STATA TECHNICAL BULLETIN

March 1999
STB-48

A publication to promote communication among Stata users

Subscriptions are available from Stata Corporation, email stata@stata.com, telephone 979-696-4600 or 800-STATAPC, fax 979-696-4601. Current subscription prices are posted at www.stata.com/bookstore/stb.html.

Previous Issues are available individually from StataCorp. See www.stata.com/bookstore/stbj.html for details.

Submissions to the STB, including submissions to the supporting files (programs, datasets, and help files), are on a nonexclusive, free-use basis. In particular, the author grants to StataCorp the nonexclusive right to copyright and distribute the material in accordance with the Copyright Statement below. The author also grants to StataCorp the right to freely use the ideas, including communication of the ideas to other parties, even if the material is never published in the STB. Submissions should be addressed to the Editor. Submission guidelines can be obtained from either the editor or StataCorp.

The *Stata Technical Bulletin* (ISSN 1097-8879) is published six times per year by Stata Corporation. Stata is a registered trademark of Stata Corporation.

Contents of this issue — page

gr34.1	Drawing Venn diagrams

Jens M. Lauritsen, County of Fyn, Denmark, jm.lauritsen@dadlnet.dk

`venndiag` produces a so-called Venn diagram based on variables in a dataset.

The Venn diagram routine has been expanded such that thickness of lines and pen choice can be changed. See Lauritsen (1999) for further explanations.

Syntax

`venndiag` *varlist* [`if` *exp*] [`in` *range*] [, `label(`*str*`)` `show(`*str*`)` `missing` `gen(`*varnames*`)` `list(`*variables*`)` `print` `saving(`*filename*`)` `c1(`*#*`)` `c2(`*#*`)` `c3(`*#*`)` `c4(`*#*`)` `noframe` `nograph` `nolabel` `t1title(`*str*`)` `t2title(`*str*`)` `t3title(`*str*`)` `r1title(`*str*`)` `r2title(`*str*`)` `r3title(`*str*`)` `r4title(`*str*`)` `r5title(`*str*`)` `r6title(`*str*`)` `pen(`*#*`)` `thick(`*#*`)`]

where the *varlist* must contain from 2–4 numerical variables and if generating a variable, that variable must not exist. Only the new options are shown below. See updated help file for further information.

Added options

`pen(`*#*`)` indicates which pens to use in the graph, e.g., `pen(123)`. The first one is for text, the second for rectangles, and the third for the frame. The default is `pen(123)`.

`thick(`*#*`)` indicates the thickness of pens on printing (for Windows 95). The default is `thick(995)`. To obtain a thicker frame, reverse the order of the numbers, i.e., `thick(559)`. Note the link to `pen()`; for example, `pen(456)` must be followed by `thick(111995)` to make pen 4 and 5 thickness 9 and pen 6 thickness 5. The first three `1`'s are not used in this case. (The pen number is defined by it's position in `thick()`.)

Historical note—extension to STB-47

Another article by John Venn (1834–1923) has been located, such that the earliest publication by him on the subject most likely was 1880. See reference list.

Acknowledgment

Thanks to Ph. D. M. D. Charlotte G. Mörtz for testing and comments and to N. Cox for hinting at whom J. Venn was.

References

Lauritsen, J. 1999. gr34: Drawing Venn diagrams. *Stata Technical Bulletin* 47: 3–8.

Ruskey, F. 1997. A survey of Venn diagrams. *The Electronic Journal of Combinatorics* 4: DS#5. (available at: `http://sue.csc.uvic.ca/~cos/venn/`)

Venn, J. 1880. On the diagrammatic and mechanical representation of propositions and reasonings. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 9: 1–18.

——. 1881. *Symbolic Logic*. London: Macmillan.

gr35	Diagnostic plots for assessing Singh–Maddala and Dagum distributions fitted by MLE

Nicholas J. Cox, University of Durham, UK, n.j.cox@durham.ac.uk

Syntax

`psm` *varname* [`if` *exp*] [`in` *range*] [, `grid` *graph_options*]

`qsm` *varname* [`if` *exp*] [`in` *range*] [, `grid` *graph_options*]

`pdagum` *varname* [`if` *exp*] [`in` *range*] [, `grid` *graph_options*]

`qdagum` *varname* [`if` *exp*] [`in` *range*] [, `grid` *graph_options*]

Options

`grid` adds grid lines at the 0.25, 0.50, 0.75 quantiles and also, in the case of `qsm` and `qdagum`, at the 0.05, 0.10, 0.90, and 0.95 quantiles.

graph_options are any of the options allowed with `graph, twoway`; see help for `graph`.

Description

`psm` produces a probability plot for *varname* compared with a three-parameter Singh–Maddala distribution. `qsm` plots the quantiles of *varname* against the quantiles of a three-parameter Singh–Maddala distribution. The parameters a, b and q are taken from global macros `S_a`, `S_b`, and `S_q`, which is where `smfit` puts maximum likelihood estimates of them.

`pdagum` produces a probability plot for *varname* compared with a three-parameter Dagum distribution. `qdagum` plots the quantiles of *varname* against the quantiles of a three-parameter Dagum distribution. The parameters b, d and h are taken from `S_b`, `S_d`, and `S_h`, which is where `dagumfit` puts maximum likelihood estimates of them.

`smfit` and `dagumfit` are discussed in Jenkins (1999b).

Example

The illustrative example uses the same income distribution data as described in Jenkins (1999a). The income variable is `eybhc` with `fweight` variable `wgt`.

Singh–Maddala and Dagum distributions were first fitted using `smfit` and `dagumfit` (as in Jenkins 1999a), except that grossing-up weights were neglected this time since the plotting programs do not handle them. The results are as follows:

```
. smfit eybhc if eybhc >0
  (output omitted )
. qsm eybhc if eybhc>0, saving(qsm1.gph,replace)
. psm eybhc if eybhc>0, saving(psm1.gph,replace)
. dagumfit eybhc
  (output omitted )
. qdagum eybhc if eybhc>0, saving(qdagum1.gph,replace)
. pdagum eybhc if eybhc>0, saving(pdagum1.gph,replace)
. graph using psm1 pdagum1 qsm1 qdagum1
```

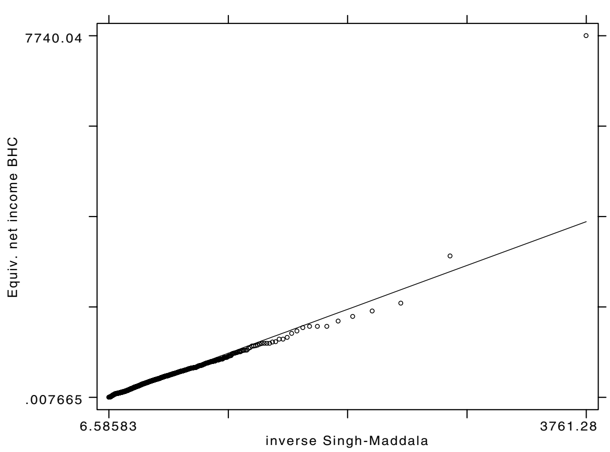

Figure 1. Output from qsm

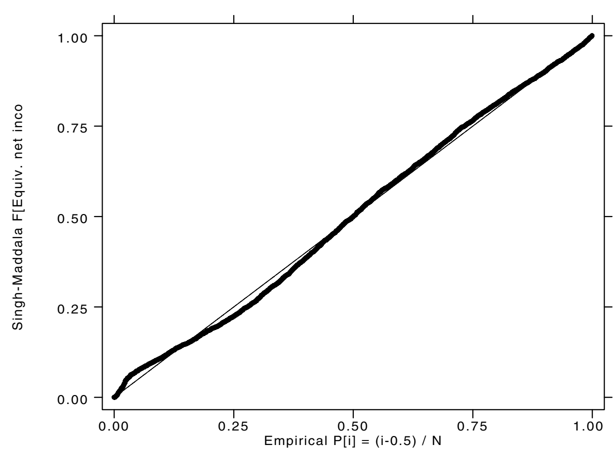

Figure 2. Output from psm

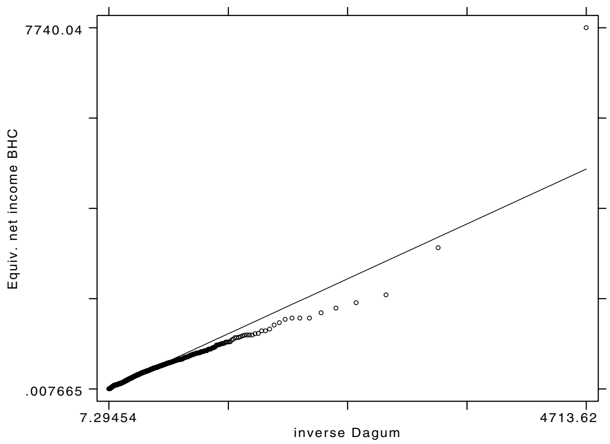

Figure 3. Output from pdagum

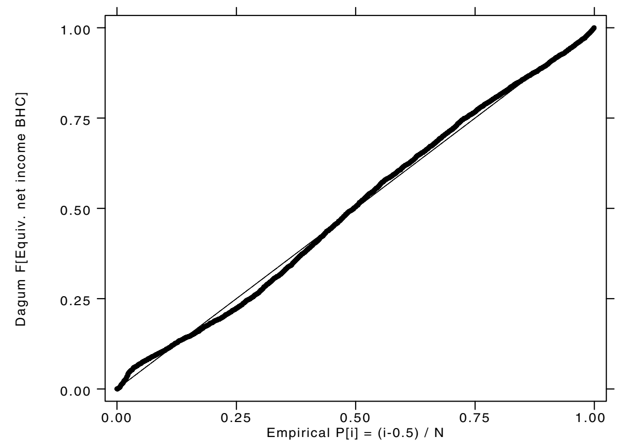

Figure 4. Output from qdagum

The plots confirm the conclusions of satisfactory goodness of fit based on other methods which were reported in the insert on fitting Singh–Maddala and Dagum distributions (Jenkins 1999b).

References

Jenkins, S. P. 1999a. sg104: Analysis of income distributions. *Stata Technical Bulletin* 48: 4–18.

——. 1999b. sg106: Fitting Singh–Maddala and Dagum distributions by maximum likelihood. *Stata Technical Bulletin* 48: 19–25.

sg104	Analysis of income distributions

Stephen P. Jenkins, University of Essex, UK, stephenj@essex.ac.uk

This insert provides a number of programs for summarizing distributions, and income distributions in particular.

- `sumdist` estimates quantiles, quantile group shares, Lorenz and generalized Lorenz ordinates.
- `xfrac` provides a tabulation using categories defined by fractions of a cut-off value (e.g., mean or median).
- `ineqdeco` estimates a selection of inequality indices (including Gini, Generalized Entropy, Atkinson indices) with optional decompositions by population subgroup into within- and between-group inequality components. `ineqdec0` is a cut-down version of this program.
- `geivars` provides estimates of selected Generalized Entropy inequality indices and their asymptotic sampling variances.
- `ineqfac` provides inequality decomposition by factor components.
- `povdeco` estimates three common poverty indices (the headcount ratio, averaged normalized poverty gap, and average squared normalized poverty gap), with optional decompositions by population subgroup.

These programs supplement various other numerical and graphical tools already in Stata for analyzing income distributions.

The programs are illustrated using income distribution data for 1991 derived by Goodman and Webb (1994) from the UK Family Expenditure Survey using the same definitions as the UK official income distribution statistics (see e.g., Department of Social Security, 1993). The data are available from the Data Archive at the University of Essex (`http://archive.essex.ac.uk`). The file used here comprises observations on 6,468 families (single persons or married couples, plus any children). A household may contain more than one family. Define the following variables:

- `ybhc` is the post-tax post-transfer money income of the household to which the family belongs, in pounds per week in 1991 prices.
- `eybhc` is needs-adjusted post-tax post-transfer household income, i.e., `ybhc` divided by an equivalence scale to account for differences in household size and composition. The scale used is the semi-official McClements one.
- `wgt` is an `fweight` used to “gross up” the estimates to represent all persons in the `UK` private household population.
- `tenure` is the housing tenure of the household in which the family lives (4 groups: social housing renter, other renter or rent-free, owned with a mortgage, owned outright).

sumdist: distribution summary statistics, by quantile group

sumdist estimates distributional summary statistics commonly used by income distribution analysts, complementing those available via pctile, xtile, and summarize, detail. In fact much of sumdist is a “wrapper” for xtile, combined with tabdisp to display the results of by-group calculations.

For variable x and distribution function $F(x)$, the statistics provided are

(1) quantiles $k = 1, 2, \dots, m-1$, for m =# quantile groups;

(2) the quantiles expressed as a percentage of median(x);

(3) the quantile group share of x in total x (group income share, %);

(4) the cumulative quantile group shares of total x (with cumulation in ascending order of x), i.e., the Lorenz ordinates $L(p)$ at each $p_k = F(x_k)$ for quantile points x_k; and

(5) the generalized Lorenz ordinates at each $p_k = F(x_k)$, i.e., $\mathrm{GL}(p_k) = \mathrm{mean}(x) * L(p_k)$.

Syntax

sumdist *varname* [*weight*] [if *exp*] [in *range*] [, ngps(*#*) qgp(*gpname*)]

fweights and aweights are allowed.

Options

ngps(*#*) specifies the number of quantile groups. Valid values are integers in the range $(0, 100]$. The default is 10.

qgp(*gpname*) creates a new categorical variable, *gpname*, containing categories summarizing quantile group membership, with the number of categories equal to m.

Example

We shall follow a conventional approach and examine the distribution of income amongst all persons in the population, assuming that each person receives the needs-adjusted income of the household to which s/he belongs. Thus we focus on the distribution of the variable eybhc weighted by wgt.

A summarize, detail shows some standard features of income distributions, namely significant dispersion combined with skewness: the mean is well above the median, and there is a long upper tail. (A more sophisticated analysis might consider the sensitivity of conclusions to differing treatments of the “outlier” largest income.)

```
. summarize eybhc [fw=wgt], de
                      Equiv. net income BHC
-------------------------------------------------------------
      Percentiles      Smallest
 1%        29.04      -123.9898
 5%     78.43056      -72.37004
10%     92.24828      -42.89144       Obs            55851705
25%     127.3008      -42.70588       Sum of Wgt.    55851705

50%     194.4472                      Mean           233.0179
                        Largest       Std. Dev.      199.0178
75%     287.2739       1846.438
90%      402.212       2013.499       Variance       39608.08
95%     503.1029       3024.663       Skewness       14.35982
99%      818.264       7740.044       Kurtosis        480.917
```

Observe the presence of negative and zero incomes in the data. It is up to the user to decide how to handle these. In general there may be arguments for or against exclusion of them, which vary with circumstances. By default sumdist retains these values, but they can be excluded using the if option. An example of default output is as follows:

```
. sumdist eybhc [fw=wgt]

Warning: eybhc has 20 values < 0. Used in calculations
Distributional summary statistics, 10 quantile groups
----------+-----------------------------------------------------------------
Quantile  |
group     |    Quantile  % of median     Share, %      L(p), %        GL(p)
----------+-----------------------------------------------------------------
        1 |       92.25        47.44         2.94         2.94         6.85
        2 |      115.77        59.54         4.47         7.41        17.26
        3 |      141.27        72.65         5.49        12.90        30.05
        4 |      167.22        86.00         6.61        19.50        45.44
        5 |      194.45       100.00         7.76        27.26        63.53
        6 |      225.38       115.91         9.04        36.30        84.59
        7 |      263.34       135.43        10.44        46.75       108.93
        8 |      315.39       162.20        12.38        59.13       137.78
        9 |      402.21       206.85        15.20        74.33       173.20
       10 |                                 25.67       100.00       233.02
----------+-----------------------------------------------------------------
Share = quantile group share of total eybhc;
L(p)=cumulative group share; GL(p)=L(p)*mean(eybhc)
```

We now have estimates of the nine deciles ($p10, p20, p30, \ldots, p90$) splitting the population into tenths ordered by income (decile groups): look at the `Quantile` column. The next column shows that $p10$ is about 47% of the median income ($= p50$). We can also see from the `Share` column that the poorest tenth of the UK population in 1991 received less than 3% of total income whereas the richest tenth received more than 25% of total income.

The `L(p)` column shows cumulative quantile group income shares, in other words, Lorenz ordinates. Lorenz curves are graphs connecting a plot of these points against cumulative population shares, and are often used for inequality summaries and inequality "dominance" comparisons (see e.g., Cowell 1995, Lambert 1993). The `GL(p)` column shows the values of `L(p)` multiplied by mean income. The generalized Lorenz curve is the Lorenz curve scaled up at each point by mean income, and is often used for "welfare" dominance comparisons (Cowell 1995, Lambert 1993). `sumdist` is designed to provide a numerical summary of these distributional features, rather than provide the data elements for drawing (generalized) Lorenz curve graphs. After all, if one has unit record data (as here), one might as well draw the graphs using all the data; see Jenkins and Van Kerm (1999).

If instead we had typed

```
. sumdist eybhc [fw=wgt], n(5) qgp(quintgp)
```

the program would have provided the four quartiles ($p20, p40, p60, p80$) splitting the population into fifths ordered by income, quintile group income shares etc., and created a new variable `quintgp` recording quintile group membership.

xfrac: tabulation using categories defined by fractions of a cut-off value

`xfrac` provides a specialized tabulation (a "wrapper" for `tabulate`). Each valid observation is first partitioned by *varname* into one of a set of 20 mutually-exclusive categories, the boundaries of which are defined by "hard-wired" fractions of a user-specified cut-off value (in the same units as *varname*), with fractions ranging from 0.1 through to 3.0. This classification is then tabulated and, optionally, can be retained as a new variable.

An example may clarify. Let *varname* be a measure of income and the cut-off be mean income. `xfrac` shows the proportion of observations with `varname` value less than 10% of mean income, between 10% and 20% of mean income, between 20% and 30% of mean income, and so on (20 categories). Cumulative proportions are also shown. The hard-wired fractions of the cut-off were chosen to match those used in the presentation of the UK official low income statistics (see, e.g., Department of Social Security, 1993). Motivated users could easily modify the `xfrac` code and change the choices if desired.

In effect `xfrac` provides a discrete representation of the distribution function for *varname*.

Syntax

`xfrac` *varname* [*weight*] [`if` *exp*] [`in` *range*] , `cut`off(*#*) [gp(*gpname*)]

`fweight`s and `aweight`s are allowed.

The user must specify a value for the cut-off value in the same units as *varname* using `cutoff(`*#*`)`.

Options

gp(*gpname*) creates a new categorical variable, **gpname**, containing categories summarizing group membership.

Example

To produce output mimicking the UK official low income statistics, we use the mean income as the cut-off value input into xfrac:

```
. summarize eybhc [fw=wgt]
Variable |     Obs        Mean   Std. Dev.       Min        Max
---------+-----------------------------------------------------
   eybhc | 5.6e+07    233.0179   199.0178  -123.9898   7740.044
. local mean = _result(3)
.
. xfrac eybhc [fw=wgt], cut(`mean´) gp(fracgp)
Warning: eybhc has 20 values < 0. Used in calculations
Proportions of the sample in subgroups defined
by values of eybhc between specified fractions
of a cut-off value = 233.01790
Fractions of|
cut-off     |      Freq.     Percent        Cum.
------------+-----------------------------------
       <.1  |     455152        0.81        0.81
     .1-.2  |     482238        0.86        1.68
     .2-.3  |     912526        1.63        3.31
     .3-.4  |    3983433        7.13       10.44
     .4-.5  |    5502971        9.85       20.30
     .5-.6  |    5186597        9.29       29.58
     .6-.7  |    4935514        8.84       38.42
     .7-.8  |    4777040        8.55       46.97
     .8-.9  |    4341904        7.77       54.75
     .9-1.0 |    4364218        7.81       62.56
   1.0-1.1 |    3234833        5.79       68.35
   1.1-1.2 |    2678779        4.80       73.15
   1.2-1.3 |    2655524        4.75       77.90
   1.3-1.4 |    2095389        3.75       81.66
   1.4-1.5 |    1683166        3.01       84.67
  1.5-1.75 |    3149798        5.64       90.31
  1.75-2.0 |    1848821        3.31       93.62
   2.0-2.5 |    1902059        3.41       97.02
   2.5-3.0 |     721933        1.29       98.32
     >=3.0 |     939810        1.68      100.00
------------+-----------------------------------
     Total |   55851705      100.00
```

There is no official poverty line in Britain, but half of the average income is used by many commentators as such a threshold. The xfrac output shows that about one fifth of the UK population in 1991 had incomes below one half of contemporary mean income (and 62.6% had incomes below the mean). But observe too that 38% of the population have incomes between 40% and 60% of mean income. Thus relatively small changes in the threshold defining the poverty line can have a large impact on estimates of the proportion who are "poor".

The command above also created a new variable summarizing income group membership. If we were now to type

```
. table fracgp tenure [fw=wgt], row col
```

we could compare the shape of the income distribution across housing tenure groups.

ineqdeco, ineqdec0: inequality indices, with decompositions by population subgroup

ineqdeco and ineqdec0 estimate a range of inequality and related indices commonly used by economists, plus decompositions of a subset of these indices by population subgroup into within- and between-group inequality components. Inequality decompositions by subgroup are useful for providing inequality profiles at a point in time, and for analyzing secular trends using shift-share analysis. Unit record (micro level) data are required. For a non-technical introduction to the topic, see Jenkins (1991). Standard textbook treatments are provided by Cowell (1995) and Lambert (1993).

Inequality indices estimated by ineqdeco are: members of the single parameter Generalized Entropy class $\mathrm{GE}(a)$ for $a = -1, 0, 1, 2$; the Atkinson class $A(e)$ for $e = 0.5, 1, 2$; the Gini coefficient, and percentile ratios such as $p90/p10$ and $p75/p25$. Also presented are related summary statistics such as subgroup means and population shares. Optionally presented are indices related to the Atkinson inequality indices, namely equally-distributed-equivalent income $Y_{\text{ede}}(e)$, social welfare indices $W(e)$, and the Sen welfare index; see below for details.

Calculations for `ineqdeco` exclude zero and negative income values since not all the indices are defined in such cases. `ineqdec0` is a stripped-down version of `ineqdeco` for situations when users wish to include zero and negative incomes in calculations, but estimates are provided for the Gini and GE(2) indices only in this case. Some programs for inequality indices have been provided in an earlier STB: see `inequal` and `rspread` in STB-23 (Whitehouse 1995, Goldstein 1995). These provide estimates for additional inequality indices. But weights cannot be used in all the programs and none of them provides full decompositions by population subgroup or estimates welfare indices.

The inequality indices differ in their sensitivities to differences in different parts of the distribution. The more positive a is, the more sensitive $\text{GE}(a)$ is to income differences at the top of the distribution; the more negative a is the more sensitive it is to differences at the bottom of the distribution. $\text{GE}(0)$ is the mean logarithmic deviation, $\text{GE}(1)$ is the Theil index, and $\text{GE}(2)$ is half the square of the coefficient of variation. The more positive $e > 0$ (the inequality aversion parameter) is, the more sensitive $A(e)$ is to income differences at the bottom of the distribution. It is readily confirmed that for each member of the Atkinson class $e = e_0$, there is a corresponding ordinally-equivalent member of the Generalized Entropy class with $a = 1 - e_0$. The Gini coefficient is most sensitive to income differences about the middle (more precisely, the mode).

`ineqdeco` has been designed not to estimate indices which are more "top-sensitive" or "bottom-sensitive" than those provided because experience shows that these can be very sensitive to the presence of just one or two very large or small income outliers.

A more detailed description is as follows. Consider a population of persons (or families or households, etc.,), $i = 1, \ldots, n$, with income y_i, and weight w_i. Let $f_i = w_i/N$, where $N = \sum_{i=1}^n w_i$. When the data are unweighted, $w_i = 1$ and $N = n$. Arithmetic mean income is m. Suppose there is an exhaustive partition of the population into mutually exclusive subgroups $k = 1, \ldots, K$.

The Generalized Entropy class of inequality indices is given by

$$\text{GE}(a) = \frac{1}{a(1-a)}\left[\left[\sum_{i=1}^n f_i (y_i/m)^a\right] - 1\right], a \neq 0, a \neq 1$$

$$\text{GE}(1) = \sum_{i=1}^n f_i (y_i/m)\log(y_i/m)$$

$$\text{GE}(0) = \sum_{i=1}^n f_i \log(m/y_i)$$

Each $\text{GE}(a)$ index can be additively decomposed as

$$\text{GE}(a) = \text{GE}_W(a) + \text{GE}_B(a)$$

where $\text{GE}_W(a)$ is within-group inequality and $\text{GE}_B(a)$ is between-group inequality; see Shorrocks (1984),

$$\text{GE}_W(a) = \sum_{k=1}^K V_k^{1-a} S_k^a \text{GE}_k(a)$$

where $V_k = N_k/N$ is the number of persons in subgroup k divided by the total number of persons (subgroup population share), and S_k is the share of total income held by k's members (subgroup income share).

$\text{GE}_k(a)$, inequality for subgroup k, is calculated as if the subgroup were a separate population, and $\text{GE}_B(a)$ is derived assuming every person within a given subgroup k received k's mean income, m_k.

Define the equally-distributed-equivalent income

$$Y_{\text{ede}}(e) = \left[\sum_{i=1}^n f_i (y_i)^{1-e}\right]^{1/1-e}, e > 0, e \neq 1$$

$$Y_{\text{ede}}(1) = \sum_{i=1}^{n} f_i \log(y_i)$$

The Atkinson indices (Atkinson 1970) are defined by

$$A(e) = 1 - \big[Y_{\text{ede}}(e)/m\big]$$

These indices are decomposable but not additively decomposable (Blackorby, Donaldson, and Auersperg 1981):

$$A(e) = A_W(a) + A_B(a) - \big[A_W(a)\big].\big[A_B(a)\big]$$

where

$$A_W(a) = 1 - \sum_{k=1}^{K} V_k Y_{\text{ede},k}/m$$

and

$$A_B(a) = 1 - \left[\frac{Y_{\text{ede}}}{\sum_{k=1}^{K} V_k Y_{\text{ede},k}/m}\right]$$

Social welfare indices (Jenkins 1997) are defined by

$$W_e = \frac{1}{1-e}\big[Y_{\text{ede}}(e)\big]^{1-e}, e \neq 0, e \neq 1$$

$$W_1 = \log\big[Y_{\text{ede}}(1)\big]$$

Each of these indices is an increasing function of a generalized mean of order $(1-e)$. All the welfare indices are additively decomposable:

$$W(e) = \sum_{k=1}^{K} V_k W_k(e)$$

The Gini coefficient is given by

$$G = 1 + (1/N) - \left(\frac{2}{mN^2}\right)\sum_{i=1}^{n}(N-i+1)y_i$$

where persons are ranked in ascending order of y_i.

The Gini coefficient (and the percentile ratios) are not properly decomposable by subgroup into within- and between-group inequality components.

Sen's (1976) welfare index is given by

$$S = m(1-G)$$

Syntax

`ineqdeco` *varname* [*weight*] [`if` *exp*] [`in` *range*] [`,` `bygroup(`*groupvar*`)` `w` `summ`]

`fweight`s and `aweight`s are allowed.

Options

`bygroup(`*groupvar*`)` requests inequality decompositions by population subgroup, with subgroup membership summarized by *groupvar*.

`w` requests calculation of equally-distributed-equivalent incomes and welfare indices in addition to the inequality index calculations.

`summ` requests presentation of `summary, detail` output for *varname*.

Saved results

`S_9010, S_7525`	Percentile ratios p90/p10, p75/p25
`S_im1, S_i0, S_i1, S_i2`	GE(a), for a = −1, 0, 1, 2
`S_ahalf, S_i1, S_a2`	A(e), for e = 0.5, 1, 2

Example

Standard output from `ineqdeco` with only the welfare index option chosen is as follows.

```
. ineqdeco eybhc [fw=wgt], w
Warning: eybhc has 20 values < 0. Not used in calculations

Percentile ratios for distribution of eybhc: all valid obs.
------------------------------------------------------------
p90/p10  p90/p50  p10/p50  p75/p25  p75/p50  p25/p50
------------------------------------------------------------
  4.336    2.063    0.476    2.249    1.474    0.655
Generalized Entropy indices GE(a), where a = income difference
 sensitivity parameter, and Gini coefficient
----------+-----------------------------------------------------------
  All obs |     GE(-1)       GE(0)       GE(1)       GE(2)        Gini
----------+-----------------------------------------------------------
          |    3.66972     0.19386     0.20530     0.36167     0.33263
----------+-----------------------------------------------------------
Atkinson indices, A(e), where e > 0 is the inequality aversion parameter
----------+-----------------------------------
  All obs |     A(0.5)        A(1)        A(2)
----------+-----------------------------------
          |    0.09294     0.17622     0.88009
----------+-----------------------------------
Equally-distributed-equivalent incomes, Yede(e)
----------+-----------------------------------
  All obs |  Yede(0.5)     Yede(1)     Yede(2)
----------+-----------------------------------
          |  212.04836   192.57941    28.03261
----------+-----------------------------------
Social welfare indices, W(e), and Sen´s welfare index
----------+-----------------------------------------------------------
  All obs |        W(0.5)           W(1)           W(2)  mean*(1-Gini)
----------+-----------------------------------------------------------
          |      29.12376        5.26051       -0.03567      156.01453
----------+-----------------------------------------------------------
```

We can examine differences in inequality by tenure group using the command

```
. ineqdeco eybhc [fw=wgt], by(tenure)
Warning: eybhc has 20 values < 0. Not used in calculations

Percentile ratios for distribution of eybhc: all valid obs.
------------------------------------------------------------
p90/p10  p90/p50  p10/p50  p75/p25  p75/p50  p25/p50
------------------------------------------------------------
  4.336    2.063    0.476    2.249    1.474    0.655
Generalized Entropy indices GE(a), where a = income difference
 sensitivity parameter, and Gini coefficient
----------+-----------------------------------------------------------
  All obs |     GE(-1)       GE(0)       GE(1)       GE(2)        Gini
----------+-----------------------------------------------------------
          |    3.66972     0.19386     0.20530     0.36167     0.33263
----------+-----------------------------------------------------------
```

```
Atkinson indices, A(e), where e > 0 is the inequality aversion parameter
----------+-----------------------------------
  All obs |     A(0.5)       A(1)       A(2)
----------+-----------------------------------
          |    0.09294    0.17622    0.88009
----------+-----------------------------------

Subgroup summary statistics, for each subgroup k = 1,...,K:
----------+-----------------------------------------------------------------
Tenure of |
HH        |  Pop. share         Mean     Rel.mean Income share    log(mean)
----------+-----------------------------------------------------------------
 Social r |     0.22858    139.71280      0.59763      0.13661      4.93959
 Other re |     0.07177    215.92972      0.92366      0.06629      5.37495
 Owned:mo |     0.50177    279.24060      1.19448      0.59935      5.63207
 Owned:ou |     0.19789    233.61986      0.99933      0.19775      5.45370
----------+-----------------------------------------------------------------

Subgroup indices: GE_k(a) and Gini_k
----------+-----------------------------------------------------------
Tenure of |
HH        |    GE(-1)       GE(0)       GE(1)       GE(2)        Gini
----------+-----------------------------------------------------------
 Social r |   0.13500     0.09188     0.09317     0.11616     0.22864
 Other re |   0.25743     0.18018     0.17526     0.21131     0.32182
 Owned:mo |   8.32796     0.16025     0.15448     0.19913     0.29406
 Owned:ou |   0.30608     0.22835     0.29114     0.85230     0.35977
----------+-----------------------------------------------------------

Within-group inequality, GE_W(a)
----------+-----------------------------------------------
  All obs |    GE(-1)       GE(0)       GE(1)       GE(2)
----------+-----------------------------------------------
          |   3.63059     0.15953     0.17450     0.33342
----------+-----------------------------------------------

Between-group inequality, GE_B(a):
----------+-----------------------------------------------
  All obs |    GE(-1)       GE(0)       GE(1)       GE(2)
----------+-----------------------------------------------
          |   0.03913     0.03433     0.03079     0.02820
----------+-----------------------------------------------

Subgroup Atkinson indices, A_k(e)
----------+-----------------------------------
Tenure of |
HH        |    A(0.5)        A(1)        A(2)
----------+-----------------------------------
 Social r |   0.04454     0.08779     0.21260
 Other re |   0.08447     0.16488     0.33987
 Owned:mo |   0.07387     0.14807     0.94336
 Owned:ou |   0.11666     0.20415     0.37971
----------+-----------------------------------

Within-group inequality, A_W(e)
----------+-----------------------------------
  All obs |    A(0.5)        A(1)        A(2)
----------+-----------------------------------
          |   0.07903     0.15204     0.69207
----------+-----------------------------------

Between-group inequality, A_B(e)
----------+-----------------------------------
  All obs |    A(0.5)        A(1)        A(2)
----------+-----------------------------------
          |   0.01511     0.02852     0.61059
----------+-----------------------------------
```

Almost 70% of the population are in households owning their own house, and this group is clearly much better off than those in rented accommodations. Average income among owner households with a mortgage is about 20% the population average income, in contrast with average income among social renters which is some 40% below the population average. Average income is lower among owners-outright than among owners with a mortgage, most likely because the former group includes a much higher proportion of older retired people.

According to most of the indices, inequality is greatest for the owned-outright group compared to the others (especially for the more top-sensitive indices such as $\mathrm{GE}(2)$) and it is lowest for the social-renting group. The former result is most likely related to factors such as age, retirement and differential pensions. The latter result is not surprising since, by design, the social housing sector is mainly for "low income" people. Observe that inequality within tenure groups accounts for very much more of total inequality than inequality between tenure groups does.

Repeated application of these decomposition methods to data for several years can be used to account for trends over time in income inequality; see Jenkins (1995) who used subgroup partitions defined by labor market status, age, household composition, etc. to study trends during the 1970s and 1980s. In essence one examines whether trends in overall inequality are more closely related to changes in subgroup inequalities, subgroup mean incomes, or subgroup population shares.

geivars: Generalized Entropy inequality indices, with sampling variances

`geivars` estimates members of the Generalized Entropy class $\mathrm{GE}(a)$ for $a = -1, 0, 1, 2$, see above for definitions, together with their asymptotic sampling variances. Unit record (micro level) data are required.

The formulas for the sampling variances are taken directly from Cowell (1989). His formulas were derived assuming that the income receiving units (households) are treated as a random sample from a bivariate distribution of income and a household weight variable (e.g., household size). It is the assumptions about, and treatment of, weights which causes complexities of estimation of sampling variances. (The issues overlap with, but are not the same as, those addressed by Stata's `svy` programs.)

We require estimates of income inequality among all persons in the household population. In effect there is a random sample of households with "self weighting" by household size, where the weights are similar to Stata's `fweight`s. Thus the variance formulas do not also adjust for the effects of complex survey design features (stratification and clustering), formulas for this case are rather complicated and the subject of current research. These problems do not arise, of course, if the data are unweighted.

Derivation of the formulas for the asymptotic variances use the result that the $\mathrm{GE}(a)$ indices can be written as functions of sample moments. For further details, see Cowell (1989).

`geivars` output includes the estimates of the four indices, and three sets of variance estimates for each index, corresponding to different informational assumptions. V_0 is the variance in the case where both mean income and household size are known. $V_1 (= V_0 + \Delta_1)$ is the variance in the case where the former is not known, and $V_2 (= V_1 + \Delta_2)$ is the variance in the case where both are unknown and estimated from the sample. (Δ_1 and Δ_2 are contributions to the sampling variance arising from relaxing the informational assumptions: see Cowell 1989.) In each case the asymptotic t ratio $= \mathrm{GE}(a)/\sqrt{[V(a)]}$ and associated p value are also reported.

Syntax

geivars *varname* [*weight*] [**if** *exp*] [**in** *range*]

`fweight`s are allowed.

Example

The specialist nature of the variance formulas led me to construct a slightly different version of the 1991 UK dataset in order to match the assumptions. I use the same household income variable `eybhc`, but the data are now organized by household rather than family (the household is the sampling unit in the original survey). The grossing-up weights have been neglected in order to focus on the self-weighting aspect. As a result, the inequality estimates are not comparable with those shown earlier.

In this example, it turns out that the sampling variances of all four inequality indices are all quite small, regardless of which informational assumption is made. These need not be the case in general, especially if the calculations are done for subgroups with relatively few members.

```
. geivars eybhc [fw=number]
Warning: eybhc has 17 values = 0. Not used in calculations
Generalized entropy inequality measures, GE(a), with asym. s.e.s
-----------------------------------------------------------------
   a     |       -1         0         1         2
-----------------------------------------------------------------
GE(a)    |     2.83066   0.18896   0.19095   0.25465

Var0     |     6.51258   0.00156   0.00655   0.00066
s.e.0    |     2.55198   0.03949   0.08094   0.02562
asym. t  |     1.10920   4.78552   2.35920   9.93927
P > |t|  |     0.26739   0.00000   0.01835   0.00000

delta1   |    -0.00176  -0.00050  -0.00645  -0.00043
Var1     |     6.51082   0.00106   0.00010   0.00023
s.e.1    |     2.55163   0.03253   0.01011   0.01506
```

```
asym. t  |     1.10935   5.80936  18.88962  16.90382
P > |t|  |     0.26733   0.00000   0.00000   0.00000

delta2   |    -0.00179  -0.00102  -0.00006  -0.00004
Var2     |     6.50902   0.00003   0.00004   0.00019
s.e.2    |     2.55128   0.00587   0.00664   0.01385
asym. t  |     1.10951  32.16550  28.77771  18.38758
P > |t|  |     0.26726   0.00000   0.00000   0.00000
```

ineqfac: inequality decomposition by factor components

`ineqfac` provides an exact decomposition of the inequality of total income into inequality contributions from each of the factor components of total income. More specifically, given

$$\textit{facvars} = \{\text{factor\_1 factor\_2 } \dots \text{ factor\_}F\}$$

define the variable `totvar` such that for each observation in the dataset,

$$\texttt{totvar} = \sum_{f=1}^{F} \text{factor\_}f$$

Shorrocks (1982a) proved that there was a unique 'decomposition rule' for which inequality in `totvar` across observations could be expressed as the sum of inequality contributions from each of the factor components, and which also satisfied some other basic axioms.

The decomposition rule is the "proportionate contribution of factor f to total inequality", s_f:

$$s_f = \rho_f \sigma(\text{factor\_}f)/\sigma(\texttt{totvar})$$

where ρ_f is the correlation between factor_f and `totvar`, and $\sigma(.)$ is the standard deviation. Equivalently, s_f is the slope coefficient from the regression of factor_f on `totvar`. Observe that for each observation,

$$\sum_{f=1}^{F} s_f = 1$$

Factor components with a positive value for s_f make a disequalizing contribution to inequality in total income; factor components with negative s_f values make an equalizing contribution.

Shorrocks (1982a) shows that choice of the decomposition rule is an issue independent of that concerning which index is used to summarize inequality. However there happens to be a nice link with the case in which inequality is measured using the coefficient of variation, for one can also rewrite s_f as

$$s_f = \rho_f[m(\text{factor\_}f)/m(\texttt{totvar})][\text{CV}(\text{factor\_}f)\text{CV}(\texttt{totvar})]$$

or

$$s_f = \rho_f[m(\text{factor\_}f)/m(\texttt{totvar})][I2(\text{factor\_}f)/I2(\texttt{totvar})]^{.5}$$

where m is the mean, and CV is the coefficient of variation, and $I2$ is half the squared coefficient of variation, or equivalently, $\text{GE}(2)$ as defined earlier.

Thus total inequality can be written in terms of the factor correlations with total income, the factor shares in total income ($= m(\text{factor\_}f)/m(\texttt{totvar})$), and the factor inequalities (summarized using either CV or $I2$).

`ineqfac` reports the estimates for each factor component of: s_f, $S_f = s_f.\text{CV}(\texttt{totvar})$, $m(\text{factor\_}f)/m(\texttt{totvar})$, $\text{CV}(\text{factor\_}f)$, and $\text{CV}(\text{factor\_}f)/\text{CV}(\texttt{totvar})$, plus, optionally, the correlations, means and standard deviations of the factor components and `totvar`. Optionally, inequality is summarized using $I2$ rather than CV.

`ineqfac` was designed as a tool for income distribution analysis in the case where the current sample contains observations on income components for each of a set of income receiving units (e.g., families, households, persons). In this case, *facvars* might include labor income, income from investments and pensions, cash transfers, and so on. See Shorrocks (1982b) and Jenkins (1995) for examples. `ineqfac` may also be applied to summarize and compare the riskiness of portfolios of wealth holdings: s_f has exactly the same form as the "beta coefficient" used in financial analysis.

Syntax

`ineqfac` *facvars* [*weight*] [`if` *exp*] [`in` *range*] [`, stats total(`*totvar*`) i2`]

`fweight`s and `aweight`s are allowed.

Options

`stats` provides the means, standard deviations, and correlations of the factor components and *totvar*.

`total(`*totvar*`)` creates a new variable, *totvar*, equal to the sum of the factor components for each observation.

`i2` summarizes inequality using $I2 = \mathrm{GE}(2)$ rather than CV.

Example

Let us consider how inequality in household money income, `ybhc`, is related to the income sources which comprise it. I distinguish five factor components: `labour`, employment and self-employment earnings; `invst`, income from investments, savings, and private pensions; `socsecb`, cash social assistance and social insurance benefits; `other`, other income; and `deducts`, income taxes and social insurance contributions.

In general, each of the factor components may have negative or zero values. Examples of valid negative values are found most commonly for `deducts`; we assume that taxes are treated as negative income. (If values of variables such as tax payments are recorded as positive in the data, it is the responsibility of the user to create a suitably signed variable prior to using `ineqfac`.) Examples of zero values might occur for, say, `labour`, in observations where no one in the household does paid work, or for `socsecb`, if no one in the household receives any social security benefits.

```
. ineqfac labour invst socsecb other deducts [fw=wgt], stats total(total)
Factor   |     100*s_f       S_f     100*m_f/m      CV_f   CV_f/CV(Total)
---------+-----------------------------------------------------------------
labour   |     77.0372      0.6515     76.0261      1.0414      1.2314
invst    |     27.8958      0.2359     10.2059      4.3230      5.1116
socsecb  |     -5.4941     -0.0465     15.3310      1.1401      1.3481
other    |      1.0902      0.0092      2.1276      5.5795      6.5973
deducts  |     -0.5292     -0.0045     -3.6907      0.5312      0.6280
---------+-----------------------------------------------------------------
Total    |    100.0000      0.8457    100.0000      0.8457      1.0000
---------------------------------------------------------------------------
Note: The proportionate contribution of factor f to inequality of Total,
      s_f = rho_f*sd(f)/sd(Total). S_f = s_f*CV(Total).
      m_f = mean(f). sd(f) = std.dev. of f. CV_f = sd(f)/m_f.

Means, s.d.s and correlations for factors and total income
(sum of wgt is  5.5852e+007)
(obs=6468)
 Variable |         Mean    Std. Dev.          Min          Max
----------+-----------------------------------------------------
   labour |     220.3662     229.4935    -223.1994     2754.562
    invst |     29.58227     127.8837       -97.52      6747.25
  socsecb |     44.43792      50.6651            0      335.534
    other |     6.167069     34.40908    -151.0626       878.31
  deducts |    -10.69765      5.68206       -45.04            0
    Total |     289.8558     245.1361    -123.9898     7740.044

        |   labour    invst  socsecb    other  deducts    Total
--------+------------------------------------------------------
  labour|   1.0000
   invst|   0.0120   1.0000
 socsecb|  -0.5111   0.0179   1.0000
   other|  -0.0518  -0.0129  -0.0373   1.0000
 deducts|  -0.2868  -0.0031   0.0826   0.0111   1.0000
   Total|   0.8229   0.5347  -0.2658   0.0777  -0.2283   1.0000
```

Unsurprisingly, labor earnings are by far the largest component of household income packages, comprising just over three-quarters of total household money income. The next largest components are social security benefits (15% of total income) and investment income (10%). Inequalities in investment income and other income are huge relative to that of the other factor components (see the last two columns). However, inequality contributions tend to be more closely related to factor shares than to factor inequalities or correlations.

According to the Shorrocks decomposition rule, labor earnings has the largest proportionate inequality contribution of all the components, some 77% of total inequality. The second largest proportionate contribution is from investment income, 28%. Observe that taxes and cash transfers have an equalizing effect on total inequality, though relatively small ones.

povdeco: Poverty indices, with decomposition by subgroup

`povdeco` estimates three poverty indices from the Foster, Greer and Thorbecke (1984) class, $\mathrm{FGT}(\alpha)$, plus related statistics (such as mean income among the poor). $\mathrm{FGT}(0)$ is the headcount ratio (the proportion poor); $\mathrm{FGT}(1)$ is the average normalized poverty gap; $\mathrm{FGT}(2)$ is the average squared normalized poverty gap. The larger α is, the greater the degree of poverty aversion (sensitivity to large poverty gaps). Optionally provided are decompositions of these indices by population subgroup. Poverty decompositions by subgroup are useful for providing poverty 'profiles' at a point in time, and for analyzing secular trends in poverty using shift-share analysis. Unit record ('micro' level) data are required.

A more detailed description is as follows. Consider a population of income-receiving units (persons, households or families, and so on), $i = 1, \ldots, n$, with income y_i, and weight w_i. Let $f_i = w_i/N$, where $N = \sum_{i=1}^{n} w_i$. When the data are unweighted, $w_i = 1$ and $N = n$.

The poverty line is z, and the poverty gap for person i is $\max(0, z - y_i)$. Suppose there is an exhaustive partition of the population into mutually-exclusive subgroups $k = 1, \ldots, K$.

The FGT class of poverty indices is given by

$$\mathrm{FGT}(\alpha) = \sum_{i=1}^{n} F_1 \Big[(z - y_i)/z\Big]^{\alpha} I_i$$

where $I_i = 1$ if $y_i < z$ and $I_i = 0$ otherwise.

Each $\mathrm{FGT}(a)$ index can be additively decomposed as

$$\mathrm{FGT}(\alpha) = \sum_{k=1}^{K} v_k \mathrm{FGT}_k(\alpha)$$

where $v_k = N_k/N$ is the number of persons in subgroup k divided by the total number of persons (subgroup population share), and $\mathrm{FGT}_k(\alpha)$, poverty for subgroup k, is calculated as if each subgroup were a separate population.

When subgroup decompositions are requested, `povdeco` also displays, for each k, the following additional subgroup summary statistics: subgroup poverty share, $S_k = v_k \mathrm{FGT}_k(\alpha)/\mathrm{FGT}(\alpha)$, and subgroup poverty risk, $R_k = \mathrm{FGT}_k(\alpha)/\mathrm{FGT}(\alpha) = S_k/v_k$.

Typically one's data are in one of two forms. In the first form, the money incomes for each income-receiving unit i, x_i, are equivalized using an equivalence scale factor, m_i, so that $y_i = x_i/m_i$, and the poverty line is a single (common) value, in the same units as equivalized income, z. This is the case discussed in the description. In the second form, incomes are not equivalized, but there are different poverty lines depending on (for example) household type. Suppose the line for unit i is z_i. Observe that if $z_i = z.m_i$, FGT poverty index calculations based on $\{y_i, z\}$ give exactly the same answers as calculations based on $\{x_i, z_i\}$, $i = 1, \ldots, n$. For the first form, use `pline(#)` to specify the single common poverty line, while for the second form, use `varpl(`*zvar*`)` to specify the poverty lines.

Syntax

`povdeco` *varname* [*weight*] [`if` *exp*] [`in` *range*] `,` { `pline(`*#*`)` | `varpl(`*zvar*`)` } [`bygroup(`*groupvar*`)`]

`fweight`s and `aweight`s are allowed.

The user must supply the poverty line value(s), either as a single number # in `pline(`*#*`)`, or provide the variable name containing the values as *zvar* in `varpl(`*zvar*`)`.

Options

`bygroup(`*groupvar*`)` requests poverty decompositions by population subgroup, with subgroup membership summarized by *groupvar*.

Saved results

S_FGT0	FGT(0), defined above
S_FGT1	FGT(1), defined above
S_FGT2	FGT(2), defined above

Example

Let consider first the case in which there is a common poverty line, taken for illustration to be equal to half average needs-adjusted income, and decompose poverty by tenure subgroups.

```
. local z = .5*`mean´
. povdeco eybhc [fw=wgt], pl(`z´) by(tenure)
Warning: eybhc has 20 values < 0. Used in calculations
Total number of observations = 6468
Weighted total no. of observations = 55851705
Number of observations poor = 1327
Weighted no. of obs poor = 11336320
Mean of eybhc amongst the poor =    86.711
Mean of poverty gaps (poverty line - eybhc) amongst the poor =    29.798
Foster-Greer-Thorbecke poverty indices, FGT(a)
----------+-----------------------------------
  All obs |        a=0         a=1         a=2
----------+-----------------------------------
          |    0.20297     0.05191     0.02387
----------+-----------------------------------
FGT(0): headcount ratio (proportion poor)
FGT(1): average normalised poverty gap
FGT(2): average squared normalised poverty gap
Decompositions by subgroup
--------------------------
Summary statistics for subgroup k = 1,...,K
----------+-----------------------------------------------------------
Tenure of |
HH        |    Pop. share           Mean      Mean|poor  Mean gap|poor
----------+-----------------------------------------------------------
 Social r |       0.22852      139.30740       93.30663       23.20227
 Other re |       0.07194      214.48389       80.45238       36.05652
 Owned:mo |       0.50169      278.40619       74.83694       41.67195
 Owned:ou |       0.19785      232.90508       84.24892       32.25999
----------+-----------------------------------------------------------
Subgroup FGT index estimates, FGT(a)
----------+-----------------------------------
Tenure of |
HH        |        a=0         a=1         a=2
----------+-----------------------------------
 Social r |    0.45587     0.09078     0.03180
 Other re |    0.22032     0.06818     0.03938
 Owned:mo |    0.08128     0.02907     0.01686
 Owned:ou |    0.21313     0.05901     0.02686
----------+-----------------------------------
Subgroup poverty ´share´, S_k = v_k.FGT_k(a)/FGT(a)
----------+-----------------------------------
Tenure of |
HH        |        a=0         a=1         a=2
----------+-----------------------------------
 Social r |    0.51326     0.39964     0.30439
 Other re |    0.07809     0.09449     0.11868
 Owned:mo |    0.20090     0.28095     0.35433
 Owned:ou |    0.20776     0.22492     0.22260
----------+-----------------------------------
```

```
Subgroup poverty ´risk´ = FGT_k(a)/FGT(a) = S_k/v_k

----------+-----------------------------------
Tenure of |
HH        |       a=0         a=1         a=2
----------+-----------------------------------
 Social r |    2.24596     1.74880     1.33198
 Other re |    1.08549     1.31345     1.64976
 Owned:mo |    0.40045     0.56001     0.70628
 Owned:ou |    1.05007     1.13681     1.12510
----------+-----------------------------------
```

The overall proportion of the population poor is 20.3% (as shown also by the xfrac output), the average normalized poverty gap is 0.052, and the average squared normalized gap, 0.024. The decomposition shows that subgroup poverty status is associated with average income, whichever index is used. For example, the group with the lowest average income, social renters, also have the highest poverty rate. And those with the highest average income, owners with a mortgage, also have the lowest poverty rate. Interestingly, however, average income among poor owners with a mortgage is lower than average income among poor social renters, 74 pounds per week compared with 93 (and hence their poverty gaps are larger). This helps explain why it is that although social renters' poverty share is about one half according to the headcount ratio, $\mathrm{FGT}(0)$, it is rather smaller when one moves to the measures sensitive to how poor people are (their poverty risks are also smaller). When one uses the poverty gap measures, the poverty share and poverty risk of owners with a mortgage becomes markedly larger.

To illustrate use of the alternative poverty line specification, let us now work with money income ybhc (rather than eybhc which is needs-adjusted), and suppose that the household type-specific poverty line is given by the former poverty line multiplied by the household equivalence scale rate (hes_bhc). To get results exactly the same as shown above, one would simply type the following:

```
. ge plinevar = `z´*hes_bhc
. povdeco ybhc [fw=wgt], varpl(plinevar) by(tenure)
```

Concluding remarks

The aim of this insert has been to make preparation of many common income distribution summary statistics a matter of routine. These numerical summaries should usually be accompanied by graphical ones and it is hoped that glcurve, Jenkins and Van Kerm (1999), should help with these.

The most notable omission from the program calculations presented here is systematic derivation of sampling variances for key statistics (apart from those in geivars). This reflects the state of the income distribution literature; the required formulas either do not yet exist or have only recently been developed. The treatment of different kinds of weights, and the interaction of 'self-weighting' features with survey design aspects, raises several complicated issues in this context which have yet to be resolved.

Nonetheless, it must also be said that conclusions drawn are likely to be at least as sensitive to other factors as to sampling ones. For example, there are important consequences of choosing different equivalence scales, definitions of income and income-receiving unit, and different treatments of rogue outliers and zero and negative incomes. Luckily, Stata is already well-suited for examining these data issues.

Acknowledgments

This work forms part of the scientific research program of the Institute for Social and Economic Research, and was supported by core funding from the University of Essex and the UK Economic and Social and Economic Research Council. The programs are revisions and extensions of some presented at the 4th UK Stata Users' Group meeting.

References

Atkinson, A. B. 1970. On the measurement of inequality. *Journal of Economic Theory* 2: 244–63.

Blackorby, C., D. Donaldson, and M. Auersperg. 1981. A new procedure for the measurement of inequality within and between population subgroups. *Canadian Journal of Economics* XIV: 665–85.

Cowell, F. A. 1989. Sampling variance and decomposable inequality measures. *Journal of Econometrics* 42: 27–41

——. 1995. *Measuring Inequality*. 2d ed. Prentice Hall/Harvester–Wheatsheaf: Hemel Hempstead.

Department of Social Security. 1993. *Households Below Average Income 1979–1990/91* HMSO, London.

Foster, J. E., J. Greer, and E. Thorbecke. 1984. A class of decomposable poverty indices. *Econometrica* 52: 761–766.

Goldstein, R. 1995. sg31: Measures of diversity: absolute and relative. *Stata Technical Bulletin* 23: 23–26. Reprinted in *Stata Technical Bulletin Reprints*, vol. 4, pp. 150–154.

Goodman, A., and S. Webb. 1994. For Richer, for Poorer. The Changing Distribution of Income in the United Kingdom, 1961–91. Commentary No. 42, Institute for Fiscal Studies, London. Abridged version in: *Fiscal Studies* 15: 29–62.

Jenkins, S. P. 1991. The measurement of income inequality. In *Economic Inequality and Poverty: International Perspectives*, ed. L. Osberg. Armonk NY: M. E. Sharpe.

Jenkins, S. P. and P. Van Kerm. 1995. Accounting for inequality trends: decomposition analyses for the UK, 1971–86. *Economica* 62: 29–63.

——. 1997. Trends in real income in Britain: a microeconomic analysis. *Empirical Economics* 22: 483–500.

——. 1999. sg107: Generalized Lorenz curves and related graphs. *Stata Technical Bulletin* 48: 25–29.

Lambert, P. J. 1993. *The Distribution and Redistribution of Income: A Mathematical Analysis*. 2d ed. Manchester University Press: Manchester and New York.

Sen, A. K. 1976. Real national income. *Review of Economic Studies* 43: 19–39.

Shorrocks, A. F. 1982a. Inequality decomposition by factor components. *Econometrica* 50: 193–212.

——. 1982b. The impact of income components on the distribution of family incomes. *Quarterly Journal of Economics* 98: 311–326.

——. 1984. Inequality decomposition by population subgroups. *Econometrica* 52: 1369–1388.

Whitehouse, E. 1995. sg30: Measures of inequality. *Stata Technical Bulletin* 23: 20–23. Reprinted in *Stata Technical Bulletin Reprints*, vol. 4, pp. 146–150.

sg105 Creation of bivariate random lognormal variables

Stephen P. Jenkins, University of Essex, UK, stephenj@essex.ac.uk

Description

`mkbilogn` is a program for the creation of bivariate random normal variables. More precisely it creates random variables, X_1 and X_2, drawn from a bivariate lognormal distribution defined as follows. X_1 and X_2 are such that, as $n \to \infty, x_1 = \log(X_1)$ and $x_2 = \log(X_2)$ are bivariate normal distributed with means m_1, and m_2, standard deviations s_1, and s_2, and correlation r. The parameters of the distribution can be optionally chosen by the user, or default to the values specified below.

The program applies methods proposed in the Stata FAQ archive:

`http://www.stata.com/support/faqs/stat/mvnorm.html`

Syntax

`mkbilogn` *var1 var2* [, `r(`*#*`)` `m1(`*#*`)` `s1(`*#*`)` `m2(`*#*`)` `s2(`*#*`)`]

Options

`r(`*#*`)` correlation of $\ln($*var1*$)$ and $\ln($*var2*$)$; default is .5.

`m1(`*#*`)` mean of $\ln($*var1*$)$; default is 0.

`s1(`*#*`)` standard deviation of $\ln($*var1*$)$; default is 1.

`m2(`*#*`)` mean of $\ln($*var2*$)$; default is 0.

`s2(`*#*`)` standard deviation of $\ln($*var2*$)$; default is 1.

Example

```
. clear
. set obs 10000
obs was 0, now 10000
. mkbilogn y1 y2, r(.3) m1(1) s1(2) m2(3) s2(4)
Creating 2 r.v.s X1 X2  s.t. x1=log(X1), x2=log(X2) are bivariate
 Normal with mean(x1) = 1 ; mean(x2) = 3 ; s.d.(x1) = 2 ;
 s.d.(x2) = 4 ; corr(x1,x2) = .3
. generate ly1 = ln(y1)
. generate ly2 = ln(y2)
```

```
. summarize
Variable |     Obs        Mean   Std. Dev.       Min        Max
---------+-----------------------------------------------------
      y1 |   10000    21.41347   217.5634   .0012415   19863.65
      y2 |   10000    34875.93    1093960   1.19e-06   9.79e+07
     ly1 |   10000    1.040054   1.990629  -6.691414   9.896646
     ly2 |   10000    3.095062    4.04193  -13.64018   18.39969

. corr
(obs=10000)
        |       y1       y2      ly1      ly2
--------+------------------------------------
      y1|   1.0000
      y2|   0.0023   1.0000
     ly1|   0.2078   0.0270   1.0000
     ly2|   0.0585   0.1011   0.2963   1.0000
```

Saved results

Two new variables (*var1*, *var2*) are added to the current dataset.

Acknowledgments

This work forms part of the scientific research program of the Institute for Social and Economic Research, and was supported by core funding from the University of Essex and the UK Economic and Social and Economic Research Council. The program was created for use in joint work with Frank Cowell (London School of Economics) developing asymptotic standard errors for inequality indices in the weighted data case.

sg106	Fitting Singh–Maddala and Dagum distributions by maximum likelihood

Stephen P. Jenkins, University of Essex, UK, stephenj@essex.ac.uk

Introduction

Economists and statisticians sometimes find it useful to fit parametric functional forms to data on a variable. `smfit` fits the three-parameter Singh–Maddala (1976) distribution and `dagumfit` fits the Dagum (1977, 1980) distribution, in each case by maximum likelihood (ML) methods, to a distribution of a random variable `incvar`, where unit record observations on `incvar` are available. The Singh–Maddala distribution is also known as the Burr Type 12 distribution and the Dagum distribution as the Burr Type 3 distribution. These three-parameter distributions have been shown to provide a good fit to empirical income data relative to other parametric functional forms; see McDonald (1984), for example. For derivation of Lorenz orderings of pairs of income distributions in terms of their Singh–Maddala and Dagum parameters, see Wifling and Kraemer (1993) and Kleiber (1996). Of course the Singh–Maddala and Dagum distributions might be suitable for describing any skewed variable, not just income.

Programmers may find `smfit` and `dagumfit` of interest because they are examples of the application of `ml` in a case which is unlike a regression model (there are no covariates or dependent variable in the conventional sense).

The Singh–Maddala distribution

The Singh–Maddala distribution has distribution function

$$F(x) = 1 - \left[\frac{1}{1+(x/b)^a}\right]^q$$

where $a \geq 0$, $b \geq 0$, $q > 1/a$ are parameters, for random variable $X \geq 0$ (income). The parameters a and q are the key distributional shape parameters; b is a scale parameter.

Letting $z = 1 + (x/b)^a$, then $F(x) = 1 - z^{-q}$, and the probability density function is

$$f(x) = (aq/b)z^{-(q+1)}(x/b)^{(a-1)}$$

The likelihood function for a sample of incomes is specified as the product of the densities for each person (weighted where relevant), and is maximized by `smfit` using Stata's `deriv0` (numerical derivatives) method. In fact, transformations of the three parameters are estimated (to impose the necessary restrictions) and the parameters derived from these.

The formulas used to derive the distributional summary statistics presented (optionally) are as follows. The rth moment about the origin is given by

$$b^r B(1+r/a, q-r/a)/B(1,q)$$

where $B(u,v)$ is the Beta distribution $= G(u)G(v)/G(u+v)$ and G is the gamma function (`exp(lngamma(.))` in Stata), which by substitution and using the result that $G(1)=1$, implies that the moments can be written

$$b^r G(1+r/a)G(q-r/a)/G(q)$$

and hence

$$E(X) = bG(1+1/a)G(q-1/a)/G(q)$$
$$\mathrm{Var}(X) = b^2 G(1+2/a)G(q-2/a)/G(q) - (E(X))^2$$

from which the standard deviation and half the squared coefficient of variation can be derived. The percentiles are derived by inverting the distribution function

$$x_p = b[(1-p)^{(-1/q)} - 1]^{(1/a)}$$

for each $p = F(x_p)$.

The Gini coefficient of inequality, Gini, is given by

$$1 - \mathrm{Gini} = G(q)G(2q-1/a)/[G(q-1/a)G(2q)]$$

The Lorenz curve ordinates $L(p)$ at each $p = F(x_p)$ use the Beta cdf, `ibeta(.)` in Stata:

$$L(p) = \mathtt{ibeta}(1+1/a, q-1/a, 1-(1-p)^{(1/q)})$$

Syntax

`smfit` *incvar* [*weight*] [`if` *exp*] [`in` *range*] [`,` `stats` `cdf(`*cdfname*`)` `pdf(`*pdfname*`)`
`level(`*#*`)` `nolog` `trace` `a0(`*#*`)` `b0(`*#*`)` `q0(`*#*`)`]

`fweight`s and `aweight`s are allowed.

To reset problem-size limits, see `help matsize`.

Options

`stats` displays selected distributional statistics implied by the Singh–Maddala parameter estimates; percentiles, cumulative shares of total income at percentiles (i.e., the Lorenz curve ordinates), the mean, standard deviation, variance, half the coefficient of variation squared, Gini coefficient, and percentile ratios $p90/p10$, $p75/p25$.

`cdf(`*cdfname*`)` creates a new variable *cdfname* containing the estimated Singh–Maddala cdf value $F(x)$ for each x in the dataset.

`pdf(`*pdfname*`)` creates a new variable *pdfname* containing the estimated Singh–Maddala pdf value $f(x)$ for each x in the dataset.

`level(`*#*`)` specifies the confidence level, in percent, for confidence intervals. The default is level(95) or as set by `set level;` see [U] **26.4 Specifying the width of confidence intervals**.

`nolog` suppresses the iteration logs.

`trace` reports the current value of the estimated parameters at each iteration; see [R] **maximize**.

`a0(`*#*`)`, `b0(`*#*`)`, `q0(`*#*`)` allow the user to specify starting values for the Singh–Maddala parameters. Default starting values are $a=2$, $q=2$, and $b=$ sample mean of *incvar*.

Saved results

The global macros set by `ml post`, plus

`S_a`, `S_b`, `S_q`	estimated parameters a, b, q, respectively

Access to estimated coefficients (transformations of the parameters) and their standard errors are available in the usual way: see [U] **20.5 Accessing coefficients and standard errors**, and [R] **matrix get**.

The Dagum distribution

The Dagum distribution has distribution function

$$F(x) = \left[1 + hx^{-d}\right]^{-b}$$

where $b > 0$, $h > 0$, $d > 1/b$ are parameters, for random variable $X > 0$ (income). Parameters b and d are the key distributional shape parameters; h is a scale parameter.

The probability density function is

$$f(x) = [(bdh)x^{(-d-1)}]/[1 + hx^{(-d)}]^{(b+1)}$$

The likelihood function for a sample of incomes is specified as the product of the densities for each person (weighted where relevant), and is maximized by `dagumfit` using Stata's `deriv0` (numerical derivatives) method. Transformations of the 3 parameters are estimated (to impose the necessary restrictions) and the parameters derived from these.

The formulas used to derive the distributional summary statistics presented (optionally) are as follows. The rth moment about the origin is given by

$$[bh^{(r/d)}]B(1 - r/d, b + r/d)$$

By substitution and using the result that $G(1) = 1$, implies that the moments can be written

$$bh^{(r/d)}G(1 - r/d)G(b + r/d)/G(b + 1)$$

and hence

$$E(X) = [bh^{(1/d)}]G(1 - 1/d)G(b + 1/d)/G(b + 1)$$
$$\text{Var}(X) = [bh^{(2/d)}]G(1 - 2/d)G(b + 2/d)/G(b + 1) - (E(X))^2$$

from which the standard deviation and half the squared coefficient of variation can be derived. The percentiles are derived by inverting the distribution function:

$$x_p = h^{(1/d)}[p^{(-1/b)} - 1]^{(-1/d)}$$

for each $p = F(x_p)$.

The Gini coefficient of inequality is given by

$$1 - \text{Gini} = [G(b)G(2b + 1/d)]/[G(2b)G(b + 1/d)]$$

The Lorenz curve ordinates $L(p)$ at each $p = F(x_p)$ use the Beta cdf

$$L(p) = \texttt{ibeta}(b + 1/d, 1 - 1/d, p^{(1/b)})$$

Syntax

`dagumfit` *incvar* [*weight*] [`if` *exp*] [`in` *range*] [, `stats` `cdf(`*cdfname*`)` `pdf(`*pdfname*`)` `level(`*#*`)` `nolog` `trace` `b0(`*#*`)` `d0(`*#*`)` `h0(`*#*`)`]

`fweight`s and `aweight`s are allowed.

To reset problem-size limits, see `help matsize`.

Options

`stats` displays selected distributional statistics implied by the Dagum model parameter estimates: percentiles, cumulative shares of total income at percentiles (i.e., the Lorenz curve ordinates), the mean, standard deviation, variance, half the coefficient of variation squared, Gini coefficient, and percentile ratios $p90/p10$, $p75/p25$.

`cdf(`*cdfname*`)` creates a new variable *cdfname* containing the estimated Dagum cdf value $F(x)$ for each x.

`pdf(`*pdfname*`)` creates a new variable *pdfname* containing the estimated Dagum pdf value $f(x)$ for each x.

`level(`*#*`)` specifies the confidence level, in percent, for confidence intervals. The default is level(95) or as set by `set level;` see [U] **26.4 Specifying the width of confidence intervals**.

`nolog` suppresses the iteration logs.

`trace` reports the current value of the estimated parameters at each iteration. See [R] **maximize**.

`b0(`*#*`)`, `d0(`*#*`)`, `h0(`*#*`)` allow the user to specify starting values for the Dagum parameters. Default starting values are $b = \exp(4)$, $d = \exp(0.1)$, and $h = 1 + \exp(13)$.

Saved results

The global macros set by `ml post`, plus

`S_b, S_d, S_h`	estimated parameters b, d, h, respectively

Access to estimated coefficients (transformations of the parameters) and their standard errors are available in the usual way; see [U] **20.5 Accessing coefficients and standard errors**, and [R] **matrix get**.

Examples

The illustrative examples use the same income distribution data as described in Jenkins (1999). The income variable is `eybhc` with `fweight` variable `wgt`.

In order to compare the results of `smfit` and `dagumfit`, the former is run excluding nonpositive values of `eybhc`. The Singh–Maddala distribution is defined for nonnegative incomes but the Dagum distribution only for positive incomes. The results are as follows:

```
. smfit eybhc [fw = wgt] if eybhc>0, stats cdf(smF) pdf(smf)
Iteration 0:  Log Likelihood = -40547.317
Iteration 1:  Log Likelihood = -40062.416
Iteration 2:  Log Likelihood = -39888.368
Iteration 3:  Log Likelihood = -39879.841
Iteration 4:  Log Likelihood = -39879.785
Iteration 5:  Log Likelihood = -39879.785
ML fit of Singh-Maddala distribution                 Number of obs    =    6448
                                                     Model chi2(0)    =       .
                                                     Prob > chi2      =       .
Log Likelihood = -39879.7845655
------------------------------------------------------------------------------
   eybhc |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
p1       |
   _cons |   .5637748   .0298546     18.884   0.000        .505261    .6222887
---------+--------------------------------------------------------------------
p2       |
   _cons |   5.357418   .0291111    184.033   0.000       5.300361    5.414475
---------+--------------------------------------------------------------------
p3       |
   _cons |    .178296   .0513498      3.472   0.001       .0776523    .2789397
------------------------------------------------------------------------------
```

```
a = 1+exp(p1) =    2.75729; std. err. =   0.05246; z =  52.55669
b = 1+exp(p2) =  213.17639; std. err. =   6.17669; z =  34.51304
q =   exp(p3) =    1.19518; std. err. =   0.06137; z =  19.47428

Singh-Maddala model estimates for distribution of eybhc
------------------------------------------------------------

    Percentiles Cumulative shares of
                total eybhc (Lorenz ordinates)

 1%   37.73642       0.00119
 5%   68.58355       0.01072
10%   89.78419       0.02785
20%  120.04293       0.07317
25%  132.97006       0.10032
30%  145.33266       0.13018
40%  169.71135       0.19776
50%  195.34499       0.27599         Mean        233.07720
60%  224.44103       0.36587         Std. Dev.   175.49745
70%  260.51414       0.46956
75%  283.25851       0.52781         Variance    30799.35345
80%  311.44974       0.59147         Half CV^2     0.28347
90%  404.94247       0.74246         Gini coeff.   0.33268
95%  513.02045       0.83928         p90/p10       4.51018
99%  855.57398       0.94708         p75/p25       2.13024

. dagumfit eybhc [fw = wgt], stats cdf(dagumF) pdf(dagumf)

Warning: eybhc has 20 values < 0. Not used in calculations
Iteration 0:  Log Likelihood = -2537735.5
(nonconcave function encountered)
Iteration 1:  Log Likelihood = -57019.692
(nonconcave function encountered)
Iteration 2:  Log Likelihood =  -45368.91
Iteration 3:  Log Likelihood = -41395.382
(nonconcave function encountered)
Iteration 4:  Log Likelihood = -41065.244
Iteration 5:  Log Likelihood = -40128.555
Iteration 6:  Log Likelihood = -39919.827
Iteration 7:  Log Likelihood = -39894.729
Iteration 8:  Log Likelihood = -39884.318
Iteration 9:  Log Likelihood = -39882.885
Iteration 10:  Log Likelihood = -39882.863
Iteration 11:  Log Likelihood = -39882.863
Iteration 12:  Log Likelihood = -39882.863

ML fit of Dagum distribution                        Number of obs    =    6448
                                                    Model chi2(0)    =       .
                                                    Prob > chi2      =       .
Log Likelihood = -39882.8626763

------------------------------------------------------------------------------
   eybhc |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
p1       |
   _cons |  -.1156061   .0447439     -2.584   0.010      -.2033025   -.0279097
---------+--------------------------------------------------------------------
p2       |
   _cons |   1.113663   .0194751     57.184   0.000       1.075493    1.151834
---------+--------------------------------------------------------------------
p3       |
   _cons |   16.22055   .3753564     43.214   0.000       15.48486    16.95623
------------------------------------------------------------------------------

b = exp(p1)   =    0.89083; std. err. =   0.03986; z =  22.34942
d = exp(p2)   =    3.04549; std. err. =   0.05931; z =  51.34757
h = 1+exp(p3) =  11078840.30261; std. err. = 4158512.76880; z =   2.66414

Dagum model estimates for distribution of eybhc
------------------------------------------------------------

    Percentiles Cumulative shares of
                total eybhc (Lorenz ordinates)

 1%   37.73208       0.00117
 5%   68.95422       0.01067
10%   90.29702       0.02777
20%  120.51138       0.07299
25%  133.33829       0.10004
30%  145.57025       0.12974
40%  169.62770       0.19686
50%  194.89853       0.27442         Mean        234.77654
```

```
60%  223.64366       0.36338         Std. Dev.   188.66945
70%  259.48672       0.46592
75%  282.24383       0.52353         Variance    35596.15976
80%  310.64428       0.58654         Half CV^2     0.32290
90%  406.43660       0.73643         Gini coeff.   0.33721
95%  520.03530       0.83332         p90/p10       4.50111
99%  894.92777       0.94313         p75/p25       2.11675
```

The likelihood values and estimates of the percentiles, inequality indices and other distribution parameters are remarkably similar for both models.

All the estimates are also very similar to their nonparametric counterparts. For example, the nonparametric estimate of the Gini coefficient is 0.333 and of the $\mathrm{GE}(2)$ index (half the squared coefficient of variation), 0.362: see the output from `ineqdeco` in Jenkins (1999). Other nonparametric statistics can be derived by `summary, detail`:

```
. summarize eybhc [fw=wgt] if eybhc>0, detail
                   Equiv. net income BHC
-------------------------------------------------------------
      Percentiles      Smallest
 1%    41.10482       .0076653
 5%      79.116       1.938724
10%    92.79689       2.631398       Obs            55687900
25%    127.8417       2.808512       Sum of Wgt.    55687900

50%     195.036                      Mean           233.7762
                       Largest       Std. Dev.      198.8109
75%    287.5094       1846.438
90%     402.397       2013.499       Variance       39525.79
95%    504.1051       3024.663       Skewness       14.44232
99%     818.264       7740.044       Kurtosis       484.1126
```

The greatest difference between the parametric and nonparametric estimates is at the very bottom and, especially, the very top of the distribution. The latter difference is almost certainly due to the presence of a single high income outlier; note for example the large under-estimation of the top-sensitive index $\mathrm{GE}(2) =$ half the squared coefficient of variation. In some cases, one might argue that the parametric estimates were more reliable on the grounds that income data in the extreme tails of the distribution are not reliable.

Goodness-of-fit may also be assessed graphically using probability plots. The `psm`, `qsm`, `pdagum`, and `qdagum` programs written by Cox (1999) provide these using estimates produced by `smfit` and `dagumfit`.

The similarity of estimates in the example appears contrary to the claim sometimes made in the literature that the Dagum distribution typically provides a better fit than the Singh–Maddala one. Results can perhaps be reconciled by observing that in virtually all cases reported to date, estimates have been derived from grouped (banded) income data rather than unit record data as here.

Other criteria besides goodness-of-fit may be relevant to a choice between `smfit` and `dagumfit`. The main difference I have found is in convergence stability and time. In all the applications I have experimented with, `smfit` has converged quickly in only a few iterations from the default starting values. By contrast, `dagumfit` typically took many more iterations and in fact sometimes failed to converge using the default starting values (try fitting the Dagum distribution to the variable price in `auto.dta`). In the illustration shown above, `smfit` took about a minute to converge using a Pentium P1/166 PC running Stata 5.0 for Windows 95, but `dagumfit` required almost 18 minutes. Part of the problem is that it is difficult to specify good default starting values for `dagumfit`. In all the cases where the program did not converge, experimentation with a range of alternative starting values led eventually to convergence. Use of the `trace` option is therefore recommended in all initial fits.

Acknowledgments

This work forms part of the scientific research program of the Institute for Social and Economic Research, and was supported by core funding from the University of Essex and the UK Economic and Social and Economic Research Council. The programs are revisions and extensions of some presented at the 4th UK Stata Users' Group meeting. Markus Jäntti and Nick Cox made helpful comments on earlier versions of the programs.

References

Cox, N. J. 1999. gr35: Diagnostic plots for assessing Singh–Maddala and Dagum distributions fitted by MLE. *Stata Technical Bulletin* 48: 2–4.

Dagum, C. 1977. A new model of personal income distribution: specification and estimation. *Economie Appliquée* 30: 413–437.

——. 1980. The generation and distribution of income, the Lorenz curve and the Gini ratio. *Economie Appliquée* 33: 327–367.

Jenkins, S. P. 1999. sg104: Analysis of income distributions. *Stata Technical Bulletin* 48: 4–18.

Kleiber, C. 1996. Dagum vs. Singh–Maddala income distributions. *Economics Letters* 53: 265–268.

McDonald, J. B. 1984. Some generalized functions for the size distribution of income. *Econometrica* 52: 647–663.

Singh, S. K. and G. S. Maddala. 1976. A function for the size distribution of income. *Econometrica* 44: 963–970.

Wifling, B. and W. Kraemer. 1993. The Lorenz-ordering of Singh–Maddala income distributions. *Economics Letters* 43: 53–57.

sg107 Generalized Lorenz curves and related graphs

Stephen P. Jenkins, ISER, University of Essex, UK, stephenj@essex.ac.uk
Philippe Van Kerm, GREBE, University of Namur, Belgium, philippe.vankerm@fundp.ac.be

Generalized Lorenz curves (henceforth GLC's) are frequently used by economists as a tool for representing and comparing empirical distributions, typically of income. The GLC of a continuously distributed variable y plots the cumulative total of y divided by total population size against $p = F(y)$, the cumulative distribution function. Mathematically, point coordinates $[p(y), \mathrm{GL}(p(y))]$ of the GLC are given by

$$p(y) = F(y), \qquad \mathrm{GL}(p(y)) = \int_0^y xf(x)dx$$

with $f(x) = dF(x)/dx$. If the GLC coordinates are computed using a series of discrete data points $y_1, \ldots, y_N$, where observations have been ordered so that $y_1 \leq y_2 \leq \ldots \leq y_N$, one obtains

$$p(y_i) = \frac{i}{N}, \qquad \mathrm{GL}(p(y_i)) = \frac{\sum_{j=1}^{i} y_j}{N}$$

and analogously for weighted data.

GLCs of income distributions have attractive properties, related to checks of "welfare dominance" and "poverty dominance." For example, if one were to draw the GLCs for two countries A and B, and found that the GLC for A lay above the GLC for B at each value of p, then one may conclude that welfare is higher and poverty lower in distribution A compared to distribution B, according to all measures of welfare and poverty satisfying a standard set of desirable axioms. See for example Shorrocks (1983) or the texts by Cowell (1995) or Lambert (1993) for further details.

A series of graphical instruments are closely related to GLCs, some of them perhaps better known. The most obvious is the Lorenz curve. The Lorenz curve of a variable y plots the cumulative share of y against $p = F(y)$, the cumulative distribution function. The LC coordinates for the corresponding discrete case are thus $p(y_i) = i/N \;\; L(p(y_i)) = \sum_{j=1}^{i} y_j / \sum_{j=1}^{N} y_j$ The Lorenz curve of y is simply the GLC of y/μ_y where μ_y is the mean of y. If two Lorenz curves do not intersect, one may conclude that inequality in the distribution with the higher curve is lower than inequality in the other distribution, according to all standard inequality indices (e.g., all those in the Atkinson and Generalized Entropy classes, and the Gini coefficient).

Imagine now that one plots the cumulative share of some other variable s (observed jointly with y) against $p = F(y)$, the cumulative distribution function. The picture obtained is the concentration curve of s against y. Say we observe a set of pairs $(y_1, s_1), \ldots, (y_N, s_N)$ indexed in such a way that $y_1 \leq y_2 \leq \ldots \leq y_N$, the coordinates of the concentration curve are $p(y_i, s_i) = i/N$, $C(p(y_i, s_i)) = \sum_{j=1}^{i} s_j / \sum_{j=1}^{N} s_j = \sum_{j=1}^{i} s_j/\mu_s/N$, where μ_s is the mean of s. Concentration curves are particularly useful for the analysis of taxes, benefits, and income redistribution (see, for example, Lambert 1993).

The so-called TIP (Three I's of Poverty) curves can also be easily introduced in this framework (Jenkins and Lambert 1997). Let z be some threshold and define the variables g as $g = z - y$ and r as $r = 1 - (y/z) = g/z$. The coordinates of the TIP curve are

$$p(y_i, z) = \frac{i}{N}, \mathrm{TIP}(p(y_i, z)) = \frac{\sum_{j=1}^{i} g_j}{N}$$

whereas the coordinates of the TIP of normalized poverty gaps are

$$p(y_i, z) = \frac{i}{N}, \mathrm{TIP}_n(p(y_i, z)) = \frac{\sum_{j=1}^{i} r_j}{N}$$

TIP curves are useful for simultaneously displaying the several dimensions of poverty in a single picture; incidence, intensity and inequality. Moreover, configurations of TIP curves are informative about "poverty dominance" for most indices of poverty which satisfy a standard set of desirable axioms.

`glcurve` greatly facilitates the drawing of all these graphs and permits straightforward visual dominance checks.

Syntax

`glcurve` *varname* [*weight*] [`if` *exp*] [`in` *range*] [`,` `pvar(`*pvarname*`)` `glvar(`*glvarname*`)` `sortvar(`*svarname*`)` `by(`*groupvar*`)` `split` `nograph` `replace` *graph_options*]

`aweight`s and `fweight`s are allowed.

Options

`pvar(`*pvarname*`)` generates the variable *pvarname* containing the x-ordinates of the created Generalized Lorenz curve.

`glvar(`*glvarname*`)` generates the variable *glvarname* containing the y-ordinates of the created Generalized Lorenz curve.

`sortvar(`*svarname*`)` specifies the variable by which the data are sorted before the ordinates are computed. By default, the data are sorted in ascending order of *varname*. If the `sortvar` option is specified, sorting and cumulation are in ascending order of *svarname*.

`by(`*groupvar*`)` specifies that the y-ordinates are to be computed separately for each subgroup defined by *groupvar*. *groupvar* must be numeric.

`split` [to be used only in conjunction with `by(`*groupvar*`)`] specifies that a series of new variables is generated containing the Generalized Lorenz y-ordinates for each sub-group specified in `by(`*groupvar*`)`. When `split` is specified, the string in `glvar(`*glvarname*`)`, truncated after 4 characters, is used as a prefix to create the new variables *glva_x1*, *glva_x2*, ... where *x1*, *x2*, ... are the values taken by *groupvar*. To avoid problems, the number of digits taken by the observations in *groupvar* should be at most 3 (otherwise the length of *glva_* must be reduced to fewer than 5 characters accordingly).

`nograph` avoids the automatic display of a crude graph made out of the created variables. `nograph` is assumed if `by(.)` is specified without `split`.

`replace` allows the variables *pvarname* and *glvarname* to be overwritten if existing names are specified in `pvar(.)` and `glvar(.)`. *pvarname* and *glvarname* must otherwise be new variable names.

graph_options are any of the options allowed with `graph, twoway`; see help for `graph`.

Examples

Given the definitions outlined earlier, it is straightforward to understand how `glcurve` works. The generated variables *pvarname* and *glvarname* are simply such that *pvarname[i]*=p(*varname[i]*) and *glvarname[i]*=gl(p(*varname[i]*)) with the operators `p(.)` and gl(.) as defined above and with the *i*s assigned so that *svarname[1]*$\leq$*svarname[2]*$\leq \ldots \leq$ *svarname[_N]*. Whenever the `by(.)` option is specified, the same construct holds but the ordinates are computed for each distinct subgroup designated by *groupvar* (population totals converted to subgroup totals). As should be clear from their definitions, Lorenz curves, concentration curves as well as TIP curves can be readily obtained as long as the *svarname* is appropriately chosen and by first applying a simple transformation to the variable of interest (e.g., for the concentration or Lorenz curves, dividing by the overall mean).

Let us give a few examples. The dataset `subcvse.dta` (extracted from a Belgian survey on low income households, the CVSEW—see the `notes` of `subcvse.dta`) provided with this insert contains four variables; a (single adult equivalent) household income measure (`eqinc`), an indicator of the sex of the household head (`headfem`), an indicator of the home tenancy status of the household (`owner`) and the amount of child benefits received by the household (`chpay`).

Suppose we wish to compare welfare levels between female-headed households and male-headed households. We can draw the Generalized Lorenz curves of the two groups by typing

```
. glcurve eqinc, by(headfem) split xlabel(0,0.25,0.50,0.75,1) ylabel
```

which results in the drawing of Figure 1 (the GLC for male-headed households is the dashed curve).

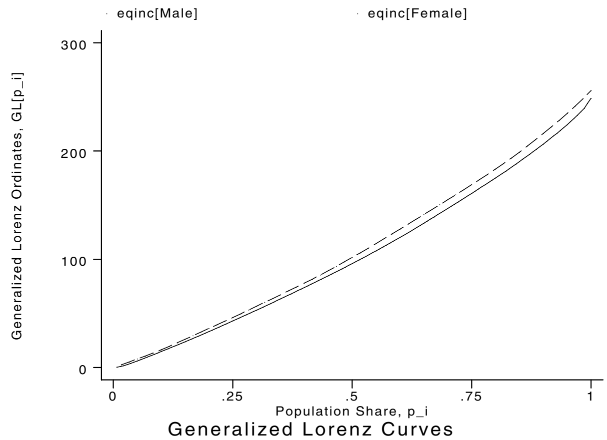

Figure 1. Generalized Lorenz curves for household incomes by head of household gender.

One may prefer to focus on comparisons of Lorenz curves for the two groups. In this case, we should first type the following in order to construct the income measure divided by the relevant subgroup average:

```
. generate eqinc_m = eqinc

. for 0 1,l(n): su eqinc_m if headfem==@ // replace
> eqinc_m=eqinc_m/_result(3) if headfem==@
```

We can then build and draw the Lorenz curves, together with the 45 degree line which corresponds to the Lorenz curve for a perfectly equal distribution, with the following commands:

```
. glcurve eqinc_m, glvar(lc) pvar(p) by(headfem) split nograph

. graph lc_* p p ,s(...) c(lll) xlabel(0,0.25,0.50,0.75,1)
> ylab(0,0.25,0.50,0.75,1) yline(0,1) xline(0, 1) noaxis
```

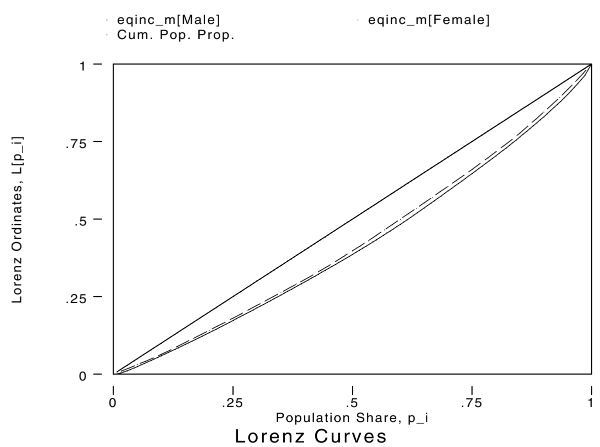

Figure 2. Comparing Lorenz curves for the two groups in Figure 1.

In order to illustrate the use of the `sortvar(.)` option, let us draw now a Concentration curve. Suppose we wish to see how child benefits are distributed relative to the income distribution. Let us draw the Concentration curve of `chpay` (solid line) along with the Lorenz curve of `eqinc` (dashed line).

```
. summarize eqinc

. replace eqinc_m = eqinc/_result(3)

. glcurve eqinc_m, gl(lc) p(p) replace nograph

. summarize chpay

. generate mchpay = chpay/_result(3)

. glcurve mchpay , gl(cc) sort(eqinc_m) nograph

. graph lc cc p p, c(lll) s(...) xlabel(0,0.25,0.50,0.75,1)
> ylabel(0,0.25,0.50,0.75,1) yline(0,1) xline(0, 1) noaxis
```

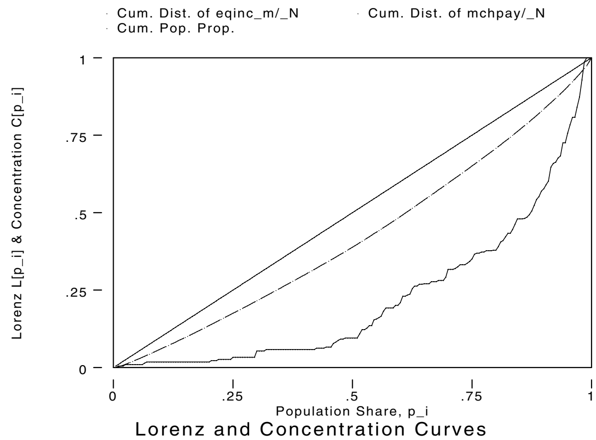

Figure 3. Lorenz and concentration curves for child benefits.

Let us finally show how TIP curves can be constructed. Suppose we wish to make poverty comparisons among two population subgroups, households who own their house (solid lines below) and households who rent their house (dashed lines below). We set the poverty line at 200 monetary units. To draw the TIP curves of absolute poverty gaps, simply type

```
. generate tip = (200 - eqinc)*(eqinc<=200)
. glcurve tip , gl(tip) p(tipp) sort(eqinc) by(owner) split
> xlabel(0,0.25,0.50,0.75,1) ylabel
```

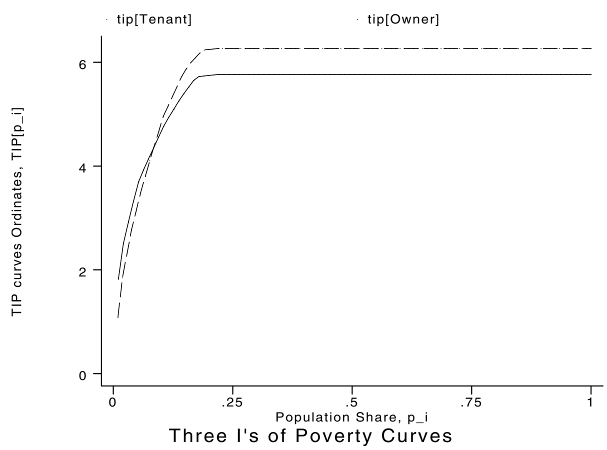

Figure 4. TIP curves of absolute poverty gaps for home owners and renters.

Imagine now that we consider setting a lower poverty line for households that own their houses, e.g., 170 monetary units. We want to construct TIP curves of relative poverty gaps:

```
. generate tiprel = (1 - (eqinc/200))*(eqinc<=200) if owner==0
. replace tiprel = (1 - (eqinc/170))*(eqinc<=170) if owner==1
. glcurve tiprel , gl(tipr) p(tipp) replace sort(eqinc) by(owner)
> split xlabel(0,0.25,0.50,0.75,1) ylabel
```

(*Graph on next page*)

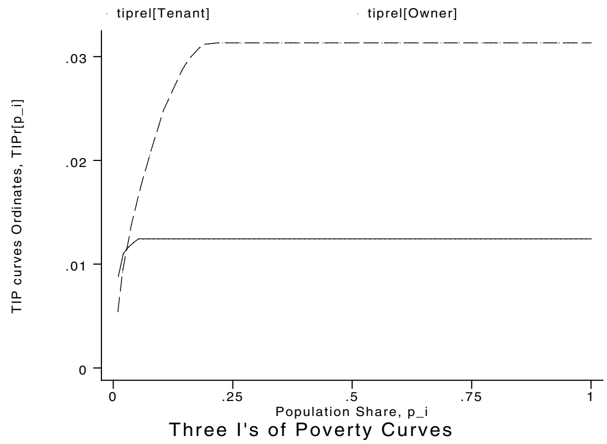

Figure 5. TIP curves of relative poverty gaps.

References

Cowell, F. A. 1995. *Measuring Inequality*. 2d ed. Prentice–Hall/Harvester–Wheatsheaf, Hemel Hempstead.

Jenkins S. P. and P. J. Lambert. 1997. Three 'I's of poverty curves, with an analysis of UK poverty trends. *Oxford Economic Papers* 49: 317–327.

Lambert, P. J. 1993. *The Distribution and Redistribution of Income—A Mathematical Analysis.* 2d ed. Manchester University Press: Manchester and New York.

Shorrocks, A. F. 1983. Ranking income distributions. *Economica* 197: 3–17.

sg108	Computing poverty indices

Philippe Van Kerm, GREBE, University of Namur, Belgium, philippe.vankerm@fundp.ac.be

Description

The objective of this insert is to help automate the estimation of a series of standard poverty measures from unit record income data. The indices computed by `poverty` are classic measures from the Foster–Greer–Thorbecke class (including the headcount ratio and the poverty gap ratio), the income gap ratio and the aggregate poverty gap, the Sen, Takayama, Thon and Watts indices, and measures from the Clark–Hemming–Ulph class. The formulas for all these measures are given below. However, I refer the reader to the literature on poverty measurement or to the original papers for an exposition of the properties of the various indices (see among the references given below).

Consider a dataset of n observations with each entry being one income recipient unit (for example, household, individual, and so on). Let y_i be the income of the ith observation, w_i be the weight of the ith element (e.g., household size) r_i be the rank of the ith element in the whole distribution (taking weights into account), and z be the poverty line.

Define the indicator $I_i = 0$ if $y_i \geq z$, and 1 otherwise, and define $N = \sum_{i=1}^{n} w_i$, $S = \sum_{i=1}^{n} w_i I_i$.

(*Continued on next page*)

The poverty measures estimated by `poverty` are computed as follows:

Foster–Greer–Thorbecke class: $\mathrm{FGT}(\alpha) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{n}\left(\frac{z-y_i}{z}\right)^{\alpha} w_i I_i$

Headcount ratio: $h = \mathrm{FGT}(0)$

Poverty gap ratio: $\mathrm{pgr} = \mathrm{FGT}(1)$

Income gap ratio: $\mathrm{igr} = \dfrac{1}{S}\displaystyle\sum_{i=1}^{n}\left(\frac{z-y_i}{z}\right) w_i I_i$

Aggregate poverty gap: $\mathrm{apg} = \sum_{i=1}^{n}(z-y_i)\, w_i I_i$

Watts index: $\mathrm{watts} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{n}\left(\ln(z)-\ln(y_i)\right) w_i I_i$

Clark–Hemming–Ulph class: $\mathrm{CHU}(\beta) = \dfrac{1}{\beta N}\displaystyle\sum_{i=1}^{n}\left(1-\left(\frac{y_i}{z}\right)^{\beta}\right) w_i I_i$

Thon index: $\mathrm{thon} = \dfrac{2}{z\,(N+1)\,N}\displaystyle\sum_{i=1}^{n}(N+1-r_i)\,(z-y_i)\,w_i I_i$

Takayama index: $\mathrm{tak} = 1 + \dfrac{1}{N} - \left[\dfrac{2\sum_{i=1}^{N}(N+1-r_i)\,w_i\,(y_i I_i + z\,(1-I_i))}{\sum_{i=1}^{N} N w_i\,(y_i I_i + z\,(1-I_i))}\right]$

Sen index: $\mathrm{sen} = \dfrac{2}{z\,(S+1)\,N}\displaystyle\sum_{i=1}^{n}(S+1-r_i)\,(z-y_i)\,w_i I_i$

In the Foster–Greer–Thorbecke class, along with $\mathrm{FGT}(0)$ and $\mathrm{FGT}(1)$, `poverty` computes $\mathrm{FGT}(\alpha)$ with α =0.5, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5. In the Clark–Hemming–Ulph class, `poverty` computes $\mathrm{CHU}(\beta)$ with β =0.1, 0.25, 0.5, 0.75, 0.9.

Syntax

`poverty` *varname* [*weight*] [`if` *exp*] [`in` *range*] [, `line(`*#*`)` `gen(`*newvarname*`)` *select_options*]

`aweight`s and `fweight`s are allowed.

Options

`line(`*#*`)` specifies the value of the poverty line. If # is set to −1, the poverty line is computed as half the median of *varname*. If # is set to −2, it is computed as two-thirds the median of *varname*. Default is −1.

`gen(`*newvarname*`)` creates the new variable *newvarname* and sets it to 1 for all observations identified as poor (i.e., observations for which *varname* is below the specified poverty line) and 0 for observations identified as non-poor. *newvarname* is set to missing for observations with missing *varname* or not included by the `if` `in` statements.

select_options are options used to select the indices to be computed. It can be any of the following (multiple selections are allowed, see examples below):

`h`	: headcount ratio	`apg`	: aggregate poverty gap
`pgr`	: poverty gap ratio	`igr`	: income gap ratio
`s`	: Sen index	`w`	: Watts index
`tak`	: Takayama index	`thon`	: Thon index
`fgt1`	: FGT(0.5)	`fgt2`	: FGT(1.5)
`fgt3`	: FGT(2)	`fgt4`	: FGT(2.5)
`fgt5`	: FGT(3)	`fgt6`	: FGT(3.5)
`fgt7`	: FGT(4)	`fgt8`	: FGT(4.5)
`fgt9`	: FGT(5)	`chu1`	: CHU(0.1)
`chu2`	: CHU(0.25)	`chu3`	: CHU(0.5)
`chu4`	: CHU(0.75)	`chu5`	: CHU(0.9)
`all`	: wrapper to select all the above indices at once.		

Example

The use of `poverty` is extremely simple. Consider the dataset `subcvse.dta` provided with `glcurve` in Jenkins and Van Kerm (1999). We have a (single adult equivalent) household income measure (`eqinc`) and a variable with the household size (`size`). Applying `poverty` to `eqinc` (f)weighted by `size` with the `all` option returns the whole series of measures computed over all observations and taking half the median of `eqinc` as the poverty line.

```
. poverty eqinc [fw=size] , all
-------------------------------------------------------------------------------
Poverty measures of eqinc
-------------------------------------------------------------------------------
 Your selection is made of 200 observations.
 The following poverty analysis has been using the 200 non-missing
 observations for eqinc in your selection.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 The poverty line is set at 134.5 units
                          (1/2 of median value)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Headcount ratio %                          1.020
Aggregate poverty gap                419.5 units
                     (or equivalently      0.48 units per obs.)
Poverty gap ratio %                        0.354
Income gap ratio %                        34.721
Watts index                                0.557
Index FGT(0.5) *100                        0.562
Index FGT(1.5) *100                        0.249
Index FGT(2.0) *100                        0.188
Index FGT(2.5) *100                        0.150
Index FGT(3.0) *100                        0.124
Index FGT(3.5) *100                        0.105
Index FGT(4.0) *100                        0.090
Index FGT(4.5) *100                        0.079
Index FGT(5.0) *100                        0.069
Clark et al. index (0.10) *100             0.528
Clark et al. index (0.25) *100             0.489
Clark et al. index (0.50) *100             0.434
Clark et al. index (0.75) *100             0.390
Clark et al. index (0.90) *100             0.368
Sen index *100                             0.466
Thon index *100                            0.706
Takayama index *100                        0.353
[fweight= size]
-------------------------------------------------------------------------------
```

If we are interested only in the headcount ratio, the poverty gap ratio and the Sen index and want to check the sensitivity of the results against a different poverty line (e.g., two-third of the median), we can type

```
. poverty eqinc [fw=size] , h pgr s line(-2)
-------------------------------------------------------------------------------
Poverty measures of eqinc
-------------------------------------------------------------------------------
 Your selection is made of 200 observations.
 The following poverty analysis has been using the 200 non-missing
 observations for eqinc in your selection.
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 The poverty line is set at 179.3333333333334 units
                            (2/3 of median value)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Headcount ratio %                          5.102
Poverty gap ratio %                        0.887
Sen index *100                             1.326
[fweight= size]
--------------------------------------------------------------------------------
```

In order to study poverty incidence in a particular sub-population, we can save the value of the poverty line computed over the whole population (see Saved Results below) and re-do the analysis by specifying the saved poverty line and selecting the appropriate observations:

```
. loc line2 = $S_4
. poverty eqinc if size<5 [fw=size] , h pgr s line(`line2´)
--------------------------------------------------------------------------------
Poverty measures of eqinc
--------------------------------------------------------------------------------
 Your selection is made of 180 observations.
 The following poverty analysis has been using the 180 non-missing
 observations for eqinc in your selection.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 The poverty line is set at 179.3333 units
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Headcount ratio %                          6.812
Poverty gap ratio %                        1.184
Sen index *100                             1.771
[fweight= size]
--------------------------------------------------------------------------------
```

Saved Results

`poverty` saves a number of results:

S_1	total number of observations in the data
S_2	number of observations used to compute the indices
S_3	weighted number of observations
S_4	value of the poverty line
S_5	weighted number of observations identified as poor

(the following results are only available if the measure has been requested)

S_6	headcount ratio [FGT(0)]	S_17	FGT(4)
S_7	aggregate poverty gap	S_18	FGT(4.5)
S_8	poverty gap ratio [FGT(1)]	S_19	FGT(5)
S_9	income gap ratio	S_20	CHU(0.10)
S_10	Watts index	S_21	CHU(0.25)
S_11	FGT(0.5)	S_22	CHU(0.50)
S_12	FGT(1.5)	S_23	CHU(0.75)
S_13	FGT(2)	S_24	CHU(0.90)
S_14	FGT(2.5)	S_25	Sen index
S_15	FGT(3)	S_26	Thon index
S_16	FGT(3.5)	S_27	Takayama index

References

Atkinson, A. B. 1987. On the measurement of poverty. *Econometrica* 55(4): 749–764.

Clark, S., R. Hemming, and D. Ulph. 1981. On indices for the measurement of poverty. *The Economic Journal* 91: 515–526.

Foster, J., J. Greer, and E. Thorbecke. 1984. A class of decomposable poverty measures. *Econometrica* 52: 761–765.

Hagenaars, A. 1986. *The Perception of Poverty*, Contributions to Economic Analysis series vol. 156. North–Holland.

Jenkins, S. P. and P. Van Kerm. 1999. sg107: Generalized Lorenz curves and related graphs. *Stata Technical Bulletin* 48: 25–29.

Ravallion, M. 1994. *Poverty Comparisons*, Fundamentals in Pure and Applied Economics series vol. 56. Harwood Academic Publishing.

Sen, A. 1976. Poverty: an ordinal approach to measurement. *Econometrica* 44: 219–231.

Takayama, N. 1979. Poverty, income inequality, and their measures: Professor Sen's axiomatic approach reconsidered. *Econometrica* 47.

Thon, D. 1979. On measuring poverty. *Review of Income and Wealth 2* 5: 429–440.

sg109	Utility to convert binomial frequency records to frequency weighted data

Mario Cleves, Stata Corporation, mcleves@stata.com

[*Editor's note: There are no help files or ado-files for this insert as this is an undocumented command in Stata 6.*]

Syntax

`bitowt` *case#_var pop_var* [`if` *exp*] [`in` *range*] [, `case(`*newvarname*`)` `weight(`*newvarname*`)`]

Description

`bitowt` converts binomial frequency records to frequency weighted data. *case#_var* specifies the variable containing the number of cases represented by each observation and `pop_var` specifies the corresponding number of total subjects (cases plus controls). This command will change the data in memory.

Options

`case(`*newvarname*`)` specifies the name of a new binomial case-indicator variable containing 1 for cases and 0 for controls. If `case()` is not specified, `case(_case)` is assumed.

`weight(`*newvarname*`)` specifies the name of a variable that will contain frequency weights. If `weight()` is not specified, `weight(_weight)` is assumed.

Remarks

`bitowt` is a utility that converts binomial frequency data to frequency weighted data. Binomial frequency data can be directly analyzed with `epitab`'s `cc`, `tabodds` and `mhodds` commands, but has to be converted if other commands such as `poisson` or `logistic` are to be used.

In each record of a binomial dataset there is a variable indicating the number of cases, a variable indicating the total number of subjects (cases plus controls), and additional variables. For example, the following is a binomial dataset:

```
.  list in 1/8
        agegrp    tobacco          D         N
  1.     25-34        0-9          0       140
  2.     25-34      10-19          2        38
  3.     25-34      20-29          0        22
  4.     25-34        30+          0        32
  5.     35-44        0-9          4       218
  6.     35-44      10-19          8        92
  7.     35-44      20-29          6        54
  8.     35-44        30+          0        34
```

Each observation has a variable indicating the observed number of cases, `D`, out of `N` subjects in the corresponding age group and tobacco-use stratum. That is, in the first observation, there are no cases out of 140 subjects age 25 to 34 who use 0 to 9 grams of tobacco per day. In the second observation, there are 2 cases out of 38 subjects age 25 to 34 who use 10 to 19 grams of tobacco per day, and so on.

We can use the `cc`, `mhodds` and `tabodds` commands directly on these data by specifying the `binomial()` option. The data, however, needs to be converted to single record or frequency record data in order to use other Stata commands.

The `bitowt` command can convert our binomial data to frequency data.

```
. bitowt D N
. list agegrp tobacco _case _weight
        agegrp    tobacco      _case    _weight
  1.     25-34        0-9          0        140
  2.     25-34        0-9          1          0
  3.     25-34      10-19          0         36
  4.     25-34      10-19          1          2
  5.     25-34      20-29          0         22
  6.     25-34      20-29          1          0
  7.     25-34        30+          0         32
```

```
  8.     25-34        30+          1          0
  9.     35-44        0-9          0        214
 10.     35-44        0-9          1          4
 11.     35-44      10-19          0         84
 12.     35-44      10-19          1          8
 13.     35-44      20-29          0         48
 14.     35-44      20-29          1          6
 15.     35-44        30+          0         34
 16.     35-44        30+          1          0
```

In this new dataset, each of the original observations is split into two observations: one for the cases and one for the controls. Because we did not specify the `case()` or the `weight()` option, the default variable names `_case` and `_weight` were used to name the new variables. The `_case` variable indicates whether the observations are for cases or for controls and the `_weight` variable specifies the corresponding number of cases or controls.

This new dataset can be used with any Stata command that allows frequency weights. For example, we could use `logistic` to further analyze these data remembering to specify the `[fweight=_weight]` option.

sg110	Hardy–Weinberg equilibrium test and allele frequency estimation

Mario Cleves, Stata Corporation, mcleves@stata.com

Syntax

`genhw` *all1 all2* [*weight*] [`if` *exp*] [`in` *range*] [`,` `binvar`]

`genhwi` $\#_{AA}$ $\#_{Aa}$ $\#_{aa}$ [`,` `label(`*genotypes*`)` `binvar`]

`genhw` allows `fweight`s.

Description

`genhw` estimates allele frequencies, genotype frequencies, and disequilibrium coefficients for codominant traits or data of completely known genotypes, and performs asymptotic Hardy–Weinberg (HW) equilibrium tests. In the case of two alleles, it also calculates an exact HW significance probability.

`genhw` expects each observation to contain the values of the two alleles at the locus being examined (*all1* and *all2*). Allele values can be numeric or string.

`genhwi` is the immediate form of `genhw` using the genotypic counts on the command line, where #AA, #Aa and #aa are the counts for the AA, Aa and aa genotypes. Note that this command only works for biallelic loci.

Options

`binvar` specifies that binomial standard errors be reported for each allele. These standard errors are calculated assuming that the population is in Hardy–Weinberg equilibrium. By default, standard errors that do not require this assumption are reported.

`label(`*genotypes*`)` specifies labels to be used in the output of the genotype frequency table. This option is only valid for the immediate form of the command.

Remarks

`genhw` estimates allele and genotype frequencies for codominant traits or data where there is no ambiguity regarding genotypes. It also performs asymptotic tests for Hardy–Weinberg equilibrium and estimates the disequilibrium coefficient (D) for each heterozygotic genotype in the sample. See *Methods and Formulas* for details of these calculations.

Example 1: biallelic locus

Sham (1998) presented MN blood group data from a random sample of 747 individuals. We would like to test whether or not the population is in Hardy–Weinberg equilibrium. We entered these data into a Stata dataset. Here are a few observations:

```
. list in 1/10

            a1         a2
  1.         M          M
  2.         M          N
  3.         N          N
  4.         M          M
```

```
     5.         M          M
     6.         M          M
     7.         M          M
     8.         M          M
     9.         M          M
    10.         M          M
```

Each observation corresponds to one of the 747 individuals and records that individual's genotype; the `a1` variable holds the value of the first allele, and the `a2` variable that of the second allele.

We now perform the test for Hardy–Weinberg equilibrium.

```
. genhw a1 a2
        Genotype |     Observed        Expected
     ------------+-----------------------------
              MM |          233          242.37
              MN |          385          366.26
              NN |          129          138.37
     ------------+-----------------------------
           total |          747          747.00
          Allele | Observed     Frequency      Std. Err.
     ------------+-------------------------------------
               M |      851        0.5696         0.0125
               N |      643        0.4304         0.0125
     ------------+-------------------------------------
           total |     1494        1.0000
     Estimated disequilibrium coefficient (D) =  -0.0125
     Hardy-Weinberg Equilibrium Test:
              Pearson chi2 (1) =    1.956  Pr= 0.1620
     likelihood-ratio chi2 (1) =    1.959  Pr= 0.1616
      Exact significance prob  =              0.1793
```

The command first tabulates the observed and expected (under HW) genotype frequencies, the allele frequencies, and corresponding estimated standard errors. Then it calculates Pearson's and the likelihood-ratio chi-squared statistics, and in the case of a biallelic locus, an exact significance probability is also reported.

For these data all three Hardy–Weinberg tests agree. They are not statistically significant; therefore, we fail to reject the null hypothesis that the population is in Hardy–Weinberg equilibrium.

We also obtained an estimate of the disequilibrium coefficient (D). At Hardy–Weinberg equilibrium, the expected value of the disequilibrium coefficient is zero.

An immediate form of the above command that will yield the same results is constructed using the observed genotype counts:

```
. genhwi 233 385 129, label(MM MN NN)
```

The `label()` option is used to label the tables. The `genhwi` command expects the genotype counts to be ordered as shown in the syntax diagram.

Because there is no statistical evidence that this population is not in Hardy–Weinberg equilibrium, we can rerun the command specifying the `binvar` option producing binomial standard error.

```
.   genhw a1 a2, binvar
        Genotype |     Observed        Expected
     ------------+-----------------------------
              MM |          233          242.37
              MN |          385          366.26
              NN |          129          138.37
     ------------+-----------------------------
           total |          747          747.00
          Allele | Observed     Frequency      Std. Err.
     ------------+-------------------------------------
               M |      851        0.5696         0.0128 (binomial)
               N |      643        0.4304         0.0128 (binomial)
     ------------+-------------------------------------
           total |     1494        1.0000
     Estimated disequilibrium coefficient (D) =  -0.0125
```

```
Hardy-Weinberg Equilibrium Test:
         Pearson chi2 (1) =    1.956  Pr= 0.1620
likelihood-ratio chi2 (1) =    1.959  Pr= 0.1616
 Exact significance prob  =               0.1793
```

Example 2: multiallelic locus

Spencer et al. (1964) examined the distribution of the red cell acid phosphatase polymorphism in 178 randomly selected individuals. They identified 3 alleles at this locus; A, B and C. We would like to test the null hypothesis that these data are consistent with Hardy–Weinberg equilibrium. Their data has been entered into Stata. Here are the first ten observations:

```
.  list in 1/10
         all1      all2
  1.        A         A
  2.        A         B
  3.        A         C
  4.        B         B
  5.        B         C
  6.        A         B
  7.        A         B
  8.        B         B
  9.        A         A
 10.        B         B
```

We now perform the test for Hardy–Weinberg equilibrium:

```
. genhw all1 all2
                                                 Disequilibrium
      Genotype | Observed   Expected   Coefficient (D)
    -----------+-------------------------------------
            AA |       17      21.95
            AB |       86      76.19           -0.0275
            AC |        5       4.92           -0.0002
            BB |       61      66.14
            BC |        9       8.53           -0.0013
            CC |        0       0.28
    -----------+-------------------------------------
         Total |      178     178.00
        Allele | Observed     Frequency      Std. Err.
    -----------+-------------------------------------
             A |      125        0.3511         0.0237
             B |      217        0.6096         0.0242
             C |       14        0.0393         0.0101
    -----------+-------------------------------------
    Hardy-Weinberg Equilibrium Test:
             Pearson chi2 (3) =    3.078  Pr= 0.3798
    likelihood-ratio chi2 (3) =    3.407  Pr= 0.3330
```

Similar to the output in the biallelic case, genotype and allele frequency tables are produced. However, instead of only one disequilibrium coefficient, in the multiallelic case, a disequilibrium coefficient is estimated for each heterozygous genotype.

For these data, we fail to reject the null hypothesis that the population is in Hardy–Weinberg equilibrium with respect to this locus.

Saved results

`genhw` saves in `r()`:

Scalars

`r(chi2)`	Pearson's chi squared
`r(df)`	degrees of freedom
`r(chi2_p)`	significance probability (Pearson)
`r(lr_chi2)`	likelihood-ratio chi squared
`r(lr_p)`	significance probability (LR)
`r(p_exact)`	exact significance probability (biallelic only)
`r(D)`	disequilibrium coefficient (biallelic only)

Methods and formulas

Borrowing the notation from Weir (1996), let A_u, $u = \{1, ..., k\}$ represent k alleles at a locus and $A_u A_v$ represent each of the possible $k(k+1)/2$ distinct genotypes.

Consider a random sample of n individuals. Then the observed alleles counts, n_u, are

$$n_u = 2n_{uu} + \sum_{u \neq v} n_{uv}$$

where n_{uv} and n_{uu} are respectively, the observed number of heterozygotes $A_u A_v$ and homozygotes $A_u A_u$ in the sample.

The population allele frequencies are therefore estimated as

$$\widehat{p}_u = \frac{n_u}{2n}$$

and their variances as

$$\text{var}(\widehat{p}_u) = \frac{1}{2n}(\widehat{p}_u + P_{uu} - 2\widehat{p}_u^2)$$

where P_{uu} is the observed frequency of the $A_u A_u$ genotype.

Each allele variance under Hardy–Weinberg equilibrium simplifies to the variance of a binomial distribution with parameters p_u and $2n$:

$$\text{var}(\widehat{p}_u) = \frac{1}{2n}\widehat{p}_u(1 - \widehat{p}_u)$$

The expected genotype frequencies under the assumption of Hardy–Weinberg equilibrium are estimated as

$$E(P_{uu}) = \widehat{p}_u^2$$

for homozygotes, and

$$E(P_{uv}) = 2\widehat{p}_u\widehat{p}_v \qquad (u \neq v)$$

for heterozygotes.

The disequilibrium coefficients for heterozygous genotypes are estimated as

$$\widehat{D}_{uv} = \widehat{p}_u\widehat{p}_v - \frac{1}{2}P_{uv}$$

The Pearson's chi-squared test statistic is computed using the observed and expected genotype counts as

$$\sum_u \frac{(n_{uu} - n\widehat{p}_u^2)^2}{n\widehat{p}_u^2} - \sum_{u \neq v} \frac{(n_{uv} - 2n\widehat{p}_u\widehat{p}_v)^2}{2n\widehat{p}_u\widehat{p}_v}$$

and the likelihood-ratio chi squared test statistic as

$$-2\ln\left(\frac{L_0}{L_1}\right)$$

where

$$L_0 = \sum_u n_{uu}\ln\left(\frac{n_u}{2n}\right)^2 + \sum_u \sum_{u \neq v} n_{uv}\ln\left(\frac{n_u n_v}{2n^2}\right)$$

and

$$L_1 = \sum_u n_{uu}\ln\left(\frac{n_{uu}}{n}\right) + \sum_u \sum_{u \neq v} n_{uv}\ln\left(\frac{n_{uv}}{n}\right)$$

Both Pearson's and the likelihood-ratio chi-squared test statistics are distributed with $k(k-1)/2$ degrees of freedom.

References

Sham, P. 1998. *Statistics in Human Genetics.* New York: John Wiley & Sons.

Spencer, N., D. A. Hopkinson, and H. Harris. 1964. Quantitative differences and gene dosage in the human red cell acid phosphatase polymorphism. *Nature* 201: 299–300.

Weir, B. S. 1996. *Genetic Data Analysis II.* Sunderland, MA: Sinauer Associates.

STB categories and insert codes

Inserts in the STB are presently categorized as follows:

General Categories:

an	announcements	*ip*	instruction on programming
cc	communications & letters	*os*	operating system, hardware, & interprogram communication
dm	data management	*qs*	questions and suggestions
dt	datasets	*tt*	teaching
gr	graphics	*zz*	not elsewhere classified
in	instruction		

Statistical Categories:

sbe	biostatistics & epidemiology	*ssa*	survival analysis
sed	exploratory data analysis	*ssi*	simulation & random numbers
sg	general statistics	*sss*	social science & psychometrics
smv	multivariate analysis	*sts*	time-series, econometrics
snp	nonparametric methods	*svy*	survey sampling
sqc	quality control	*sxd*	experimental design
sqv	analysis of qualitative variables	*szz*	not elsewhere classified
srd	robust methods & statistical diagnostics		

In addition, we have granted one other prefix, *stata*, to the manufacturers of Stata for their exclusive use.

Guidelines for authors

The Stata Technical Bulletin (STB) is a journal that is intended to provide a forum for Stata users of all disciplines and levels of sophistication. The STB contains articles written by StataCorp, Stata users, and others.

Articles include new Stata commands (ado-files), programming tutorials, illustrations of data analysis techniques, discussions on teaching statistics, debates on appropriate statistical techniques, reports on other programs, and interesting datasets, announcements, questions, and suggestions.

A submission to the STB consists of

1. An insert (article) describing the purpose of the submission. The STB is produced using plain TeX so submissions using TeX (or LaTeX) are the easiest for the editor to handle, but any word processor is appropriate. If you are not using TeX and your insert contains a significant amount of mathematics, please FAX (409–845–3144) a copy of the insert so we can see the intended appearance of the text.

2. Any ado-files, `.exe` files, or other software that accompanies the submission.

3. A help file for each ado-file included in the submission. See any recent STB diskette for the structure a help file. If you have questions, fill in as much of the information as possible and we will take care of the details.

4. A do-file that replicates the examples in your text. Also include the datasets used in the example. This allows us to verify that the software works as described and allows users to replicate the examples as a way of learning how to use the software.

5. Files containing the graphs to be included in the insert. If you have used STAGE to edit the graphs in your submission, be sure to include the `.gph` files. Do not add titles (e.g., "Figure 1: ...") to your graphs as we will have to strip them off.

The easiest way to submit an insert to the STB is to first create a single "archive file" (either a `.zip` file or a compressed `.tar` file) containing all of the files associated with the submission, and then email it to the editor at `stb@stata.com` either by first using `uuencode` if you are working on a Unix platform or by attaching it to an email message if your mailer allows the sending of attachments. In Unix, for example, to email the current directory and all of its subdirectories:

```
tar -cf - . | compress | uuencode xyzz.tar.Z > whatever
mail stb@stata.com < whatever
```

International Stata Distributors

International Stata users may also order subscriptions to the *Stata Technical Bulletin* from our International Stata Distributors.

Company:	Applied Statistics & Systems Consultants
Address:	P.O. Box 1169 17100 NAZERATH-ELLIT Israel
Phone:	+972 (0)6 6100101
Fax:	+972 (0)6 6554254
Email:	assc@netvision.net.il
Countries served:	Israel

Company:	Axon Technology Company Ltd
Address:	9F, No. 259, Sec. 2 Ho-Ping East Road TAIPEI 106 Taiwan
Phone:	+886-(0)2-27045535
Fax:	+886-(0)2-27541785
Email:	hank@axon.axon.com.tw
Countries served:	Taiwan

Company:	Chips Electronics
Address:	Lokasari Plaza 1st Floor Room 82 Jalan Mangga Besar Raya No. 82 JAKARTA Indonesia
Phone:	62 - 21 - 600 66 47
Fax:	62 - 21 - 600 66 47
Email:	puyuh23@indo.net.id
Countries served:	Indonesia

Company:	Dittrich & Partner Consulting
Address:	Kieler Strasse 17 5. floor D-42697 Solingen Germany
Phone:	+49 2 12 / 26 066 - 0
Fax:	+49 2 12 / 26 066 - 66
Email:	sales@dpc.de
URL:	http://www.dpc.de
Countries served:	Germany, Austria, Italy

Company:	IEM
Address:	P.O. Box 2222 PRIMROSE 1416 South Africa
Phone:	+27-11-8286169
Fax:	+27-11-8221377
Email:	iem@hot.co.za
Countries served:	South Africa, Botswana, Lesotho, Namibia, Mozambique, Swaziland, Zimbabwe

Company:	MercoStat Consultores
Address:	9 de junio 1389 CP 11400 MONTEVIDEO Uruguay
Phone:	598-2-613-7905
Fax:	Same
Email:	mercost@adinet.com.uy
Countries served:	Uruguay, Argentina, Brazil, Paraguay

Company:	Metrika Consulting
Address:	Mosstorpsvagen 48 183 30 Taby STOCKHOLM Sweden
Phone:	+46-708-163128
Fax:	+46-8-7924747
Email:	sales@metrika.se
Countries served:	Sweden, Baltic States, Denmark, Finland, Iceland, Norway

Company:	Ritme Informatique
Address:	34, boulevard Haussmann 75009 Paris France
Phone:	+33 (0)1 42 46 00 42 +33 (0)1 42 46 00 33
Email:	info@ritme.com
URL:	http://www.ritme.com
Countries served:	France, Belgium, Luxembourg

(*List continued on next page*)

International Stata Distributors

(*Continued from previous page*)

Company:	Scientific Solutions S.A.
Address:	Avenue du Général Guisan, 5 CH-1009 Pully/Lausanne Switzerland
Phone:	41 (0)21 711 15 20
Fax:	41 (0)21 711 15 21
Email:	info@scientific-solutions.ch
Countries served:	Switzerland

Company:	Smit Consult
Address:	Doormanstraat 19 5151 GM Drunen Netherlands
Phone:	+31 416-378 125
Fax:	+31 416-378 385
Email:	J.A.C.M.Smit@smitcon.nl
URL:	http://www.smitconsult.nl
Countries served:	Netherlands

Company:	Survey Design & Analysis Services P/L
Address:	249 Eramosa Road West Moorooduc VIC 3933 Australia
Phone:	+61 (0)3 5978 8329
Fax:	+61 (0)3 5978 8623
Email:	sales@survey-design.com.au
URL:	http://survey-design.com.au
Countries served:	Australia, New Zealand

Company:	Timberlake Consultants
Address:	47 Hartfield Crescent WEST WICKHAM Kent BR4 9DW United Kingdom
Phone:	+44 (0)181 462 0495
Fax:	+44 (0)181 462 0493
Email:	info@timberlake.co.uk
URL:	http://www.timberlake.co.uk
Countries served:	United Kingdom, Eire

Company:	Timberlake Consulting S.L.
Address:	Calle Mendez Nunez, 1, 3 41011 Sevilla Spain
Phone:	+34 (9) 5 422 0648
Fax:	+34 (9) 5 422 0648
Email:	timberlake@zoom.es
Countries served:	Spain

Company:	Timberlake Consultores, Lda.
Address:	Praceta Raúl Brandao, n°1, 1°E 2720 ALFRAGIDE Portugal
Phone:	+351 (0)1 471 73 47
Fax:	+351 (0)1 471 73 47
Email:	timberlake.co@mail.telepac.pt
Countries served:	Portugal

Company:	Unidost A.S. Rihtim Cad. Polat Han D:38 Kadikoy 81320 ISTANBUL Turkey
Phone:	+90 (216) 414 19 58
Fax:	+30 (216) 336 89 23
Email:	info@unidost.com
URL:	http://abone.turk.net/unidost
Countries served:	Turkey

Company:	Vishvas Marketing-Mix Services
Address:	“Prashant” Vishnu Nagar Baji Prabhu Deshpande Path, Naupada THANE - 400602 India
Phone:	+91-251-440087
Fax:	+91-22-5378552
Email:	vishvas@vsnl.com
Countries served:	India