AUTOREGRESSIVE ESTIMATION AND PREDICTION EXERCISE

In the directory with this exercise there are data files for the weekly average of new Covid-19 infections in the US. These are daily data; the past week's average every day. You are going to work with the log of these data, and the first few entries include zeros and small numbers, so use only data from March 30, 2020 and later. The .RData file version of the data has them already logged, truncated to start with March 30, and structured as an R time series object.

Here is a statistics-package-free statement of the exercise.

(1) Fit a 9th order linear autoregressive model to these data, using a flat prior or default prior, assuming normally distributed disturbance terms, and conditioning on initial observations.
(2) Using the estimated AR coefficients, forecast the series over the next 270 days (around 9 months) after the end of the data (February 11, 2021). (This is a single forecast path, with no error bands.)
(3) Sample 1000 draws from the posterior distribution of AR coefficients and residual variance to generate a forecast with error bands around the forecast that reflects (only) the uncertainty about the parameters of the model. (Make these 90% and 68% bands.)
(4) Generate error bands that include effects both of uncertainty about the model parameters and uncertainty about future disturbance terms.
(5) Plot, on a single graph, 20 of the randomly drawn forecast time series that include the effects of both kinds of uncertainty. (This is a different way of visualizing forecast uncertainty.)
(6) Using the same draws from the posterior and future shocks, evaluate the posterior probability that the rate of new infections is smaller at the end of the forecast period than at the start.
(7) Using the unconditional joint pdf of the initial conditions implied by the point-estimate of the parameters, calculate how many standard deviations from the process mean is the initial observation. If your point estimates imply an unstable root, you won't be able to do this part, so start by calculating the roots and checking whether they are all in the stable region.
(8) Use a histogram of the residuals and a normal q-q plot of them to assess whether the normality assumption is a reasonable approximation.

R tools for the exercise

While this exercise is doable in principle with any matrix programming language or programmable statistical package, there is an R package, IDex2019, on the exercise website, that contains programs that minimize the required programming work.

Date: February 16, 2021.

To install that package in R, you download the zip file and uncompress it. Then from within R the line `install.packages(''IDex2019'', repos=NULL, type=source)` installs it. If R's working directory is not the directory containing the decompressed IDex2019 folder, you need to give the full path to that folder as the quoted argument. Then `library(IDex2019)` loads the package. Documentation for the functions in the package is available via the usual R help system.

Estimation of the model can be done with a call to `rfvar3()`. By default `rfvar3()` uses an improper prior that shrinks toward persistence. You can omit that prior by setting `lambda=NULL, mu=NULL`. You can choose whether or not to use the prior. Generation of draws from the posterior for the parameters can then be done with a call to `postdraw()` A forecast using a single set of parameter values can be generated with `fcast()`. A set of draws from the posterior on the forecast can be generated with `fcastBand()`. This program will make plots showing error bands and will return an array of sampled forecasts, unsorted. It can show uncertainty based on parameter values alone (with `whichs=0` or, with `whichs` left at its default value, based on both parameter and future shock uncertainty.

The IDex2019 package programs were built to work with VAR's, so using it with the model in this exercise, which has only a single variable, requires some care. The `y0` argument of the forecasting commands should be dimensioned as a 9 x 1 matrix. It should also be a time series object with start date and frequency. Check that this is true with `str(y0)`. If not, use `y0 <-ts(matrix(y0, ncol=1), end=c(2021,42), freq=365)`. The `c(2021, 42)` specifies day 42 (i.e. February 11) of 2021.