MATH 5075 R Project 8

Your Name Here

11/22/2016

*Remember: I expect to see commentary either in the text, in the code with comments created using `#`, or (preferably) both! **Failing to do so may result in lost points!***

*Because randomization is used in this assignment, I set the seed here, in addition to beginning each code block. **Do not change the seed!***

```
set.seed(10202016)
```

Problem

Consider the RCA(1) model:

$$x_t = (\rho + \phi_t)x_{t-1} + w_t$$

1. *Let ϕ_t be i.i.d for all $t \in \mathbb{Z}$, with $\phi_t \sim N(0, 1)$, likewise for w_t, and let all phi_t and w_t be independent of each other. What is the condition for which the RCA(1) model has a stationary solution (you do not need to prove the result; simply state it)? Assume $\rho = -0.1$. Use simulation to estimate the quantity that determines stationarity (attach a 95% confidence interval to this quantity, using some appropriate method), and decide if this model (with these particular parameters) is stationary.*

```
# Your code here
```

2. *In this part you will write a function similar in spirit to `arima.sim()` that simulates an RCA(1) process in a flexible manner; call this function `rca1.sim()`. The function takes the following arguments:*

- `rho`: *A numeric value corresponding to the value of ρ*
- `n`: *An integer representing number of realizations to simulate; this is `NULL` by default, but if `innov.coeff` and `innov.res` are supplied, then this argument will be ignored*
- `rand.gen.coeff`: *A function for randomly generating the coefficients of the model, ϕ_t; by default, this should be `rnorm`*
- `rand.gen.res`: *A function for randomly generating the residuals of the model, w_t; by default, this should be `rnorm`*
- `innov.coeff`: *An optional time series (that is, `ts`) object that, if supplied, will be used for ϕ_t instead of randomly generating its values via `rand.gen.coeff`; by default, if `rand.gen.coeff` is supplied, this argument is a randomly generated series*
- `innov.res`: *An optional time series (that is, `ts`) object that, if supplied, will be used for w_t instead of randomly generating its values via `rand.gen.res`; by default, if `rand.gen.res` is supplied, this argument is a randomly generated series*
- `n.start`: *An integer representing number of realizations to simulate for the "burn-in" period; should be 500 by default*
- `start.innov.coeff`: *If supplied, a time series object that will be used for the "burn-in" period, corresponding to the random coefficients; by default, if `rand.gen.coeff` is supplied, this argument is a randomly generated series*
- `start.innov.res`: *If supplied, a time series object that will be used for the "burn-in" period, corresponding to the residuals; by default, if `rand.gen.res` is supplied, this argument is a randomly generated series*
- `coeff.args`: *A list of named arguments corresponding to arguments to be passed to `rand.gen.coeff`; most usefully, if `rand.gen.coeff` is `rnorm`, you can include an argument `sd` to control the standard deviation of ϕ_t*

- `res.args`: *A list of named arguments corresponding to arguments to be passed to* ***`rand.gen.res`****; most usefully, if* ***`rand.gen.coeff`*** *is* ***`rnorm`****, you can include an argument* ***`sd`*** *to control the standard deviation of* w_t

The function returns a time series (that is, class ***`ts`****) object containing the simulated values.*

I have started writing this function, giving many of the parameters sensible values. Fill in the rest of the function so it works. ***(Hint: Your job is to generate an RCA(1) process using `start.innov.res`, `innov.res`, `start.innov.coeff`, and `innov.coeff`, which are already made for you, then return a `ts` object with the last `n` elements of the recursion.)***

```
rca1.sim <- function(rho = 0, n = NULL, rand.gen.coeff = rnorm, rand.gen.res = rnorm,
    innov.coeff = do.call(rand.gen.coeff, c(list(n), coeff.args)), innov.res = do.call(rand.gen.res,
        c(list(n), res.args)), n.start = 500, start.innov.coeff = do.call(rand.gen.coeff,
        c(list(n.start), coeff.args)), start.innov.res = do.call(rand.gen.res,
        c(list(n.start), res.args)), coeff.args = list(sd = 1), res.args = list(sd = 1)) {

    # Error checking
    if (length(start.innov.coeff) != length(start.innov.res)) {
        stop("start.innov.coeff and start.innov.res need to be of the same length")
    }
    if (length(innov.coeff) != length(innov.res)) {
        stop("innov.coeff and innov.res must be of the same length")
    }
    # Coerce innov.coeff, innov.res, start.innov.coeff, and start.innov.res to
    # be ts objects
    innov.coeff <- as.ts(innov.coeff)
    innov.res <- as.ts(innov.res)
    start.innov.coeff <- as.ts(start.innov.coeff)
    start.innov.res <- as.ts(start.innov.res)

    if (!is.null(n)) {
        n <- length(innov.res)
    }
    if (is.null(n.start)) {
        n.start <- length(start.innov.res)
    }

    # Your code here
}
```

3. *Using* ***`rca1.sim()`****, simulate RCA(1) processes with* $T = 100$ *observations each with the following characteristics:*

- $\rho = 0$, $phi_t \sim N(0, 1)$, $w_t \sim N(0, 1)$
- $\rho = .1$, $phi_t \sim N(0, 1)$, $w_t \sim N(0, 1)$
- $\rho = .5$, $phi_t \sim N(0, 1)$, $w_t \sim N(0, 1)$
- $\rho = -.1$, $phi_t \sim N(0, 1)$, $w_t \sim N(0, 1)$
- $\rho = -.5$, $phi_t \sim N(0, 1)$, $w_t \sim N(0, 1)$
- $\rho = .5$, $phi_t \sim N(0, .5^2)$, $w_t \sim N(0, 20)$
- $\rho = .5$, $phi_t \sim t(3)$, $w_t \sim N(0, 1)$
- $\rho = .5$, $phi_t \sim N(0, 1)$, $w_t \sim t(3)$

Use the default burn-in period, and report your results via a plot.

```
# Your code here
```

4. *Recall the least-squares and weighted-least-squares estimators for* ρ*. Write a function that, given a*

data set (presumably a `ts`*-class object), estimates* ρ *with the least-squares estimator and returns the estimate. Do the same with the weighted-least-squares estimator. Then simulate a sample from each process described in part 3, estimate* ρ *from each simulated data set with both estimators, and compare against what* ρ *is known to be. Which estimator seems to perform best?*

```
# Your code here
```

5. *The data set* `lynx` *contains the annual numbers of lynx trappings for 1821-1934 in Canada. D.F. Nicholls and D.G. Quinn in their 1982 monograph* Random coefficient autoregressive models: an introduction *suggested that an RCA(2) model would fit the data. Plot the original data and the log-transformed data, and find the least-squares estimate and weighted-least-squares estimates for* ρ *in the RCA(1) model, using the demeaned log-transformed data.*

```
# Your code here
```