Problem Set 6

This problem set will be due at 5pm on Tuesday May 8th. Hand in either in-class on prior Thursday or in my mailbox (6th floor, 19 West 4th Street). Show your work (required for full and part marks).

1 Theory Questions

Question 1

Many time series forecasters will blend an AR and an MA model into something called the $ARMA$ model. An $ARMA(1,1)$ includes one lag of the observed variable, Y_t and one lag of the shock e_t. Hence, this is like an $AR(1)$ with serially dependent errors. In this exercise we will explore a few key properties of this process.

Consider a simple $ARMA(1,1)$ model:

$$y_t = \beta_0 + \rho y_{t-1} + \varepsilon_t + \gamma \varepsilon_{t-1} \qquad \text{with } \varepsilon_t \sim N(0, \sigma^2) \text{ for } t = 1, \ldots, T$$

We will assume that this process is stationary. Our goal is then to derive the properties of the stationary ARMA(1,1). To be clear we have our usual assumption that the shock in each period, ε_t, is assumed to be random in that period and so is independent of y_{t-1} and ε_{t-1}. Further, we assume that all of our parameters are finite (so $\rho < \infty, \gamma < \infty, \sigma_\varepsilon^2 < \infty$).

Part (a): What is the mean of this process? Be clear where in your proof you impose stationarity.

Part (b): What is the variance? Be clear where in your proof you impose stationarity.

Part (c): What is the first autocovariance? Note that your answer must only be a function of the parameters $(\rho, \gamma, \sigma_\varepsilon^2)$.

Part (d): Under which restrictions on the vector (ρ, γ) will our ARMA(1,1) model be weakly dependent? (Note: this question is somewhat hard. A formal proof is not required, rather you can just calculate the second autocovariance and assume that the geometric pattern you observe between the first and second autocovariance continues forever – which it indeed does!)

Question 2

Suppose that we have a random walk with drift, so our model is:

$$Y_t = \alpha_0 + Y_{t-1} + e_t$$

where e_t is iid with zero mean and variance of σ_e^2 and is not serially correlated (so $Cov(e_t, e_{t-k}) = 0$ for all k). In addition, at the "start of time" $Y_0 = e_0$.

Part (a): Find the mean and variance of the random walk. Is the random walk stationary?

Part II. In part (a) you found that this random walk with drift is not stationary. It turns out it is also not weakly dependent (proof is very similar to a problem on Lecture 21, slide 4). Time series econometrics is clearly going to be hard here, so suppose that you decide to take first-differences of your outcome variable, so define $\Delta Y_t = Y_t - Y_{t-1}$, where Y_t is still given by the equation above (so $Y_t = \alpha_0 + Y_{t-1} + e_t$).

Part (b): Find the mean, variance and first autocovariance of ΔY_t.

Part (c): Is the model stationary? Is the model weakly dependent?

2 Practical Question

Question 3

US Unemployment and Inflation Data set **cpi_unemp.csv** was downloaded from the St Louis Fed's website. It contains monthly observations of the US CPI and unemployment rate, starting from January 1948 and ending in February 2011 for a total of 758 observations. It has three variables which are:

date: date of observation (month/day/year)
cpi: US inflation index (with base date of August 1983)
unrate: US unemployment rate

I will be nice and help you along for this practical question as we have not gone over time series data in `R` much in this course. So first, we tell R that we are working with monthly time series data starting from January 1948. To do so you will need to use the `ts` (could also use `zoo` command):

```
cpi_ts = ts(cpi_unemp, start=c(1948,1), frequency=12)
```

The first argument is the actually data series you want to denote as a time series. The start command tells `R` the starting year and the increments on year, while the final argument says that within each year there are 12 observations (since this is monthly).

Next we are going to generate lagged inflation (*dinf*) by transforming our inflation time series (*inf*) so it is the *log* difference of the consumer price index. To do so, run the following command in `R`:

```
inf <- diff(log(cpi_ts))*12
```

Note that this has created a dataset where both CPI and unemployment are in log differences. For now, we will just work with CPI, but for part (d) when I ask you to use unemployment data please be sure to use the original data set **cpi_ts**.

Part (a): For the dates beginning Jan 1974 and ending Jan 2011, run an AR(1), AR(3) and an AR(6) forecast models for the log change in CPI (a.k.a. inflation). Which model has the highest adjusted R^2?

To run these regressions, you will want to use the `dynlm` command which is part of the `dynlm` package.[1] Here is the syntax:

```
dm1 = dynlm(data=inf, cpi~L(cpi,1:k), start = 1974 + 0/12, end =
    2011 + 0/12)
```

where k is the number of lags and must be supplied by the user (notice that this *inside* the `L`). The start and end commands tell `R` which subset of the time series to use for estimation. Notice the notation here: adding $j/12$ from $j = 0, ..., 11$ is how one indexes months.[2] Once this is completed, the `summary` and `coefficients` commands can extract more information like the values of the coefficients, R^2, etc., as we have done for this entire course.

Part (b): Now use the AR(1), AR(3) and AR(6) models to forecast February 2011 log change in CPI (a.k.a. inflation). The easiest way to do this is to use the data from January 2011 (and going backwards the number of required lags) and by hand multiplying these data points with the estimated coefficients.[3]

Part (c): Report the forecast error for each of the AR(1), AR(3) and AR(6) models. Which model has the lowest forecast error? Note that we can find the forecast error since our data goes *through* February 2011 (but we only use data through January 2011 to predict our model).

Part (d): Using the unemployment time series (recall: go back to **cpi_ts** data set so your unemployment data is not log differenced), perform an AR(1) version of the Dickey-Fuller test at the 5% significance level without a trend on the unemployment series.

[1] So obviously run "install.packages("dynlm")" and "library(dynlm)" commands.

[2] So my current code starts with data from Jan 1 1974 and ends with data on Jan 1 2011.

[3] The inquisitive can try using the `predict` command – but I would not suggest this since it can be very finicky.