
Stationary Models

6.1 Purpose

As seen in the previous chapters, a time series will often have well-defined components, such as a trend and a seasonal pattern. A well-chosen linear regression may account for these non-stationary components, in which case the residuals from the fitted model should not contain noticeable trend or seasonal patterns. However, the residuals will usually be correlated in time, as this is not accounted for in the fitted regression model. Similar values may cluster together in time; for example, monthly values of the Southern Oscillation Index, which is closely associated with El Niño, tend to change slowly and may give rise to persistent weather patterns. Alternatively, adjacent observations may be negatively correlated; for example, an unusually high monthly sales figure may be followed by an unusually low value because customers have supplies left over from the previous month. In this chapter, we consider stationary models that may be suitable for residual series that contain no obvious trends or seasonal cycles. The fitted stationary models may then be combined with the fitted regression model to improve forecasts. The autoregressive models that were introduced in §4.5 often provide satisfactory models for the residual time series, and we extend the repertoire in this chapter. The term *stationary* was discussed in previous chapters; we now give a more rigorous definition.

6.2 Strictly stationary series

A time series model $\{x_t\}$ is *strictly stationary* if the joint statistical distribution of x_{t_1}, \dots, x_{t_n} is the same as the joint distribution of $x_{t_1+m}, \dots, x_{t_n+m}$ for all t_1, \dots, t_n and m , so that the distribution is unchanged after an arbitrary time shift. Note that strict stationarity implies that the mean and variance are constant in time and that the autocovariance $\text{Cov}(x_t, x_s)$ only depends on lag $k = |t - s|$ and can be written $\gamma(k)$. If a series is not strictly stationary but the mean and variance are constant in time and the autocovariance only

depends on the lag, then the series is called *second-order* stationary.¹ We focus on the second-order properties in this chapter, but the stochastic processes discussed are strictly stationary. Furthermore, if the white noise is Gaussian, the stochastic process is completely defined by the mean and covariance structure, in the same way as any normal distribution is defined by its mean and variance-covariance matrix.

Stationarity is an idealisation that is a property of models. If we fit a stationary model to data, we assume our data are a realisation of a stationary process. So our first step in an analysis should be to check whether there is any evidence of a trend or seasonal effects and, if there is, remove them. Regression can break down a non-stationary series to a trend, seasonal components, and residual series. It is often reasonable to treat the time series of residuals as a realisation of a stationary error series. Therefore, the models in this chapter are often fitted to residual series arising from regression analyses.

6.3 Moving average models

6.3.1 MA(q) process: Definition and properties

A moving average (MA) process of order q is a linear combination of the current white noise term and the q most recent past white noise terms and is defined by

$$x_t = w_t + \beta_1 w_{t-1} + \dots + \beta_q w_{t-q} \quad (6.1)$$

where $\{w_t\}$ is white noise with zero mean and variance σ_w^2 . Equation (6.1) can be rewritten in terms of the backward shift operator \mathbf{B}

$$x_t = (1 + \beta_1 \mathbf{B} + \beta_2 \mathbf{B}^2 + \dots + \beta_q \mathbf{B}^q) w_t = \phi_q(\mathbf{B}) w_t \quad (6.2)$$

where ϕ_q is a polynomial of order q . Because MA processes consist of a finite sum of stationary white noise terms, they are stationary and hence have a time-invariant mean and autocovariance.

The mean and variance for $\{x_t\}$ are easy to derive. The mean is just zero because it is a sum of terms that all have a mean of zero. The variance is $\sigma_w^2(1 + \beta_1^2 + \dots + \beta_q^2)$ because each of the white noise terms has the same variance and the terms are mutually independent. The autocorrelation function, for $k \geq 0$, is given by

$$\rho(k) = \begin{cases} 1 & k = 0 \\ \frac{\sum_{i=0}^{q-k} \beta_i \beta_{i+k}}{\sum_{i=0}^q \beta_i^2} & k = 1, \dots, q \\ 0 & k > q \end{cases} \quad (6.3)$$

where β_0 is unity. The function is zero when $k > q$ because x_t and x_{t+k} then consist of sums of independent white noise terms and so have covariance

¹ For example, the skewness, or more generally $E(x_t x_{t+k} x_{t+l})$, might change over time.

zero. The derivation of the autocorrelation function is left to Exercise 1. An MA process is *invertible* if it can be expressed as a stationary autoregressive process of infinite order without an error term. For example, the MA process $x_t = (1 - \beta\mathbf{B})w_t$ can be expressed as

$$w_t = (1 - \beta\mathbf{B})^{-1}x_t = x_t + \beta x_{t-1} + \beta^2 x_{t-2} + \dots \quad (6.4)$$

provided $|\beta| < 1$, which is required for convergence.

In general, an MA(q) process is invertible when the roots of $\phi_q(B)$ all exceed unity in absolute value (Exercise 2). The autocovariance function only identifies a unique MA(q) process if the condition that the process be invertible is imposed. The estimation procedure described in §6.4 leads naturally to invertible models.

6.3.2 R examples: Correlogram and simulation

The autocorrelation function for an MA(q) process (Equation (6.3)) can readily be implemented in R, and a simple version, without any detailed error checks, is given below. Note that the function takes the lag k and the model parameters β_i for $i = 0, 1, \dots, q$, with $\beta_0 = 1$. For the non-zero values (i.e., values within the **else** part of the **if-else** statement), the autocorrelation function is computed in two stages using a **for** loop. The first loop generates a sum (**s1**) for the autocovariance, whilst the second loop generates a sum (**s2**) for the variance, with the division of the two sums giving the returned autocorrelation (**ACF**).

```
> rho <- function(k, beta) {
  q <- length(beta) - 1
  if (k > q) ACF <- 0 else {
    s1 <- 0; s2 <- 0
    for (i in 1:(q-k+1)) s1 <- s1 + beta[i] * beta[i+k]
    for (i in 1:(q+1)) s2 <- s2 + beta[i]^2
    ACF <- s1 / s2}
  ACF}
```

Using the code above for the autocorrelation function, correlograms for a range of MA(q) processes can be plotted against lag – the code below provides an example for an MA(3) process with parameters $\beta_1 = 0.7$, $\beta_2 = 0.5$, and $\beta_3 = 0.2$ (Fig. 6.1a).

```
> beta <- c(1, 0.7, 0.5, 0.2)
> rho.k <- rep(1, 10)
> for (k in 1:10) rho.k[k] <- rho(k, beta)
> plot(0:10, c(1, rho.k), pch = 4, ylab = expression(rho[k]))
> abline(0, 0)
```

The plot in Figure 6.1(b) is the autocovariance function for an MA(3) process with parameters $\beta_1 = -0.7$, $\beta_2 = 0.5$, and $\beta_3 = -0.2$, which has negative

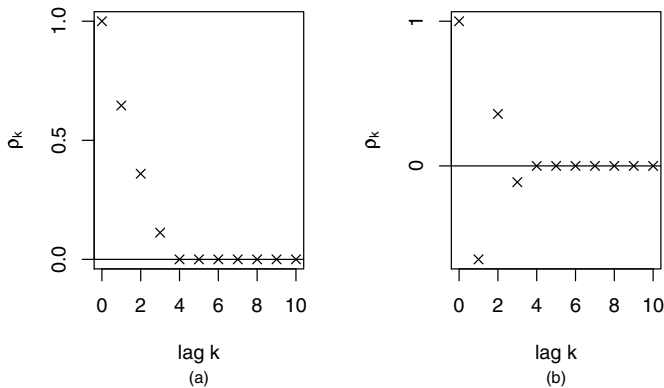


Fig. 6.1. Plots of the autocorrelation functions for two MA(3) processes: (a) $\beta_1 = 0.7$, $\beta_2 = 0.5$, $\beta_3 = 0.2$; (b) $\beta_1 = -0.7$, $\beta_2 = 0.5$, $\beta_3 = -0.2$.

correlations at lags 1 and 3. The function `expression` is used to get the Greek symbol ρ .

The code below can be used to simulate the MA(3) process and plot the correlogram of the simulated series. An example time plot and correlogram are shown in Figure 6.2. As expected, the first three autocorrelations are significantly different from 0 (Fig. 6.2b); other statistically significant correlations are attributable to random sampling variation. Note that in the correlogram plot (Fig. 6.2b) 1 in 20 (5%) of the sample correlations for lags greater than 3, for which the underlying population correlation is zero, are expected to be statistically significantly different from zero at the 5% level because multiple t-test results are being shown on the plot.

```
> set.seed(1)
> b <- c(0.8, 0.6, 0.4)
> x <- w <- rnorm(1000)
> for (t in 4:1000) {
  for (j in 1:3) x[t] <- x[t] + b[j] * w[t - j]
}
> plot(x, type = "l")
> acf(x)
```

6.4 Fitted MA models

6.4.1 Model fitted to simulated series

An MA(q) model can be fitted to data in R using the `arima` function with the `order` function parameter set to `c(0,0,q)`. Unlike the function `ar`, the

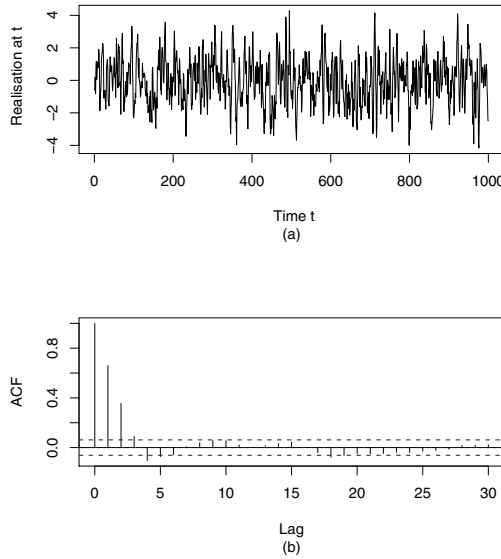


Fig. 6.2. (a) Time plot and (b) correlogram for a simulated MA(3) process.

function `arima` does not subtract the mean by default and estimates an intercept term. MA models cannot be expressed in a multiple regression form, and, in general, the parameters are estimated with a numerical algorithm. The function `arima` minimises the conditional sum of squares to estimate values of the parameters and will either return these if `method=c("CSS")` is specified or use them as initial values for maximum likelihood estimation.

A description of the conditional sum of squares algorithm for fitting an MA(q) process follows. For any choice of parameters, the sum of squared residuals can be calculated iteratively by rearranging Equation (6.1) and replacing the errors, w_t , with their estimates (that is, the residuals), which are denoted by \hat{w}_t :

$$S(\hat{\beta}_1, \dots, \hat{\beta}_q) = \sum_{t=1}^n \hat{w}_t^2 = \sum_{t=1}^n \left\{ x_t - (\hat{\beta}_1 \hat{w}_{t-1} + \dots + \hat{\beta}_q \hat{w}_{t-q}) \right\}^2 \quad (6.5)$$

conditional on $\hat{w}_0, \dots, \hat{w}_{t-q}$ being taken as 0 to start the iteration. A numerical search is used to find the parameter values that minimise this conditional sum of squares.

In the following code, a moving average model, `x.ma`, is fitted to the simulated series of the last section. Looking at the parameter estimates (coefficients in the output below), it can be seen that the 95% confidence intervals (approximated by $\text{coeff.} \pm 2 \text{ s.e. of coeff.}$) contain the underlying parameter values (0.8, 0.6, and 0.4) that were used in the simulations. Furthermore, also as expected,

the intercept is not significantly different from its underlying parameter value of zero.

```
> x.ma <- arima(x, order = c(0, 0, 3))
> x.ma
```

```
Call:
arima(x = x, order = c(0, 0, 3))
```

```
Coefficients:
      ma1      ma2      ma3  intercept
    0.790  0.566  0.396     -0.032
s.e.  0.031  0.035  0.032      0.090
```

```
sigma^2 estimated as 1.07:  log likelihood = -1452,  aic = 2915
```

It is possible to set the value for the mean to zero, rather than estimate the intercept, by using `include.mean=FALSE` within the `arima` function. This option should be used with caution and would only be appropriate if you wanted $\{x_t\}$ to represent displacement from some known fixed mean.

6.4.2 Exchange rate series: Fitted MA model

In the code below, an MA(1) model is fitted to the exchange rate series. If you refer back to §4.6.2, a comparison with the output below indicates that the AR(1) model provides the better fit, as it has the smaller standard deviation of the residual series, 0.031 compared with 0.042. Furthermore, the correlogram of the residuals indicates that an MA(1) model does not provide a satisfactory fit, as the residual series is clearly not a realistic realisation of white noise (Fig. 6.3).

```
> www <- "http://www.massey.ac.nz/~pscowper/ts/pounds_nz.dat"
> x <- read.table(www, header = T)
> x.ts <- ts(x, st = 1991, fr = 4)
> x.ma <- arima(x.ts, order = c(0, 0, 1))
> x.ma
```

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

```
Coefficients:
      ma1  intercept
    1.000      2.833
s.e.  0.072      0.065
```

```
sigma^2 estimated as 0.0417:  log likelihood = 4.76,  aic = -3.53
```

```
> acf(x.ma$res[-1])
```

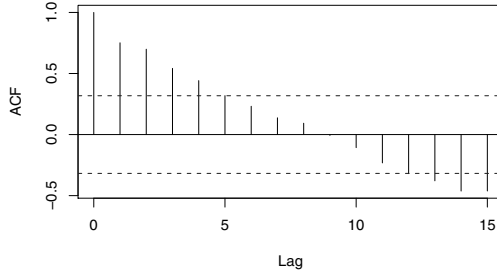


Fig. 6.3. The correlogram of residual series for the MA(1) model fitted to the exchange rate data.

6.5 Mixed models: The ARMA process

6.5.1 Definition

Recall from Chapter 4 that a series $\{x_t\}$ is an autoregressive process of order p , an AR(p) process, if

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \dots + \alpha_p x_{t-p} + w_t \quad (6.6)$$

where $\{w_t\}$ is white noise and the α_i are the model parameters. A useful class of models are obtained when AR and MA terms are added together in a single expression. A time series $\{x_t\}$ follows an autoregressive moving average (ARMA) process of order (p, q) , denoted ARMA(p, q), when

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \dots + \alpha_p x_{t-p} + w_t + \beta_1 w_{t-1} + \beta_2 w_{t-2} + \dots + \beta_q w_{t-q} \quad (6.7)$$

where $\{w_t\}$ is white noise. Equation (6.7) may be represented in terms of the backward shift operator \mathbf{B} and rearranged in the more concise polynomial form

$$\theta_p(\mathbf{B})x_t = \phi_q(\mathbf{B})w_t \quad (6.8)$$

The following points should be noted about an ARMA(p, q) process:

- The process is stationary when the roots of θ all exceed unity in absolute value.
- The process is invertible when the roots of ϕ all exceed unity in absolute value.
- The AR(p) model is the special case ARMA($p, 0$).
- The MA(q) model is the special case ARMA($0, q$).
- Parameter parsimony.* When fitting to data, an ARMA model will often be more parameter efficient (i.e., require fewer parameters) than a single MA or AR model.

- (e) *Parameter redundancy.* When θ and ϕ share a common factor, a stationary model can be simplified. For example, the model $(1 - \frac{1}{2}B)(1 - \frac{1}{3}B)x_t = (1 - \frac{1}{2}B)w_t$ can be written $(1 - \frac{1}{3}B)x_t = w_t$.

6.5.2 Derivation of second-order properties*

In order to derive the second-order properties for an ARMA(p, q) process $\{x_t\}$, it is helpful first to express the x_t in terms of white noise components w_t because white noise terms are independent. We illustrate the procedure for the ARMA(1, 1) model.

The ARMA(1, 1) process for $\{x_t\}$ is given by

$$x_t = \alpha x_{t-1} + w_t + \beta w_{t-1} \quad (6.9)$$

where w_t is white noise, with $E(w_t) = 0$ and $\text{Var}(w_t) = \sigma_w^2$. Rearranging Equation (6.9) to express x_t in terms of white noise components,

$$x_t = (1 - \alpha B)^{-1}(1 + \beta B)w_t$$

Expanding the right-hand-side,

$$\begin{aligned} x_t &= (1 + \alpha B + \alpha^2 B^2 + \dots)(1 + \beta B)w_t \\ &= \left(\sum_{i=0}^{\infty} \alpha^i B^i \right) (1 + \beta B) w_t \\ &= \left(1 + \sum_{i=0}^{\infty} \alpha^{i+1} B^{i+1} + \sum_{i=0}^{\infty} \alpha^i \beta B^{i+1} \right) w_t \\ &= w_t + (\alpha + \beta) \sum_{i=1}^{\infty} \alpha^{i-1} w_{t-i} \end{aligned} \quad (6.10)$$

With the equation in the form above, the second-order properties follow. For example, the mean $E(x_t)$ is clearly zero because $E(w_{t-i}) = 0$ for all i , and the variance is given by

$$\begin{aligned} \text{Var}(x_t) &= \text{Var} \left[w_t + (\alpha + \beta) \sum_{i=1}^{\infty} \alpha^{i-1} w_{t-i} \right] \\ &= \sigma_w^2 + \sigma_w^2 (\alpha + \beta)^2 (1 - \alpha^2)^{-1} \end{aligned} \quad (6.11)$$

The autocovariance γ_k , for $k > 0$, is given by

$$\begin{aligned} \text{Cov}(x_t, x_{t+k}) &= (\alpha + \beta) \alpha^{k-1} \sigma_w^2 + (\alpha + \beta)^2 \sigma_w^2 \alpha^k \sum_{i=1}^{\infty} \alpha^{2i-2} \\ &= (\alpha + \beta) \alpha^{k-1} \sigma_w^2 + (\alpha + \beta)^2 \sigma_w^2 \alpha^k (1 - \alpha^2)^{-1} \end{aligned} \quad (6.12)$$

The autocorrelation ρ_k then follows as

$$\begin{aligned}\rho_k &= \gamma_k / \gamma_0 = \text{Cov}(x_t, x_{t+k}) / \text{Var}(x_t) \\ &= \frac{\alpha^{k-1}(\alpha + \beta)(1 + \alpha\beta)}{1 + \alpha\beta + \beta^2}\end{aligned}\quad (6.13)$$

Note that Equation (6.13) implies $\rho_k = \alpha\rho_{k-1}$.

6.6 ARMA models: Empirical analysis

6.6.1 Simulation and fitting

The ARMA process, and the more general ARIMA processes discussed in the next chapter, can be simulated using the R function `arima.sim`, which takes a list of coefficients representing the AR and MA parameters. An ARMA(p, q) model can be fitted using the `arima` function with the `order` function parameter set to `c(p, 0, q)`. The fitting algorithm proceeds similarly to that for an MA process. Below, data from an ARMA(1, 1) process are simulated for $\alpha = -0.6$ and $\beta = 0.5$ (Equation (6.7)), and an ARMA(1, 1) model fitted to the simulated series. As expected, the sample estimates of α and β are close to the underlying model parameters.

```
> set.seed(1)
> x <- arima.sim(n = 10000, list(ar = -0.6, ma = 0.5))
> coef(arima(x, order = c(1, 0, 1)))

      ar1      ma1 intercept
-0.59697  0.50270  -0.00657
```

6.6.2 Exchange rate series

In §6.3, a simple MA(1) model failed to provide an adequate fit to the exchange rate series. In the code below, fitted MA(1), AR(1) and ARMA(1, 1) models are compared using the AIC. The ARMA(1, 1) model provides the best fit to the data, followed by the AR(1) model, with the MA(1) model providing the poorest fit. The correlogram in Figure 6.4 indicates that the residuals of the fitted ARMA(1, 1) model have small autocorrelations, which is consistent with a realisation of white noise and supports the use of the model.

```
> x.ma <- arima(x.ts, order = c(0, 0, 1))
> x.ar <- arima(x.ts, order = c(1, 0, 0))
> x.arma <- arima(x.ts, order = c(1, 0, 1))
> AIC(x.ma)

[1] -3.53

> AIC(x.ar)
```

```

[1] -37.4

> AIC(x.arma)

[1] -42.3

> x.arma

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

Coefficients:
      ar1      ma1  intercept
    0.892  0.532    2.960
s.e.  0.076  0.202    0.244

sigma^2 estimated as 0.0151:  log likelihood = 25.1,  aic = -42.3

> acf(resid(x.arma))

```

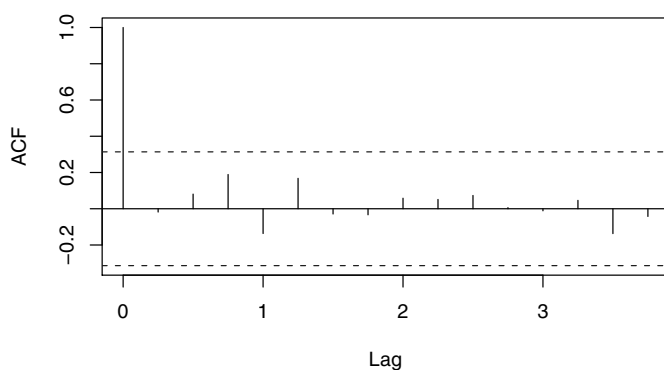


Fig. 6.4. The correlogram of residual series for the ARMA(1, 1) model fitted to the exchange rate data.

6.6.3 Electricity production series

Consider the Australian electricity production series introduced in §1.4.3. The data exhibit a clear positive trend and a regular seasonal cycle. Furthermore, the variance increases with time, which suggests a log-transformation may be appropriate (Fig. 1.5). A regression model is fitted to the logarithms of the original series in the code below.

```

> www <- "http://www.massey.ac.nz/~pscowper/ts/cbe.dat"
> CBE <- read.table(www, header = T)
> Elec.ts <- ts(CBE[, 3], start = 1958, freq = 12)
> Time <- 1:length(Elec.ts)
> Imth <- cycle(Elec.ts)
> Elec.lm <- lm(log(Elec.ts) ~ Time + I(Time^2) + factor(Imth))
> acf(resid(Elec.lm))

```

The correlogram of the residuals appears to cycle with a period of 12 months suggesting that the monthly indicator variables are not sufficient to account for the seasonality in the series (Fig. 6.5). In the next chapter, we find that this can be accounted for using a non-stationary model with a stochastic seasonal component. In the meantime, we note that the best fitting ARMA(p, q) model can be chosen using the smallest AIC either by trying a range of combinations of p and q in the `arima` function or using a `for` loop with upper bounds on p and q – taken as 2 in the code shown below. In each step of the `for` loop, the AIC of the fitted model is compared with the currently stored smallest value. If the model is found to be an improvement (i.e., has a smaller AIC value), then the new value and model are stored. To start with, `best.aic` is initialised to infinity (`Inf`). After the loop is complete, the best model can be found in `best.order`, and in this case the best model turns out to be an AR(2) model.

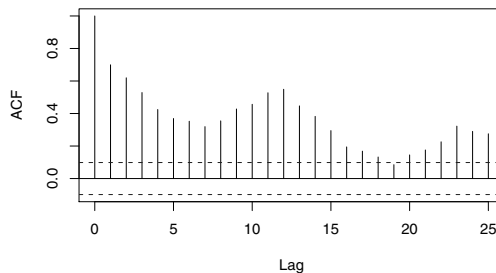


Fig. 6.5. Electricity production series: correlogram of the residual series of the fitted regression model.

```

> best.order <- c(0, 0, 0)
> best.aic <- Inf
> for (i in 0:2) for (j in 0:2) {
  fit.aic <- AIC(arima(resid(Elec.lm), order = c(i, 0,
    j)))
  if (fit.aic < best.aic) {
    best.order <- c(i, 0, j)
    best.arma <- arima(resid(Elec.lm), order = best.order)
    best.aic <- fit.aic
  }
}

```

```

}
> best.order

[1] 2 0 0

> acf(resid(best.arma))

```

The `predict` function can be used both to forecast future values from the fitted regression model and forecast the future errors associated with the regression model using the ARMA model fitted to the residuals from the regression. These two forecasts can then be summed to give a forecasted value of the logarithm for electricity production, which would then need to be anti-logged and perhaps adjusted using a bias correction factor. As `predict` is a generic R function, it works in different ways for different input objects and classes. For a fitted regression model of class `lm`, the `predict` function requires the new set of data to be in the form of a data frame (object class `data.frame`). For a fitted ARMA model of class `arma`, the `predict` function requires just the number of time steps ahead for the desired forecast. In the latter case, `predict` produces an object that has both the predicted values and their standard errors, which can be extracted using `pred` and `se`, respectively. In the code below, the electricity production for each month of the next three years is predicted.

```

> new.time <- seq(length(Elec.ts), length = 36)
> new.data <- data.frame(Time = new.time, Imth = rep(1:12,
  3))
> predict.lm <- predict(Elec.lm, new.data)
> predict.arma <- predict(best.arma, n.ahead = 36)
> elec.pred <- ts(exp(predict.lm + predict.arma$pred), start = 1991,
  freq = 12)
> ts.plot(cbind(Elec.ts, elec.pred), lty = 1:2)

```

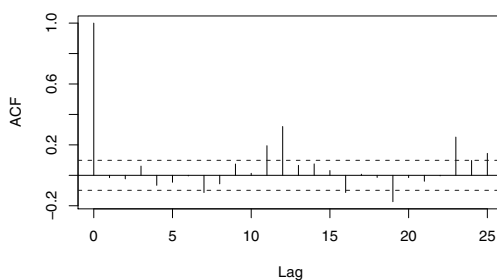


Fig. 6.6. Electricity production series: correlogram of the residual series of the best-fitting ARMA model.

The plot of the forecasted values suggests that the predicted values for winter may be underestimated by the fitted model (Fig. 6.7), which may be due to the remaining seasonal autocorrelation in the residuals (see Fig. 6.6). This problem will be addressed in the next chapter.

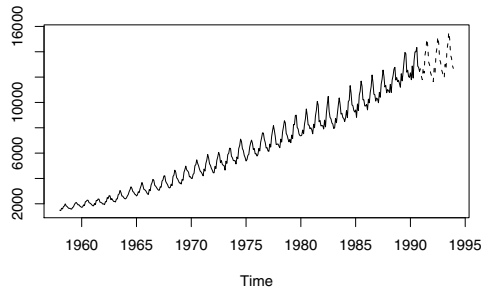


Fig. 6.7. Electricity production series: observed (solid line) and forecasted values (dotted line). The forecasted values are not likely to be accurate because of the seasonal autocorrelation present in the residuals for the fitted model.

6.6.4 Wave tank data

The data in the file `wave.dat` are the surface height of water (mm), relative to the still water level, measured using a capacitance probe positioned at the centre of a wave tank. The continuous voltage signal from this capacitance probe was sampled every 0.1 second over a 39.6-second period. The objective is to fit a suitable $\text{ARMA}(p, q)$ model that can be used to generate a realistic wave input to a mathematical model for an ocean-going tugboat in a computer simulation. The results of the computer simulation will be compared with tests using a physical model of the tugboat in the wave tank.

The `pacf` suggests that p should be at least 2 (Fig. 6.8). The best-fitting $\text{ARMA}(p, q)$ model, based on a minimum variance of residuals, was obtained with both p and q equal to 4. The `acf` and `pacf` of the residuals from this model are consistent with the residuals being a realisation of white noise (Fig. 6.9).

```
> www <- "http://www.massey.ac.nz/~pscowper/ts/wave.dat"
> wave.dat <- read.table(www, header = T)
> attach (wave.dat)
> layout(1:3)
> plot (as.ts(waveht), ylab = 'Wave height')
> acf (waveht)
> pacf (waveht)
> wave.arma <- arima(waveht, order = c(4,0,4))
> acf (wave.arma$res[-(1:4)])
> pacf (wave.arma$res[-(1:4)])
> hist(wave.arma$res[-(1:4)], xlab='height / mm', main='')
```

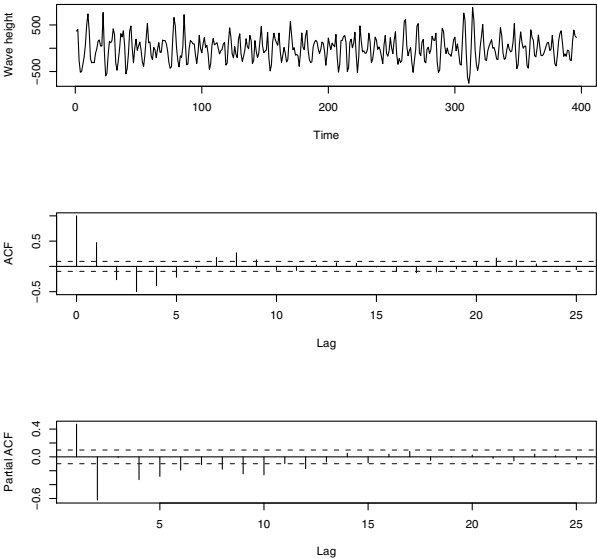


Fig. 6.8. Wave heights: time plot, acf, and pacf.

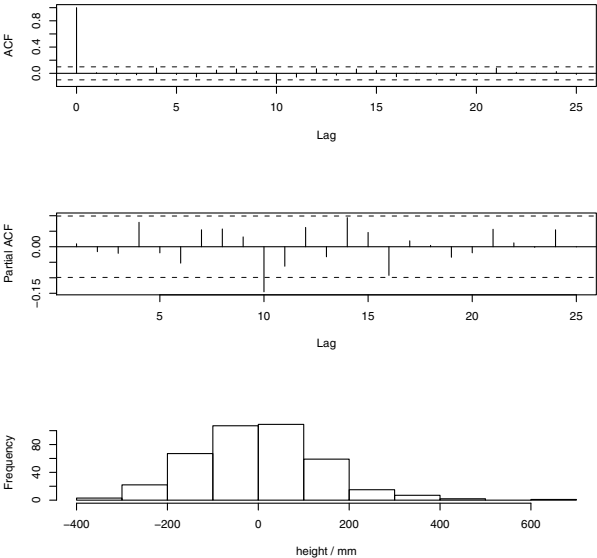


Fig. 6.9. Residuals after fitting an ARMA(4, 4) model to wave heights: acf, pacf, and histogram.

6.7 Summary of R commands

<code>arma.sim</code>	simulates data from an ARMA (or ARIMA) process
<code>arma</code>	fits an ARMA (or ARIMA) model to data
<code>seq</code>	generates a sequence
<code>expression</code>	used to plot maths symbol

6.8 Exercises

- Using the relation $\text{Cov}(\sum x_t, \sum y_t) = \sum \sum \text{Cov}(x_t, y_t)$ (Equation (2.15)) for time series $\{x_t\}$ and $\{y_t\}$, prove Equation (6.3).
- The series $\{w_t\}$ is white noise with zero mean and variance σ_w^2 . For the following moving average models, find the autocorrelation function and determine whether they are invertible. In addition, simulate 100 observations for each model in R, compare the time plots of the simulated series, and comment on how the two series might be distinguished.
 - $x_t = w_t + \frac{1}{2}w_{t-1}$
 - $x_t = w_t + 2w_{t-1}$
- Write the following models in ARMA(p, q) notation and determine whether they are stationary and/or invertible (w_t is white noise). In each case, check for parameter redundancy and ensure that the ARMA(p, q) notation is expressed in the simplest form.
 - $x_t = x_{t-1} - \frac{1}{4}x_{t-2} + w_t + \frac{1}{2}w_{t-1}$
 - $x_t = 2x_{t-1} - x_{t-2} + w_t$
 - $x_t = \frac{3}{2}x_{t-1} - \frac{1}{2}x_{t-2} + w_t - \frac{1}{2}w_{t-1} + \frac{1}{4}w_{t-2}$
 - $x_t = \frac{3}{2}x_{t-1} - \frac{1}{2}x_{t-2} + \frac{1}{2}w_t - w_{t-1}$
 - $x_t = \frac{7}{10}x_{t-1} - \frac{1}{10}x_{t-2} + w_t - \frac{3}{2}w_{t-1}$
 - $x_t = \frac{3}{2}x_{t-1} - \frac{1}{2}x_{t-2} + w_t - \frac{1}{3}w_{t-1} + \frac{1}{6}w_{t-2}$
- Fit a suitable regression model to the air passenger series. Comment on the correlogram of the residuals from the fitted regression model.
 - Fit an ARMA(p, q) model for values of p and q no greater than 2 to the residual series of the fitted regression model. Choose the best fitting model based on the AIC and comment on its correlogram.
 - Forecast the number of passengers travelling on the airline in 1961.
- Write an R function that calculates the autocorrelation function (Equation (6.13)) for an ARMA(1, 1) process. Your function should take parameters representing α and β for the AR and MA components.

- b) Plot the autocorrelation function above for the case with $\alpha = 0.7$ and $\beta = -0.5$ for lags 0 to 20.
 - c) Simulate $n = 100$ values of the ARMA(1, 1) model with $\alpha = 0.7$ and $\beta = -0.5$, and compare the sample correlogram to the theoretical correlogram plotted in part (b). Repeat for $n = 1000$.
6. Let $\{x_t : t = 1, \dots, n\}$ be a stationary time series with $E(x_t) = \mu$, $\text{Var}(x_t) = \sigma^2$, and $\text{Cor}(x_t, x_{t+k}) = \rho_k$. Using Equation (5.5) from Chapter 5:
- a) Calculate $\text{Var}(\bar{x})$ when $\{x_t\}$ is the MA(1) process $x_t = w_t + \frac{1}{2}w_{t-1}$.
 - b) Calculate $\text{Var}(\bar{x})$ when $\{x_t\}$ is the MA(1) process $x_t = w_t - \frac{1}{2}w_{t-1}$.
 - c) Compare each of the above with the variance of the sample mean obtained for the white noise case $\rho_k = 0$ ($k > 0$). Of the three models, which would have the most accurate estimate of μ based on the variances of their sample means?
 - d) A simulated example that extracts the variance of the sample mean for 100 Gaussian white noise series each of length 20 is given by


```
> set.seed(1)
> m <- rep(0, 100)
> for (i in 1:100) m[i] <- mean(rnorm(20))
> var(m)
[1] 0.0539
```

 For each of the two MA(1) processes, write R code that extracts the variance of the sample mean of 100 realisations of length 20. Compare them with the variances calculated in parts (a) and (b).

7. If the sample autocorrelation function of a time series appears to cut off after lag q (i.e., autocorrelations at lags higher than q are not significantly different from 0 and do not follow any clear patterns), then an MA(q) model might be suitable. An AR(p) model is indicated when the *partial* autocorrelation function cuts off after lag p . If there are no convincing cutoff points for either function, an ARMA model may provide the best fit. Plot the autocorrelation and partial autocorrelation functions for the simulated ARMA(1, 1) series given in §6.6.1. Using the AIC, choose a best-fitting AR model and a best-fitting MA model. Which best-fitting model (AR or MA) has the smallest number of parameters? Compare this model with the fitted ARMA(1, 1) model of §6.6.1, and comment.



<http://www.springer.com/978-0-387-88697-8>

Introductory Time Series with R
Cowpertwait, P.S.P.; Metcalfe, A.V.
2009, XVI, 256 p., Softcover
ISBN: 978-0-387-88697-8