7

Non-stationary Models

7.1 Purpose

As we have discovered in the previous chapters, many time series are non-stationary because of seasonal effects or trends. In particular, random walks, which characterise many types of series, are non-stationary but can be transformed to a stationary series by first-order differencing (§4.4). In this chapter we first extend the random walk model to include autoregressive and moving average terms. As the differenced series needs to be aggregated (or 'integrated') to recover the original series, the underlying stochastic process is called autoregressive *integrated* moving average, which is abbreviated to ARIMA.

The ARIMA process can be extended to include seasonal terms, giving a non-stationary seasonal ARIMA (SARIMA) process. Seasonal ARIMA models are powerful tools in the analysis of time series as they are capable of modelling a very wide range of series. Much of the methodology was pioneered by Box and Jenkins in the 1970's.

Series may also be non-stationary because the variance is serially correlated (technically known as *conditionally heteroskedastic*), which usually results in periods of *volatility*, where there is a clear change in variance. This is common in financial series, but may also occur in other series such as climate records. One approach to modelling series of this nature is to use an autoregressive model for the variance, i.e. an autoregressive conditional heteroskedastic (ARCH) model. We consider this approach, along with the generalised ARCH (GARCH) model in the later part of the chapter.

7.2 Non-seasonal ARIMA models

7.2.1 Differencing and the electricity series

Differencing a series $\{x_t\}$ can remove trends, whether these trends are stochastic, as in a random walk, or deterministic, as in the case of a linear trend. In

P.S.P. Cowpertwait and A.V. Metcalfe, *Introductory Time Series with R*,
Use R, DOI 10.1007/978-0-387-88698-5_7,

the case of a random walk, $x_t = x_{t-1} + w_t$, the first-order differenced series is white noise $\{w_t\}$ (i.e., $\nabla x_t = x_t - x_{t-1} = w_t$) and so is stationary. In contrast, if $x_t = a + bt + w_t$, a linear trend with white noise errors, then $\nabla x_t = x_t - x_{t-1} = b + w_t - w_{t-1}$, which is a stationary moving average process rather than white noise. Notice that the consequence of differencing a linear trend with white noise is an MA(1) process, whereas subtraction of the trend, $a + bt$, would give white noise. This raises an issue of whether or not it is sensible to use differencing to remove a deterministic trend. The `arima` function in `R` does not allow the fitted differenced models to include a constant. If you wish to fit a differenced model to a deterministic trend using `R` you need to difference, then mean adjust the differenced series to have a mean of 0, and then fit an ARMA model to the adjusted differenced series using `arima` with `include.mean` set to `FALSE` and `d = 0`.

A corresponding issue arises with simulations from an ARIMA model. Suppose $x_t = a + bt + w_t$ so $\nabla x_t = y_t = b + w_t - w_{t-1}$. It follows directly from the definitions that the inverse of $y_t = \nabla x_t$ is $x_t = x_0 + \sum_{i=1}^{t} y_i$. If an MA(1) model is fitted to the differenced time series, $\{y_t\}$, the coefficient of w_{t-1} is unlikely to be identified as precisely -1. It follows that the simulated $\{x_t\}$ will have increasing variance (Exercise 3) about a straight line.

We can take first-order differences in `R` using the difference function `diff`. For example, with the Australian electricity production series, the code below plots the data and first-order differences of the natural logarithm of the series. Note that in the `layout` command below the first figure is allocated two 1s and is therefore plotted over half (i.e., the first two fourths) of the frame.

```
> www <- "http://www.massey.ac.nz/~pscowper/ts/cbe.dat"
> CBE <- read.table(www, he = T)
> Elec.ts <- ts(CBE[, 3], start = 1958, freq = 12)
> layout(c(1, 1, 2, 3))
> plot(Elec.ts)
> plot(diff(Elec.ts))
> plot(diff(log(Elec.ts)))
```

The increasing trend is no longer apparent in the plots of the differenced series (Fig. 7.1).

7.2.2 Integrated model

A series $\{x_t\}$ is *integrated* of order d, denoted as I(d), if the dth difference of $\{x_t\}$ is white noise $\{w_t\}$; i.e., $\nabla^d x_t = w_t$. Since $\nabla^d \equiv (1 - \mathbf{B})^d$, where $\mathbf{B}$ is the backward shift operator, a series $\{x_t\}$ is integrated of order d if

$$(1 - \mathbf{B})^d x_t = w_t \tag{7.1}$$

The random walk is the special case I(1). The `diff` command from the previous section can be used to obtain higher-order differencing either by repeated application or setting the parameter d to the required values; e.g.,

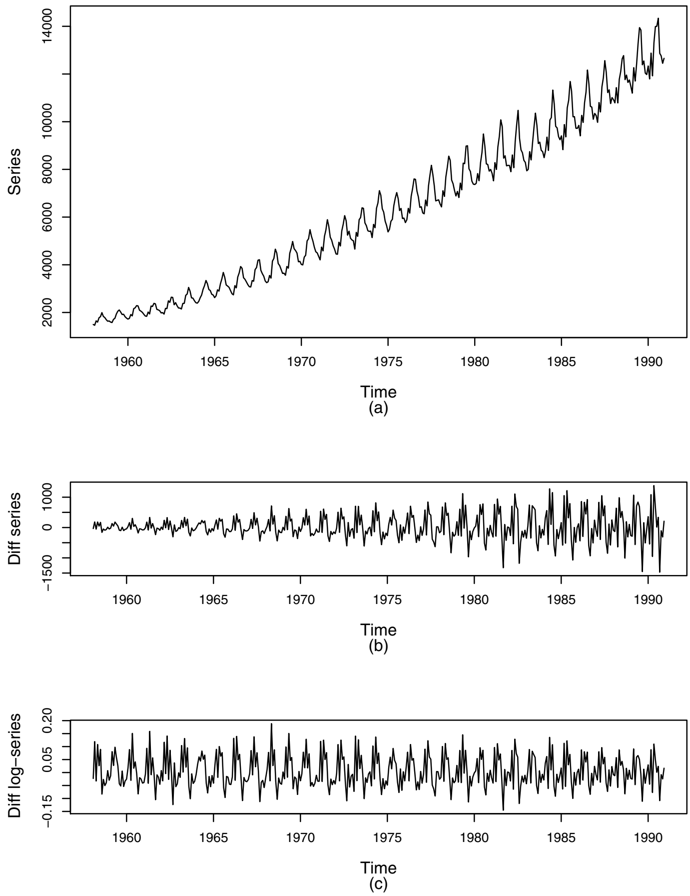


Fig. 7.1. (a) Plot of Australian electricity production series; (b) plot of the first-order differenced series; (c) plot of the first-order differenced log-transformed series.

`diff(diff(x))` and `diff(x, d=2)` would both produce second-order differenced series of `x`. Second-order differencing may sometimes successfully reduce a series with an underlying curve trend to white noise. A further parameter (`lag`) can be used to set the lag of the differencing. By default, `lag` is set to unity, but other values can be useful for removing additive seasonal effects. For example, `diff(x, lag=12)` will remove both a linear trend and additive seasonal effects in a monthly series.

7.2.3 Definition and examples

A time series $\{x_t\}$ follows an ARIMA(p, d, q) process if the dth differences of the $\{x_t\}$ series are an ARMA(p, q) process. If we introduce $y_t = (1 - \mathbf{B})^d x_t$, then $\theta_p(\mathbf{B})y_t = \phi_q(\mathbf{B})w_t$. We can now substitute for y_t to obtain the more succinct form for an ARIMA(p, d, q) process as

$$\theta_p(\mathbf{B})(1-\mathbf{B})^d x_t = \phi_q(\mathbf{B})w_t \tag{7.2}$$

where θ_p and ϕ_q are polynomials of orders p and q, respectively. Some examples of ARIMA models are:

(a) $x_t = x_{t-1}+w_t+\beta w_{t-1}$, where β is a model parameter. To see which model this represents, collect together like terms, factorise them, and express them in terms of the backward shift operator $(1-\mathbf{B})x_t = (1+\beta\mathbf{B})w_t$. Comparing this with Equation (7.2), we can see that $\{x_t\}$ is ARIMA(0, 1, 1), which is sometimes called an *integrated moving average* model, denoted as IMA(1, 1). In general, ARIMA(0, d, q) $\equiv$ IMA(d, q).

(b) $x_t = \alpha x_{t-1}+x_{t-1}-\alpha x_{t-2}+w_t$, where α is a model parameter. Rearranging and factorising gives $(1-\alpha\mathbf{B})(1-\mathbf{B})x_t = w_t$, which is ARIMA(1, 1, 0), also known as an integrated autoregressive process and denoted as ARI(1, 1). In general, ARI(p, d) $\equiv$ ARIMA(p, d, 0).

7.2.4 Simulation and fitting

An ARIMA(p, d, q) process can be fitted to data using the `R` function `arima` with the parameter `order` set to `c(p, d, q)`. An ARIMA(p, d, q) process can be simulated in `R` by writing appropriate code. For example, in the code below, data for the ARIMA(1, 1, 1) model $x_t = 0.5x_{t-1}+x_{t-1}-0.5x_{t-2}+w_t+0.3w_{t-1}$ are simulated and the model fitted to the simulated series to recover the parameter estimates.

```
> set.seed(1)
> x <- w <- rnorm(1000)
> for (i in 3:1000) x[i] <- 0.5 * x[i - 1] + x[i - 1] - 0.5 *
     x[i - 2] + w[i] + 0.3 * w[i - 1]
> arima(x, order = c(1, 1, 1))

Call:
arima(x = x, order = c(1, 1, 1))

Coefficients:
        ar1    ma1
      0.423  0.331
s.e.  0.043  0.045

sigma^2 estimated as 1.07:  log likelihood = -1450,  aic = 2906
```

Writing your own code has the advantage in that it helps to ensure that you understand the model. However, an ARIMA simulation can be carried out using the inbuilt `R` function `arima.sim`, which has the parameters `model` and `n` to specify the model and the simulation length, respectively.

```
> x <- arima.sim(model = list(order = c(1, 1, 1), ar = 0.5,
     ma = 0.3), n = 1000)
> arima(x, order = c(1, 1, 1))

Call:
arima(x = x, order = c(1, 1, 1))

Coefficients:
        ar1    ma1
      0.557  0.250
s.e.  0.037  0.044

sigma^2 estimated as 1.08:  log likelihood = -1457,  aic = 2921
```

7.2.5 IMA(1, 1) model fitted to the beer production series

The Australian beer production series is in the second column of the dataframe `CBE` in §7.2.1. The beer data is dominated by a trend of increasing beer production over the period, so a simple integrated model IMA(1, 1) is fitted to allow for this trend and a carryover of production from the previous month. The IMA(1, 1) model is often appropriate because it represents a linear trend with white noise added. The residuals are analysed using the correlogram (Fig. 7.2), which has peaks at yearly cycles and suggests that a seasonal term is required.

```
> Beer.ts <- ts(CBE[, 2], start = 1958, freq = 12)
> Beer.ima <- arima(Beer.ts, order = c(0, 1, 1))
> Beer.ima

Call:
arima(x = Beer.ts, order = c(0, 1, 1))

Coefficients:
          ma1
       -0.333
s.e.    0.056

sigma^2 estimated as 360:  log likelihood = -1723,  aic = 3451

> acf(resid(Beer.ima))
```

From the output above the fitted model is $x_t = x_{t-1} + w_t - 0.33w_{t-1}$. Forecasts can be obtained using this model, with t set to the value required for the forecast. Forecasts can also be obtained using the `predict` function in `R` with the parameter `n.ahead` set to the number of values in the future. For example, the production for the next year in the record is obtained using `predict` and the total annual production for 1991 obtained by summation:

```
> Beer.1991 <- predict(Beer.ima, n.ahead = 12)
> sum(Beer.1991$pred)
[1] 2365
```

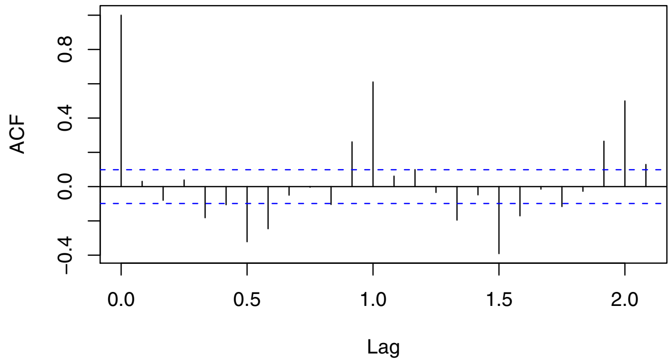


Fig. 7.2. Australian beer series: correlogram of the residuals of the fitted IMA(1, 1) model

7.3 Seasonal ARIMA models

7.3.1 Definition

A seasonal ARIMA model uses differencing at a lag equal to the number of seasons (s) to remove additive seasonal effects. As with lag 1 differencing to remove a trend, the lag s differencing introduces a moving average term. The seasonal ARIMA model includes autoregressive and moving average terms at lag s. The seasonal ARIMA(p, d, q)(P, D, Q)$_s$ model can be most succinctly expressed using the backward shift operator

$$\Theta_P(\mathbf{B}^s)\theta_p(\mathbf{B})(1-\mathbf{B}^s)^D(1-\mathbf{B})^d x_t = \Phi_Q(\mathbf{B}^s)\phi_q(\mathbf{B})w_t \tag{7.3}$$

where Θ_P, θ_p, Φ_Q, and ϕ_q are polynomials of orders P, p, Q, and q, respectively. In general, the model is non-stationary, although if $D = d = 0$ and the roots of the characteristic equation (polynomial terms on the left-hand side of Equation (7.3)) all exceed unity in absolute value, the resulting model would be stationary. Some examples of seasonal ARIMA models are:

(a) A simple AR model with a seasonal period of 12 units, denoted as ARIMA(0, 0, 0)(1, 0, 0)$_{12}$, is $x_t = \alpha x_{t-12} + w_t$. Such a model would be appropriate for monthly data when only the value in the month of the previous year influences the current monthly value. The model is stationary when $|\alpha^{-1/12}| > 1$.

(b) It is common to find series with stochastic trends that nevertheless have seasonal influences. The model in (a) above could be extended to $x_t = x_{t-1} + \alpha x_{t-12} - \alpha x_{t-13} + w_t$. Rearranging and factorising gives

$(1 - \alpha\mathbf{B}^{12})(1 - \mathbf{B})x_t = w_t$ or $\Theta_1(\mathbf{B}^{12})(1 - \mathbf{B})x_t = w_t$, which, on comparing with Equation (7.3), is ARIMA(0, 1, 0)(1, 0, 0)$_{12}$. Note that this model could also be written $\nabla x_t = \alpha\nabla x_{t-12} + w_t$, which emphasises that the *change* at time t depends on the change at the same time (i.e., month) of the previous year. The model is non-stationary since the polynomial on the left-hand side contains the term $(1 - \mathbf{B})$, which implies that there exists a unit root $B = 1$.

(c) A simple quarterly seasonal moving average model is $x_t = (1-\beta\mathbf{B}^4)w_t = w_t - \beta w_{t-4}$. This is stationary and only suitable for data without a trend. If the data also contain a stochastic trend, the model could be extended to include first-order differences, $x_t = x_{t-1} + w_t - \beta w_{t-4}$, which is an ARIMA(0, 1, 0)(0, 0, 1)$_4$ process. Alternatively, if the seasonal terms contain a stochastic trend, differencing can be applied at the seasonal period to give $x_t = x_{t-4} + w_t - \beta w_{t-4}$, which is ARIMA(0, 0, 0)(0, 1, 1)$_4$.

You should be aware that differencing at lag s will remove a linear trend, so there is a choice whether or not to include lag 1 differencing. If lag 1 differencing is included, when a linear trend is appropriate, it will introduce moving average terms into a white noise series. As an example, consider a time series of period 4 that is the sum of a linear trend, four additive seasonals, and white noise:

$$x_t = a + bt + s_{[t]} + w_t$$

where $[t]$ is the remainder after division of t by 4, so $s_{[t]} = s_{[t-4]}$. First, consider first-order differencing at lag 4 only. Then,

$$\begin{aligned}(1 - \mathbf{B}^4)x_t &= x_t - x_{t-4}\\ &= a + bt - (a + b(t-4)) + s_{[t]} - s_{[t-4]} + w_t - w_{t-4}\\ &= 4b + w_t - w_{t-4}\end{aligned}$$

Formally, the model can be expressed as ARIMA(0, 0, 0)(0, 1, 1)$_4$ with a constant term $4b$. Now suppose we apply first-order differencing at lag 1 before differencing at lag 4. Then,

$$\begin{aligned}(1 - \mathbf{B}^4)(1 - \mathbf{B})x_t &= (1 - \mathbf{B}^4)(b + s_{[t]} - s_{[t-1]} + w_t - w_{t-1})\\ &= w_t - w_{t-1} - w_{t-4} + w_{t-5}\end{aligned}$$

which is a ARIMA(0, 1, 1)(0, 1, 1)$_4$ model with no constant term.

7.3.2 Fitting procedure

Seasonal ARIMA models can potentially have a large number of parameters and combinations of terms. Therefore, it is appropriate to try out a wide range of models when fitting to data and to choose a best-fitting model using an appropriate criterion such as the AIC. Once a best-fitting model has been found, the correlogram of the residuals should be verified as white noise. Some confidence in the best-fitting model can be gained by deliberately overfitting the model by including further parameters and observing an increase in the AIC.

In R, this approach to fitting a range of seasonal ARIMA models is straightforward, since the fitting criteria can be called by nesting functions and the 'up arrow' on the keyboard used to recall the last command, which can then be edited to try a new model. Any obvious terms, such as a differencing term if there is a trend, should be included and retained in the model to reduce the number of comparisons. The model can be fitted with the `arima` function, which requires an additional parameter `seasonal` to specify the seasonal components. In the example below, we fit two models with first-order terms to the logarithm of the electricity production series. The first uses autoregressive terms and the second uses moving average terms. The parameter $d = 1$ is retained in both the models since we found in §7.2.1 that first-order differencing successfully removed the trend in the series. The seasonal ARI model provides the better fit since it has the smallest AIC.

```
> AIC (arima(log(Elec.ts), order = c(1,1,0),
                             seas = list(order = c(1,0,0), 12)))
[1] -1765
> AIC (arima(log(Elec.ts), order = c(0,1,1),
                             seas = list(order = c(0,0,1), 12)))
[1] -1362
```

It is straightforward to check a range of models by a trial-and-error approach involving just editing a command on each trial to see if an improvement in the AIC occurs. Alternatively, we could write a simple function that fits a range of ARIMA models and selects the best-fitting model. This approach works better when the conditional sum of squares method `CSS` is selected in the `arima` function, as the algorithm is more robust. To avoid over parametrisation, the *consistent* Akaike Information Criteria (CAIC; see Bozdogan, 1987) can be used in model selection. An example program follows.

```
get.best.arima <- function(x.ts, maxord = c(1,1,1,1,1,1))
{
  best.aic <- 1e8
  n <- length(x.ts)
  for (p in 0:maxord[1]) for(d in 0:maxord[2]) for(q in 0:maxord[3])
    for (P in 0:maxord[4]) for(D in 0:maxord[5]) for(Q in 0:maxord[6])
    {
       fit <- arima(x.ts, order = c(p,d,q),
                        seas = list(order = c(P,D,Q),
                        frequency(x.ts)), method = "CSS")
       fit.aic <- -2 * fit$loglik + (log(n) + 1) * length(fit$coef)
       if (fit.aic < best.aic)
       {
```

```
        best.aic <- fit.aic
        best.fit <- fit
        best.model <- c(p,d,q,P,D,Q)
      }
    }
  list(best.aic, best.fit, best.model)
}

> best.arima.elec <- get.best.arima( log(Elec.ts),
                                    maxord = c(2,2,2,2,2,2))
> best.fit.elec <- best.arima.elec[[2]]
> acf( resid(best.fit.elec) )
> best.arima.elec [[3]]

[1] 0 1 1 2 0 2

> ts.plot( cbind( window(Elec.ts,start = 1981),
                exp(predict(best.fit.elec,12)$pred) ), lty = 1:2)
```

From the code above, we see the best-fitting model using terms up to second order is ARIMA$(0, 1, 1)(2, 0, 2)_{12}$. Although higher-order terms could be tried by increasing the values in `maxord`, this would seem unnecessary since the residuals are approximately white noise (Fig. 7.3b). For the predicted values (Fig. 7.3a), a biased correction factor could be used, although this would seem unnecessary given that the residual standard deviation is small compared with the predictions.

7.4 ARCH models

7.4.1 S&P500 series

Standard and Poors (of the McGraw-Hill companies) publishes a range of financial indices and credit ratings. Consider the following time plot and correlogram of the daily returns of the S&P500 Index[1] (from January 2, 1990 to December 31, 1999), available in the `MASS` library within `R`.

```
> library(MASS)
> data(SP500)
> plot(SP500, type = 'l')
> acf(SP500)
```

The time plot of the returns is shown in Figure 7.4(a), and at first glance the series appears to be a realisation of a stationary process. However, on

[1] The S&P500 Index is calculated from the stock prices of 500 large corporations. The time series in `R` is the returns of the S&P500 Index, defined as $100\ln(\mathrm{SPI}_t/\mathrm{SPI}_{t-1})$, where SPI_t is the value of the S&P500 Index on trading day t.

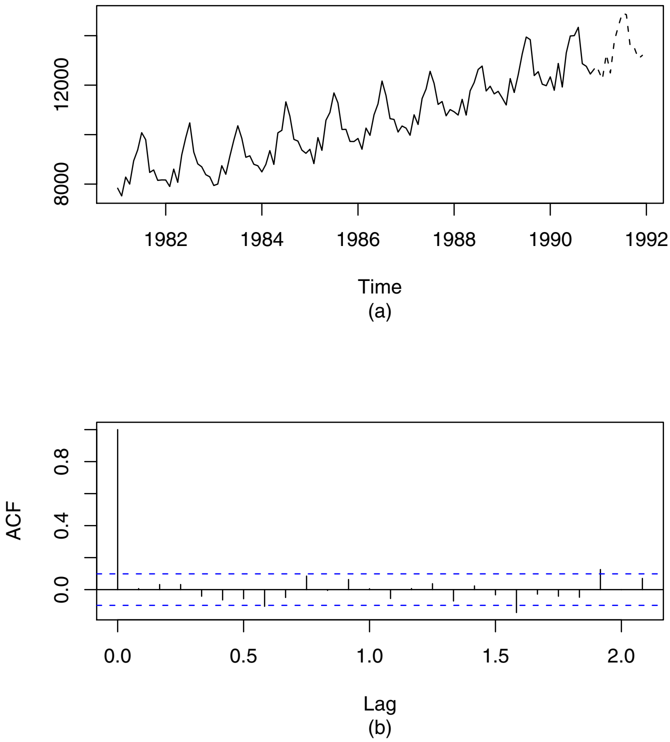


Fig. 7.3. Electricity production series: (a) time plot for last 10 years, with added predicted values (dotted); (b) correlogram of the residuals of the best-fitting seasonal ARIMA model.

closer inspection, it seems that the variance is smallest in the middle third of the series and greatest in the last third. The series exhibits periods of increased variability, sometimes called *volatility* in the financial literature, although it does not increase in a regular way. When a variance is not constant in time but changes in a regular way, as in the airline and electricity data (where the variance increased with the trend), the series is called *heteroskedastic*. If a series exhibits periods of increased variance, so the variance is correlated in time (as observed in the S&P500 data), the series exhibits volatility and is called *conditional heteroskedastic*.

Note that the correlogram of a volatile series does not differ significantly from white noise (Fig. 7.4b), but the series is non-stationary since the variance is different at different times. If a correlogram appears to be white noise (e.g., Fig. 7.4b), then volatility can be detected by looking at the correlogram of the squared values since the squared values are equivalent to the variance

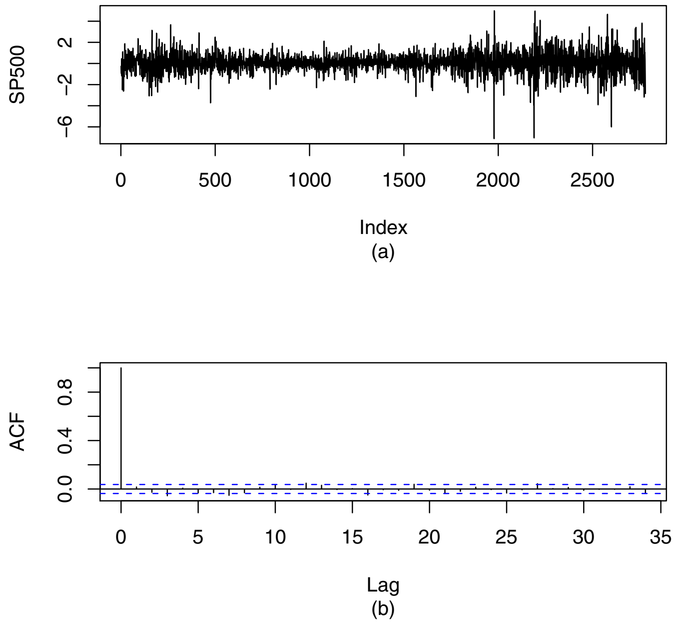


Fig. 7.4. Standard and Poors returns of the S&P500 Index: (a) time plot; (b) correlogram.

(provided the series is adjusted to have a mean of zero). The mean of the returns of the S&P500 Index between January 2, 1990 and December 31, 1999 is 0.0458. Although this is small compared with the variance, it accounts for an increase in the S&P500 Index from 360 to 1469 over the 2527 trading days. The correlogram of the squared mean-adjusted values of the S&P500 index is given by

```
> acf((SP500 - mean(SP500))^2)
```

From this we can see that there is evidence of serial correlation in the squared values, so there is evidence of conditional heteroskedastic behaviour and volatility (Fig. 7.5).

7.4.2 Modelling volatility: Definition of the ARCH model

In order to account for volatility, we require a model that allows for conditional changes in the variance. One approach to this is to use an autoregressive model for the variance process. This leads to the following definition. A series $\{\epsilon_t\}$ is first-order autoregressive conditional heteroskedastic, denoted ARCH(1), if

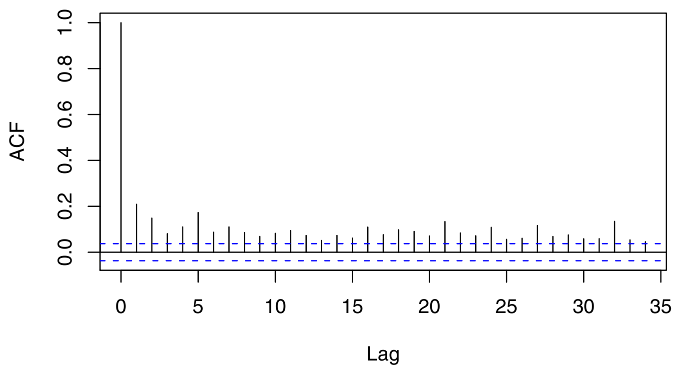


Fig. 7.5. Returns of the Standard and Poors S&P500 Index: correlogram of the squared mean-adjusted values.

$$\epsilon_t = w_t \sqrt{\alpha_0 + \alpha_1 \epsilon_{t-1}^2} \tag{7.4}$$

where $\{w_t\}$ is white noise with zero mean and unit variance and α_0 and α_1 are model parameters.

To see how this introduces volatility, square Equation (7.4) to calculate the variance

$$\begin{aligned}
\operatorname{Var}(\epsilon_t) &= E\left(\epsilon_t^2\right) \\
&= E\left(w_t^2\right) E\left(\alpha_0 + \alpha_1 \epsilon_{t-1}^2\right) \\
&= E\left(\alpha_0 + \alpha_1 \epsilon_{t-1}^2\right) \\
&= \alpha_0 + \alpha_1 \operatorname{Var}(\epsilon_{t-1})
\end{aligned} \tag{7.5}$$

since $\{w_t\}$ has unit variance and $\{\epsilon_t\}$ has zero mean. If we compare Equation (7.5) with the AR(1) process $x_t = \alpha_0 + \alpha_1 x_{t-1} + w_t$, we see that the variance of an ARCH(1) process behaves just like an AR(1) model. Hence, in model fitting, a decay in the autocorrelations of the *squared* residuals should indicate whether an ARCH model is appropriate or not. The model should only be applied to a prewhitened residual series $\{\epsilon_t\}$ that is uncorrelated and contains no trends or seasonal changes, such as might be obtained after fitting a satisfactory SARIMA model.

7.4.3 Extensions and GARCH models

The first-order ARCH model can be extended to a pth-order process by including higher lags. An ARCH(p) process is given by

$$\epsilon_t = w_t \sqrt{\alpha_0 + \sum_{i=1}^{p} \alpha_p \epsilon_{t-i}^2} \tag{7.6}$$

where $\{w_t\}$ is again white noise with zero mean and unit variance.

A further extension, widely used in financial applications, is the generalised ARCH model, denoted GARCH(q, p), which has the ARCH(p) model as the special case GARCH(0, p). A series $\{\epsilon_t\}$ is GARCH(q, p) if

$$\epsilon_t = w_t \sqrt{h_t} \tag{7.7}$$

where

$$h_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j h_{t-j} \tag{7.8}$$

and α_i and β_j ($i = 0, 1, \ldots, p$; $j = 1, \ldots, q$) are model parameters. In `R`, a GARCH model can be fitted using the `garch` function in the `tseries` library (Trapletti and Hornik, 2008). An example now follows.

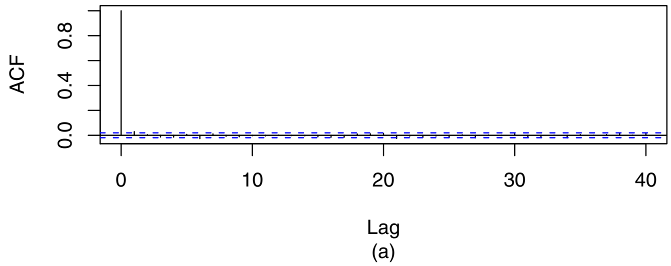
7.4.4 Simulation and fitted GARCH model

In the following code data are simulated for the GARCH(1, 1) model $a_t = w_t\sqrt{h_t}$, where $h_t = \alpha_0 + \alpha_1 a_{t-1} + \beta_1 h_{t-1}$ with $\alpha_1 + \beta_1 < 1$ to ensure stability; e.g., see Enders (1995). The simulated series are placed in the vector `a` and the correlograms plotted (Fig. 7.6).

```
> set.seed(1)
> alpha0 <- 0.1
> alpha1 <- 0.4
> beta1 <- 0.2
> w <- rnorm(10000)
> a <- rep(0, 10000)
> h <- rep(0, 10000)
> for (i in 2:10000) {
      h[i] <- alpha0 + alpha1 * (a[i - 1]^2) + beta1 * h[i -
          1]
      a[i] <- w[i] * sqrt(h[i])
  }
> acf(a)
> acf(a^2)
```

The series in `a` exhibits the GARCH characteristics of uncorrelated values (Fig. 7.6a) but correlated squared values (Fig. 7.6b).

In the following example, a GARCH model is fitted to the simulated series using the `garch` function, which can be seen to recover the original parameters since these fall within the 95% confidence intervals. The default is GARCH(1, 1), which often provides an adequate model, but higher-order models can be specified with the parameter `order=c(p,q)` for some choice of p and q.



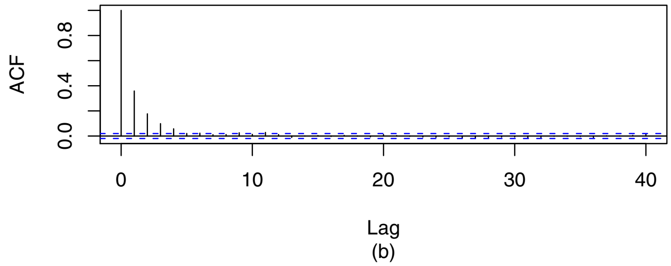


Fig. 7.6. Correlograms for GARCH series: (a) simulated series; (b) squared values of simulated series.

```
> library(tseries)
> a.garch <- garch(a, grad = "numerical", trace = FALSE)
> confint(a.garch)

    2.5 % 97.5 %
a0 0.0882  0.109
a1 0.3308  0.402
b1 0.1928  0.295
```

In the example above, we have used the parameter `trace=F` to suppress output and a numerical estimate of gradient `grad="numerical"` that is slightly more robust (in the sense of algorithmic convergence) than the default.

7.4.5 Fit to S&P500 series

The GARCH model is fitted to the S&P500 return series. The residual series of the GARCH model $\{\hat{w}_t\}$ are calculated from

$$\hat{w}_t = \frac{\epsilon_t}{\sqrt{\hat{h}_t}}$$

If the GARCH model is suitable the residual series should appear to be a realisation of white noise with zero mean and unit variance. In the case of a GARCH(1, 1) model,

$$\hat{h}_t = \hat{\alpha}_0 + \hat{\alpha}_1 \epsilon_{t-1}^2 + \hat{\beta}_1 \hat{h}_{t-1}$$

with $\hat{h}_1 = 0$ for $t = 2, \ldots, n$.[2] The calculations are performed by the function `garch`. The first value in the residual series is not available (`NA`), so we remove the first value using `[-1]` and the correlograms are then found for the resultant residual and squared residual series:

```
> sp.garch <- garch(SP500, trace = F)
> sp.res   <- sp.garch$res[-1]
> acf(sp.res)
> acf(sp.res^2)
```

Both correlograms suggest that the residuals of the fitted GARCH model behave like white noise, indicating a satisfactory fit has been obtained (Fig. 7.7).

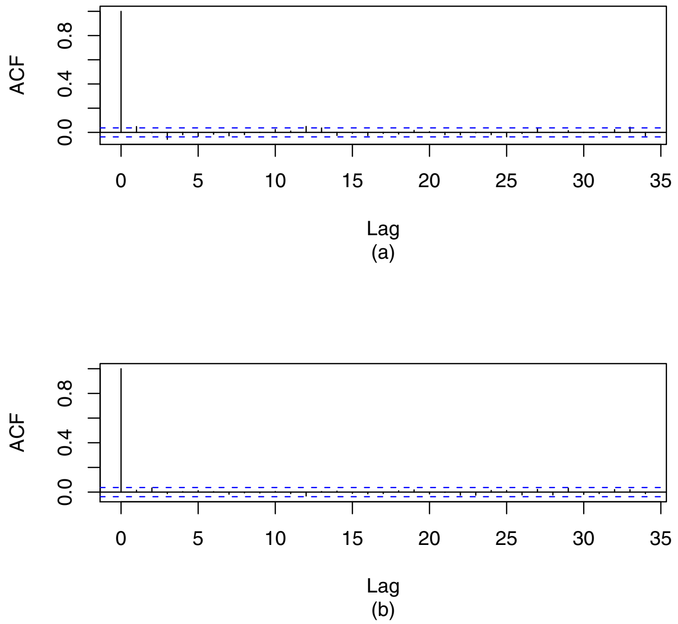


Fig. 7.7. GARCH model fitted to mean-adjusted S&P500 returns: (a) correlogram of the residuals; (b) correlogram of the squared residuals.

[2] Notice that a residual for time $t = 1$ cannot be calculated from this formula.

7.4.6 Volatility in climate series

Recently there have been studies on volatility in climate series (e.g., Romilly, 2005). Temperature data (1850–2007; see Brohan et al. 2006) for the southern hemisphere were extracted from the database maintained by the University of East Anglia Climatic Research Unit and edited into a form convenient for reading into R. In the following code, the series are read in, plotted (Fig. 7.8), and a best-fitting seasonal ARIMA model obtained using the `get.best.arima` function given in §7.3.2. Confidence intervals for the parameters were then evaluated (the transpose `t()` was taken to provide these in rows instead of columns).

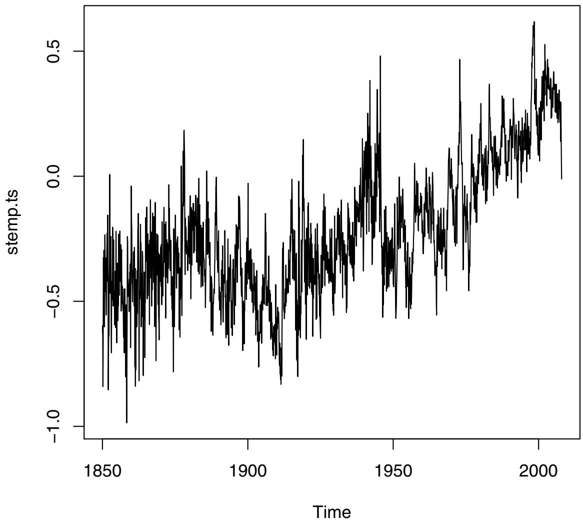


Fig. 7.8. The southern hemisphere temperature series.

```
> stemp <- scan("http://www.massey.ac.nz/~pscowper/ts/stemp.dat")
> stemp.ts <- ts(stemp, start = 1850, freq = 12)
> plot(stemp.ts)

> stemp.best <- get.best.arima(stemp.ts, maxord = rep(2,6))
> stemp.best[[3]]

[1] 1 1 2 2 0 1

> stemp.arima <- arima(stemp.ts, order = c(1,1,2),
                                  seas = list(order = c(2,0,1), 12))
```

```
> t( confint(stemp.arima) )

          ar1   ma1   ma2  sar1    sar2  sma1
2.5 %  0.832 -1.45 0.326 0.858 -0.0250 -0.97
97.5 % 0.913 -1.31 0.453 1.004  0.0741 -0.85
```

The second seasonal AR component is not significantly different from zero, and therefore the model is refitted leaving this component out:

```
> stemp.arima <- arima(stemp.ts, order = c(1,1,2),
                                  seas = list(order = c(1,0,1), 12))

> t( confint(stemp.arima) )

         ar1   ma1   ma2  sar1   sma1
2.5 %  0.83 -1.45 0.324 0.924 -0.969
97.5 % 0.91 -1.31 0.451 0.996 -0.868
```

To check for goodness-of-fit, the correlogram of residuals from the ARIMA model is plotted (Fig. 7.9a). In addition, to investigate volatility, the correlogram of the squared residuals is found (Fig. 7.9b).

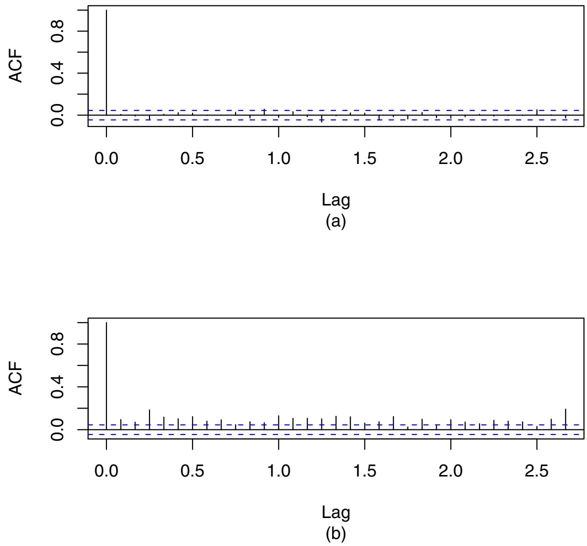


Fig. 7.9. Seasonal ARIMA model fitted to the temperature series: (a) correlogram of the residuals; (b) correlogram of the squared residuals.

```
> stemp.res <- resid(stemp.arima)
> layout(1:2)
```

```
> acf(stemp.res)
> acf(stemp.res^2)
```

There is clear evidence of volatility since the squared residuals are correlated at most lags (Fig. 7.9b). Hence, a GARCH model is fitted to the residual series:

```
> stemp.garch <- garch(stemp.res, trace = F)
> t(confint(stemp.garch))

             a0     a1    b1
2.5 %  1.06e-05 0.0330 0.925
97.5 % 1.49e-04 0.0653 0.963

> stemp.garch.res <- resid(stemp.garch)[-1]
> acf(stemp.garch.res)
> acf(stemp.garch.res^2)
```

Based on the output above, we can see that the coefficients of the fitted GARCH model are all statistically significant, since zero does not fall in any of the confidence intervals. Furthermore, the correlogram of the residuals shows no obvious patterns or significant values (Fig. 7.10). Hence, a satisfactory fit has been obtained.

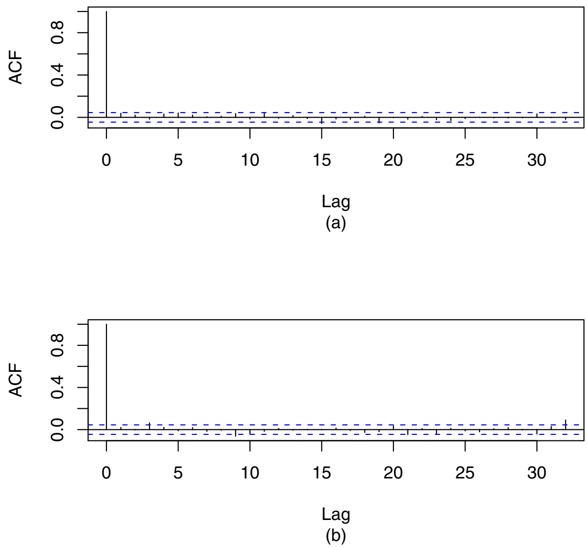


Fig. 7.10. GARCH model fitted to the residuals of the seasonal ARIMA model of the temperature series: (a) correlogram of the residuals; (b) correlogram of the squared residuals.

7.4.7 GARCH in forecasts and simulations

If a GARCH model is fitted to the residual errors of a fitted time series model, it will not influence the *average* prediction at some point in time since the mean of the residual errors is zero. Thus, single-point forecasts from a fitted time series model remain unchanged when GARCH models are fitted to the residuals. However, a fitted GARCH model will affect the variance of simulated predicted values and thus result in periods of changing variance or volatility in simulated series.

The main application of GARCH models is for simulation studies, especially in finance, insurance, teletraffic, and climatology. In all these applications, the periods of high variability tend to lead to untoward events, and it is essential to model them in a realistic manner. Typical `R` code for simulation was given in §7.4.4.

7.5 Summary of `R` commands

`garch` fits a GARCH (or ARCH) model to data

7.6 Exercises

In each of the following, $\{w_t\}$ is white noise with zero mean.

1. Identify each of the following as specific ARIMA models and state whether or not they are stationary.

   a) $z_t = z_{t-1} - 0.25z_{t-2} + w_t + 0.5w_{t-1}$

   b) $z_t = 2z_{t-1} - z_{t-2} + w_t$

   c) $z_t = 0.5z_{t-1} + 0.5z_{t-2} + w_t - 0.5w_{t-1} + 0.25w_{t-2}$

2. Identify the following as certain multiplicative seasonal ARIMA models and find out whether they are invertible and stationary.

   a) $z_t = 0.5z_{t-1} + z_{t-4} - 0.5z_{t-5} + w_t - 0.3w_{t-1}$

   b) $z_t = z_{t-1} + z_{t-12} - z_{t-13} + w_t - 0.5w_{t-1} - 0.5w_{t-12} + 0.25w_{t-13}$

3. Suppose $x_t = a + bt + w_t$. Define $y_t = \nabla x_t$.

   a) Show that $x_t = x_0 + \sum_{i=1}^{t} y_i$ and identify x_0.
   b) Now suppose an MA(1) model is fitted to $\{y_t\}$ and the fitted model is $y_t = b + w_t + \beta w_{t-1}$. Show that a simulated $\{x_t\}$ will have increasing variance about the line $a + bt$ unless β is precisely -1.

4. The number of overseas visitors to New Zealand is recorded for each month over the period 1977 to 1995 in the file `osvisit.dat` on the book website (http://www.massey.ac.nz/∼pscowper/ts/osvisit.dat). Download the file into `R` and carry out the following analysis. Your solution should include any `R` commands, plots, and comments. Let x_t be the number of overseas visitors in time period t (in months) and $z_t = \ln(x_t)$.
   a) Comment on the main features in the correlogram for $\{z_t\}$.
   b) Fit an ARIMA(1, 1, 0) model to $\{z_t\}$ giving the estimated AR parameter and the standard deviation of the residuals. Comment on the correlogram of the residuals of this fitted ARIMA model.
   c) Fit a seasonal ARIMA$(1, 1, 0)(0, 1, 0)_{12}$ model to $\{z_t\}$ and plot the correlogram of the residuals of this model. Has seasonal differencing removed the seasonal effect? Comment.
   d) Choose the best-fitting Seasonal ARIMA model from the following: ARIMA$(1, 1, 0)(1, 1, 0)_{12}$, ARIMA$(0, 1, 1)(0, 1, 1)_{12}$, ARIMA$(1, 1, 0)(0, 1, 1)_{12}$, ARIMA$(0, 1, 1)(1, 1, 0)_{12}$, ARIMA$(1, 1, 1)(1, 1, 1)_{12}$, ARIMA$(1, 1, 1)(1, 1, 0)_{12}$, ARIMA$(1, 1, 1)(0, 1, 1)_{12}$. Base your choice on the AIC, and comment on the correlogram of the residuals of the best-fitting model.
   e) Express the best-fitting model in part (d) above in terms of z_t, white noise components, and the backward shift operator (you will need to write this out by hand, but it is not necessary to expand all the factors).
   f) Test the residuals from the best-fitting seasonal ARIMA model for stationarity.
   g) Forecast the number of overseas visitors for each month in the next year (1996), and give the total number of visitors expected in 1996 under the fitted model. [Hint: To get the forecasts, you will need to use the exponential function of the generated seasonal ARIMA forecasts and multiply these by a bias correction factor based on the mean square residual error.]

5. Use the `get.best.arima` function from §7.3.2 to obtain a best-fitting ARIMA$(p, d, q)(P, D, Q)_{12}$ for all $p, d, q, P, D, Q \leq 2$ to the logarithm of the Australian chocolate production series (in the file at http://www.massey.ac.nz/∼pscowper/ts/cbe.dat). Check that the correlogram of the residuals for the best-fitting model is representative of white noise. Check the correlogram of the squared residuals. Comment on the results.

6. This question uses the data in `stockmarket.dat` on the book website http://www.massey.ac.nz/∼pscowper/ts/, which contains stock market

data for seven cities for the period January 6, 1986 to December 31, 1997. Download the data into R and put the data into a variable x. The first three rows should be:

```
> x[1:3,]
  Amsterdam Frankfurt London HongKong   Japan Singapore NewYork
1    275.76   1425.56 1424.1  1796.59 13053.8    233.63  210.65
2    275.43   1428.54 1415.2  1815.53 12991.2    237.37  213.80
3    278.76   1474.24 1404.2  1826.84 13056.4    240.99  207.97
```

a) Plot the Amsterdam series and the first-order differences of the series. Comment on the plots.
b) Fit the following models to the Amsterdam series, and select the best fitting model: ARIMA(0, 1, 0); ARIMA(1, 1, 0), ARIMA(0, 1, 1), ARIMA(1, 1, 1).
c) Produce the correlogram of the residuals of the best-fitting model and the correlogram of the squared residuals. Comment.
d) Fit the following GARCH models to the residuals, and select the best-fitting model: GARCH(0, 1), GARCH(1, 0), GARCH(1, 1), and GARCH(0, 2). Give the estimated parameters of the best-fitting model.
e) Plot the correlogram of the residuals from the best fitting GARCH model. Plot the correlogram of the squared residuals from the best fitting GARCH model, and comment on the plot.

7. Predict the monthly temperatures for 2008 using the model fitted to the climate series in §7.4.6, and add these predicted values to a time plot of the temperature series from 1990. Give an upper bound for the predicted values based on a 95% confidence level. Simulate ten possible future temperature scenarios for 2008. This will involve generating GARCH errors and adding these to the predicted values from the fitted seasonal ARIMA model.