AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search
http://ageconsearch.umn.edu
aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

Check for updates

The Stata Journal (2016)
16, Number 2, pp. 264–300

Assessing inequality using percentile shares

Ben Jann
University of Bern
Bern, Switzerland
ben.jann@soz.unibe.ch

Abstract. At least since Thomas Piketty's best-selling *Capital in the Twenty-First Century* (2014, Cambridge, MA: The Belknap Press), percentile shares have become a popular approach for analyzing distributional inequalities. In their work on the development of top incomes, Piketty and collaborators typically report top-percentage shares, using varying percentages as thresholds (top 10%, top 1%, top 0.1%, etc.). However, analysis of percentile shares at other positions in the distribution may also be of interest. In this article, I present a new command, `pshare`, that estimates percentile shares from individual-level data and displays the results using histograms or stacked bar charts.

Keywords: st0432, pshare, percentile shares, Lorenz curve, concentration curve, inequality, income distribution, wealth distribution, graphics

1 Introduction

Empirical inequality literature heavily relies on the Gini coefficient for the analysis of the development of inequality over time or the analysis of differences in inequality between countries. However, various distributional changes can increase or decrease the Gini coefficient, and it might be important to obtain more detailed knowledge about these processes. Moreover, even if the Gini coefficient remains stable, significant changes in the shape of a distribution may occur. In addition, the specific values of the Gini coefficient, apart from the minimum and the maximum, are difficult to interpret in an absolute sense.

For these reasons, percentile shares have become increasingly popular for analyzing distributional inequality. Percentile shares quantify the proportions of total outcome (for example, of total income) that go to different groups defined in terms of their relative ranks in the distribution. They thus have an intuitive and appealing interpretation and can be used for detailed analysis of distributional changes. The most prominent applications of percentile shares can be found in the works of Thomas Piketty and collaborators (for example, Atkinson, Piketty, and Saez [2011], Piketty and Saez [2014], and Piketty [2014]) and their "World Wealth and Income Database".[1] Piketty and collaborators typically study top-income shares, such as the proportion of income that goes to the top 1% or the top 10%, but the income shares of other percentile groups may be interesting too.

1. See http://topincomes.parisschoolofeconomics.eu/.

© 2016 StataCorp LP

In this article, I present a new command, `pshare`, that estimates percentile shares of an outcome variable from individual level data. `pshare` provides standard errors and confidence intervals (CIs) for the estimated percentile shares and supports estimation from complex samples. Furthermore, `pshare` provides subcommands for computing differences in percentile shares across variables or subpopulations and for graphing results as stacked bar charts or histograms.[2]

2 Methods and formulas

2.1 Lorenz ordinates

Let Y be the outcome variable of interest (for example, income). The distribution function of Y is given as $F(y) = \Pr(Y \leq y)$, and the quantile function (the inverse of the distribution function) is given as $Q(p) = F^{-1}(p) = \inf\{y|F(y) \geq p\}$ with $p \in [0, 1]$. Based on these definitions, the ordinates of the Lorenz curve are given as

$$L(p) = \frac{\int_{-\infty}^{Q_p} y\, dF(y)}{\int_{-\infty}^{\infty} y\, dF(y)}$$

Intuitively, a point on the Lorenz curve quantifies the proportion of total outcome of the poorest $p \times 100$ percent of the population. This can easily be seen in the finite population form of $L(p)$, which is given as

$$L(p) = \frac{\sum_{i=1}^{N} Y_i I_{Y_i \leq Q_p}}{\sum_{i=1}^{N} Y_i}$$

with I_A as an indicator function being equal to 1 if A is true and 0 otherwise.

2.2 Percentile shares

Percentile share $S(p_1, p_2)$, with $p_1 \leq p_2$, is equal to the proportion of total outcome that falls into the quantile interval $(Q_{p_1}, Q_{p_2}]$ or, stated differently, the proportion of total outcome pertaining to the population segment from relative rank p_1 to relative rank p_2 in the list of ordered outcomes. This is equal to the difference between the Lorenz ordinates for p_1 and p_2; that is,

$$S(p_1, p_2) = L(p_2) - L(p_1)$$

2. Some of the functionality of `pshare` is also covered by the user-written commands `sumdist` (Jenkins 1999) and `svylorenz` (Jenkins 2006). However, `pshare` specifically focuses on percentile shares and provides a more comprehensive implementation. Furthermore, `sumdist` and `svylorenz` use somewhat different methods to compute the percentile shares (ties are not broken, and flat regions in the distribution function are not interpolated; see below).

or, in the finite population,

$$S(p_1, p_2) = \frac{\sum_{i=1}^{N} Y_i I_{Y_i \leq Q_{p_2}}}{\sum_{i=1}^{N} Y_i} - \frac{\sum_{i=1}^{N} Y_i I_{Y_i \leq Q_{p_1}}}{\sum_{i=1}^{N} Y_i} = \frac{\sum_{i=1}^{N} Y_i \left(I_{Y_i \leq Q_{p_2}} - I_{Y_i \leq Q_{p_1}} \right)}{\sum_{i=1}^{N} Y_i}$$

To simplify notation, we will let $S_\ell^j = S(p_{\ell-1}, p_\ell)$. Furthermore, we will let

$$\mathbf{s}(\mathbf{p}) = \begin{bmatrix} S_1 & S_2 & \cdots & S_k \end{bmatrix}$$

be the $1 \times k$ vector of a disjunctive and exhaustive set of percentile shares across the domain of p using cutoffs $\mathbf{p} = \begin{bmatrix} p_0 & p_1 & \cdots & p_k \end{bmatrix}$ with $p_{\ell-1} < p_\ell$ for all $\ell = 0, \ldots, k$ and $p_0 = 0$ and $p_k = 1$.

Depending on context, it may be sensible to normalize percentile shares by the size of the respective population segment (that is, the proportion of the population covered by the segment, which is equal to $p_\ell - p_{\ell-1}$), which yields percentile share density

$$D_\ell = \frac{S_\ell^j}{p_\ell - p_{\ell-1}}$$

D_ℓ is a density in the sense that $\mathbf{d}(\mathbf{p})$—a disjunctive and exhaustive set of percentile share densities across the domain of p—integrates to 1. Note, however, that D_ℓ may be negative if the outcome variable can take on negative values (for example, debt). A value of $D_\ell = 1$ means that each member in the respective population segment has (on average) an outcome value equal to the average outcome in the population. A value of $D_\ell = 2$ means that each member in the segment has (on average) twice the population average; a value of $D_\ell = -0.5$ means that each member in the segment has (on average) minus half the population average.

Furthermore, percentile shares can be expressed as totals or averages in absolute terms. The finite population form of percentile share totals and averages is given as

$$T_\ell = \sum_{i=1}^{N} Y_i I_{Y_i \leq Q_{p_\ell}} - \sum_{i=1}^{N} Y_i I_{Y_i \leq Q_{p_{\ell-1}}} = \sum_{i=1}^{N} Y_i \left(I_{Y_i \leq Q_{p_\ell}} - I_{Y_i \leq Q_{p_{\ell-1}}} \right) = S_\ell^j \sum_{i=1}^{N} Y_i$$

and

$$A_\ell = \frac{T_\ell}{(p_\ell - p_{\ell-1}) \times N}$$

respectively. T_ℓ is simply the sum of all outcomes in the respective population segment; A_ℓ is the average outcome among the members of the segment.

Finally, with reference to the generalized Lorenz curve, generalized percentile shares can be defined as

$$G_\ell = \text{GL}(p_\ell) - \text{GL}(p_{\ell-1})$$

where the finite-population form of the generalized Lorenz ordinate $\text{GL}(p)$ is

$$\text{GL}(p) = \frac{1}{N}\sum_{i=1}^{N} Y_i I_{Y_i \leq Q_p}$$

so that

$$G_\ell = \frac{1}{N}\sum_{i=1}^{N} Y_i I_{Y_i \leq Q_{p_\ell}} - \frac{1}{N}\sum_{i=1}^{N} Y_i I_{Y_i \leq Q_{p_{\ell-1}}} = \frac{1}{N}\sum_{i=1}^{N} Y_i \left(I_{Y_i \leq Q_{p_\ell}} - I_{Y_i \leq Q_{p_{\ell-1}}} \right)$$

Note that there is an interesting relationship between percentile share averages and generalized percentile shares: percentile share average A_ℓ is equal to $G_\ell/(p_\ell - p_{\ell-1})$; that is, A_ℓ is equal to the difference in the generalized Lorenz ordinates for p_ℓ and $p_{\ell-1}$ divided by the population share $p_\ell - p_{\ell-1}$.

2.3 Point estimation

The above exposition assumes Y to be continuous. Because empirical data are always discrete, the empirical distribution function is nonsmooth, and there may be ties or empty sets at the quantiles of interest. For estimation of percentile shares using empirical data, it makes sense to break ties proportionally and apply linear interpolation in regions where the empirical distribution function is flat.

Let w_i be sampling weights (equal to 1 in unweighted data), and let subscripts in parentheses refer to sorted observations in ascending order of Y. S_ℓ^j can then be estimated from a sample of size n as

$$\widehat{S}_\ell^j = \widehat{L}(p_\ell) - \widehat{L}(p_{\ell-1})$$

with

$$\widehat{L}(p) = (1-\gamma)\widetilde{Y}_{j_p-1} + \gamma\widetilde{Y}_{j_p}$$

where

$$\gamma = \frac{p - \widehat{p}_{j_p-1}}{\widehat{p}_{j_p} - \widehat{p}_{j_p-1}}, \quad \widetilde{Y}_{j_p} = \frac{\sum_{i=1}^{j_p} w_{(i)} Y_{(i)}}{\sum_{i=1}^{n} w_i Y_i}, \quad \text{and} \quad \widehat{p}_{j_p} = \frac{\sum_{i=1}^{j_p} w_{(i)}}{\sum_{i=1}^{n} w_i}$$

and where j_p is set such that $\widehat{p}_{j_p-1} < p \leq \widehat{p}_{j_p}$. This corresponds to estimating Lorenz ordinates by taking quantiles from the running sum of the ordered Y values (divided by the total of Y) according to quantile definition 4 in Hyndman and Fan (1996).

Alternatively, ignoring linear interpolation in flat regions, $L(p)$ can be estimated as

$$\widehat{L}(p) = \widetilde{Y}_{j_p} = \frac{\sum_{i=1}^{j_p} w_{(i)} Y_{(i)}}{\sum_{i=1}^{n} w_i Y_i}$$

which corresponds to quantile definition 1 in Hyndman and Fan (1996).[3]

An estimate for D_ℓ is given as $\widehat{S}^j_\ell/(p_\ell - p_{\ell-1})$. For an estimate of T_ℓ, omit the denominator, $\sum_{i=1}^n w_i Y_i$, in the formula for $\widetilde{Y}_j$. An estimate for A_ℓ can be obtained as $\widehat{T}_\ell/\{(p_\ell - p_{\ell-1})\sum_{i=1}^n w_i\}$. For an estimate of G_ℓ, replace the denominator in the formula for $\widetilde{Y}_j$ by $\sum_{i=1}^n w_i$.

2.4 Variance estimation

An approximate variance matrix for $\widehat{\mathbf{s}}(\mathbf{p})$ can be obtained by using an estimating equations approach as outlined by Binder and Kovacevic (1995) (also see Kovačević and Binder [1997]). Let θ be the parameter of interest (a percentile share in our case), and let $\boldsymbol{\lambda}$ be a vector of nuisance parameters on which θ depends (the two quantiles determining the Lorenz ordinates in our case). According to Kovačević and Binder (1997), the sampling variance of $\widehat{\theta}$ can be approximated by the sampling variance of the total estimator

$$\sum_{i=1}^{n} w_i u_i^*$$

where w_i are sampling weights and u_i^* is the solution of

$$\left\{ -u_i^\theta + \frac{\partial U^\theta}{\partial \boldsymbol{\lambda}} \left(\frac{\partial \mathbf{U}^{\boldsymbol{\lambda}}}{\partial \boldsymbol{\lambda}} \right)^{-1} \mathbf{u}_i^{\boldsymbol{\lambda}} \right\} \left(\frac{\partial U^\theta}{\partial \theta} \right)^{-1}$$

with all unknowns in the final solution replaced by their sample counterparts. u_i^θ and $\mathbf{u}_i^{\boldsymbol{\lambda}}$ are estimating functions such that in the (finite) population, θ and $\boldsymbol{\lambda}$ are the solutions to

$$U^\theta = \sum_{i=1}^{N} u_i^\theta = 0 \quad \text{and} \quad \mathbf{U}^{\boldsymbol{\lambda}} = \sum_{i=1}^{N} \mathbf{u}_i^{\boldsymbol{\lambda}} = \mathbf{0}$$

3. The first approach is the default method in the `pshare` command presented below. The second approach ignoring linear interpolation can be requested by specifying the `step` option. Note that results from the second approach depend on the sort order within ties of Y if there are sampling weights. To enforce stable results in this case, the `pshare` command sorts observations in ascending order of the sampling weights among ties of Y, but this is an arbitrary decision.

In our case, $\theta = S^j_\ell$ and $\boldsymbol{\lambda} = \begin{bmatrix} Q^j_{p_\ell} & Q^j_{p_{\ell-1}} \end{bmatrix}$, where j refers to the analyzed subpopulation. Let $J_i = 1$ if observation i belongs to subpopulation j and $J_i = 0$ otherwise (with $J_i = 1$ for all observations if the entire sample is analyzed). Because

$$S^j_\ell = \frac{\sum_{i=1}^N Y_i \left(I_{y_i \leq Q^j_{p_\ell}} - I_{y_i \leq Q^j_{p_{\ell-1}}} \right) J_i}{\sum_{i=1}^N Y_i J_i} \qquad \text{and} \qquad \sum_{i=1}^N \left(I_{Y_i \leq Q^j_p} - p \right) J_i = 0$$

the estimating functions are

$$u^\theta_i = Y_i \left(I_{Y_i \leq Q^j_{p_\ell}} - I_{Y_i \leq Q^j_{p_{\ell-1}}} \right) J_i - Y_i J_i S^j_\ell \qquad \text{and} \qquad \mathbf{u}^{\boldsymbol{\lambda}}_i = \begin{bmatrix} \left(I_{Y_i \leq Q^j_{p_\ell}} - p_\ell \right) J_i \\ \left(I_{Y_i \leq Q^j_{p_{\ell-1}}} - p_{\ell-1} \right) J_i \end{bmatrix}$$

Furthermore, given these definitions,

$$\frac{\partial U^\theta}{\partial \theta} = -\sum_{i=1}^N Y_i J_i \qquad \text{and} \qquad \frac{\partial U^\theta}{\partial \boldsymbol{\lambda}} \left(\frac{\partial \mathbf{U}^{\boldsymbol{\lambda}}}{\partial \boldsymbol{\lambda}} \right)^{-1} = \begin{bmatrix} E(Y|Y = Q^j_{p_\ell}) \\ -E(Y|Y = Q^j_{p_{\ell-1}}) \end{bmatrix}'$$

Finally, because $E(Y|Y = Q_p) = Q_p$, we get

$$\begin{aligned} u^*_i &= \frac{-\left\{ Y_i \left(I_{Y_i \leq \widehat{Q}^j_{p_\ell}} - I_{Y_i \leq \widehat{Q}^j_{p_{\ell-1}}} \right) J_i - Y_i J_i \widehat{S}^j_\ell \right\}}{-\sum_{k=1}^n w_k Y_k J_k} \\ &\quad + \frac{\widehat{Q}^j_{p_\ell} \left(I_{Y_i \leq \widehat{Q}^j_{p_\ell}} - p_\ell \right) J_i - \widehat{Q}^j_{p_{\ell-1}} \left(I_{Y_i \leq \widehat{Q}^j_{p_{\ell-1}}} - p_{\ell-1} \right) J_i}{-\sum_{k=1}^n w_k Y_k J_k} \\ &= \frac{\left\{ \left(Y_i - \widehat{Q}^j_{p_\ell} \right) I_{Y_i \leq \widehat{Q}^j_{p_\ell}} - \left(Y_i - \widehat{Q}^j_{p_{\ell-1}} \right) I_{Y_i \leq \widehat{Q}^j_{p_{\ell-1}}} + p_\ell \widehat{Q}^j_{p_\ell} - p_{\ell-1} \widehat{Q}^j_{p_{\ell-1}} \right\} J_i}{\sum_{k=1}^n w_k Y_k J_k} \\ &\quad - \frac{Y_i J_i \widehat{S}^j_\ell}{\sum_{k=1}^n w_k Y_k J_k} \end{aligned}$$

The sampling variance of the total of u_i^*, which serves as an approximation of the sampling variance of $\widehat{S}_\ell^j$, can then be estimated using standard techniques as implemented in `total` (see [R] **total**), possibly accounting for complex survey design. The joint variance matrix for all elements of $\widehat{\mathbf{s}}(\mathbf{p})$ can be obtained by applying `total` to a series of appropriate u^* variables. Likewise, for joint estimation across several outcome variables or multiple subpopulations, include multiple series of u^* variables, one series for each outcome variable or subpopulation.[4]

Variance estimators for percentile densities, totals, averages, or generalized shares can be derived analogously. The appropriate u^* variables are obtained by replacing a_i and b in

$$u_i^* = \frac{\left\{\left(Y_i - \widehat{Q}^j_{p\ell}\right) I_{Y_i \leq \widehat{Q}^j_{p\ell}} - \left(Y_i - \widehat{Q}^j_{p\ell-1}\right) I_{Y_i \leq \widehat{Q}^j_{p\ell-1}} + p_\ell \widehat{Q}^j_{p\ell} - p_{\ell-1}\widehat{Q}^j_{p\ell-1}\right\} J_i - a_i}{b}$$

according to the overview in table 1, where n_c is the number of clusters and $\omega_{[i]}$ is the sum of weights in the cluster to which observation i belongs.[5]

4. When computing the u^* variables, the `pshare` command presented below uses definition 4 in Hyndman and Fan (1996) to determine $\widehat{Q}^j_p$ (or definition 1, depending on the method used for estimating the Lorenz ordinates). Furthermore, in analogy to the approach used for point estimation, ties in Y are broken when determining $I(Y_i \leq \widehat{Q}^j_p)$ (based on observations sorted by w_i within ties, which is an arbitrary decision to enforce stable results).
5. Depending on sample design, the denominator in a_i for T may require modification, for example, to take account of stratification. A workaround, followed by the `pshare` command presented below, is to simply set a_i to zero for T. This is a slight deviation from the approach outlined above (because u^* will sum to $\widehat{T}$ instead of zero), but the resulting variance estimates are the same in this case. On a related matter, note that `total` with clusters or weights yields different results than `svy: total` because the former assumes the number of observations or the sum of weights (and not the number of clusters) to be fixed. Likewise, `total` with the `over()` option produces different results than `svy: total`, even in the absence of clusters or weights, because the subgroup sizes are assumed fixed. Despite this disagreement, the `pshare` command presented below, which relies on the `total` command for variance estimation, always yields results that are consistent with `svy: total`, irrespective of whether weights and clusters are specified directly or via the `svy` option.

Table 1. Definitions of a_i and b for different types of percentile shares

	a_i	b
S	$Y_i J_i \widehat{S}^j_\ell$	$\sum_{i=1}^{n} w_i Y_i J_i$
D	$Y_i J_i (p_\ell - p_{\ell-1}) \widehat{D}^j_\ell$	$\sum_{i=1}^{n} w_i Y_i J_i (p_\ell - p_{\ell-1})$
T	$\frac{1}{n_c \omega_{[i]}} \widehat{T}^j_\ell$	1
A	$J_i (p_\ell - p_{\ell-1}) \widehat{A}^j_\ell$	$\sum_{i=1}^{n} w_i J_i (p_\ell - p_{1-\ell})$
G	$J_i \widehat{G}^j_\ell$	$\sum_{i=1}^{n} w_i J_i$

An alternative to the approach outlined above is to estimate the variances using the bootstrap or jackknife method (see [R] **bootstrap** and [R] **jackknife**).

2.5 Extensions

Contrasts

To analyze distributional differences among subpopulations or across time, one can compute contrasts between percentile shares. The most intuitive approach is to compute contrasts as arithmetic differences. For example, given percentile share estimates from two subpopulations (or two variables), A and B, the vector of arithmetic contrasts is

$$\widehat{\mathbf{s}}^A(\mathbf{p}) - \widehat{\mathbf{s}}^B(\mathbf{p})$$

with variance matrix

$$\begin{bmatrix}\mathbf{I}_k & -\mathbf{I}_k\end{bmatrix} \widehat{\mathbf{V}}\left\{\begin{bmatrix}\widehat{\mathbf{s}}^A(\mathbf{p}) & \widehat{\mathbf{s}}^B(\mathbf{p})\end{bmatrix}\right\} \begin{bmatrix}\mathbf{I}_k & -\mathbf{I}_k\end{bmatrix}'$$

where $\mathbf{I}_k$ is the identity matrix of size k and $\widehat{\mathbf{V}}\{\dots\}$ is the joint variance matrix of the percentile shares across both subpopulations (or variables).

Alternatively, contrasts could be computed as ratios or logarithms of ratios. Generally, let

$$\begin{bmatrix} c\left(\widehat{S}^A_1, \widehat{S}^B_1\right) & c\left(\widehat{S}^A_2, \widehat{S}^B_2\right) & \cdots & c\left(\widehat{S}^A_k, \widehat{S}^B_k\right)\end{bmatrix}$$

be the vector of percentile share contrasts between subpopulations (or variables) A and B, with $c(a, b)$ as a function of a and b, such as $c(a, b) = a/b$ (ratio) or $c(a, b) = \ln(a/b)$ (logarithm of ratio). The variance matrix of the vector can then be approximated by the delta method as

$$\mathbf{\Delta}\widehat{\mathbf{V}}\left\{\begin{bmatrix}\widehat{\mathbf{s}}^A(\mathbf{p}) & \widehat{\mathbf{s}}^B(\mathbf{p})\end{bmatrix}\right\}\mathbf{\Delta}'$$

where $\boldsymbol{\Delta}$ is a $k \times 2k$ matrix

$$\begin{bmatrix} \frac{\partial c(\widehat{S}_1^A, \widehat{S}_1^B)}{\partial \widehat{S}_1^A} & 0 & \cdots & 0 & \frac{\partial c(\widehat{S}_1^A, \widehat{S}_1^B)}{\partial \widehat{S}_1^B} & 0 & \cdots & 0 \\ 0 & \frac{\partial c(\widehat{S}_2^A, \widehat{S}_2^B)}{\partial \widehat{S}_2^A} & \cdots & 0 & 0 & \frac{\partial c(\widehat{S}_2^A, \widehat{S}_2^B)}{\partial \widehat{S}_2^B} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial c(\widehat{S}_k^A, \widehat{S}_k^B)}{\partial \widehat{S}_k^A} & 0 & 0 & \cdots & \frac{\partial c(\widehat{S}_k^A, \widehat{S}_k^B)}{\partial \widehat{S}_k^A} \end{bmatrix}$$

In Stata, the `nlcom` command can be used to perform the necessary computations. The derivatives in $\boldsymbol{\Delta}$ are determined numerically by `nlcom` (see [R] **nlcom**).

Renormalization

Percentile shares expressed as proportions or densities are normalized with respect to the total of the analyzed outcome variable in the given (sub)population. Depending on context, it may be sensible to use a different total for normalization. For example, when analyzing different subpopulations, we may want to express results in terms of proportions of the grand total across all subpopulations. Likewise, if analyzing, say, labor income, we may want to express results in terms of total income (labor income plus capital income).

To normalize results to a different total, simply replace denominator $\sum_{i=1}^n w_i Y_i$ in the above percentile share estimators with the appropriate total. For example, to normalize to the total of variable Z instead of the total of variable Y (where Z may be the sum of several variables, possibly including Y), use $\sum_{i=1}^n w_i Z_i$ as the denominator. Similarly, if normalizing percentile shares to the total of a reference (sub)population r instead of subpopulation j, replace the standard denominator $\sum_{i=1}^n w_i Y_i J_i$ with $\sum_{i=1}^n w_i Y_i R_i$, where J_i and R_i are indicators for whether observation i belongs to subpopulation j or r, respectively. When normalizing percentile densities to the total of a reference (sub)population, you need to consider the relative group sizes so that the densities reflect multiples of the average outcome in the reference (sub)population. That is, use

$$\widehat{D}_\ell^{jr} = \frac{\widehat{S}_\ell^{jr}}{(p_\ell - p_{\ell-1})\widehat{P}^{jr}} \qquad \text{with} \qquad \widehat{P}^{jr} = \frac{\sum_{i=1}^n w_i J_i}{\sum_{i=1}^n w_i R_i}$$

to compute the percentile density in subpopulation j with respect to the total of subpopulation r.

For variance estimation, several cases have to be distinguished: 1) normalizing to the total of Z, 2) normalizing to a fixed total τ, 3) normalizing to the total of Y in reference population r, 4) normalizing to the total of Z in reference population r, and 5) normalizing to a fixed total τ in reference population r. In general, when one normalizes densities with respect to a reference population (cases 3 to 5), the relative group size is a further nuisance parameter that has to be considered. Solving the equations for the

different cases leads to the expressions for a_i and b as shown in table 2 (see the section on variance estimation above for background).[6]

Table 2. Definitions of a_i and b for renormalized percentile shares

		a_i	b
(1)	S	$Z_i J_i \widehat{S}^j_\ell$	$\sum_i w_i Z_i J_i$
	D	$Z_i J_i (p_\ell - p_{\ell-1}) \widehat{D}^j_\ell$	$\sum_i w_i Z_i J_i (p_\ell - p_{\ell-1})$
(2)	S	$\frac{\tau}{n_c \omega_{[i]}} \widehat{S}^j_\ell$	τ
	D	$\frac{\tau}{n_c \omega_{[i]}} (p_\ell - p_{\ell-1}) \widehat{D}^j_\ell$	$\tau (p_\ell - p_{\ell-1})$
(3)	S	$Y_i R_i \widehat{S}^{jr}_\ell$	$\sum_i w_i Y_i R_i$
	D	$\left(Y_i R_i - \frac{\sum_k w_k Y_k R_k}{\sum_k w_k R_k} R_i + \frac{\sum_k w_k Y_k R_k}{\sum_k w_k J_k} J_i \right) \times (p_\ell - p_{\ell-1}) \widehat{P}^{jr} \widehat{D}^{jr}_\ell$	$\sum_i w_i Y_i R_i (p_\ell - p_{\ell-1}) \widehat{P}^{jr}$
(4)	S	Like (3), but with all instances of Y replaced by Z	
	D		
(5)	S	$\frac{\tau}{n_c \omega_{[i]}} \widehat{S}^{jr}_\ell$	τ
	D	$\left(\frac{\tau}{n_c \omega_{[i]}} - \frac{\tau}{\sum_k w_k R_k} R_i + \frac{\tau}{\sum_k w_k J_k} J_i \right) \times (p_\ell - p_{\ell-1}) \widehat{P}^{jr} \widehat{D}^{jr}_\ell$	$\tau (p_\ell - p_{\ell-1}) \widehat{P}^{jr}$

(All sums are across the entire sample.)

Concentration shares

A further interesting possibility is to determine the relative ranks of the population members using an alternative outcome variable. By default, observations will be ordered by their Y values. However, we may also order observations by some alternative variable Z. The (finite-population) Lorenz ordinates are then defined as

$$L^Z(p) = \frac{\sum_{i=1}^{N} Y_i I_{Z_i \leq Q^Z_p}}{\sum_{i=1}^{N} Y_i}$$

6. Depending on sample design, expression $\tau/(n_c \omega_{[i]})$ in a_i for cases (2) and (5) may require modification. An alternative, however, is to simply set $\tau/(n_c \omega_{[i]})$ to zero. See footnote 5 above.

and the percentile shares reflect the proportion of total Y that is received by different percentile groups of Z (in this case, the Lorenz curve is called a concentration curve; see Kakwani [1977] and Lambert [2001]). For example, this could be used to analyze how taxes (Y) are distributed across income groups (Z).

For the purpose of estimation, it appears sensible to average Y within ties of Z when computing the concentration curve ordinates so that results are independent of the sort order of the observations. Furthermore, for variance estimation, we need to replace $\widehat{Q}_p$ in the formulas for the u^* variables with $\widehat{E}(Y|Z = Q_p^Z)$, the expected value of Y at the p quantile of Z.[7]

3 The pshare command

Four subcommands are provided. `pshare estimate` computes the percentile shares and their variance matrix; `pshare contrast` computes differences in percentile shares between outcome variables or subpopulations based on the results from `pshare estimate`; `pshare stack` draws a stacked bar chart of the results from `pshare estimate`, and `pshare histogram` draws a histogram of the results from `pshare estimate` or `pshare contrast`.

3.1 Syntax of pshare estimate

The syntax of `pshare estimate` is

```
pshare [estimate] varlist [if] [in] [weight] [,
    {proportion|percent|density|sum|average|generalized} normalize(spec)
    gini {nquantiles(#)|percentiles(numlist)} pvar(pvar) step
    over(varname) total contrast[(spec)] stack[(options)]
    histogram[(options)] vce(vcetype) cluster(clustvar) svy[(subpop)] nose
    level(#) noheader notable nogtable display_options]
```

`pweights`, `iweights`, and `fweights` are allowed; see [U] **11.1.6 weight**. For each specified variable, percentile shares (quintile shares by default) are tabulated along with their standard errors and CIs.[8] If the `over()` option is specified (see below), only one variable is allowed in *varlist*. `pshare` assumes subcommand `estimate` as the default; typing the word "`estimate`" is required only in the case of a name conflict between the first element of *varlist* and the other subcommands of `pshare` (see below). Options are as follows.

7. In the `pshare` command presented below, $E(Y|Z = Q_p^Z)$ is estimated by local linear regression using the Epanechnikov kernel and the default rule-of-thumb bandwidth as described in [R] **lpoly**.
8. Variance estimation is not supported for `iweights` and `fweights`. To compute standard errors and CIs in the case of `fweights`, apply `pshare` to the expanded data (see [D] **expand**).

Main

`proportion`, `percent`, `density`, `sum`, `average`, or `generalized` selects the type of results to be computed. The default is `proportion`, that is, to report percentile shares as proportions. Use the `percent` option to report percentile shares as percentages. Furthermore, use the `density` option to report densities, defined as outcome shares divided by population shares (so that in a bar chart, the areas of the bars are proportional to the outcome shares). Outcome sums (totals) and average outcomes can be requested by the `sum` and `average` options, respectively. Finally, use the `generalized` option to report generalized percentile shares, defined as differences between generalized Lorenz ordinates. Only one of `proportion`, `percent`, `density`, `sum`, `average`, or `generalized` is allowed.

`normalize(`*spec*`)` normalizes results with respect to the specified total (not allowed in combination with `sum`, `average`, or `generalized`). *spec* is

[*over*:] [*total*]

where *over* may be

`.`	the subpopulation at hand (the default)
#	the subpopulation identified by value *#*
`#`*#*	the *#*th subpopulation
`total`	the total across all subpopulations

and *total* may be

`.`	the total of the variable at hand (the default)
`*`	the total of the sum across all analyzed outcome variables
varlist	the total of the sum across the variables in *varlist*
#	a total equal to *#*

total specifies the variables from which the total is to be computed or sets the total to a fixed value. If multiple variables are specified, the total across all specified variables is used (*varlist* may contain external variables that are not among the list of analyzed outcome variables). *over* selects the reference population from which the total is to be computed; *over* is allowed only if the `over()` option has been specified (see below). Subpopulation sizes (sum of weights) are considered for the computation of densities (the `density` option) if *over* is provided so that the densities reflect multiples of the average outcome in the reference population.

`gini` reports the Gini coefficients of the distributions (also known as concentration indices if `pvar()` is specified; see below) to be computed and reported in a separate table. Variance estimation for Gini coefficients is not supported.[9]

9. Following Lerman and Yitzhaki (1989), the concentration index of Y with respect to Z is computed as $C = 2\sum_{i=1}^{n} \widetilde{w}_i(Y_i - \overline{Y})(F_i - \overline{F})/\overline{Y}$, where $\widetilde{w}_i = w_i/\sum_{i=1}^{n} w_i$ are normalized weights, $\overline{Y} = \sum_{i=1}^{n} \widetilde{w}_i Y_i$ is the mean of Y, $\overline{F} = \sum_{i=1}^{n} \widetilde{w}_i F_i$ is the mean of F, and $F_i = \sum_{j=1}^{n} \widetilde{w}_j I_{Z_j \leq Z_i} - \sum_{j=1}^{n} \widetilde{w}_i I_{Z_j = Z_i}/2$ is the midinterval relative rank of Z_i in the empirical distribution of Z. For the Gini coefficient of Y, set $Z = Y$.

Percentiles

`nquantiles(#)` specifies the number of (equally sized) percentile groups to be used or `percentiles(`*numlist*`)` to specify a list of percentile cutoffs. The default is `nquantiles(5)`, which corresponds to `percentiles(20 40 60 80)` or, with shorthand as described in [U] **11.1.8 numlist**, `percentiles(20(20)80)`.

`pvar(`*pvar*`)` causes the percentile groups to be based on variable *pvar* instead of the outcome variable. That is, observations will be sorted in increasing order of *pvar*, and percentiles will be determined from the running sum of the outcome variable across this sort order (using averaged values within ties of *pvar*). Use this option to analyze relations between different variables (for example, how wealth is distributed across different income groups). If `pvar()` is specified, the computed percentile shares correspond to differences between ordinates of the "concentration curve" of the outcome variable with respect to *pvar*.

`step` determines the Lorenz ordinates from the step function of cumulative outcomes. The default is to use linear interpolation in regions where the step function is flat.

Over

`over(`*varname*`)` repeats results for each subpopulation defined by the values of *varname*. Only one outcome variable is allowed if `over()` is specified.

`total` reports additional overall results across all subpopulations. `total` is allowed only if `over()` is specified.

Contrast/Graph

`contrast`[`(`*spec*`)`] computes differences in percentile shares between outcome variables or between subpopulations. *spec* is

[*base*] [`, ratio lnratio`]

where *base* is the name of the outcome variable or the value of the subpopulation to be used as base for the contrasts. If *base* is omitted, adjacent contrasts across outcome variables or subpopulations are computed (or contrasts with respect to the total if total results across subpopulations have been requested).

Use the suboption `ratio` to compute contrasts as ratios or the suboption `lnratio` to compute contrasts as logarithms of ratios. The default is to compute contrasts as differences.

`stack`[`(`*options*`)`] draws a stacked bar chart of the results. *options* are as described for `pshare stack` below.

`histogram`[`(`*options*`)`] draws a histogram of the results. *options* are as described for `pshare histogram` below.

SE/SVY

`vce(`*vcetype*`)` determines how standard errors and CIs are computed. *vcetype* may be

`analytic`
`cluster` *clustvar*
`bootstrap` [`,` *bootstrap_options*]
`jackknife` [`,` *jackknife_options*]

The default is `vce(analytic)`. See [R] **bootstrap** and [R] **jackknife** for *bootstrap_options* and *jackknife_options*, respectively.

`cluster(`*clustvar*`)` is a synonym for `vce(cluster` *clustvar*`)`.

`svy`[`(`*subpop*`)`] causes the survey design to be taken into account for variance estimation; see [SVY] **svyset**. Specify *subpop* to restrict survey estimation to a subpopulation, where *subpop* is

[*varname*] [*if*]

The subpopulation is defined by observations for which *varname* $\neq 0$ and for which the `if` condition is met. See [SVY] **subpopulation estimation** for more information on subpopulation estimation.

The `svy` option is allowed only if the variance estimation method set by `svyset` is Taylor linearization (the default). For other variance estimation methods, the usual `svy` prefix command may be used; see [SVY] **svy**. For example, type "`svy brr: pshare` ..." to use balanced repeated-replication variance estimation. `pshare` does not allow the `svy` prefix for Taylor linearization because of technical reasons. This is why the `svy` option is provided.

`nose` suppresses the computation of standard errors and CIs. Use the `nose` option to speed up computations when analyzing census data. The `nose` option may also be useful to speed up computations when using a prefix command that uses replication techniques for variance estimation, such as [SVY] **svy jackknife**. The `vce(bootstrap)` and `vce(jackknife)` options imply `nose`.

Reporting

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is `level(95)` or as set by `set level`.

`noheader` suppresses the output header; only the coefficient table is displayed.

`notable` suppresses the coefficient table.

`nogtable` suppresses the table containing Gini coefficients.

display_options are standard reporting options such as `cformat()`, `pformat()`, `sformat()`, or `coeflegend`; see [R] **estimation options**.

3.2 Syntax of pshare contrast

`pshare contrast` computes differences in percentile shares between outcome variables or subpopulations. It requires results from `pshare estimate` to be in memory, which will be replaced by the results from `pshare contrast`.[10] The syntax is

`pshare contrast` [*base*] [`, ratio lnratio stack`[`(`*options*`)`] `histogram`[`(`*options*`)`] *display_options*]

where *base* is the name of the outcome variable or the value of the subpopulation to be used as base for the contrasts. If *base* is omitted, `pshare contrast` computes adjacent contrasts across outcome variables or subpopulations (or contrasts with respect to the total if total results across subpopulations have been requested). Options are as follows:

`ratio` causes contrasts to be reported as ratios. The default is to report contrasts as differences.

`lnratio` causes contrasts to be reported as logarithms of ratios. The default is to report contrasts as differences.

`stack`[`(`*options*`)`] draws a stacked bar chart of the results. *options* are as described for `pshare stack` below.

`histogram`[`(`*options*`)`] draws a histogram of the results. *options* are as described for `pshare histogram` below.

display_options are standard reporting options such as `cformat()`, `pformat()`, `sformat()`, or `coeflegend`; see [R] **estimation options**.

10. Alternatively, to compute the contrasts directly, you may apply the `contrast()` option to `pshare estimate` (see above).

3.3 Syntax of pshare stack

`pshare stack` draws a stacked bar chart of percentile shares. It requires results from `pshare estimate` to be in memory.[11] The syntax is

`pshare stack` [`,` {`vertical`|`horizontal`} `proportion` `reverse` `keep(`*list*`)`
`sort`[`(`*options*`)`] `gini(%`*fmt*`)` `nogini` `labels("`*label1*`" "`*label2*`" ...)`
`plabels("`*label1*`" "`*label2*`" ...)` `barwidth(`*#*`)` *barlook_options*
`p`*#*`(`*barlook_options*`)` `values`[`(%`*fmt*`)`] *marker_label_options* `addplot(`*plot*`)`
twoway_options]

Options are as follows.

Main

`vertical` or `horizontal` specifies whether a vertical or a horizontal bar plot is drawn; the default is `horizontal`.

`proportion` scales the population axis as a proportion (0 to 1). The default is to scale the axis as a percentage (0 to 100).

`reverse` orders percentile groups from top to bottom (the richest are leftmost, the poorest are rightmost). The default is to order percentile groups from bottom to top (the poorest are leftmost, the richest are rightmost).

`keep(`*list*`)` selects and orders the results to be included as separate bars. Use `keep()` with multiple outcome variables or subpopulations. For multiple outcome variables, *list* is a list of the names of the outcome variables to be included. When `over()` was specified in `pshare estimate`, *list* is a list of the values of the subpopulations to be included. *list* may also contain `total` for the overall results (if overall results were requested). Furthermore, *list* may also contain elements such as `#1`, `#2`, `#3`, etc., to refer to the 1st, 2nd, 3rd, etc., outcome variable or subpopulation.

`sort`[`(`*options*`)`] orders the bars for the different outcome variables or subpopulations by the level of inequality, where *options* are `gini` to sort by Gini coefficients (if Gini coefficients have been computed), `descending` to sort in descending order, and `tfirst` or `tlast` to place the total across subpopulations first or last, respectively. The default is to sort in ascending order of the shares of the top percentile group.

`gini(%`*fmt*`)` sets the format for the Gini coefficients included in the graph as secondary axis labels; see [D] **format**. The default is `gini(%9.3g)`. Gini coefficients will be included only if information on Gini coefficients is available in the provided results (that is, if the `gini` option has been applied to `pshare estimate`).

11. You may also draw a chart directly by applying the `stack()` option to `pshare estimate` or `pshare contrast` (see above).

`nogini` suppresses the Gini coefficients. This is relevant only if the `gini` option has been specified when calling `pshare estimate`.

Labels/rendering

`labels("`*label1*`" "`*label2*`" ...)` specifies custom axis labels for the outcome variables or subpopulations.

`plabels("`*label1*`" "`*label2*`" ...)` specifies custom legend labels for the bar segments (that is, the percentile groups).

`barwidth(`#`)` sets the width of the bars as proportion of the spacing between bar positions. The default is `barwidth(0.75)`, leaving white space of 1/3 barwidth between the bars.

barlook_options and `p`#`(`*barlook_options*`)` affect the rendition of the plotted bars, where `p`#`()` applies to the #th segment (the #th percentile group) of the stacked bars; see [G-3] ***barlook_options***.

`values`[`(%`*fmt*`)`] prints the values of the percentile shares as marker labels at the center of the bar segments. The default is `values(%9.3g)`; see [D] **format**.

marker_label_options affect the rendition of the values included as marker labels using the `values()` option; see [G-3] ***marker_label_options***. Do not use `mlabel()` or `mlabvposition()`.

Add plots

`addplot(`*plot*`)` adds other plots to the generated graph; see [G-3] ***addplot_option***.

Y axis, X axis, Title, Caption, Legend, Overall

twoway_options are general twoway options, other than `by()`, as documented in [G-3] ***twoway_options***.

3.4 Syntax of pshare histogram

`pshare histogram` draws a histogram of percentile shares or percentile share contrasts. It requires results from `pshare estimate` or `pshare contrast` to be in memory.[12] The syntax is

`pshare histogram` [`,` {`vertical`|`horizontal`} `proportion` `keep(`*list*`)`
`max(`#[`,` *options*]`)` `min(`#[`,` *options*]`)` `prange(`*min max*`)` `gini(%`*fmt*`)` `nogini`
barlook_options `step` `spikes`[`(`#`)`] `labels("`*label1*`" "`*label2*`" ...)`
`byopts(`*byopts*`)` `overlay` `o`#`(`*options*`)` `psep`[`("`*label1*`" "`*label2*`" ...)`]
`p`#`(`*options*`)` `level(`#`)` `ci(`*citype*`)` `ciopts(`*options*`)` `cibelow` `noci`
`addplot(`*plot*`)` *twoway_options*]

Options are as follows.

Main

`vertical` or `horizontal` specifies whether a vertical or a horizontal plot is drawn. The default is to draw a vertical bar plot.

`proportion` scales the population axis as a proportion (0 to 1). The default is to scale the axis as a percentage (0 to 100).

`keep(`*list*`)` selects and orders the results to be included as separate subgraphs, where *list* is a list of the names of the outcome variables or the values of the subpopulations to be included. *list* may also contain `total` for the overall results if overall results were requested. Furthermore, you may use elements such as `#1`, `#2`, `#3`, ... to refer to the 1st, 2nd, 3rd, ... outcome variable or subpopulation.

`max(`#[`,` *options*]`)` top-codes results at # and `min(`#[`,` *options*]`)` bottom-codes results at #. This is useful if there are large differences in the plotted values and you want to restrict the axis range. The truncated values will be included in the graph as marker labels. *options* are `format(%`*fmt*`)` to set the format for the marker labels (default is `format(%9.3g)`; see [D] **format**), *marker_label_options* to affect the rendition of the marker labels (see [G-3] ***marker_label_options***), and `nolabels` to omit the marker labels.

`prange(`*min max*`)` restricts the range of percentile groups to be included in the graph. Only results for percentile groups whose lower and upper cumulative population bounds (in percent) are within *min* and *max* will be plotted. *min* and *max* must be within $[0, 100]$. For example, to include only the lower half of the distribution, type `prange(0 50)`.

12. You may also draw the histogram directly by applying the `histogram()` option to `pshare estimate` or `pshare contrast` (see above).

`gini(%`*fmt*`)` sets the format for the Gini coefficients included in the subgraph labels; see [D] **format**. The default is `gini(%9.3g)`. Gini coefficients will be included only if information on Gini coefficients is available in the provided results (that is, if the `gini` option has been applied to `pshare estimate`).

`nogini` suppresses the Gini coefficients. This is relevant only if the `gini` option has been specified when calling `pshare estimate`.

Labels/rendering

barlook_options affect the rendition of the plotted bars; see [G-3] ***barlook_options***.

`step` uses a step function (line plot) instead of a histogram to draw the results. Use *line_options* instead of *barlook_options* to affect the rendition of the plotted line; see [G-3] ***line_options***. `step` may be included in `o#()`, if `overlay` has been specified, to apply `step` to selected outcome variables or subpopulations (see below).

`spikes`[`(#)`] uses (equally spaced) spikes instead of histogram bars to draw the results. # specifies the number of spikes, and the default is `spikes(100)`. Use *line_options* instead of *barlook_options* to affect the rendition of the plotted spikes; see [G-3] ***line_options***. Confidence intervals will be omitted.

`labels("`*label1*`" "`*label2*`" ...)` specifies custom labels for the subgraphs of the outcome variables or subpopulations.

`byopts(`*byopts*`)` determines how subgraphs are combined; see [G-3] ***by_option***.

`overlay` includes results from multiple outcome variables or subpopulations in the same plot instead of creating subgraphs. `overlay` and `psep()` are not both allowed. Specifying `overlay` implies `noci`.

`o#(`*options*`)` affects the rendition of the bars of the #th outcome variable or subpopulation if `overlay` has been specified. *options* are `step` (draw step function instead of bars) and *barlook_options* (affect rendition of the plotted bars).

`psep`[`("`*label1*`" "`*label2*`" ...)`] causes different rendering to be used for each percentile group and includes a corresponding legend in the graph. The default is to draw all bars in the same style. `psep()` and `overlay` are not both allowed.

`p#(`*options*`)` affects the rendition of the bars of the #th percentile group if `psep()` has been specified. *options* are *barlook_options* (affect rendition of the plotted bars) and `ciopts(`*options*`)` (affect rendition of the confidence spikes).

Confidence intervals

`level(#)` specifies the confidence level, as a percentage, for CIs. The default is the level that has been used for computing the `pshare` results. `level()` cannot be used together with `ci(bc)`, `ci(bca)`, or `ci(percentile)`. To change the level for these CIs, you need to specify `level()` when computing the results.

`ci(`*citype*`)` chooses the type of CIs to be plotted for results that have been computed using the bootstrap technique. *citype* may be `normal` (normal-based CIs, the default), `bc` (bias-corrected [BC] CIs), `bca` (BC and accelerated CIs), or `percentile` (percentile CIs). `bca` is available only if BC_a CIs have been requested when running `pshare estimate` (see [R] **bootstrap**).

`ciopts(`*options*`)` affects the rendition of the plotted confidence spikes. *options* depend on the plot type used for the confidence spikes. The default plot type is capped spikes; see [G-2] **graph twoway rcap**. To use uncapped spikes, for example, type `ciopts(recast(rspike))`; see [G-2] **graph twoway rspike**. `ciopts()` may be included in `p#()`, if `psep` has been specified, to affect the rendition of the confidence spikes for selected percentile groups.

`cibelow` places CI spikes behind the plotted bars. The default is to draw the spikes in front of the bars.

`noci` omits CI spikes from the plot.

Add plots

`addplot(`*plot*`)` provides a way to add other plots to the generated graph; see [G-3] ***addplot_option***.

Y axis, X axis, Title, Caption, Legend, Overall

twoway_options are general twoway options, other than `by()`, as documented in [G-3] ***twoway_options***.

4 Examples

4.1 Basic application

By default, `pshare` computes outcome shares of quintile groups. The following example shows the results for wages in the 1988 extract of the U.S. National Longitudinal Study of Young Women data shipped with Stata:

```
. sysuse nlsw88
(NLSW, 1988 extract)
. pshare estimate wage, percent
Percentile shares (percent)       Number of obs   =      2,246
```

wage	Coef.	Std. Err.	[95% Conf.	Interval]
0-20	8.018458	.1403194	7.743288	8.293627
20-40	12.03655	.1723244	11.69862	12.37448
40-60	16.2757	.2068139	15.87013	16.68127
60-80	22.47824	.2485367	21.99085	22.96562
80-100	41.19106	.6246426	39.96612	42.41599

The `percent` option was specified to express results as percentages. We can see, for example, that the 20% best-earning women in the data receive 41% of the total of wages, whereas the 20% poorest-earning women receive only 8%. If wages were distributed evenly, then all quintile groups would receive 20%.

To compute decile shares, we could type

```
. pshare estimate wage, percent nquantiles(10)
Percentile shares (percent)       Number of obs   =      2,246
```

wage	Coef.	Std. Err.	[95% Conf.	Interval]
0-10	3.426509	.0702149	3.288816	3.564202
10-20	4.591949	.0813845	4.432352	4.751546
20-30	5.544608	.0842676	5.379357	5.709858
30-40	6.491941	.0934605	6.308663	6.675219
40-50	7.542334	.1023013	7.341719	7.742948
50-60	8.733366	.1131891	8.5114	8.955333
60-70	10.24571	.1284118	9.993888	10.49752
70-80	12.23253	.1367424	11.96438	12.50069
80-90	14.65518	.1493718	14.36226	14.9481
90-100	26.53588	.682887	25.19672	27.87503

The results indicate that the 10% best-earning women get 26.5% of the wages, whereas the 10% poorest-earning women get only 3.4%.

`pshare` does not require the percentile groups to be of equal size. To compute the shares of, say, the bottom 50%, the mid 40%, and the top 10%, we could type

```
. pshare estimate wage, percent percentiles(50 90)
Percentile shares (percent)       Number of obs   =      2,246
```

wage	Coef.	Std. Err.	[95% Conf.	Interval]
0-50	27.59734	.3742279	26.86347	28.33121
50-90	45.86678	.4217771	45.03967	46.6939
90-100	26.53588	.682887	25.19672	27.87503

The `percentiles()` option specifies the cutoffs defining the percentile groups. That is, `percentiles(50 90)` indicates to use three groups, 0–50, 50–90, and 90–100. We see that the lower-paid half of women gets about the same share of total wages as the best-paid 10%.

4.2 Stacked bar charts

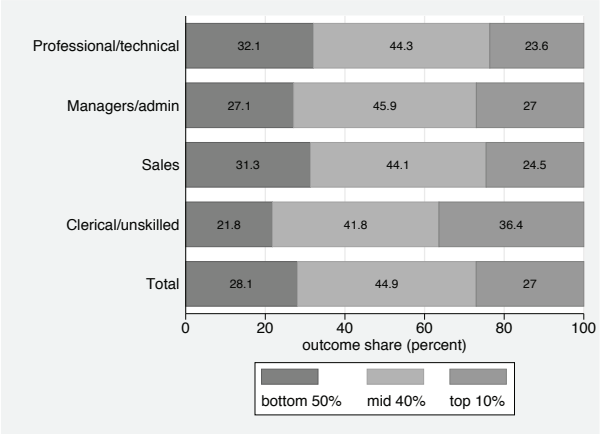
`pshare` supports two types of graphical displays of percentile shares. The first type is a stacked bar chart. For example, to compare wage distributions by some occupational groups, we could type

```
. pshare estimate wage if occupation<=4, percent percentiles(50 90)
> over(occupation) total gini
Percentile shares (percent)       Number of obs   =      1,409
            1: occupation = Professional/technical
            2: occupation = Managers/admin
            3: occupation = Sales
            4: occupation = Clerical/unskilled
```

wage	Coef.	Std. Err.	[95% Conf.	Interval]
1				
0-50	32.08652	.9560224	30.21114	33.9619
50-90	44.30132	.8461561	42.64146	45.96118
90-100	23.61216	1.468329	20.73181	26.49251
2				
0-50	27.11145	1.015934	25.11854	29.10436
50-90	45.90042	.8232238	44.28555	47.5153
90-100	26.98812	1.337874	24.36368	29.61256
3				
0-50	31.34111	.730376	29.90836	32.77385
50-90	44.1261	.7914729	42.57351	45.6787
90-100	24.53279	1.378169	21.8293	27.23627
4				
0-50	21.78931	1.909258	18.04401	25.53461
50-90	41.83106	2.046101	37.81733	45.84479
90-100	36.37963	2.898928	30.69295	42.06631
total				
0-50	28.06045	.4731704	27.13226	28.98865
50-90	44.91512	.4944292	43.94522	45.88502
90-100	27.02443	.8354367	25.38559	28.66326

	Gini
1	.273825
2	.3373482
3	.2833736
4	.4357447
total	.3279324

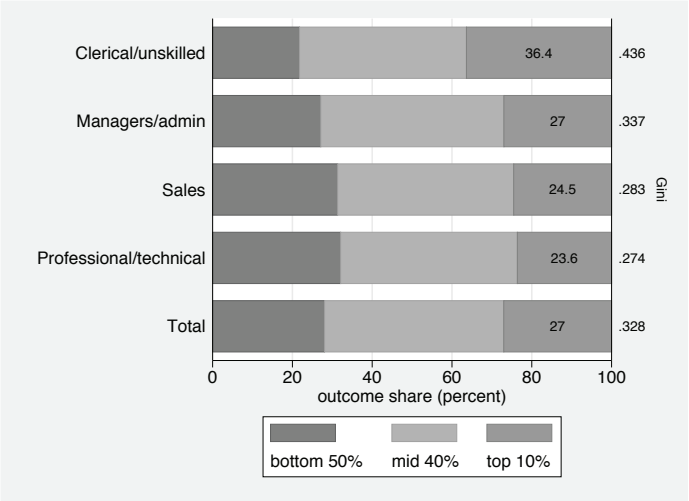
```
. pshare stack, plabels("bottom 50%" "mid 40%" "top 10%") values nogini
```



The `over(occupation)` option causes results to be computed by the subpopulations defined by the values of `occupation`, the `total` option requests total results across subpopulations to be included, and the `gini` option causes Gini coefficients to be computed. The `plabels()` option of `pshare stack` provides custom labels for the legend keys, the `values` option causes the values of the shares to be included as marker labels in the graph, and the `nogini` option suppresses the Gini coefficients that would be included in the graph as secondary axis labels (see next example).

To sort the bars by level of inequality, we could type

```
. pshare stack, plabels("bottom 50%" "mid 40%" "top 10%") values
> sort(gini tlast descending) mlabsize(zero) p3(mlabsize(small))
```

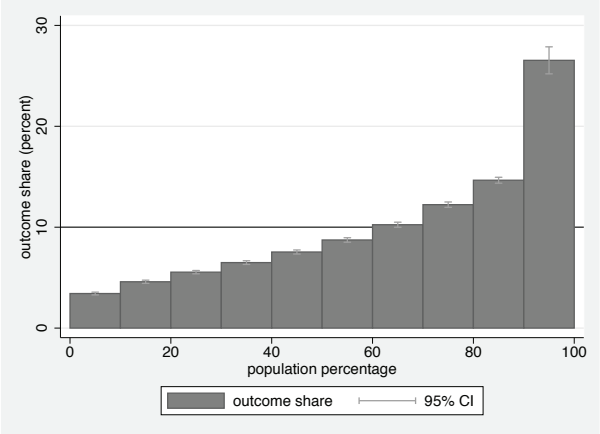


The `gini` argument in `sort()` causes bars to be sorted by Gini coefficients, `tlast` specifies placing the overall results last, and `descending` requests sorting from highest inequality to lowest inequality. The example also illustrates how to print marker labels only for specific percentile groups. The global option `mlabsize(zero)` sets the size of the marker labels to zero so that they are invisible, but `p3(mlabsize(small))` resets the marker label size for the third percentile group to `small`.

4.3 Histograms

The second type of graphical display supported by `pshare` is a percentile share histogram. The basic idea is to display a bar chart in which the area of each bar is proportional to the outcome share of the respective percentile group. An example with decile shares is as follows:

```
. pshare estimate wage, percent nquantiles(10)
  (output omitted)
. pshare histogram, yline(10)
```



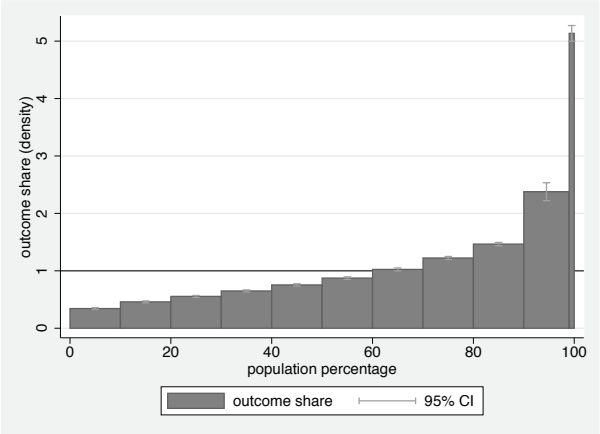
The `yline(10)` option was added to print a reference line at 10%. This would be the share each group would receive in an equal distribution.

If percentile groups are of unequal size, then densities instead of percentages or proportions should be used to construct the histogram (otherwise, the areas of the bars would no longer be proportional to the shares). Here is an example in which the top 1% is a separate group:

```
. pshare estimate wage, density percentiles(10(10)90 99)
Percentile shares (density)       Number of obs   =      2,246
```

wage	Coef.	Std. Err.	[95% Conf.	Interval]
0-10	.3426509	.0070215	.3288816	.3564202
10-20	.4591949	.0081384	.4432352	.4751546
20-30	.5544608	.0084268	.5379357	.5709858
30-40	.6491941	.009346	.6308663	.6675219
40-50	.7542334	.0102301	.7341719	.7742948
50-60	.8733366	.0113189	.85114	.8955333
60-70	1.024571	.0128412	.9993888	1.049752
70-80	1.223253	.0136742	1.196438	1.250069
80-90	1.465518	.0149372	1.436226	1.49481
90-99	2.377868	.0794248	2.222114	2.533622
99-100	5.135065	.0696951	4.998392	5.271739

```
. pshare histogram, yline(1)
```



Percentile share densities have an intuitive interpretation. They indicate how much each member in a group gets (on average) in relation to the overall average. In the example, we see that the average pay of the lowest 10% is only about 35% of the overall average. On the other hand, the members in the top percentage group earn wages that are more than five times the average wage. An alternative interpretation is as follows: Think of 100 representative dollars to be distributed among 100 people. In an equal distribution everyone would get one dollar. If, however, you divide the 100 dollars according to the observed distribution, then the density of a particular group indicates how many representative dollars a person in that group would get. In the example above, we see that the 10 women at the bottom would only get 35 cents each, whereas the top women would get more than 5 dollars (about 15 times as much). We also see that about 60% of the women are below the equal distribution line (that is, they receive below-average wages).

Note that the percentile density histogram is closely related to the so-called quantile plot (see [R] **diagnostic plots** and Cox [1999]), also known as Pen's "Parade of Dwarfs (and a few Giants)" (Pen 1971, 48–59). The difference is that a quantile plot usually displays individual observations using the original scale of the outcome variable. In the percentile density histogram, the values are averaged within bins and normalized by the population average.

4.4 Contrasts

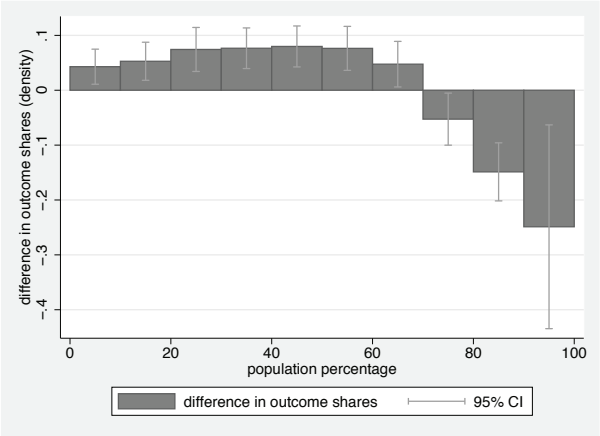
Differences between subpopulations

A useful feature of `pshare` is that contrasts between distributions can be computed. For example, the difference in the wage distribution between unionized and nonunionized women could be analyzed as follows:

```
. pshare estimate wage, density over(union) n(10)
  (output omitted)
. pshare contrast 0
Differences in percentile shares (density)      Number of obs     =      1,878
           0: union = nonunion
           1: union = union
```

wage	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
1						
0-10	.0429197	.016305	2.63	0.009	.0109419	.0748975
10-20	.0528084	.0177041	2.98	0.003	.0180866	.0875301
20-30	.0743417	.0204516	3.64	0.000	.0342315	.1144519
30-40	.0765406	.018892	4.05	0.000	.0394891	.1135922
40-50	.0798209	.0190538	4.19	0.000	.0424521	.1171897
50-60	.0763097	.0204552	3.73	0.000	.0361924	.116427
60-70	.0475279	.0211824	2.24	0.025	.0059843	.0890715
70-80	-.0526677	.0242038	-2.18	0.030	-.1001369	-.0051984
80-90	-.1487654	.0269943	-5.51	0.000	-.2017074	-.0958234
90-100	-.2488358	.094742	-2.63	0.009	-.4346464	-.0630251

```
(contrasts with respect to union = 0)
. pshare histogram
```

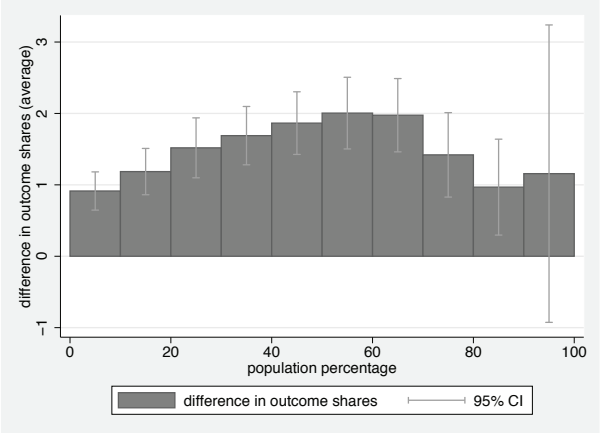


From the results, we see that the bottom 70% are relatively better off if unionized; the top 30% are relatively worse off. The differences are expressed in representative dollars; that is, the bottom 70% gain around 5 representative cents, and the top 10%

lose about a quarter of a representative dollar. However, note that these differences reflect only differences in the distributional shape; they are net of a possible overall difference in the wage levels between unionized and nonunionized workers.

To take the different wage levels of unionized and nonunionized workers into account, specify the `average` option so that the results are expressed as average wages. Furthermore, note that instead of using the `pshare contrast` command, you can also compute contrasts directly by applying the `contrast()` option to `pshare estimate`:

```
. pshare estimate wage, average over(union) n(10) contrast(0) histogram
```
(*output omitted*)



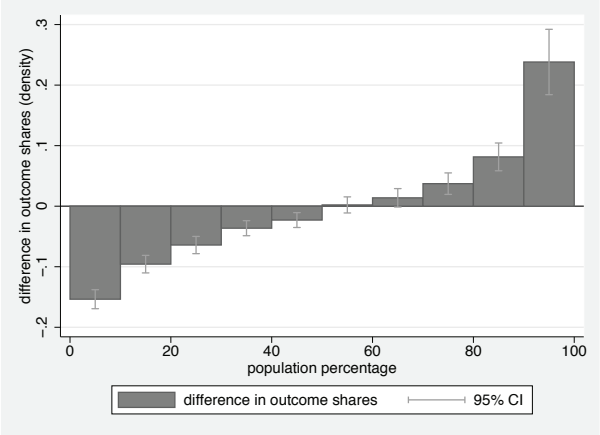
From these results, we see that unionized workers are better off across the board (by about one to two dollars per hour). Hence, from a welfare perspective, one could argue that the wage distribution of unionized women is strictly preferable over the wage distribution of nonunionized women (the wage distribution of unionized women generalized Lorenz dominants the wage distribution of nonunionized women; see, for example, Lambert [2001]). We also see that the (absolute) gains are somewhat larger in the middle of the distribution than at the top and at the bottom.

Differences between outcome variables

Instead of comparing subpopulations, `pshare` can also be used to compare distributions of different variables. For example, we could be interested in how the distribution changes once we move from hourly wages to weekly earnings:

```
. generate weekly = hours * wage
(4 missing values generated)
. label variable weekly "weekly earnings"
. pshare estimate wage weekly, density n(10) contrast(wage)
```
(*output omitted*)

```
. pshare histogram, yline(0)
```



We see that weekly earnings are considerably more unequal than hourly wages. Apparently, and as expected by economic theory, women with higher wages do supply more labor, so they get a larger share of weekly earnings than of hourly wages.

4.5 Concentration shares

The relation between two continuous variables can be analyzed by the `pshare` command using the `pvar()` option (in this case, percentile shares correspond to differences in concentration curve ordinates). In the last example, we saw that weekly earnings are distributed more unequally than hourly wages, which implies that women with higher wages work longer hours. Hence, it might be interesting to see how labor supply is distributed across wage groups:

```
. pshare estimate hours, pvar(wage) average n(10)
Percentile shares (average)       Number of obs   =      2,242
```

hours	Coef.	Std. Err.	[95% Conf.	Interval]
0-10	33.05259	.889763	31.30775	34.79744
10-20	33.6382	.8199639	32.03023	35.24616
20-30	34.78557	.7480189	33.31869	36.25245
30-40	37.14429	.6222536	35.92404	38.36454
40-50	37.73974	.6375459	36.4895	38.98998
50-60	38.6289	.670502	37.31403	39.94377
60-70	39.17663	.5903086	38.01902	40.33424
70-80	38.59946	.5712248	37.47928	39.71965
80-90	40.03568	.5799854	38.89832	41.17305
90-100	39.38002	.660688	38.08439	40.67564

```
(percentile groups with respect to wage)
```

The results indicate that average labor supply by women in the bottom 30% of the wage distribution is only about 33 to 35 hours per week, whereas in the upper half of the wage distribution, it is about 40 hours per week. To obtain results expressed in relation to the overall average, use the `density` option:

```
. pshare estimate hours, pvar(wage) density n(10)
Percentile shares (density)       Number of obs   =      2,242
```

hours	Coef.	Std. Err.	[95% Conf.	Interval]
0-10	.8880782	.0222773	.8443919	.9317646
10-20	.9038126	.0205245	.8635637	.9440616
20-30	.934641	.0188478	.8976801	.971602
30-40	.9980166	.0159431	.9667519	1.029281
40-50	1.014016	.0162895	.9820715	1.04596
50-60	1.037906	.0170757	1.00442	1.071392
60-70	1.052623	.0153487	1.022524	1.082722
70-80	1.037115	.0149871	1.007725	1.066505
80-90	1.075704	.0151754	1.045945	1.105464
90-100	1.058088	.0169731	1.024803	1.091372

```
(percentile groups with respect to wage)
```

We see, for example, that the weekly labor supply of women in the top 10% of the wage distribution is about 6% higher than average labor supply. The weekly labor supply of women in the bottom 10% of the wage distribution is 11% below the average.

The same technique could also be used, for example, to study the relation between income and wealth or between received bequests and existing income or wealth (for example, how much of the sum of all bequests in a given year goes to the wealthiest 10% of the population). Furthermore, it could be used to study the composition of income by sources or to study the effects of redistribution (for example, how much the different income percentiles contribute to overall taxes and how the empirical tax progression looks).

4.6 Processing results from pshare

`pshare estimate` and `pshare contrast` post their results in the `e()` return (see [P] **ereturn**; also see [U] **13.5 Accessing coefficients and standard errors**), so they can be processed by postestimation commands such as `test` (see [R] **test**), `lincom` (see [R] **lincom**), and `nlcom` (see [R] **nlcom**) or tabulated and graphed by programs such as `estout` (Jann 2005, 2007) and `coefplot` (Jann 2014).

For example, to compute the Palma ratio of wages—top 10% share divided by bottom 40% share (see, for example, Cobham, Schlogl, and Sumner [2015])—we could type

```
. pshare estimate wage, percentiles(40 90)
Percentile shares (proportion)    Number of obs   =      2,246
```

wage	Coef.	Std. Err.	[95% Conf.	Interval]
0-40	.2005501	.0029161	.1948315	.2062687
40-90	.5340912	.0048778	.5245258	.5436566
90-100	.2653588	.0068289	.2519672	.2787503

```
. nlcom (Palma: _b[90-100] / _b[0-40])
      Palma:  _b[90-100] / _b[0-40]
```

wage	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Palma	1.323155	.0506042	26.15	0.000	1.223972	1.422337

Furthermore, the Lorenz ordinates used to compute the percentile shares are stored by `pshare` in `e(L_ll)` (lower bounds) and `e(L_ul)` (upper bounds). To tabulate the Lorenz ordinates together with the percentile shares, we could type

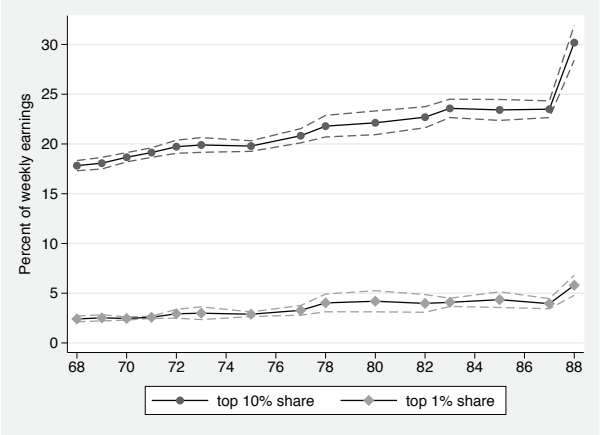
```
. pshare estimate wage
  (output omitted)
. estout, cell((b(label(share)) L_ll L_ul)) mlabels(none)
```

	share	L_ll	L_ul
0-20	.0801846	0	.0801846
20-40	.1203655	.0801846	.2005501
40-60	.162757	.2005501	.3633071
60-80	.2247824	.3633071	.5880894
80-100	.4119106	.5880894	1

Finally, `estimates store` (see [R] **estimates store**) can be used to make copies of results from different calls to `pshare` for later usage by commands such as `estout` or `coefplot`. In the following example, `coefplot` is used to plot the top decile share and the top centile share of weekly earnings against time:

```
. use http://www.stata-press.com/data/r14/nlswork.dta, clear
(National Longitudinal Survey.  Young Women 14-26 years of age in 1968)
. gen weekly = exp(ln_wage) * hours
(67 missing values generated)
. pshare estimate weekly, percent percentile(90) over(year) vce(cluster idcode)
  (output omitted)
. estimates store p90
. pshare estimate weekly, percent percentile(99) over(year) vce(cluster idcode)
  (output omitted)
```

```
. estimates store p99
. coefplot (p90, keep(*:90-100) label("top 10% share"))
>          (p99, keep(*:99-100) label("top 1% share")) ,
>          at(_eq) recast(connected) ciopts(recast(rline) lpattern(dash))
>          xlabel(68(2)88) ylabel(0(5)30, angle(horizontal))
>          ytitle("Percent of weekly earnings")
```



Through the years, as the respondents grew older, the share of the top decile increased from about 18% to 30%. The share of the top centile increased from 2.5% to about 5%.[13]

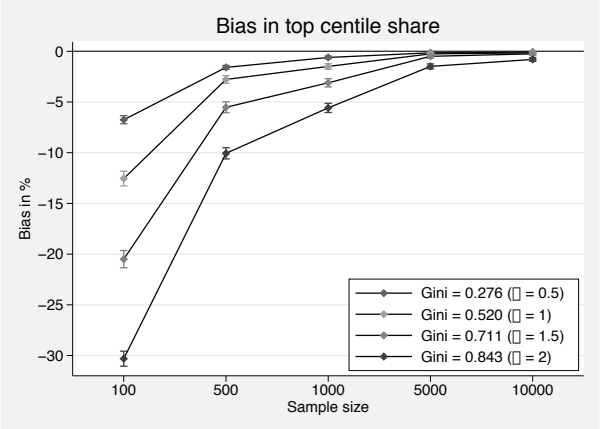
5 Small-sample bias

Estimates of percentile shares are affected by small-sample bias, especially at the top of the distribution. The bias can be substantial if the distribution is highly skewed and the number of observations is small. Thus, to obtain reliable estimates for shares of small top groups such as, say, the top 0.1% share, one must use large samples.

The simulation below provides some results for the relative bias in the estimate of the top 1% share for different sample sizes using a lognormal distribution. The scale parameter of the lognormal distribution is varied between $\sigma = 0.5$ (corresponding to a Gini coefficient of 0.276) and $\sigma = 2$ (corresponding to a Gini coefficient of 0.843).

13. The `vce(cluster idcode)` option has been added because the data are from a panel study where `idcode` identifies individuals. Adding the option in the example is not strictly necessary because the variances of the yearly estimates are not affected much by the clustering. However, it will be relevant once differences between years are analyzed.

```
. set seed 3230982
. program mysim, rclass
  1.     syntax [, n(integer 1000) Sigma(real 1) ]
  2.     drop _all
  3.     qui set obs `n´
  4.     tempvar y
  5.     gen `y´ = exp(rnormal(0, `sigma´))
  6.     pshare estimate `y´, nose percentile(99)
  7.     local b = 1 - normal(invnorm(0.99) - `sigma´)
  8.     return scalar bias = (_b[99-100] - `b´) / `b´
  9. end
. local i 0
. capture matrix drop R
. foreach sigma in 0.5 1 1.5 2 {
  2.     local ++i
  3.     local gini = 2*normal(`sigma´/sqrt(2)) - 1
  4.     foreach n in 100 500 1000 5000 10000 {
  5.         quietly simulate r(bias), reps(10000): mysim, n(`n´) sigma(`sigma´)
  6.         quietly ci means _sim_1
  7.         matrix tmp = r(mean), r(lb), r(ub)
  8.         matrix rownames tmp = s`i´:`n´
  9.         matrix R = nullmat(R) , tmp´
 10.     }
 11. }
. local i 0
. local plots
. foreach sigma in 0.5 1 1.5 2 {
  2.     local ++i
  3.     local lbl `: di %9.3f 2*normal(`sigma´/sqrt(2)) - 1´
  4.     local lbl Gini = `lbl´ ({&sigma} = `sigma´)
  5.     local plots `plots´ (matrix(R), keep(s`i´:) label("`lbl´"))
  6. }
. coefplot `plots´, ci((R[2] R[3])) vertical nooffset rescale(100)
>     msymbol(d) xtitle(Sample size) recast(connected) ciopts(recast(rcap))
>     ytitle(Bias in %) ylabel(#10, angle(horizontal)) yline(0)
>     title(Bias in top centile share) legend(cols(1) position(0) bplace(se))
```



For example, in a sample of 100 observations, the top centile share is underestimated by about 30% for a lognormal distribution with a Gini coefficient of 0.843. For lower levels of inequality, the underestimation is less severe but still substantial. This is not much of a surprise because in a sample of 100 observations, the top centile group contains only a single observation. However, also with a sample size of 1,000, the top centile share is underestimated by about 5% for the distribution with a Gini coefficient of 0.843.

The simulation results suggest that for moderately skewed distributions (such as the income distribution with a typical Gini coefficient between about 0.3 and 0.6), there should be a minimum of about 10 observations in the top group to keep the error within acceptable bounds of just a few percent. Estimating the top 0.1% share, for example, requires a sample size of at least 10,000 observations. However, for accurate estimation of top shares in extremely skewed distributions (such as the wealth distribution with Gini coefficients as high as 0.8 or event 0.9), minimum sample-size requirements may be considerably higher (such as 50 or even 100 observations in the top group).

6 Discussion

In this article, I presented only a selection of the features of the `pshare` command. It has been designed in such a way that it offers a wide variety of possible applications and can be used in many different situations. For example, much effort has been put into the support for complex survey data, a topic that has not been touched on in the examples. Nonetheless, a number of limitations and remaining issues need to be mentioned.

First, `pshare` is designed to be applied to individual-level data. Often, however, data on the distribution of income or wealth are available in the form of aggregate tables (typically from tax statistics). In such tables, individual-level units are grouped into outcome brackets, and for each bracket, the number of units and the outcome total are reported. `pshare` can be applied to such grouped data by computing the average outcome per bracket and weighting the data by the number of units. However, such a procedure assumes perfect equality within brackets and thus provides only a lower bound of the true inequality in the distribution (see, for example, Cowell [2011]). It would be worthwhile to develop a companion command for grouped data that also offers upper-bound estimates and intermediate estimates.

Second, analytic variance estimation implemented in `pshare` is only approximate and, possibly, more accurate estimation procedures could be developed. For example, variance estimation for percentile shares based on the concentration curve (that is, if the `pvar()` option is specified) requires the estimation of the expectation of the outcome variable at specific quantiles of the auxiliary variable. In the current implementation of `pshare`, this is accomplished by local linear regression using a constant bandwidth (see footnote 7). Some preliminary simulations indicate that this procedure generates consistent estimates of standard errors. However, the accuracy and stability of the standard error estimates could possibly be improved by using a variable bandwidth depending on the local density of the data. Furthermore, `pshare` reports symmetric,

normal-based CIs that may not be very accurate in small samples. A topic for future research could thus be to develop refined estimation of CIs.

Third, as discussed above, percentile shares are affected by small-sample bias. Future research will have to show whether a suitable correction procedure can be designed. A main challenge is to ensure that the correction does not increase the mean squared error (MSE) of the estimates. The problem can be illustrated by a simple bootstrap correction procedure. Let $\widehat{S}$ be the uncorrected estimate in the original sample and $\overline{S}$ be the mean of the estimates from a number of bootstrap samples. The bias in the bootstrap samples with respect to the original sample is then given as $\overline{S} - \widehat{S}$. The idea is to use the bootstrap bias as an approximation of the bias of the sample with respect to the population. Hence, a corrected estimate of S can be obtained as $\widehat{S}^{\text{corr}} = \widehat{S} - (\overline{S} - \widehat{S}) = 2\widehat{S} - \overline{S}$. Alternatively, the correction could also be based on ratios or on odds ratios between $\overline{S}$ and $\widehat{S}$. Findings from simulations with such procedures are that the bootstrap correction mostly removes the bias, unless the distribution is extremely skewed. At the same time, however, MSE increases. The reason is obvious: the larger the top share in a given sample turns out to be, the larger will be the bootstrap correction. This inflates sampling variance. Possibly, however, parametric extreme-value estimation may be used to design a correction procedure that does not increase the MSE.

7 Acknowledgments

The histogram and stacked bar plots produced by `pshare` have been inspired by the graphs shown in a video posted by Evan Klassen on YouTube[14] and a TED talk by Dan Ariely.[15] The `max()` and `min()` options of `pshare histogram` have been independently suggested by Hans-Jürgen Andreß and Jonas Meier. Furthermore, I would like to thank Stephen P. Jenkins for his helpful advice.

8 References

Atkinson, A. B., T. Piketty, and E. Saez. 2011. Top incomes in the long run of history. *Journal of Economic Literature* 49: 3–71.

Binder, D. A., and M. S. Kovacevic. 1995. Estimating some measures of income inequality from survey data: An application of the estimating equations approach. *Survey Methodology* 21: 137–145.

Cobham, A., L. Schlogl, and A. Sumner. 2015. Inequality and the tails: The Palma proposition and ratio revisited. United Nations DESA Working Paper No. 143. http://www.un.org/esa/desa/papers/2015/wp143_2015.pdf.

Cowell, F. A. 2011. *Measuring Inequality*. 3rd ed. Oxford: Oxford University Press.

14. See http://www.youtube.com/watch?v=slTF_XXoKAQ.
15. See https://www.ted.com/talks/dan_ariely_how_equal_do_we_want_the_world_to_be_you_d_be_surprised.

Cox, N. J. 1999. gr42: Quantile plots, generalized. *Stata Technical Bulletin* 51: 16–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 9, pp. 113–116. College Station, TX: Stata Press.

Hyndman, R. J., and Y. Fan. 1996. Sample quantiles in statistical packages. *American Statistician* 50: 361–365.

Jann, B. 2005. Making regression tables from stored estimates. *Stata Journal* 5: 288–308.

———. 2007. Making regression tables simplified. *Stata Journal* 7: 227–244.

———. 2014. Plotting regression coefficients and other estimates. *Stata Journal* 14: 708–737.

Jenkins, S. P. 1999. sg104: Analysis of income distributions. *Stata Technical Bulletin* 48: 4–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 243–260. College Station, TX: Stata Press.

———. 2006. svylorenz: Stata module to derive distribution-free variance estimates from complex survey data, of quantile group shares of a total, cumulative quantile group shares. Statistical Software Components S456602, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s456602.html.

Kakwani, N. C. 1977. Measurement of tax progressivity: An international comparison. *Economic Journal* 87: 71–80.

Kovačević, M. S., and D. A. Binder. 1997. Variance estimation for measures of income inequality and polarization—The estimating equations approach. *Journal of Official Statistics* 13: 41–58.

Lambert, P. J. 2001. *The Distribution and Redistribution of Income*. 3rd ed. Manchester: Manchester University Press.

Lerman, R. I., and S. Yitzhaki. 1989. Improving the accuracy of estimates of Gini coefficients. *Journal of Econometrics* 42: 43–47.

Pen, J. 1971. *Income Distribution*. London: Allen Lane.

Piketty, T. 2014. *Capital in the Twenty-First Century*. Cambridge, MA: Belknap Press.

Piketty, T., and E. Saez. 2014. Inequality in the long run. *Science* 344: 838–843.

About the author

Ben Jann is professor of sociology at the University of Bern, Switzerland. His research interests include social science methodology, statistics, social stratification, and labor market sociology. Recent publications include articles in *Sociological Methodology*, *Sociological Methods and Research*, the *Stata Journal*, *Public Opinion Quarterly*, and the *American Sociological Review*.