Lab 3: Measuring Poverty from Household Surveys
Short Course on Poverty & Development for Nordic Ph.D. Students
University of Copenhagen
June 13-23, 2000

In this lab we'll use the same data that we used in the previous lab (e.g. `c:\copen\data\expend.dta`) and our estimated poverty line to construct poverty indices. In particular, we'll estimate FGT measures of poverty. We'll make comparisons by measuring poverty for each of the seven regions and in urban and rural areas in our dataset. Realizing that we have only a sample of households and not the entire population, we will estimate standard errors for the various poverty measures so that we can test differences between poverty in the various regions. Note that an implicit assumption behind this procedure (and the theory) is that our welfare indicator is the true measure, and that the standard errors around the poverty measures derive solely from the fact that we have a sample of households, not from measurement error.

Stata notes: In this lab we take programming one step further to create '.ado' files, which are simply programs that we write and can save in the directory `c:\ado`. Once written and save, these progams can be invoked just as you would use a built in command in Stata. For example, we will write a program called `fgt.ado`, which we will save as `c:\ado\fgt.ado`. Then with a dataset loaded in memory, in the Stata command line (in any directory), we can type `fgt var wgt z alpha`, where `var` is the variable of interest (e.g. per capita household consumption expenditures), `wgt` is the sampling weight variable, `z` is the poverty line (which can be a number or a scalar), and `alpha` is the poverty sensitivity parameter (which can be a number or a scalar), we will receive as output the P-alpha poverty measure.

Approach:

1. *Calculating FGT measures:* First note that the FGT poverty measure for a given a is defined over a continuous variable y which has support in the non-negative real numbers, as

$$P_a = \int_0^q \left(\frac{z-y}{z} \right)^a dy$$

An estimate of this in discrete terms is

$$P_a = \frac{1}{N} \sum_{i=1}^{q} \left(\frac{z-y_i}{z} \right)^a$$

To make it easier to program, note that this is equivalent to

$$P_a = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{z-y_i}{z} \right)^a 1(y_i \le z)$$

where 1(•) is an indicator variable which takes on a value of one if it's argument is true (i.e. the household is poor) and zero otherwise (i.e. the poverty gap of the nonpoor is set to zero). Thus the estimate of P_a is simply the average of the poverty gaps raised to the power a, where by definition the poverty gaps of the non-poor are zero.

Now using our dataset `c:\copen\data\expend.dta`, we can estimate FGT measures for a = 0, 1, 2 as follows using our poverty line of 736 calculated in lab 2,

```
use c:\copen\data\expend
scalar z = 736
gen pcexp = hhexpend / hhsize
gen ind = 0
replace ind = 1 if pcexp <= z
gen gap0 = ind * ((z-pcexp)/z)^0
gen gap1 = ind * ((z-pcexp)/z)^1
gen gap2 = ind * ((z-pcexp)/z)^2
sum gap0 gap1 gap2
```

Note that this standard definition assumes that each individual is weighted equally (i.e. the sampling weights for each individual are $\frac{1}{N}$). If the weights differ, we only have to change one simple line to get the weighted average of the poverty gaps raised to the power a,

```
sum gap0 gap1 gap2 [aw=hhsize]
```

where our weight (or expansion factor) is the house size.

Clearly a nice feature of the FGT-class of poverty measures is it's simplicity and the ease with which we can calculate it. Now let's make it even easier, by making it a command recognized by Stata. To do this we write the following program,

```
#delimit ;
program define palpha;
        version 6.0;
        local var `1';
        local wgt `2';
        local z   `3';
        local alpha = 0;
        tempvar ind gap0 gap;
         quietly gen `gap0' = (`z' - `var')/abs(`z');
         quietly gen `ind'  = 0;
         qui replace `ind'  = 1 if `gap0' >= 0;

        while `alpha'<=2 {;
               quietly gen `gap'  = (`gap0'^`alpha')*`ind';
               summ `gap' [aw=`wgt'], meanonly;
                local phat = r(mean);
                global P`alpha' = `phat';
                drop `gap';
                local alpha = `alpha' + 1;
                     };
        display  "P0 = "   %5.4f $P0;
        display  "P1 = "   %5.4f $P1;
        display  "P2 = "   %5.4f $P2;
end;
```

and save it as `c:\ado\palpha.ado`

Then from the Stata command prompt, we can type

```
palpha pcexp hhsize z
```

and we'll should get the same results as we got previously.

2. *Testing the difference between FGT measures:* We follow Davidson & Duclos (1998) to show how to test the difference between FGT measures for independent distributions and the same poverty line (we'll use this in the next lab for stochastic dominance testing). Davidson & Duclos show that if we define

$$D^s(z) = \tfrac{1}{N(s-1)!}\sum_{i=1}^{N}(z-y_i)^{s-1}I(y_i \le z)$$

which is equivalent to $\frac{z}{a!}P_a$ where $s = a+1$, then the variance of $D^s(z)$ is simply

$$Var = \tfrac{1}{N}\sum_{i=1}^{N}\left[\tfrac{1}{a!}(z-y_i)^a\right]^2 - \left(D^s(z)\right)^2$$

since $D^s(z)$ is the sum of *iid* random variables. It is easy to show that

$$Var = \left(\frac{z}{a!}\right)^2\left[\frac{1}{N}\sum_{i=1}^{N}\left[\left(\frac{z-y_i}{z}\right)^a\right]^2 - (P_a)^2\right] = \left(\frac{z}{a!}\right)^2 M$$

Simple t-statistics can now be used to test the difference between FGT measures for independent distributions *A* and *B* (see Lab 4 discussion for a discussion of the denominator),

$$t = \frac{D_A^s - D_B^s}{\sqrt{\text{var}(D_A^s) + \text{var}(D_B^s)}}$$

$$= \frac{\frac{z}{a!}\left(P_a^A - P_a^B\right)}{\frac{z}{a!}\sqrt{M_A + M_B}} = \frac{P_a^A - P_a^B}{\sqrt{M_A + M_B}}$$

Having set the stage, let's now write a program which we can save as `c:\ado\fgt.ado`, that calculate an FGT measure, the associated $D^s(z)$ and var($D^s(z)$).

```
* syntax is "fgt var wgt z alpha"

#delimit ;
program define fgt;
  version 6.0
  local var      `1';
  local wgt      `2';
  local z      = `3';
  local alpha  = `4';

  local s    = `alpha' + 1;
  local afac = round(exp(lnfact(`alpha')),1);
  tempvar N ind gap ngap;
    quietly gen      `gap'  = `z' - `var';
    quietly gen      `ind'  = 0;
     quietly replace `ind'  = 1 if `gap' >= 0;
    quietly gen      `ngap' = `gap' / `z';
     quietly replace `gap'  = (`gap'^`alpha')  * `ind';
     quietly replace `ngap' = (`ngap'^`alpha') * `ind';
  summarize  `gap' [aw=`wgt'], meanonly;
  local dhat  = r(mean) / `afac';
  summarize `ngap' [aw=`wgt'], meanonly;
  local phat  = r(mean);
     quietly replace `gap'  = `gap' ^ 2;
  summarize `gap' [aw=`wgt'], meanonly;
  local vdhat= (1/r(N))*((r(mean)/(`afac'^2)) - (`dhat'^2));

  global P`alpha' = `phat';
  global D`s'     = `dhat';
  global VD`s'    = `vdhat';

  display "P"`alpha' "  = " `phat';
  display "D"`s'     "  = " `dhat';
  display "VD"`s'    " = "  `vdhat';
end;
```

With this in hand, we can calculate regional FGT measures and test the differences between them. For example, for the headcount ratio for regions 1 and 2 in our dataset,

```
preserve
keep if region==1
fgt pcexp hhsize z 0
scalar D1  = $D1
scalar VD1 = $VD1
restore
preserve
keep if region==2
fgt pcexp hhsize z 0
scalar D2  = $D1
scalar VD2 = $VD1
restore
scalar tstat12 = (D1-D2)/sqrt(VD1+VD2)
display tstat12
```

Now you can experiment with the remaining regions and FGT measures.