelder-Mead" in case of constrained optimization.
variance	Shall an estimated asymptotic covariance matrix be returned? Default is TRUE.
init	Initialization of numerical optimization. Default is neutral.
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 in case of constrained optimization. Default is 0.01.
tol_sigma2	Minimum value of sigma2 parameter (population variance of X); only relevant in case of constrained optimization. Default is 0.01.

Value

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

thetahat A vector of estimates for the polyserial correlation coefficient (rho), population mean of X (mu), population variance of Y (sigma2), as well as thresholds for y (named tau1, tau2, ..., tau_{r-1}).

stderr A vector of standard errors for each estimate in thetahat.

vcov Estimated asymptotic covariance matrix of thetahat. 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.

pointpolyserial Estimated polyserial correlation coefficient, calculated with provided scoring of Y

objective Value of minimized loss function.

optim Object of class optim.

inputs List of provided inputs.

Examples

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

polyserial_mle(x,y)
```

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
```

print.robpolyserial *Print method for classes* "robpolyserial" *and* "polyserial".

Description

Print method for classes "robpolyserial" and "polyserial".

Usage

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

Arguments

x	Object of class "robpolyserial" or "polyserial".
digits	Number of digits to be printed.
...	Additional parameters to be passed down.

Value

A print to the console.

Examples

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

print(fit)
fit # equivalent
```

---

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

Description

Estimate asymptotic variance-covariance matrix of polychoric model.

Usage

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

Arguments

object	Object of class "robpolycor" or "polycor".
...	Additional parameters to be passed down.

Details

Method for classes "robpolycor" and "polycor". Returns the estimated asymptotic variance-covariance matrix of a point estimate thetahat. 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.

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)
```

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

Description

Estimate asymptotic variance-covariance matrix of polyserial model.

Usage

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

Arguments

`object`	Object of class `"robpolyserial"` or `"polyserial"`.
`...`	Additional parameters to be passed down.

Details

Method for classes `"robpolyserial"` and `"polyserial"`. Returns the estimated asymptotic variance-covariance matrix of a point estimate `thetahat`. 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.

Value

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

Examples

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

fit <- polyserial(x,y)
vcov(fit)
```

Index