CHAPTER 1

INTRODUCTION

Data obtained from observations collected sequentially over time are extremely common. In business, we observe weekly interest rates, daily closing stock prices, monthly price indices, yearly sales figures, and so forth. In meteorology, we observe daily high and low temperatures, annual precipitation and drought indices, and hourly wind speeds. In agriculture, we record annual figures for crop and livestock production, soil erosion, and export sales. In the biological sciences, we observe the electrical activity of the heart at millisecond intervals. In ecology, we record the abundance of an animal species. The list of areas in which time series are studied is virtually endless. The purpose of time series analysis is generally twofold: to understand or model the stochastic mechanism that gives rise to an observed series and to predict or forecast the future values of a series based on the history of that series and, possibly, other related series or factors.

This chapter will introduce a variety of examples of time series from diverse areas of application. A somewhat unique feature of time series and their models is that we usually cannot assume that the observations arise independently from a common population (or from populations with different means, for example). Studying models that incorporate dependence is the key concept in time series analysis.

1.1 Examples of Time Series

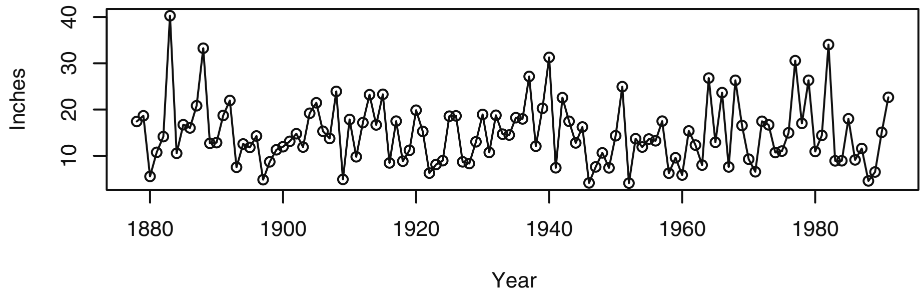
In this section, we introduce a number of examples that will be pursued in later chapters.

Annual Rainfall in Los Angeles

Exhibit 1.1 displays a time series plot of the annual rainfall amounts recorded in Los Angeles, California, over more than 100 years. The plot shows considerable variation in rainfall amount over the years—some years are low, some high, and many are in-between in value. The year 1883 was an exceptionally wet year for Los Angeles, while 1983 was quite dry. For analysis and modeling purposes we are interested in whether or not consecutive years are related in some way. If so, we might be able to use one year's rainfall value to help forecast next year's rainfall amount. One graphical way to investigate that question is to pair up consecutive rainfall values and plot the resulting scatterplot of pairs.

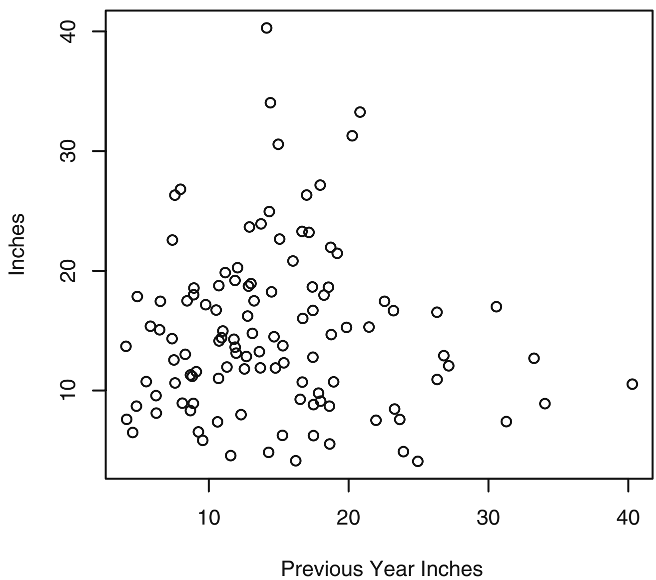
Exhibit 1.2 shows such a scatterplot for rainfall. For example, the point plotted near the lower right-hand corner shows that the year of extremely high rainfall, 40 inches in 1883, was followed by a middle of the road amount (about 12 inches) in 1884. The point near the top of the display shows that the 40 inch year was preceded by a much more typical year of about 15 inches.

Exhibit 1.1 Time Series Plot of Los Angeles Annual Rainfall




```
> library(TSA)
> win.graph(width=4.875, height=2.5,pointsize=8)
> data(larain); plot(larain,ylab='Inches',xlab='Year',type='o')
```

Exhibit 1.2 Scatterplot of LA Rainfall versus Last Year's LA Rainfall



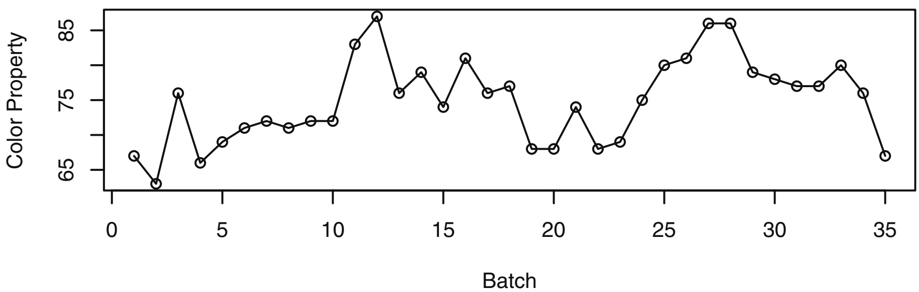

```
> win.graph(width=3,height=3,pointsize=8)
> plot(y=larain,x=zlag(larain),ylab='Inches',
  xlab='Previous Year Inches')
```

The main impression that we obtain from this plot is that there is little if any information about this year's rainfall amount from last year's amount. The plot shows no "trends" and no general tendencies. There is little correlation between last year's rainfall amount and this year's amount. From a modeling or forecasting point of view, this is not a very interesting time series!

An Industrial Chemical Process

As a second example, we consider a time series from an industrial chemical process. The variable measured here is a color property from consecutive batches in the process. Exhibit 1.3 shows a time series plot of these color values. Here values that are neighbors in time tend to be similar in size. It seems that neighbors are related to one another.

Exhibit 1.3 Time Series Plot of Color Property from a Chemical Process

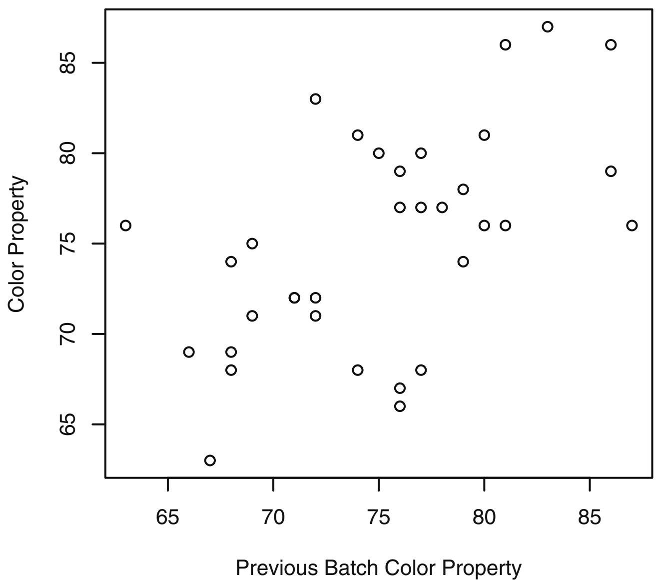



```
> win.graph(width=4.875, height=2.5,pointsize=8)
> data(color)
> plot(color,ylab='Color Property',xlab='Batch',type='o')
```

This can be seen better by constructing the scatterplot of neighboring pairs as we did with the first example.

Exhibit 1.4 displays the scatterplot of the neighboring pairs of color values. We see a slight upward trend in this plot—low values tend to be followed in the next batch by low values, middle-sized values tend to be followed by middle-sized values, and high values tend to be followed by high values. The trend is apparent but is not terribly strong. For example, the correlation in this scatterplot is about 0.6.

Exhibit 1.4 Scatterplot of Color Value versus Previous Color Value

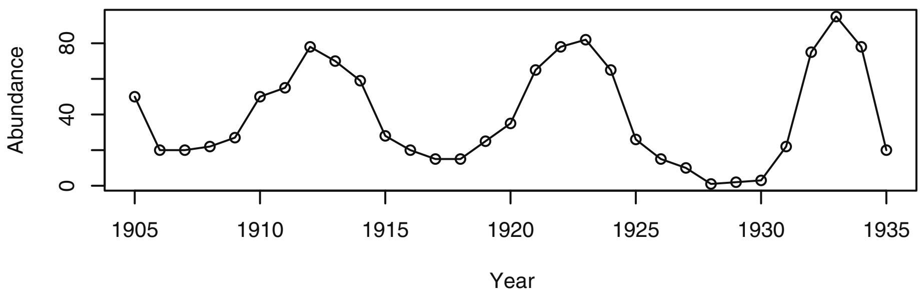



```
> win.graph(width=3,height=3,pointsize=8)
> plot(y=color,x=zlag(color),ylab='Color Property',
   xlab='Previous Batch Color Property')
```

Annual Abundance of Canadian Hare

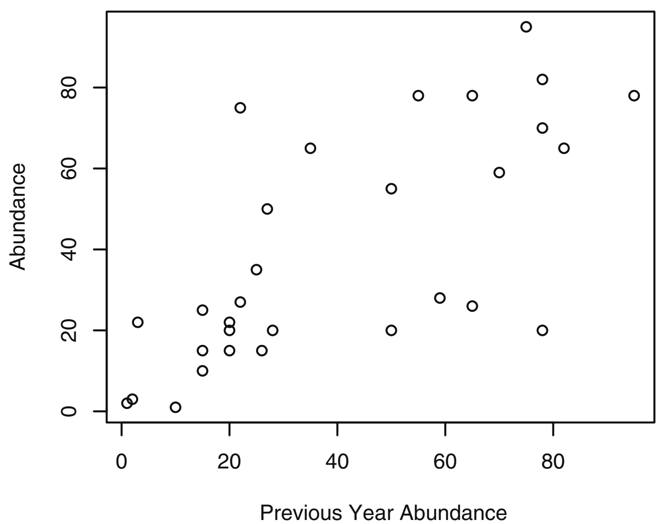
Our third example concerns the annual abundance of Canadian hare. Exhibit 1.5 gives the time series plot of this abundance over about 30 years. Neighboring values here are very closely related. Large changes in abundance do not occur from one year to the next. This neighboring correlation is seen clearly in Exhibit 1.6 where we have plotted abundance versus the previous year's abundance. As in the previous example, we see an upward trend in the plot—low values tend to be followed by low values in the next year, middle-sized values by middle-sized values, and high values by high values.

Exhibit 1.5 Abundance of Canadian Hare




```
> win.graph(width=4.875, height=2.5,pointsize=8)
> data(hare); plot(hare,ylab='Abundance',xlab='Year',type='o')
```

Exhibit 1.6 Hare Abundance versus Previous Year's Hare Abundance

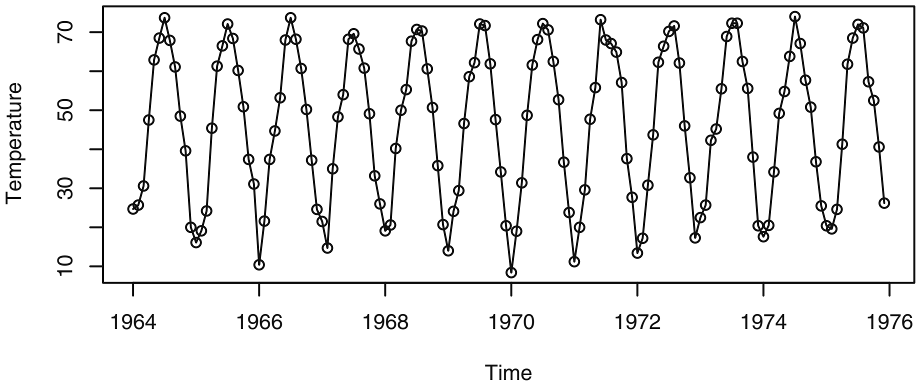



```
> win.graph(width=3, height=3,pointsize=8)
> plot(y=hare,x=zlag(hare),ylab='Abundance',
   xlab='Previous Year Abundance')
```

Monthly Average Temperatures in Dubuque, Iowa

The average monthly temperatures (in degrees Fahrenheit) over a number of years recorded in Dubuque, Iowa, are shown in Exhibit 1.7.

Exhibit 1.7 Average Monthly Temperatures, Dubuque, Iowa



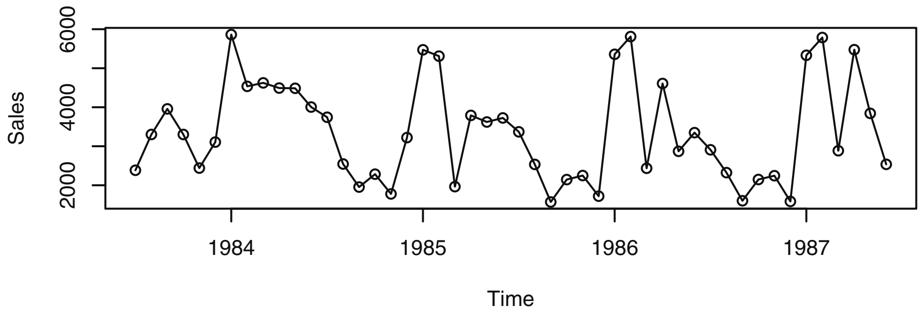

```
> win.graph(width=4.875, height=2.5,pointsize=8)
> data(tempdub); plot(tempdub,ylab='Temperature',type='o')
```

This time series displays a very regular pattern called **seasonality**. Seasonality for monthly values occurs when observations twelve months apart are related in some manner or another. All Januarys and Februarys are quite cold but they are similar in value and different from the temperatures of the warmer months of June, July, and August, for example. There is still variation among the January values and variation among the June values. Models for such series must accommodate this variation while preserving the similarities. Here the reason for the seasonality is well understood—the Northern Hemisphere's changing inclination toward the sun.

Monthly Oil Filter Sales

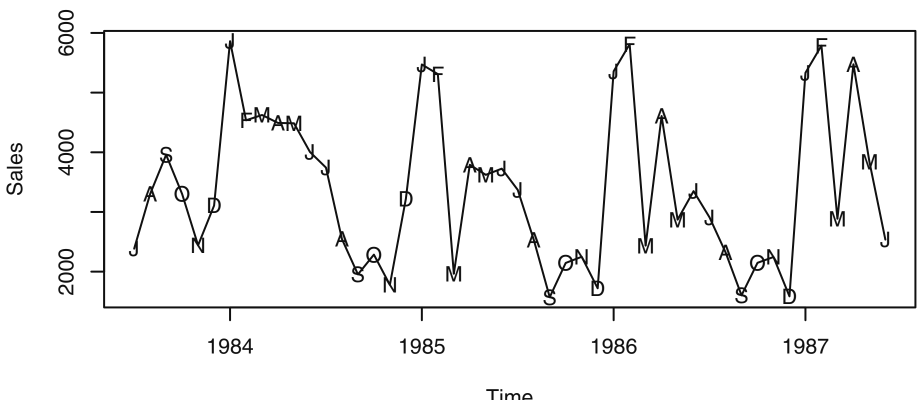
Our last example for this chapter concerns the monthly sales to dealers of a specialty oil filter for construction equipment manufactured by John Deere. When these data were first presented to one of the authors, the manager said, "There is no reason to believe that these sales are seasonal." Seasonality would be present if January values tended to be related to other January values, February values tended to be related to other February values, and so forth. The time series plot shown in Exhibit 1.8 is not designed to display seasonality especially well. Exhibit 1.9 gives the same plot but amended to use meaningful plotting symbols. In this plot, all January values are plotted with the character J, all Februarys with F, all Marches with M, and so forth.[†] With these plotting symbols, it is much easier to see that sales for the winter months of January and February all tend to be high, while sales in September, October, November, and December are generally quite low. The seasonality in the data is much easier to see from this modified time series plot.

Exhibit 1.8 Monthly Oil Filter Sales




```
> data(oilfilters); plot(oilfilters,type='o',ylab='Sales')
```

Exhibit 1.9 Monthly Oil Filter Sales with Special Plotting Symbols



J=January (and June and July),
F=February, M=March (and May), and so forth

```
> plot(oilfilters,type='l',ylab='Sales')
> points(y=oilfilters,x=time(oilfilters),
   pch=as.vector(season(oilfilters)))
```

† In reading the plot, you will still have to distinguish between Januarys, Junes, and Julys, between Marches and Mays, and Aprils and Augusts, but this is easily done by looking at neighboring plotting characters.

In general, our goal is to emphasize plotting methods that are appropriate and useful for finding patterns that will lead to suitable models for our time series data. In later chapters, we will consider several different ways to incorporate seasonality into time series models.

1.2 A Model-Building Strategy

Finding appropriate models for time series is a nontrivial task. We will develop a multistep model-building strategy espoused so well by Box and Jenkins (1976). There are three main steps in the process, each of which may be used several times:

1. **model specification** (or identification)
2. **model fitting**, and
3. **model diagnostics**

In model specification (or identification), the classes of time series models are selected that may be appropriate for a given observed series. In this step we look at the time plot of the series, compute many different statistics from the data, and also apply any knowledge of the subject matter in which the data arise, such as biology, business, or ecology. It should be emphasized that the model chosen at this point is *tentative* and subject to revision later on in the analysis.

In choosing a model, we shall attempt to adhere to the **principle of parsimony**; that is, the model used should require the smallest number of parameters that will adequately represent the time series. Albert Einstein is quoted in Parzen (1982, p. 68) as remarking that "everything should be made as simple as possible but not simpler."

The model will inevitably involve one or more parameters whose values must be estimated from the observed series. Model fitting consists of finding the best possible estimates of those unknown parameters within a given model. We shall consider criteria such as least squares and maximum likelihood for estimation.

Model diagnostics is concerned with assessing the quality of the model that we have specified and estimated. How well does the model fit the data? Are the assumptions of the model reasonably well satisfied? If no inadequacies are found, the modeling may be assumed to be complete, and the model may be used, for example, to forecast future values. Otherwise, we choose another model in the light of the inadequacies found; that is, we return to the model specification step. In this way, we cycle through the three steps until, ideally, an acceptable model is found.

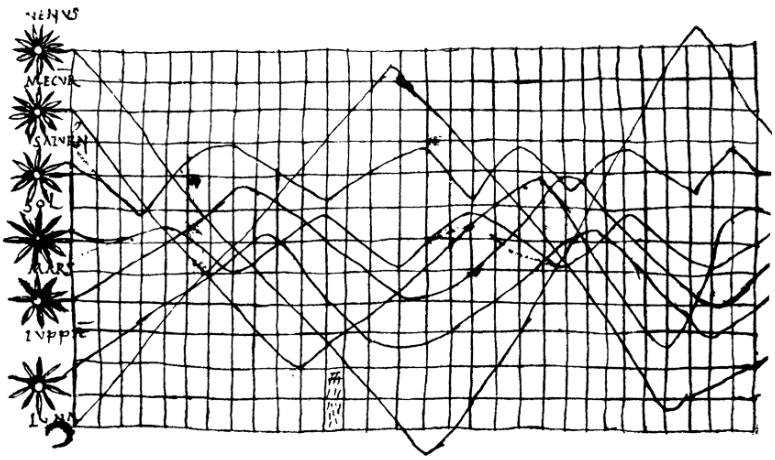
Because the computations required for each step in model building are intensive, we shall rely on readily available statistical software to carry out the calculations and do the plotting.

1.3 Time Series Plots in History

According toTufte (1983, p. 28), "The time-series plot is the most frequently used form of graphic design. With one dimension marching along to the regular rhythm of seconds, minutes, hours, days, weeks, months, years, or millennia, the natural ordering of the time scale gives this design a strength and efficiency of interpretation found in no other graphic arrangement."

Exhibit 1.10 reproduces what appears to be the oldest known example of a time series plot, dating from the tenth (or possibly eleventh) century and showing the inclinations of the planetary orbits.† Commenting on this artifact, Tufte says "It appears as a mysterious and isolated wonder in the history of data graphics, since the next extant graphic of a plotted time-series shows up some 800 years later."

Exhibit 1.10 A Tenth-Century Time Series Plot



1.4 An Overview of the Book

Chapter 2 develops the basic ideas of mean, covariance, and correlation functions and ends with the important concept of stationarity. Chapter 3 discusses trend analysis and investigates how to estimate and check common deterministic trend models, such as those for linear time trends and seasonal means.

Chapter 4 begins the development of parametric models for stationary time series, namely the so-called autoregressive moving average (ARMA) models (also known as Box-Jenkins models). These models are then generalized in Chapter 5 to encompass certain types of stochastic nonstationary cases—the ARIMA models.

Chapters 6, 7, and 8 form the heart of the model-building strategy for ARIMA modeling. Techniques are presented for tentatively specifying models (Chapter 6), efficiently estimating the model parameters using least squares and maximum likelihood (Chapter 7), and determining how well the models fit the data (Chapter 8).

Chapter 9 thoroughly develops the theory and methods of minimum mean square error forecasting for ARIMA models. Chapter 10 extends the ideas of Chapters 4

† From Tufte (1983, p. 28).

through 9 to stochastic seasonal models. The remaining chapters cover selected topics and are of a somewhat more advanced nature.

Exercises

1.1 Use software to produce the time series plot shown in Exhibit 1.2, on page 2. The data are in the file named larain.[†]

1.2 Produce the time series plot displayed in Exhibit 1.3, on page 3. The data file is named color.

1.3 Simulate a completely random process of length 48 with independent, normal values. Plot the time series plot. Does it look “random”? Repeat this exercise several times with a new simulation each time.

1.4 Simulate a completely random process of length 48 with independent, chi-square distributed values, each with 2 degrees of freedom. Display the time series plot. Does it look “random” and nonnormal? Repeat this exercise several times with a new simulation each time.

1.5 Simulate a completely random process of length 48 with independent, *t*-distributed values each with 5 degrees of freedom. Construct the time series plot. Does it look “random” and nonnormal? Repeat this exercise several times with a new simulation each time.

1.6 Construct a time series plot with monthly plotting symbols for the Dubuque temperature series as in Exhibit 1.7, on page 6. The data are in the file named tempdub.

[†] If you have installed the R package TSA, available for download at www.r-project.org, the larain data are accessed by the R command: data(larain). An ASCII file of the data is also available on the book Website at www.stat.uiowa.edu/~kchan/TSA.htm.