

Package: conogive (via r-universe)

June 9, 2026

Title Congeneric Normal-Ogive Model

Version 1.0.0

Description The congeneric normal-ogive model is a popular model for psychometric data (McDonald, R. P. (1997) [doi:10.1007/978-1-4757-2691-6_15](https://doi.org/10.1007/978-1-4757-2691-6_15)). This model estimates the model, calculates theoretical and concrete reliability coefficients, and predicts the latent variable of the model. This is the companion package to Moss (2020) [doi:10.31234/osf.io/nvg5d](https://doi.org/10.31234/osf.io/nvg5d).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL <https://github.com/JonasMoss/conogive>

BugReports <https://github.com/JonasMoss/conogive/issues>

RoxygenNote 7.1.0

Imports psych, mvtnorm, checkmate, assertthat

Suggests testthat (>= 2.1.0), psychTools, covr

NeedsCompilation no

Author Jonas Moss [aut, cre] (<https://orcid.org/0000-0002-6876-6964>)

Maintainer Jonas Moss <jonas.gjertsen@gmail.com>

Repository <https://cranhaven.r-universe.dev>

Date/Publication 2026-06-09 10:02:00 UTC

RemoteUrl <https://github.com/cranhaven/cranhaven.r-universe.dev>

RemoteRef package/conogive

RemoteSha aba0a9e2e19e6b4cf7f9d02c039a279fca55231f

RemoteSubdir conogive

Contents

conogive	2
message_cuts	3
predict.conogive	3
reliability	4
Index	6

conogive

Estimate a Congeneric Normal-Ogive Model

Description

conogive is used to estimate congeneric normal-ogive models (McDonald, R. P. (1997)).

Usage

```
conogive(data, use = "complete.obs", ...)
```

Arguments

data	A data frame of observations or a named list with elements lambda, sigma, and cuts. See the details.
use	Passed to stats::cov; defaults to "complete.obs".
...	Passed to psych::fa, where fm = "ml" by default.

Details

The data argument can be either a list containing the parameters of a normal-ogive model, or raw data. If actual data is passed to data, it is passed to psych::polychoric to estimate its polychoric correlation matrix and cutoffs. This is passed to psych::fa to do a barebones multivariate normal-ogive model. The ... arguments are passed to psych::fa, which is called with fm = "ml" by default.

Likert data should start with 1, not 0.

Value

An object of class conogive.

References

McDonald, R. P. (1997). Normal-ogive multidimensional model. In W. J. van der Linden & R. K. Hambleton (Eds.), *Handbook of Modern Item Response Theory* (pp. 257–269). Springer. https://doi.org/10.1007/978-1-4757-2691-6_15

Moss, J. (2020). Please avoid the standardized alpha and the ordinal alpha. <https://psyarxiv.com/nvg5d>

Examples

```

if(require("psychTools")) {
  extraversion = psychTools::bfi[c("E1", "E2", "E3", "E4", "E5")]
  extraversion[, "E1"] = 7 - extraversion[, "E1"] # Reverse-coded item.
  extraversion[, "E2"] = 7 - extraversion[, "E2"] # Reverse-coded item.
  fit = conogive(extraversion)
}

```

message_cuts

Message Cuts to the Desired Shape

Description

Message Cuts to the Desired Shape

Usage

```
message_cuts(cuts, k)
```

Arguments

cuts	A matrix, list, or vector of cuts
k	Optional k saying how many times the vector of cuts should be repeated. Only matters when cuts is a vector.

predict.conogive

Predict Method for Conogive Objects

Description

Predict the latent variable in a congeneric normal-ogive model using the formula of ((arxiv ref.))

Usage

```

## S3 method for class 'conogive'
predict(object, newdata, weights = c("optimal", "equal"), ...)

```

Arguments

object	An object of class "conogive".
newdata	An optional data frame with variables to predict with. The fitted values are used if omitted.
weights	Weights to use; choose between optimal weights and equal weights.
...	Ignored.

Examples

```

if(require("psychTools")) {
  extraversion = psychTools::bfi[c("E1", "E2", "E3", "E4", "E5")]
  extraversion[, "E1"] = 7 - extraversion[, "E1"] # Reverse-coded item.
  extraversion[, "E2"] = 7 - extraversion[, "E2"] # Reverse-coded item.
  object = conogive(extraversion)
  hist(predict(object, extraversion)) # Plot distribution of predictions.
}

```

reliability

*Calculate the Ordinal Reliability***Description**

The function `ordinal_r` calculates the concrete ordinal reliability. The functions `theoretical_ordinal_r` and `theoretical_ordinal_alpha` calculates the theoretical ordinal reliability and alpha based on the polychoric correlation matrix.

Usage

```

ordinal_r(
  object,
  xi = c("sample", "theoretical"),
  weights = c("optimal", "equal")
)

theoretical_ordinal_r(object, weights = c("optimal", "equal", "sigma"))

theoretical_ordinal_alpha(object)

```

Arguments

<code>object</code>	An object of class <code>conogive</code> .
<code>xi</code>	How to calculate the Xi matrix. Option "theoretical" calculates the theoretical Xi matrix from rho, while "sample" calculates the sample Xi matrix.
<code>weights</code>	The weights used to calculate the ordinal reliability. Option "optimal" uses the optimal weights and "equal" the equal weights.

Details

The population value of theoretical ordinal alpha equals the theoretical ordinal reliability when the underlying multivariate normal is parallel. The concrete ordinal reliability is the squared correlation between the true latent variable and the best linear predictor of the observed Likert-type data. See ((ref)) for definitions.

Value

The concrete ordinal reliability, theoretical ordinal reliability, or theoretical ordinal alpha.

Examples

```
if(require("psychTools")) {  
  agreeableness = psychTools::bfi[c("A1", "A2", "A3", "A4", "A5")]  
  agreeableness[, "A1"] = 7 - agreeableness[, "A1"] # Reverse-coded item.  
  object = conogive(agreeableness)  
  ordinal_r(object, weights = "equal") # 0.6394087  
  theoretical_ordinal_alpha(object) # 0.7589922  
  theoretical_ordinal_r(object, weights = "equal") # 0.7689878  
  ordinal_r(object, weights = "optimal") # 0.6763742  
  theoretical_ordinal_r(object) # 0.8101108  
}
```

Index

conogive, [2](#)

massage_cuts, [3](#)

ordinal_r (reliability), [4](#)

predict.conogive, [3](#)

reliability, [4](#)

theoretical_ordinal_alpha
(reliability), [4](#)

theoretical_ordinal_r (reliability), [4](#)