SGPE QM Lab 5, Part 2: Time Series Analysis
Mark Schaffer
version of 23.10.2011

Introduction

In this lab we look at some of the tools of time-series analysis, with a particular focus on non-stationary data. The lab has two sections.

In the first section, we consider the implications of *nonstationarity* for regressions estimated by OLS. Nonstationarity is a key concept in time-series econometrics. If the data you are using are nonstationary, and you don't take this into account when doing your estimations, you can easily end up with results that are nonsense without realising it.

In the second section, we walk through the analysis of US real GDP in the lecture notes, demonstrating which Stata commands are used. We will approximately replicate the results in Prof. Molana's notes for Section 3 of QM1, "Steps in modelling a time series as a univariate process". This is covered in Prof. Molana's notes starting on p. 83, and in the slides starting with slide 145.

Section 1: Nonstationarity – a Monte Carlo analysis

A stochastic process is stationary if the joint distribution is the same whenever you look at it. For example, a process x_t is covariance stationary if (a) $E(x_t)$ is constant; (b) $Var(x_t)$ is constant; (c) for any t, $s \geq 1$, $Cov(x_t , x_{t+h})$ depends only on h and not on t. If h=1, say, then the correlation between x yesterday and x today is always the same; it doesn't matter if "yesterday" is t=1 or t=1 million.

OLS relies heavily on the assumption of stationarity. If the data are non-stationary, OLS breaks down, in interesting ways that we will explore in this task.

Preparation

Copy the following file from the QM coursework folder to your Lab9 folder and open it in the do-file editor:

```
Lab5_stat.do
```

Monte Carlo

The basic form of the DGP (data-generating process) we will be using is extremely simple:

$$y_t = \beta_0 + \beta_1 x_t + u_t$$
$$\beta_0 = \beta_1 = 0$$
$$\Rightarrow$$
$$y_t = u_t$$

The dependent variable y_t is just noise, and the explanatory variable x_t should, on average, get a zero coefficient (be "insignificantly different from 0"). For example, if we use a 5% significance level and a 2-tailed test, we should incorrectly reject

H_0: $\beta_1 = 0$

only 5% of the time if the test is correctly sized.

We consider two different DGPs:

DGP 1 (stationary data):

$$x_t \sim N(0,1)$$
$$u_t \sim N(0,1)$$
$$y_t = u_t$$

implemented in Stata as

```
gen x = rnormal()
gen u = rnormal()
gen y = u
```

DGP 2 (non-stationary data):

$$e1_t \sim N(0,1)$$
$$x_1 = 0 \qquad \text{(initial value at t=1)}$$
$$x_t = x_{t-1} + e1_t \qquad \text{(for t>1)}$$
$$e2_t \sim N(0,1)$$
$$u_1 = 0 \qquad \text{(initial value at t=1)}$$
$$u_t = u_{t-1} + e2_t \qquad \text{(for t>1)}$$
$$y_t = u_t$$

implemented in Stata as

```
gen x = 0 if t==1             /*  Initial value of x is 0  */
gen e1 = rnormal()            /*  White noise error  */
replace x=L.x + e1 if t>1     /*  x is a random walk  */
gen u = 0 if t==1             /*  Initial value of u is 0  */
gen e2 = rnormal()            /*  White noise error  */
replace u=L.u + e2 if t>1     /*  u is a random walk  */
gen y = u
```

Note that in DGP 2, to generate the lagged values of x and u, we use Stata's lag operator L. We will discuss this and Stata's other time-series operators later in the lab.

We want to see whether the distribution of the OLS estimate of β_1 converges to the true value as $t \to \infty$. The MC program **mysimstat** therefore returns to **simulate** the OLS estimate and standard error for a sample size of t=50, t=100 and t=500. These are called **b50**, **b100** and **b500**, and **se50**, **se100** and **se500**, respectively.

The choice of DGP is controlled by the macro **$stationary** in the **simulate** block of code.

```
global stationary=1
```

means DGP 1 (stationary data) is used.

```
global stationary=0
```

means DGP 2 (non-stationary data) is used.

You should do the following for the first DGP (stationary data).

(a) Run the 5,000 simulations. Do this by executing the do file up to and including the **simulate** block of code. After the simulation, you have 5,000 estimates of the OLS estimates **b50**, **b100** and **b500**, and **se50**, **se100** and **se500**.
(b) **summarize** the OLS estimators of β_1. Are the means close to the true value of β_1?
(c) Plot the distributions of **b50**, **b100** and **b500** using the **twoway** graph command. Does the OLS estimator converge towards the true value of β_1 as the sample size increases?
(d) Next, consider the size properties of the OLS estimators for the 3 sample sizes, t=50, t=100 and t=500. Calculate t-statistics and p-values for a test that the estimator is equal to the true value of β_1. How often would you incorrectly reject the null hypothesis at the 5% significance level? 5% of the time, or more than that?
(e) Look at the distribution of the t-statistics for t=50, t=100 and t=500. Compare them to the standard normal. Do they change as the sample size increases? Should they?
(f) Graph the p-values for the 3 possibilities in (d). Is the distribution uniform?

Now repeat (a)-(f) above for DGP 2 (non-stationary data). Your results should change dramatically, especially those for (c) and (e). What do you conclude?

Section 2: Univariate time series

Preparation

If you haven't already done so, copy the following files from the QM coursework folder to your Lab9 folder:

```
Lab5_US_GDP.do
US_GDP_49_09.csv
```

The do file starts with the following lines:

```
capture log close
log using US_GDP, text replace
```

The second line starts a log file, i.e, a file with the output that Stata sends to the screen. You have seen the **capture** trick before: if a log is already started, close it; if a log isn't started, ignore the command.

The US GDP data are in a CSV ("comma-separate values") file exported from an Excel worksheet. In the do file you will find code that loads the data into Stata using the **insheet** command, and that tells Stata that the data are time-series data – so that Stata's time-series operators and commands will work.

Although the US GDP dataset comes with a variable **YEAR** that takes the values 1949, 1950, 1951, …, 2009, we won't use this. Instead, you will need to create a time-trend variable **t** that runs 1, 2, 3, …, 81. And since some of the analysis uses a quadratic time trend, you will also need to create a variable **tsq**. Finally, we will be working with the log of GDP, which in keeping with the lecture notes we will call **ly**. We need to create this variable, too.

Open Stata and execute the following lines. You should be familiar enough with Stata now to understand what is going on, and/or to know how to find out using Stata's on-line help. Note that our time-trend variable **t** will be =1 in 1949.

```
insheet using US_GDP_49_09.csv, names case clear
gen t = _n
tsset t
gen ly=ln(GDP)
label var ly "Log(GDP)"
```

Note: the results you will generate will often differ slightly from those in the lecture notes. This is because the results reported in the lecture notes come from an analysis using a dataset with observations back to 1929. This means that the time-trend variable is =1 in 1929 (and hence is =21 in 1949, which makes a difference for the results using a quadratic trend), and that the first difference of GDP is available for 1949 (whereas in your dataset, taking the first difference means you lose the first observation).

Task 1: Getting to know Stata's time-series operators

tsset t tells Stata the dataset is a time-series dataset and the date variable is **t**. This means that we can now use Stata's **time-series operators**.

Online help on Stata's time-series operators is available in the help topic **varlists** ("variable lists"); one of the few cases where using the obvious keywords in Stata's online help may not get you to the topic you want right away. The four time-series operators are **L**, **F**, **D** and **S**. This is a rare case where case doesn't matter – either upper-case or lower-case can be used (but upper-case is more readable). The time-series operators work like prefixes on variables, where a "." separates the prefix from the variable. For example,

```
list GDP  L.GDP  L2.GDP  L3.GDP
```

means list GDP and the one-period, two-period and three-period lags of GDP. Note that "**L.**" is the same thing as "**L1.**". **F** is the lead ("forward") operator, **D** is difference operator, and **S** is the "seasonal difference" operator. **Warning**: the number used with the difference operator is **the number of times you difference**, **not** the time gap in the difference. **D.GDP** is the difference between GDP at time t and GDP at time t-1, but **D2.GDP** is the difference of this difference and the preceding difference; it is **not** the difference between GDP at time t and GDP at time t-2.

List the following variables in Stata, or, if they don't yet exist, create and then list them. See if you understand what they mean (including why some observations are missing).

```
list YEAR  t  GDP  L.GDP  L2.GDP  L3.GDP

list YEAR  t  GDP  L.GDP  F.GDP

list YEAR  t  GDP  L.GDP  L2.GDP  L3.GDP

list YEAR  t  GDP  F.GDP  F2.GDP  F3.GDP

list YEAR  t  GDP  F.L.GDP

list YEAR  t  GDP  D.GDP

gen diff1 = GDP - L.GDP
list YEAR  t  GDP  D.GDP diff1

gen diff2 = GDP - L2.GDP
gen doublediff = diff1 - L.diff1
list YEAR  t  GDP  D.GDP  D2.GDP  diff2  doublediff
```

Task 2: Inspect the data

As the lecture notes make clear (see slide 145), the first step in modelling a time series as a univariate process (and indeed more generally) is to look at the data.

The graphs in Figure 8.1 (slide 147) are simple line graphs of US GDP and **ly** ≡ log(GDP). Replicate them:

```
line GDP YEAR, yline(0)

line ly YEAR, yline(0)
```

The **yline(0)** option draws a horizontal line at y=0.

The graphs in Figure 8.2 are line graphs of the change in GDP, $\Delta GDP \equiv GDP_t - GDP_{t-1}$, and of the change in log GDP, Δ**ly**. Replicate them, using the Stata time-series first-differences operator **D**.

```
line D.GDP YEAR

line D.ly YEAR
```

The lecture notes explain why, on the basis of these graphs, we opt for modelling **ly**. Can you see why without checking the notes?

Task 3: Modelling log GDP

Slides 87-90 discuss checking whether **ly** (=log GDP) has a deterministic trend so that it can be expressed as a trend-stationary (TS) process.

The easiest way to do this is to analyse two regression equations: (1) Δ**ly** regressed on the trend **t** and a constant; (2) **ly** regressed on a quadratic time trend and a constant, i.e., on **t** and **tsq**.

Estimate (1) (the first regression on slide 149, but the coefficients and t-stats will be slightly different for the reasons given above:

```
regress D.ly t
```

Note the use of Stata's first-differences operator **D**.

Estimate (2) and plot log GDP and its deterministic trend (replicating the first picture in Figure 9 on slide 150):

```
regress ly t tsq
predict lyhat, xb
line ly lyhat YEAR
```

Note that we could do this using just Stata's **twoway** graphing command. The first graph in () uses Stata's **qfit** for fitting a quadratic trend; the second uses **line** to graph the raw log(GDP) data:

```
twoway (line ly t) (qfit ly t)
```

Estimate (2) and plot detrended log GDP, i.e., the residuals (replicating the second picture in Figure 9). We will call detrended log GDP **uhat**:

```
regress ly t tsq
predict double uhat if e(sample), resid
label var uhat "Detrended log GDP"
line uhat YEAR, yline(0)
```

We use **"if e(sample)"** to restrict the residuals to just the estimated sample. We also use **double** precision because we will use the residuals later for testing purposes. Finally, we label **uhat** so that it is clearly named in our graphs.

The two possibilities discussed on slides 146-52 are that Δ**ly** is trend-stationary (TS) around a linear deterministic trend, i.e., that Δ**ly** is difference-stationary (DS), or **ly** is TS around a quadratic deterministic trend.

Task 4: Unit root tests

Step 2 in analysing a univariate time series is determining the nonstationary properties of the series. This is discussed on slides 153-59.

Consider the AR(1) process on slide 153:

$$(x_t - \mu_t) = \alpha(x_{t-1} - \mu_{t-1}) + \xi_t$$

where μ_t is a deterministic trend. Rewrite it for convenience using $\alpha \equiv 1+\lambda$ (see slide 154):

$$(x_t - \mu_t) = (1+\lambda)(x_{t-1} - \mu_{t-1}) + \xi_t$$

which can be reorganized as

$$\Delta(x_t - \mu_t) = \lambda(x_{t-1} - \mu_{t-1}) + \xi_t$$

or (see slide 156)

$$\Delta x_t = \lambda x_{t-1} + \mu_t - (1+\lambda)\mu_{t-1} + \xi_t$$

The Dickey-Fuller (DF) test is based on the unit root equation

$$\Delta x_t = \delta_0 + \delta_1 t + \lambda x_{t-1} + \xi_t$$

where ξ_t is a white-noise error.

The null hypothesis is H_0: $\lambda=0$, meaning $\alpha=1$ and the variable x contains a unit root and is non-stationary. It also means that $(x_t - \mu_t)$ will be difference-stationary.

The alternative hypothesis is H_1: $\lambda<0$, meaning $\alpha<1$ and the variable x is I(0) and trend-stationary.

We can estimate λ in the unit root equation above using OLS, but we *cannot* use the usual t or normal distributions for testing, i.e., deciding whether or not to reject the null. The reason is that under the null, the estimated λ from OLS has a non-standard distribution. Dickey and Fuller tabulated the critical values to use.

The three flavours of the DF test depend on whether (1) we are testing for a unit root and x_t is difference-stationary around a quadratic trend ($\delta_0 \neq 0$, $\delta_1 \neq 0$); (2) we are testing for a unit root and x_t is difference-stationary around a linear trend ($\delta_0 \neq 0$, $\delta_1 = 0$); (3) we are testing for a unit root and x_t is difference-stationary around a constant ($\delta_0 = \delta_1 = 0$). The DF critical values are different in the 3 cases.

The Dickey-Fuller test is available in Stata as the command **`dfuller`**. Below you will first replicate the underlying OLS regression – but this isn't good enough, because you still need the critical values, so you will then use **`dfuller`** to do the full test in one step.

Application to detrended log GDP (slide 164):

By construction, this is flavour 3 ($\delta_0 = \delta_1 = 0$). (Why?)

```
regress D.uhat L.uhat, nocons
dfuller uhat, nocons
```

Note the use of Stata's lag operator **`L`**. Note also the use of the **`nocons`** option in both cases; by default, Stata includes a constant in OLS regression, and the default for **`dfuller`** is difference-stationary around a linear trend (i.e., x is a random walk without drift).

What do you conclude on the basis of this test?

Application to log GDP, i.e., **ly** *(slide 164-66):*

Flavour 3 ($\delta_0 = 0$, $\delta_1 = 0$):

```
regress D.ly L.ly, nocons
dfuller ly, nocons
```

Flavour 2 ($\delta_0 \neq 0$, $\delta_1 = 0$):

```
regress D.ly L.ly
dfuller ly
```

Flavour 1 ($\delta_0 \neq 0$, $\delta_1 \neq 0$):

```
regress D.ly L.ly t
dfuller ly, trend
```

What do you conclude on the basis of these tests?

Task 5: Unit root tests (continued)

The Augmented Dickey-Fuller (ADF) test is discussed on slides 165-66. As its name suggests, it is based on augmenting the DF equation with one or more lags of Δx:

$$\Delta x_t = \delta_0 + \delta_1 t + \lambda x_{t-1} + [\beta_1 \Delta x_{t-1} + \beta_2 \Delta x_{t-2} + \ldots + \beta_\rho \Delta x_{t-\rho}] + \xi_t$$

Application to detrended log GDP (slide 166):

This is still flavour 3 ($\delta_0 = \delta_1 = 0$). (Why?) We use 1 lag, i.e., ρ=1:

```
regress D.uhat L.uhat LD.uhat, nocons
dfuller uhat, nocons lags(1)
```

Note the use of the combination first-difference operator **D** and the lag operator **L**. (Of course, in practice you wouldn't need to do this because **dfuller** does it for you.)

What do you conclude on the basis of this test?

Task 6: Correlogram analysis

Step 3 of the analysis of a univariate time series is discussed starting on slide 170. We want to determine the order of the AR and MA components of the variable. The tests above suggest that detrended log GDP (**uhat**) can be treated as a trend-stationary (TS) process, and we will work with that.

We begin by looking at the autocorrelation and partial autocorrelation coefficients. (The definitions of the autocorrelation coefficient (AC) ρ and the partial autocorrelation coefficient (PAC) α are on slides 78-80. The AC ρ_s is the correlation between y_t and y_{t-s}; the PAC α_s is the same thing but conditioned on the s-1 intervening values of y.)

The ACs and PACs can be listed using the **corrgram** command:

```
corrgram uhat
```

but often a graphical presentation is more informative. Look at the ACs (replicating the first graph in Figure 10 on slide 176):

```
ac uhat
```

The style of graph is different from that in the slides, but the content is the same.

Look at the PACs (replicating the second graph in Figure 10 on slide 176):

```
pac uhat
```

Calculate Box-Pierce-Lung statistic Q_H (see slide 173 for the definition and slide 176 for the values in this application):

```
wntestq uhat, lags(1)

wntestq uhat, lags(2)
```

This all suggests **uhat** is an AR(1) or AR(2). (Why?)

Estimate an AR(2) (replicating the equation on slide 177):

```
regress uhat L.uhat L2.uhat, nocons
```

Try experimenting with including a constant and/or further lags of **uhat**. What do you conclude?

Plot the fitted values and residuals (replicating Figure 11 on slide 178):

```
predict double fitted if e(sample), xb
predict double etahat if e(sample), resid
line uhat fitted YEAR
line etahat YEAR, yline(0)
```

Optional: how would you check that **etahat** is stationary?

In fact, the best way to estimate this in Stata is to use the **var** (vector autoregression – you will hear much more about VARs shortly) command, because it comes with useful postestimation commands, including graphing:

```
var uhat, lags(1/2) nocons dfk small
```

The **lags(1/2)** option says "use lags 1 and 2"; the **dfk** and **small** options replicate the default behaviour of **regress**, which reports t and F statistics with small sample degrees of freedom adjustments.

To check its stability, after estimating use the **varstable** command:

```
varstable
```

To look at the impulse response function (IRF), we first create a file with IRF results:

```
irf create ar2, set(myirf, replace)
```

And then look at the IRF and cumulative IRF (replicating Figure 12 on slide 184):

```
irf graph irf
irf graph cirf
```