

STAT 626: Outline of Lectures 10-11
Time Series Regression and EDA (§3.1, 3.2)

1. Review of Stationarity, Preview of TS Models
2. Example 3.1: Estimating the Linear Trend
3. A Quick Review of Multiple Regression

$$x = Z\beta + w,$$

LSE of β :

$$\widehat{\beta} = (Z'Z)^{-1}Z'x.$$

4. Tests, CIs and Variable Selection
5. AIC (Akaike Information Criterion), BIC
6. Example 3.5: Pollution, Temperature and Mortality (PTM) Data
7. Example 3.6: Regression with Lagged Variables
8. Example 3.7: Detrending and Differencing
9. The Backshift Operator B : $Bx_t = x_{t-1}$.

Regression and Forecasting:

PROBLEM (POPULATION INFORMATION): Given the value of a random variable X , **find β to minimize the mean-square error (MSE) of predicting Y by $\hat{Y} = \beta X$:**

$$\text{MSE}(\beta) = E(Y - \beta X)^2.$$

SOLUTION: The minimizer satisfies the *normal equation*:

$$\text{Var}(X) \quad \hat{\beta} = \text{Cov}(X, Y)$$

or

$$\hat{\beta} = \frac{\text{Cov}(X, Y)}{\text{Var}(X)},$$

and

$$\text{MSE}(\hat{\beta}) = E(Y - \hat{Y})^2 = (1 - \rho^2)\text{Var}(Y).$$

Time Series Prediction: Given the time series data x_1, \dots, x_n from a zero-mean stationary process $\{x_t\}$ with **known** autocovariance function, $\gamma(h)$ find the forecast value of the process at the next time point, x_{n+1} . More precisely, find ϕ_{n1}, \dots, ϕ_n to minimize the MSE of forecasting:

$$E(x_{n+1} - \phi_{n1}x_n - \dots - \phi_{nn}x_1)^2.$$

Review of Stationarity, Preview of TS Models (Chapter 4)

1. **Linear Processes:** $x_t = \mu + \sum_{j=-\infty}^{+\infty} \psi_j w_{t-j}$ is stationary with the *autocovariance function*

$$\gamma(h) = \sigma_w^2 \sum_{j=-\infty}^{\infty} \psi_{j+h} \psi_j.$$

2. **MA(q) Models:** $x_t = w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q}$, $\theta_q \neq 0$, is stationary, its autocovariance is zero at lags greater than q .
3. **Autoregressive Models of order p or AR(p) Models:**

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t, \quad \phi_p \neq 0.$$

QUESTION: Is a time series $\{x_t\}$ defined via an AR(p) model always stationary? If so, what is its autocovariance function?

To get a feel for the answer consider the AR(1):

$$x_t = \phi x_{t-1} + w_t,$$

what happens when $\phi = 1$?

4. **The Backshift Operator B :** $Bx_t = x_{t-1}$.

5. **MA(q) and B :**

$$x_t = w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q} = (1 + \theta_1 B + \dots + \theta_q B^q) w_t = \theta(B) w_t.$$

6. **AR(p) and B :**

$$x_t - \phi_1 x_{t-1} - \dots - \phi_p x_{t-p} = w_t, \quad (1 - \phi_1 B - \dots - \phi_p B^p) x_t = \phi(B) x_t = w_t.$$

7. **The ROOTS of the polynomial equation**

$$\phi(B) = 0,$$

hold the key to the question of stationarity of the solutions of AR models.

Chapter 3

Time Series Regression and EDA

3.1 Ordinary Least Squares for Time Series

We first consider the problem where a time series, say, x_t , for $t = 1, \dots, n$, is possibly being influenced by a collection of fixed series, say, $z_{t1}, z_{t2}, \dots, z_{tq}$. The data collection with $q = 3$ exogenous variables is as follows:

Time	Dependent Variable	Independent Variables		
1	x_1	z_{11}	z_{12}	z_{13}
2	x_2	z_{21}	z_{22}	z_{23}
:	:	:	:	:
n	x_n	z_{n1}	z_{n2}	z_{n3}

We express the general relation through the *linear regression model*

$$x_t = \beta_0 + \beta_1 z_{t1} + \beta_2 z_{t2} + \cdots + \beta_q z_{tq} + w_t, \quad (3.1)$$

where $\beta_0, \beta_1, \dots, \beta_q$ are unknown fixed regression coefficients, and $\{w_t\}$ is white normal noise with variance σ_w^2 ; we will relax this assumption later.

Example 3.1. Estimating the Linear Trend of a Commodity

Consider the monthly export price of Norwegian salmon per kilogram from September 2003 to June 2017 shown in Figure 3.1. There is an obvious upward trend in the series, and we might use simple linear regression to estimate that trend by fitting the model,

$$x_t = \beta_0 + \beta_1 z_t + w_t, \quad z_t = 2003 \frac{8}{12}, 2004 \frac{8}{12}, \dots, 2017 \frac{5}{12}.$$

This is in the form of the regression model (3.1) with $q = 1$. The data x_t are in `salmon` and z_t is month, with values in `time(salmon)`. Our assumption that the error, w_t , is white noise is probably not true, but we will assume it is true for now. The problem of autocorrelated errors will be discussed in detail in Section 5.4.

In ordinary least squares (OLS), we minimize the error sum of squares

$$S = \sum_{t=1}^n w_t^2 = \sum_{t=1}^n (x_t - [\beta_0 + \beta_1 z_t])^2$$

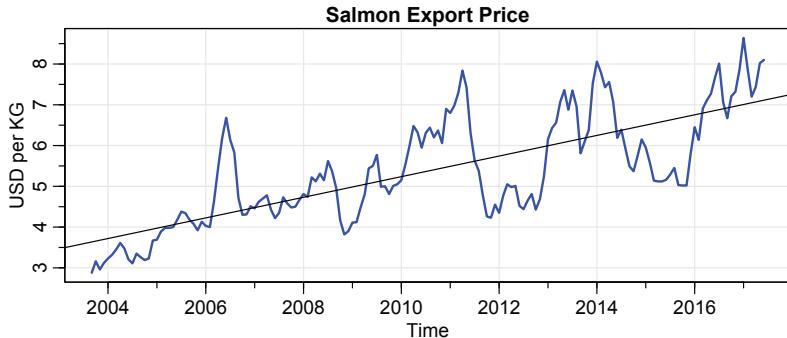


Figure 3.1 *The monthly export price of Norwegian salmon per kilogram from September 2003 to June 2017, with fitted linear trend line.*

with respect to β_i for $i = 0, 1$. In this case we can use simple calculus to evaluate $\partial S / \partial \beta_i = 0$ for $i = 0, 1$, to obtain two equations to solve for the β s. The OLS estimates of the coefficients are explicit and given by

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

where $\bar{x} = \sum_t x_t / n$ and $\bar{z} = \sum_t z_t / n$ are the respective sample means.

Using R, we obtained the estimated slope coefficient of $\hat{\beta}_1 = .25$ (with a standard error of .02) yielding a highly significant estimated increase of about 25 cents *per year*.¹ Finally, Figure 3.1 shows the data with the estimated trend line superimposed. To perform this analysis in R, use the following commands:

```
summary(fit <- lm(salmon~time(salmon), na.action=NULL))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -503.08947   34.44164  -14.61  <2e-16
time(salmon)   0.25290    0.01713   14.76  <2e-16
---
Residual standard error: 0.8814 on 164 degrees of freedom
Multiple R-squared:  0.5706,    Adjusted R-squared:  0.568
F-statistic: 217.9 on 1 and 164 DF,  p-value: < 2.2e-16
tsplot(salmon, col=4, ylab="USD per KG", main="Salmon Export Price")
abline(fit)
```

◇

Simple linear regression extends to multiple linear regression in a fairly straightforward manner. As in the previous example, OLS estimation minimizes the error sum of squares

$$S = \sum_{t=1}^n w_t^2 = \sum_{t=1}^n (x_t - [\beta_0 + \beta_1 z_{t1} + \beta_2 z_{t2} + \cdots + \beta_q z_{tq}])^2, \quad (3.2)$$

¹The unit of time here is one year, $z_t - z_{t-12} = 1$. Thus $\hat{x}_t - \hat{x}_{t-12} = \hat{\beta}_1(z_t - z_{t-12}) = \hat{\beta}_1$.

with respect to $\beta_0, \beta_1, \dots, \beta_q$. This minimization can be accomplished by solving $\partial S / \partial \beta_i = 0$ for $i = 0, 1, \dots, q$, which yields $q + 1$ equations with $q + 1$ unknowns. These equations are typically called the *normal equations*. The minimized error sum of squares (3.2), denoted SSE , can be written as

$$SSE = \sum_{t=1}^n (x_t - \hat{x}_t)^2, \quad (3.3)$$

where

$$\hat{x}_t = \hat{\beta}_0 + \hat{\beta}_1 z_{t1} + \hat{\beta}_2 z_{t2} + \cdots + \hat{\beta}_q z_{tq},$$

and $\hat{\beta}_i$ denotes the OLS estimate of β_i for $i = 0, 1, \dots, q$. The ordinary least squares estimators of the β s are unbiased and have the smallest variance within the class of linear unbiased estimators. An unbiased estimator for the variance σ_w^2 is

$$s_w^2 = MSE = \frac{SSE}{n - (q + 1)}, \quad (3.4)$$

where MSE denotes the *mean squared error*. Because the errors are normal, if $se(\hat{\beta}_i)$ represents the estimated standard error of the estimate of β_i , then

$$t = \frac{(\hat{\beta}_i - \beta_i)}{se(\hat{\beta}_i)} \quad (3.5)$$

has the t -distribution with $n - (q + 1)$ degrees of freedom. This result is often used for individual tests of the null hypothesis $H_0: \beta_i = 0$ for $i = 1, \dots, q$.

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, $z_{t,1:r} = \{z_{t1}, z_{t2}, \dots, z_{tr}\}$ is influencing the dependent variable x_t . The reduced model is

$$x_t = \beta_0 + \beta_1 z_{t1} + \cdots + \beta_r z_{tr} + w_t \quad (3.6)$$

where $\beta_1, \beta_2, \dots, \beta_r$ are a subset of coefficients of the original q variables.

The null hypothesis in this case is $H_0: \beta_{r+1} = \cdots = \beta_q = 0$. We can test the reduced model (3.6) against the full model (3.1) by comparing the error sums of squares under the two models using the F -statistic

$$F = \frac{(SSE_r - SSE)/(q - r)}{SSE/(n - q - 1)} = \frac{MSR}{MSE}, \quad (3.7)$$

where SSE_r is the error sum of squares under the reduced model (3.6). Note that $SSE_r \geq SSE$ because the reduced model has fewer parameters. If $H_0: \beta_{r+1} = \cdots = \beta_q = 0$ 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, (3.7) has a central F -distribution with $q - r$ and $n - q - 1$ degrees of freedom when (3.6) is the correct model.

Table 3.1 *Analysis of Variance for Regression*

Source	df	Sum of Squares	Mean Square	F
$z_{t,r+1:q}$	$q - r$	$SSR = SSE_r - SSE$	$MSR = SSR / (q - r)$	$F = \frac{MSR}{MSE}$
Error	$n - (q + 1)$	SSE	$MSE = SSE / (n - q - 1)$	

These results are often summarized in an ANOVA table as given in [Table 3.1](#) for this particular case. The difference in the numerator is often called the regression sum of squares (SSR). The null hypothesis is rejected at level α if $F > F_{n-q-1}^{q-r}(\alpha)$, 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 $H_0: \beta_1 = \dots = \beta_q = 0$. In this case $r = 0$, and the model in [\(3.6\)](#) becomes

$$x_t = \beta_0 + w_t.$$

The residual sum of squares under this reduced model is

$$SSE_0 = \sum_{t=1}^n (x_t - \bar{x})^2, \quad (3.8)$$

and SSE_0 is often called the *adjusted total sum of squares*, or SST (i.e., $SST = SSE_0$). In this case,

$$SST = SSR + SSE,$$

and we may measure the proportion of variation accounted for by all the variables using

$$R^2 = \frac{SSR}{SST}. \quad (3.9)$$

The measure R^2 is called the *coefficient of determination*.

The techniques discussed in the previous paragraph can be used for model selection; e.g., stepwise regression. Another approach is based on *parsimony* (also called *Occam's razor*) where we try to find the most *accurate* model with the least amount of *complexity*. For regression models, this means that we find the model that has the best fit with the fewest number of parameters. You may have been introduced to parsimony and model choice via Mallows C_p in a course on regression.

To measure accuracy, we use the error sum of squares, $SSE = \sum_{t=1}^n (x_t - \hat{x}_t)^2$, because it measures how close the fitted values (\hat{x}_t) are to the actual data (x_t). In particular, for a normal regression model with k coefficients, consider the (maximum likelihood) estimator for the variance as

$$\hat{\sigma}_k^2 = \frac{SSE(k)}{n}, \quad (3.10)$$

where by $SSE(k)$, we mean the residual sum of squares under the model with k regression coefficients. The complexity of the model can be characterized by k , the number of parameters in the model. Akaike (1974) suggested balancing the accuracy of the fit against the number of parameters in the model.

Definition 3.2. Akaike's Information Criterion (AIC)

$$\text{AIC} = \log \hat{\sigma}_k^2 + \frac{n + 2k}{n}, \quad (3.11)$$

where $\hat{\sigma}_k^2$ is given by (3.10) and k is the number of parameters in the model.²

Thus, the parsimonious model will be an accurate one (with small error $\hat{\sigma}_k$) that is not overly complex (small k). Hence, the model yielding the minimum AIC specifies the best model.

The choice for the penalty term given by (3.11) is not the only one, and a considerable literature is available advocating different penalty terms. A corrected form, suggested by Sugiura (1978), and expanded by Hurvich and Tsai (1989), can be based on small-sample distributional results for the linear regression model. The corrected form is defined as follows.

Definition 3.3. AIC, Bias Corrected (AICc)

$$\text{AICc} = \log \hat{\sigma}_k^2 + \frac{n + k}{n - k - 2}, \quad (3.12)$$

where $\hat{\sigma}_k^2$ is given by (3.10), k is the number of parameters in the model.

We may also derive a penalty term based on Bayesian arguments, as in Schwarz (1978), which leads to the following.

Definition 3.4. Bayesian Information Criterion (BIC)

$$\text{BIC} = \log \hat{\sigma}_k^2 + \frac{k \log n}{n}, \quad (3.13)$$

using the same notation as in Definition 3.2.

BIC is also called the Schwarz Information Criterion (SIC). 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; see McQuarrie and Tsai (1998) for detailed comparisons.

Example 3.5. Pollution, Temperature, and Mortality

The data shown in Figure 3.2 are extracted series from a study by Shumway et al. (1988) of the possible effects of temperature and pollution on weekly mortality in Los Angeles County. Note 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.

Notice the inverse relationship between mortality and temperature; the mortality

²Formally, AIC is defined as $-2 \log L_k + 2k$ where L_k is the maximum value of the likelihood and k is the number of parameters in the model. For the normal regression problem, AIC can be reduced to the form given by (3.11). For comparison, BIC is defined as $-2 \log L_k + k \log n$, so complexity has a much larger penalty.

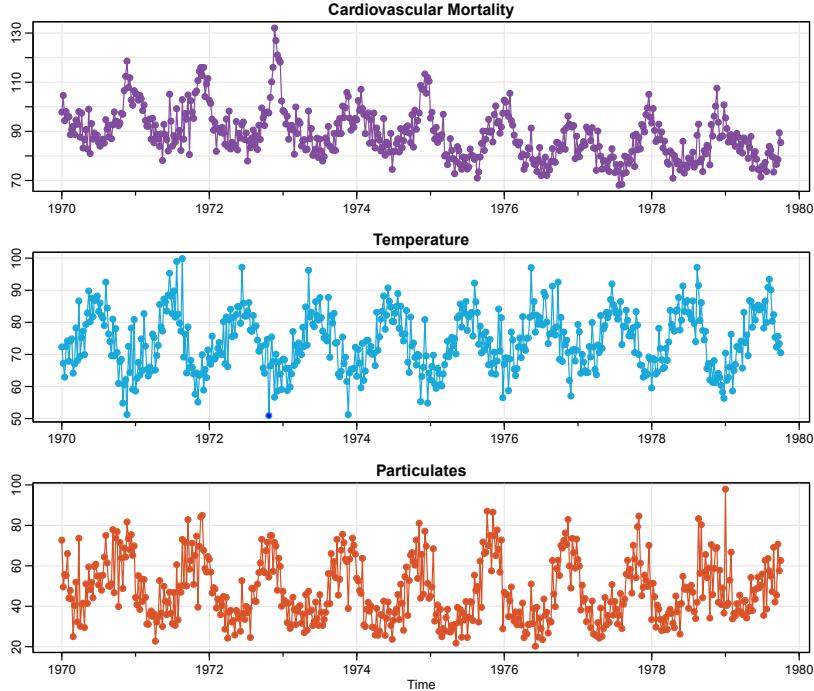


Figure 3.2 Average weekly cardiovascular mortality (top), temperature (middle), and particulate pollution (bottom) in Los Angeles County. There are 508 six-day smoothed averages obtained by filtering daily values over the 10-year period 1970–1979.

rate is higher for cooler temperatures. In addition, it appears that particulate pollution is directly related to mortality; the mortality rate increases for higher levels of pollution. These relationships can be better seen in Figure 3.3, where the data are plotted together. The time series plots were produced using the following R code:

```
##-- Figure 3.2 --##
culer = c(rgb(.66,.12,.85), rgb(.12,.66,.85), rgb(.85,.30,.12))
par(mfrow=c(3,1))
tsplot(cmort, main="Cardiovascular Mortality", col=culer[1],
       type="o", pch=19, ylab="")
tsplot(temp, main="Temperature", col=culer[2], type="o", pch=19,
       ylab="")
tsplot(part, main="Particulates", col=culer[3], type="o", pch=19,
       ylab="")
##-- Figure 3.3 --##
tsplot(cmort, main="", ylab="", ylim=c(20,130), col=culer[1])
lines(temp, col=culer[2])
lines(part, col=culer[3])
legend("topright", legend=c("Mortality", "Temperature", "Pollution"),
       lty=1, lwd=2, col=culer, bg="white")
```

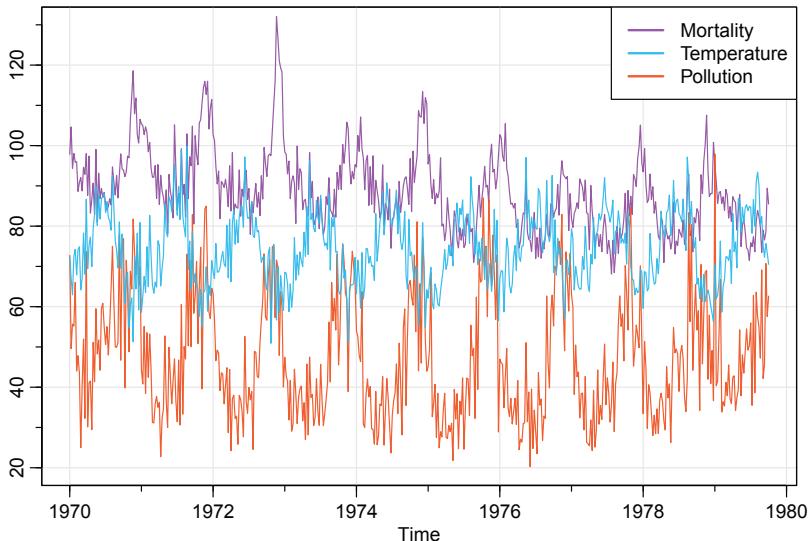


Figure 3.3 Mortality data on same plot.

To investigate these relationships further, a scatterplot matrix is shown in Figure 3.4 and indicates that cardiovascular mortality is linearly related to pollutant particulates, but is nonlinearly related to temperature. We note that the curvilinear shape of the temperature–mortality curve indicates that higher temperatures as well as lower temperatures are associated with increases in cardiovascular mortality. The scatterplot matrix shown in Figure 3.4 was generated in R as follows. The script `panel.cor` calculates the correlations between all the variables, and when called in `pairs`, inserts the corresponding correlation value.

```
panel.cor <- function(x, y, ...){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y), 2)
  text(0.5, 0.5, r, cex = 1.75)
}
pairs(cbind(Mortality=cmort, Temperature=tempr, Particulates=part),
      col="dodgerblue3", lower.panel=panel.cor)
```

It is important that temperature and particulate pollution are nearly uncorrelated. If these two independent variables were highly correlated (i.e., collinear), then it would be difficult to distinguish between the effects of each on mortality.

For ease, let M_t denote cardiovascular mortality, T_t denote temperature, and P_t denote the particulate levels. Based on the scatterplot matrix, it seems clear that both T_t and P_t should be in the model, but for demonstration purposes, we entertain four

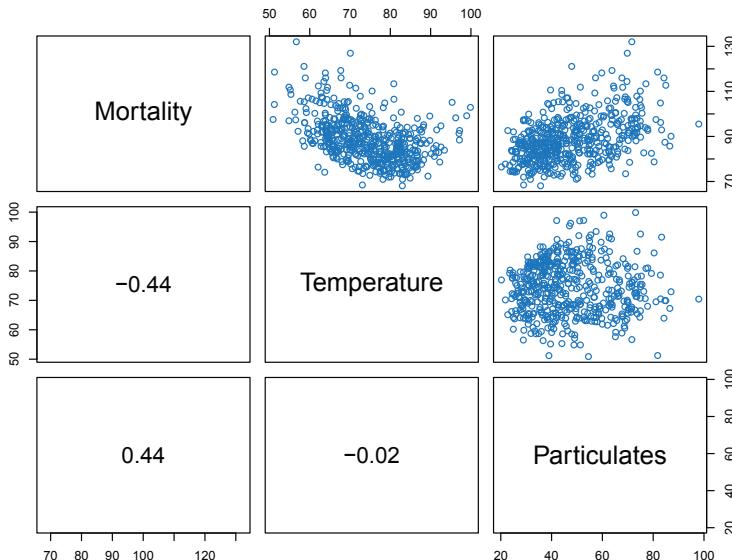


Figure 3.4 Scatterplot matrix showing relations between mortality, temperature, and pollution. The lower panels display the correlations.

models. They are

$$M_t = \beta_0 + \beta_1 t + w_t \quad (3.14)$$

$$M_t = \beta_0 + \beta_1 t + \beta_2(T_t - T.) + w_t \quad (3.15)$$

$$M_t = \beta_0 + \beta_1 t + \beta_2(T_t - T.) + \beta_3(T_t - T.)^2 + w_t \quad (3.16)$$

$$M_t = \beta_0 + \beta_1 t + \beta_2(T_t - T.) + \beta_3(T_t - T.)^2 + \beta_4 P_t + w_t \quad (3.17)$$

where we adjust temperature for its mean, $T. = 74.26$, to avoid collinearity problems. For this range of temperatures, T_t and T_t^2 are highly collinear, but $T_t - T.$ and $(T_t - T.)^2$ are not. To see this, run this simple R code:

```
par(mfrow = 2:1)
plot(temp, temp^2) # collinear
cor(temp, temp^2)
[1] 0.9972099
temp = temp - mean(temp)
plot(temp, temp^2) # not collinear
cor(temp, temp^2)
[1] 0.07617904
```

Note that (3.14) is a trend only model, (3.15) adds a linear temperature term, (3.16) adds a curvilinear temperature term and (3.17) adds a pollution term. We summarize some of the statistics given for this particular case in Table 3.2.

We note that each model does substantially better than the one before it and

Table 3.2 *Summary Statistics for Mortality Models*

Model	k	SSE	df	MSE	R^2	AIC	BIC
(3.14)	2	40,020	506	79.0	.21	5.38	5.40
(3.15)	3	31,413	505	62.2	.38	5.14	5.17
(3.16)	4	27,985	504	55.5	.45	5.03	5.07
(3.17)	5	20,508	503	40.8	.60	4.72	4.77

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 the residual sums of squares and (3.7). Hence, a model with only trend could be compared to the full model using $q = 4, r = 1, n = 508$, so

$$F_{3,503} = \frac{(40,020 - 20,508)/3}{20,508/503} = 160,$$

which exceeds $F_{3,503}(.001) = 5.51$. We obtain the best prediction model,

$$\hat{M}_t = 2831.5 - 1.396_{(.10)} \text{trend} - .472_{(.032)}(T_t - 74.26) + .023_{(.003)}(T_t - 74.26)^2 + .255_{(.019)}P_t,$$

for mortality, where the standard errors are given in parentheses.

As expected, a negative trend is present over time as well as a negative coefficient for adjusted temperature. 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 (of which there is a substantial amount), but we defer this question to Section 5.4 when we discuss regression with correlated errors.

Below is the R code to fit the final regression model (3.17), and compute the corresponding values of AIC and BIC.³ Our definitions differ from R by terms that do not change from model to model. In the example, we show how to obtain (3.11) and (3.13) from the R output. Finally, the use of `na.action` in `lm()` is to retain the time series attributes for the residuals and fitted values.

```
temp = tempr - mean(tempr) # center temperature
temp2 = temp^2
trend = time(cmort)         # time is trend
fit = lm(cmort ~ trend + temp + temp2 + part, na.action=NULL)
summary(fit)                # regression results
summary(aov(fit))           # ANOVA table (compare to next line)
```

³The easiest way to extract AIC and BIC from an `lm()` run in R is to use the command `AIC()` or `BIC()`.

```
summary(aov(lm(cmort~cbind(trend, temp, temp2, part)))) # Table 3.1
num = length(cmort)                                # sample size
AIC(fit)/num - log(2*pi)                          # AIC
BIC(fit)/num - log(2*pi)                          # BIC
```

Finally, in Figure 3.3 it appears that mortality may peak a few weeks after pollution peaks. In this case, we may want to include a lagged value of pollution into the model. This concept is explored further in Problem 3.2. ◇

It is possible to include lagged variables in time series regression models with some care. We will continue to discuss this type of problem throughout the text. To first address this problem, we consider a simple example of lagged regression.

Example 3.6. Regression with Lagged Variables

In Example 2.32, we discovered that the Southern Oscillation Index (SOI) measured at time $t - 6$ months is associated with the Recruitment series at time t , indicating that the SOI leads the Recruitment series by six months. Although there is strong evidence that the relationship is NOT linear (this is discussed further in Example 3.13), *for demonstration purposes only*, we consider the following regression,

$$R_t = \beta_0 + \beta_1 S_{t-6} + w_t, \quad (3.18)$$

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

$$\hat{R}_t = 65.79 - 44.28_{(2.78)} S_{t-6} \quad (3.19)$$

with $\hat{\sigma}_w = 22.5$ on 445 degrees of freedom. Of course, it is essential to check the model assumptions before making any conclusions, but we defer most of this discussion until later. We do, however, display a time series plot of the regression residuals in Figure 3.5, which clearly demonstrates a pattern and contradicts the assumption that w_t is white noise.

Performing lagged regression in R is a little difficult because the series must be aligned prior to running the regression. The easiest way to do this is to create an object (that we call `fish`) using `ts.intersect`, which aligns the lagged series.

```
fish = ts.intersect(rec, soiL6=lag(soi,-6))
summary(fit1 <- lm(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
tsplot(resid(fit1), col=4)  # residual time plot
```

The headache of aligning the lagged series can be avoided by using the R package `dynlm`. The setup is easier and the results are identical.

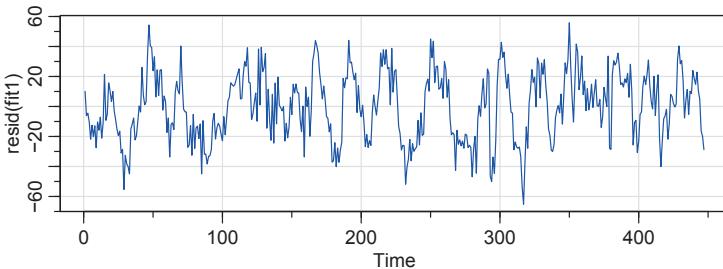


Figure 3.5 Residual plot for Example 3.6.

```
library(dynlm)
summary(fit2 <- dynlm(rec ~ L(soi, 6)))
```

◇

3.2 Exploratory Data Analysis

For time series, 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 correlation between contiguous time points if the correlation were different for every pair of observations. Hence, it is crucial that a time series satisfies the conditions of stationarity stated in Definition 2.13 for at least some reasonable stretch of time. Often, this is not the case, and in this section we discuss some methods for coercing nonstationary data to stationarity.

A number of our examples came from clearly nonstationary series. The Johnson & Johnson series in Figure 1.1 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, for example, clearly increases as one progresses over the length of the series. Also, the global temperature series shown in Figure 1.2 contain clear evidence of an increasing trend over time.

Perhaps the easiest form of nonstationarity to work with is the *trend stationary* model wherein the process has stationary behavior around a trend. We may write this type of model as

$$x_t = \mu_t + y_t \quad (3.20)$$

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 , as we shall see in numerous examples. 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

$$\hat{y}_t = x_t - \hat{\mu}_t. \quad (3.21)$$

Consider the following example.

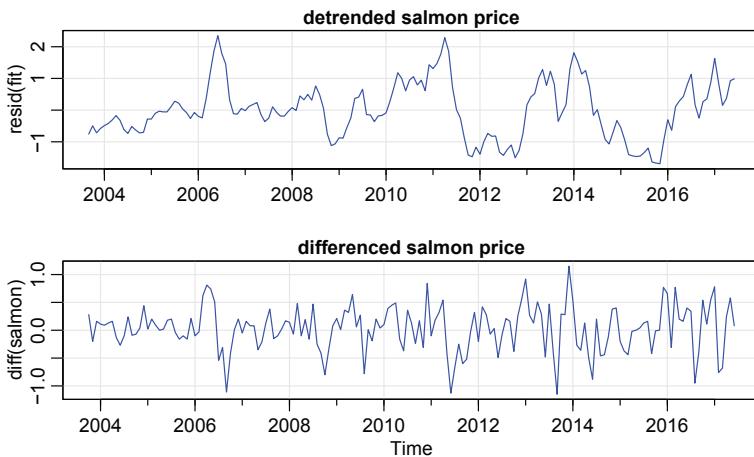


Figure 3.6 Detrended (top) and differenced (bottom) salmon price series. The original data are shown in [Figure 3.1](#).

Example 3.7. Detrending a Commodity

Let x_t represent the salmon price data presented in [Example 3.1](#). Here we suppose the model is of the form of (3.20),

$$x_t = \mu_t + y_t,$$

where, as we suggested in [Example 3.1](#), a straight line might be useful for detrending the data; i.e.,

$$\mu_t = \beta_0 + \beta_1 t,$$

where the time indices are the values in `time(salmon)`. In that example, we estimated the trend using ordinary least squares and found

$$\hat{\mu}_t = -503 + .25 t.$$

[Figure 3.1](#) (top) 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 series⁴

$$\hat{y}_t = x_t + 503 - .25 t.$$

The top graph of [Figure 3.6](#) shows the detrended series. [Figure 3.7](#) shows the ACF of the detrended data (top panel). ◇

In [Example 1.10](#) we saw that a random walk might also be a good model for trend.

⁴Because the error term, y_t , is not assumed to be white noise, the reader may feel that weighted least squares is called for in this case. The problem is, we do not know the behavior of y_t and that is precisely what we are trying to assess at this stage. A notable result by Grenander and Rosenblatt (2008, Ch 7) is that under mild conditions on y_t , for polynomial regression or periodic regression, ordinary least squares is equivalent to weighted least squares with regard to efficiency for large samples.

That is, rather than modeling trend as fixed (as in Example 3.7), we might model trend as a stochastic component using the random walk with drift model,

$$\mu_t = \delta + \mu_{t-1} + w_t, \quad (3.22)$$

where w_t is white noise and is independent of y_t . If the appropriate model is (3.20), then differencing the data, x_t , yields a stationary process; that is,

$$\begin{aligned} x_t - x_{t-1} &= (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) \\ &= \delta + w_t + y_t - y_{t-1}. \end{aligned} \quad (3.23)$$

It is easy to show $z_t = y_t - y_{t-1}$ is stationary using Property 2.7. That is, because y_t is stationary,

$$\begin{aligned} \gamma_z(h) &= \text{cov}(z_{t+h}, z_t) = \text{cov}(y_{t+h} - y_{t+h-1}, y_t - y_{t-1}) \\ &= 2\gamma_y(h) - \gamma_y(h+1) - \gamma_y(h-1) \end{aligned} \quad (3.24)$$

is independent of time; we leave it as an exercise (Problem 3.5) to show that $x_t - x_{t-1}$ in (3.23) is stationary.

 One advantage of differencing over detrending to remove trend is that no parameters are estimated in the differencing operation. One disadvantage, however, is that differencing does not yield an estimate of the stationary process y_t as can be seen in (3.23). If an estimate of y_t is essential, then detrending may be more appropriate. This would be the case, for example, if we were interested in the business cycle of commodities. The salmon prices appear to have a 3- to 4-year business cycle, which is known as the Kitchin cycle (Kitchin, 1923) and is seen in many commodity series.

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 3.7. That is, e.g., if $\mu_t = \beta_0 + \beta_1 t$ in the model (3.20), differencing the data produces stationarity (see Problem 3.4):

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

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

$$\nabla x_t = x_t - x_{t-1}. \quad (3.25)$$

As we have seen, the first difference eliminates a linear trend. A second difference, that is, the difference of (3.25), can eliminate a quadratic trend, and so on. In order to define higher differences, we need a variation in notation that we will use often in our discussion of ARIMA models in Chapter 5.

Definition 3.8. We define the **backshift operator** by

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

and extend it to powers $B^2x_t = B(Bx_t) = Bx_{t-1} = x_{t-2}$, and so on. Thus,

$$B^k x_t = x_{t-k}. \quad (3.26)$$

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*. In addition, it is clear that we may rewrite (3.25) as

$$\nabla x_t = (1 - B)x_t, \quad (3.27)$$

and we may extend the notion further. For example, the second difference becomes

$$\nabla^2 x_t = (1 - B)^2 x_t = (1 - 2B + B^2)x_t = x_t - 2x_{t-1} + x_{t-2} \quad (3.28)$$

by the linearity of the operator.

Definition 3.9. Differences of order d are defined as

$$\nabla^d = (1 - B)^d, \quad (3.29)$$

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.

The first difference (3.25) is an example of a *linear filter* applied to eliminate a trend. Other filters, formed by averaging values near x_t , can produce adjusted series that eliminate other kinds of unwanted fluctuations, as in Chapter 6. The differencing technique is an important component of the ARIMA model discussed in Chapter 5.

Example 3.10. Differencing a Commodity

The first difference of the salmon prices series, also shown in Figure 3.6, produces different results than removing trend by detrending via regression. For example, the Kitchin business cycle we observed in the detrended series is not obvious in the differenced series (although it is still there, which can be verified using Chapter 7 techniques).

The ACF of the differenced series is also shown in Figure 3.7. In this case, the difference series exhibits a strong annual cycle that was not evident in the original or detrended data. The R code to reproduce Figure 3.6 and Figure 3.7 is as follows.

```
fit = lm(salmon~time(salmon), na.action=NULL) # the regression
par(mfrow=c(2,1)) # plot transformed data
tsplot(resid(fit), main="detrended salmon price")
tsplot(diff(salmon), main="differenced salmon price")
par(mfrow=c(2,1)) # plot their ACFs
acf1(resid(fit), 48, main="detrended salmon price")
acf1(diff(salmon), 48, main="differenced salmon price")
```



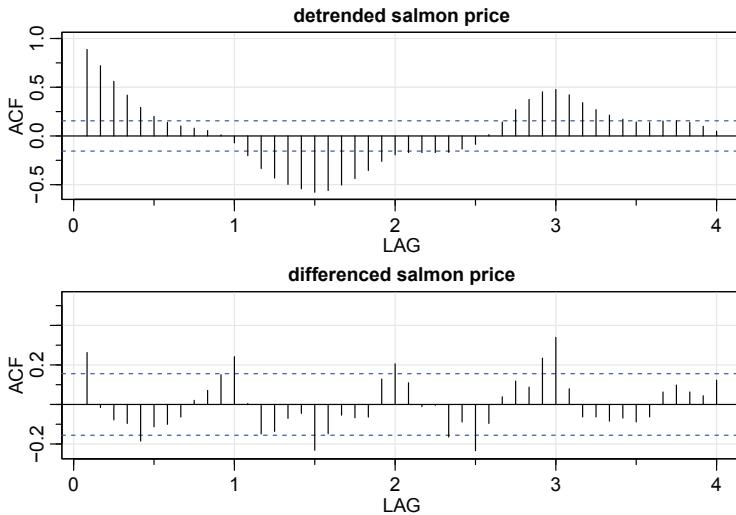


Figure 3.7 *Sample ACFs of the detrended (top) and of the differenced (bottom) salmon price series.*

Example 3.11. Differencing Global Temperature

The global temperature series shown in Figure 1.2 appears to behave more as a random walk than a trend stationary series. Hence, rather than detrend the data, it would be more appropriate to use differencing to coerce it into stationarity. The detrended data are shown in Figure 3.8 along with the corresponding sample ACF. In this case it appears that the differenced process shows minimal autocorrelation at lag 1, which may imply the global temperature series is nearly a random walk with drift.

It is interesting to note that if the series is a random walk with drift, the mean of the differenced series, which is an estimate of the drift, is about .014, or an increase of about one and a half degree centigrade per 100 years. If however, we restrict attention to the temperatures after 1980 when global temperature increase is evident (see Hansen and Lebedeff, 1987), the drift increases by more than twofold. The R code to reproduce Figure 3.8 is as follows.

```
par(mfrow=c(2,1))
tsplot(diff(gtemp_land), col=4, main="differenced global temperature")
mean(diff(gtemp_land))      # drift since 1880
[1] 0.0143
acf1(diff(gtemp_land))
mean(window(diff(gtemp_land), start=1980)) # drift since 1980
[1] 0.0329
```

◇

Sometimes heteroscedasticity is seen in time series data. A particularly useful transformation in this case is

$$y_t = \log x_t, \quad (3.30)$$

which tends to suppress larger fluctuations that occur over portions of the series where

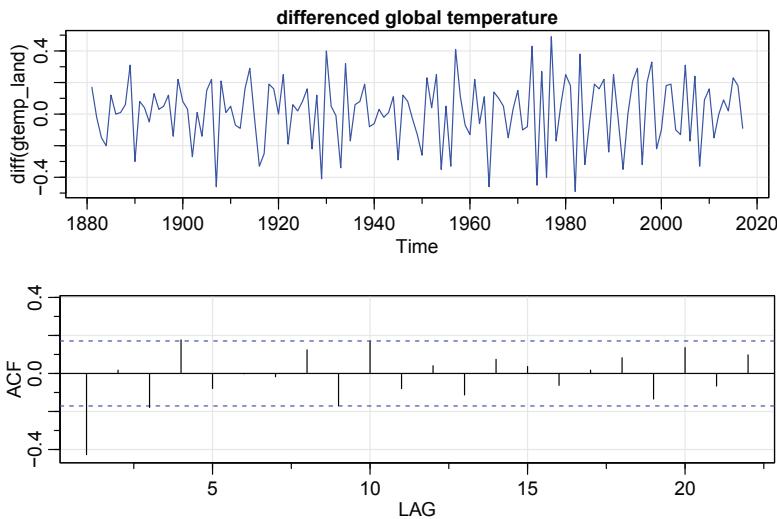


Figure 3.8 *Differenced global temperature series and its sample ACF.*

the underlying values are larger. Other possibilities are *power transformations* in the Box–Cox family of the form

$$y_t = \begin{cases} (x_t^\lambda - 1)/\lambda & \lambda \neq 0, \\ \log x_t & \lambda = 0. \end{cases} \quad (3.31)$$

Methods for choosing the power λ are available (see [Johnson and Wichern, 2002, §4.7](#)) but we do not pursue them here. Often, transformations are also used to improve the approximation to normality or to improve linearity in predicting the value of one series from another.

Example 3.12. Paleoclimatic Glacial Varves

Melting glaciers deposit yearly layers of sand and silt during the spring melting seasons, which can be reconstructed yearly over a period ranging from the time deglaciation began in New England (about 12,600 years ago) to the time it ended (about 6,000 years ago). Such sedimentary deposits, called *varves*, can be used as proxies for paleoclimatic parameters, such as temperature, because, in a warm year, more sand and silt are deposited from the receding glacier. The top of [Figure 3.9](#) shows the thicknesses of the yearly varves collected from one location in Massachusetts for 634 years, beginning 11,834 years ago. For further information, see [Shumway and Verosub \(1992\)](#).

Because the variation in thicknesses increases in proportion to the amount deposited, a logarithmic transformation could remove the nonstationarity observable in the variance as a function of time. [Figure 3.9](#) shows the original and the logged transformed varves, and it is clear that this improvement has occurred. Also plotted are the corresponding normal Q-Q plots. Recall that these plots are of the quantiles

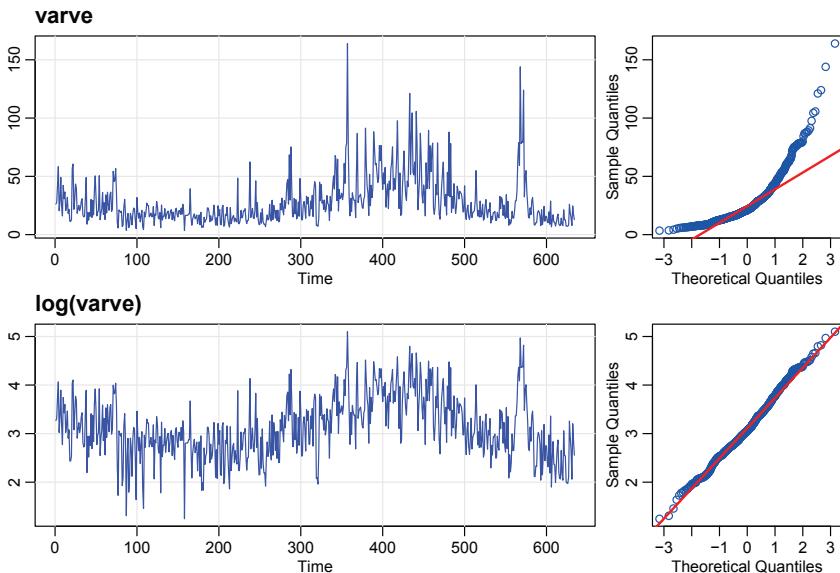


Figure 3.9 Glacial varve thicknesses (top) from Massachusetts for $n = 634$ years compared with log transformed thicknesses (bottom). The plots on the right-side are corresponding normal Q-Q plots.

of the data against the theoretical quantiles of the normal distribution. Normal data should fall approximately on the exhibited line of equality. In this case, we can argue that the approximation to normality is improved by the log transformation.

Figure 3.9 was generated in R as follows:

```
layout(matrix(1:4, 2), widths=c(2.5, 1))
par(mgp=c(1.6, .6, 0), mar=c(2, 2, .5, 0)+.5)
tsplot(varve, main="", ylab="", col=4, margin=0)
mtext("varve", side=3, line=.5, cex=1.2, font=2, adj=0)
tsplot(log(varve), main="", ylab="", col=4, margin=0)
mtext("log(varve)", side=3, line=.5, cex=1.2, font=2, adj=0)
qqnorm(varve, main="", col=4); qqline(varve, col=2, lwd=2)
qqnorm(log(varve), main="", col=4); qqline(log(varve), col=2, lwd=2) ◇
```

Next, we consider another preliminary data processing technique that is used for the purpose of visualizing the relations between series at different lags, namely the *lagplot*. When using the ACF, we are measuring the linear relation between lagged values of a time series. The restriction of this idea to linear predictability, however, may mask possible nonlinear relationships between future values, x_{t+h} , and current values, x_t . This idea extends to two series where one may be interested in examining lagplots of y_t versus x_{t-h} .

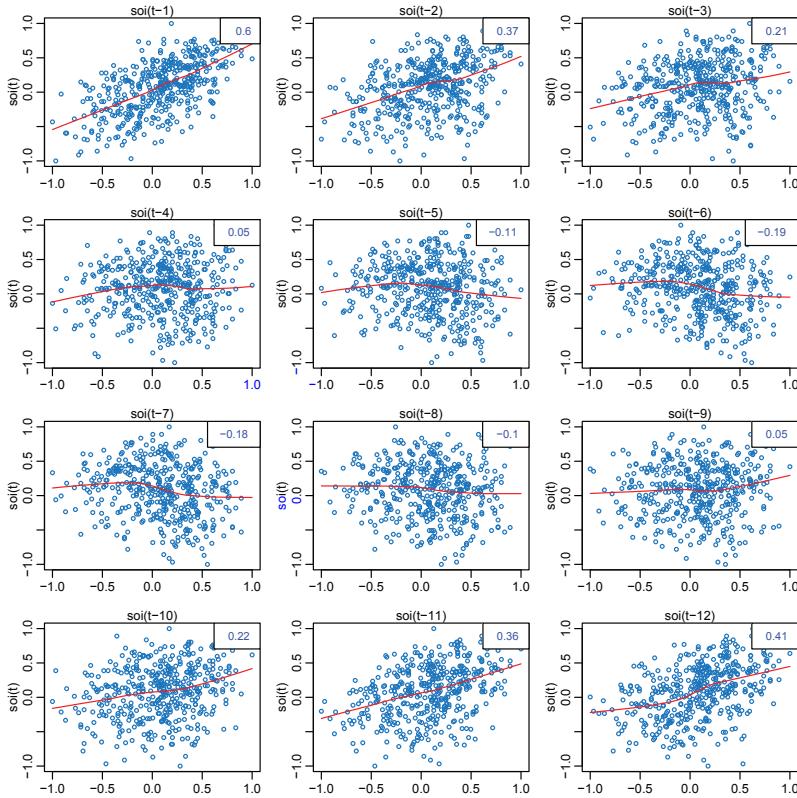


Figure 3.10 Lagplot relating current SOI values, S_t , to past SOI values, S_{t-h} , at lags $h = 1, 2, \dots, 12$. The values in the upper right corner are the sample autocorrelations and the lines are a lowess fit.

Example 3.13. Lagplots: SOI and Recruitment

Figure 3.10 displays a lagplot of the SOI, S_t , on the vertical axis plotted against S_{t-h} on the horizontal axis. The sample autocorrelations are displayed in the upper right-hand corner and superimposed on the lagplots are locally weighted scatterplot smoothing (lowess) lines that can be used to help discover any nonlinearities. We discuss smoothing in the next section, but for now, think of lowess as a method for fitting local regression.

In Figure 3.10, we notice that the local fits are approximately linear so that the sample autocorrelations are meaningful. Also, we see strong positive linear relations at lags $h = 1, 2, 11, 12$, that is, between S_t and $S_{t-1}, S_{t-2}, S_{t-11}, S_{t-12}$, and a negative linear relation at lags $h = 6, 7$.

Similarly, we might want to look at values of one series, say Recruitment, denoted R_t plotted against another series at various lags, say the SOI, S_{t-h} , to look for possible nonlinear relations between the two series. Because, for example, we might wish to

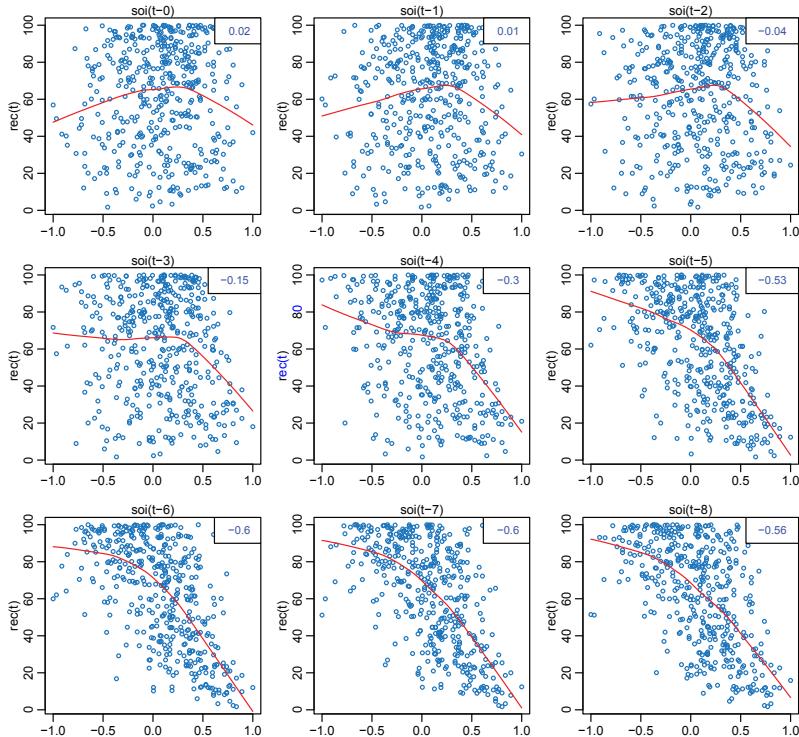


Figure 3.11 Lagplot of the Recruitment series, R_t , on the vertical axis plotted against the SOI series, S_{t-h} , on the horizontal axis at lags $h = 0, 1, \dots, 8$. The values in the upper right corner are the sample cross-correlations and the lines are a lowess fit.

predict the Recruitment series, R_t , from current or past values of the SOI series, S_{t-h} , for $h = 0, 1, 2, \dots$ it would be worthwhile to examine the scatterplot matrix. Figure 3.11 shows the lagplot of the Recruitment series R_t on the vertical axis plotted against the SOI index S_{t-h} on the horizontal axis. In addition, the figure exhibits the sample cross-correlations as well as lowess fits.

Figure 3.11 shows a fairly strong nonlinear relationship between Recruitment, R_t , and the SOI series at $S_{t-5}, S_{t-6}, S_{t-7}, S_{t-8}$, indicating the SOI series tends to lead the Recruitment series and the coefficients are negative, implying that increases in the SOI lead to decreases in the Recruitment. The nonlinearity observed in the lagplots (with the help of the superimposed lowess fits) indicate that the behavior between Recruitment and the SOI is different for positive values of SOI than for negative values of SOI.

The R code for this example is

```
lag1.plot(soi, 12, col="dodgerblue3")      # Figure 3.10
lag2.plot(soi, rec, 8, col="dodgerblue3")    # Figure 3.11
```



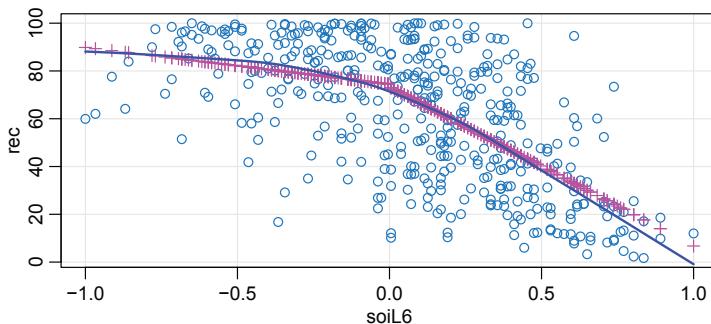


Figure 3.12 Display for Example 3.14: Plot of Recruitment (R_t) vs. SOI lagged 6 months (S_{t-6}) with the fitted values of the regression as points (+) and a lowess fit (—).

Example 3.14. Regression with Lagged Variables (cont.)

In Example 3.6 we regressed Recruitment on lagged SOI,

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

However, in Example 3.13, we saw 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

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

The result of the fit is given in the R code below. We have loaded `zoo` to ease the pain of working with lagged variables in R. Figure 3.12 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 in Problem 5.16.

```
library(zoo) # zoo allows easy use of the variable names
dummy = ifelse(soi<0, 0, 1)
fish = as.zoo(ts.intersect(rec, soiL6=lag(soi,-6), dL6=lag(dummy,-6)))
summary(fit <- lm(rec~ soiL6*dL6, data=fish, na.action=NULL))
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  

F-statistic: 99.43 on 3 and 443 DF, p-value: < 2.2e-16  

plot(fish$soiL6, fish$rec, panel.first=Grid(), col="dodgerblue3")  

points(fish$soiL6, fitted(fit), pch=3, col=6)  

lines(lowess(fish$soiL6, fish$rec), col=4, lwd=2)  

tsplot(resid(fit))    # not shown, but looks like Figure 3.5  

acf1(resid(fit))      # and obviously not noise

```

◊

As a final exploratory tool, we discuss assessing periodic behavior in time series data using regression analysis; this material may be thought of as an introduction to *spectral analysis*, which we discuss in detail in [Chapter 6](#). In [Example 1.11](#), 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 data from the pollution study example shown in [Figure 3.2](#) exhibit strong yearly cycles. Also, the Johnson & Johnson data shown in [Figure 1.1](#) make one cycle every year (four quarters) on top of an increasing trend and the speech data in [Figure 1.2](#) is highly repetitive. The monthly SOI and Recruitment series in [Figure 1.7](#) show strong yearly cycles, but hidden in the series are clues to the El Niño cycle.

Example 3.15. Using Regression to Discover a Signal in Noise

In [Example 1.11](#), we generated $n = 500$ observations from the model

$$x_t = A \cos(2\pi\omega t + \phi) + w_t, \quad (3.32)$$

where $\omega = 1/50$, $A = 2$, $\phi = .6\pi$, and $\sigma_w = 5$; the data are shown on the bottom panel of [Figure 1.11](#). 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 (3.32) in a nonlinear way, so we use a trigonometric identity (see [Section C.5](#)) and write

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

where $\beta_1 = A \cos(\phi)$ and $\beta_2 = -A \sin(\phi)$.

Now the model (3.32) can be written in the usual linear regression form given by (no intercept term is needed here)

$$x_t = \beta_1 \cos(2\pi t/50) + \beta_2 \sin(2\pi t/50) + w_t. \quad (3.33)$$

Using linear regression, we find $\hat{\beta}_1 = -.74_{(.33)}$, $\hat{\beta}_2 = -1.99_{(.33)}$ 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 $\beta_1 = 2 \cos(.6\pi) = -.62$, and $\beta_2 = -2 \sin(.6\pi) = -1.90$. It is clear that we are able to detect the signal in the noise using regression, even though the signal-to-noise ratio is small. The top of [Figure 3.13](#) shows the data generated by (3.32); it is hard to discern the signal and the data look like noise. However, the bottom of the figure shows the same data, but with the fitted line superimposed. It is now easy to see the signal through the noise.

To reproduce the analysis and [Figure 3.13](#) in R, use the following:

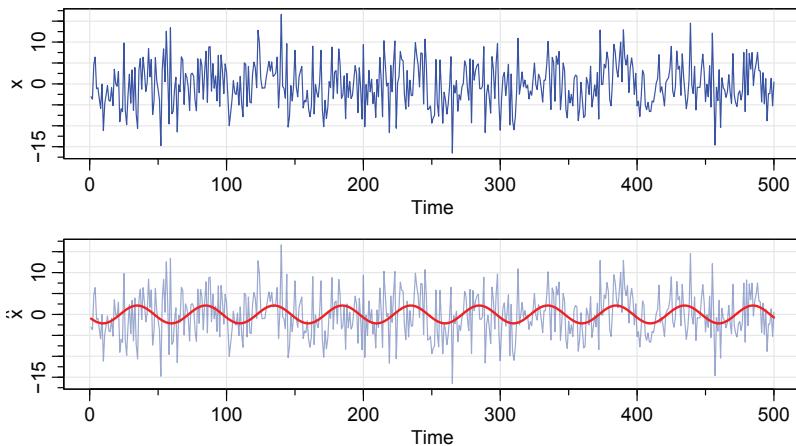


Figure 3.13 Data generated by (3.32) [top] and the fitted line superimposed on the data [bottom].

```
set.seed(90210)                      # 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
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
z1   -0.7442      0.3274  -2.273  0.0235
z2   -1.9949      0.3274 -6.093 2.23e-09
Residual standard error: 5.177 on 498 degrees of freedom
par(mfrow=c(2,1))
tsplot(x, col=4)
tsplot(x, ylab=expression(hat(x)), col=rgb(0,0,1,.5))
lines(fitted(fit), col=2, lwd=2)
```

◇

3.3 Smoothing Time Series

In Example 1.8, we introduced the concept of smoothing a time series using a moving average. This method is useful for discovering certain traits in a time series, such as long-term trend and seasonal components (see Section 6.3 for details). In particular, if x_t represents the observations, then

$$m_t = \sum_{j=-k}^k a_j x_{t-j}, \quad (3.34)$$

where $a_j = a_{-j} \geq 0$ and $\sum_{j=-k}^k a_j = 1$ is a symmetric moving average.

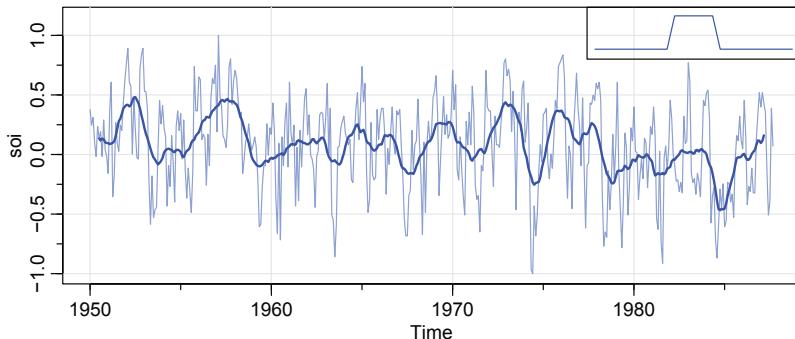


Figure 3.14 The SOI series smoothed using (3.34) with $k = 6$ (and half-weights at the ends). The insert shows the shape of the moving average (“boxcar”) kernel [not drawn to scale] described in (3.36).

Example 3.16. Moving Average Smoother

For example, Figure 3.14 shows the monthly SOI series discussed in Example 1.4 smoothed using (3.34) with $k = 6$ and weights $a_0 = a_{\pm 1} = \dots = a_{\pm 5} = 1/12$, and $a_{\pm 6} = 1/24$. This particular method removes (filters out) the obvious annual temperature cycle and helps emphasize the El Niño cycle. The reason half-weights are used at the ends is so the same month does not get included in the average twice. For example, if we center on a July ($j = 0$), then January ($j = -6$) of that year and January ($j = 6$) of the next year will be included in the smoother. Consequently, each January gets a half-weight, and so on.

To reproduce Figure 3.14 in R:

```
w = c(.5, rep(1,11), .5)/12
soif = filter(soi, sides=2, filter=w)
tsplot(soi, col=rgb(.5, .6, .85, .9), ylim=c(-1, 1.15))
lines(soif, lwd=2, col=4)
# insert
par(fig = c(.65, 1, .75, 1), new = TRUE)
w1 = c(rep(0,20), w, rep(0,20))
plot(w1, type="l", ylim = c(-.02,.1), xaxt="n", yaxt="n", ann=FALSE)
```



Although the moving average smoother does a good job in highlighting the El Niño 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 (3.34).

Example 3.17. Kernel Smoothing

Kernel smoothing is a moving average smoother that uses a weight function, or kernel, to average the observations. Figure 3.15 shows kernel smoothing of the SOI series, where m_t is now

$$m_t = \sum_{i=1}^n w_i(t) x_{t_i}, \quad (3.35)$$

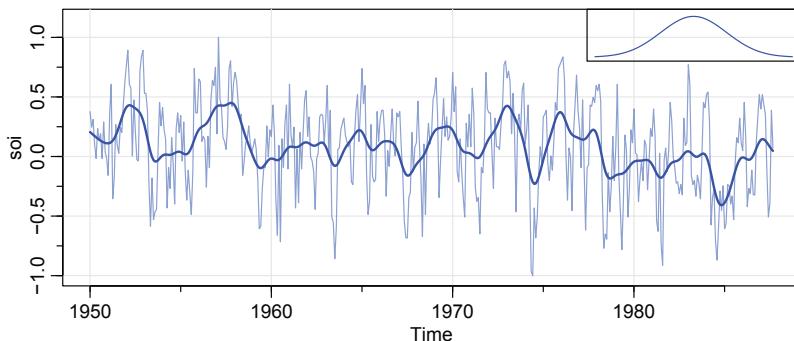


Figure 3.15 Kernel smoother of the SOI. The insert shows the shape of the normal kernel [not drawn to scale].

where

$$w_i(t) = K\left(\frac{t-t_i}{b}\right) / \sum_{k=1}^n K\left(\frac{t-t_k}{b}\right) \quad (3.36)$$

are the weights and $K(\cdot)$ is a kernel function. In this example, and typically, the normal kernel, $K(z) = \exp(-z^2/2)$, is used.

To implement this in R, we use the `ksmooth` function where a bandwidth can be chosen. Think of b as standard deviation, and the bigger the bandwidth, the smoother the result. In our case, we are smoothing over time, which is of the form $t/12$ for `soi`. In Figure 3.15, we used the value of $b = 1$ to correspond to approximately smoothing over about a year. The R code for this example is

```
tsplot(soi, col=rgb(0.5, 0.6, 0.85, .9), ylim=c(-1, 1.15))
lines(ksmooth(time(soi), soi, "normal", bandwidth=1), lwd=2, col=4)
# insert
par(fig = c(.65, 1, .75, 1), new = TRUE)
curve(dnorm(x), -3, 3, xaxt="n", yaxt="n", ann=FALSE, col=4)
```

We note that if the unit of time for SOI were months, then an equivalent smoother would use a bandwidth of 12:

```
SOI = ts(soi, freq=1)
tsplot(SOI) # the time scale matters (not shown)
lines(ksmooth(time(SOI), SOI, "normal", bandwidth=12), lwd=2, col=4) ◇
```

Example 3.18. Lowess

Another approach to smoothing is based on k -nearest neighbor regression, wherein, for $k < n$, one uses only the data $\{x_{t-k/2}, \dots, x_t, \dots, x_{t+k/2}\}$ 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 3.16 shows smoothing of SOI using the R function `lowess` (see Cleveland, 1979). 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

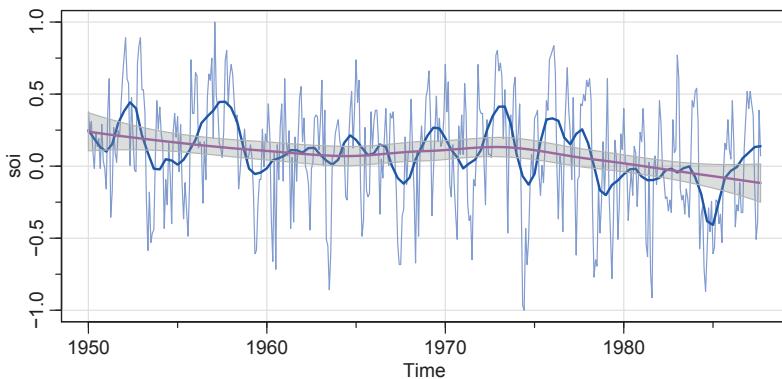


Figure 3.16 *Locally weighted scatterplot smoothers of the SOI series. The El Niño cycle is estimated using lowess and the trend with confidence intervals is estimated using loess.*

the smoothed values m_t . The larger the fraction of nearest neighbors included, the smoother the fit will be. In Figure 3.16, one smoother uses 5% of the data to obtain an estimate of the El Niño 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 the R function `loess` with the default smoother span of `f=2/3` of the data. The script `loess` is similar to `lowess`. A major difference for us is that the former strips the time series attributes whereas the latter does not, but the `loess` script allows the calculation of confidence intervals. Figure 3.16 can be reproduced in R as follows. We have commented out the trend estimate using `lowess`.

```
tsplot(soi, col=rgb(0.5, 0.6, 0.85, .9))
lines(lowess(soi, f=.05), lwd=2, col=4)      # El Niño cycle
# lines(lowess(soi), lty=2, lwd=2, col=2) # trend (with default span)
##-- trend with CIs using loess --#
lo = predict(loess(soi~ time(soi)), se=TRUE)
trnd = ts(lo$fit, start=1950, freq=12)       # put back ts attributes
lines(trnd, col=6, lwd=2)
L = trnd - qt(.975, lo$df)*lo$se
U = trnd + qt(.975, lo$df)*lo$se
xx = c(time(soi), rev(time(soi)))
yy = c(L, rev(U))
polygon(xx, yy, border=8, col=gray(.6, alpha=.4))
```



Example 3.19. Smoothing One Series as a Function of Another

Smoothing techniques can also be applied to smoothing a time series as a function of another time series. In Example 3.5, we discovered a nonlinear relationship between mortality and temperature. Figure 3.17 shows 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

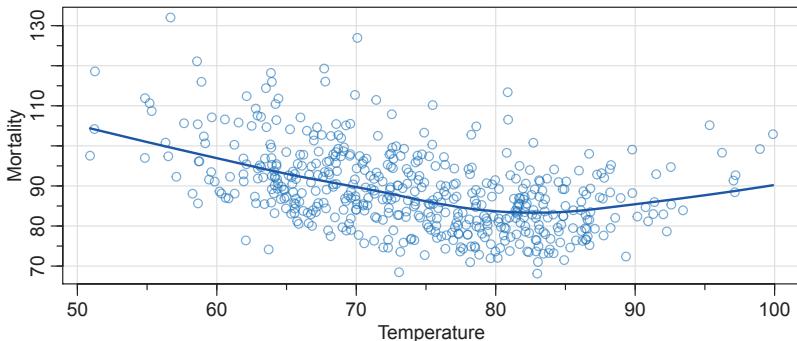


Figure 3.17 Smooth of mortality as a function of temperature using lowess.

mortality increases at extreme temperatures. The minimum mortality rate seems to occur at approximately 83° F. Figure 3.17 can be reproduced in R as follows.

```
plot(temp, cmort, xlab="Temperature", ylab="Mortality",
     col="dodgerblue3", panel.first=Grid())
lines(lowess(temp, cmort), col=4, lwd=2)
```

◇

Example 3.20. Classical Structural Modeling

A classical approach to time series analysis is to decompose data into components labeled trend (T_t), seasonal (S_t), irregular or noise (N_t). If we let x_t denote the data, we can then sometimes write

$$x_t = T_t + S_t + N_t.$$

Of course, not all time series data fit into such a paradigm and the decomposition may not be unique. Sometimes an additional cyclic component, say C_t , such as a business cycle is added to the model.

Figure 3.18 shows the result of the decomposition using loess on the quarterly occupancy rate of Hawaiian hotels from 2002 to 2016. R provides a few scripts to fit the decomposition. The script `decompose` uses moving averages as in Example 3.16. Another script, `stl`, uses loess to obtain each component and is similar to the approach used in Example 3.18. To use `stl`, the seasonal smoothing method must be specified. That is, specify either the character string "periodic" or the span of the loess window for seasonal extraction. The span should be odd and at least 7 (there is no default). By using a seasonal window, we are allowing $S_t \approx S_{t-4}$ rather than $S_t = S_{t-4}$, which is forced by specifying a periodic seasonal component.

Note that in Figure 3.18, the seasonal component is very regular showing a 2% to 4% gain in the first and third quarters, while showing a 2% to 4% loss in the second and fourth quarters. The trend component is perhaps more like a business cycle than what may be considered a trend. As previously implied, the components are not well defined and the decomposition is not unique; one person's trend may be another person's business cycle. The basic R code for this example is:

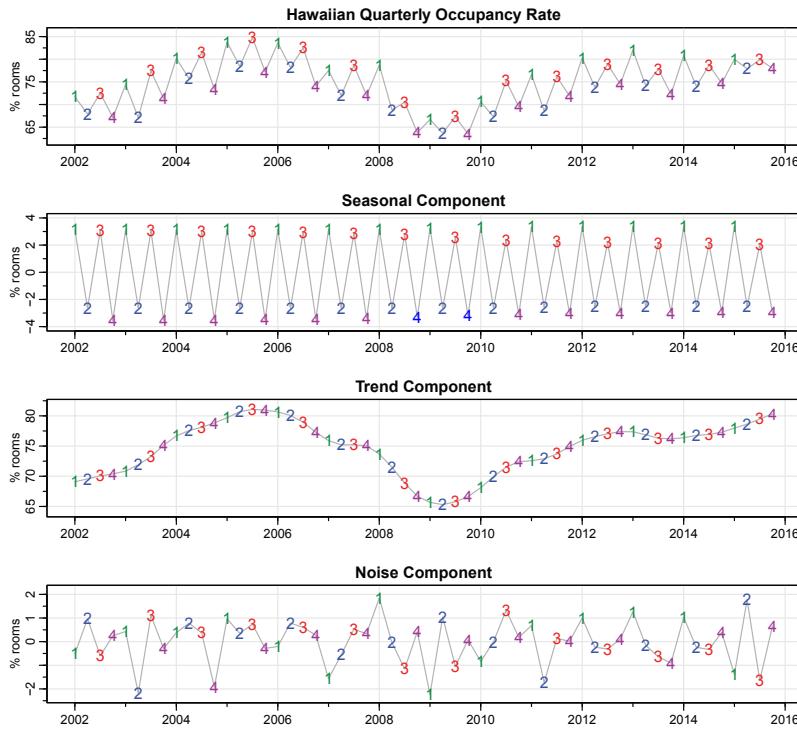


Figure 3.18 *Structural model of the Hawaiian quarterly occupancy rate.*

```
x = window(hor, start=2002)
plot(decompose(x))           # not shown
plot(stl(x, s.window="per")) # seasons are periodic - not shown
plot(stl(x, s.window=15))
```

However, a figure similar to Figure 3.18 can be generated as follows:

```
culer = c("cyan4", 4, 2, 6)
par(mfrow = c(4,1), cex.main=1)
x = window(hor, start=2002)
out = stl(x, s.window=15)$time.series
tsplot(x, main="Hawaiian Occupancy Rate", ylab="% rooms", col=gray(.7))
text(x, labels=1:4, col=culer, cex=1.25)
tsplot(out[,1], main="Seasonal", ylab="% rooms", col=gray(.7))
text(out[,1], labels=1:4, col=culer, cex=1.25)
tsplot(out[,2], main="Trend", ylab="% rooms", col=gray(.7))
text(out[,2], labels=1:4, col=culer, cex=1.25)
tsplot(out[,3], main="Noise", ylab="% rooms", col=gray(.7))
text(out[,3], labels=1:4, col=culer, cex=1.25)
```



Problems

3.1 (Structural Regression Model). For the Johnson & Johnson data, say y_t , shown in Figure 1.1, let $x_t = \log(y_t)$. In this problem, we are going to fit a special type of structural model, $x_t = T_t + S_t + N_t$ where T_t is a trend component, S_t is a seasonal component, and N_t is noise. In our case, time t is in quarters (1960.00, 1960.25, ...) so one unit of time is a year.

- (a) Fit the regression model

$$x_t = \underbrace{\beta t}_{\text{trend}} + \underbrace{\alpha_1 Q_1(t) + \alpha_2 Q_2(t) + \alpha_3 Q_3(t) + \alpha_4 Q_4(t)}_{\text{seasonal}} + \underbrace{w_t}_{\text{noise}}$$

where $Q_i(t) = 1$ if time t corresponds to quarter $i = 1, 2, 3, 4$, and zero otherwise. The $Q_i(t)$'s are called indicator variables. We will assume for now that w_t is a Gaussian white noise sequence. Hint: Detailed code is given in Appendix A, near the end of Section A.5.

- (b) If the model is correct, what is the estimated average annual increase in the logged earnings per share?
- (c) If the model is correct, does the average logged earnings rate increase or decrease from the third quarter to the fourth quarter? And, by what percentage does it increase or decrease?
- (d) What happens if you include an intercept term in the model in (a)? Explain why there was a problem.
- (e) Graph the data, x_t , and superimpose the fitted values, say \hat{x}_t , on the graph. Examine the residuals, $x_t - \hat{x}_t$, and state your conclusions. Does it appear that the model fits the data well (do the residuals look white)?

3.2. For the mortality data examined in Example 3.5:

- (a) Add another component to the regression in (3.17) that accounts for the particulate count four weeks prior; that is, add P_{t-4} to the regression in (3.17). State your conclusion.
- (b) Using AIC and BIC, is the model in (a) an improvement over the final model in Example 3.5?

3.3. In this problem, we explore the difference between a random walk and a trend stationary process.

- (a) Generate *four* series that are random walk with drift, (1.4), of length $n = 500$ with $\delta = .01$ and $\sigma_w = 1$. Call the data x_t for $t = 1, \dots, 500$. Fit the regression $x_t = \beta t + w_t$ using least squares. Plot the data, the true mean function (i.e., $\mu_t = .01 t$) and the fitted line, $\hat{x}_t = \hat{\beta} t$, on the same graph.
- (b) Generate *four* series of length $n = 500$ that are linear trend plus noise, say $y_t = .01 t + w_t$, where t and w_t are as in part (a). Fit the regression $y_t = \beta t + w_t$

using least squares. Plot the data, the true mean function (i.e., $\mu_t = .01 t$) and the fitted line, $\hat{y}_t = \hat{\beta} t$, on the same graph.

- (c) Comment on the differences between the results of part (a) and part (b).

3.4. Consider a process consisting of a linear trend with an additive noise term consisting of independent random variables w_t with zero means and variances σ_w^2 , that is,

$$x_t = \beta_0 + \beta_1 t + w_t,$$

where β_0, β_1 are fixed constants.

- (a) Prove x_t is nonstationary.
- (b) Prove that the first difference series $\nabla x_t = x_t - x_{t-1}$ is stationary by finding its mean and autocovariance function.
- (c) Repeat part (b) if w_t is replaced by a general stationary process, say y_t , with mean function μ_y and autocovariance function $\gamma_y(h)$.

3.5. Show (3.23) is stationary.

3.6. The glacial varve record plotted in Figure 3.9 exhibits some nonstationarity that can be improved by transforming to logarithms and some additional nonstationarity that can be corrected by differencing the logarithms.

- (a) Argue that the glacial varves series, say x_t , exhibits heteroscedasticity by computing the sample variance over the first half and the second half of the data. Argue that the transformation $y_t = \log x_t$ stabilizes the variance over the series. Plot the histograms of x_t and y_t to see whether the approximation to normality is improved by transforming the data.
- (b) Plot the series y_t . Do any time intervals, of the order 100 years, exist where one can observe behavior comparable to that observed in the global temperature records in Figure 1.2?
- (c) Examine the sample ACF of y_t and comment.
- (d) Compute the difference $u_t = y_t - y_{t-1}$, examine its time plot and sample ACF, and argue that differencing the logged varve data produces a reasonably stationary series. Can you think of a practical interpretation for u_t ?

3.7. Use the three different smoothing techniques described in Example 3.16, Example 3.17, and Example 3.18, to estimate the trend in the global temperature series displayed in Figure 1.2. Comment.

3.8. In Section 3.3, we saw that the El Niño/La Niña cycle was approximately 4 years. To investigate whether there is a strong 4-year cycle, compare a sinusoidal (one cycle every four years) fit to the Southern Oscillation Index to a lowess fit (as in Example 3.18). In the sinusoidal fit, include a term for the trend. Discuss the results.

3.9. As in Problem 3.1, let y_t be the raw Johnson & Johnson series shown in Figure 1.1, and let $x_t = \log(y_t)$. Use each of the techniques mentioned in Example 3.20

to decompose the logged data as $x_t = T_t + S_t + N_t$ and describe the results. If you did Problem 3.1, compare the results of that problem with those found in this problem.