Package: ICEHmeasures (via r-universe)

October 20, 2026

Title The Equiplot Graph and Complex Inequality Measures

Version 2.1.0

Description Generates the equiplot, an iconic dot-plot graph for visualizing inequalities, as well as three complex inequality measures: the slope index of inequality, the concentration index and the mean absolute difference to the mean. For more details see World Health Organization (2013) <https://www.who.int/docs/default-source/gho-documents/health-equity/handbook-on-health-inequality-monitoring/handbook-on-health-inequality-monitoring.pdf>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports dplyr, tidyr, tibble, ggplot2 (>= 3.4.0), survey, rlang, grDevices, car, tidyselect

Suggests viridisLite

Depends R (>= 3.5)

LazyData true

NeedsCompilation no

Author Leonardo Ferreira [aut, cre], Luisa Arroyave [aut]

Maintainer Leonardo Ferreira <lferreira@equidade.org>

Config/pak/sysreqs cmake make libicu-dev

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

Date/Publication 2026-08-21 20:40:09 UTC

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

RemoteRef HEAD

RemoteSha 3f6c91383a20faa0961dd9eceebf2ba00a4ceb4e

Contents

cixr 2
equiplot 4
example_data 7
example_data2 7
iceh_palette 8
mad 9
siilogit 10

Index **13**

`cixr` *Concentration index (relative) and Erreygers corrected index*

Description

Calculates the relative concentration index, a relative measure of inequality expressed as the cumulative concentration of the outcome across the ranking variable distribution

Usage

```
cixr(
  data,
  rank_var,
  outcome_var,
  weight_var = NULL,
  cluster_var = NULL,
  corrected = FALSE,
  graph = FALSE,
  quant = NULL
)
```

Arguments

`data`	A data.frame or tibble containing the variables.
`rank_var`	Ranking variable (unquoted column name; e.g. `wealth`).
`outcome_var`	Outcome variable (unquoted column name).
`weight_var`	Optional weight variable (unquoted column name). If `NULL`, equal weights are assumed.
`cluster_var`	Optional cluster variable for variance estimation (unquoted column name). If provided, clustered standard errors are computed using `svydesign()` from the **survey** package.
`corrected`	Logical. If `TRUE`, computes Erreygers corrected index.
`graph`	Logical. If `TRUE`, draws the concentration curve.
`quant`	Optional integer containing the number of quantiles to use for plotting (grouped plot). If `NULL`, it attempts to find the optional number of groups

Details

The Concentration Index (CI) is a relative measure of socioeconomic inequality that quantifies the extent to which a health variable is unequally distributed across an ordered inequality dimension.

The CI is defined as twice the area between the concentration curve and the line of equality. It can equivalently be expressed as twice the covariance between the health variable and the fractional rank in the socioeconomic distribution, divided by the mean of the health variable. The CI ranges theoretically between -1 and 1.

This function is designed for use with ordered dimensions such as the wealth index. The ranking variable must represent a meaningful ordering from the most disadvantaged to the most advantaged.

The implementation follows the original formulation proposed by Wagstaff et al. (1991). Estimation is performed using a convenient regression-based approach as described by O'Donnell et al. (2008). The Erreygers correction is optionally available.

Interpretation: A positive CI indicates that the health variable is concentrated among more advantaged groups, while a negative CI indicates concentration among disadvantaged groups. A value of zero reflects no relative socioeconomic inequality. The magnitude reflects the degree of relative inequality across the entire distribution.

Value

A tibble with a single row and two columns: cix (concentration index) and cix_se (standard error). If `corrected=TRUE`, the tibble contains two additional columns: ccix (corrected concentration index), ccix_se (standard error).

References

Wagstaff A, Paci P, van Doorslaer E (1991). On the measurement of inequalities in health. Social Science & Medicine, 33(5), 545–557.

O'Donnell O, van Doorslaer E, Wagstaff A, Lindelow M (2008). Analyzing Health Equity Using Household Survey Data: A Guide to Techniques and Their Implementation. The World Bank.

Erreygers G (2009). Correcting the concentration index. Journal of Health Economics, 28(2), 504–515.

Examples

```
data(example_data)
cixr(
  data = example_data,
  rank_var = wic,
  outcome_var = stunt5,
  weight_var = sweight,
  cluster_var = cluster,
  graph = TRUE
)
```

equiplot *Equiplot Function*

Description

The function creates an equiplot graph to visualize disaggregated health indicator estimates across population subgroups defined by an inequality dimension.

Usage

```
equiplot(
  data,
  group_var,
  outcome_var = NULL,
  strat_var = NULL,
  wide = FALSE,
  palette = "wealth",
  order = c("alphabetical", "ascending", "descending"),
  order_ref = NULL,
  proportion = FALSE,
  xlim = NULL,
  point_size = 4,
  line_color = "black",
  xlab = "Outcome",
  ylab = "",
  legend_title = "Stratifier"
)
```

Arguments

data	A data.frame containing the data.
group_var	Unquoted column name for the grouping variable (y-axis).
outcome_var	Unquoted column name for the outcome variable (x-axis). Required when wide = FALSE; ignored when wide = TRUE.
strat_var	Unquoted stratifier column name(s) (color groups). When wide = FALSE, a single unquoted column name identifying the stratifier variable (e.g., strat_var = education). When wide = TRUE, one or more unquoted column names giving the stratifier columns, in the exact order they should be plotted and colored (e.g., strat_var = c(Poorest, Q2, Q3, Q4, Richest)).
wide	Logical. If TRUE, assumes each stratum is in a separate column.
palette	"wealth", "educ", "area", or "viridis".
order	"alphabetical", "ascending", or "descending".
order_ref	Specific stratum used to order groups.
proportion	Logical. If TRUE converts outcome from 0–1 proportions to percentages. Default = FALSE.

xlim	Numeric vector of x-axis limits.
point_size	Size of points.
line_color	Line color connecting strata.
xlab	X-axis label.
ylab	Y-axis label.
legend_title	Legend title.

Details

An equiplot is a graphical tool used to display disaggregated estimates of a health indicator across population subgroups defined by an inequality dimension (e.g., wealth quintile, education level, place of residence). It provides a clear visual representation of absolute differences between subgroups and facilitates the identification of inequality patterns.

It supports both **long** and **wide** data formats. In the wide format, the stratifier columns to be plotted must be specified explicitly via `strat_var`, listing the columns in the order they should appear (e.g., `strat_var = c(Poorest, Q2, Q3, Q4, Richest)`, from the most disadvantaged to the most advantaged group). In the long format, `strat_var` receives the name of the stratifier column.

The outcome scale is controlled through the `proportion` argument. When `proportion = TRUE`, outcomes expressed as proportions (0–1) are converted to percentages (0–100). The default (`proportion = FALSE`) keeps the original outcome scale, enabling use with proportions, percentages, rates, or counts.

Interpretation: Each point represents the outcome value for a specific subgroup of the stratifier variable (e.g., wealth quintiles, place of residence). The distance between points reflects the absolute inequality between these subgroups, the greater the distance, the larger the disparity. Equiplots facilitate visual comparison of inequality patterns across multiple groups simultaneously.

Value

A ggplot object representing an Equiplot. Because the function returns a standard ggplot object, users can further customize the Equiplot by adding layers and adjustments using the + operator (e.g., themes, scales, labels, or annotations).

Examples

```
# Example 1: 5 Wealth Quintiles, Wide Format, Sorted by "Poorest" Descending
# Goal: Highlight countries with the best results for their lowest quintile
# Values already expressed as percentages (0-100)

df_wealth <- data.frame(
  country = c("Angola", "Brazil", "Vietnam", "Peru", "Egypt"),
  Poorest = c(10, 20, 45, 15, 35),
  Q2 = c(25, 35, 55, 30, 45),
  Q3 = c(40, 50, 65, 45, 60),
  Q4 = c(60, 70, 80, 65, 75),
  Richest = c(85, 90, 95, 85, 92)
)

equiplot(
```

```
  df_wealth,
  country,
  strat_var = c(Poorest, Q2, Q3, Q4, Richest),
  wide = TRUE,
  palette = "wealth",
  order = "descending",
  order_ref = "Poorest",
  proportion = FALSE,
  xlab = "DTP3 Coverage (%)",
  legend_title = "Wealth Quintile"
)


# Example 2: Education Categories, Long Format
# Goal: Example using proportions (0–1) converted automatically to %

df_educ <- data.frame(
  country = rep(c("Zambia", "Bolivia", "Albania"), each = 3),
  education = rep(c("None", "Primary", "Secondary+"), 3),
  value = c(0.60, 0.75, 0.90,
            0.40, 0.60, 0.85,
            0.80, 0.85, 0.95)
)

equiplot(
  df_educ,
  country,
  value,
  education,
  palette = "educ",
  order = "alphabetical",
  proportion = TRUE,
  xlab = "Antenatal care 4+ visits (%)"
)


# Example 3: Urban vs Rural (Area), Wide Format
# Goal: Identify the lowest overall coverage
# Values already expressed as percentages

df_area <- data.frame(
  region = c("North", "South", "East", "West"),
  Rural = c(30, 55, 20, 45),
  Urban = c(60, 75, 50, 65)
)

equiplot(
  df_area,
  region,
  strat_var = c(Rural, Urban),
  wide = TRUE,
  palette = "area",
  order = "ascending",
```

```
  proportion = FALSE,
  xlab = "Outcome (%)",
  legend_title = "Residence"
)
```

`example_data` *Example dataset*

Description

A data frame containing individual level survey data used in the examples of the Slope Index of Inequality and the Concentration Index functions. Data represents children under 5-years

Usage

```
example_data
```

Format

A data frame with 500 rows and 5 variables:

sweight Survey weight

cluster Cluster ID

wiq Wealth quintile

wic Wealth index continuous score

stunt5 Child is stunted (1=yes, 0=no)

Source

Sampled from Bangladesh's 2022 DHS survey

`example_data2` *Example dataset 2*

Description

A data frame containing admin-1 level estimates from 3 Lao's surveys (2006, 2011 and 2017) used in the example of the Mean Absolute Difference

Usage

```
example_data2
```

Format

A data frame with 38 rows and 3 variables:

year Year of the survey

r Prevalence of stunting for each admin-1 unit for each year

r_mean Prevalence of stunting (at national level) for each year

Source

Provided by the ICEH Retriever

`iceh_palette` *ICEH Adaptive Color Palettes*

Description

ICEH Adaptive Color Palettes

Usage

```
iceh_palette(type = c("wealth", "educ", "area", "viridis"), n = NULL)
```

Arguments

`type`	character: "wealth", "educ", "area", or "viridis".
`n`	integer, optional. Number of colors to return. If `NULL`, the base palette is returned.

Details

When `type = "viridis"`, the palette is generated using `viridisLite::viridis()`. The package `viridisLite` must be installed to use this option.

Value

A character vector of hexadecimal color codes. If `n` is `NULL`, the function returns the base palette corresponding to the selected `type`. If `n` is specified, the function returns `n` interpolated colors generated from the base palette using `grDevices::colorRampPalette()`.

mad *Mean Absolute Difference*

Description

Computes the mean absolute difference from subgroup values to a specified reference value (typically the overall mean).

Usage

```
mad(
  data,
  outcome_var,
  reference_var = NULL,
  weight_var = NULL,
  groupby = NULL,
  na.rm = TRUE
)
```

Arguments

data	A data.frame or tibble containing the variables.
outcome_var	Outcome variable (unquoted column name).
reference_var	Optional reference variable. If NULL, the code uses the mean of the subgroups as the reference.
weight_var	Optional weight variable (unquoted column name). If NULL, equal weights are assumed.
groupby	Optional grouping variable. Use it if your data frame has multiple countries/years/indicators..
na.rm	Logical. If TRUE (default), rows with NA in the outcome or weight variable are dropped before any calculation. If FALSE, NA values are not removed and propagate to the result.

Details

The mean absolute difference (MAD) is defined as

$$MAD = \frac{1}{K} \sum_{k=1}^{K} |x_k - r|$$

where x_k denotes the outcome value for subgroup k, r the reference value, and K is the number of subgroups.

If a weight variable is supplied, a weighted version is computed:

$$MAD_w = \frac{\sum_{k=1}^{K} w_k |x_k - r|}{\sum_{k=1}^{K} w_k}$$

where w_k represents the subgroup weights.

If `reference_var` is `NULL`, the reference r is defined as the overall mean of the subgroup outcome values. Otherwise, the supplied reference variable is used.

The function is designed for grouped data where each row represents a subgroup. If `groupby` is specified, MAD is calculated separately within each group defined by that variable. (e.g., countries, years, indicators)

MAD quantifies dispersion relative to a reference and is expressed in the same units as the outcome variable.

Value

A tibble with the Mean Absolute Difference (MAD).

If `groupby` is provided, the tibble contains one row per group. Otherwise, it contains a single row with the overall MAD.

Examples

```
data(example_data2)
mad(
  data = example_data2,
  outcome_var = r,
  reference_var = r_mean
)
```

`siilogit` *SII and RII estimation using logistic regression*

Description

Calculates the slope index for inequality (SII), an absolute measure of inequality expressed as the difference between the extremes of the ranking variable. It can also compute the relative index of inequality (RII).

Usage

```
siilogit(
  data,
  rank_var,
  outcome_var,
  weight_var = NULL,
  cluster_var = NULL,
  rii = FALSE,
  graph = FALSE
)
```

Arguments

`data`	A data.frame or tibble containing the variables.
`rank_var`	Ranking variable (unquoted column name; e.g. `wealth`).
`outcome_var`	Outcome variable (unquoted column name).
`weight_var`	Optional weight variable (unquoted column name). If `NULL`, equal weights are assumed.
`cluster_var`	Optional cluster variable for variance estimation (unquoted column name). If provided, clustered standard errors are computed using `svydesign()` from the **survey** package.
`rii`	Logical. If `TRUE`, computes the Relative Index of Inequality (RII).
`graph`	Logical. If `TRUE`, draws a plot of the fitted model.

Details

The Slope Index of Inequality (SII) is an absolute measure of inequality that represents the difference in predicted coverage between the most advantaged and most disadvantaged individuals, based on the full distribution of an ordered inequality dimension (e.g., wealth quintiles).

This implementation is primarily designed for coverage indicators bounded between 0 and 1 (e.g., service utilization, intervention coverage). A logistic regression model is used to ensure predicted values remain within the (0, 1) range.

The function is intended for ordered dimensions such as wealth quintiles, education levels, or other ranked stratification variables.

Interpretation: A positive SII indicates higher coverage among more advantaged groups, while a negative SII indicates higher coverage among disadvantaged groups. An SII of zero reflects no absolute inequality. The magnitude represents the absolute percentage-point difference in predicted coverage between the extremes of the distribution of the inequality dimension.

Important assumption: The SII assumes a relatively linear relationship between the subgroups of the inequality dimension and the outcome of interest. If the pattern of coverage across subgroups is highly non-linear, the SII may not adequately summarize inequality.

Value

A tibble with a single row and two columns: sii (slope index of inequality) and sii_se (standard error). If `rii=TRUE`, the tibble contains two additional columns: rii (relative index of inequality), rii_se (standard error).

References

World Health Organization (2013). Handbook on Health Inequality Monitoring.

Examples

```
data(example_data)
siilogit(
  data = example_data,
  rank_var = wiq,
```

```
  outcome_var = stunt5,
  weight_var = sweight,
  cluster_var = cluster,
  rii = TRUE,
  graph = TRUE
)
```

Index

∗ **datasets**
example_data, 7
example_data2, 7

cixr, 2

equiplot, 4
example_data, 7
example_data2, 7

iceh_palette, 8

mad, 9

siilogit, 10