

Problem set 4  
607: Applied Macroeconometrics  
Fall 2016  
Jon Faust

The following is due at the beginning of next class. You can turn in any paper in my mailbox or in class; email me and requested computer work. You may work in groups; hand in a single submission for the group. The submission should list those who contributed.

1. Terms

(a) Size. Given a model  $Y \sim P_\theta, \theta \in \Theta, Y \in \mathcal{Y}$ , a conventional test is completely defined by a hypothesis,  $H_0 : \theta \in \Theta_0$ , where  $\Theta_0$  is a subset of  $\Theta$ , and a rejection region,  $\mathbf{Y}^c$ , which is a subset of  $\mathcal{Y}$ . If we observe  $Y \in \mathbf{Y}^c$ , we consider  $H_0$  to be rejected. In this notation, what is the size of the test? And what is the power of the test against  $\theta = \theta_k$ .

(b) Define *consistent test*

(c) Define *uniformly most powerful test*.

2. Zellner's Seemingly Unrelated Regressors (SUR) estimator. We have an  $N$ -equation system:

$$y_{nt} = X_{nt}\beta_n + \varepsilon_{nt},$$

or in vector notation:

$$\mathbf{Y}_n = \mathbf{X}_n\beta_n + \varepsilon_n$$

Suppose that all of the  $\varepsilon$ s are independent of all of the  $X$ s, mean zero and iid. The  $\varepsilon$ s are, however, contemporaneously correlated across equations. That is, if  $e_t = (\varepsilon_{1t}, \dots, \varepsilon_{nt})'$ ,  $Ee_t e_t' = \Omega$ , where  $\Omega$  need not be diagonal.

(a) Would OLS on each equation separately give rise to CAN  $\hat{\beta}$ s?

(b) We can stack our system into a single equation,

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

where  $\mathbf{Y}$ ,  $\beta$  and  $\varepsilon$  are  $NT \times 1$  vectors formed by stacking the  $N$  underlying column vectors, 1 through  $N$ . For, example, for  $\mathbf{Y}$ :

$$\mathbf{Y}' = (\mathbf{Y}'_1, \mathbf{Y}'_2, \dots, \mathbf{Y}'_N)'$$

The matrix  $\mathbf{X}$  is block diagonal with  $\mathbf{X}_j$  as the  $j^{\text{th}}$  diagonal block. Under the same conditions as discussed in part (a), OLS on this equation will give rise to CAN estimates. Show that OLS on the stacked system gives the same answer for the  $\beta$ s (and hence  $\hat{\varepsilon}$ s) as equation-by-equation OLS.

- (c) This new system satisfies,

$$E\varepsilon\varepsilon' = \Omega \otimes I_T$$

where  $I_T$  is a  $T \times T$  identity matrix. The GLS estimator is then:

$$\hat{\beta}_{GLS} = (\mathbf{X}(\Omega \otimes I_T)^{-1}\mathbf{X})^{-1}\mathbf{X}(\Omega \otimes I_T)^{-1}\mathbf{Y}$$

Describe the natural FGLS estimator.

- (d) Show that SUR is numerically equal to OLS when each equation has the same regressors.
- (e) Our first vector autoregression (VAR). Stay with the SUR system just described, but suppose that the  $X$  variables in each equation are a constant and  $p$  lags of each of the  $y$  variables.

Define  $\tilde{X}$  as the natural RHS for each equation. That is  $\tilde{X}$  is a matrix with  $t^{\text{th}}$  row equal to:

$$(1, y_{1t-1}, \dots, y_{1t-p}, \dots, y_{Nt-1}, \dots, y_{Nt-p})$$

(where that 1 is for the constant).

In this case, we call the system a VAR, vector autoregression.

For the VAR, then,

$$\mathbf{X} = I \otimes \tilde{X}$$

and FGLS and OLS are equivalent.

Question: Why do lazy time series workers like VARs?

3. Class presentation. We are moving toward the realization that all our magical asymptotic results are magical until we remember that these are asymptotic approximations. And whenever you use an approximation you should find some reasonable method to evaluate the adequacy of the approximation.

In statistics, Monte Carlo is one essential tool in this regard. The key idea behind Monte Carlo work is that if you want to know the distribution of some random item AND if you know how to generate independent replicates of that random variable, then the following trivial recipe always works.. Generate a zillion replicates and take the histogram of these to be a pretty good guide as to the true distribution. (Of course, we can summarize the collection with more formal measures than a histogram as well, such as approximating  $F(c)$  by the share of replicates less than  $c$ .)

Here is the simplest Monte Carlo recipe for understanding the distribution of a statistic:

- Pick a DGP (remember that a DGP a recipe for simulating a data sample from the implied process)
- Generate a zillion independent samples of size  $T$  using the DGP.
- Calculate your statistic of interest on each of the zillion samples.
- Then look at the histogram or empirical distribution of the zillion replicates to get a sense of the distribution of the statistic in a sample of size  $T$ .

In this problem, we'll use Monte Carlo to see how well 'robust standard errors' of the variety proposed by White perform.

Maintained model:

$$y_t = \alpha + \beta x_t + \varepsilon_t$$

$\varepsilon_t \sim N(0, \sigma_{\varepsilon,t}^2)$  and  $\varepsilon_t$  uncorrelated with  $x_t$ . Thus, the model allows heteroskedasticity of unknown form in  $\varepsilon$ . Further,

$$\begin{aligned} x_t &= u_t + v_t \\ u_t &\sim N(0, \sigma_u^2) \end{aligned}$$

where  $v_t$  is 1 with prob.  $\pi$  and 0 with prob.  $1 - \pi$ . Also:  $u, v$ , and  $\varepsilon$  are mutually independent.

The observables are  $x$  and  $y$ , the parameters are  $\alpha, \beta, \sigma_{\varepsilon,t}^2, \sigma_u^2, \pi$ .

- (a) Will the OLS estimate of  $\beta$  be consistent? Efficient? Asymptotically normal? (Sketch proofs)
- (b) Supposing that we choose to use the OLS estimator, we need to use an estimator of the variance-covariance matrix of the coefficients that is appropriate under heteroskedasticity. These are called HC for heteroskedasticity consistent. We'll consider several versions of White's variance-covariance matrix. There are several asymptotically equivalent versions of the White HC estimator. STATA, for example, offers 3 versions of robust standard errors (the `robust`, `robust hc2`, and `robust hc3` options of the `regress` command, see `help regress`).

Problem: Give the formula for these three different versions and explain briefly why they are asymptotically equivalent.

- (c) One main use of the standard errors of a coefficient is creating  $t$ -statistics. We'll assess the properties of White's estimators by checking the distribution under the null hypothesis of  $t$  statistics under particular instances of the maintained model.

We will follow the recipe: pick a DGP, generate a zillion replicates of the relevant  $t$  statistic. Then we'll summarize the distribution by asking how often the statistics exceed the, say, asymptotically justified 5 percent critical value. Of course, we'd like this to be 5 percent.

So, we'll try various instances of the DGP. None will have heteroskedasticity, the both OLS and all the versions of White should give valid results asymptotically (that is the empirical rejection frequency will be 5 percent in a sufficiently large sample).

What will differ across the DGPs is the sample size and  $\pi$ : the probability with which 'outliers' happen in the  $x$ s. I won't bother to state the specifics of the DGPs here: you'll see them explicitly in the provided computer code.

Specifically, nearly complete code for doing this Monte Carlo in Matlab is provided with the problem set. There are a number of instances of ??? in the code where you need to do something. Study the code until you understand what's going on, then complete it run it and answer the following questions:

- Overall, which estimator seems to be most robust in the sense of delivering approximately proper size across a wide range of  $T$  and  $\pi$ ?

- Does it matter which version of White's standard errors you use?
- What lessons do you think might generalize from this exercise?

Note 1: I would like you to learn good programming habits (if you don't already possess these). This is why I start you off with partially completed code.

Note 2: The code as written computes the results and puts them in a matrix, but doesn't print them. You need to figure out how to print the tables. I have included a simple table printing routine (`prtTable.m`). It was written for a slightly different purpose, but it will give you an idea about how to simply print rough, but nice looking tables. If you are more into elegance, I have included a more complicated routine that will generate latex code that can be compiled into nicer looking tables. These are just provided to give a sense of where you might go in making it easy to do stuff like this.

Note 3: We will continue this problem in a subsequent problem set, illustrating the glory of writing good, re-usable computer code.