5

Regression

5.1 Purpose

Trends in time series can be classified as *stochastic* or *deterministic*. We may consider a trend to be stochastic when it shows inexplicable changes in direction, and we attribute apparent transient trends to high serial correlation with random error. Trends of this type, which are common in financial series, can be simulated in R using models such as the random walk or autoregressive process (Chapter 4). In contrast, when we have some plausible physical explanation for a trend we will usually wish to model it in some deterministic manner. For example, a deterministic increasing trend in the data may be related to an increasing population, or a regular cycle may be related to a known seasonal frequency. Deterministic trends and seasonal variation can be modelled using regression.

The practical difference between stochastic and deterministic trends is that we extrapolate the latter when we make forecasts. We justify short-term extrapolation by claiming that underlying trends will usually change slowly in comparison with the forecast lead time. For the same reason, short-term extrapolation should be based on a line, maybe fitted to the more recent data only, rather than a high-order polynomial.

In this chapter various regression models are studied that are suitable for a time series analysis of data that contain deterministic trends and regular seasonal changes. We begin by looking at linear models for trends and then introduce regression models that account for seasonal variation using indicator and harmonic variables. Regression models can also include explanatory variables. The logarithmic transformation, which is often used to stabilise the variance, is also considered.

Time series regression usually differs from a standard regression analysis because the residuals form a time series and therefore tend to be serially correlated. When this correlation is positive, the estimated standard errors of the parameter estimates, read from the computer output of a standard regression analysis, will tend to be less than their true value. This will lead

P.S.P. Cowpertwait and A.V. Metcalfe, *Introductory Time Series with R*, Use R, DOI 10.1007/978-0-387-88698-5_5,

to erroneously high statistical significance being attributed to statistical tests in standard computer output (the p values will be smaller than they should be). Presenting correct statistical evidence is important. For example, an environmental protection group could be undermined by allegations that it is falsely claiming statistically significant trends. In this chapter, generalised least squares is used to obtain improved estimates of the standard error to account for autocorrelation in the residual series.

5.2 Linear models

5.2.1 Definition

A model for a time series $\{x_t : t = 1, \ldots n\}$ is *linear* if it can be expressed as

$$x_t = \alpha_0 + \alpha_1 u_{1,t} + \alpha_2 u_{2,t} + \ldots + \alpha_m u_{m,t} + z_t \tag{5.1}$$

where $u_{i,t}$ is the value of the ith predictor (or explanatory) variable at time t ($i = 1, \ldots, m$; $t = 1, \ldots, n$), z_t is the error at time t, and $\alpha_0, \alpha_1, \ldots, \alpha_m$ are model parameters, which can be estimated by least squares. Note that the errors form a time series $\{z_t\}$, with mean 0, that does not have to be Gaussian or white noise. An example of a linear model is the pth-order polynomial function of t:

$$x_t = \alpha_0 + \alpha_1 t + \alpha_2 t^2 \ldots + \alpha_p t^p + z_t \tag{5.2}$$

The predictor variables can be written $u_{i,t} = t^i$ ($i = 1, \ldots, p$). The term 'linear' is a reference to the summation of model parameters, each multiplied by a single predictor variable.

A simple special case of a linear model is the straight-line model obtained by putting $p = 1$ in Equation (5.2): $x_t = \alpha_0 + \alpha_1 t + z_t$. In this case, the value of the line at time t is the trend m_t. For the more general polynomial, the trend at time t is the value of the underlying polynomial evaluated at t, so in Equation (5.2) the trend is $m_t = \alpha_0 + \alpha_1 t + \alpha_2 t^2 \ldots + \alpha_p t^p$.

Many non-linear models can be transformed to linear models. For example, the model $x_t = e^{\alpha_0 + \alpha_1 t + z_t}$ for the series $\{x_t\}$ can be transformed by taking natural logarithms to obtain a linear model for the series $\{y_t\}$:

$$y_t = \log x_t = \alpha_0 + \alpha_1 t + z_t \tag{5.3}$$

In Equation (5.3), standard least squares regression could then be used to fit a linear model (i.e., estimate the parameters α_0 and α_1) and make predictions for y_t. To make predictions for x_t, the inverse transform needs to be applied to y_t, which in this example is $\exp(y_t)$. However, this usually has the effect of biasing the forecasts of mean values, and we discuss correction factors in §5.10.

Natural processes that generate time series are not expected to be precisely linear, but linear approximations are often adequate. However, we are not

restricted to linear models, and the Bass model (§3.3) is an example of a non-linear model, which we fitted using the non-linear least squares function `nls`.

5.2.2 Stationarity

Linear models for time series are non-stationary when they include functions of time. Differencing can often transform a non-stationary series with a deterministic trend to a stationary series. For example, if the time series $\{x_t\}$ is given by the straight-line function plus white noise $x_t = \alpha_0 + \alpha_1 t + z_t$, then the first-order differences are given by

$$\nabla x_t = x_t - x_{t-1} = z_t - z_{t-1} + \alpha_1 \tag{5.4}$$

Assuming the error series $\{z_t\}$ is stationary, the series $\{\nabla x_t\}$ is stationary as it is not a function of t. In §4.3.6 we found that first-order differencing can transform a non-stationary series with a stochastic trend (the random walk) to a stationary series. Thus, differencing can remove both stochastic and deterministic trends from time series. If the underlying trend is a polynomial of order m, then mth-order differencing is required to remove the trend.

Notice that differencing the straight-line function plus white noise leads to a different stationary time series than subtracting the trend. The latter gives white noise, whereas differencing gives a series of consecutive white noise terms (which is an example of an MA process, described in Chapter 6).

5.2.3 Simulation

In time series regression, it is common for the error series $\{z_t\}$ in Equation (5.1) to be autocorrelated. In the code below a time series with an increasing straight-line trend $(50 + 3t)$ with autocorrelated errors is simulated and plotted:

```
> set.seed(1)
> z <- w <- rnorm(100, sd = 20)
> for (t in 2:100) z[t] <- 0.8 * z[t - 1] + w[t]
> Time <- 1:100
> x <- 50 + 3 * Time + z
> plot(x, xlab = "time", type = "l")
```

The model for the code above can be expressed as $x_t = 50 + 3t + z_t$, where $\{z_t\}$ is the AR(1) process $z_t = 0.8z_{t-1} + w_t$ and $\{w_t\}$ is Gaussian white noise with $\sigma = 20$. A time plot of a realisation of $\{x_t\}$ is given in Figure 5.1.

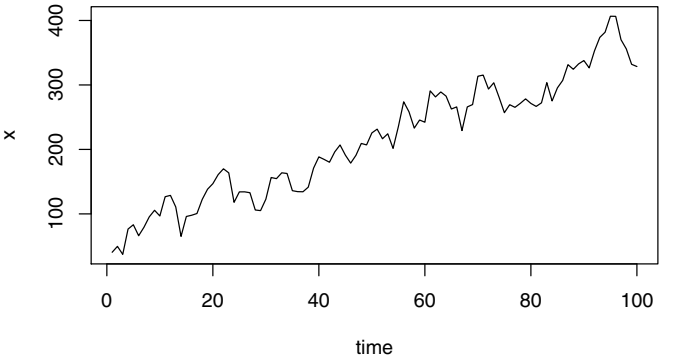

Fig. 5.1. Time plot of a simulated time series with a straight-line trend and AR(1) residual errors.

5.3 Fitted models

5.3.1 Model fitted to simulated data

Linear models are usually fitted by minimising the sum of squared errors, $\sum z_t^2 = \sum(x_t - \alpha_0 - \alpha_1 u_{1,t} - \ldots - \alpha_m u_{m,t})^2$, which is achieved in `R` using the function `lm`:

```
> x.lm <- lm(x ~ Time)
> coef(x.lm)

(Intercept)        Time
      58.55        3.06

> sqrt(diag(vcov(x.lm)))

(Intercept)        Time
     4.8801      0.0839
```

In the code above, the estimated parameters of the linear model are extracted using `coef`. Note that, as expected, the estimates are close to the underlying parameter values of 50 for the intercept and 3 for the slope. The standard errors are extracted using the square root of the diagonal elements obtained from `vcov`, although these standard errors are likely to be underestimated because of autocorrelation in the residuals. The function `summary` can also be used to obtain this information but tends to give additional information, for example t-tests, which may be incorrect for a time series regression analysis due to autocorrelation in the residuals.

After fitting a regression model, we should consider various diagnostic plots. In the case of time series regression, an important diagnostic plot is the correlogram of the residuals:

```
> acf(resid(x.lm))
> pacf(resid(x.lm))
```

As expected, the residual time series is autocorrelated (Fig. 5.2). In Figure 5.3, only the lag 1 partial autocorrelation is significant, which suggests that the residual series follows an AR(1) process. Again this should be as expected, given that an AR(1) process was used to simulate these residuals.

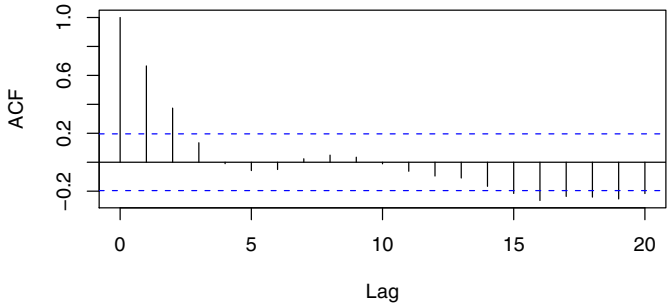

Fig. 5.2. Residual correlogram for the fitted straight-line model.

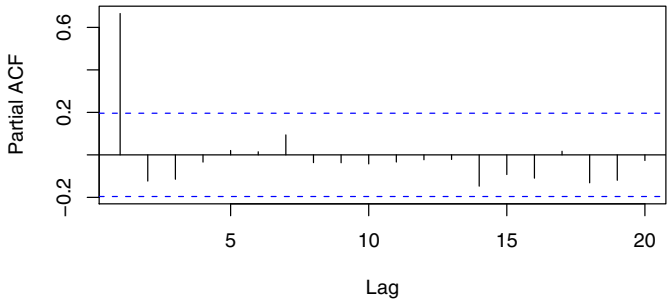

Fig. 5.3. Residual partial correlogram for the fitted straight-line model.

5.3.2 Model fitted to the temperature series (1970–2005)

In §1.4.5, we extracted temperatures for the period 1970–2005. The following regression model is fitted to the global temperature over this period,

and approximate 95% confidence intervals are given for the parameters using `confint`. The explanatory variable is the time, so the function `time` is used to extract the 'times' from the `ts` temperature object.

```
> www <- "http://www.massey.ac.nz/~pscowper/ts/global.dat"
> Global <- scan(www)
> Global.ts <- ts(Global, st = c(1856, 1), end = c(2005,
    12), fr = 12)
> temp <- window(Global.ts, start = 1970)
> temp.lm <- lm(temp ~ time(temp))
> coef(temp.lm)

(Intercept)  time(temp)
   -34.9204      0.0177

> confint(temp.lm)

                2.5 %   97.5 %
(Intercept) -37.2100 -32.6308
time(temp)    0.0165   0.0188

> acf(resid(lm(temp ~ time(temp))))
```

The confidence interval for the slope does not contain zero, which would provide statistical evidence of an increasing trend in global temperatures if the autocorrelation in the residuals is negligible. However, the residual series is positively autocorrelated at shorter lags (Fig. 5.4), leading to an underestimate of the standard error and too narrow a confidence interval for the slope.

Intuitively, the positive correlation between consecutive values reduces the effective record length because similar values will tend to occur together. The following section illustrates the reasoning behind this but may be omitted, without loss of continuity, by readers who do not require the mathematical details.

5.3.3 Autocorrelation and the estimation of sample statistics*

To illustrate the effect of autocorrelation in estimation, the sample mean will be used, as it is straightforward to analyse and is used in the calculation of other statistical properties.

Suppose $\{x_t : t = 1, \ldots, n\}$ is a time series of *independent* random variables with mean $E(x_t) = \mu$ and variance $\mathrm{Var}(x_t) = \sigma^2$. Then it is well known in the study of random samples that the sample mean $\bar{x} = \sum_{t=1}^{n} x_t/n$ has mean $E(\bar{x}) = \mu$ and variance $\mathrm{Var}(\bar{x}) = \sigma^2/n$ (or standard error $\sigma/\sqrt{n}$). Now let $\{x_t : t = 1, \ldots, n\}$ be a stationary time series with $E(x_t) = \mu$, $\mathrm{Var}(x_t) = \sigma^2$, and autocorrelation function $\mathrm{Cor}(x_t, x_{t+k}) = \rho_k$. Then the variance of the sample mean is given by

$$\mathrm{Var}\,(\bar{x}) = \frac{\sigma^2}{n}\left[1 + 2\sum_{k=1}^{n-1}(1 - k/n)\rho_k\right] \tag{5.5}$$

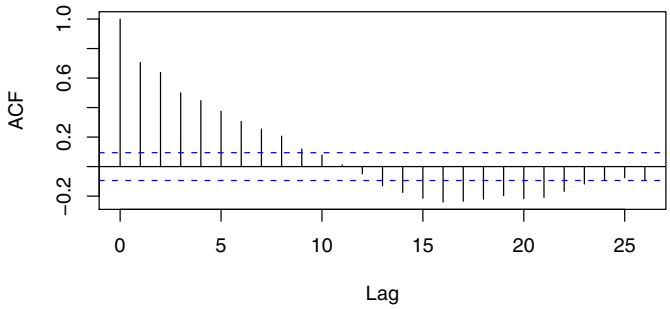

Fig. 5.4. Residual correlogram for the regression model fitted to the global temperature series (1970–2005).

In Equation (5.5) the variance σ^2/n for an independent random sample arises as the special case where $\rho_k = 0$ for all $k > 0$. If $\rho_k > 0$, then $\mathrm{Var}(\bar{x}) > \sigma^2/n$ and the resulting estimate of μ is less accurate than that obtained from a random (independent) sample of the same size. Conversely, if $\rho_k < 0$, then the variance of the estimate may actually be smaller than the variance obtained from a random sample of the same size. This latter result is due to the tendency for a value above the mean to be followed by a value below the mean, thus providing a more efficient estimate of the overall mean level. Conversely, for a positive correlation, values are more likely to persist above or below the mean, resulting in a less efficient estimate of the overall mean. Thus, for a positively correlated series, a larger sample would be needed to achieve the same level of accuracy in the estimate of μ obtained from a sample of negatively (or zero) correlated series. Equation (5.5) can be proved using Equation (2.15) and the properties of variance:

$$
\begin{aligned}
\mathrm{Var}\,(\bar{x}) &= \mathrm{Var}\,[(x_1 + x_2 + \cdots + x_n)/n] = \mathrm{Var}\,(x_1 + x_2 + \cdots + x_n)\,/n^2 \\
&= n^{-2}\mathrm{Cov}\big(\textstyle\sum_{i=1}^{n} x_i, \sum_{j=1}^{n} x_j\big) = n^{-2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathrm{Cov}\,(x_i, x_j) \\
&= n^{-2}\big[\gamma_0 + \gamma_1 + \cdots + \gamma_{n-2} + \gamma_{n-1} + \\
&\qquad \gamma_1 + \gamma_0 + \cdots + \gamma_{n-3} + \gamma_{n-2} + \\
&\qquad \vdots \qquad\qquad\qquad\qquad \vdots \\
&\qquad \gamma_{n-2} + \gamma_{n-3} + \cdots + \gamma_2 + \gamma_1 + \\
&\qquad \gamma_{n-1} + \gamma_{n-2} + \cdots + \gamma_1 + \gamma_0\big] \\
&= n^{-2}\big[n\gamma_0 + 2\textstyle\sum_{k=1}^{n-1}(n-k)\,\gamma_k\big]
\end{aligned}
$$

Equation (5.5) follows after substituting $\gamma_0 = \sigma^2$ and $\rho_k = \gamma_k/\sigma^2$ in the last line above.

5.4 Generalised least squares

We have seen that in time series regression it is common and expected that the residual series will be autocorrelated. For a positive serial correlation in the residual series, this implies that the standard errors of the estimated regression parameters are likely to be underestimated (Equation (5.5)), and should therefore be corrected.

A fitting procedure known as *generalised least squares* (GLS) can be used to provide better estimates of the standard errors of the regression parameters to account for the autocorrelation in the residual series. The procedure is essentially based on maximising the likelihood given the autocorrelation in the data and is implemented in `R` in the `gls` function (within the `nlme` library, which you will need to load).

5.4.1 GLS fit to simulated series

The following example illustrates how to fit a regression model to the simulated series of §5.2.3 using generalised least squares:

```
> library(nlme)
> x.gls <- gls(x ~ Time, cor = corAR1(0.8))
> coef(x.gls)

(Intercept)        Time
      58.23        3.04

> sqrt(diag(vcov(x.gls)))

(Intercept)        Time
     11.925       0.202
```

A lag 1 autocorrelation of 0.8 is used above because this value was used to simulate the data (§5.2.3). For historical series, the lag 1 autocorrelation would need to be estimated from the correlogram of the residuals of a fitted linear model; i.e., a linear model should first be fitted by ordinary least squares (OLS) and the lag 1 autocorrelation read off from a correlogram plot of the residuals of the fitted model.

In the example above, the standard errors of the parameters are considerably greater than those obtained from OLS using `lm` (§5.3) and are more accurate as they take the autocorrelation into account. The parameter estimates from GLS will generally be slightly different from those obtained with OLS, because of the weighting. For example, the slope is estimated as 3.06 using `lm` but 3.04 using `gls`. In principle, the GLS estimators are preferable because they have smaller standard errors.

5.4.2 Confidence interval for the trend in the temperature series

To calculate an approximate 95% confidence interval for the trend in the global temperature series (1970–2005), GLS is used to estimate the standard error accounting for the autocorrelation in the residual series (Fig. 5.4). In the `gls` function, the residual series is approximated as an AR(1) process with a lag 1 autocorrelation of 0.7 read from Figure 5.4, which is used as a parameter in the `gls` function:

```
> temp.gls <- gls(temp ~ time(temp), cor = corAR1(0.7))
> confint(temp.gls)

                2.5 %   97.5 %
(Intercept) -39.8057 -28.4966
time(temp)    0.0144   0.0201
```

Although the confidence intervals above are now wider than they were in §5.3, zero is not contained in the intervals, which implies that the estimates are statistically significant, and, in particular, that the trend is significant. Thus, there is statistical evidence of an increasing trend in global temperatures over the period 1970–2005, so that, if current conditions persist, temperatures may be expected to continue to rise in the future.

5.5 Linear models with seasonal variables

5.5.1 Introduction

As time series are observations measured sequentially in time, seasonal effects are often present in the data, especially annual cycles caused directly or indirectly by the Earth's movement around the Sun. Seasonal effects have already been observed in several of the series we have looked at, including the airline series (§1.4.1), the temperature series (§1.4.5), and the electricity production series (§1.4.3). In this section, linear regression models with predictor variables for seasonal effects are considered.

5.5.2 Additive seasonal indicator variables

Suppose a time series contains s seasons. For example, with time series measured over each calendar month, $s = 12$, whereas for series measured over six-month intervals, corresponding to summer and winter, $s = 2$. A seasonal indicator model for a time series $\{x_t : t = 1, \ldots, n\}$ containing s seasons and a trend m_t is given by

$$x_t = m_t + s_t + z_t \tag{5.6}$$

where $s_t = \beta_i$ when t falls in the ith season ($t = 1, \ldots, n; i = 1, \ldots, s$) and $\{z_t\}$ is the residual error series, which may be autocorrelated. This model

takes the same form as the additive decomposition model (Equation (1.2)) but differs in that the trend is formulated with parameters. In Equation (5.6), m_t does not have a constant term (referred to as the intercept), i.e., m_t could be a polynomial of order p with parameters $\alpha_1, \ldots, \alpha_p$. Equation (5.6) is then equivalent to a polynomial trend in which the constant term depends on the season, so that the s seasonal parameters $(\beta_1, \ldots, \beta_s)$ correspond to s possible constant terms in Equation (5.2). Equation (5.6) can therefore be written as

$$x_t = m_t + \beta_{1+(t-1) \bmod s} + z_t \tag{5.7}$$

For example, with a time series $\{x_t\}$ observed for each calendar month beginning with $t = 1$ at January, a seasonal indicator model with a straight-line trend is given by

$$x_t = \alpha_1 t + s_t + z_t = \begin{cases} \alpha_1 t + \beta_1 + z_t & t = 1, 13, \ldots \\ \alpha_1 t + \beta_2 + z_t & t = 2, 14, \ldots \\ \vdots & \\ \alpha_1 t + \beta_{12} + z_t & t = 12, 24, \ldots \end{cases} \tag{5.8}$$

The parameters for the model in Equation (5.8) can be estimated by OLS or GLS by treating the seasonal term s_t as a 'factor'. In R, the `factor` function can be applied to seasonal indices extracted using the function `cycle` (§1.4.1).

5.5.3 Example: Seasonal model for the temperature series

The parameters of a straight-line trend with additive seasonal indices can be estimated for the temperature series (1970–2005) as follows:

```
> Seas <- cycle(temp)
> Time <- time(temp)
> temp.lm <- lm(temp ~ 0 + Time + factor(Seas))
> coef(temp.lm)

          Time  factor(Seas)1  factor(Seas)2  factor(Seas)3
        0.0177       -34.9973       -34.9880       -35.0100
 factor(Seas)4  factor(Seas)5  factor(Seas)6  factor(Seas)7
      -35.0123       -35.0337       -35.0251       -35.0269
 factor(Seas)8  factor(Seas)9 factor(Seas)10 factor(Seas)11
      -35.0248       -35.0383       -35.0525       -35.0656
factor(Seas)12
      -35.0487
```

A zero is used within the formula to ensure that the model does not have an intercept. If the intercept is included in the formula, one of the seasonal terms will be dropped and an estimate for the intercept will appear in the output. However, the fitted models, with or without an intercept, would be equivalent, as can be easily verified by rerunning the algorithm above without the zero in

the formula. The parameters can also be estimated by GLS by replacing `lm` with `gls` in the code above.

Using the above fitted model, a two-year-ahead future prediction for the temperature series is obtained as follows:

```
> new.t <- seq(2006, len = 2 * 12, by = 1/12)
> alpha <- coef(temp.lm)[1]
> beta <- rep(coef(temp.lm)[2:13], 2)
> (alpha * new.t + beta)[1:4]

factor(Seas)1 factor(Seas)2 factor(Seas)3 factor(Seas)4
        0.524         0.535         0.514         0.514
```

Alternatively, the `predict` function can be used to make forecasts provided the new data are correctly labelled within a `data.frame`:

```
> new.dat <- data.frame(Time = new.t, Seas = rep(1:12, 2))
> predict(temp.lm, new.dat)[1:24]

    1     2     3     4     5     6     7     8     9    10    11    12
0.524 0.535 0.514 0.514 0.494 0.504 0.503 0.507 0.495 0.482 0.471 0.489
   13    14    15    16    17    18    19    20    21    22    23    24
0.542 0.553 0.532 0.531 0.511 0.521 0.521 0.525 0.513 0.500 0.488 0.507
```

5.6 Harmonic seasonal models

In the previous section, one parameter estimate is used per season. However, seasonal effects often vary smoothly over the seasons, so that it may be more parameter-efficient to use a smooth function instead of separate indices.

Sine and cosine functions can be used to build smooth variation into a seasonal model. A sine wave with frequency f (cycles per sampling interval), amplitude A, and phase shift ϕ can be expressed as

$$A\sin(2\pi ft + \phi) = \alpha_s \sin(2\pi ft) + \alpha_c \cos(2\pi ft) \tag{5.9}$$

where $\alpha_s = A\cos(\phi)$ and $\alpha_c = A\sin(\phi)$. The expression on the right-hand side of Equation (5.9) is linear in the parameters α_s and α_c, whilst the left-hand side is non-linear because the parameter ϕ is within the sine function. Hence, the expression on the right-hand side is preferred in the formulation of a seasonal regression model, so that OLS can be used to estimate the parameters. For a time series $\{x_t\}$ with s seasons there are $[s/2]$ possible cycles.[1] The harmonic seasonal model is defined by

[1] The notation [ ] represents the integer part of the expression within. In most practical cases, s is even and so [ ] can be omitted. However, for some 'seasons', s may be an odd number, making the notation necessary. For example, if the 'seasons' are the days of the week, there would be $[7/2] = 3$ possible cycles.

$$x_t = m_t + \sum_{i=1}^{[s/2]} \left\{ s_i \sin(2\pi it/s) + c_i \cos(2\pi it/s) \right\} + z_t \tag{5.10}$$

where m_t is the trend which includes a parameter for the constant term, and s_i and c_i are unknown parameters. The trend may take a polynomial form as in Equation (5.2). When s is an even number, the value of the sine at frequency $1/2$ (when $i = s/2$ in the summation term shown in Equation (5.10)) will be zero for all values of t, and so the term can be left out of the model. Hence, with a constant term included, the maximum number of parameters in the harmonic model equals that of the seasonal indicator variable model (Equation (5.6)), and the fits will be identical.

At first sight it may seem strange that the harmonic model has cycles of a frequency higher than the seasonal frequency of $1/s$. However, the addition of further harmonics has the effect of perturbing the underlying wave to make it less regular than a standard sine wave of period s. This usually still gives a dominant seasonal pattern of period s, but with a more realistic underlying shape. For example, suppose data are taken at monthly intervals. Then the second plot given below might be a more realistic underlying seasonal pattern than the first plot, as it perturbs the standard sine wave by adding another two harmonic terms of frequencies $2/12$ and $4/12$ (Fig. 5.5):

```
> TIME <- seq(1, 12, len = 1000)
> plot(TIME, sin(2 * pi * TIME/12), type = "l")
> plot(TIME, sin(2 * pi * TIME/12) + 0.2 * sin(2 * pi * 2 *
      TIME/12) + 0.1 * sin(2 * pi * 4 * TIME/12) + 0.1 *
      cos(2 * pi * 4 * TIME/12), type = "l")
```

The code above illustrates just one of many possible combinations of harmonics that could be used to model a wide range of possible underlying seasonal patterns.

5.6.1 Simulation

It is straightforward to simulate a series based on the harmonic model given by Equation (5.10). For example, suppose the underlying model is

$$\begin{aligned} x_t = {} & 0.1 + 0.005t + 0.001t^2 + \sin(2\pi t/12) + \\ & 0.2\sin(4\pi t/12) + 0.1\sin(8\pi t/12) + 0.1\cos(8\pi t/12) + w_t \end{aligned} \tag{5.11}$$

where $\{w_t\}$ is Gaussian white noise with standard deviation 0.5. This model has the same seasonal harmonic components as the model represented in Figure 5.5b but also contains an underlying quadratic trend. Using the code below, a series of length 10 years is simulated, and it is shown in Figure 5.6.

```
> set.seed(1)
> TIME <- 1:(10 * 12)
> w <- rnorm(10 * 12, sd = 0.5)
```

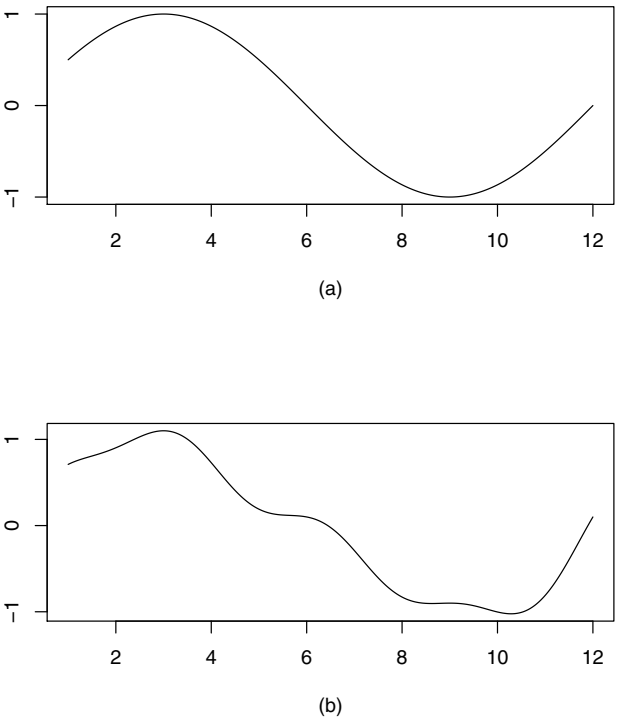

Fig. 5.5. Two possible underlying seasonal patterns for monthly series based on the harmonic model (Equation (5.10)). Plot (a) is of the first harmonic over a year and is usually too regular for most practical applications. Plot (b) is of the same wave but with a further two harmonics added. Plot (b) illustrates just one of many ways that an underlying sine wave can be perturbed to produce a less regular, but still dominant, seasonal pattern of period 12 months.

```
> Trend <- 0.1 + 0.005 * TIME + 0.001 * TIME^2
> Seasonal <- sin(2*pi*TIME/12) + 0.2*sin(2*pi*2*TIME/12) +
              0.1*sin(2*pi*4*TIME/12) + 0.1*cos(2*pi*4*TIME/12)
> x <- Trend + Seasonal + w
> plot(x, type = "l")
```

5.6.2 Fit to simulated series

With reference to Equation (5.10), it would seem reasonable to place the harmonic variables in matrices, which can be achieved as follows:

```
> SIN <- COS <- matrix(nr = length(TIME), nc = 6)
> for (i in 1:6) {
```

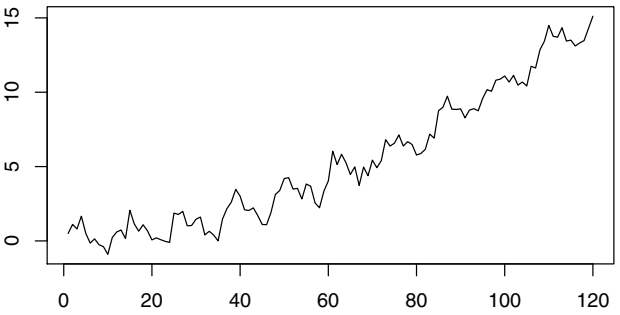

Fig. 5.6. Ten years of simulated data for the model given by Equation (5.11).

```
    COS[, i] <- cos(2 * pi * i * TIME/12)
    SIN[, i] <- sin(2 * pi * i * TIME/12)
}
```

In most cases, the order of the harmonics and polynomial trend will be unknown. However, the harmonic coefficients are known to be independent, which means that all harmonic coefficients that are not statistically significant can be dropped. It is largely a subjective decision on the part of the statistician to decide what constitutes a significant variable. An approximate t-ratio of magnitude 2 is a common choice and corresponds to an approximate 5% significance level. This t-ratio can be obtained by dividing the estimated coefficient by the standard error of the estimate. The following example illustrates the procedure applied to the simulated series of the last section:

```
> x.lm1 <- lm(x ~ TIME + I(TIME^2) + COS[, 1] + SIN[, 1] +
   COS[, 2] + SIN[, 2] + COS[, 3]  + SIN[, 3] + COS[, 4] +
   SIN[, 4] + COS[, 5] + SIN[, 5]  + COS[, 6] + SIN[, 6])
> coef(x.lm1)/sqrt(diag(vcov(x.lm1)))

(Intercept)       TIME  I(TIME^2)   COS[, 1]   SIN[, 1]   COS[, 2]
      1.239      1.125     25.933      0.328     15.442     -0.515
   SIN[, 2]   COS[, 3]   SIN[, 3]   COS[, 4]   SIN[, 4]   COS[, 5]
      3.447      0.232     -0.703      0.228      1.053     -1.150
   SIN[, 5]   COS[, 6]   SIN[, 6]
      0.857     -0.310      0.382
```

The preceding output has three significant coefficients. These are used in the following model:[2]

[2] Some statisticians choose to include both the `COS` and `SIN` terms for a particular frequency if either has a statistically significant value.

```
> x.lm2 <- lm(x ~ I(TIME^2) + SIN[, 1] + SIN[, 2])
> coef(x.lm2)/sqrt(diag(vcov(x.lm2)))

(Intercept)   I(TIME^2)    SIN[, 1]    SIN[, 2]
       4.63      111.14       15.79        3.49
```

As can be seen in the output from the last command, the coefficients are all significant. The estimated coefficients of the best-fitting model are given by

```
> coef(x.lm2)

(Intercept)   I(TIME^2)    SIN[, 1]    SIN[, 2]
    0.28040     0.00104     0.90021     0.19886
```

The coefficients above give the following model for predictions at time t:

$$\hat{x}_t = 0.280 + 0.00104t^2 + 0.900\sin(2\pi t/12) + 0.199\sin(4\pi t/12) \qquad (5.12)$$

The AIC can be used to compare the two fitted models:

```
> AIC(x.lm1)

[1] 165

> AIC(x.lm2)

[1] 150
```

As expected, the last model has the smallest AIC and therefore provides the best fit to the data. Due to sampling variation, the best-fitting model is not identical to the model used to simulate the data, as can easily be verified by taking the AIC of the known underlying model:

```
> AIC(lm(x ~ TIME +I(TIME^2) +SIN[,1] +SIN[,2] +SIN[,4] +COS[,4]))

[1] 153
```

In R, the algorithm `step` can be used to automate the selection of the best-fitting model by the AIC. For the example above, the appropriate command is `step(x.lm1)`, which contains all the predictor variables in the form of the first model. Try running this command, and check that the final output agrees with the model selected above.

A best fit can equally well be based on choosing the model that leads to the smallest estimated standard deviations of the errors, provided the degrees of freedom are taken into account.

5.6.3 Harmonic model fitted to temperature series (1970–2005)

In the code below, a harmonic model with a quadratic trend is fitted to the temperature series (1970–2005) from §5.3.2. The units for the 'time' variable are in 'years', so the divisor of 12 is not needed when creating the harmonic variables. To reduce computation error in the OLS procedure due to large numbers, the `TIME` variable is standardized after the `COS` and `SIN` predictors have been calculated.

```
> SIN <- COS <- matrix(nr = length(temp), nc = 6)
> for (i in 1:6) {
     COS[, i] <- cos(2 * pi * i * time(temp))
     SIN[, i] <- sin(2 * pi * i * time(temp))
 }
> TIME <- (time(temp) - mean(time(temp)))/sd(time(temp))
> mean(time(temp))

[1] 1988

> sd(time(temp))

[1] 10.4

> temp.lm1 <- lm(temp ~ TIME + I(TIME^2) +
                        COS[,1] + SIN[,1] + COS[,2] + SIN[,2] +
                        COS[,3] + SIN[,3] + COS[,4] + SIN[,4] +
                        COS[,5] + SIN[,5] + COS[,6] + SIN[,6])
> coef(temp.lm1)/sqrt(diag(vcov(temp.lm1)))

(Intercept)        TIME   I(TIME^2)    COS[, 1]    SIN[, 1]    COS[, 2]
     18.245      30.271       1.281       0.747       2.383       1.260
   SIN[, 2]    COS[, 3]    SIN[, 3]    COS[, 4]    SIN[, 4]    COS[, 5]
      1.919       0.640       0.391       0.551       0.168       0.324
   SIN[, 5]    COS[, 6]    SIN[, 6]
      0.345      -0.409      -0.457

> temp.lm2 <- lm(temp ~ TIME + SIN[, 1] + SIN[, 2])
> coef(temp.lm2)

(Intercept)        TIME    SIN[, 1]    SIN[, 2]
     0.1750      0.1841      0.0204      0.0162

> AIC(temp.lm)

[1] -547

> AIC(temp.lm1)

[1] -545

> AIC(temp.lm2)

[1] -561
```

Again, the AIC is used to compare the fitted models, and only statistically significant terms are included in the final model.

To check the adequacy of the fitted model, it is appropriate to create a time plot and correlogram of the residuals because the residuals form a time series (Fig. 5.7). The time plot is used to detect patterns in the series. For example, if a higher-ordered polynomial is required, this would show up as a curve in the time plot. The purpose of the correlogram is to determine whether there is autocorrelation in the series, which would require a further model.

```
> plot(time(temp), resid(temp.lm2), type = "l")
> abline(0, 0, col = "red")
> acf(resid(temp.lm2))
> pacf(resid(temp.lm2))
```

In Figure 5.7(a), there is no discernible curve in the series, which implies that a straight line is an adequate description of the trend. A tendency for the series to persist above or below the x-axis implies that the series is positively autocorrelated. This is verified in the correlogram of the residuals, which shows a clear positive autocorrelation at lags 1–10 (Fig. 5.7b).

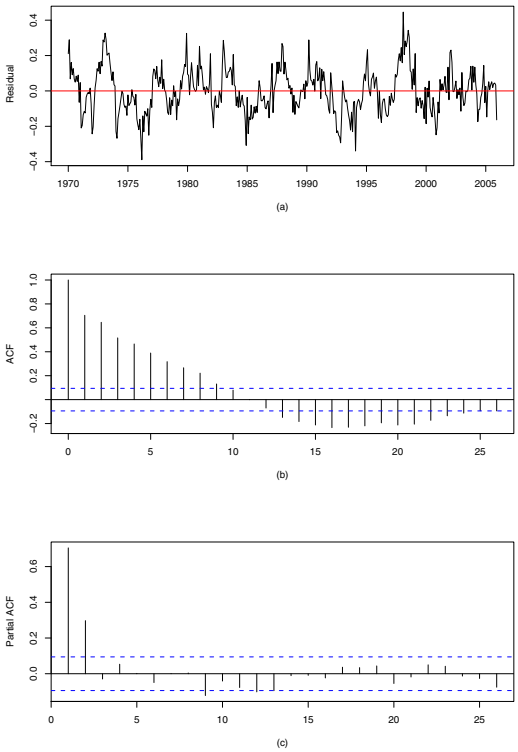

Fig. 5.7. Residual diagnostic plots for the harmonic model fitted to the temperature series (1970–2005): (a) the residuals plotted against time; (b) the correlogram of the residuals (time units are months); (c) partial autocorrelations plotted against lag (in months).

The correlogram in Figure 5.7 is similar to that expected of an AR(p) process (§4.5.5). This is verified by the plot of the partial autocorrelations, in which only the lag 1 and lag 2 autocorrelations are statistically significant (Fig. 5.7). In the code below, an AR(2) model is fitted to the residual series:

```
> res.ar <- ar(resid(temp.lm2), method = "mle")
> res.ar$ar

[1] 0.494 0.307

> sd(res.ar$res[-(1:2)])

[1] 0.0837

> acf(res.ar$res[-(1:2)])
```

The correlogram of the residuals of the fitted AR(2) model is given in Figure 5.8, from which it is clear that the residuals are approximately white noise. Hence, the final form of the model provides a good fit to the data. The fitted model for the monthly temperature series can be written as

$$x_t = 0.175 + \frac{0.184(t - 1988)}{10.4} + 0.0204\sin(2\pi t) + 0.0162\sin(4\pi t) + z_t \quad (5.13)$$

where t is 'time' measured in units of 'years', the residual series $\{z_t\}$ follow an AR(2) process given by

$$z_t = 0.494z_{t-1} + 0.307z_{t-2} + w_t \quad (5.14)$$

and $\{w_t\}$ is white noise with mean zero and standard deviation 0.0837.

If we require an accurate assessment of the standard error, we should refit the model using `gls`, allowing for an AR(2) structure for the errors (Exercise 6).

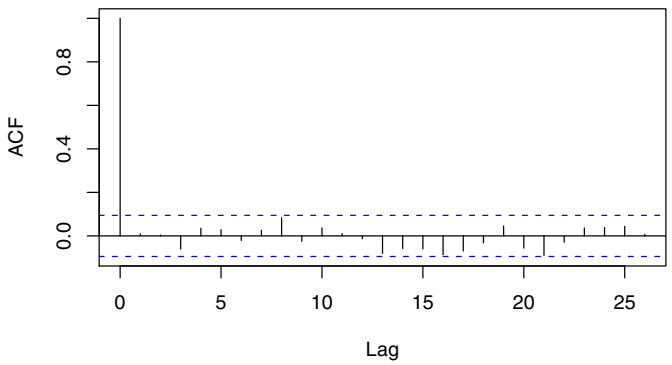

Fig. 5.8. Correlogram of the residuals of the AR(2) model fitted to the residuals of the harmonic model for the temperature series.

5.7 Logarithmic transformations

5.7.1 Introduction

Recall from §5.2 that the natural logarithm (base e) can be used to transform a model with multiplicative components to a model with additive components. For example, if $\{x_t\}$ is a time series given by

$$x_t = m'_t \; s'_t \; z'_t \tag{5.15}$$

where m'_t is the trend, s'_t is the seasonal effect, and z'_t is the residual error, then the series $\{y_t\}$, given by

$$y_t = \log x_t = \log m'_t + \log s'_t + \log z'_t = m_t + s_t + z_t \tag{5.16}$$

has additive components, so that if m_t and s_t are also linear functions, the parameters in Equation (5.16) can be estimated by OLS. In Equation (5.16), logs can be taken only if the series $\{x_t\}$ takes all positive values; i.e., $x_t > 0$ for all t. Conversely, a log-transformation may be seen as an appropriate model formulation when a series can only take positive values and has values near zero because the anti-log forces the predicted and simulated values for $\{x_t\}$ to be positive.

5.7.2 Example using the air passenger series

Consider the air passenger series from §1.4.1. Time plots of the original series and the natural logarithm of the series can be obtained using the code below and are shown in Figure 5.9.

```
> data(AirPassengers)
> AP <- AirPassengers
> plot(AP)
> plot(log(AP))
```

In Figure 5.9(a), the variance can be seen to increase as t increases, whilst after the logarithm is taken the variance is approximately constant over the period of the record (Fig. 5.9b). Therefore, as the number of people using the airline can also only be positive, the logarithm would be appropriate in the model formulation for this time series. In the following code, a harmonic model with polynomial trend is fitted to the air passenger series. The function `time` is used to extract the time and create a standardised time variable `TIME`.

```
> SIN <- COS <- matrix(nr = length(AP), nc = 6)
> for (i in 1:6) {
      SIN[, i] <- sin(2 * pi * i * time(AP))
      COS[, i] <- cos(2 * pi * i * time(AP))
  }
> TIME <- (time(AP) - mean(time(AP)))/sd(time(AP))
> mean(time(AP))
```

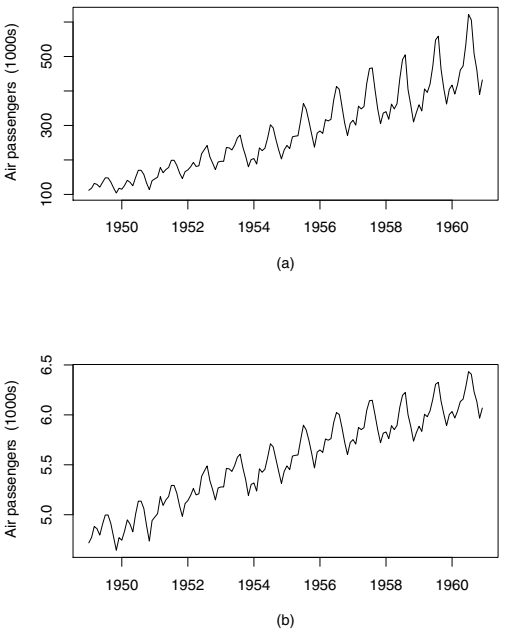

Fig. 5.9. Time plots of (a) the airline series (1949–1960) and (b) the natural logarithm of the airline series.

```
[1] 1955

> sd(time(AP))

[1] 3.48

> AP.lm1 <- lm(log(AP) ~ TIME + I(TIME^2) + I(TIME^3) + I(TIME^4) +
        SIN[,1] + COS[,1] + SIN[,2] + COS[,2] + SIN[,3] + COS[,3] +
        SIN[,4] + COS[,4] + SIN[,5] + COS[,5] + SIN[,6] + COS[,6])
> coef(AP.lm1)/sqrt(diag(vcov(AP.lm1)))

(Intercept)       TIME  I(TIME^2)  I(TIME^3)  I(TIME^4)   SIN[, 1]
    744.685     42.382     -4.162     -0.751      1.873      4.868
   COS[, 1]   SIN[, 2]   COS[, 2]   SIN[, 3]   COS[, 3]   SIN[, 4]
    -26.055     10.395     10.004     -4.844     -1.560     -5.666
   COS[, 4]   SIN[, 5]   COS[, 5]   SIN[, 6]   COS[, 6]
      1.946     -3.766      1.026      0.150     -0.521

> AP.lm2 <- lm(log(AP) ~ TIME + I(TIME^2) + SIN[,1] + COS[,1] +
    SIN[,2] + COS[,2] + SIN[,3] + SIN[,4] + COS[,4] + SIN[,5])
> coef(AP.lm2)/sqrt(diag(vcov(AP.lm2)))
```

```
(Intercept)        TIME   I(TIME^2)    SIN[, 1]    COS[, 1]    SIN[, 2]
     922.63      103.52       -8.24        4.92      -25.81       10.36
   COS[, 2]    SIN[, 3]    SIN[, 4]    COS[, 4]    SIN[, 5]
       9.96       -4.79       -5.61        1.95       -3.73

> AIC(AP.lm1)

[1] -448

> AIC(AP.lm2)

[1] -451

> acf(resid(AP.lm2))
```

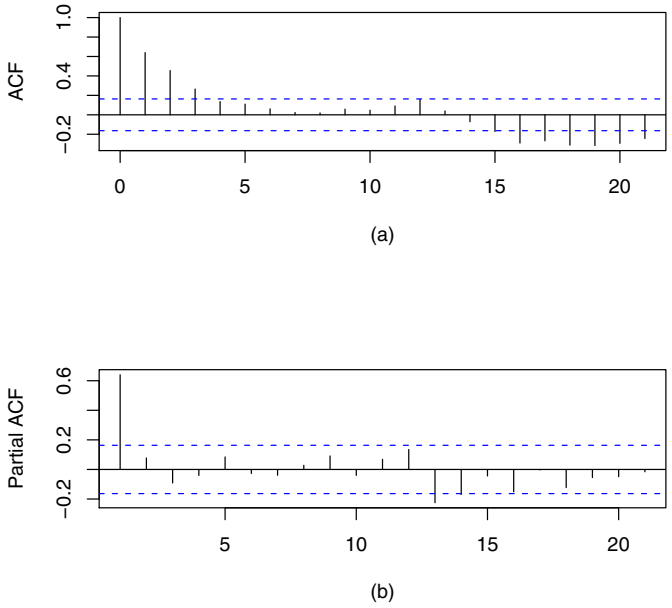

Fig. 5.10. The correlogram (a) and partial autocorrelations (b) of the residual series.

The residual correlogram indicates that the data are positively autocorrelated (Fig. 5.10). As mentioned in §5.4, the standard errors of the parameter estimates are likely to be under-estimated if there is positive serial correlation in the data. This implies that predictor variables may falsely appear 'significant' in the fitted model. In the code below, GLS is used to check the significance of the variables in the fitted model, using the lag 1 autocorrelation (approximately 0.6) from Figure 5.10.

```
> AP.gls <- gls(log(AP) ~ TIME + I(TIME^2) + SIN[,1] + COS[,1] +
    SIN[,2] + COS[,2] + SIN[,3] + SIN[,4] + COS[,4] + SIN[,5],
    cor = corAR1(0.6))
> coef(AP.gls)/sqrt(diag(vcov(AP.gls)))

(Intercept)        TIME   I(TIME^2)    SIN[, 1]    COS[, 1]    SIN[, 2]
     398.84       45.85       -3.65        3.30      -18.18       11.77
  COS[, 2]    SIN[, 3]    SIN[, 4]    COS[, 4]    SIN[, 5]
     11.43       -7.63      -10.75        3.57       -7.92
```

In Figure 5.10(b), the partial autocorrelation plot suggests that the residual series follows an AR(1) process, which is fitted to the series below:

```
> AP.ar <- ar(resid(AP.lm2), order = 1, method = "mle")
> AP.ar$ar

[1] 0.641

> acf(AP.ar$res[-1])
```

The correlogram of the residuals of the fitted AR(1) model might be taken for white noise given that only one autocorrelation is significant (Fig. 5.11). However, the lag of this significant value corresponds to the seasonal lag (12) in the original series, which implies that the fitted model has failed to fully account for the seasonal variation in the data. Understandably, the reader might regard this as curious, given that the data were fitted using the full seasonal harmonic model. However, seasonal effects can be stochastic just as trends can, and the harmonic model we have used is deterministic. In Chapter 7, models with stochastic seasonal terms will be considered.

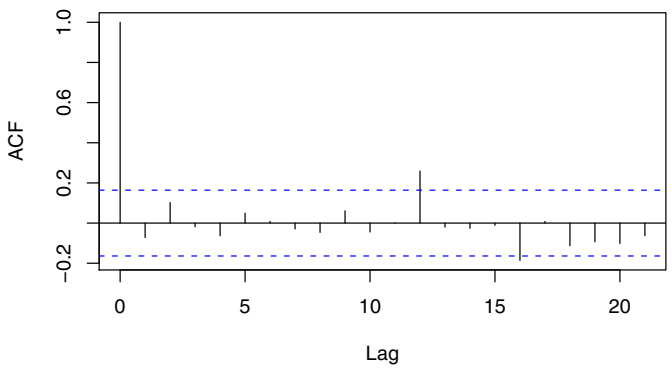

Fig. 5.11. Correlogram of the residuals from the AR(1) model fitted to the residuals of the logarithm model.

5.8 Non-linear models

5.8.1 Introduction

For the reasons given in §5.2, linear models are applicable to a wide range of time series. However, for some time series it may be more appropriate to fit a non-linear model directly rather than take logs or use a linear polynomial approximation. For example, if a series is known to derive from a known non-linear process, perhaps based on an underlying known deterministic law in science, then it would be better to use this information in the model formulation and fit a non-linear model directly to the data. In `R`, a non-linear model can be fitted by least squares using the function `nls`.

In the previous section, we found that using the natural logarithm of a series could help stabilise the variance. However, using logs can present difficulties when a series contains negative values, because the log of a negative value is undefined. One way around this problem is to add a constant to all the terms in the series, so if $\{x_t\}$ is a series containing (some) negative values, then adding c_0 such that $c_0 > \max\{-x_t\}$ and then taking logs produces a transformed series $\{\log(c_0 + x_t)\}$ that is defined for all t. A linear model (e.g., a straight-line trend) could then be fitted to produce for $\{x_t\}$ the model

$$x_t = -c_0 + e^{\alpha_0 + \alpha_1 t + z_t} \tag{5.17}$$

where α_0 and α_1 are model parameters and $\{z_t\}$ is a residual series that may be autocorrelated.

The main difficulty with the approach leading to Equation (5.17) is that c_0 should really be estimated like any other parameter in the model, whilst in practice a user will often arbitrarily choose a value that satisfies the constraint ($c_0 > \max\{-x_t\}$). If there is a reason to expect a model similar to that in Equation (5.17) but there is no evidence for multiplicative residual terms, then the constant c_0 should be estimated with the other model parameters using non-linear least squares; i.e., the following model should be fitted:

$$x_t = -c_0 + e^{\alpha_0 + \alpha_1 t} + z_t \tag{5.18}$$

5.8.2 Example of a simulated and fitted non-linear series

As non-linear models are generally fitted when the underlying non-linear function is known, we will simulate a non-linear series based on Equation (5.18) with $c_0 = 0$ and compare parameters estimated using `nls` with those of the known underlying function.

Below, a non-linear series with AR(1) residuals is simulated and plotted (Fig. 5.12):

```
> set.seed(1)
> w <- rnorm(100, sd = 10)
```

```
> z <- rep(0, 100)
> for (t in 2:100) z[t] <- 0.7 * z[t - 1] + w[t]
> Time <- 1:100
> f <- function(x) exp(1 + 0.05 * x)
> x <- f(Time) + z
> plot(x, type = "l")
> abline(0, 0)
```

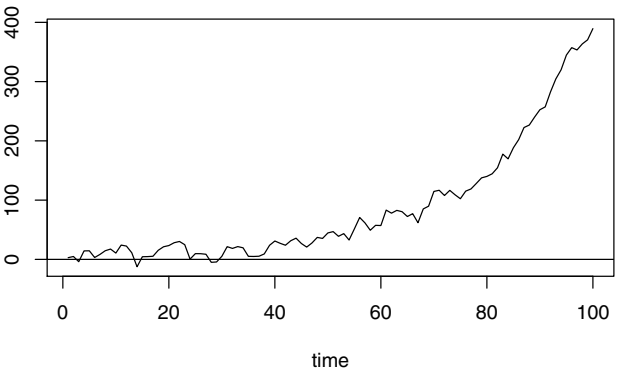

Fig. 5.12. Plot of a non-linear series containing negative values.

The series plotted in Figure 5.12 has an apparent increasing exponential trend but also contains negative values, so that a direct log-transformation cannot be used and a non-linear model is needed. In `R`, a non-linear model is fitted by specifying a formula with the parameters and their starting values contained in a `list`:

```
> x.nls <- nls(x ~ exp(alp0 + alp1 * Time), start = list(alp0 = 0.1,
      alp1 = 0.5))
> summary(x.nls)$parameters

       Estimate   Std. Error   t value   Pr(>|t|)
alp0     1.1764     0.074295      15.8   9.20e-29
alp1     0.0483     0.000819      59.0   2.35e-78
```

The estimates for α_0 and α_1 are close to the underlying values that were used to simulate the data, although the standard errors of these estimates are likely to be underestimated because of the autocorrelation in the residuals.[3]

[3] The generalised least squares function `gls` can be used to fit non-linear models with autocorrelated residuals. However, in practice, computational difficulties often arise when using this function with non-linear models.

5.9 Forecasting from regression

5.9.1 Introduction

A forecast is a prediction into the future. In the context of time series regression, a forecast involves extrapolating a fitted model into the future by evaluating the model function for a new series of times. The main problem with this approach is that the trends present in the fitted series may change in the future. Therefore, it is better to think of a forecast from a regression model as an expected value conditional on past trends continuing into the future.

5.9.2 Prediction in R

The generic function for making predictions in R is `predict`. The function essentially takes a fitted model and new data as parameters. The key to using this function with a regression model is to ensure that the new data are properly defined and labelled in a `data.frame`.

In the code below, we use this function in the fitted regression model of §5.7.2 to forecast the number of air passengers travelling for the 10-year period that follows the record (Fig. 5.13). The forecast is given by applying the exponential function (anti-log) to `predict` because the regression model was fitted to the logarithm of the series:

```
> new.t <- time(ts(start = 1961, end = c(1970, 12), fr = 12))
> TIME <- (new.t - mean(time(AP)))/sd(time(AP))
> SIN <- COS <- matrix(nr = length(new.t), nc = 6)
> for (i in 1:6) {
      COS[, i] <- cos(2 * pi * i * new.t)
      SIN[, i] <- sin(2 * pi * i * new.t)
  }
> SIN <- SIN[, -6]
> new.dat <- data.frame(TIME = as.vector(TIME), SIN = SIN,
      COS = COS)
> AP.pred.ts <- exp(ts(predict(AP.lm2, new.dat), st = 1961,
      fr = 12))
> ts.plot(log(AP), log(AP.pred.ts), lty = 1:2)
> ts.plot(AP, AP.pred.ts, lty = 1:2)
```

5.10 Inverse transform and bias correction

5.10.1 Log-normal residual errors

The forecasts in Figure 5.13(b) were obtained by applying the anti-log to the forecasted values obtained from the log-regression model. However, the process

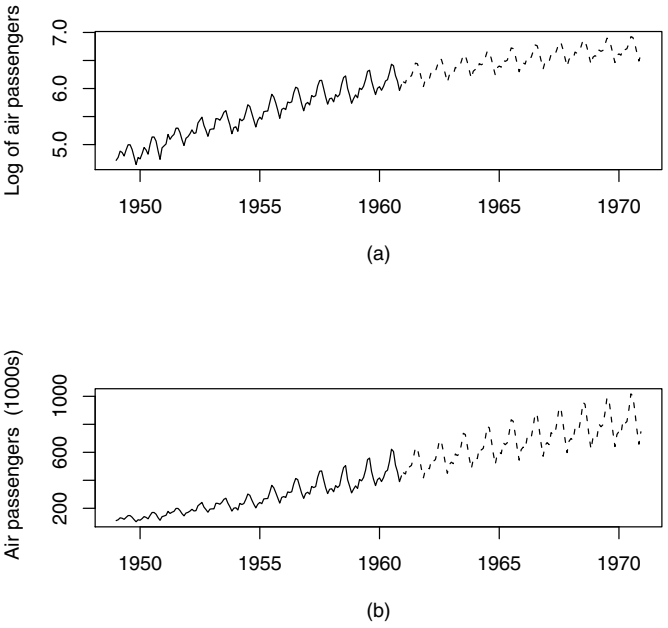

Fig. 5.13. Air passengers (1949–1960; solid line) and forecasts (1961–1970; dotted lines): (a) logarithm and forecasted values; (b) original series and anti-log of the forecasted values.

of using a transformation, such as the logarithm, and then applying an inverse transformation introduces a bias in the forecasts of the mean values. If the regression model closely fits the data, this bias will be small (as shown in the next example for the airline predictions). Note that a bias correction is only for means and should not be used in simulations.

The bias in the means arises as a result of applying the inverse transform to a residual series. For example, if the time series are Gaussian white noise $\{w_t\}$, with mean zero and standard deviation σ, then the distribution of the inverse-transform (the anti-log) of the series is log-normal with mean $e^{\frac{1}{2}\sigma^2}$. This can be verified theoretically, or empirically by simulation as in the code below:

```
> set.seed(1)
> sigma <- 1
> w <- rnorm(1e+06, sd = sigma)
> mean(w)

[1] 4.69e-05
```

```
> mean(exp(w))

[1] 1.65

> exp(sigma^2/2)

[1] 1.65
```

The code above indicates that the mean of the anti-log of the Gaussian white noise and the expected mean from a log-normal distribution are equal. Hence, for a Gaussian white noise residual series, a correction factor of $e^{\frac{1}{2}\sigma^2}$ should be applied to the forecasts of means. The importance of this correction factor really depends on the value of σ^2. If σ^2 is very small, the correction factor will hardly change the forecasts at all and so could be neglected without major concern, especially as errors from other sources are likely to be significantly greater.

5.10.2 Empirical correction factor for forecasting means

The $e^{\frac{1}{2}\sigma^2}$ correction factor can be used when the residual series of the fitted log-regression model is Gaussian white noise. In general, however, the distribution of the residuals from the log regression (Exercise 5) is often negatively skewed, in which case a correction factor can be determined empirically using the mean of the anti-log of the residual series. In this approach, adjusted forecasts $\{\hat{x}'_t\}$ can be obtained from

$$\hat{x}'_t = e^{\hat{\log x_t}} \sum_{t=1}^{n} e^{z_t}/n \tag{5.19}$$

where $\{\hat{\log x_t} : t = 1, \ldots, n\}$ is the predicted series given by the fitted log-regression model, and $\{z_t\}$ is the residual series from this fitted model.

The following example illustrates the procedure for calculating the correction factors.

5.10.3 Example using the air passenger data

For the airline series, the forecasts can be adjusted by multiplying the predictions by $e^{\frac{1}{2}\sigma^2}$, where σ is the standard deviation of the residuals, or using an empirical correction factor as follows:

```
> summary(AP.lm2)$r.sq

[1] 0.989

> sigma <- summary(AP.lm2)$sigma
> lognorm.correction.factor <- exp((1/2) * sigma^2)
> empirical.correction.factor <- mean(exp(resid(AP.lm2)))
```

```
> lognorm.correction.factor

[1] 1.001171

> empirical.correction.factor

[1] 1.001080

> AP.pred.ts <- AP.pred.ts * empirical.correction.factor
```

The adjusted forecasts in `AP.pred.ts` allow for the bias in taking the anti-log of the predictions. However, the small σ (and $R^2 = 0.99$) results in a small correction factor (of the order 0.1%), which is probably negligible compared with other sources of errors that exist in the forecasts. Whilst in this example the correction factor is small, there is no reason why it will be small in general.

5.11 Summary of R commands

`lm`	fits a linear (regression) model
`coef`	extracts the parameter estimates from a fitted model
`confint`	returns a (95%) confidence interval for the parameters of a fitted model
`gls`	fits a linear model using generalised least squares (allowing for autocorrelated residuals)
`factor`	returns variables in the form of 'factors' or indicator variables

5.12 Exercises

1. a) Produce a time plot for $\{x_t : t = 1, \ldots, 100\}$, where $x_t = 70 + 2t - 3t^2 + z_t$, $\{z_t\}$ is the AR(1) process $z_t = 0.5z_{t-1} + w_t$, and $\{w_t\}$ is white noise with standard deviation 25.
   b) Fit a quadratic trend to the series $\{x_t\}$. Give the coefficients of the fitted model.
   c) Find a 95% confidence interval for the parameters of the quadratic model, and comment.
   d) Plot the correlogram of the residuals and comment.
   e) Refit the model using GLS. Give the standard errors of the parameter estimates, and comment.

2. The standard errors of the parameter estimates of a fitted regression model are likely to be underestimated if there is positive serial correlation in the data. This implies that explanatory variables may appear as 'significant' when they should not. Use GLS to check the significance of the variables

of the fitted model from §5.6.3. Use an appropriate estimate of the lag 1 autocorrelation within `gls`.

3. This question is based on the electricity production series (1958–1990).
   a) Give two reasons why a log-transformation may be appropriate for the electricity series.
   b) Fit a seasonal indicator model with a quadratic trend to the (natural) logarithm of the series. Use stepwise regression to select the best model based on the AIC.
   c) Fit a harmonic model with a quadratic trend to the logarithm of the series. Use stepwise regression to select the best model based on the AIC.
   d) Plot the correlogram and partial correlogram of the residuals from the overall best-fitting model and comment on the plots.
   e) Fit an AR model to the residuals of the best-fitting model. Give the order of the best-fitting AR model and the estimated model parameters.
   f) Plot the correlogram of the residuals of the AR model, and comment.
   g) Write down in full the equation of the best-fitting model.
   h) Use the best fitting model to forecast electricity production for the years 1991–2000, making sure you have corrected for any bias due to taking logs.

4. Suppose a sample of size n follows an AR(1) process with lag 1 autocorrelation $\rho_1 = \alpha$. Use Equation (5.5) to find the variance of the sample mean.

5. A hydrologist wishes to simulate monthly inflows to the Font Reservoir over the next 10-year period. Use the data in `Font.dat` (§2.3.3) to answer the following:
   a) Regress `inflow` on `month` using indicator variables and time t, and fit a suitable AR model to the residual error series.
   b) Plot a histogram of the residual errors of the fitted AR model, and comment on the plot. Fit back-to-back Weibull distributions to the errors.
   c) Simulate 20 realisations of `inflow` for the next 10 years.
   d) Give reasons why a log transformation may be suitable for the series of inflows.
   e) Regress `log(inflow)` on `month` using indicator variables and time t (as above), and fit a suitable AR model to the residual error series.
   f) Plot a histogram of the residual errors of the fitted AR model, and comment on the plot. Fit a back-to-back Weibull distribution to the residual errors.

g) Simulate 20 realisations of `log(inflow)` for the next 10-years. Take anti-logs of the simulated values to produce a series of simulated flows.
h) Compare both sets of simulated flows, and discuss which is the more satisfactory.

6. Refit the harmonic model to the temperature series using `gls`, allowing for errors from an AR(2) process.
   a) Construct a 99% confidence interval for the coefficient of time.
   b) Plot the residual error series from the model fitted using GLS against the residual error series from the model fitted using OLS.
   c) Refit the AR(2) model to the residuals from the fitted (GLS) model.
   d) How different are the fitted models?
   e) Calculate the annual means. Use OLS to regress the annual mean temperature on time, and construct a 99% confidence interval for its coefficient.