Package: polecon (via r-universe)

November 19, 2026

Title Quantitative Methods for Political Economy

Version 0.1.0

Description Computes weighted rank-dependent concentration and inequality indices with explicit treatment of tied ranks. Implements the relative, generalized, modified, Wagstaff-normalized and Erreygers-normalized estimators described by O'Donnell et al. (2016) <doi:10.1177/1536867X1601600112>. A replicate-weight survey design can estimate sampling uncertainty while recomputing fractional ranks for every replicate. This is an independently restructured R implementation informed by the 'Stata' command distributed under the GPL-3 license.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Suggests survey, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Stefanos Stavrianos [aut, cre, cph], Owen O'Donnell [ctb, cph], Stephen O'Neill [ctb, cph], Tom Van Ourti [ctb, cph], Brendan Walsh [ctb, cph]

Maintainer Stefanos Stavrianos <stefanstavrianos@aua.gr>

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

Date/Publication 2026-11-18 21:40:02 UTC

RemoteUrl https://github.com/cran/polecon

RemoteRef HEAD

RemoteSha 19ddf7c875d637b96213f488ad824c9f9950db67

Contents

conindex 2
conindex_results 3

Index **5**

---

conindex *Weighted Concentration and Inequality Indices*

Description

Compute relative, generalized, modified, Wagstaff and Erreygers concentration indices with weighted mid-ranks. When `rankvar` is omitted, the outcome is also the ranking variable and the result is a Gini-type index.

Usage

```
conindex(data, varname, rankvar = NULL, weight = NULL,
         truezero = TRUE, limits = NULL, generalized = FALSE,
         bounded = FALSE, wagstaff = FALSE, erreygers = FALSE,
         svy = NULL, na_action = c("omit", "fail"))
```

Arguments

`data`	Data frame containing numeric variables. If `svy` is supplied, it must correspond row-for-row to the survey design.
`varname`	Name of the numeric outcome variable.
`rankvar`	Optional name of numeric ranking variable, omitted for Gini.
`weight`	Optional weight column name or numeric vector. Omit with `svy`.
`truezero`	If `FALSE` with a single lower limit, compute a modified index. Defaults to `TRUE`.
`limits`	Theoretical limits of the outcome: length two when `bounded=TRUE`, or one lower limit for a modified index.
`generalized`	Compute generalized index $2Cov_w(h, R)$.
`bounded`	Transform outcome to $(h - a)/(b - a)$; requires finite `limits=c(a,b)`.
`wagstaff`	Compute Wagstaff normalization of a bounded outcome.
`erreygers`	Compute Erreygers normalization of a bounded outcome.
`svy`	Optional `svyrep.design` from package **survey**; recomputes ranks and moments in every replicate.
`na_action`	Drop invalid observations (default) or fail; with `svy`, invalid observations always fail.

Details

The weighted mid-rank for all observations tied at ranking value y is the sum of the weights strictly below y, plus half the total weight of the tie group, divided by the sum of all weights. With R denoting that rank and μ the weighted outcome mean, the standard index is $C = 2Cov_w(h, R)/\mu$. Generalized is $G = 2Cov_w(h, R)$. For the normalized bounded outcome $z = (h - a)/(b - a)$, Erreygers is $E = 8Cov_w(z, R)$ and Wagstaff is $W = 2Cov_w(z, R)/(\mu_z(1 - \mu_z))$. For a modified index with lower limit a, use $M = 2Cov_w(h, R)/(\mu - a)$.

With survey replicate weights, the point estimate is recomputed for each replicate; reported standard errors are design-dependent and use the replicate construction supplied by the analyst. Confidence intervals and p-values use a normal approximation. Without a replicate-weight survey design, this function does not estimate standard errors. Theoretical bounds must be meaningful for normalized indices; sample-based scaling alone does not imply substantive absolute outcome limits.

Value

A `conindex` object containing `CI`, `CIse`, `p.value`, `N`, `Nunique`, `rank`, `type`, and concentration-curve coordinates in `curve`.

References

O'Donnell, O., O'Neill, S., Van Ourti, T. and Walsh, B. (2016). Conindex: Estimation of concentration indices. *The Stata Journal*, 16(1), 112–138. doi:10.1177/1536867X1601600112.

Examples

```
d <- data.frame(income = 1:4, outcome = c(1, 2, 3, 4), w = c(1, 1, 1, 1))
conindex(d, "outcome", "income", weight = "w")
conindex(d, "outcome", weight = "w")
b <- data.frame(income = 1:4, health = c(0, 0, 1, 1))
conindex(b, "health", "income", bounded = TRUE,
         limits = c(0, 1), erreygers = TRUE)
```

`conindex_results` *Inspect Concentration Index Results*

Description

Print an index, extract a one-row data frame, or obtain an approximate confidence interval if a replicate-weight standard error is available.

Usage

```
conindex_results(x)
## S3 method for class 'conindex'
print(x, ...)
## S3 method for class 'conindex'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

`x`	Object of class `conindex`.
`object`	Object of class `conindex`.
`parm`	Must be `NULL`.
`level`	Confidence level between zero and one.
`...`	Unused extra arguments for generic-method compatibility.

Value

`conindex_results` gives a one-row data frame; `print` returns the input invisibly; `confint` returns an interval matrix.

Examples

```
d <- data.frame(y = 1:4)
z <- conindex(d, "y")
conindex_results(z)
```

Index

confint.conindex *(*conindex_results*)*, 3
conindex, 2
conindex_results, 3

print.conindex *(*conindex_results*)*, 3