Henrik Madsen
Time Series Analysis
Chapman Hall/CRC
ISBN 978-1-4200-5976

Simulation of AR(p) and Seasonal Processes[1]. Assignment related to Chapter 5 of the book

This assignments has four parts starting with some theoretical questions. Second part deals with simulation of AR-processes. The last two parts are small programming exercises formulating various models. You will study how the choice of coefficients (of the lag-polynomial) affects the structure of the simulated data and the autocorrelation functions. At the end there are some hints for using R (or Splus).

Part 1: Theoretical part

Exercise 1

Consider the following AR(2)-process:

$$X_t + \phi_1 X_{t-1} + \phi_2 X_{t-2} = \epsilon_t$$

where ϵ_t is white noise.

Question 1.

Determine and sketch the allowed area of the parameters, i.e. the set of (ϕ_1, ϕ_2), where the AR(2)-process is stationary.

Question 2.

Determine and sketch that part of the allowed area of the parameters, where the autocorrelation function shows damping harmonic oscillations.

Exercise 2

In establishing models to predict the water level, it is found that deviations Y_t between the (per hour) calculated water levels and the observed water

[1]The assignment is found at `www.imm.dtu.dk/~hm/time.series.analysis`

levels can be described by a linear process of the form

$$(1 - 0.8B)(1 - 0.2B^6)(1 - B)Y_t = \epsilon_t$$

where ϵ_t is white noise with the variance σ_ϵ^2. Based on about 1500 observations, it is found that

$$\hat{\sigma^2}_\epsilon = 0.31dm^2$$

Based on the observations of Y_t in Table 1, it is desired to predict the value of Y_t corresponding to $t = 12$, together with a 95% confidence interval for the prediction:

t (in hours)	1	2	3	4	5	6	7	8	9	10
y_t (in dm)	2	1	-1	-2	-3	1	4	4	0	-3

Table 1: Observed water levels.

Part 2: Simulating AR-processes

You can use the commands `filter` and `arima.sim` to simulate the AR-processes. Begin using the filter command. You should simulate a noise-process, say 1500 standard normal random variables,

```
e=rnorm(1500)
```

If you want to simulate the AR(2)-process

$$Y_t = a_1 Y_{t-1} + a_2 Y_{t-2} + e_t \tag{1}$$

first create a vector of the coefficients of the lag polynomial and then use the filter function as follows

```
a=c(a1,a2)
y=filter(e,filter=a,method=''recursive'')
```

Look at the result you got with this command. Since there are no Y_0, Y_{-1} values the recursive filter just uses the terms that do exist, i.e. the noise process, and sets the first needed values of Y to zero. AR-processes are assumed to have started at virtually $t = -\infty$. Therefore you should only use say the last 250 of the 1500 observations. By letting the process run for a while we hope that the initial value (e[1]) does not affect the latter part of the data that much and it is as if the process had started at $t = -\infty$.

Now try simulating AR(1)-processes with positive, negative, small and large coefficients and comment on the results. You could use $a_1 = 0.1, -0.1, 0.9, -0.9$ for example. How about if you try to simulate with a coefficient bigger than 1? What happens? Plot the simulated data sets and the estimated autocorrelation functions. Compare the plots to plots of the noise process. Now simulate an AR(2) process with coefficients of your choice. Make sure the process is stationary. This is not so difficult. You can just construct a characteristic polynomial by starting with $(x - \text{root1})(x - \text{root2})$ and expand the product. Plot the data and the autocorrelation function. If you like, you could try a couple of different values for a_1 and a_2. Comment.

You can also use the `arima.sim` command to simulate instead of the `filter` command. It has the advantage that it will tell you if you pick a non-stationary set of coefficients. Also, it will automatically let the process run for a while so that the end result you get does not depend on an initial value. If you were to simulate the AR(2) process above with arima.sim you would write

```
y=arima.sim(model=list(ar=c(a1,a2)),n=250)
```

Try using arima.sim to simulate your AR(2) process and change the coefficients `a1` and `a2` to something else. Comment.

Part 3: Random Walk

If you were given a process

$$Y_t = \sum_{s=0}^{t} e_s \tag{2}$$

where e is the generated white noise process, and you were asked to find the mean value, variance and covariance-function of the process, what's

your answer? Simulate a noise process of 3000 values with mean zero and variance σ^2. Then figure out how to simulate the Y-process in R as simply as possible. Make the simulated data into a time series with the command `ts`. Then plot the data (using `ts.plot`) and the autocorrelation function. What do you notice? Is the series stationary? If so, in what sense? If not, in what sense? You might want to simulate a couple of these series, say 10 of them, and plot them together in the same plot. This should clarify the matter. What if you constructed a series X_t from the Y-process by taking first differences:

$$X_t = Y_t - Y_{t-1} \tag{3}$$

is the X-process stationary? Look up the command `diff`. Plot the new time series X and its autocorrelation function. You could also do this with the 10 Y series and create 10 X series. Comment.

Part 4: Seasonal processes

Remember that a process $\{Y_t\}$ is said to follow a multiplicative $(p, d, q) \times (P, D, Q)_s$ seasonal model if

$$\varphi(B)\phi(B^s)\nabla^d\nabla_s^D Y_t = \theta(B)\Theta(B^s)\varepsilon_t \tag{4}$$

where $\{\varepsilon_t\}$ is white noise, and φ and θ are polynomials of order p and q, respectively. Furthermore, φ and Θ are polynomials in B^s defined by

$$\phi(B^s) = 1 + \phi_1 B^s + \cdots + \phi_P B^{sP}, \tag{5}$$
$$\Theta(B^s) = 1 + \Theta_1 B^s + \cdots + \Theta_Q B^{sQ}. \tag{6}$$

and the seasonal difference operator is

$$\nabla_s = (1 - B^s) \tag{7}$$

Now try simulating the following models (where monthly data is assumed) and plot the associated autocorrelation function.

1. A $(1, 0, 0) \times (0, 0, 0)_{12}$ seasonal model with the parameter $\varphi_1 = 0.9$.
2. A $(0, 0, 0) \times (1, 0, 0)_{12}$ model with $\phi_1 = 0.7$.

3. A $(1,0,0) \times (0,0,1)_{12}$ model with $\varphi_1 = 0.9$ and $\Theta_1 = 0.4$.
4. A $(1,0,0) \times (1,0,0)_{12}$ model with $\varphi_1 = 0.9$ and $\phi_1 = 0.7$.
5. A $(0,0,1) \times (0,0,1)_{12}$ model with $\theta_1 = 0.4$ and $\Theta_1 = 0.3$.
6. A $(0,0,1) \times (1,0,0)_{12}$ model with with $\theta_1 = 0.4$ and $\phi_1 = 0.7$.

Is it possible to draw some conclusions on the behavior of the autocorrelation function for seasonal models based on the simulations above?

Note: `arima.sim` can only take non-seasonal models, but given Equations (4)-(6) a combined polynomial can be found.

A few hints/commands for the R beginners

When quitting R you are asked if you want to save your workspace. Do this if you want to remember all the data next time you start R in the same directory.

```
ts.plot(y)                  % Plots y as a time series.
y                           % Print out y - and the associated
                            % elements.
names(y)                    % Print out the names of the associated elements of
acf(y)                      % Calculates and plots the
                            % autocorrelation for y.
arima.mle(..)               % Maximum Likelihood Estimates of
                            % parameters in an ARIMA model.
arima.diag(..)              % Model validation for an estimated
                            % ARIMA model.
arima.forecast(..)          % Make forecasts in an ARIMA process.
module()                    % Lists the available modules.
q()                         % Exit from R.
```

`help.search(''something'')` can be used to search for functions you don't know the name of. Finally, `apropos(''something'')` will list all functions where the search string is part of the name.