Chapter 7

Time Series

7.1 Key ideas

7.1.1 Time series operators

1. Back-shift operator, B. Moves the time index of an observation back by one time interval:

$$BY_t = Y_{t-1}.$$

   If we back-shift p times the result is

$$B^p Y_t = Y_{t-p}.$$

2. Differencing operator, $\Delta = 1 - B$. Takes the difference between an observation and the previous observation:

$$\Delta Y_t = (1 - B)Y_t = Y_t - Y_{t-1}$$

   The effect of differencing d times can be calculated by expanding the corresponding polynomial in B:

$$\Delta^d Y_t = (1 - B)^d Y_t = \sum_{j=0}^{d} \binom{d}{j} (-1)^j Y_{t-j}.$$

   The differencing operator is applied to models to reduce them to stationarity. It is often applied to data in an attempt to generate a series for which a stationary model is appropriate.

3. Seasonal differencing operator, $\Delta_s = 1 - B^s$. Takes the difference between two points in the same season:

$$\Delta_s Y_t = Y_t - Y_{t-s}.$$

4. Seasonal summation operator, $S(B) = 1 + B + B^2 + \ldots + B^{s-1}$. Adds points over the seasonal period:

$$S(B)Y_t = Y_t + Y_{t-1} + \ldots + Y_{t-s+1}$$

7.1.2 Covariance stationarity

Stationarity features prominently in the statistical theory of time series. A model $\{Y_t\}$ is covariance stationary if:

1. $E(Y_t) = \mu_Y$ a constant for all t,
2. $\text{Var}(Y_t) = \sigma_Y^2$ a constant for all t,
3. $\text{Corr}(Y_t, Y_{t-h})$ is a function of h alone for all t and integer h.

In summary, the mean, variance and the correlation structure of the series do not change over time.

7.1.3 Auto-correlation

The auto-correlation function (ACF) is of key interest in time series analysis. This function describes the strength of the relationships between different points in the series. For a covariance stationary time series model $\{Y_t\}$, the autocorrelation function (ACF), $\rho(.)$, is defined as

$$\rho(h) = \text{Corr}(Y_t, Y_{t-h}) \quad \text{for} \quad h = 0, 1, \ldots.$$

For a sample, $\{y_1, \ldots, y_n\}$ the sample estimate of the ACF is

$$\hat{\rho}(h) = r(h) = \frac{c(h)}{c(0)} \quad \text{where} \quad c(h) = \frac{1}{n} \sum_{i=h+1}^{n} (y_i - \bar{y})(y_{i-h} - \bar{y}).$$

7.1.4 Autoregressive moving average and related models

Models of the autoregressive moving average (ARMA) class are widely used to represent time series. They are not the only time series models available and are often not the most appropriate. However, they form a good starting point for the analysis of time series. In what follows we describe zero mean processes. The definitions are easily adapted to the non-zero mean case.

Time series are usually characterised by their dependence structure; what happens today is dependent on what happened yesterday. A simple way to model dependence on past observations is to use ideas from regression. We can build a simple regression model for our time series in which the explanatory variable is the observation immediately prior to our current observation. This is an autoregressive model of order 1, AR(1):

$$Y_t = \phi Y_{t-1} + \varepsilon_t, \quad \{\varepsilon_t\} \sim iN(0, \sigma_\varepsilon^2).$$

This idea can be extended to p previous observations, AR(p):

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t.$$

This can be written as

$$Y_t - \phi_1 Y_{t-1} - \phi_2 Y_{t-2} - \ldots - \phi_p Y_{t-p} = \varepsilon_t,$$

or

$$\phi(B) Y_t = \varepsilon_t,$$

where $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$. We can specify conditions on the function $\phi(.)$ in order to ensure that this model is stationary.

A more general class of model is defined by including dependence on past error terms. These are referred to as autoregressive moving average (ARMA) models. If we include dependence on the q previous error term, the result is an ARMA(p, q):

$$Y_t - \phi_1 Y_{t-1} - \ldots - \phi_p Y_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q},$$

or

$$\phi(B) Y_t = \theta(B) \varepsilon_t,$$

where $\theta(B) = 1 + \theta_1 B + \ldots + \theta_q B^q$.

Models which are stationary after differencing may be represented as autoregressive integrated moving averages (ARIMA). If $\{Y_t\} \sim$ ARIMA(p, d, q), then $\{\Delta^d Y_t\} \sim$ ARMA(p, q). This relationship is represented by the model equation

$$\phi(B) \Delta^d Y_t = \theta(B) \varepsilon_t.$$

In practice d is usually a small number ($d = 0$, 1 or 2).

The order of an ARIMA model is determined by three numbers p, d and q. If the process is seasonal, in addition to these numbers we have s the seasonal period, and P, D and Q the orders of the seasonal parts of the model. † The seasonal ARIMA$(p, d, q) \times (P, D, Q)_s$ model is

$$\Phi(B^s) \phi(B) \Delta_s^D \Delta^d Y_t = \Theta(B^s) \theta(B) \varepsilon_t,$$

where $\Phi(.)$ and $\Theta(.)$ are polynomials of order P and Q respectively.

7.1.5 Generalised autoregressive conditionally heteroskedastic models

Let $\{X_t\}$ be a zero mean time series and let $\mathcal{F}_t$ denote the the past of the series, $\{X_{t-i} : i = 0, 1, \ldots\}$. The generalised autoregressive conditionally heteroskedastic, GARCH(p, q), model is defined by

$$\begin{aligned} X_t &= \sigma_t \varepsilon_t, \\ \sigma_t^2 &= c + \sum_{i=1}^{p} b_i X_{t-i}^2 + \sum_{i=1}^{q} a_i \sigma_{t-i}^2, \end{aligned}$$

where $\sigma_t^2 = \mathrm{Var}(X_t | \mathcal{F}_{t-1}) = E(X_t^2 | \mathcal{F}_{t-1})$, and $\{\varepsilon_t\} \sim$ IID$(0, 1)$ with ε_t independent of $\mathcal{F}_{t-1}$. All of the parameters are non-negative, $c > 0$ and $\sum a_i + \sum b_i < 1$. In financial applications,

GARCH models are routinely fitted to the (log) returns for the price of an asset. The popularity of the GARCH class is due, at least in part, to the ease with which the likelihood is constructed; if $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a sequence of observed returns then, by prediction error decomposition,

$$\ell(\boldsymbol{\theta}) = \log f(\boldsymbol{x}|\boldsymbol{\theta}, \mathcal{F}_0) = \sum_{t=1}^{n} \left[\log \sigma_t - \log f_\varepsilon(x_t/\sigma_t)\right],$$

where the density function of the error terms, f_ε, is assumed known.

7.2 Time series objects

7.2.1 Creating and plotting time series objects

In order to illustrate some time series ideas we will use the data in the file `UKairpass.dat`. This file contains the monthly numbers of airline passengers (thousands) using UK airports from January 1995 to June 2003. It contains two series, one for domestic passengers and one for international passengers.

The function `ts(...)` provides a flexible mechanism for creating time series objects from vectors, matrices or data frames. We will illustrate using a data frame.

```
> UKairpass <- read.table("UKairpass.dat", header=TRUE)
> Domestic <- ts(UKairpass[1], start=1995, frequency=12)
> International <- ts(UKairpass[2], start=1995, frequency=12)
```

`Domestic` and `International` are now time series objects. The argument `frequency` refers to the seasonal period s. We can view time series objects by simply typing their name or use the function `tsp(...)` to extract the start, end and frequency. Notice how, since we specified `frequency = 12` when we created these objects, R assumes that the appropriate labels for the observations are month and year.

```
> Domestic
> tsp(Domestic)
```

The function `ts.plot(...)` will generate a time series plot. Below we add colour to differentiate between the series and include a legend.

```
> ts.plot(Domestic, International, col=c(2,3))
> legend(locator(n=1),legend=c("Domestic","International"), col=c(2,3),
+ lty=1)
```

7.2.2 Simple manipulation of time series objects

We can shift the series in time using the `lag(...)` function. This function has two arguments: the series that we want to shift and `k` the number of time steps we would like to push the series

into the past (the default is 1). This function works in a counter intuitive way; if we want the series back shift the series in the usual sense (so that Jan 1995 reading is at Feb 1995, and so on) we have to use `k=-1`.

```
> lag(Domestic)
> lag(Domestic, k=-1)
> lag(Domestic, k=-12)
```

In order to take differences we use the `diff(...)` function. The `diff(...)` function takes two arguments: the series we would like to difference and `lag`, the back-shift that we would like to perform (default is 1). Here `lag` argument of the `diff(...)` function (not to be confused with the `lag(...)` function!) is interpreted in the usual way so `diff(Domestic, lag=12)` is equal to `Domestic - lag(Domestic, k=-12)`. Note that this is seasonal differencing, Δ_{12}, not order 12 differencing Δ^{12}.

```
> diff(Domestic)
> ts.plot(diff(Domestic, lag=12), diff(International, lag=12), col=c(4,6))
> legend(locator(n=1),legend=c("Domestic","International"), col=c(4,6),
+ lty=1)
```

We can perform simple arithmetic and transformations using time series objects. The results are also time series objects.

```
> Total <- Domestic + International
> attributes(Total)
> ts.plot(Domestic, International, Total, col=c(2,3,4))
> logDomest <- log(Domestic)
> attributes(logDomest)
> ts.plot(logDomest, col=2)
```

7.3 Descriptive analysis and model identification

The function to generate the ACF in R is `acf(...)`. This function returns an object of mode list and (unless we alter the default setting) plots the ACF.

```
> acf(International)
> acfInt <- acf(International, plot=FALSE)
> mode(acfInt)
> attributes(acfInt)
> acfInt$acf
```

Note that 1.0 on the x-axis in this plot, denotes 1 year. We can use the time series plots and ACF plots to guide the process of model identification. The time series plot of `International` suggests a trend and seasonal pattern. We may remove these by differencing and look at the ACF of the resulting data.

```
> diffInt <- diff(International)
> diffIntSeas <- diff(diffInt, lag=12)
> acf(diffInt)
> acf(diffIntSeas)
```

This final plot indicates that after differencing and seasonal differencing there is still some significant correlation of points one year apart. † We may tentatively identify an ARIMA$(0,1,0) \times (0,1,1)_{12}$ on the basis of this plot.

In order to generate a multivariate time series from two series that are observed at the same time points we may use the `ts.union(...)` function. Multivariate time series may be passed as arguments to plotting functions. Passing a multivariate time series to the `acf(...)` function will draw individual ACF plots and also the cross-correlations. We can also do perform simple manipulation operations on a multivariate series; the result is that each of the component series will be altered.

```
> bothSeries <- ts.union(Domestic, International)
> ts.plot(bothSeries, col=c(3,4))
> acf(bothSeries)
> diffBothSeries <- diff(diff(bothSeries), lag=12)
> acf(diffBothSeries)
```

7.4 Fitting ARIMA models

There are functions available in R to fit autoregressive integrated moving average (ARIMA) models. We proceed by examples.

7.4.1 Simple AR and ARIMA for Nile data

For every year from 1871 to 1970 the volume of water flowing through the Nile at Aswan was recorded (I think the units are cubic kilometers). These 100 data points are in the file `nile.dat`.

```
> nile <- ts(read.table("nile.dat", header=TRUE), start=1871)
> ts.plot(nile, col=2)
> acf(nile, col=2)
```

We fit ARIMA models using the function `arima(...)`. The commands below fit an AR(1) and an ARIMA(0,1,1) to the Nile data. Note the brackets around the command; this is equivalent to performing the command without brackets then typing the name of the object just created. The `order` is a vector of three numbers $(p,\ d,\ q)$. Which of these models do you think is preferable?

```
> (nileAR1 <- arima(nile, order=c(1,0,0)))
> (nileARIMA011 <- arima(nile, order=c(0,1,1)))
```

One of the key questions in time series modelling is, does our model account adequately for

serial correlation in the observed series? The function `tsdiag(...)` will give an index plot of standardized residuals (over time), a plot of the sample ACF values of the residuals and a plot of the Ljung-Box statistic calculated from the residuals for a given number of lags (this number can be set using the `gof.lag` argument). Both sample ACF and Ljung-Box statistics are used to detect serial correlation in the residuals, that is, serial correlation that we have not accounted for using our model. The Ljung-Box statistic at lag h tests the hypothesis of no significant correlation below lag h. The commands below generate diagnostic plots for our two models. Does this change your opinion about which model is preferable?

```
> tsdiag(nileAR1)
> tsdiag(nileARIMA011)
```

One of the aims of time series modelling is to generate forecasts. In R this is done using the `predict(...)` function which takes a model object and a number (number of prediction) as arguments. The return value is a list with named components `pred` (the predictions) and `se` (the associated standard errors). Below we forecast 20 years beyond the end of the series for each of our models. What do you notice about these predictions? (A picture may help.) The predictions are very different; can you explain this?

```
> predict(nileAR1,20)$pred
> predict(nileARIMA011,20)$pred
```

7.4.2 Seasonal models for air passengers data

Seasonal ARIMA models are also fitted using the `arima(...)` function. The argument `seasonal` is a list with named elements `order` (the seasonal order of the process, $(P,\ D,\ Q)$) and `period` (the seasonal period). Earlier we indicated that an ARIMA$(0,1,0) \times (0,1,1)_{12}$ may be appropriate for the international air passengers data. We will fit this model and look at the properties of residuals. What do you notice in the index plot of the residuals?

```
> (intARIMA <- arima(International, order=c(0,1,0),
+ seasonal=list(order=c(0,1,1), period=12)))
> tsdiag(intARIMA)
```

A good way to represent forecasts is in a plot along with the observed data. Below we generate a plot of forecasts and an approximate 95% confidence interval for the forecasts.

```
> intfore <- predict(intARIMA, 36)
> ts.plot(International, intfore$pred, infore$pred+2*intfore$se,
+ infore$pred-2*intfore$se, lty=c(1,2,3,3), col=c(2,3,4,4))
```

7.5 Exercise

1. The file `rpi.dat` contains the monthly RPI for the UK (retail price index excluding housing costs) from January 1987 to August 2002 measured on a scale which takes the January

1987 figure to be 100. Generate a time series plot of the data. Generate an plot of the sample ACF for the original series, for the first differences and for the seasonal difference of the first difference series. Comment on these plots.

2. The file `passport.in7` contains the number of applications arriving at the UK passport service monthly from July 1993 to June 2002. Fit an ARIMA$(0,1,1) \times (0,1,1)_{12}$ model to these data and generate a plot of 5 years of forecasts. Is there anything in the residual plots for this model that makes you feel uneasy about your forecasts?

3. In this exercise I would like you to write some of your own time series functions.

   (a) The `lag(...)` function behaves counter intuitively (as described above). Write a simple function that takes a time series object and a positive integer as arguments and performs back-shifting operation in the normal way (should only take one line).

   (b) By default the `acf(...)` function plots included the autocorrelation at lag 1. This is irritating; we know that the ACF at lag 1 is equal to 1, we don't need it to be included on the plot. Try to write a function that plots the ACF from lag 2 upwards. [Hint: have a look at the attributes of the return value of the `acf(...)` `function`.]

   (c) Write a function that takes as arguments a time series and an integer, d, and difference the series d times, that is, performs the operation Δ^d.

   (d) † Write a function that fits several different ARIMA models to a data set a choose the one with minimum AIC.

4. A couple of interesting time series simulation problems (one easy, one tricky).

   (a) Simulate an AR(1), an ARIMA(1,1,0) and an ARIMA(1,2,0) with $n = 500$. Compare the time series plots and ACF plots for these series.

   (b) † Design and conduct a simulation experiments to find the properties of the parameter estimates of an ARMA(p, q) for different values of n, ϕ and θ.

7.6 Reading

7.6.1 Directly related reading

- Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*, 4th edition. Springer. [Sections 14.1, 14.2 and 14.3 but not the spectral stuff.]

7.6.2 Supplementary reading†

- Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*, 4th edition. Springer. [The remainder of chapter 14 interesting spectral and financial time series stuff.]

- Brockwell, P. J. and Davis, R. A. (1996) *Introduction to Time Series and Forecasting*. Springer-Verlag. [One of the best time series introductions.]