Package 'polycor'

January 11, 2022

Version 0.8-1

Date 2022-01-10

Title Polychoric and Polyserial Correlations

Depends R (>= 3.3.0)

Imports stats, mvtnorm, Matrix, parallel, admisc (>= 0.22)

ByteCompile yes

LazyLoad yes

Description Computes polychoric and polyserial correlations by quick ``two-step'' methods or ML, optionally with standard errors; tetrachoric and biserial correlations are special cases.

License GPL (>= 2)

URL https://r-forge.r-project.org/projects/polycor/, https://CRAN.R-project.org/package=polycor

NeedsCompilation no

Author John Fox [aut, cre], Adrian Dusa [ctb]

Maintainer John Fox <jfox@mcmaster.ca>

Repository CRAN

Date/Publication 2022-01-11 19:22:44 UTC

R topics documented:

hetcor *Heterogeneous Correlation Matrix*

Description

hetcor computes a heterogenous correlation matrix, consisting of Pearson product-moment correlations between numeric variables, polyserial correlations between numeric and ordinal variables, and polychoric correlations between ordinal variables.

The detectCores function is imported from the **parallel** package and re-exported.

Usage

```
hetcor(data, ..., ML = FALSE, std.err = TRUE,
  use=c("complete.obs", "pairwise.complete.obs"),
  bins=4, pd=TRUE, parallel=FALSE, ncores=detectCores(logical=FALSE),
  thresholds=FALSE)
## S3 method for class 'data.frame'
hetcor(data, ML = FALSE, std.err = TRUE,
  use = c("complete.obs", "pairwise.complete.obs"),
  bins=4, pd=TRUE, parallel=FALSE, ncores=detectCores(logical=FALSE),
  thresholds=FALSE, ...)
## Default S3 method:
hetcor(data, ..., ML = FALSE, std.err = TRUE,
  use=c("complete.obs", "pairwise.complete.obs"),
  bins=4, pd=TRUE, parallel=FALSE, ncores=detectCores(logical=FALSE),
  thresholds=FALSE)
## S3 method for class 'hetcor'
print(x, digits = max(3, getOption("digits") - 3), ...)
## S3 method for class 'hetcor'
as.matrix(x, ...)
detectCores(all.tests=FALSE, logical=TRUE)
```

Arguments

data	a data frame consisting of factors, ordered factors, logical variables, character variables, and/or numeric variables, or the first of several variables.
...	variables and/or arguments to be passed down.
ML	if TRUE, compute maximum-likelihood estimates; if FALSE, compute quick two-step estimates.
std.err	if TRUE, compute standard errors.
bins	number of bins to use for continuous variables in testing bivariate normality; the default is 4.
pd	if TRUE and if the correlation matrix is not positive-definite, an attempt will be made to adjust it to a positive-definite matrix, using the nearPD function in the Matrix package. Note that default arguments to nearPD are used (except corr=TRUE); for more control call nearPD directly.

parallel	if TRUE (the default is FALSE), perform parallel computations on a computer with multiple CPUs/cores.
ncores	the number of cores to use for parallel computations; the default is the number of physical cores detected.
use	if "complete.obs", remove observations with any missing data; if "pairwise.complete.obs", compute each correlation using all observations with valid data for that pair of variables.
thresholds	if TRUE (the default is FALSE), include the estimated thresholds for polyserial and polychoric correlation in the returned object.
x	an object of class "hetcor" to be printed, or from which to extract the correlation matrix.
digits	number of significant digits.
all.tests	logical, apply all known tests; default is FALSE.
logical	if TRUE, detect logical CPUs/cores; if FALSE, detect physical CPUs/cores.

Value

hetcor returns an object of class "hetcor" with the following components:

correlations	the correlation matrix.
type	the type of each correlation: "Pearson", "Polychoric", or "Polyserial".
std.errors	the standard errors of the correlations, if requested.
n	the number (or numbers) of observations on which the correlations are based.
tests	p-values for tests of bivariate normality for each pair of variables.
NA.method	the method by which any missing data were handled: "complete.obs" or "pairwise.complete.obs".
ML	TRUE for ML estimates, FALSE for two-step estimates.
thresholds	optionally, according to the thresholds argument, a matrix of mode list with a list of thresholds for each polyserial and polychoric correlation in the elements below the diagonal and the type of each correlation (Pearson, polyserial, or polychoric) above the diagonal.

Warning

Be careful with character variables (as opposed to factors), the values of which are ordered alphabetically. Thus, e.g., the values "disagree", "neutral", "agree" are ordered "agree", "disagree", "neutral".

Note

Although the function reports standard errors for product-moment correlations, transformations (the most well known is Fisher's z-transformation) are available that make the approach to asymptotic normality much more rapid.

Author(s)

John Fox <jfox@mcmaster.ca>

References

Drasgow, F. (1986) Polychoric and polyserial correlations. Pp. 68-74 in S. Kotz and N. Johnson, eds., *The Encyclopedia of Statistics, Volume 7.* Wiley.

Olsson, U. (1979) Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika* **44**, 443-460.

Rodriguez, R.N. (1982) Correlation. Pp. 193-204 in S. Kotz and N. Johnson, eds., *The Encyclopedia of Statistics, Volume 2.* Wiley.

Ghosh, B.K. (1966) Asymptotic expansion for the moments of the distribution of correlation coefficient. *Biometrika* **53**, 258-262.

Olkin, I., and Pratt, J.W. (1958) Unbiased estimation of certain correlation coefficients. *Annals of Mathematical Statistics* **29**, 201-211.

See Also

polychor, polyserial, nearPD, detectCores

Examples

```
if(require(mvtnorm)){
    set.seed(12345)
    R <- matrix(0, 4, 4)
    R[upper.tri(R)] <- runif(6)
    diag(R) <- 1
    R <- cov2cor(t(R) %*% R)
    round(R, 4)  # population correlations
    data <- rmvnorm(1000, rep(0, 4), R)
    round(cor(data), 4)   # sample correlations
    }
if(require(mvtnorm)){
    x1 <- data[,1]
    x2 <- data[,2]
    y1 <- cut(data[,3], c(-Inf, .75, Inf))
    y2 <- cut(data[,4], c(-Inf, -1, .5, 1.5, Inf))
    data <- data.frame(x1, x2, y1, y2)
    hetcor(data)  # Pearson, polychoric, and polyserial correlations, 2-step est.
    }
if(require(mvtnorm)){
   hetcor(x1, x2, y1, y2, ML=TRUE) # Pearson, polychoric, polyserial correlations, ML est.
    }

## Not run:
    hc <- hetcor(data, ML=TRUE)
      # parallel computation:
    hc.m <- hetcor(data, ML=TRUE, parallel=TRUE,
                   ncores=min(2, detectCores()))
    hc.m
    all.equal(hc, hc.m)

      # error handling:
    data$y1[data$y2 == "(0.5,1.5]"] <- NA
```

```
    hetcor(data)

## End(Not run)
```

---

polychor *Polychoric Correlation*

Description

Computes the polychoric correlation (and its standard error) between two ordinal variables or from their contingency table, under the assumption that the ordinal variables dissect continuous latent variables that are bivariate normal. Either the maximum-likelihood estimator or a (possibly much) quicker "two-step" approximation is available. For the ML estimator, the estimates of the thresholds and the covariance matrix of the estimates are also available.

Usage

```
polychor(x, y, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, start, thresholds=FALSE)
```

Arguments

x	a contingency table of counts or an ordered categorical variable; the latter can be numeric, logical, a factor, an ordered factor, or a character variable, but if a factor, its levels should be in proper order, and the values of a character variable are ordered alphabetically.
y	if x is a variable, a second ordered categorical variable.
ML	if TRUE, compute the maximum-likelihood estimate; if FALSE, the default, compute a quicker "two-step" approximation.
control	optional arguments to be passed to the optim function.
std.err	if TRUE, return the estimated variance of the correlation (for the two-step estimator) or the estimated covariance matrix (for the ML estimator) of the correlation and thresholds; the default is FALSE.
maxcor	maximum absolute correlation (to insure numerical stability).
start	optional start value(s): if a single number, start value for the correlation; if a list with the elements rho, row.thresholds, and column.thresholds, start values for these parameters; start values are supplied automatically if omitted, and are only relevant when the ML estimator or standard errors are selected.
thresholds	if TRUE (the default is FALSE) return estimated thresholds along with the estimated correlation even if standard errors aren't computed.

Details

The ML estimator is computed by maximizing the bivariate-normal likelihood with respect to the thresholds for the two variables ($\tau_i^x, i = 1, \ldots, r-1$; $\tau_j^y, j = 1, \ldots, c-1$) and the population correlation (ρ). Here, r and c are respectively the number of levels of x and y. The likelihood is maximized numerically using the `optim` function, and the covariance matrix of the estimated parameters is based on the numerical Hessian computed by `optim`.

The two-step estimator is computed by first estimating the thresholds ($\tau_i^x, i = 1, \ldots, r-1$ and $\tau_j^y, i = j, \ldots, c-1$) separately from the marginal distribution of each variable. Then the one-dimensional likelihood for ρ is maximized numerically, using `optim` if standard errors are requested, or `optimise` if they are not. The standard error computed treats the thresholds as fixed.

Value

If `std.err` or `thresholds` is `TRUE`, returns an object of class `"polycor"` with the following components:

`type`	set to `"polychoric"`.
`rho`	the polychoric correlation.
`row.cuts`	estimated thresholds for the row variable (`x`), for the ML estimate.
`col.cuts`	estimated thresholds for the column variable (`y`), for the ML estimate.
`var`	the estimated variance of the correlation, or, for the ML estimate, the estimated covariance matrix of the correlation and thresholds.
`n`	the number of observations on which the correlation is based.
`chisq`	chi-square test for bivariate normality.
`df`	degrees of freedom for the test of bivariate normality.
`ML`	`TRUE` for the ML estimate, `FALSE` for the two-step estimate.

Othewise, returns the polychoric correlation.

Author(s)

John Fox <jfox@mcmaster.ca>

References

Drasgow, F. (1986) Polychoric and polyserial correlations. Pp. 68–74 in S. Kotz and N. Johnson, eds., *The Encyclopedia of Statistics, Volume 7.* Wiley.

Olsson, U. (1979) Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika* **44**, 443-460.

See Also

`hetcor`, `polyserial`, `print.polycor`, `optim`

Examples

```
if(require(mvtnorm)){
    set.seed(12345)
    data <- rmvnorm(1000, c(0, 0), matrix(c(1, .5, .5, 1), 2, 2))
    x <- data[,1]
    y <- data[,2]
    cor(x, y)  # sample correlation
    }
if(require(mvtnorm)){
    x <- cut(x, c(-Inf, .75, Inf))
    y <- cut(y, c(-Inf, -1, .5, 1.5, Inf))
    polychor(x, y)  # 2-step estimate
    }
if(require(mvtnorm)){
    polychor(x, y, ML=TRUE, std.err=TRUE)  # ML estimate
    }
```

---

polyserial *Polyserial Correlation*

Description

Computes the polyserial correlation (and its standard error) between a quantitative variable and an ordinal variable, based on the assumption that the joint distribution of the quantitative variable and a latent continuous variable underlying the ordinal variable is bivariate normal. Either the maximum-likelihood estimator or a quicker "two-step" approximation is available. For the ML estimator the estimates of the thresholds and the covariance matrix of the estimates are also available.

Usage

```
polyserial(x, y, ML = FALSE, control = list(),
  std.err = FALSE, maxcor=.9999, bins=4, start, thresholds=FALSE)
```

Arguments

x	a numerical variable.
y	an ordered categorical variable; can be numeric, logical, a factor, an ordered factor, or a character variables, but if a factor, its levels should be in proper order, and the values of a character variable are ordered alphabetically.
ML	if TRUE, compute the maximum-likelihood estimate; if FALSE, the default, compute a quicker "two-step" approximation.
control	optional arguments to be passed to the optim function.
std.err	if TRUE, return the estimated variance of the correlation (for the two-step estimator) or the estimated covariance matrix of the correlation and thresholds (for the ML estimator); the default is FALSE.
maxcor	maximum absolute correlation (to insure numerical stability).

bins	the number of bins into which to dissect x for a test of bivariate normality; the default is 4.
start	optional start value(s): if a single number, start value for the correlation; if a list with the elements rho and thresholds, start values for these parameters; start values are supplied automatically if omitted, and are only relevant when the ML estimator or standard errors are selected.
thresholds	if TRUE (the default is FALSE) return estimated thresholds along with the estimated correlation even if standard errors aren't computed.

Details

The ML estimator is computed by maximizing the bivariate-normal likelihood with respect to the thresholds for y ($\tau_j^y, i = 1, \ldots, c-1$) and the population correlation (ρ). The likelihood is maximized numerically using the optim function, and the covariance matrix of the estimated parameters is based on the numerical Hessian computed by optim.

The two-step estimator is computed by first estimating the thresholds ($\tau_j^y, i = 1, \ldots, c-1$) from the marginal distribution of y. Then if the standard error of $\hat{\rho}$ is requested, the one-dimensional likelihood for ρ is maximized numerically, using optim if standard errors are requested; the standard error computed treats the thresholds as fixed. If the standard error isn't request, $\hat{\rho}$ is computed directly.

Value

If std.err or thresholds is TRUE, returns an object of class "polycor" with the following components:

type	set to "polyserial".
rho	the polyserial correlation.
cuts	estimated thresholds for the ordinal variable (y), for the ML estimator.
var	the estimated variance of the correlation, or, for the ML estimator, the estimated covariance matrix of the correlation and thresholds.
n	the number of observations on which the correlation is based.
chisq	chi-square test for bivariate normality.
df	degrees of freedom for the test of bivariate normality.
ML	TRUE for the ML estimate, FALSE for the two-step estimate.

Othewise, returns the polyserial correlation.

Author(s)

John Fox <jfox@mcmaster.ca>

References

Drasgow, F. (1986) Polychoric and polyserial correlations. Pp. 68–74 in S. Kotz and N. Johnson, eds., *The Encyclopedia of Statistics, Volume 7.* Wiley.

See Also

hetcor, polychor, print.polycor, optim

Examples

```
if(require(mvtnorm)){
    set.seed(12345)
    data <- rmvnorm(1000, c(0, 0), matrix(c(1, .5, .5, 1), 2, 2))
    x <- data[,1]
    y <- data[,2]
    cor(x, y)  # sample correlation
    }
if(require(mvtnorm)){
    y <- cut(y, c(-Inf, -1, .5, 1.5, Inf))
    polyserial(x, y)  # 2-step estimate
    }
if(require(mvtnorm)){
    polyserial(x, y, ML=TRUE, std.err=TRUE) # ML estimate
    }
```

---

print.polycor *Standard Methods for polycor Objects*

Description

Some standard methods for objects of class polycor, produced by polychor and polyserial, including print, summary, coef, and vcov. The summary method simply invokes the print method.

Usage

```
## S3 method for class 'polycor'
print(x, digits = max(3, getOption("digits") - 3), ...)
## S3 method for class 'polycor'
summary(object, ...)
## S3 method for class 'polycor'
coef(object, correlation=TRUE, thresholds=TRUE, ...)
## S3 method for class 'polycor'
vcov(object, correlation=TRUE, thresholds=TRUE, ...)
```

Arguments

x, object	an object of class polycor, as returned by polychor or polyserial.
digits	number of significant digits to be printed.
correlation	return the estimated correlation or sampling variance of the correlation.
thresholds	return the estimated thresholds or sampling variances/covariances of the thresholds.
...	pass arguments from summary to print; otherwise not used.

Author(s)

John Fox <jfox@mcmaster.ca>

See Also

polychor, polyserial

Examples

```
if(require(mvtnorm)){
    set.seed(12345)
    data <- rmvnorm(1000, c(0, 0), matrix(c(1, .5, .5, 1), 2, 2))
    x <- data[,1]
    y <- data[,2]
    cor(x, y) # sample correlation
    }
if(require(mvtnorm)){
    x <- cut(x, c(-Inf, .75, Inf))
    y <- cut(y, c(-Inf, -1, .5, 1.5, Inf))
    print(polychor(x, y, ML=TRUE, std.err=TRUE),
      digits=3)  # polychoric correlation, ML estimate
    }
if(require(mvtnorm)){
    coef(polychor(x, y, ML=TRUE, std.err=TRUE))
    }
if(require(mvtnorm)){
    vcov(polychor(x, y, ML=TRUE, std.err=TRUE))
    }
```

Index