# ABS State space workshop

Lab session 2 — 30 May 2014

---

Before doing any exercises in R, load the **fpp** package using `library(fpp)`.

1. Use `StructTS` to forecast each of the time series you used in the last lab session. How do your results compare to those obtained earlier in terms of their point forecasts and prediction intervals?

   Check the residuals of each fitted model to ensure they look like white noise.

2. For this exercise, use the monthly Australian short-term overseas visitors data, May 1985–April 2005. (Data set: `visitors` in `expsmooth` package.)

   (a) Use `StructTS` to fit a basic structural model to these data and record the training set RMSE.

   (b) We will now compare the model to the ETS model found earlier using time series cross-validation. The following code will compute the result, beginning with four years of data in the training set.

   ```
   k <- 48 # minimum size for training set
   n <- length(visitors) # Total number of observations
   e2 <- visitors*NA # Vector to record one-step forecast errors
   for(i in 48:(n-1))
   {
     train <- ts(visitors[1:i],freq=12)
     fit <- StructTS(train, type="BSM")
     fc <- forecast(fit,h=1)$mean
     e2[i] <- visitors[i+1]-fc
   }
   sqrt(mean(e2^2,na.rm=TRUE))
   ```

   (c) How does the RMSE computed in (2b) compare to that computed earlier? Does the structural model perform better than the ETS model?

3. In this exercise, you will write your own code for updating regression coefficients using a Kalman filter. We will model quarterly growth rates in US personal consumption expenditure ($y$) against quarterly growth rates in US real personal disposable income ($z$). So the model is $y_t = a + bz_t + \varepsilon_t$. The corresponding state space model is

$$
\begin{aligned}
y_t &= a_t + b_t z_t + \varepsilon_t \\
a_t &= a_{t-1} \\
b_t &= b_{t-1}
\end{aligned}
$$

which can be written in matrix form as follows:

$$
\begin{aligned}
y_t &= \boldsymbol{f}_t' \boldsymbol{x}_t + \varepsilon_t, & \varepsilon_t &\sim \mathrm{N}(0, \sigma_t^2) \\
\boldsymbol{x}_t &= \boldsymbol{G}_t \boldsymbol{x}_{t-1} + \boldsymbol{u}_t & \boldsymbol{w}_t &\sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{W}_t)
\end{aligned}
$$

where

$$
\boldsymbol{f}_t' = [1\ z_t], \quad \boldsymbol{x}_t = \begin{bmatrix} a_t \\ b_t \end{bmatrix}, \quad \boldsymbol{G}_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{W}_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.
$$

(a) Plot the data using `plot(usconsumption[,1],usconsumption[,2])` and fit a linear regression model to the data using the `lm` function.

(b) Write some R code to implement the Kalman filter using the above state space model. You can

(c) Estimate the parameters $a$ and $b$ by applying a Kalman filter and calculating $\hat{x}_{T|T}$. You will need to write your own code to implement the Kalman filter. [The only parameter that has not been specified is $\sigma^2$. It makes no difference what value you use in your code. Why?]

(d) Check that your estimates are identical to the usual OLS estimates obtained with the `lm` function?

(e) Use your code to obtain the sequence of parameter estimates given by $\hat{\boldsymbol{x}}_{1|1}, \hat{\boldsymbol{x}}_{2|2}, \ldots, \hat{\boldsymbol{x}}_{T|T}$.

(f) Plot the parameters over time. Does it appear that a model with time-varying parameters would be better?

(g) How would you estimate $\sigma^2$ using your code?

4. (a) Repeat Exercise 3, but this time use the **dlm** package to implement the Kalman filter.

   (b) Check that you get the same series of parameter estimates.