Using weights in Stata

Yannick Dupraz

September 18, 2013

Stata offers 4 weighting options: frequency weights (`fweight`), analytic weights (`aweight`), probability weights (`pweight`) and importance weights (`iweight`). This document aims at laying out precisely how Stata obtains coefficients and standard errors when you use one of these options, and what kind of weighting to use, depending on the problem [1].

1 Frequency weights: sample with many duplicate observations

Consider the following linear regression model (in matrix form):

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{u} \tag{1}$$

There are n observations and k variables, so that $\boldsymbol{y}$ and $\boldsymbol{u}$ are $[n \times 1]$, $\boldsymbol{\beta}$ is $[(k+1) \times 1]$ and $\boldsymbol{X}$ is $[n \times (k+1)]$. Now let's consider a case where several observations are identical. Let's imagine for example that observations i and $i+1$ are identical, so that:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1k} \\ \vdots & & & \vdots \\ 1 & x_{i1} & \dots & x_{ik} \\ 1 & x_{i1} & \dots & x_{ik} \\ \vdots & & & \vdots \\ 1 & x_{n1} & \dots & x_{nk} \end{bmatrix} \quad \text{and} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ y_i \\ \vdots \\ y_n \end{bmatrix}$$

[1] I do not consider importance weights, which are mainly a tool for the Stata programmer.

1.1 Point estimates

The OLS estimator for $\boldsymbol{\beta}$ is:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$

Applying matrix mutliplication, we find that $\boldsymbol{X}'\boldsymbol{X}$ is equal to:

$$\begin{bmatrix} (1+...+2+...+1) & (x_{11}+...+2x_{i1}+...+x_{n1}) & \dots & (x_{1k}+...+2x_{ik}+...+x_{nk}) \\ \vdots & \vdots & & \vdots \\ (x_{11}+...+2x_{i1}+...+x_{n1}) & (x_{11}^2+...+2x_{i1}^2+...+x_{n1}^2) & \dots & (x_{11}x_{1k}+...+2x_{i1}x_{ik}+...+x_{n1}x_{nk}) \\ \vdots & \vdots & & \vdots \\ (x_{1k}+...+2x_{ik}+...+x_{nk}) & (x_{1k}x_{11}+...+2x_{ik}x_{i1}+...+x_{nk}x_{n1}) & \dots & (x_{1k}^2+...+2x_{ik}^2+...+x_{nk}^2) \end{bmatrix}$$

and

$$\boldsymbol{X}'\boldsymbol{y} = \begin{bmatrix} (y_1+...+2y_i+...+y_n) \\ (x_{11}y_1+...+2x_{i1}y_i+...+y_n) \\ \vdots \\ (x_{1k}y_1+...+2x_{ik}y_i+...+x_{nk}y_n) \end{bmatrix}$$

And you can check that $\boldsymbol{X}'\boldsymbol{X}$=$\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{X}}$ and $\boldsymbol{X}'\boldsymbol{y} = \tilde{\boldsymbol{X}}'\tilde{\boldsymbol{y}}$, where $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{y}}$ were obtained by removing the duplicate row $i+1$ and multiplying each element of row i by $\sqrt{2}$:

$$\tilde{\boldsymbol{X}} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1k} \\ \vdots & & & \vdots \\ \sqrt{2} & \sqrt{2}x_{i1} & \dots & \sqrt{2}x_{ik} \\ \vdots & & & \vdots \\ 1 & x_{n1} & \dots & x_{nk} \end{bmatrix} \quad \text{and} \quad \tilde{\boldsymbol{y}} = \begin{bmatrix} y_1 \\ \vdots \\ \sqrt{2}y_i \\ \vdots \\ y_n \end{bmatrix}$$

More generally, if you have a sample with many identical observations, you might want to remove duplicates, as they do not bring any additional information. After removing all duplicates, you find yourself with a new dataset of m observations, that we will index by j. In order to get the point estimates you would have obtained using all the observations, you need to weight each (compressed) observation j by w_j, the number of times it appears in the non-compressed dataset. Weighting means that the estimator for $\boldsymbol{\beta}$ is:

$$\hat{\boldsymbol{\beta}} = (\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{y}}$$

where

$$\tilde{\boldsymbol{X}} = \begin{bmatrix} \sqrt{w_1} & \sqrt{w_1}x_{11} & \dots & \sqrt{w_1}x_{1k} \\ \vdots & & & \vdots \\ \sqrt{w_m} & \sqrt{w_m}x_{m1} & \dots & \sqrt{w_m}x_{mk} \end{bmatrix} \quad \text{and} \quad \tilde{\boldsymbol{y}} = \begin{bmatrix} \sqrt{w_1}y_1 \\ \vdots \\ \sqrt{w_m}y_m \end{bmatrix}$$

Hence you see that, as far as point estimates are concerned, weighting each observation by w_j is equivalent to estimating the following model:

$$\sqrt{w_j}y_j = \sqrt{w_j}\boldsymbol{x_j}\boldsymbol{\beta} + \sqrt{w_j}u_j \tag{2}$$

1.2 Variance-covariance matrix

Consider model (1), with two identical observations. The OLS variance-covariance matrix of $\hat{\boldsymbol{\beta}}$ is:

$$\Sigma = \hat{\sigma}^2(\boldsymbol{X}'\boldsymbol{X})^{-1}$$

with

$$\hat{\sigma}^2 = \hat{\boldsymbol{u}}'\hat{\boldsymbol{u}}/(n-(k+1))$$

Where $\hat{\boldsymbol{u}}$ is the (column) vector of residuals. Now since y_i and $\boldsymbol{x_i}$ are identical for observations i and $i+1$, residuals are also identical, so that:

$$\hat{\boldsymbol{u}} = \begin{bmatrix} \hat{u}_1 \\ \vdots \\ \hat{u}_i \\ \hat{u}_i \\ \vdots \\ \hat{u}_n \end{bmatrix}$$

We already showed that $\boldsymbol{X}'\boldsymbol{X}$=$\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{X}}$, where $\tilde{\boldsymbol{X}}$ was obtained by removing the duplicate row $i+1$ of $\boldsymbol{X}$ and multiplying each element of row i by $\sqrt{2}$. We can also show that $\hat{\boldsymbol{u}}'\hat{\boldsymbol{u}} = \tilde{\hat{\boldsymbol{u}}}'\tilde{\hat{\boldsymbol{u}}}$ where $\tilde{\hat{\boldsymbol{u}}}$ was obtained by removing the duplicate row $i+1$ and multiplying row i by $\sqrt{2}$. Actually, if we estimate the model on $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{y}}$, the matrix of residuals will be $\tilde{\hat{\boldsymbol{u}}}$ since

$$\hat{\boldsymbol{u}} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}$$

Be careful though: if you estimate σ^2 on the compressed dataset, using model (2), you need to divide the sum of squares by the right denominator: even if the number of observations in your compressed dataset is m, the denominator will be $n-(k+1)$, where n is the number of observations in the uncompressed dataset: $n = \sum_{i=1}^{m} w_i$. Actually, using m or n as the number of observations in the estimation of σ^2 is what makes the difference between Stata's frequency weights and Stata's analytic weights.

2 Analytic weights: sample where each observation is a group-mean

Consider now the following model:

$$y_{ij} = T_j\tau + x_j\beta + u_{ij}$$

where i indexes n individuals and j indexes m villages. Although outcome y_{ij} varies at the individual level, treatment T and controls x are at the village level (think, for instance, of a spatial RDD analysis where controls consist in a semi parametric function of geographical location). You might want to estimate the model on village means:

$$\bar{y}_j = T_j\tau + x_j\beta + \bar{u}_j$$

There are two reasons to do that: the first is that it will be computationally more efficient. The second is that, in this situation, the random unit is really the village, not the individual. It is very likely that a good deal of the unexplained variation that is captured by the error term come from village characteristics and not individual characteristics. By estimating the model on individual rather than villages, you are, in a way, increasing the precision of your estimation by multiplying observations that are already in your sample, not adding new observations. Usually, clustering standard errors by village would be the way to go, but here, since all controls are at the village level, you can estimate the model on village means. Actually, if all variables are at the village level, estimating the model on individuals and clustering standard errors by village will yield the same coefficients and standard errors as estimating on village means, using analytic weights and standard errors robust to heteroskedasticity (`vce(robust)`).

Formally, `aweight` wil produce the same point estimates as `fweight`:

$$\hat{\boldsymbol{\beta}} = (\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{y}}$$

where $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{y}}$ where obtained by multiplying each row of $\boldsymbol{X}$ and $\boldsymbol{y}$ by $\sqrt{w_j}$, w_j being the number of individual contributing to the average. The variance-covariance matrix is given by:

$$\Sigma = \frac{1}{(m-(k+1))}\tilde{\hat{\boldsymbol{u}}}'\tilde{\hat{\boldsymbol{u}}}(\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{X}})^{-1}$$

Note that the number of degrees of freedom is $m-(k+1)$ (m is the number of villages) and not $n-(k+1)$ (n is the total number of individuals). Standard errors will therefore always be higher with `aweight` as compared to `fweight` [2].

3 Probability weights: sample where observations had a different probability of being sampled

Now consider the following model:

$$y_j = \boldsymbol{x}_j\beta + u_j$$

to be estimated on a sample of m observations. Let's assume that there was cluster sampling in the following way: the population was divided in two clusters, rural and urban areas. Half of the urban population was interviewed while only one tenth of the rural population was interviewed. We can see it as follows: one individual interviewed in an urban area represents 2 people while an individual interviewed in a rural area represents 10 persons. In order to take this into account, each observation will be weighted by the inverse of its probability of being sampled: each rural area observation will receive weight 10 and each urban area observation will receive weight 2.

Point estimates will be estimated in the exact same way as with `fweight` and `aweight`. The variance-covariance matrix, however, will be different. Using `aweight`, the variance-covariance matrix would be:

$$\Sigma = \hat{\sigma}^2(\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{X}})^{-1}$$

with

$$\begin{aligned}\hat{\sigma}^2 &= \tilde{\hat{\boldsymbol{u}}}'\tilde{\hat{\boldsymbol{u}}}/(m-k) \\ &= \sum_{j=1}^{m} w_j\hat{u}_j^2/(m-k)\end{aligned}$$

[2] It also means that, if you have the `aweight` standard error for a coefficient, you can recover the `fweight` standard error by multiplying by $\sqrt{\left(\frac{m-k}{n-k}\right)}$, and vice versa

The higher the weight, the higher the observation's contribution to the residual sum of squares. It makes sense if observations are means, as each mean does represent several individuals (if the variance of the individual level error term is σ^2, the variance of the average will be σ^2/w_j, where w_j is the number of individual contributing to the average). But if you think of each observation as a random draw from a subsample, then this variance-covariance matrix is not appropriate anymore (the variance of the j^{th} observation is σ^2 and not σ^2/w_j).

When you use `pweight`, Stata uses a Sandwich (White) estimator to compute the variance-covariance matrix. More precisely, if you consider the following model:

$$y_j = \boldsymbol{x}_j\beta + u_j$$

where j indexes m observations and there are k variables, and estimate it using `pweight`, with weights w_j, the estimate for β is given by:

$$\hat{\boldsymbol{\beta}} = (\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{X}})^{-1}\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{y}}$$

where $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{y}}$ are obtained by multiplying each row of $\boldsymbol{X}$ and $\boldsymbol{y}$ by $\sqrt{w_j}$. Note that point estimates are the same than the one obtained using `aweight`. The variance-covariance matrix is given by:

$$\frac{m}{m-(k+1)}(\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{X}})^{-1}(\tilde{\boldsymbol{X}}'\boldsymbol{W}\tilde{\boldsymbol{X}})(\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{X}})^{-1}$$

where $\boldsymbol{W}$ is an $m \times m$ diagonal matrix with squared residuals $\hat{u}_j^2$ on the diagonal ($\hat{\boldsymbol{u}} = \tilde{\boldsymbol{y}} - \tilde{\boldsymbol{X}}\boldsymbol{\beta}$).

Since, when using `pweight`, standard errors are obtained using a robust sandwich estimator, entering

```
reg y var1 var2 [pweight=n]
```

or

```
reg y var1 var2 [pweight=n], vce(robust)
```

will produce identical point estimates and standard errors. Moreover,

```
reg y var1 var2 [pweight=n]
```

will produce the same results (coefficients and standard errors) as

```
reg y var1 var2 [aweight=n], vce(robust)
```

meaning that, even when you are dealing with a situation similar to the one described in part 2, it would be a good idea to use `pweight` (or `aweight` along with `vce(robust)`) in order to get standard errors that are robust to unspecified heteroskedasticity.