F70TS – Time Series
Computer Lab 3

In this computer lab we use R to make forecasts for an ARIMA process. We will first consider the case when the coefficient values are known, before moving to the case where the coefficients must be estimated from observed data, and examine the accuracy of the forecasts in each case.

Key R Commands The key R packages used in this lab are `forecast` and `ggplot2`. A brief introduction to some relevant commands is provided here, but for more information see the R documentation.

As in the first computer practical, we must simulate data using the `arima.sim()` function. As a simple example, the following generates a series of length 100 from an ARIMA(2,1,2) process:

```
n <- 100
model <- list(order = c(2,1,2), ar = c(0.1, 0.2), ma = c(0.3, 0.4))
generated_data <- arima.sim(model, n)
```

The `autoplot()` function from the `ggplot2` package is an easy way to visualise this data

```
autoplot(generated_data)
```


Before forecasting using an ARIMA model, it is first necessary to fit it to the data, using the function `arima()`.

```
arima_model <- arima(generated_data, order=c(2,1,2))
```


(**N.B.** The data provided to `arima()` must be a *Time-Series* object. A vector or matrix can be converted to a *Time-Series* using the function `ts()`.)

By default, `arima()` estimates the values of the coefficients from the data. The coefficients can alternatively be fixed by supplying them as an argument to the original function


```
arima_model <- arima(generated_data, order=c(2,1,2), fixed = c(phi1, phi2, psi1, psi2))
```

The following command simulates a future m steps from the time series, according to the ARIMA model fitted on the generated data.

```
simulate(arima_model, nsim=m)
```

A forecast from the time series model can then be produced using the function `forecast` from the `forecast` package. The following forecasts the next 10 timesteps of `arima_model`.

```
arima_forecast <- forecast(arima_model, h = 10)
```

The output of this forecast is a list including the forecasted mean, and the lower and upper limits of 80% and 95% prediction intervals. These can be accessed using `arima_forecast$mean`, `arima_forecast$lower` and `arima_forecast$upper` respectively. The forecast may be plotted using `autoplot`:

```
autoplot(arima_forecast)
```

Exercise

In this exercise we consider an ARIMA(2,1,2) process with coefficients $\phi_1 = 0.7$, $\phi_2 = 0.2$, $\psi_1 = 0.5$ and $\psi_2 = 0.8$, and $\epsilon_t \sim N(0, 1)$.

1. Simulate a time series of length $n = 1000$ from this process. Plot this data.

2. Fit an ARIMA(2,1,2) model to the first $k = 100$ datapoints of the series. Fix the coefficients of the model to take their true values. Provide the model with the true value of $\text{Var}(\epsilon_t)$ using the command

   ```
   your_model_name$sigma2 = 1
   ```

3. Use the function `forecast()` to obtain a 95% prediction interval for the value of X_{k+50}. Test whether the true value of X_{k+50} is contained in this interval. Produce a plot illustrating the forecasted prediction intervals.

4. Investigate the accuracy of these forecasts. Using the command `simulate(your_model_name, nsim = 100)` simulate another 100 steps of the time series. Repeating 10^4 times, calculate the proportion for which the simulated value of X_{k+50} lies within the 95% predicted interval.

   (If this takes a long time, try 10^3 simulations.)

5. Repeat (2)-(4) for a model where the coefficients and variance must be estimated from the data. For part (4) remember that the simulated data should still be generated according to the true model, rather than the model with estimated coefficients. How does the accuracy compare to the case when the coefficients and variance are known?

6. How does the accuracy change if you alter the number of datapoints, k, used to estimate the parameters? How does the accuracy for $k = 10$ compare to $k = 900$?