Week 2: Time Series Regression

- Shumway, R.H. & Stoffer, D.S. (2016). Time series analysis and its applications with R examples. springer.
  - Chapter 2: Time Series Regression and Exploratory Data Analysis
- Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.
  - Chapter 7: Time series regression models
  - Chapter 8: Exponential smoothing

Chapter 2

Time series regression models

In this chapter we discuss time series regression models. The basic concept is that we forecast the time series of interest x_t assuming that it has a linear relationship with other time series z_t. For example, we might wish to forecast monthly sales x_t using total advertising spend z_t as a predictor. Or we might forecast daily electricity demand x_t using temperature z_{t1} and the day of week z_{t2} as predictors. The forecast variable x_t is sometimes also called the regressand, dependent or explained variable. The predictor variables z are sometimes also called the regressors, independent or explanatory variables.

2.1 Classical Regression in the Time Series Context

Consider some output or dependent time series, x_t, for $t = 1, \dots, n$, is being influenced by a collection of possible inputs or independent series, say, $z_{t1}, z_{t2}, \dots, z_{tq},$. We express this relation through the linear regression model

$$x_t = \beta_0 + \beta_1 z_{t1} + \beta_2 z_{t2} + \dots + \beta_q z_{tq} + w_t, \tag{2.1.1}$$

where $\beta_0, \beta_1, \dots, \beta_q$ are unknown fixed regression coefficients, and $\{w_t\}$ is a random error or noise process consisting of iid normal variables with $\mu_w = 0$ and variance σ_w^2. For time series regression, it is rarely the case that the noise is white, and we will need to eventually relax that assumption.

Example 2.1 *[Estimating linear trend] Consider the monthly price (per pound) of a chicken in the US from mid-2001 to mid-2016 (180 months), say x_t. As can be seen in Figure 2.1 there is an obvious upward trend in the series, and we might use simple linear regression to estimate that trend by fitting the model*

$$x_t = \beta_0 + \beta_1 z_t + w_t, \quad z_t = 2001\frac{7}{12}, 2001\frac{8}{12}, \dots, 2016\frac{6}{12}. \tag{2.1.2}$$

This is in the form of the regression model (2.1.1) *with $q = 1$ and z_t denotes time.*

```
library(astsa)
library(itsmr)
summary(fit <- lm(chicken˜time(chicken))) # regress price on time
```

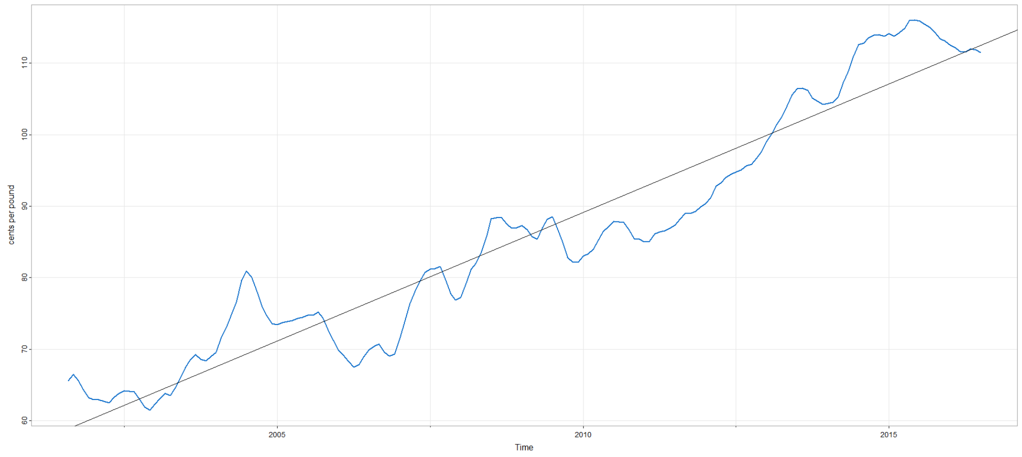

Figure 2.1: The price of chicken: monthly whole bird spot price, Georgia docks, US cents per pound, August 2001 to July 2016, with fitted linear trend line

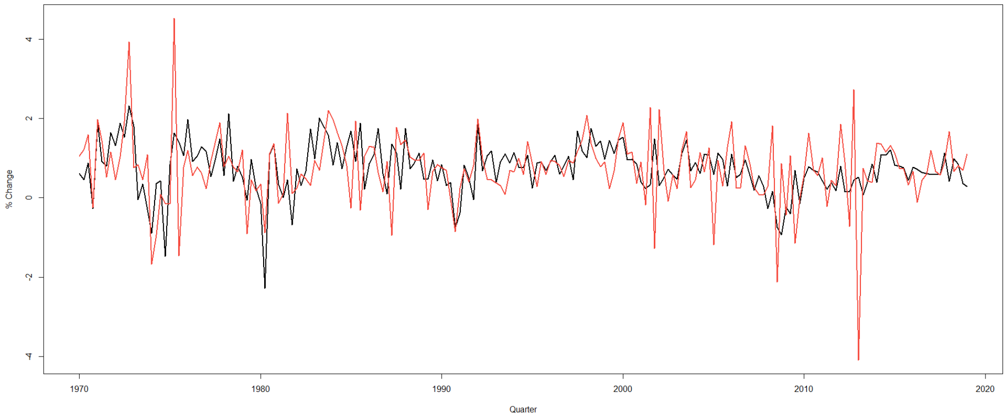

Figure 2.2: Percentage changes in personal consumption expenditure and personal income for the US.

```r
tsplot(chicken, ylab="cents per pound", col=4, lwd=2)
abline(fit)
```

Listing 2.1: Example 2.1

Note 2.1 *Note that we are making the assumption that the errors, w_t, are an iid normal sequence, which may not be true.*

Example 2.2 (US consumption expenditure (simple linear regression)) *Figure 2.2 shows time series of quarterly percentage changes (growth rates) of real personal consumption expenditure, x_t, and real personal disposable income, z_t, for the US from 1970 Q1 to 2019 Q2. We want to find the linear relationship between x_t and z_t.:*

$$x_t = \beta_0 + \beta_1 z_t + w_t. \tag{2.1.3}$$

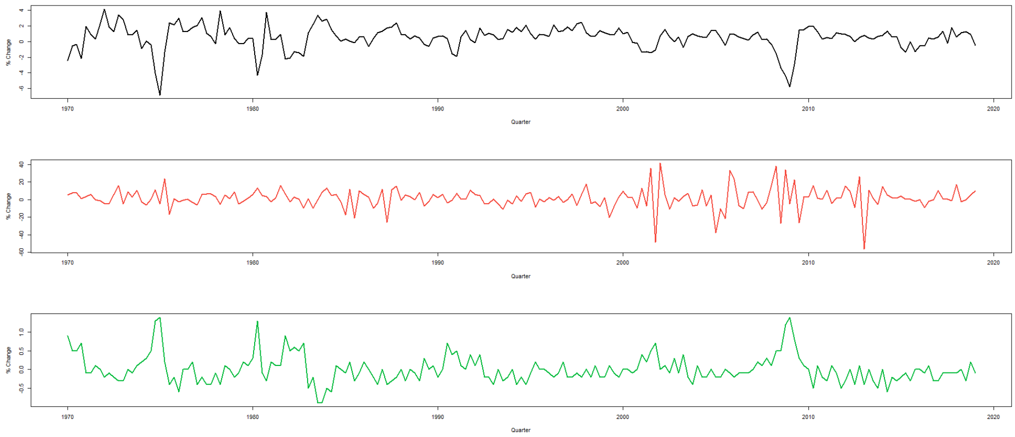

Figure 2.3: Percentage changes in personal consumption expenditure and personal income for the US.

```
library(fpp3)
attach(us_change)
class(Consumption)
Consumption_ts=ts(Consumption, start=1970, end=2019, frequency = 4)
class(Consumption_ts)
class(Income)
Income_ts=ts(Income, start=1970, end=2019, frequency = 4)
class(Income_ts)
ts.plot(cbind(Consumption_ts,Income_ts), col=1:2, ylab="% Change", xlab
    ="Quarter")
```

Listing 2.2: Example 2.2

Example 2.3 (US consumption expenditure (multiple linear regression)) *There are additional predictors that may be useful for forecasting US consumption expenditure. These are quarterly percentage changes in industrial production and personal savings, and quarterly changes in the unemployment rate (as this is already a percentage). Building a multiple linear regression model can potentially generate more accurate forecasts as we expect consumption expenditure to not only depend on personal income but on other predictors as well. Therefore, we try to find the linear relationship between x_t and Income (z_{t1}), Production (z_{t2}), Savings (z_{t3}) and Unemployment (z_{t4}).*

$$x_t = \beta_0 + \beta_1 z_{t1} + \beta_1 z_{t2} + \beta_1 z_{t3} + \beta_1 z_{t4} + w_t. \tag{2.1.4}$$

```
par(mfrow=c(3,1))

Production_ts = ts(Production, start=1970, end=2019, frequency = 4)
Savings_ts = ts(Savings, start=1970, end=2019, frequency = 4)
Unemployment_ts = ts(Unemployment, start=1970, end=2019, frequency = 4)

ts.plot(Production_ts , col=1, lwd=2, ylab="% Change", xlab="Quarter")
ts.plot(Savings_ts , col=2, lwd=2, ylab="% Change", xlab="Quarter")
ts.plot(Unemployment_ts, col=3, lwd=2, ylab="% Change", xlab="Quarter")
```

Listing 2.3: Example 2.3

2.1.1 Assumptions

When we use a linear regression model, we are implicitly making some assumptions about the input variables:

- We assume that the model is a reasonable approximation to reality; that is, the relationship between the forecast variable and the predictor variables satisfies this linear equation.
- We make the following assumptions about the error terms, $w_1, w_2, \ldots, w_n$:
  - they have mean zero; otherwise the forecasts will be systematically biased.
  - they are not autocorrelated; otherwise the forecasts will be inefficient, as there is more information in the data that can be exploited.
  - they are unrelated to the predictor variables; otherwise there would be more information that should be included in the systematic part of the model.
  - The errors are normally distributed with a constant variance σ_w^2 (we need this assumption to produce prediction intervals).

2.1.2 Least Square Estimation

In practice, of course, we have a collection of observations but we do not know the values of the coefficients $\beta_0, \beta_1, \ldots, \beta_q$. These need to be estimated from the data.

The **least squares** principle provides a way of choosing the coefficients effectively by minimising the sum of the squared errors. That is, we choose the values of $\beta_0, \beta_1, \ldots, \beta_q$ that minimise

$$Q = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n} (x_t - \beta_0 - \beta_1 z_{t1} - \beta_2 z_{t2} - \ldots - \beta_q z_{tq})^2. \tag{2.1.5}$$

To minimize Equation (2.1.5), lets rewrite as:

$$Q = \sum_{t=1}^{n} w_t^2 = \sum_{t=1}^{n} (x_t - \beta' z_t)^2, \tag{2.1.6}$$

where $z_t = (1, z_{t1}, z_{t2}, \ldots, z_{tq})', \beta = (\beta_0, \beta_1, \ldots, \beta_q)'$ and where $'$ denotes transpose.

This minimization can be accomplished by differentiating (2.1.6) with respect to the vector β or by using the properties of projections. Either way, the solution must satisfy $\sum_{t=1}^{n} (x_t - \beta' z_t) z_t' = 0$. This procedure gives the normal equations

$$\left(\sum_{t=1}^{n} z_t z_t' \right) \beta = \sum_{t=1}^{n} z_t x_t. \tag{2.1.7}$$

If $\sum_{t=1}^{n} z_t z_t'$ is non-singular, the least squares estimate of β is

$$\hat{\beta} = \left(\sum_{t=1}^{n} z_t z_t' \right)^{-1} \sum_{t=1}^{n} z_t x_t. \tag{2.1.8}$$

The minimized error sum of squares (2.1.6), denoted SSE, can be written as

$$SSE = \sum_{t=1}^{n} (x_t - \hat{\beta}' z_t)^2, \tag{2.1.9}$$

The ordinary least squares estimators are unbiased, i.e., $E(\hat{\beta}) = \beta$, and have the smallest variance within the class of linear unbiased estimators. If the errors w_t are normally distributed, $\hat{\beta}$ is also the maximum likelihood estimator for β and is normally distributed with

$$Cov(\hat{\beta}) = \sigma_w^2 C, \tag{2.1.10}$$

where

$$C = \left(\sum_{t=1}^{n} z_t z_t' \right)^{-1}. \tag{2.1.11}$$

An unbiased estimator for the variance σ_w^2 is

$$s_w^2 = MSE = \frac{SSE}{n - (q+1)}, \tag{2.1.12}$$

where MSE denotes the mean squared error. Under the normal assumption,

$$t = \frac{(\hat{\beta}_i - \beta_i)}{s_w \sqrt{c_{ii}}}, \tag{2.1.13}$$

has the t-distributionwith $n - (q + 1)$ degrees of freedom; c_{ii} denotes the i-th diagonal element of C, as defined in (2.1.11). This result is often used for individual tests of the null hypothesis $H_0 : \beta_i = 0$ for $i = 1, \ldots, q$.

Note 2.2 *It can be shown that the least square estimators for β_0 and β_1 in simple linear regression models, such as* (2.1.2) *and* (2.1.3) *are*

$$\hat{\beta}_0 = \bar{x} - \hat{\beta}_1 \bar{z}, \qquad \hat{\beta}_1 = \frac{\sum_{t=1}^{n} (x_t - \bar{x})(z_t - \bar{z})}{\sum_{t=1}^{n} (z_t - \bar{z})^2} \tag{2.1.14}$$

Example 2.4 (Cont. Example 2.1) *Using* (2.1.14)*, it can be shown that the least square estimators for β_0 and β_1 in Example 2.1 are*

$$\hat{\beta}_0 = -7131.02, \qquad \hat{\beta}_1 = 3.59.$$

```
summary(fit <- lm(chicken~time(chicken), na.action=NULL))
# Call:
#lm(formula = chicken ~ time(chicken), na.action = NULL)

#Coefficients:
#                 Estimate Std. Error t value Pr(>|t|)
#(Intercept)    -7.131e+03  1.624e+02  -43.91   <2e-16 ***
#time(chicken)   3.592e+00  8.084e-02   44.43   <2e-16 ***

#Residual standard error: 4.696 on 178 degrees of freedom
#Multiple R-squared:  0.9173, Adjusted R-squared:  0.9168
#F-statistic:  1974 on 1 and 178 DF,  p-value: < 2.2e-16
```

Listing 2.4: Cont. Example 2.1

Example 2.5 (Cont. Example 2.2) *Based on* (2.1.14)*, the least square estimators for* β_0 *and* β_1 *in Example 2.2 can be calculated as*

$$\hat{\beta}_0 = 0.54, \qquad \hat{\beta}_1 = 0.27.$$

```
summary(fit1 <- lm(Consumption ~ Income, na.action=NULL))
#Call:
#lm(formula = Consumption ~ Income, na.action = NULL)

#Coefficients:
#             Estimate Std. Error t value Pr(>|t|)
#(Intercept)  0.54454    0.05403   10.079  < 2e-16 ***
#Income       0.27183    0.04673    5.817  2.4e-08 ***

#Residual standard error: 0.5905 on 196 degrees of freedom
#Multiple R-squared:  0.1472, Adjusted R-squared:  0.1429
#F-statistic: 33.84 on 1 and 196 DF,  p-value: 2.402e-08
```

Listing 2.5: Cont. Example 2.2

Example 2.6 (Cont. Example 2.3) *Based on* (2.1.14)*, the least square estimators for the coefficients in Example 2.3 are*

$$\hat{\beta}_0 = 0.25, \quad \hat{\beta}_1 = 0.74, \quad \hat{\beta}_2 = 0.047, \quad \hat{\beta}_3 = -0.053, \quad \hat{\beta}_4 = -0.17.$$

```
summary(fit1 <- lm(Consumption ~ Income + Production + Savings +
    Unemployment, na.action=NULL))

#Call:
#lm(formula = Consumption ~ Income + Production + Savings +
    Unemployment,
#    na.action = NULL)

#Coefficients:
#               Estimate Std. Error t value Pr(>|t|)
#(Intercept)   0.253105   0.034470   7.343 5.71e-12 ***
#Income        0.740583   0.040115  18.461  < 2e-16 ***
#Production    0.047173   0.023142   2.038   0.0429 *
#Savings      -0.052890   0.002924 -18.088  < 2e-16 ***
#Unemployment -0.174685   0.095511  -1.829   0.0689 .

#Residual standard error: 0.3102 on 193 degrees of freedom
#Multiple R-squared:  0.7683, Adjusted R-squared:  0.7635
#F-statistic:   160 on 4 and 193 DF,  p-value: < 2.2e-16
```

Listing 2.6: Cont. Example 2.3

2.1.3 F Test Statistic

Various competing models are often of interest to isolate or select the best subset of independent variables. Suppose a proposed model specifies that only a subset $r < q$ independent variables, say, $z_{t,1:r} = \{z_{t1}, z_{t2}, \ldots, z_{tr}\}$ is influencing the dependent variable x_t. The reduced model is

$$x_t = \beta_0 + \beta_1 z_{t1} + \ldots + \beta_r z_{tr} + w_t, \tag{2.1.15}$$

where $\beta_1, \beta_2, \ldots, \beta_r$ are a subset of coefficients of the original q variables. The null hypothesis in this case is $H_0 : \beta_{r+1} = \ldots = \beta_q = 0$. We can test the reduced model (2.1.15) against the full model (2.1.1) by comparing the error sums of squares under the two models using the F-statistic

$$F = \frac{(SSE_r - SSE)/(q-r)}{SSE/(n-q-1)} = \frac{MSR}{MSE} \tag{2.1.16}$$

where SSE_r is the error sum of squares under the reduced model (2.1.15). Note that $SSE_r \geq SSE$ because the full model has more parameters. If $H_0 : \beta_{r+1} = \ldots = \beta_q = 0$, is true, then $SSE_r \approx SSE$ because the estimates of those β_s will be close to 0. Hence, we do not believe H_0 if $SSR = SSE_r - SSE$ is big. Under the null hypothesis, (2.1.16) has a central F-distribution with $q - r$ and $n - q - 1$ degrees of freedom when (2.1.15) is the correct model.

These results are often summarized in an Analysis of Variance (ANOVA) table. The difference in the numerator is often called the regression sum of squares (SSR). The null hypothesis is rejected at level α if $F > F^{q-r}_{n-q-1}(\alpha)$, the $1-\alpha$ percentile of the F distribution with $q - r$ numerator and $n - q - 1$ denominator degrees of freedom.

A special case of interest is the null hypothesis $H_0 : \beta_1 = \ldots = \beta_q = 0$. In this case $r = 0$, and the model in (2.1.15) becomes

$$x_t = \beta_0 + w_t.$$

Coefficient of Determination

Predictions of x_t can be obtained by using the estimated coefficients in the regression equation and setting the error term to zero. In general we write,

$$\hat{x}_t = \hat{\beta}_0 + \hat{\beta}_1 z_{t1} + \hat{\beta}_2 z_{t2} + \ldots + \hat{\beta}_q z_{tq}. \tag{2.1.17}$$

Plugging in the values of $z_{t1}, z_{t2} \ldots, z_{tq}$ for $t = 1, 2, \ldots, n$ returns predictions of x_t within the training set, referred to as **fitted values**. Note that these are predictions of the data used to estimate the model, not genuine forecasts of future values of x_t.

We may measure the proportion of variation in x_t that is accounted for (or explained) by the regression model using

$$R^2 = \frac{SSE_0 - SSE}{SSE_0} \tag{2.1.18}$$

where the residual sum of squares under the reduced model is

$$SSE_0 = \sum_{t=1}^{n} (x_t - \bar{x})^2. \tag{2.1.19}$$

In this case SSE_0 is the sum of squared deviations from the mean $\bar{x}$ and is otherwise known as the adjusted total sum of squares. The measure R^2 is called the **coefficient of determination**.

If the predictions are close to the actual values, we would expect R^2 to be close to 1. On the other hand, if the predictions are unrelated to the actual values, then $R^2 = 0$ (assuming there is an intercept). In all cases, R^2 lies between 0 and 1. The R^2 value is used frequently, though often incorrectly, in forecasting. The value of R^2 will **never decrease** when adding an extra predictor to the model and this can lead to **over-fitting**. There are no set rules for what is a good R^2 value, and typical values of R^2 depend on the type of data used. Validating a model's forecasting performance on the test data is much better than measuring the R^2 value on the training data.

An alternative which is designed to overcome the over-fitting problems is the adjusted-R^2, defined as:

$$adj - R^2 = 1 - (1 - R^2)\frac{n-1}{n-r-1}, \tag{2.1.20}$$

where r is the number of predictors. This is an improvement on R^2 as it will no longer increase with each added predictor. Using this measure, the best model will be the one with the largest value of adjusted-R^2.

Note 2.3 *The R^2 and F-statistic of Example 2.1, 2.2 and 2.3 are presented in the* R *codes.*

Model Selection

The techniques discussed in the previous part can be used to test various models against one another using the F test given in (2.1.16). These tests have been used in the past in a stepwise manner, where variables are added or deleted when the values from the F-test either exceed or fail to exceed some predetermined levels. The procedure, called stepwise multiple regression, is useful in arriving at a set of useful variables. For more information, refer to Section 7.5, Hyndman & Athanasopoulos (2018).

An alternative is to focus on a procedure for model selection that does not proceed sequentially, but simply evaluates each model on its own merits. Suppose we consider a normal regression model with k coefficients and denote the maximum likelihood estimator for the variance as

$$\hat{\sigma}_k^2 = \frac{SSE(k)}{n}, \tag{2.1.21}$$

where $SSE(k)$ denotes the residual sum of squares under the model with k regression coefficients. The following criteria are based on measuring the goodness of fit for a particular model by balancing the error of the fit against the number of parameters in the model.

Definition 2.1 (Akaike's Information Criterion (AIC))

$$AIC = \log(\hat{\sigma}_k^2) + \frac{n+2k}{n} \tag{2.1.22}$$

where $\hat{\sigma}_k^2$ is given by (2.1.21) *and k is the number of parameters in the model.*

The value of k yielding the **minimum AIC** specifies the best model. The idea is roughly that minimizing $\hat{\sigma}_k^2$ would be a reasonable objective, except that it decreases monotonically as k increases. Therefore, we ought to penalize the error variance by a term proportional to the number of parameters. The choice for the penalty term given by (2.1.22) is not the only one. A corrected form of AIC is defined as follows.

Definition 2.2 (AIC, Bias Corrected (AICc))

$$AICc = \log(\hat{\sigma}_k^2) + \frac{n+k}{n-k-2}, \tag{2.1.23}$$

where $\hat{\sigma}_k^2$ is given by (2.1.21)*, k is the number of parameters in the model and n is the sample size.*

As with the AIC, the AICc should be minimised.

Another correction of AIC is based on Bayesian arguments, which leads to the following.

Definition 2.3 (Bayesian Information Criterion (BIC))

$$BIC = \log(\hat{\sigma}_k^2) + \frac{k\log(n)}{n}, \tag{2.1.24}$$

where $\hat{\sigma}_k^2$ is given by (2.1.21)*, k is the number of parameters in the model and n is the sample size.*

BIC is also called the Schwarz Information Criterion (SIC). Notice that the penalty term in BIC is much larger than in AIC, consequently, **BIC tends to choose smaller models**. Various simulation studies have tended to verify that **BIC** does well at getting the **correct order in large samples**, whereas **AICc** tends to be superior in **smaller samples** where the **relative number of parameters is large**.

Example 2.7 *[Pollution, Temperature and Mortality] The data in Figure 2.4 is used to study the possible effects of temperature and pollution on weekly mortality in Los Angeles County. As can be seen, there is a the strong seasonal components in all of the series, corresponding to winter-summer variations and the downward trend in the cardiovascular mortality over the 10-year period.*

```r
par(mfrow=c(3,1)) # plot the data
plot(cmort, main="Cardiovascular Mortality", xlab="", ylab="", lwd=2)
plot(tempr, main="Temperature", xlab="", ylab="", lwd=2)
plot(part, main="Particulates", xlab="", ylab="", lwd=2)
```

Listing 2.7: Time series plot in Example 2.7

A scatterplot matrix, shown in Fig. 2.5, indicates a possible linear relation between mortality and the pollutant particulates and a possible relation to temperature. Note the curvilinear shape of the temperature mortality curve, indicating that higher temperatures as well as lower temperatures are associated with increases in cardiovascular mortality.

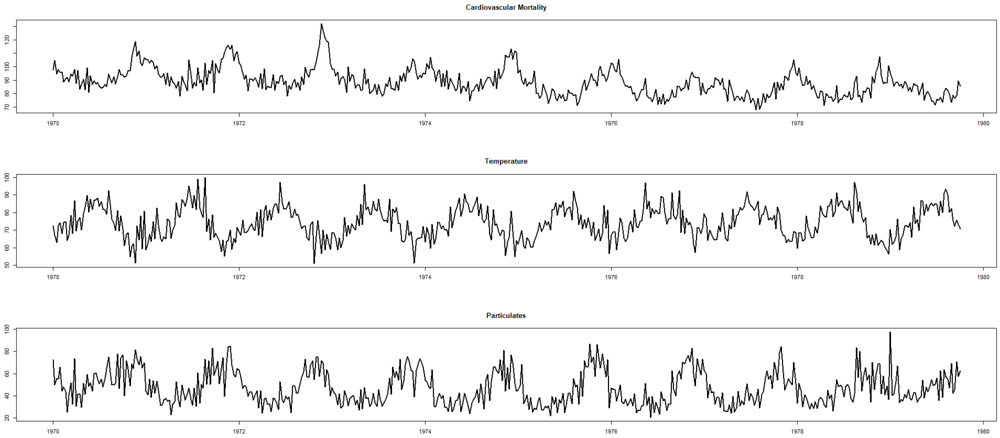

Figure 2.4: Average weekly cardiovascular mortality (top), temperature (middle) and particulate pollution (bottom) in Los Angeles County. There are 508 six-day smoothed averages obtained by filtering daily values over the 10 year period 1970–1979

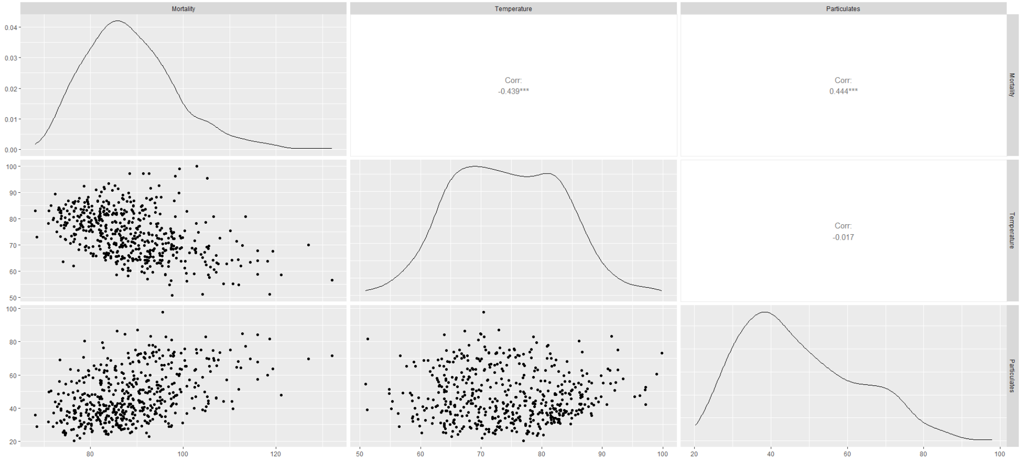

Figure 2.5: Scatterplot matrix showing relations between mortality, temperature, and pollution

```r
library(ggplot2)
library(GGally)

df=as.data.frame(cbind(Mortality=cmort, Temperature=tempr, Particulates
    =part))
ggpairs(df)
```

Listing 2.8: Scatterplot matrix in Example 2.7

Based on the scatterplot matrix, we entertain, tentatively, four models where M_t denotes cardiovascular mortality, T_t denotes temperature and P_t denotes the particulate *levels. They are*

$$\textit{Trend} \quad M_t = \beta_0 + \beta_1 t + w_t, \tag{2.1.25}$$

$$\textit{Linear} \quad M_t = \beta_0 + \beta_1 t + \beta_2 (T_t - T_\cdot) + w_t, \tag{2.1.26}$$

$$\textit{Curvilinear} \quad M_t = \beta_0 + \beta_1 t + \beta_2 (T_t - T_\cdot) + \beta_3 (T_t - T_\cdot)^2 + w_t, \tag{2.1.27}$$

$$\textit{Curvilinear} \quad M_t = \beta_0 + \beta_1 t + \beta_2 (T_t - T_\cdot) + \beta_3 (T_t - T_\cdot)^2 + \beta_4 P_t + w_t, \tag{2.1.28}$$

where we adjust temperature for its mean, $T_\cdot = 74.26$, *to avoid collinearity problems.*

```
temp = tempr-mean(tempr) # center temperature
temp2 = temp^2
trend = time(cmort) # time
fit = lm(cmort~ trend + temp + temp2 + part, na.action=NULL)
summary(fit) # regression results
summary(aov(fit))
#             Df Sum Sq Mean Sq F value Pr(>F)
#trend         1  10667   10667  261.62 <2e-16 ***
#temp          1   8607    8607  211.09 <2e-16 ***
#temp2         1   3429    3429   84.09 <2e-16 ***
#part          1   7476    7476  183.36 <2e-16 ***
#Residuals   503  20508      41
```

Listing 2.9: ANOVA table in Example 2.7

Model	k	SSE	df	MSE	R^2	AIC	BIC
(2.1.25)	2	40,020	506	79.0	.21	5.38	5.40
(2.1.26)	3	31,413	505	62.2	.38	5.14	5.17
(2.1.27)	4	27,985	504	55.5	.45	5.03	5.07
(2.1.28)	5	20,508	503	40.8	.60	4.72	4.77

Table 2.1: Summary statistics for mortality models

We note that each model does substantially better than the one before it and that the model including temperature, temperature squared, and particulates does the best, accounting for some 60% of the variability and with the best value for AIC and BIC (because of the large sample size, AIC and AICc are nearly the same).

Note that one can compare any two models using F test in Equation (2.1.16) *. Hence, a model with only trend could be compared to the full model, which means we want to test* $H_0 : \beta_2 = \beta_3 = \beta_4 = 0$, *using* $q = 4$, $r = 1$, $n = 508$, *and*

$$F_{3,503} = \frac{(40020 - 20508)/3}{20508/503} = 160$$

which exceeds $F_{3,503}(0.001) = 5.51$. *We obtain the best prediction model,*

$$\hat{M}_t = 2831.5 - 1.396t - 0.472(T_t - 74.26) + 0.023(T_t - 74.26)^2 + 0.255P_t,$$

for mortality. As expected, a negative trend is present in time as well as a negative coefficient for adjusted temperature. The quadratic effect of temperature can clearly be seen in the scatterplots of Figure (2.5). *Pollution weights positively and can be interpreted as the incremental contribution to daily deaths per unit of particulate pollution.*

It would still be essential to check the residuals $\hat{w}_t = M_t - \hat{M}_t$ for autocorrelation. The estimated model violates the assumption of no autocorrelation in the errors, and our forecasts may be inefficient, so there is some information left over which should be accounted for in the model in order to obtain better forecasts. HOW?? will be discusssed later.

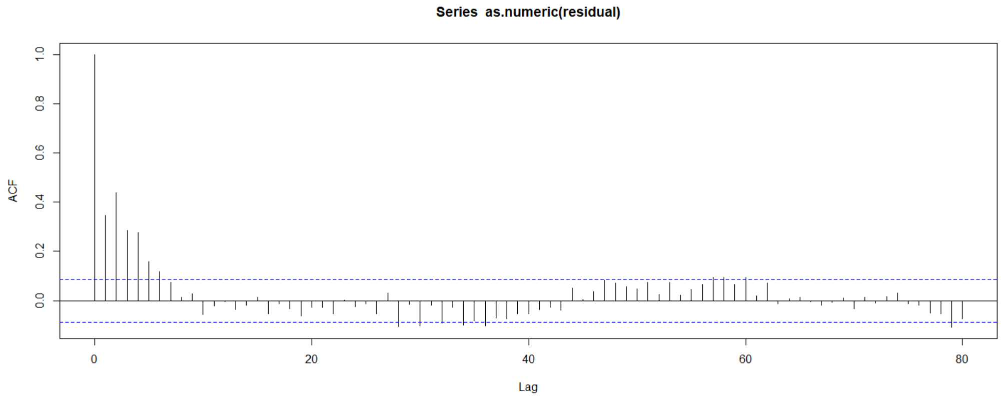

Figure 2.6: ACF of residuals of model (2.1.28)

Note 2.4 *For other model evaluations based on residuals, refer to Section 7.3, Hyndman & Athanasopoulos (2018).*

Example 2.8 *[Regression With Lagged Variables] In this example, we are going to analyse the monthly values of an environmental series called the Southern Oscillation Index (SOI) and associated Recruitment (number of new fish). Both series are for a period of 453 months ranging over the years 1950–1987. Figure 2.7 displays both series, which exhibit repetitive behavior, with regularly repeating cycles that are easily visible.*

It seems that the two series are also related; it is easy to imagine the fish population is dependent on the ocean temperature. This possibility suggests trying some version of regression analysis as a procedure for relating the two series.

Figure 2.8 shows the autocorrelation and cross-correlation functions (ACFs and CCF) for these two series. Both of the ACFs exhibit periodicities corresponding to the correlation between values separated by 12 units. Observations 12 months or one year apart are strongly positively correlated, as are observations at multiples such as 24, 36, 48, . . . Observations separated by six months are negatively correlated, showing that positive excursions tend to be associated with negative excursions six months removed.

The sample CCF in Fig. 2.8, however, shows some departure from the cyclic component of each series and there is an obvious peak at $h = -6$. This result implies that SOI measured at time $t - 6$ months is associated with the Recruitment series at time t. We could say the SOI leads the Recruitment series by six months. The sign of the CCF is negative, leading to the conclusion that the two series move in different directions; that is, increases in SOI lead to decreases in Recruitment and vice versa.

```r
# Time series plot
par(mfrow = c(2,1)) # set up the graphics
```

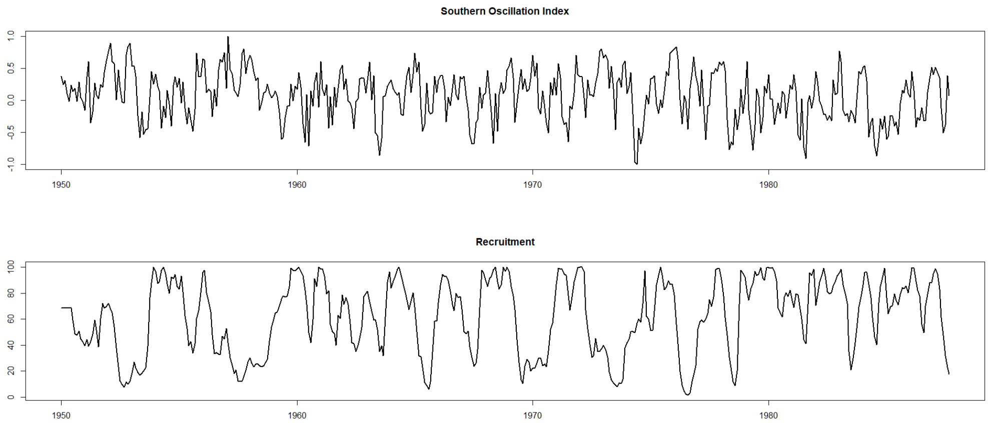

Figure 2.7: Monthly SOI and Recruitment (estimated new fish), 1950–1987

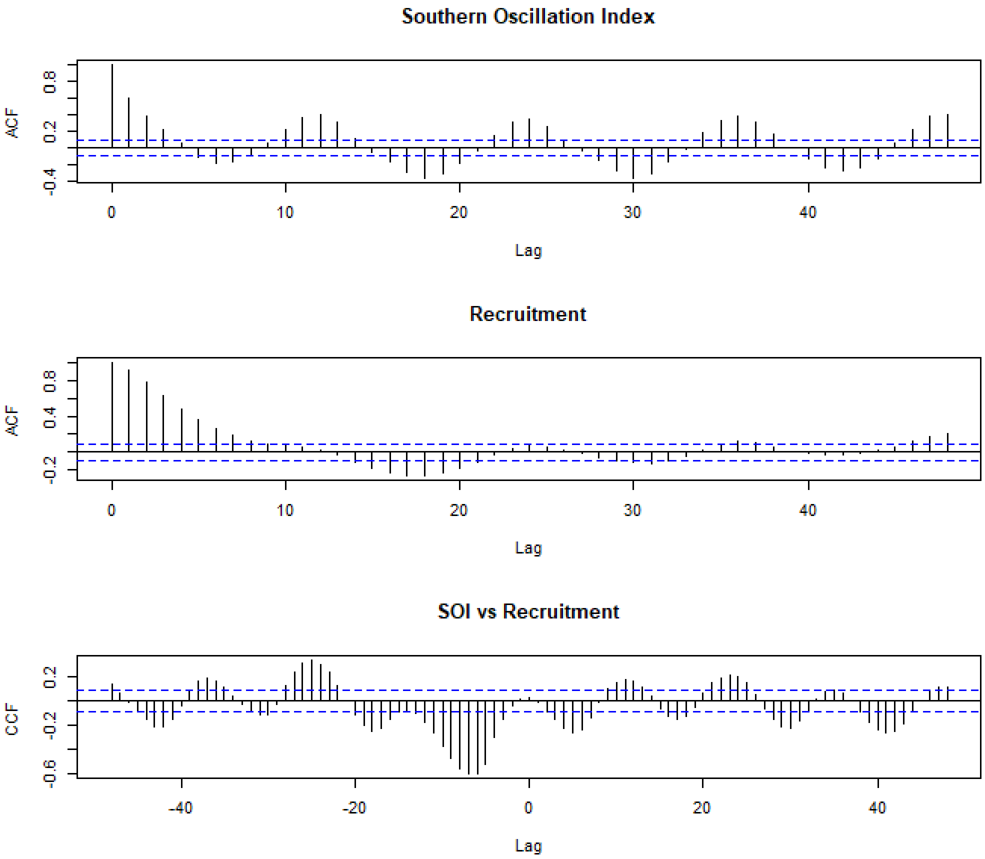

Figure 2.8: Sample ACFs of the SOI series (top) and of the Recruitment series (middle), and the sample CCF of the two series (bottom); negative lags indicate SOI leads Recruitment. The lag axes are in terms of seasons (12 months)

```
plot(soi, ylab="", xlab="", lwd=2, main="Southern Oscillation Index")
plot(rec, ylab="", xlab="", lwd=2, main="Recruitment")


```

```r
# ACF and CCF plots
par(mfrow=c(3,1))
acf(soi, 48, main="Southern Oscillation Index")
acf(rec, 48, main="Recruitment")
ccf(soi, rec, 48, main="SOI vs Recruitment", ylab="CCF")
```

Listing 2.10: Codes for Example 2.8

lets assume that the relationship between Recruitment and SOI is linear:

$$R_t = \beta_0 + \beta_1 S_{t-6} + w_t, \tag{2.1.29}$$

where R_t denotes Recruitment for month t and S_{t-6} denotes SOI six months prior. Assuming the w_t sequence is white, the fitted model is

$$\hat{R}_t = 65.790 + -44.283 S_{t-6}, \tag{2.1.30}$$

with $\sigma_w = 22.5$ on 445 degrees of freedom. This result indicates the strong predictive ability of SOI for Recruitment six months in advance. Of course, it is still essential to check the model assumptions.

```r
# ts.intersect and lag are functions in stats package
fish = ts.intersect(rec, soiL6=stats::lag(soi,6), dframe=TRUE)
summary(fit <- lm(rec~soiL6, data=fish, na.action=na.omit))

#Call:
#lm(formula = rec ~ soiL6, data = fish, na.action = NULL)

#Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
#(Intercept)    65.790      1.088   60.47   <2e-16 ***
#soiL6         -44.283      2.781  -15.92   <2e-16 ***

#Residual standard error: 22.5 on 445 degrees of freedom
#Multiple R-squared:  0.3629, Adjusted R-squared:  0.3615
#F-statistic: 253.5 on 1 and 445 DF,  p-value: < 2.2e-16
```

Listing 2.11: Model fitting in Example 2.8

2.2 Exploratory Data Analysis

With time series data, it is the dependence between the values of the series that is important to measure; we must, at least, be able to estimate autocorrelations with precision. It would be difficult to measure that dependence if the dependence structure is not regular or is changing at every time point. Hence, to achieve any meaningful statistical analysis of time series data, it will be crucial that at least the mean and the autocovariance functions satisfy the conditions of stationarity. Often, this is not the case, and we will mention some methods in this section for playing down the effects of nonstationarity.

Example 2.9 *Consider the Johnson & Johnson series in Figure 1.2. As can be seen clearly, this time series has a mean that increases exponentially over time, and the increase in the magnitude of the fluctuations around this trend causes changes in the covariance function; the variance of the process clearly increases as one progresses over the length of the series.*

2.2.1 Removing Trend

The easiest form of nonstationarity to work with is the trend stationary model wherein the process has stationary behavior around a trend:

$$x_t = \mu_t + y_t, \tag{2.2.1}$$

where x_t are the observations, μ_t denotes the trend, and y_t is a stationary process. Quite often, strong trend will obscure the behavior of the stationary process, y_t. Hence, there is some advantage to removing the trend as a first step in an exploratory analysis of such time series. The steps involved are to obtain a reasonable estimate of the trend component, say $\hat{\mu}_t$, and then work with the residuals

$$\hat{y}_t = x_t - \hat{\mu}_t. \tag{2.2.2}$$

Example 2.10 (Detrending Chicken Prices) *As suggested in Example 2.1, in the analysis of the chicken price data, we should consider $\mu_t = \beta_0 + \beta_1 t$ as the trend. If we estimated $\hat{\mu}_t$ using ordinary least squares, we found that*

$$\hat{\mu}_t = -7131 + 3.59t. \tag{2.2.3}$$

Figure 2.1 shows the data with the estimated trend line superimposed. To obtain the detrended series, we simply subtract $\hat{\mu}_t$ from the observations, x_t, to obtain the detrended series

$$\hat{y}_t = x_t + 7131 - 3.59t. \tag{2.2.4}$$

The middle graph of Figure 2.9 shows the detrended series. Figure 2.10 shows the ACF of the original data (top panel) as well as the ACF of the detrended data (middle panel).

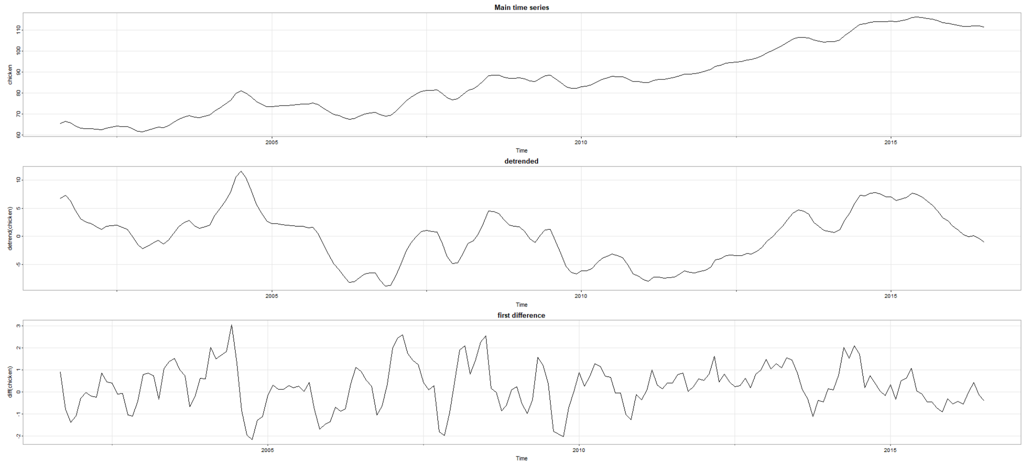

Figure 2.9: Original (top), Detrended (middle) and differenced (bottom) chicken price series.

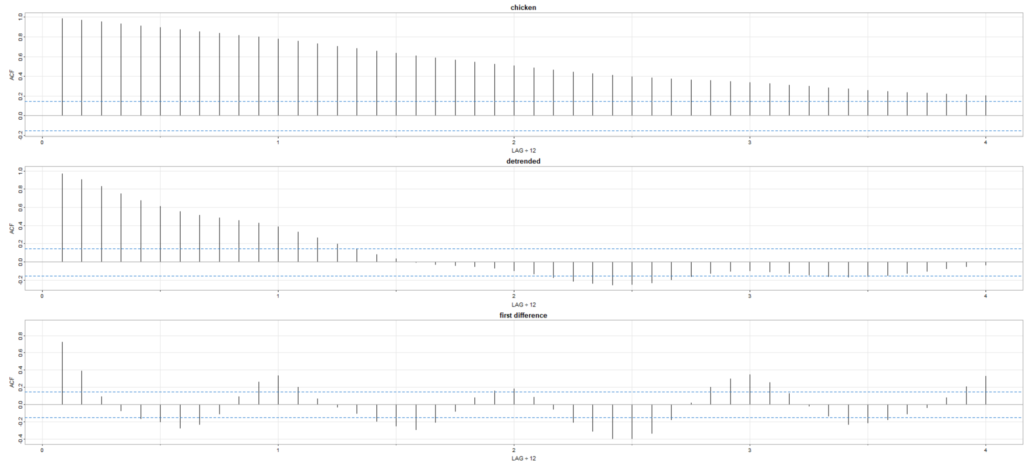

Figure 2.10: Sample ACFs of chicken prices (top), and of the detrended (middle) and the differenced (bottom) series

In Chapter 1, we saw that a random walk might also be a good model for trend. That is, rather than modeling trend as fixed, we might model trend as a stochastic component using the random walk with drift model,

$$\mu_t = \delta + \mu_{t-1} + w_t, \tag{2.2.5}$$

where w_t is white noise and is independent of y_t. If the appropriate model is (2.2.1), then differencing the data, x_t, yields a stationary process; that is,

$$\begin{aligned} x_t - x_{t-1} &= (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) \\ &= \delta + w_t + y_t - y_{t-1}. \end{aligned} \tag{2.2.6}$$

It is easy to show $z_t = y_t - y_{t-1}$ is stationary. That is, because y_t is stationary,

$$\begin{aligned} \gamma_z(h) = Cov(z_{t+h}, z_t) &= Cov(y_{t+h} - y_{t+h-1}, y_t - y_{t-1}) \\ &= 2\gamma_y(h) - \gamma_y(h+1) + \gamma_y(h-1), \end{aligned}$$

is independent of time; and it can be shown that $x_t - x_{t-1}$ in (2.2.6) is stationary.

- One advantage of differencing over detrending to remove trend is that no parameters are estimated in the differencing operation.
- One disadvantage of differencing over detrending is that differencing does not yield an estimate of the stationary process y_t as can be seen in 2.2.6.
- If an estimate of y_t is essential, then detrending may be more appropriate.
- If the goal is to coerce the data to stationarity, then differencing may be more appropriate.

Differencing is also a viable tool if the trend is fixed, as in Example 2.4. That is, e.g., if $\mu_t = \beta_0 + \beta_1 t$ in the model (2.2.1), differencing the data produces stationarity:

$$x_t - x_{t-1} = (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) = \beta_1 + y_t - y_{t-1}.$$

Because differencing plays a central role in time series analysis, it receives its own notation. The first difference is denoted as

$$\nabla x_t = x_t - x_{t-1}. \tag{2.2.7}$$

As we have seen, the first difference eliminates a linear trend. A second difference, that is, the difference of (2.2.7), can eliminate a quadratic trend, and so on. In order to define higher differences, we need a variation in notation.

More on Differencing

Nonstationarity arises in many ways. The first obvious way in which a process is nonstationary is when the mean function $\mu_x(t) \neq \mu_x$ for all t. For example it could be a deterministic trend in time (linear, exponential, polynomial etc.) or a function of other processes which are not stationary. Another obvious way in which a process is nonstationary is when the variance is not constant through time or autocovariance function does not only depend on time separation in the process. There is also the phenomena of random walk behaviour and its variants in which the current value of the process is a random increment from the previous value of the process.

The simplest way to detect non-stationarity is to examine a time plot of the data. A simple method to remove many common forms of nonstationarity is to difference the observations at certain lags, typically one time lag or at a seasonal lag.

Definition 2.4 (The backshift operator) *We define the backshift operator, B, by*

$$Bx_t = x_{t-1}$$

Therefore, B takes as input a time series and produces as output the series shifted backwards in time by one time unit and iterating this we get

$$B^k x_t = x_{t-k} \tag{2.2.8}$$

which shifts the series backwards in time by k time units.

The idea of an inverse operator can also be given if we require $B^{-1}B = 1$, so that

$$x_t = B^{-1}Bx_t = B^{-1}x_{t-1}.$$

That is, B^{-1} is the forward-shift operator. Besides, the **difference operator**, ∇, introduced in (2.2.7), is defined as

$$\nabla = (1 - B)$$

which, we referred to as lag 1 differences. we may extend the notion further. For example, the second difference becomes

$$\nabla^2 x_t = (1 - B)^2 x_t = (1 - 2B + B2)x_t = x_t - 2x_{t-1} + x_{t-2} \tag{2.2.9}$$

by the linearity of the operator. To check, just take the difference of the first difference $\nabla(\nabla x_t) = \nabla(x_t - x_{t-1}) = (x_t - x_{t-1}) - (x_{t-1} - x_{t-2})$.

Definition 2.5 (Differences of order d) *Differences of order d are defined as*

$$\nabla^d = (1 - B)^d, \tag{2.2.10}$$

where we may expand the operator $(1 - B)^d$ algebraically to evaluate for higher integer values of d. When $d = 1$, we drop it from the notation.

For seasonal time series, we need to define a difference operator that can be useful in obtaining a series that is free of seasonal patterns and is stationary.

Definition 2.6 (Seasonal difference operator) *For seasonal time series with period, S, the seasonal difference operator, ∇_S, is defined as*

$$\nabla_S = (1 - B^S) \tag{2.2.11}$$

and its action on a time series $\{x_t\}$ is

$$\nabla_S x_t = (1 - B^S)x_t = x_t - B^S x_t = x_t - x_{t-S}$$

which are referred to as seasonal differences.

The lag 1 difference is an example of a linear filter applied to eliminate a trend. Other filters, formed by averaging values near xt , can produce adjusted series that eliminate other kinds of unwanted fluctuations. The differencing technique is an important component of the ARIMA model of Box and Jenkins and will be discussed more later.

Example 2.11 (Differencing Chicken Prices) *The first difference of the chicken prices series, also shown in Figure 2.9, produces different results than removing trend by detrending via regression. For example, the differenced series does not contain the long (five-year) cycle we observe in the detrended series. The ACF of this series is also shown in Figure 2.10. In this case, the differenced series exhibits an annual cycle that was obscured in the original or detrended data.*

```r
# astsa now has a detrend script, so Figure 2.4 can be done as
par(mfrow=c(3,1))
tsplot( chicken, main="Main time series" )
tsplot( detrend(chicken), main="detrended" )
tsplot( diff(chicken), main="first difference" )

# and Figure 2.5 as

par(mfrow=c(3,1))      # plot ACFs
acf1(chicken, 48, main="chicken")
acf1(detrend(chicken), 48, main="detrended")
acf1(diff(chicken), 48, main="first difference")
```

Listing 2.12: The R code to reproduce Figures 2.9 and 2.10

Example 2.12 (Regression with Lagged Variables (cont)) *In Example 2.8, we regressed Recruitment on lagged SOI,*

$$R_t = \beta_0 + \beta_1 S_{t-6} + w_t.$$

However, it can be shown that the relationship is nonlinear and different when SOI is positive or negative. In this case, we may consider adding a dummy variable to account for this change. In particular, we fit the model

$$R_t = \beta_0 + \beta_1 S_{t-6} + \beta_2 D_{t-6} + \beta_3 D_{t-6} S_{t-6} + w_t.$$

where D_t is a dummy variable that is 0 if $S_t < 0$ and 1 otherwise. This means that

$$R_t = \begin{cases} \beta_0 + \beta_1 S_{t-6} + w_t & \text{if } S_{t-6} < 0, \\ (\beta_0 + \beta_2) + (\beta_1 + \beta_3) S_{t-6} + w_t & \text{if } S_{t-6} \geq 0. \end{cases}$$

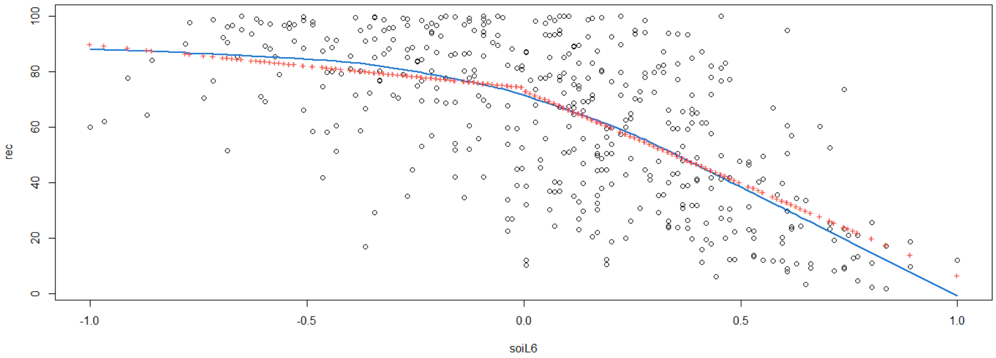

Figure 2.11: Plot of Recruitment (R_t) vs SOI lagged 6 months (S_{t-6}) with the fitted values of the regression as points and a lowess fit

The result of the fit is given in the R code below.

```
dummy = ifelse(soi<0, 0, 1)
fish = ts.intersect(rec, soiL6=stats::lag(soi,-6), dL6=stats::lag(dummy
    ,-6), dframe=TRUE)
summary(fit <- lm(rec˜ soiL6*dL6, data=fish, na.action=na.omit))
#Call:
#lm(formula = rec ˜ soiL6 * dL6, data = fish, na.action = na.omit)

#Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
#(Intercept)    74.479      2.865  25.998  < 2e-16 ***
#soiL6         -15.358      7.401  -2.075   0.0386 *
#dL6            -1.139      3.711  -0.307   0.7590
#soiL6:dL6     -51.244      9.523  -5.381  1.2e-07 ***
#---

#Residual standard error: 21.84 on 443 degrees of freedom
#  (6 observations deleted due to missingness)
#Multiple R-squared:  0.4024, Adjusted R-squared:  0.3984
#F-statistic: 99.43 on 3 and 443 DF,  p-value: < 2.2e-16
```

Listing 2.13: The fitted model for Example 2.12.

Figure 2.11 shows R_t vs S_{t-6} with the fitted values of the regression and a lowess fit superimposed. The piecewise regression fit is similar to the lowess fit, but we note that the residuals are not white noise. This is followed up Chapter 3.

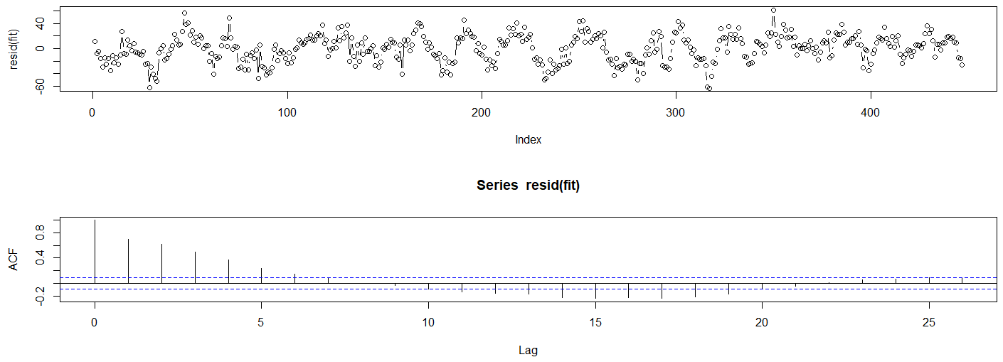

Figure 2.12: The scatter plot and ACF of the residuals of Example 2.12

```
par(mfrow=c(2,1))
plot(resid(fit), type="b")
acf(resid(fit))
```

Listing 2.14: The scatter plot and ACF of residuals for Example 2.12

In the following, we discuss assessing periodic behavior in time series data using regression analysis. In Example 1.10, we briefly discussed the problem of identifying cyclic or periodic signals in time series. A number of the time series we have seen so far exhibit periodic behavior. For example, the Johnson & Johnson data shown in Figure 1.2 make one cycle every year (four quarters) on top of an increasing trend and the speech data in Figure 1.3 is highly repetitive. The monthly SOI and Recruitment series in Figure 2.7 show strong yearly cycles, which obscures the slower El Nino cycle.

Example 2.13 (Using Regression to Discover a Signal in Noise) *In Example 1.10, we generated $n = 500$ observations from the model*

$$x_t = A\cos(2\pi\omega t + \phi) + w_t \tag{2.2.12}$$

where $\omega = 1/50$, $A = 2$, $\phi = 0.6\pi$, and $\sigma_w = 5$; the data are shown on the bottom panel of Figure **??**. *At this point we assume the frequency of oscillation $\omega = 1/50$ is known, but A and ϕ are unknown parameters. In this case the parameters appear in* (2.2.12) *in a nonlinear way, so we use a trigonometric identity and write*

$$Acos(2\pi\omega t + \phi) = \beta_1 cos(2\pi\omega t) + \beta_2 sin(2\pi\omega t),$$

where $\beta_1 = Acos(\phi)$ and $\beta_2 = -Asin(\phi)$. Now the model (2.2.12) *can be written in the usual linear regression form given by (no intercept term is needed here)*

$$x_t = \beta_1 cos(2\pi\omega t) + \beta_2 sin(2\pi\omega t) + w_t \tag{2.2.13}$$

Using linear regression, we find $\hat{\beta} = -0.74(0.33)$, $\hat{\beta}2 = -1.99(0.33)$ with $\hat{\sigma}_w = 5.18$; the values in parentheses are the standard errors. We note the actual values of the coefficients for this example are $\beta_1 = 2cos(0.6\pi) = -0.62$, and $\beta_2 = -2sin(.6\pi) = -1.90$. It is clear

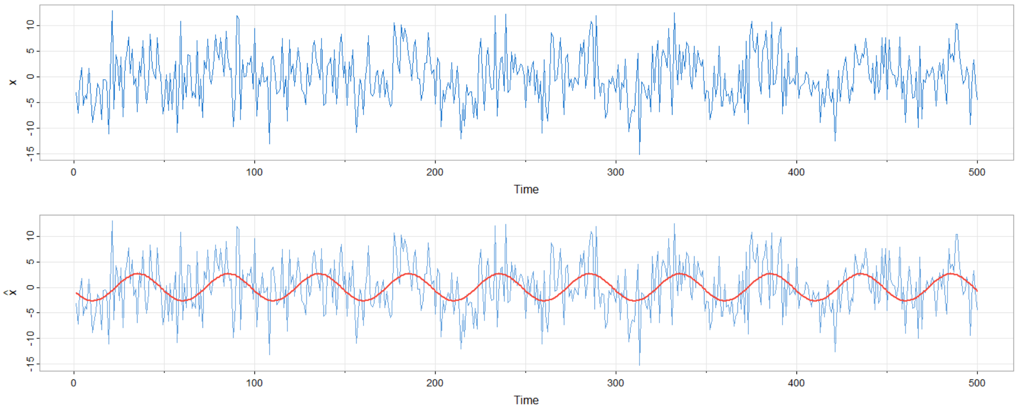

Figure 2.13: Data generated by (2.2.12) [top] and the fitted line superimposed on the data [bottom]

that we are able to detect the signal in the noise using regression, even though the signal-to-noise ratio is small. Figure 2.13 shows data generated by (2.2.12) *with the fitted line superimposed.*

To reproduce the analysis and Figure 2.13 in R, use the following:

```r
set.seed(1000)  # so you can reproduce these results
x  = 2*cos(2*pi*1:500/50 + .6*pi) + rnorm(500,0,5)
z1 = cos(2*pi*1:500/50)
z2 = sin(2*pi*1:500/50)
summary(fit <- lm(x~0+z1+z2))  # zero to exclude the intercept
par(mfrow=c(2,1))
tsplot(x, col=4)
tsplot(x, col=astsa.col(4,.7), ylab=expression(hat(x)))
lines(fitted(fit), col=2, lwd=2)
```

Listing 2.15: The scatter plot and ACF of residuals for Example 2.12

Example 2.14 (Seasonally differenced series) *Figure 2.14 is a time series plot of employment in Trades in Wisconsin measured each month over five years (top panel), the seasonal (lag=12) differenced data (middle panel) and the seasonal differenced followed by lag one differenced data (bottom panel).*

Figure 2.15 shows (left panel) estimates of the autocorrelation function for the time series of employment in Trades in Wisconsin, that of the seasonal lag (= 12*) differenced data (middle panel), and that of the seasonally differenced then lag 1 differenced data (right panel). Note that lag 12 differencing appears to remove the seasonal pattern obvious in the original series but that there is evidence of non-stationarity of the mean for these differences which is removed when additional lag 1 differencing is applied.*

Note 2.5 *The differencing operators commute:* $\nabla_S \nabla = \nabla \nabla_S$ *so it does not matter in which order they are applied.*

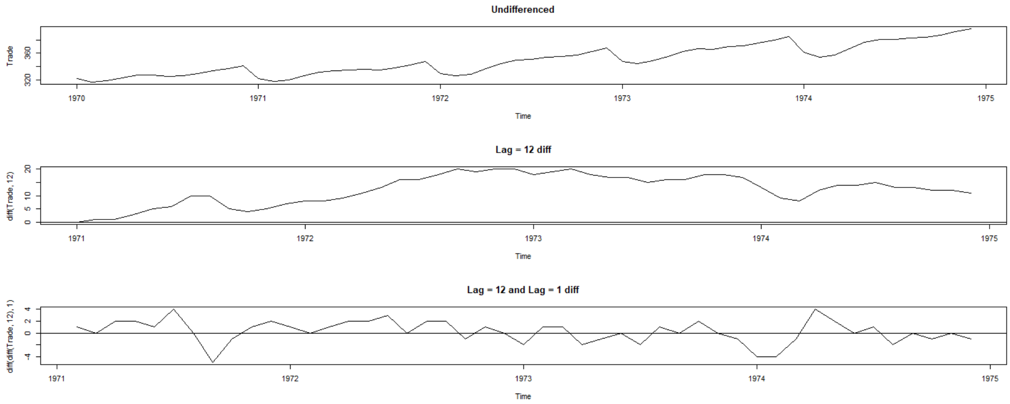

Figure 2.14: Monthly Employment in Trades in Wisconsin (top), Seasonal differences (middle) and combined seasonal and lag 1 differences (bottom)

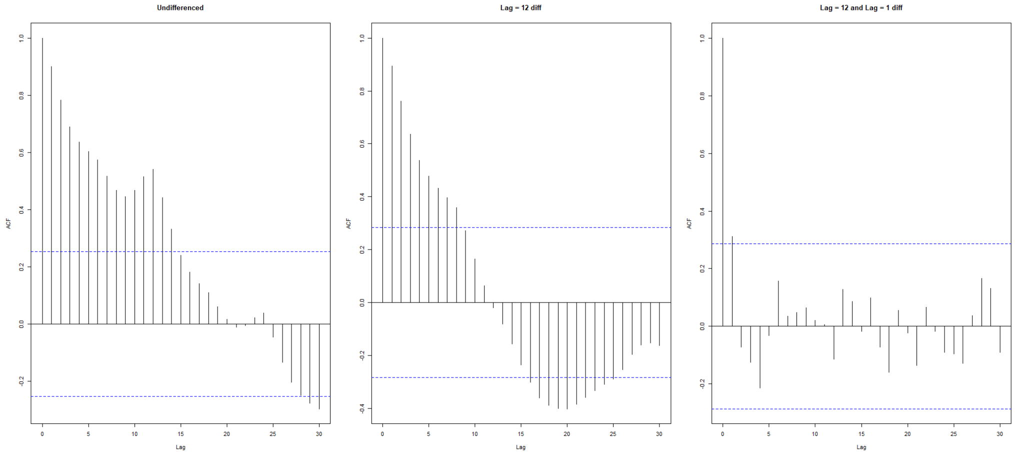

Figure 2.15: Autocorrelation function for Monthly Employment in Trades (left) and seasonal (lag =12) differences (middle) and combined seasonal and monthly differences (right)

2.3 Smoothing in the Time Series Context

In Chapter 1, we introduced the concept of filtering or smoothing a time series, and also discussed using a moving average to smooth white noise. This method is useful in discovering certain traits in a time series, such as long-term trend and seasonal components. In particular, if xt represents the observations, then

$$m_t = \sum_{j=-k}^{k} a_j x_{t-j} \tag{2.3.1}$$

is a symmetric moving average of the data. In the following examples, we are going to present different methods of smoothing in time series.

Example 2.15 (Moving Average Smoother) *For example, Figure 2.16 shows the monthly*

SOI series discussed in Example 2.8, smoothed using (2.3.1) *with weights* $a_0 = a_{\pm 1} = \ldots = a_{\pm 5} = 1/12$*, and* $a_{\pm 6} = 1/24$; $k = 6$. *This particular method removes (filters out) the obvious annual temperature cycle and helps emphasize the El Nino cycle. To reproduce Figure 2.16 in R:*

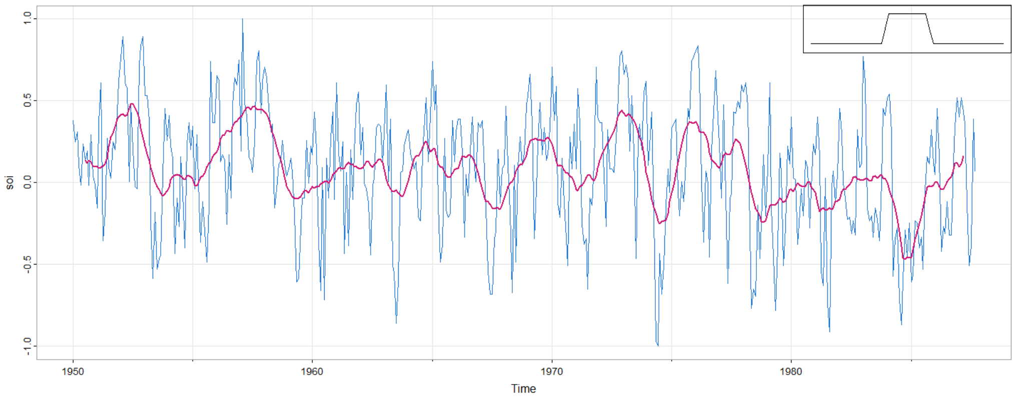

Figure 2.16: Moving average smoother of SOI. The insert shows the shape of the moving average ("boxcar") kernel [not drawn to scale] described in (2.39)

```
wgts = c(.5, rep(1,11), .5)/12
soif = filter(soi, sides=2, filter=wgts)
tsplot(soi, col=4)
lines(soif, lwd=2, col=6)
par(fig = c(.75, 1, .75, 1), new = TRUE) # the insert
nwgts = c(rep(0,20), wgts, rep(0,20))
plot(nwgts, type="l", ylim = c(-.02,.1), xaxt='n', yaxt='n', ann=FALSE)
```

Listing 2.16: The code to reproduce Figure 2.16

Although the moving average smoother does a good job in highlighting the El Nino effect, it might be considered too choppy. We can obtain a smoother fit using the normal distribution for the weights, instead of boxcar-type weights of (2.3.1).

Example 2.16 (Kernel Smoothing) *Kernel smoothing is a moving average smoother that uses a weight function, or kernel, to average the observations. Figure 2.17 shows kernel smoothing of the SOI series, where* m_t *is now*

$$m_t = \sum_{i=1}^{n} w_i(t) x_i \tag{2.3.2}$$

where

$$w_i(t) = K\left(\frac{t-i}{b}\right) \Big/ \sum_{j=1}^{n} K\left(\frac{t-j}{b}\right) \tag{2.3.3}$$

are the weights and $K(\cdot)$ *is a kernel function. This estimator is often called the Nadaraya–Watson estimator. In this example, and typically, the normal kernel,* $K(z) = \frac{1}{\sqrt{2\pi}} \exp\{-z^2/2\}$*, is used.*

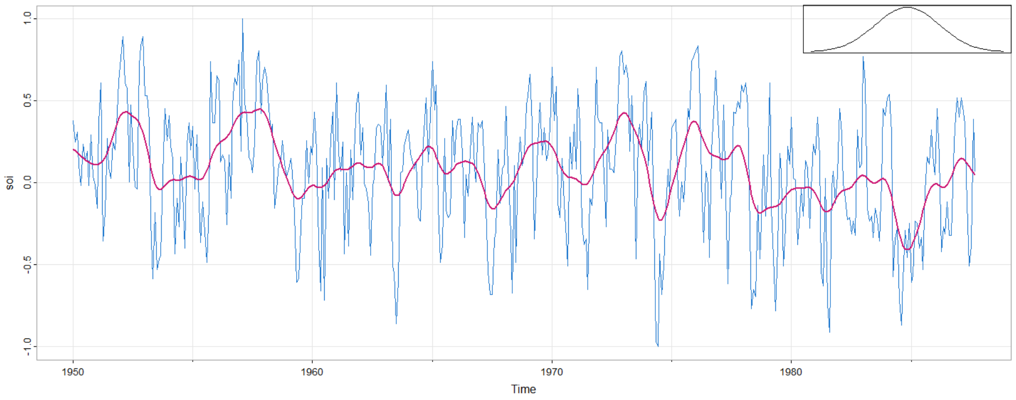

Figure 2.17: Kernel smoother of SOI. The insert shows the shape of the normal kernel [not drawn to scale]

To implement this in R, use the `ksmooth` *function where a bandwidth can be chosen. The wider the bandwidth, b, the smoother the result. In Figure 2.17, we used the value of $b = 1$ to correspond to approximately smoothing a little over one year. Figure 2.17 can be reproduced in R as follows.*

```r
tsplot(soi, col=4)
lines(ksmooth(time(soi), soi, "normal", bandwidth=1), lwd=2, col=6)
par(fig = c(.75, 1, .75, 1), new = TRUE) # the insert
curve(dnorm, -3, 3,  xaxt='n', yaxt='n', ann=FALSE)
```

Listing 2.17: The code to reproduce Figure 2.17

Example 2.17 (Lowess) *Another approach to smoothing a time plot is* **nearest neighbor regression**. *The technique is based on k-nearest neighbors regression, wherein one uses only the data $\{x_{t-k/2}, \ldots, x_t, \ldots, x_{t+k/2}\}$ to predict x_t via regression, and then sets $m_t = \hat{x}_t$.*

Lowess *is a method of smoothing that is rather complex, but the basic idea is close to nearest neighbor regression. Figure 2.18 shows smoothing of SOI using the R function lowess. First, a certain proportion of nearest neighbors to x_t are included in a weighting scheme; values closer to x_t in time get more weight. Then, a robust weighted regression is used to predict x_t and obtain the smoothed values m_t. The larger the fraction of nearest neighbors included, the smoother the fit will be. In Figure 2.18, one smoother uses 5% of the data to obtain an estimate of the El Nino cycle of the data. In addition, a (negative) trend in SOI would indicate the long-term warming of the Pacific Ocean. To investigate this, we used lowess with the default smoother span of $f = 2/3$ of the data.*

Figure 2.18 can be reproduced in R as follows.

```r
plot(soi)
lines(lowess(soi, f=.05), lwd=2, col=4) # El Nino cycle
lines(lowess(soi), lty=2, lwd=2, col=2) # trend (with default span)
```

Listing 2.18: The code to reproduce Figure 2.18

Example 2.18 (Smoothing Splines) *An obvious way to smooth data would be to fit a polynomial regression in terms of time. For example, a cubic polynomial would have*

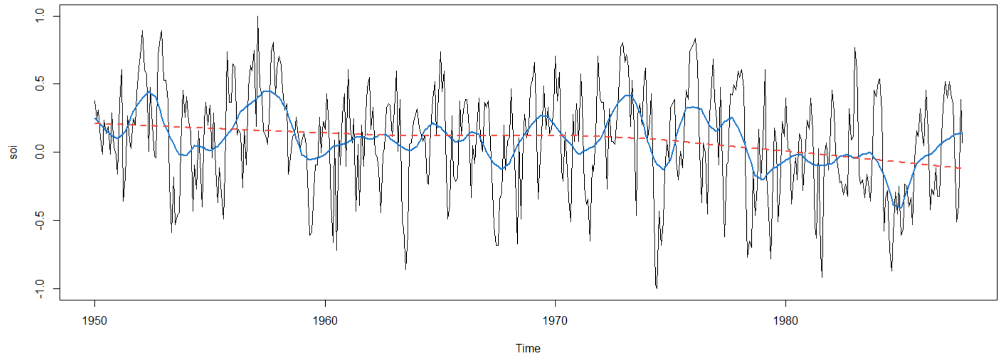

Figure 2.18: Locally weighted scatterplot smoothers (lowess) of the SOI series

$x_t = m_t + w_t$ where

$$m_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3. \tag{2.3.4}$$

We could then fit m_t via ordinary least squares.

An extension of polynomial regression is to first divide time $t = 1, \ldots, n$, into k intervals, $[t_0 = 1, t_1], [t_1 + 1, t_2], \ldots, [t_{k-1} + 1, t_k = n]$; the values $t_0, t_1, \ldots, t_k$ are called **knots**. *Then, in each interval, one fits a polynomial regression, typically the order is 3, and this is called* **cubic splines**.

A related method is **smoothing splines**, *which minimizes a compromise between the fit and the degree of smoothness given by*

$$\sum_{t=1}^{n} [x_t - m_t]^2 + \lambda \int (m"_t)\, dt \tag{2.3.5}$$

where m_t is a cubic spline with a knot at each t and primes denote differentiation. The degree of smoothness is controlled by $\lambda > 0$.

Think of taking a long drive where m_t is the position of your car at time t. In this case, $m"_t$ is instantaneous acceleration/deceleration, and $\int (m"_t)\, dt$ is a measure of the total amount of acceleration and deceleration on your trip. A smooth drive would be one where a constant velocity, is maintained (i.e., $m"_t = 0$). A choppy ride would be when the driver is constantly accelerating and decelerating, such as beginning drivers tend to do.

If $\lambda = 0$, we don't care how choppy the ride is, and this leads to $m_t = x_t$, the data, which are not smooth. If $\lambda = \infty$, we insist on no acceleration or deceleration $m"_t = 0$; in this case, our drive must be at constant velocity, $m_t = c + vt$, and consequently very smooth. Thus, λ is seen as a trade-off between linear regression (completely smooth) and the data itself (no smoothness). The larger the value of λ, the smoother the fit.

In R, *the smoothing parameter is called* `spar` *and it is monotonically related to λ; Figure 2.19 shows smoothing spline fits on the SOI series using* `spar`*=0.5 to emphasize the El Nino cycle, and* `spar`*=1 to emphasize the trend.*

Figure 2.19 can be reproduced in R as follows.

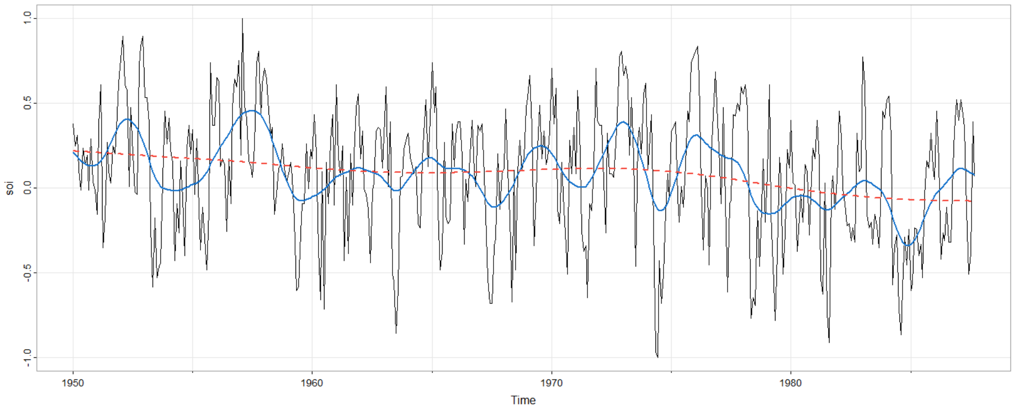

Figure 2.19: Smoothing splines fit to the SOI series

```
plot(soi)
lines(lowess(soi, f=.05), lwd=2, col=4) # El Nino cycle
lines(lowess(soi), lty=2, lwd=2, col=2) # trend (with default span)
```

Listing 2.19: The code to reproduce Figure 2.19

Example 2.19 (Smoothing One Series as a Function of Another) *In addition to smoothing time plots, smoothing techniques can be applied to smoothing a time series as a function of another time series. We have already seen this idea when we used lowess to visualize the nonlinear relationship between Recruitment and SOI at various lags. In this example, we smooth the scatterplot of two contemporaneously measured time series, mortality as a function of temperature. In Example 2.7, we discovered a nonlinear relationship between mortality and temperature. Continuing along these lines, Figure 2.20 show a scatterplot of mortality, M_t , and temperature, T_t, along with M_t smoothed as a function of T_t using lowess. Note that mortality increases at extreme temperatures, but in an asymmetric way; mortality is higher at colder temperatures than at hotter temperatures. The minimum mortality rate seems to occur at approximately* 83° *F.*

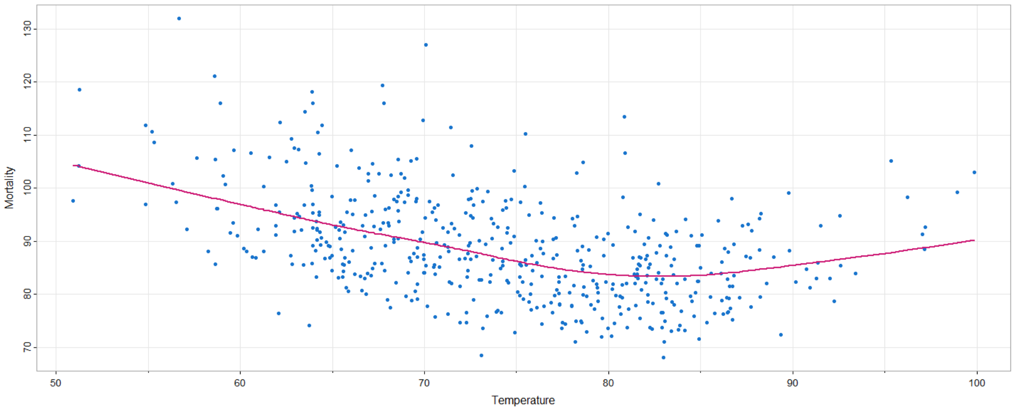

Figure 2.20: Smooth of mortality as a function of temperature using lowess

Figure 2.20 can be reproduced in R as follows.

```
tsplot(tempr, cmort, type="p", xlab="Temperature", ylab="Mortality",
    pch=20, col=4)
lines(lowess(tempr, cmort), col=6, lwd=2)
```

Listing 2.20: The code to reproduce Figure 2.20