Dealing with Extreme Values: Trimming and Bottom- / Top- coding

Goal

Many inequality measures are sensitive to the values at the bottom and/or top of the income distribution, and some are not defined for non-positive values of income (e.g., any measure that calculates a logarithm). Therefore, comparative researchers sometimes ‘trim’ the distribution (by deleting the top and bottom 1% for example) or impose ‘bottom codes’ and ‘top codes’ to provide a common calculation of lower and upper limits, method often referred to as ‘winsorising’.

Activity

Use the data for Sweden 2005. Remove all missing and zero values of household disposable income. Using both the trimming and winsorising methods, create the following two new variables:

- variable ***trim***, where the top 1% and bottom 1% of weighted household disposable income (***dpi***) is set to missing (trimming);
- variable ***wins*** where the top 1% and bottom 1% of weighted household disposable income (***dpi***) are set respectively to the value of the 1st and 99th percentile (winsorising).

Compare the mean, median, and the first four and last four observations of the household income before the changes, after trimming, and after winsorising.

Guidelines

- You can easily find the values of the 1st and 99th percentiles of disposable income by using the **`frequencies`** command with the option **`percentiles`**. However, these values are only displayed, and cannot be used for further calculations. The first way of solving this problem is to run the program in two sessions; the first to display the value of the percentile, and a second session with the manually typed values in the program. This way of working is both cumbersome, and error-prone!
- In order to facilitate things, LIS has prepared a routine that matches any percentile (this routine is therefore also valid for the median which is nothing else than the 50th percentile) to your existing data. The routine can be called in the following way :

```
include file = 'i:\match-pctl-incvar.sps'.
```

- This routine requires two parameters to be assigned before calling it : which percentile, and for which income variable you want your calculations. For instance , for the 25th percentile of DPI, use the following two lines :

```
compute inc_var = dpi .
compute pctl = 25 .
```

- The routine will create a new variables for the chosen percentile; its name being ***pctli*** .Since in this exercise we need to trim both ends of the distribution, we will need to run the routine twice, once for the first percentile, and a second time for the 99th percentile. As the name of the new variable as produced within the routine remains constant, be aware to copy the contents of pctli into a new variable, otherwise the contents gets lost with the second run, for instance like :

```
compute pctl99 = pctli .
```

- To see the smallest and largest observation, you can use the minimum and maximum from:

  ```
  descriptives variables = dpi trim wins.
  ```

- The median will not be produced by the `descriptives` command. Therefore one needs to use the `frequencies` command, while specifying the option `statistics`. Remind that running frequencies on continuous variables (like wage) will produce a huge listing!! This must be avoided, and can be done by adding the `format` option, like this:

  ```
  frequencies variables = dpi
      /  statistics = median
      /  format = notable .
  ```

Program

```
title "** INCOME DISTRIBUTION II - Exercise 12 **" .

get file = se05h /keep = hweight dpi .

select if dpi ne 0 .
select if not missing(dpi) .
weight by hweight .
compute wins = dpi.
compute trim = dpi.
compute inc_var = dpi.
compute pctl = 99.
include file = 'i:\match-pctl-incvar.sps' .

***  topcoding, winsorizing.
compute pctl99 = pctli.
if dpi gt pctl99 wins = pctl99 .
frequencies variables = dpi wins trim
   / statistics = median default
   / format = notable .
save outfile = "u:\pa_ex19" .
get file = "u:\pa_ex19" .
compute pctl = 1.
include file = 'i:\match-pctl-incvar.sps' .

***  bottomcoding, winsorizing.
compute pctl1 = pctli.
if dpi lt pctl1 wins = pctl1 .
frequencies variables = dpi wins trim
   / statistics = median default
   / format = notable .

***  topcoding, trimming.
if dpi gt pctl99 trim = $sysmis .
frequencies variables = dpi wins trim
   / statistics = median default
   / format = notable .

***  bottomcoding, trimming.
if dpi lt pctl1 trim = $sysmis .
frequencies variables = dpi wins trim
   / statistics = median default
   / format = notable .

weight off .
frequencies variables = dpi wins trim
   / statistics = min max
/ format = notable .
```

Results

		Original values	**After trimming**	**After winsorising**
Number of valid observations		*16,268*	*15,918*	*16,268*
Average income		*269,551*	*262,484*	*265,713*
Median income		*223,861*	*223,861*	*223,861*
Income level of the first four observations **(smallest incomes)**	Smallest:	*-1,053,732*	*43,066*	*43,066*
	2nd smallest:	*-813,940*	*43,487*	*43,066*
	3rd smallest:	*-270,365*	*43,644*	*43,066*
	4th smallest:	*-239,543*	*43,671*	*43,066*
Income level of the last four observations **(highest incomes)**	4th largest:	*6,542,836*	*803,085*	*806,076*
	3rd largest:	*6,746,146*	*803,952*	*806,076*
	2nd largest:	*7,609,412*	*804,307*	*806,076*
	Largest:	*1,072,029,135*	*806,076*	*806,076*