Macroeconometrics: Problem Set 1

Javier García-Cicco Luis Libonatti

Maestría en Economía - Universidad de San Andrés

September 2021

Agenda

Granger-Causality

Diagnostic Tests

Unit Root Tests

Granger-Causality

- The variable y does not "Granger-cause" x if we cannot anticipate x with values of y. Essentially, we contrast the following two models

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \sum_{\ell=1}^{p} \begin{bmatrix} \phi_\ell^{xx} & 0 \\ \phi_\ell^{yx} & \phi_\ell^{yy} \end{bmatrix} \begin{bmatrix} x_{t-\ell} \\ y_{t-\ell} \end{bmatrix} + \begin{bmatrix} \pi^x \\ \pi^y \end{bmatrix} D_t + \begin{bmatrix} e_t^x \\ e_t^y \end{bmatrix} \qquad \text{(C)}$$

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \sum_{\ell=1}^{p} \begin{bmatrix} \phi_\ell^{xx} & \phi_\ell^{xy} \\ \phi_\ell^{yx} & \phi_\ell^{yy} \end{bmatrix} \begin{bmatrix} x_{t-\ell} \\ y_{t-\ell} \end{bmatrix} + \begin{bmatrix} \pi^x \\ \pi^y \end{bmatrix} D_t + \begin{bmatrix} e_t^x \\ e_t^y \end{bmatrix} \qquad \text{(F)}$$

- A test is carried out with these hypothesis

$$\begin{aligned} \mathrm{H}_0 &: \phi_1^{xy} = 0 \cap \ldots \cap \phi_p^{xy} = 0 \text{ (all } \phi_\ell^{xy} \text{ are zero)} \\ \mathrm{H}_A &: \phi_1^{xy} \neq 0 \cup \ldots \cup \phi_p^{xy} \neq 0 \text{ (at least one } \phi_\ell^{xy} \text{ is not zero),} \end{aligned}$$

  where if H_0 is true, we say that y does not "Granger-cause" x.
- We can employ either a Wald test (only F is estimated), a LM test (only C is estimated) or a likelihood ratio test (both are estimated).

Granger-Causality

- In the case of multiple variables, there are two ways on how to test for Granger-causality:
  1. Compare one variable versus the rest as a whole.
  2. Evaluate each variable individually.
- For example, let's say we have three variables x, y and z and we wish to test if x is "Granger-caused" by the other variables. With the first method, we define the vector $w \equiv (y, z)$ and apply the same logic as in the two variable setup. The constrained model is then

$$\begin{bmatrix} x_t \\ w_t \end{bmatrix} = \sum_{\ell=1}^{p} \begin{bmatrix} \phi_\ell^{xx} & 0 \\ \Phi_\ell^{wx} & \Phi_\ell^{ww} \end{bmatrix} \begin{bmatrix} x_{t-\ell} \\ w_{t-\ell} \end{bmatrix} + \begin{bmatrix} \pi^x \\ \Pi^w \end{bmatrix} D_t + \begin{bmatrix} e_t^x \\ e_t^w \end{bmatrix}$$

whereas with the second method we would have

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \sum_{\ell=1}^{p} \begin{bmatrix} \phi_\ell^{xxx} & \phi_\ell^{xyy} & 0 \\ \phi_\ell^{yxx} & \phi_\ell^{yyy} & \phi_\ell^{yzz} \\ \phi_\ell^{zxx} & \phi_\ell^{zyy} & \phi_\ell^{zzz} \end{bmatrix} \begin{bmatrix} x_{t-\ell} \\ y_{t-\ell} \\ z_{t-\ell} \end{bmatrix} + \begin{bmatrix} \pi^x \\ \pi^y \\ \pi^z \end{bmatrix} D_t + \begin{bmatrix} e_t^x \\ e_t^y \\ e_t^z \end{bmatrix},$$

the difference being on how to treat each ϕ_ℓ^{xyy} in the first equation.

Granger-Causality

- The R package `vars` implements Granger-causality as a Wald test that is executed with the command `causality()`.
- Unfortunately, the second method described previously is not contemplated in the **`vars`** library and requires manual computation.
- **`vars`** also allows to use a bootstrap version instead of an asymptotic approximation.
- The function `causality()` also checks for "instantaneous causality". A Wald test is conducted for the following hypothesis

$$\mathrm{H}_0 : \mathrm{vec}(\Sigma_e^{xy}) = 0$$
$$\mathrm{H}_A : \mathrm{vec}(\Sigma_e^{xy}) \neq 0,$$

involving the (block) covariance of the residuals in question.

Serial Correlation: Portmanteau Test

- This is a multivariate generalization of the Ljung-Box test. It is used to test that there is no auto- or cross correlation in the residuals. The hypothesis of the test are

$$\begin{aligned} \mathrm{H}_0 &: R_1 = 0 \cap \ldots \cap R_h = 0 \text{ (all } R_\ell \text{ are zero)} \\ \mathrm{H}_A &: R_1 \neq 0 \cup \ldots \cup R_h \neq 0 \text{ (at least one } R_\ell \text{ is not zero)}, \end{aligned}$$

  where each R_ℓ is a matrix of cross-correlations of order ℓ.

- The test statistic has an approximate asymptotically chi-squared distribution under H_0

$$Q_m(h) = T \sum_{\ell=1}^{h} \operatorname{tr}\left(\hat{\Gamma}_\ell' \hat{\Gamma}_0^{-1} \hat{\Gamma}_\ell \hat{\Gamma}_0^{-1}\right) \approx\xrightarrow{d} \chi^2_{m^2(h-p)}$$

  where each $\hat{\Gamma}_\ell$ are the sample matrices of cross-covariances of order ℓ computed according to the following formula

$$\hat{\Gamma}_\ell = \frac{1}{T} \sum_{t=\ell+1}^{T} (\hat{e}_t - \bar{\hat{e}})(\hat{e}_{t-\ell} - \bar{\hat{e}})'.$$

Serial Correlation: Portmanteau Test

- There is a trade-off between a good approximation to the chi-squared random variable and a loss of power depending on the specified value for h. Large values should be preferred.
- The test is implemented in the package `vars` via the command `serial.test()` when selecting `type = "PT.asymptotic"` in R.
- For smaller sample sizes, a "corrected" version is available (`type = "PT.adjusted"`). The adjusted test statistic is

$$Q_m^C(h) = T^2 \sum_{\ell=1}^{h} \frac{1}{T-\ell} \mathrm{tr}\left(\hat{\Gamma}_\ell' \hat{\Gamma}_0^{-1} \hat{\Gamma}_\ell \hat{\Gamma}_0^{-1}\right) \approx \xrightarrow{d} \chi^2_{m^2(h-p)}$$

- This test requires a slight modification when applied to VARs estimated with constrained coefficients. Unluckily, **`vars`** does not include this modification.

Serial Correlation: Breusch–Godfrey Test

- A multivariate generalization of the standard Breusch–Godfrey test. We estimate the auxiliary regression

$$\hat{e}_t = F_1 y_{t-1} + \ldots + F_p y_{t-p} + \Upsilon D_t + G_1 \hat{e}_{t-1} + \ldots + G_h \hat{e}_{t-h} + \varepsilon_t$$

  and contrast the following hypothesis to test for auto- or cross correlation in the residuals

$$\begin{aligned} \mathrm{H}_0 &: G_1 = 0 \cap \ldots \cap G_h = 0 \text{ (all } G_j \text{ are zero)} \\ \mathrm{H}_A &: G_1 \neq 0 \cup \ldots \cup G_h \neq 0 \text{ (at least one } G_j \text{ is not zero)}, \end{aligned}$$

  the null hypothesis corresponds to no serial correlation in the residuals.

- Under H_0, the test statistic converges asymptotically to a chi-squared random variable

$$BG(h) = T[m - \mathrm{tr}(\hat{\Sigma}_e^{-1}\hat{\Sigma}_\varepsilon)] \xrightarrow{d} \chi^2_{hm^2}.$$

Serial Correlation: Breusch–Godfrey Test

- Suitable for small h should be used. Degrees of freedom are easily exhausted.
- The test is implemented in `vars` with the command `serial.test()` when selecting `type = "BG"`.
- A "corrected" statistic with better properties in small samples is also available with the option `type = "ES"`

$$BG^C(h) = \frac{1-(1-\delta)^{\frac{1}{\zeta}}}{(1-\theta)^{\frac{1}{\zeta}}} \left(\frac{m^2 h}{\kappa\zeta - \eta} \right)^{-1} \xrightarrow{d} F^{m^2 h}_{\text{int}(\kappa\zeta-\eta)}$$

with

$$\theta = 1 - |\hat{\Sigma}_\varepsilon| / |\hat{\Sigma}_e| \qquad \zeta = \left(\frac{m^4 h^2 - 4}{m^2 + (mh)^2 - 5} \right)^{\frac{1}{2}}$$
$$\eta = \frac{1}{2} m^2 h - 1 \qquad \kappa = T - m - mh - \frac{1}{2}(m - mh - 1),$$

which converges to an F random variable under H_0.

Normality: Jarque-Bera Test

- This a goodness-of-fit test of whether sample residuals have the skewness and kurtosis matching a normal distribution.
- The test statistic converges in distribution to a chi-squared random variable

  $$JB = \lambda_3 + \lambda_4 \xrightarrow{d} \chi^2_{2m},$$

  with

  $$\begin{aligned} \lambda_3 &= \tfrac{T}{6}\hat{b}_1'\hat{b}_1 \xrightarrow{d} \chi^2_m \\ \lambda_4 &= \tfrac{T}{24}(\hat{b}_2' - 3_m)'(\hat{b}_2' - 3_m) \xrightarrow{d} \chi^2_m, \end{aligned}$$

  where $\hat{b}_1$ and $\hat{b}_2$ are the third and fourth moment vectors of the standardized residuals, $\hat{e}^s_t = \hat{P}^{-1}(\hat{e}_t - \bar{\hat{e}})$.
- P is a matrix with positive diagonal elements such that $\hat{P}\hat{P}' = \hat{\Sigma}_e$. The Cholesky decomposition of $\hat{\Sigma}_e$ is an example of $\hat{P}$.
- The implementation in `vars` is via `normality.test()`.
- The test can be inaccurate unless there is a really large sample.

Heteroskedasticity: ARCH Test

- The ARCH test involves an auxiliary regression

$$\text{vech}(\hat{e}_t\hat{e}_t') = \eta + S_1\text{vech}(\hat{e}_{t-1}\hat{e}_{t-1}') + \ldots + S_q\text{vech}(\hat{e}_{t-q}\hat{e}_{t-q}') + \nu_t$$

where the following hypothesis are tested

$$\begin{aligned} \text{H}_0 &: S_1 = 0 \cap \ldots \cap S_q = 0 \text{ (all } S_j \text{ are zero)} \\ \text{H}_A &: S_1 \neq 0 \cup \ldots \cup S_q \neq 0 \text{ (at least one } S_j \text{ is not zero).} \end{aligned}$$

- If H_0 is true, then the test statistic shown below converges to a chi-squared random variable

$$ARCH(q) = \tfrac{1}{2}Tm(m+1)\theta \xrightarrow{d} \chi^2_{qm^2(m+1)^2/4}$$

with

$$\theta = 1 - \frac{2}{m(m+1)}\text{tr}(\hat{\Omega}\hat{\Omega}_0^{-1}),$$

and we say that there are no ARCH effects (i.e., heteroskedasticity).

- Using `vars`, it is performed via the command `arch.test()`.

Unit Root Tests: Augmented Dickey–Fuller (ADF) Test

- The R library **`urca`** implements various unit root tests.
- In particular, we'll focus on the ADF test.
- The command `ur.df` allows for three different specifications

$$\textbf{Case 1}: \Delta y_t = \pi y_{t-1} + \sum_{j=1}^{k} \gamma_j \Delta y_{t-j} + u_t$$

$$\textbf{Case 2}: \Delta y_t = \beta_1 + \pi y_{t-1} + \sum_{j=1}^{k} \gamma_j \Delta y_{t-j} + u_t$$

$$\textbf{Case 3}: \Delta y_t = \beta_1 + \beta_2 t + \pi y_{t-1} + \sum_{j=1}^{k} \gamma_j \Delta y_{t-j} + u_t.$$

- The number of lags k can be chosen by information criteria.

Unit Root Test: Augmented Dickey–Fuller Test

- The null hypothesis are as follows depending on each case

$$\textbf{Case 1}: \quad \tau_1 : \pi = 0.$$

$$\begin{aligned} \textbf{Case 2}: \quad & \tau_2 : \pi = 0 \\ & \phi_1 : \pi = 0 \ \cap \ \beta_1 = 0. \end{aligned}$$

$$\begin{aligned} \textbf{Case 3}: \quad & \tau_3 : \pi = 0 \\ & \phi_2 : \pi = 0 \ \cap \ \beta_2 = 0 \\ & \phi_3 : \pi = 0 \ \cap \ \beta_2 = 0 \ \cap \ \beta_1 = 0. \end{aligned}$$

- The ADF test is a **left-tailed** test. The null hypothesis is rejected if the test statistic is larger than the specified critical value for a given significance level.
- A general drawback of unit root tests is that they have low power against persistent stationary processes.