Example script `bias.inp`

Allin Cottrell

The usual situation in econometrics is that we don't know the "true" characteristics of the data-generating process (functional form, parameter vector, distribution of the error term). Rather, we're trying to infer these from a data sample by applying some (hopefully suitable) estimator. However, when thinking about the properties of estimators it can be very useful to work the other way round: use a computer to generate data according to a DGP of our choosing, and see how well the estimator performs at retrieving the (known) parameter values. This is known as Monte Carlo analysis.

The results we get from a *single* computer-generated dataset don't tell us much (sampling error could be quite substantial), so we generally run a large number of replications and look at the average behavior of the estimator.

The script `bias.inp` is a simple example of this sort of thing. It is designed to illustrate via simulation the bias that is induced in the OLS $\hat{\beta}$ when there is a correlation between the error term u and a regressor (column of the X matrix).

Below are some remarks on the gretl code.

The `nulldata` command creates an "empty" dataset with a number of observations given by the scalar argument (so here the dataset will have 200 observations).

We then generate an artificial independent variable, `x`, using the function `uniform()`, which creates a pseudo-random series drawn from the uniform distribution on the range 0–1. This series will remain unchanged in the course of the simulation.

Next we fix on a number of replications, `K`: we choose 5000. And we define a matrix (column vector), `slope`, that will hold the regression slope estimate from each of the replications.

For reference, we run a first `loop` with no bias. Note that there's no point in 5000 replications unless we change something on each iteration; so we generate a new error term, `u`, and recompute the dependent variable, `y`, each time. The `normal()` function gives us a series of pseudo-random drawings from the standard normal distribution, and the DGP is set as

$$y = 10 + 1.5x + u$$

We record each slope coefficient in the appropriate slot in the `slope` vector.

When the loop is finished, we use the `meanc()` function ("mean of column") to calculate the mean of the 5000 estimated slopes. Since we have introduced no bias, this should be close to the known parameter value of 1.5.

We then repeat the `loop` exercise, but this time we define the error term via

```
u = 0.2*x + normal()
```

By including a fraction of the `x` value along with the normal drawing we induce a (positive) correlation between u and X, so that $E(u|X) \neq 0$. In this case we expect to see evidence of bias. Run the script a few times (the results will differ slightly due to the randomness). What do you see?

A simple extension of the analysis is to see what happens if we make the error term negatively correlated with the `x` variable. Try substituting

```
u = -0.3*x + normal()
```

in the second loop and rerun the script.