# inequality

Inequality measurement, decomposition, and poverty analysis in R

Charles Coverdale

*London, United Kingdom*

April 2026

# The question

**How should we compute the standard suite of inequality, poverty, and redistribution measures in a single reproducible R interface?**

*Eighteen functions, four families, one uniform `iq_` interface with survey weights as a first-class argument.*

# Why it matters

- **National statistical offices** publish annual Gini, Palma, and poverty tabulations
- **Fiscal policy analysts** measure Kakwani progressivity and Reynolds-Smolensky redistribution
- **Development economists** compute Ravallion-Chen growth incidence curves and FGT poverty
- **Health economists** quantify socioeconomic gradients via Erreygers-corrected concentration indices

# What is already out there

- **`ineq`**: Gini, Theil, Lorenz only. Last updated 2014. No survey weights, no CIs[1]
- **`IC2`**, **`dineq`**: decomposition-focused, narrow scope, inconsistent output schemas
- **`acid`**: archived on CRAN, no active maintenance
- **Private scripts**: widely-used but untested, uncited, unreproducible

The gap: **no single package covers indices, decomposition, poverty, and redistribution with survey weights**.

[1] Zeileis (2014), *ineq: Measuring Inequality, Concentration, and Poverty*, R package v0.2-13.

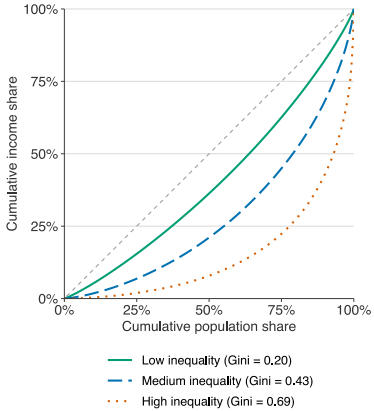
# What inequality offers

1. **Coverage**: 18 functions across 4 families (inequality indices, decomposition, poverty, fiscal redistribution)
2. **Interface**: uniform `iq_` prefix; optional `weights` argument on every function; S3 objects with print methods
3. **Provenance**: pure computation, no API calls, no network dependency, cited references in every help page

In the CRAN newbies queue as of April 2026. Davidson bootstrap CIs on the Gini[2].

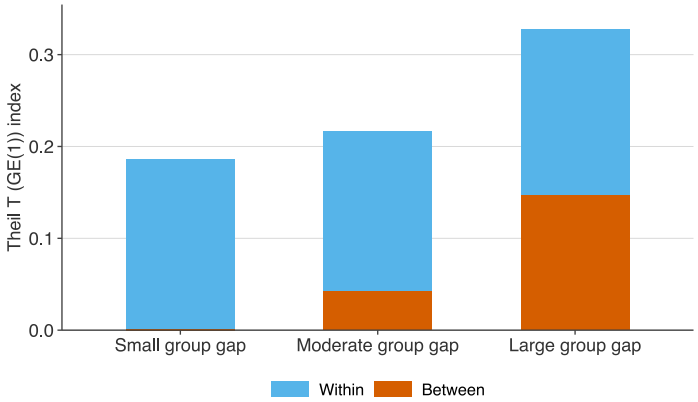
[2] Davidson (2009), *Reliable inference for the Gini index*, Journal of Econometrics 150(1).

# Indices: Lorenz curves and Gini



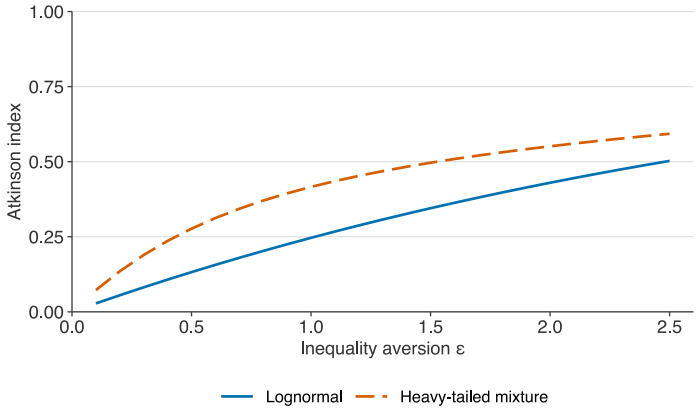

*Lorenz curves for three synthetic distributions. Exports:* `iq_gini()`, `iq_lorenz()`, `iq_sgini()`, `iq_palma()`, `iq_hoover()`, `iq_shares()`.

# Decomposition: Bourguignon between-within



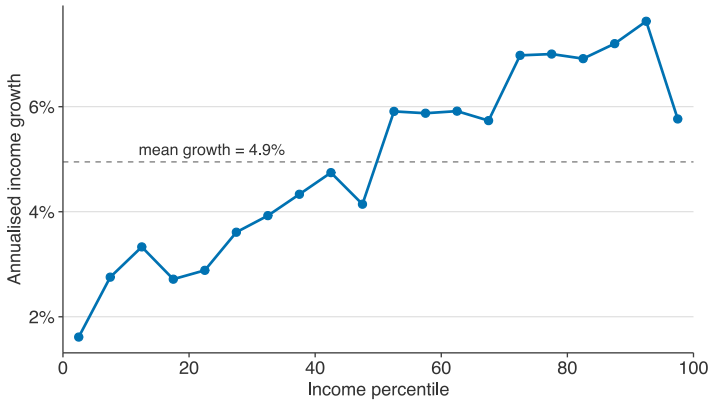

*Bourguignon (1979) between-within decomposition of Theil T by subgroup. Exports:* `iq_theil()`, `iq_decompose()`, `iq_concentration()`.

# Welfare: Atkinson over inequality aversion




*Atkinson index over inequality-aversion ε for two synthetic distributions. Exports:* `iq_atkinson()`, `iq_kolm()`, `iq_wolfson()`.

# Poverty: Ravallion-Chen growth incidence




*Ravallion-Chen (2003) growth incidence curve: per-percentile income growth between two periods. Exports:* `iq_fgt()`, `iq_sen()`, `iq_gic()`, `iq_kakwani()`.

# Central formulas

**Gini coefficient** (twice the area between Lorenz and the 45-degree line):

$$G = \frac{1}{2n^2\bar{x}} \sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j| \tag{1}$$

**Theil T index** (GE(1), additively decomposable; Theil 1967):

$$T = \frac{1}{n} \sum_{i=1}^{n} \frac{x_i}{\bar{x}} \ln\left(\frac{x_i}{\bar{x}}\right) \tag{2}$$

**Atkinson index** (welfare-grounded, aversion $\varepsilon$; Atkinson 1970):

$$A_\varepsilon = 1 - \frac{1}{\bar{x}} \left[ \frac{1}{n} \sum_{i=1}^{n} x_i^{1-\varepsilon} \right]^{\frac{1}{1-\varepsilon}} \tag{3}$$

# Package at a glance

Function families:

- **Inequality indices**: Gini, S-Gini, Theil (T, L), Atkinson, Kolm, Palma, Hoover, percentile ratios
- **Distribution**: Lorenz curves, income shares (bottom 50, top 10, top 1)
- **Decomposition**: Bourguignon between-within; Erreygers concentration
- **Poverty**: FGT ($\alpha = 0, 1, 2$), Sen, Watts; growth incidence
- **Redistribution**: Kakwani, Reynolds-Smolensky
- **Polarisation**: Wolfson

Deps: `cli`, `grDevices`, `graphics`, `stats`. R $\geq$ 4.1.0.

**Uniform interface**

Every function:

```
iq_*(x, weights, ...)
```

Returns an S3 object with a `print()` method. Weights are first-class.

# Minimal working example

```r
library(inequality)
d <- iq_sample_data("income")

# Three diagnostics on the same data, each weighted
g   <- iq_gini(d$income,      weights = d$weight, ci = TRUE)
dec <- iq_decompose(d$income, d$group, weights = d$weight)
pov <- iq_poverty(d$income,   line = 20000, weights = d$weight)

print(g); print(dec); print(pov)
```

Every function accepts `weights`, returns a printable S3 object, and exposes numeric components via `$`. Davidson bootstrap CIs available on `iq_gini()` via `ci = TRUE`.

# Six-country Gini trajectories, 1970 to 2024

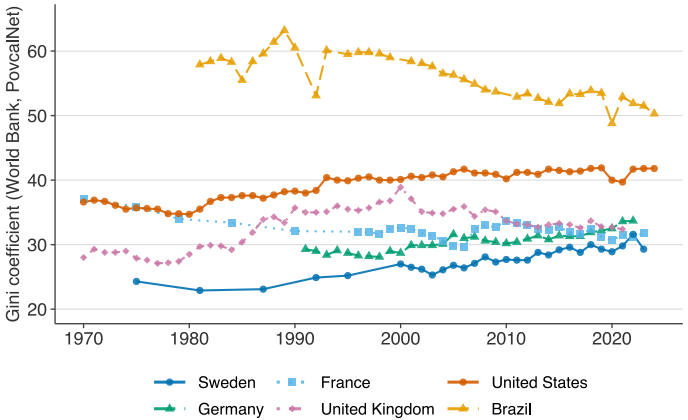
**Data.** World Bank Poverty and Inequality Platform, series `SI.POV.GINI`. 241 country-year observations across six economies: Sweden, Germany, France, the United Kingdom, the United States, Brazil.

**Question.** *Do liberal-market, continental European, and Scandinavian economies trace distinct inequality trajectories over five decades?*

**Why this case.** Longest continuous coverage on PIP. Tests the comparative-welfare-state hypothesis[3]. Runs through `iq_gini()` on each slice.

[3] Atkinson (2008), *The Changing Distribution of Earnings in OECD Countries*, OUP; Piketty (2014), *Capital in the Twenty-First Century*.

# Six-country Gini trajectories: the evidence




*Gini coefficient by country, 1970 to 2024. Source: World Bank* `SI.POV.GINI`.

# What inequality does not yet do

- **Measurement only**: no replicate-weight or Taylor-series variance for complex survey designs
- **CIs on Gini only**: Theil, Atkinson, Palma CIs via bootstrap not yet exposed as a uniform option
- **No top-coding correction**: Pareto-tail interpolation is upstream user responsibility
- **No DINA reconciliation**: national-accounts bridging for cross-country comparability is on the user

**v0.2.0 roadmap**: complex-survey variance estimators, CIs for the full index set, DINA-style helpers, Rcpp acceleration for panel-scale linked employer-employee data.

# Contact, code, paper

**Charles Coverdale**

charles.f.coverdale@gmail.com
github.com/charlescoverdale/inequality
charlescoverdale.github.io/publications

**Acknowledgements.** Data: World Bank Poverty and Inequality Platform (CC BY 4.0). Thanks to the distributional-economics community, CRAN maintainers, and reviewers of *inequality* v0.1.0.

*Scan for paper PDF*