Part II

Time Series Models, Probability Theory, and Statistical Inference

Very early on (see page 20) we discussed the primary aims in time series analysis (1) descriptive, (2) inferential, (3) predictive, and (4) control. In Part I, we discussed elementary methods for describing the behavior of a time series data set. The main mechanisms put forth were basic descriptive statistics used in time series analysis – for example, the sample autocorrelation function or the periodogram. In traditional (i.i.d.) statistics, probability theory forms the bridge between descriptive statistics and inferential statistics. The same is true for time series analysis. And so now, we move to more advanced topics, beginning with probability theory for univariate time series in Chapter 4. As we progress through probability theory, the need for transforming a time series using the techniques presented in Chapter 3 will become clear.

We follow the discussion on probability theory for univariate time series with an introduction to the most commonly used models in traditional time series analysis in Chapter 5. After these models have been fully developed, we proceed to inferential statistical time series in Chapter 6. We conclude this part with a chapter on important algorithms for time series analysis.

Chapter 4

Probability Theory for Univariate Time Series

The descriptive statistics presented in Chapter 2 are important describing a time series. Throughout Chapter 2, we made some probabilistic assumptions without actually stating them. In this chapter, we address those assumptions. It is important to note that, if these assumptions are not satisfied, many of the descriptive statistics previously discussed make little sense. However, if the assumptions are satisfied, then the descriptive statistics can be used not only for describing the behavior of a time series, but also for statistical inference on the parameter of the series.

4.1 Introduction

As in many areas of statistics, our basic aim in time series analysis is twofold: descriptive and inferential. In Chapter 2 we considered ways to describe time series data. To make inferences from data, we need an underlying probabilistic structure. That structure might be attainable by applying the following strategy.

1. Assume that some member of a family of models will adequately represent the observed behavior of a time series data set.

2. Identify which member of the family best represents the data (model identification).

3. Estimate the parameters of the chosen model.

4. Check the adequacy of the fit of the estimated model.

5. Make statistical and scientific inferences based on the characteristics of the chosen model.

There are two primary objectives of this chapter. The first is to define and explain the probabilistic assumptions that make the descriptive statistics in Chapter 2 valid. If these assumptions are satisfied, then not only are the statistics useful for describing the behavior of a time series data set, but they are also estimators of time series *parameters*. Defining these parameters is the second objective. The parameters will only be well defined if the probabilistic assumptions hold. The parameters we define in this chapter are nonparametric, since we are not assuming a particular model for the data.

In the typical random sampling model of elementary statistics, one's aim is to make inferences about population parameters from statistics calculated from random samples. In time series analysis the ideas of "population" and "sample" are not so clear cut. We illustrated that difference in Chapter 3 when we discussed whether a realization of a random walk was due to a linear trend or nonconstant variance. In Figure 3.3, the top left graph was a single realization of length $n = 100$ of a random walk. The bottom right figure contained that realization, along with 19 other realizations (each of length $n = 100$) from the same random walk model, with the same initial value. Except for the (fixed) initial value, each value x_t has an infinite number of values it can be. In time series, we visualize that a data set $x_1, x_2, \ldots, x_n$ is just one possible set of many that could have been generated by some random mechanism producing data. The set of all possible realizations that could be observed is called the **ensemble** of realizations. Figure 4.1 is a time plot of a set of sunspot numbers observed annually from 1749 to 1963. We assume that the annual total sunspot numbers are

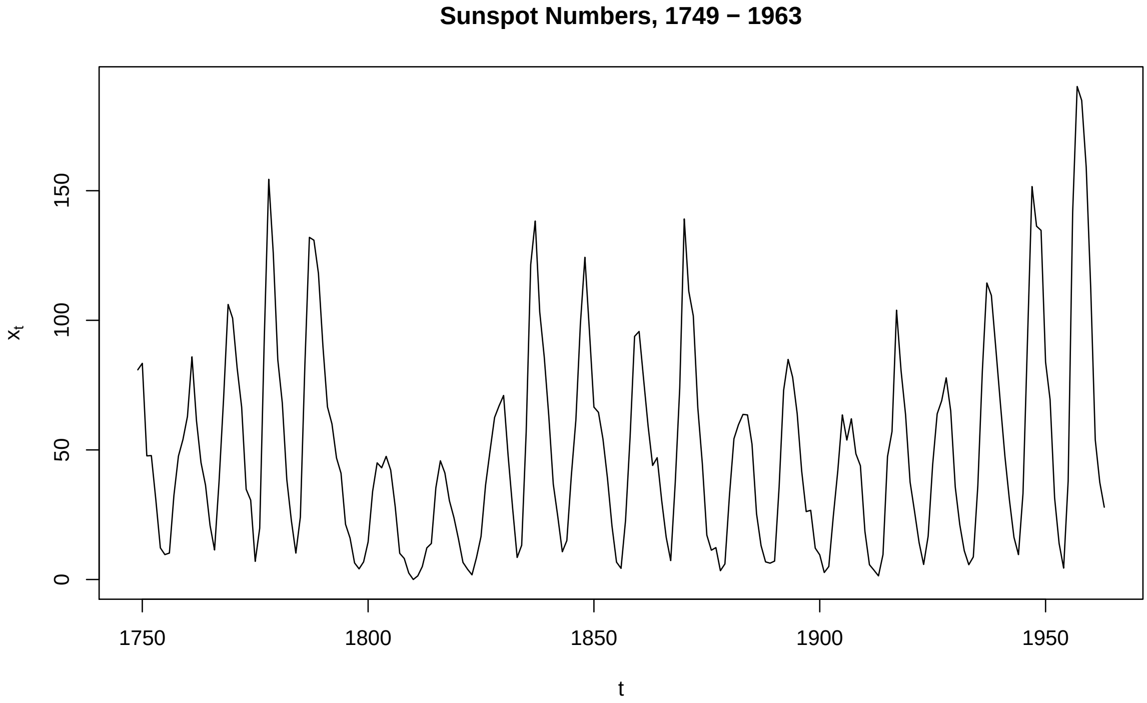

Figure 4.1: Sunspot numbers from 1749 to 1963.

a random function of time – that is, that the observation in any one year is a random variable.

Our aim is to make inferences about the joint behavior of these random variables when we have only *one* observation of each. We can do this by developing ideal models for their joint distributions. These ideas of "random functions in time" and "ensembles" also give rise to the caution that must be taken in interpretting statistics like the histogram (Section 2.3.1) or informative quantile plot (Section 2.3.2) of a time series realization. These graphical measures do not represent the same ideas they did in elementary statistics, where we usually need not consider joint behavior of observations because they are assumed independent and identically distributed.

Definition 4.1 TIME SERIES.
A **time series** *is an indexed collection* $\{X_t,\, t \in \mathcal{T}\}$ *of random variables having finite second moments; that is* $E[X_t^2] < \infty$ *for each element of the index set* $\mathcal{T}$*.*

Usually we will consider $\mathcal{T}$ to be the set $\mathcal{Z}$ of all integers; that is, we will assume that the phenomenon being observed has been going on for a long time, and will continue indefinitely. We will often refer to a time series asX_t, or just X if there is no possibility of confusion. We use capital letters to refer to random variables and lowercase letters for particular values for the random variables. For the sunspot data X_1 refers to the random variable representing all possible sunspot numbers in 1749, while $x_1 = 80.9$ denotes the actual observed value in 1749.

4.2 Covariance Stationary Time Series

Traditionally, two assumptions are made about the joint behavior of the random variables making up a time series. The first we will label as an "assumption," while the second assumption – that of covariance stationarity – is given as the definition of "covariance stationary" (Definition 4.3). These assumptions are defining the roles and restrictions on the *parameters* and of the time series.

We begin by stating the assumption on the first two moments (which are *parameters*) of the series use in describing the behavior of the series.

Assumption 4.1 *The behavior of a time series* X *can be adequately described by a knowledge of its mean value function* m *and covariance kernel* K*; that is,*

$$\begin{aligned} m(t) &= E[X_t]\,, \quad t \in \mathcal{Z} \\ K(s,t) &= Cov(X_t,\, X_s)\,, \quad s,t \in \mathcal{Z} \end{aligned}$$

Notice that this assumption implicitly states that from one time point to another, there is no guaranty that the value of the process mean is constant; in fact, it says the opposite – that the mean is itself a *function of time.* Likewise, if you are to consider the covariance between one set of values X_{s_1} and X_{t_1} that is not necessarily the same as the covariance between to other sets of values X_{s_2} and X_{t_2}, say. The covariance is a function of *when* each of the two point being considered occur. The fact that these two parameters change in accordance with time is in direct opposition to the behavior of parameters in traditional statistics when they are considered fixed constants.

Definition 4.2 GAUSSIAN TIME SERIES.
A time series X is said to be a **Gaussian time series** *if the joint distribution of any finite number of X_t is multivariate normal; that is, for any positive integer n, and any n integers $t_1, \ldots, t_n$, we have that the joint distribution of $X_{t_!}, \ldots, X_{t_n}$ is multivariate normal.*

If X is a Gaussian time series then Assumption 4.1 is valid. We will later see that a wide variety of non-Gaussian time series can also be effectively analyzed. In Part III, we will look at developments for analyzing time series data from processes which do not satisfy these assumptions.

The multivariate normal distribution plays a critical role in the theory and analysis of time series. A discussion of its basic properties is given in Appendix A.6.

Definition 4.3 COVARIANCE STATIONARY.
A time series X is said to be **covariance***,* **weakly***, or* **second-order stationary** *if its mean value function is a constant; that is $E[X_t] = \mu$, and if there exists a function $\{R_v, v \in \mathcal{Z}\}$ such that*

$$K(s,t) = Cov(X_t,\, X_s) = R_{t-s};$$

that is, the covariance function of any pair of Xs that are the same distance apart in time is the same. The function R is called the **autocovariance function** *of X.*

Notice that the assumption of covariance stationary greatly simplifies Assumption 4.1. It removes the dependence of the mean function on time, reducing the mean to a constant. In other words, for every random variable in the collection $\{X_t,\ t \in \mathcal{T}\}$, the mean is constant.

Since we are assuming a series is second-order stationary, then the constant mean should highlight the importance of transforming a time series using the methods in Section 3.1. If a series is observed to have an increasing trend, or a periodicity, then the series mean is not constant. The nonconstancy of the mean must be removed so that what remains has a constant mean.

The second part of covariance stationarity says that the autocovariance kernel is a function of only *distance apart in time* and not of *when* each of the two points occur. This greatly simplifies the description of the covariance. Notice that, if this assumption is not satisfied, then the Definition 2.1 of the sample autocovariance (autocorrelation) function would do little good in estimating a function that changes with when (s,t), as opposed to changing only with distance apart in time $(t-s)$. We will often drop the prefix "auto" in the word "autocovariance." Note that there is another (stronger) type of stationary, namely *strictly stationarity.* See Exercise T4.1.

4.2.1 The Autocorrelation Function

We usually concentrate on the correlation between two random variables rather than on their covariance.

Definition 4.4 Autocorrelation Function.
Let X be a covariance stationary time series having autocovariance function R. The **autocorrelation function of** X *is given by*

$$\rho_v = \mathit{Corr}\left(X_t,\, X_{t+v}\right), \quad t, v \in \mathcal{Z}.$$

A plot of ρ_v versus v for $v = 0, 1, \ldots, M$ is called the **correlogram of** X**.**

Notice that without the assumption of covariance stationary, this definition is insufficient for modeling the covariance kernel. Furthermore, by the definition of the correlation of two random variables, we have

$$\rho_v = \frac{\operatorname{Cov}\left(X_t,\, X_{t+v}\right)}{\sqrt{\operatorname{Var}\left(X_t\right)\operatorname{Var}\left(X_{t+v}\right)}} = \frac{R_v}{\sqrt{R_0 R_0}} = \frac{R_v}{R_0},$$

and thus $\rho_0 = 1$ and $\rho_{-v} = \rho_v$. Thus the autocorrelation function of X is just the autocovariance function divided by the variance R_0 of the series.

We will be primarily concerned with covariance stationary time series and thus will be trying to make inferences about μ and the autocovariance function R. In Section 4.7, we provide a statement of Theorem 4.2, the “Two Spectral Representations Theorem”. Some of the mathematics in the theorem are quite advanced, and so we reserved a statement of the theorem for Section 4.7, and a partial proof can be found in Appendix B.3. In short, the theorem summarizes the basic facts about R, and introduce two functions that are mathematically equivalent of R, and which are of importance in their own right, in theory and in practice.

Recall Assumption 4.1 – that we are assuming that all inferences about a covariance stationary time series X can be based on making inference about μ and R. From this assumption we can see the practical implications of the Two Spectral Representations Theorem. Part (c) through (e) of the theorem provide us with a function that is mathematically equivalent to R. The first of these functions $F(\omega)$, is the spectral distribution function, and the other $f(\omega)$ is the spectral density function. These functions (F and f) as parameters of the time series, are important in their own right in many circumstances. We have already seen their sample counterparts (in Chapter 2), and so have an idea of what these functions can tell us about an observed series. Further note that part (c) provides a harmonic analysis of the infinite sequence R; that is R can be represented as a sum (actually a special kind of integral) of sinusoids whose coefficients are “increments” of the spectral distribution function F. Part (d) says that one possible realization $\{x_t, t \in \mathcal{Z}\}$ can be thought of a sum of infinitely many sinusoids (it is always useful to think of any kind of integral as a sum of a large number of terms) where within a particular realization the amplitudes of those sinusoids are fixed, but between possible realizations they vary according to a probability law. These sinusoids are called **frequency components** and we can think of $f(\omega)$ as being proportional to the average value (over many realizations) of the squared amplitudes of the frequency component of frequency ω.

Part (e) gives conditions where the following relationships hold.

$$F(\omega) = \int_0^{\omega} f(x)\,dx, \quad f(\omega) = \frac{dF(\omega)}{d\omega}$$

and

$$R_v = \int_0^1 f(\omega)\cos(2\pi v\omega)\ d\omega, \quad v \in \mathcal{Z}.$$

The function F is called the **spectral distribution function** and f is called the **spectral density function**. Note that F and f are analogous to a cumulative distribution function (cdf) and probability density function (pdf) in probability theory, with some exceptions. First at $v = 0, f(\omega)$ integrates to R_0. Therefore, the spectral distribution and density functions provide a distribution of the variance of the time series as a function of frequency components, as opposed to the distribution of probability as a function of a random variable. To take this analogy one step further, note that R is in the form of a characteristic function (it is a sequence since f is symmetric on a finite interval).

We will sometimes extend f to the whole real line by making it a periodic function with period one; that is $f(\omega + k) = f(\omega)$ for all integers k. It should be noted that much of the literature of time series analysis defines f to be a function on $-\pi$ to π or -0.5 to 0.5. By doing a simple change of variables in all of the formulas above, we could obtain the corresponding formulas on other intervals (see Exercise T4.3).

4.2.2 Ensemble Mean Interpretation of ρ and f

Perhaps the most useful interpretation of ρ and f is as an ensemble averages of the sample autocorrelation function and sample spectral density functions introduced in Chapter 2. We will show in Chapter 6 that under very general conditions

$$\mathrm{E}\left[\hat{f}(\omega)\right] \longrightarrow f(\omega) \quad \text{and} \quad \mathrm{E}\left[\hat{\rho}_v\right] \longrightarrow \rho_v$$

as the length of the time series realization goes to infinity. When comparing these convergence results to similar convergence results in elementary statistics, there is a subtle difference that must be recognized. In elementary statistics, convergence results rely on having a sufficiently large sample size, so that the mathematical expression “as $n \to \infty$” means that we’re collecting more information from the same population. On the other hand, in time series that same phrase means that we’re collecting a longer path – but we still have only a single path. In other words, we have a sample of size one (path) that has a length of n. So while in elementary statistics, the expectation of similar results is an average over all possible samples with increasing size, in time series the expectation is over all possible ensembles of increasing length. This difference is most apparent in the mathematical proofs of the convergence theorems. In subsequent chapters, this difference will be highlighted, or references will be provided which will allow the interested reader to further delve into these subtleties of time series theory.

To continue our discussion on how to properly interpret these convergence results, consider the electroencephalograph (EEG) on a patient. We recognize that a sample time series

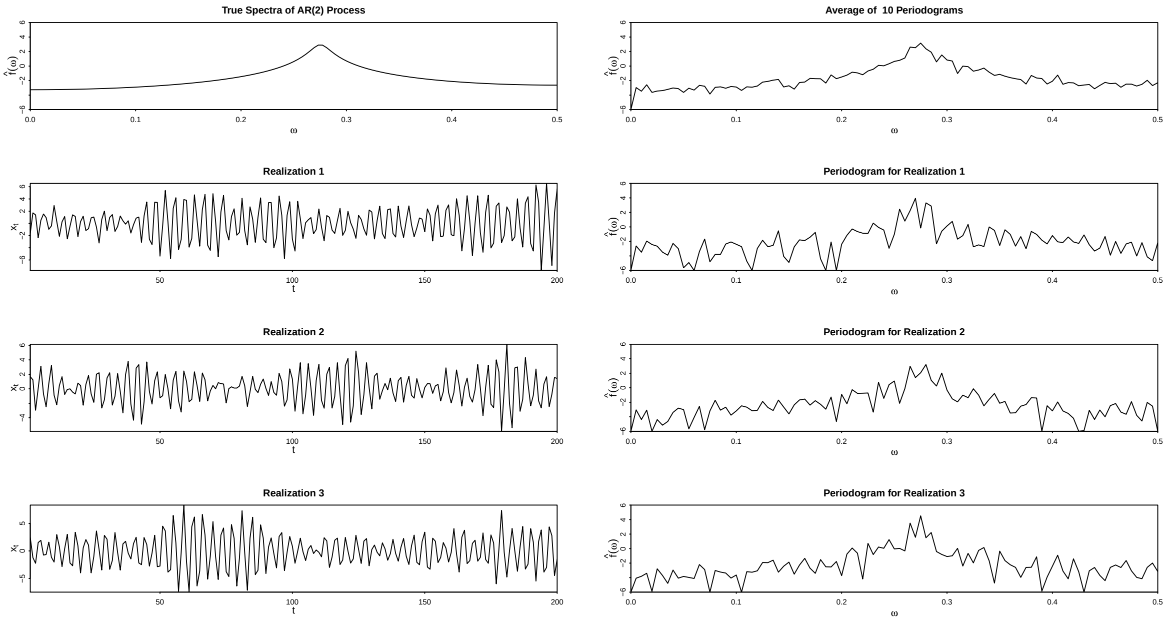

Figure 4.2: Illustrating the interpretation of the sample spectral density function as an ensemble average.

consisting of an EEG record would vary from one time to another in that patient's history and we would be most interested in making inferences about some average behavior of the observed series. In a situation like this, viewing the spectral density over many long realizations of EEG records is a natural thing to do.

To further illustrate this interpretation, consider Figure 4.2, which contains the graph of the true spectral density function f of a time series X, together with three realizations of length 200 from X and the sample spectral density for each of these realizations.

Finally, we have included the average of ten such sample spectral densities. Note how each of the sample spectral densities is extremely variable, but the average spectrum follows the true spectral density fairly closely. See Example 4.1 for a function that further develops the interpretation of the spectral density as the ensemble average of the periodogram.

4.2.3 The White Noise Process

There are many models that are used to represent data. In Chapter 5, we will discuss many of these traditional linear models. However, there is one model that is useful before we begin that discuss – the white noise "model." This is the simplest type of time series model, and occurs when X consists of uncorrelated random variables, each having mean μ and constant variance σ^2.

Definition 4.5 White Noise Process. *A time series X is said to be a* **white noise process with variance** σ^2 *if*

$$E[X_t] = 0, \quad t \in \mathcal{Z}, \quad \textit{and} \quad R_v = Cov(X_t, X_{t+v}) = \begin{cases} \sigma^2 & v = 0 \\ 0 & v \neq 0. \end{cases}$$

Such a process is denoted $X \sim WN(\sigma^2)$.

If we define the Kronecker delta function

$$\delta_v = \begin{cases} 1, & v = 0 \\ 0, & v \neq 0, \end{cases}$$

we can write $R_v = \delta_v \sigma^2$ when $X \sim \mathrm{WN}(\sigma^2)$. Note that

$$f(\omega) = \sum_{v=-\infty}^{\infty} R_v \cos(2\pi v\omega) = \sigma^2 \sum_{v=-\infty}^{\infty} \delta_v \cos(2\pi v\omega) = \sigma^2;$$

that is, the spectral density function of a white noise process is flat (constant). Thus, in analogy with the physical spectrum of white light (which contains equal amounts of all light frequencies), a sequence of uncorrelated random variables (which is often used to model physical noise) is referred to as white noise. Processes not having constant spectral densities are often called "colored noise".

4.3 Linear Filters

We will often express one time series Y as a linear function of the values of another series X. We would like to be able to derive the probabilistic properties of Y from known properties of X.

Definition 4.6 FILTERED VERSION OF A TIME SERIES.
The time series Y is called a **filtered version of X with filter coefficients** $\{\beta_j, j \in \mathcal{Z}\}$ *if we can write as a limit in mean square (see Part (e) of Definition A.6.4 in Appendix A.6)*

$$Y_t = \sum_{j=-\infty}^{\infty} \beta_j X_{t-j}, \quad t \in \mathcal{Z}.$$

In Section 3.1.3 we defined the moving average smoother

$$Y_t = \sum_{j=-M}^{M} \frac{1}{2M+1} X_{t-j},$$

which is a filtering operation with coefficients

$$\beta_j = \begin{cases} \dfrac{1}{2M+1} & |j| = 0, 1, \ldots M, \\ 0 & \text{otherwise.} \end{cases}$$

Another simple example of filtering a time series results in a model known as a moving average process. We will discuss these further in Section 5.3.

In Section 4.2.3, we saw that if $\varepsilon \sim \mathrm{WN}(\sigma^2)$, then $R_{\varepsilon,v} = \delta_v \sigma^2$ and $f_\varepsilon(\omega) = \sigma^2$ where we have now put the subscript ε on the R and the f to indicate to which time series they correspond. Theorem 4.1 will allow us to easily find R_Y and f_Y for a filtered version of another series. Before that theorem is stated, we will further simplified these issues with the introduction of the covariance generating function of the autocovariance function R.

Definition 4.7 COVARIANCE GENERATING FUNCTION.
The covariance generating function Γ of a covariance stationary time series X having autocovariance function R is the complex valued function

$$\Gamma(z) = \sum_{v=-\infty}^{\infty} R_v z^v, \quad z \in \mathcal{C}.$$

Note that $f(\omega) = \Gamma(e^{-2\pi i v\omega})$.

We now present the "Univariate Filter Theorem." Like Theorem 2.2, this theorem is basic enough, and fundamental enough, that it is stated here, as opposed to the "More Advanced" Section 4.7.

Theorem 4.1 UNIVARIATE FILTER THEOREM.
Suppose that X is a covariance stationary time series with autocovariance function R_X and spectral density function f_X. Suppose that Y is a filtered version of X with filter coefficients β. Then assuming that the quantities involved exist, we have

(a) Y is also covariance stationary.

(b) The autocovariance function of Y is given by

$$R_{Y,v} = \sum_{k=-\infty}^{\infty} R_{\beta,k} R_{X,v-k}, \quad v \in \mathcal{Z},$$

where

$$R_{\beta,k} = \sum_{j=-\infty}^{\infty} \beta_j \beta_{j+|k|}, \quad k \in \mathcal{Z}.$$

(c) The covariance generating function of Y is given by

$$\Gamma_Y(z) = \Gamma_X(z)\Gamma_\beta(z)\Gamma_\beta\left(z^{-1}\right),$$

where Γ_X is the covariance generating function of X and

$$\Gamma_\beta(z) = \sum_{j=-\infty}^{\infty} \beta_j z^j.$$

(d) The spectral density function of Y is given by

$$f_Y(\omega) = \left|h\left(e^{2\pi i\omega}\right)\right|^2 f_X(\omega),$$

where the function

$$h(z) = \sum_{k=-\infty}^{\infty} \beta_k z^k, \quad z \in \mathcal{C},$$

is called the **impulse response function of the filter**.

Note that part (d) follows immediately from part (c) by substituting $e^{2\pi i\omega}$ for z. We note that $h\left(e^{2\pi i\omega}\right)$ is called the **frequency transfer function** of the filter. We will use the Univariate Filter Theorem extensively throughout the text, particularly in Chapter 5 where we introduce time series models having a finite number of parameters.

4.3.1 The Effect of Differencing

The Univariate Filter Theorem is also important for studying the effects of some transformations. For example, we can see the effect of differencing very clearly. Suppose that X is a covariance stationary time series with spectral density function f_X. Suppose that Y is obtained as the d^{th} difference of X; that is $Y_t = X_t - X_{t-d}$. Then Y is covariance stationary and

$$f_Y(\omega) = \left|1 - e^{2\pi i d\omega}\right|^2 f_X(\omega),$$

since $h(z) = 1 - z^d$. Thus $f_Y(\omega)$ will be zero anywhere that $e^{2\pi i d\omega}$ is one, namely at any ω such that $d\omega$ is an integer. Thus

$$f_Y\left(\frac{j}{d}\right) = 0, \quad j = 0, 1, \ldots, d.$$

In particular, first differencing makes $f_Y(0) = f_Y(1) = 0$, while twelfth differencing makes $f_Y(0) = f_Y\left(\frac{1}{12}\right) = \cdots = f_Y(1) = 0$. (For more the application of this, see Section 3.1.2 and Exercises T3.2 and T3.3.) Thus differencing totally removed frequency components of these frequencies of a time series. In fact, any differencing makes $f_Y(0) = 0$. We will see in Chapter 6 that the spectral density at zero frequency is an important quantity in making inferences about the mean of a time series. Because of this zeroing out of frequencies, many analysts suggest that great care be taken when using differencing.

4.3.2 What Does the MA Smoother Do?

Another example of using the Univariate Filter Theorem to study the effect of transformations is to consider the moving average smoother. Let X be a covariance stationary time series with spectral density f_X and let

$$Y_t = \sum_{j=-M}^{M} \frac{1}{2M+1} X_{t-j}.$$

The frequency response function of this filter is

$$h\left(e^{2\pi i\omega}\right) = \frac{1}{2M+1} \sum_{j=-M}^{M} e^{2\pi ij\omega} = \frac{1}{2M+1} D_M(\omega)$$

where the function

$$D_M(\omega) = \sum_{j=-M}^{M} e^{2\pi ij\omega}$$

is called the **Dirichlet kernel** and is well known in many scientific areas. We have seen (Exercise T4.6)

$$D_M(\omega) = \frac{\sin\left(\{M+0.5\}2\pi\omega\right)}{\sin\left(\pi\omega\right)}.$$

Figure 4.3 (see Example 4.2) gives graphs of D_M for $M = 5, 10, 20$, and 40.

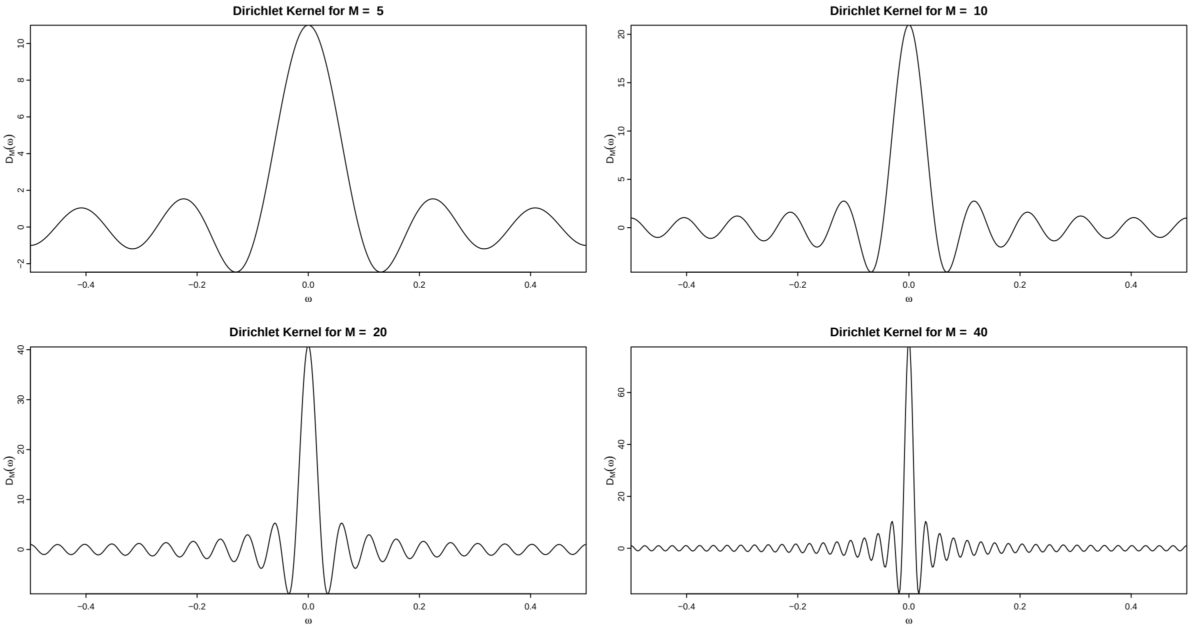

Figure 4.3: The Dirichlet kernel for $M = 5, 10, 20$ and 40.

Note that the kernel becomes more concentrated about zero as M increases. Note also that the kernel is negative for certain frequency ranges and has large "side lobes," that is, secondary peaks. Thus we have

$$f_Y(\omega) = \left(\frac{1}{2M+1}\right)^2 |D_M(\omega)|^2 f_X(\omega),$$

and D_M becomes more and more concentrated around frequency zero as M (the number of terms on each side of X_t used in the average) gets large. Thus the moving average smoother is essentially allowing only frequency components of low frequency to be "passed" to Y from X.

Definition 4.8 Low and High Pass Filters & Bandpass Filter.
If Y is a filtered version of X with frequency transfer function $h\left(e^{2\pi i\omega}\right)$, then the filter is called

(a) a **low (high) pass filter** *if only low (high) frequency components are passed through a filter, that is if $f_Y(\omega) = 0$ for all $\omega > \omega_1$ ($\omega \leq \omega_1$) for some frequency ω_1.*

(b) a **bandpass filter** *if only frequency components in a certain interval (band) of frequencies are passed through the filter.*

Thus the moving average smoother is an example of a low pass filter except that its frequency transfer function never becomes exactly zero for high frequencies. We can now see why $h\left(e^{2\pi i\omega}\right)$ is called the frequency transfer function of the filter; it determines what happens to the various frequency components in X as a result of the filter.

4.3.3 Designing Filters

Sometimes we want to design filters, that is, determine filter coefficients so that an output time series Y has a certain type of spectral density. We illustrate this by considering the design of a bandpass filter. Suppose that we want

$$f_Y(\omega) = \begin{cases} f_X(\omega) & \omega \in [u-v,\, u+v] \text{ or } \omega \in [1-(u+v),\, 1-(u-v)] \\ 0 & \text{otherwise;} \end{cases}$$

that is, we want to determine a bandpass filter centered at frequency u and having bandwidth v. Recall that a spectral density is symmetric around $\omega = 0.5$, that is, $f(\omega) = f(1-\omega)$ for $\omega \in [0,\, 1]$, and thus we must put both intervals in the definition of f_Y. Thus we are looking for a filter having frequency response function

$$h\left(e^{2\pi i\omega}\right) = \begin{cases} 1 & \omega \in [u-v,\, u+v] \text{ or } \omega \in [1-(u+v),\, 1-(u-v)] \\ 0 & \text{otherwise.} \end{cases}$$

This mean we must find (see Appendix A.2)

$$\begin{aligned} \beta_k &= \int_0^1 h\left(e^{2\pi i\omega}\right) e^{-2\pi i k\omega}\, d\omega \\ &= \int_{u-v}^{u+v} e^{-2\pi i k\omega}\, d\omega + \int_{1-(u+v)}^{1-(u-v)} e^{-2\pi i k\omega}\, d\omega. \end{aligned}$$

For $k = 0$, this gives $\beta_0 = 4v$, while for $k \neq 0$,

$$\beta_k = \frac{2}{\pi k} \cos\left(2\pi k u\right) \sin\left(2\pi k v\right).$$

Thus the filter giving rise to the ideal transfer function h has infinitely many coefficients. In practice, we need to determine how many terms in the filter are needed to produce a satisfactory approximation to the transfer function. In the left column of graphs in Figure 4.4, we superimpose the transfer function h of the ideal bandpass filter for $u = 0.25$ and $v = 0.1$ and the M-term approximation

$$h_M\left(e^{2\pi i\omega}\right) = \sum_{k=-M}^{M} \beta_k e^{2\pi i k\omega}$$

to h for $M = 12, 24$ and 48. Note the approximation does not approach the idea function in a smooth way; rather it is very wiggly and has a strange behavior at the points of discontinuity of h, namely at the values 0.15 and 0.35. This phenomenon of overshooting the function at a point of discontinuity is called Gibb's phenomenon in the theory of Fourier series. Note also that the approximating functions are negative in places, even though the true function is never negative.

The traditional method for eliminating Gibb's phenomenon and the negativity in approximating functions is to apply nonnegative weights to the βs (the βs are called the Fourier coefficients of h) before finding the approximating series. This is what we have done to obtain the second column of graphs in Figure 4.4.

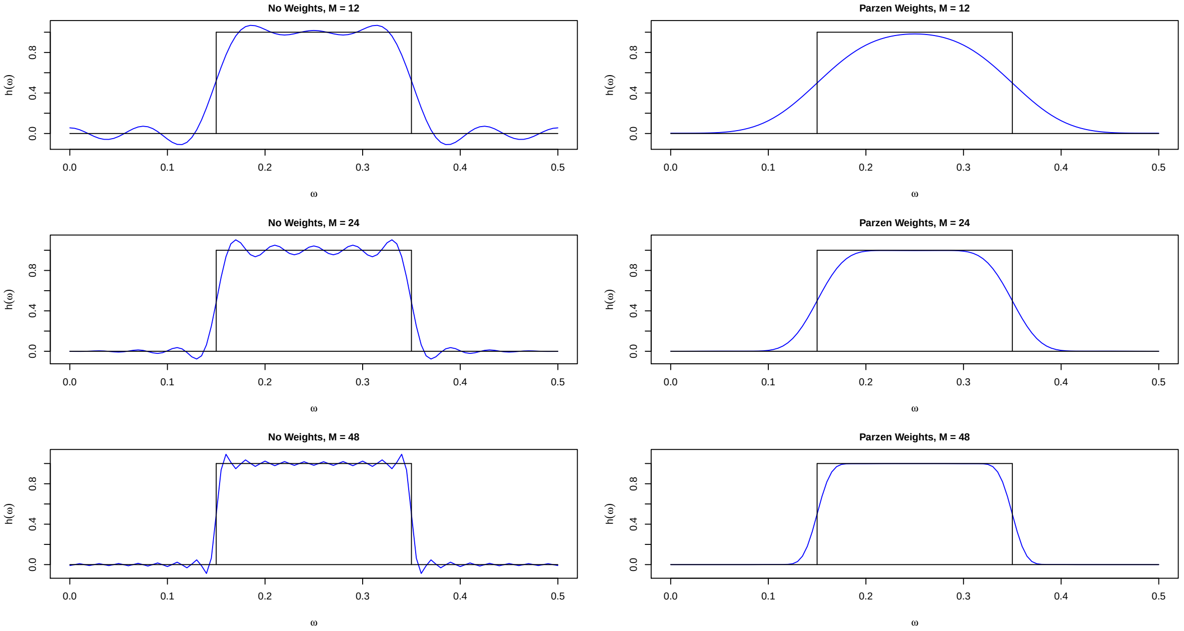

Figure 4.4: Approximations to the ideal bandpass filter transfer function using unweighted and weighted Fourier coefficients.

We have used the Parzen weights (see Table 6.1). Note how the approximating functions are approaching h much more smooth and are never negative. The penalty for this is that for the same M, the unweighted function is closer to h as measured by the integral of the squared difference between h and h_M than is the function using the weights. See Example 4.3 for the function that generated Figure 4.4.

4.3.4 The Lag Operators

Linear filters can be succinctly represented if we introduce what is called the lag (or backshift) operator.

Definition 4.9 LAG (BACKSHIFT) OPERATOR & FILTER POLYNOMIAL OPERATOR. *The* **lag operator** L *operating on a time series* X *is defined by*

$$L^k X_t = X_{t-k}, \quad t \in \mathcal{Z}.$$

If Y *is a filtered version of* X *we defined the* **filter polynomial operator** *to be*

$$h(L) = \sum_{k=-\infty}^{\infty} \beta_k L^k.$$

Thus we can formally write

$$\begin{aligned} Y_t &= \sum_{k=-\infty}^{\infty} \beta_k X_{t-k} \\ &= \sum_{k=-\infty}^{\infty} \beta_k L^k X_t \\ &= h(L) X_t. \end{aligned}$$

In Section 3.2.6, we saw the function `filt` function will form a filtered version of a time series.

4.4 Time Series Prediction

One of the major aims of time series analysis is to make use of correlation between and within series to predict unobserved values of a series. In this section, we describe some of the basic theory for prediction. In Chapter 7 we discuss in detail several algorithms for calculating predictors for the time series models that we present in Chapter 5. Much of the theory of prediction is due to the remarkable work of Whittle (1963a).

If we have a realization $X_1, \ldots, X_n$ from a time series X, we often wish to find a function of the data that is close to some future variable X_{n+h}. We call n the **memory** or the **origin** of the predictor and h the **horizon** or **number of steps ahead** being predicted.

Definition 4.10 Best Unbiased Predictor, Best Unbiased Linear Predictor, Infinite Memory Predictor, & Predictor Error Variances.
Let $\boldsymbol{X}_n = (X_1, \ldots, X_n)^T$ be a realization of length n from a time series X.

(a) The **best unbiased predictor** *of X_{n+h} given $\boldsymbol{X}_n$ is that function $\tilde{X}_{nh}$ of $\boldsymbol{X}_n$ that has the same mean as X_{n+h} and has smaller prediction error variance that any other unbiased function of $\boldsymbol{X}_n$.*

(b) The **best unbiased linear predictor** *of X_{n+h} given $\boldsymbol{X}_n$ is that linear function $\hat{X}_{nh}$ of $\boldsymbol{X}_n$ that has the same mean as X_{n+h} and has smaller prediction error variance than any other unbiased linear function of $\boldsymbol{X}_n$.*

(c) If $\hat{X}_{nh}$ converges in mean square as $n \to \infty$ to a random variable X_{nh}, then X_{nh} is called the **infinite memory h step ahead predictor** *of X_{n+h}.*

(d) The **error variances** *of $\tilde{X}_{nh}, \hat{X}_{nh}$ and X_{nh} are denoted $\tilde{\sigma}^2_{nh}, \hat{\sigma}^2_{nh}$, and σ^2_{nh}, respectively.*

The Theorem 4.3 is a straightforward application of the material in Appendix A.6.3 on predictor of random vectors. However, it is a bit more advanced, so we will relegate the statement of the theorem to Section 4.7. Here, we provide an interpretation of the results in the theorem. To begin, part (a) simply says that the best predictors and their error variances are conditional means and variance of X_{n+h} given the realization $X_1, \ldots, X_n$, while part (b) gives formulas for finding the best linear unbiased predictors and their error variances. Notice that the prediction normal equations are very similar to the normal equations in regression analysis, with R_0 playing the role of $\boldsymbol{y}^T\boldsymbol{y}$, $\boldsymbol{\Gamma}_n$ playing the role of $\boldsymbol{X}^T\boldsymbol{X}$, and $\boldsymbol{r}_{n,h}$ playing the role of $\boldsymbol{X}^T\boldsymbol{y}$. The last part of part (b) provides an often used method for calculating linear predictors, namely the device of pretending that the process is Gaussian and then using conditional means and variances.

Part (c) of the theorem contains the often used Levinson algorithm (see Levinson (1947)). This algorithm allows us to find recursively the prediction coefficients. We will refer to this algorithm frequently throughout the remainder of the text. The last part of part (c) shows that for each memory n, the last prediction coefficient is indeed a partial correlation. The partials are used extensively in identifying time series models (Chapter 5). In Chapter 7 we give more details about part (c). (See Theorems 7.2 and Theorem 7.3.)

Part (d) of the theorem requires a little more explanation. The beginning result provides a mathematical definition for *purely nondeterministic*. In short, the definition means that, as the serie's past extends infinitely back in time, and the variance of the one step ahead prediction is positive (nonzero), then the series cannot be perfectly predicted, no matter how much of the past we know. Part (d) continues from this definition. Part (di) goes on to say that if a series is purely nondeterministic, then the series X can be written as a linear filter of past values of a white noise series. This linear filter is known as the *infinite order moving average represenation* of X. We will study a "moving average process" in detail in Section 5.3. The remaining two parts of (d) provides algorithms for finding infinite memory linear predictors via the infinite order moving average representation. The infinite memory linear predictors are often very easy to calculate. Further the infinite order moving average representation of a process in terms of its innovations is often used in other contexts (see Theorem 7.4, for example). From (dii) we the infinite memory one step ahead prediction error $X_{n+1} - X_{n1} = \varepsilon_{n+1}$, which is the origin of the term *innovation*. The ε_{n+1} is what is left over after having use the infinite past of X to predict X_{n+1}. Finally, part (e) is the extremely important Wold Decomposition, which provides a decomposition of any covariance stationary time series into two parts, one that can be perfectly predicted if we know enough of its past, and one part that cannot be perfectly predicted, no matter how much of its past we know. Most of the time series models that we consider have only this unpredictable or nondeterministic part.

4.5 Using *Timeslab in R* and *R*

4.5.1 *Timeslab in R* Functions

The `toepl` Function

The function `toepl` will form a Toeplitz matrix from the input values. The call to the function is

```
toepl(R, R0, M = (length(R)+1))
```

where `R` is a real vector containing the off-diagonal elements and `R0` is a real scalar containing the diagonal element of the desired Toeplitz matrix. The argument `M` is a real scalar indicating how many elements of `R` to use (`M` $-$ 1 = `length(R)`). The default is that the function uses all elements of `R`. The function returns the `M` × `M` symmetric Toeplitz matrix. For more on Toeplitz matrices, see Section 4.7 and Example 4.4.

The `filt` Function

The `filt` function can be called via

```
filt(beta, beta0, x)
```

will return a real vector that is a filtered version of the input vector `x`. The lead coefficient of the filter is specified by the user in the real scalar `beta0`, and the remaining coefficients are provided in the real vector `beta`.

4.5.2 *R* Functions

The `eigen` Function

The *R* function `eigen` can be used for computing the eigenvalues and eigenvectors of a matrix. The call to the function is

```
eigen(x, symmetric, only.values = FALSE, ...)
```

where x is a matrix for which the eigenvalues and eigenvectors are desired. The argument symmetric is a logical. If symmetric = TRUE, the matrix is assumed to be symmetric. If symmetric is not specified, then the matrix is inspected for symmetry. The next argument, only.values is also a logical argument. If only.values = TRUE, then only the eigenvalues are computed. Otherwise, both the eigenvalues and eigenvectors are computed. The function eigen returns a vector of eigenvalues only if only.values = TRUE. Otherwise eigen returns a list containing two objects. The first is the vector values, which contains the eigenvectors. The second is vectors which is either a matrix whose columns contain the eigenvectors of x or NULL if only.values = TRUE.

We have provided only a brief description of the most basic use of this function. For more information, the reader is referred to the *R* help pages for more detail on using the eigen function.

4.6 Pulling it All Together

Example 4.1 The Spectral Density Function.
We saw in Section 4.2.2 that the spectral density function of a time series can be thought of as the average over many long realizations of the periodogram (the unsmoothed sample spectral density function at the natural frequencies). The function perdave *can be used to generate a series of realizations from a specified autoregressive process (for more on autoregressive process, see Section 5.4) and keep a running average of the periodograms found for each one. There are four plots on the screen while the function is running: (1) the true spectral density of the process, (2) the current data set, (3) the periodogram of the current data set, and (4) the average of the periodograms that have been produced. The point of the function is that the average of the periodograms gets closer and closer to the true spectral density as the number of realizations increases. Note that all of the spectral plots are on a log scale.*

```
perdave <- function(n, alpha, nsamps = 20, Q = n, rvar = 1, seed = 0)
#-----------------------------------------------------------------------
#  R function to illustrate the unbiasedness of the periodogram.  For
#  a specified AR(p) process, the function generates nsamps samples of
#  size n.
#
#  INPUT: n = an integer scalar indicating the length of the each
#             series.
#         alpha = a real vector containing the coefficients of the AR
#                 process.
#         nsamps = the number of samples to generate.  The default
#                  value is nsamps = 20.
#         Q = an integer specifying the number of frequencies on the
#             interval [0, 1] at which to compute the spectral density
#             function.  The default value is Q = n (which is best for
#             comparing returned numerical output).
#         rvar = a real scalar containing the variance of the error
#                terms.  The default value is rvar = 1.
#         seed = a nonnegative integer for the random number generator.
#                If seed = 0 (the default) then the output seed from
#                the last call is used.  "seed" is an optional argument.
#
#  VALUE: The function perdave creates a plotting window with four
#          graphs in a 2 x 2 array.  The top left graph is the true
#          AR(p) spectral density function.  The top right graph is
#          the average of the periodograms generated thus far.  The
#          bottom left graph is of the current AR(p) realization.
#          The bottom right graph is of the periodogram of the
#          current AR process.  When the function exits, a list
#          containing the following objects is returned.
#          truef = a vector of length [Q/2]+1 containing the true
#                  AR spectra,
#          averagef = a vector of length [n/2]+1 containing the
#                     average of the periodograms.
#          nsamps = the number of samples used for computing averagef.
#-----------------------------------------------------------------------
{
  p        <- length(alpha)
  R0       <- arcorr(alpha, rvar, m = (p + 1))$var
  f        <- arsp(alpha, rvar, Q)
  averagef <- rep(0, length = (floor(n/2)+1))

  par(mar = c(3, 2, 2, 1), oma = rep(0, length = 4),
      tck = -0.015, cex.axis = 0.75, xaxs = "i", yaxs = "i",
      cex.main = 0.75, mgp = c(1, 0.25, 0), cex.lab = 0.85)

  split.screen(c(2, 2), erase = TRUE)
  screen(n = 1)
  plotsp(f, Q, R0, main = paste("True AR(",p,") Spectral Density"))

  samp <- 0
  b    <- "C"
  while(b == "C" && samp < nsamps) {
    samp     <- samp + 1
    x        <- ardt(alpha, n, rvar, seed)$x
    fhat     <- perdgm(x)
    averagef <- ((samp - 1)*averagef + fhat) / samp
    screen(n = 2, new = TRUE)
    plotsp(averagef, n, R0,
      main = paste("Average of",samp,"Periodograms"))
    screen(n = 3, new = TRUE)
    plot(x, type = "l", xlab = "t", ylab = expression(x[t]),
         main = paste("AR(",p,") Realization #",samp))
    screen(n = 4, new = TRUE)
    plotsp(fhat, n, R0,
      main = paste("Periodogram of Realization #",samp))

    if(samp < nsamps) {
      b <- readline("Do you want to continue? (Strike <Enter>,
             or type C or Y.) ")
      if(b == "c" || b == "Y" || b == "y" || b == "") b <- "C"
    }
  }
  if(samp < nsamps) nsamps <- samp
  close.screen(all = TRUE)

  return(list(f = f, averagef = averagef, nsamps = nsamps))
}
```

Example 4.2 The Dirichlet Kernel.
The function plot.dirichlet *was used to generate Figure 4.3.*

```
plot.dirichlet <- function(M = c(5, 10, 20, 40))
#-----------------------------------------------------------------------
# R function to create a plot of the Dirichlet kernel for specified
# values of M.
#
# INPUT: M = a positive integer scalar or vector containing values of M.
#            The default value is the vector M = c(5, 10, 20, 40).
#
# VALUE: The function plot.dirichlet returns a graphics window containing
#        a set of graphs in an length(M)/2 x 2 array of the Dirichlet
#        kernel.
#-----------------------------------------------------------------------
{
  rows <- 1
  cols <- 1
  if(length(M) > 1) {
    rows <- floor(length(M)/2)
    cols <- 2
  }
  if(length(M) > 1 && (length(M)/2 - rows) != 0)
    stop("\nThe length of M must be either 1 or an even number.\n")
  if(min(M) <= 0 || min(M - floor(M)) != 0)
    stop("\n All values of M must be positive integers.\n")

  omega <- seq(-0.5, 0.5, length = 501)

  par(mfrow = c(rows, cols), oma = rep(0, length = 4), tck = -0.015,
      mar = c(3, 2, 2, 1), cex.axis = 0.75, xaxs = "i", yaxs = "i",
      cex.main = 1, mgp = c(1, 0.25, 0), cex.lab = 0.85)
  for(i in 1:length(M)) {
    plot(omega, dirichlet(M[i]), type = "l",
      xlab = expression(omega), ylab = expression(D[M]*(omega)),
      main = paste("Dirichlet Kernel for M = ",M[i]))
  }
  return(invisible())
}

dirichlet <- function(M, n = 501)
#-----------------------------------------------------------------------
#  R function to compute the Dirichlet kernel at value of M.
#
#  INPUT: M = a positive integer.
#         n = a positive integer indicating at how many points the
#             kernel should be computed.  The default is n = 501.
#
#  VALUE: The function dirichlet returns a vector of length n
#         containing values of the Dirichlet kernel for n frequencies
#         from -0.5 to 0.5.
#-----------------------------------------------------------------------
{
  if(n <= 0 || (floor(n)-n) != 0)
    stop("\n The value of n must be an integer greater than one.\n")
  if(min(M) <= 0 || min(floor(M)-M) != 0)
    stop("\n All values of M must be positive integers.\n")

  omega <- seq(-0.5, 0.5, length = n)

  return(sin(2*pi*omega*(M+0.5))/sin(pi*omega))
}
```

Example 4.3 The Bandpass Filter Example.
The functions band *and* parz *were used to form the graphs in Figure 4.4.*

```
band <- function(m = c(12, 24, 48), omega = 0.25, bw = 0.1, npts = 200)
#-----------------------------------------------------------------------
#  R Function to compute the ideal transfer function of a bandpass
#  filter centered at frequency omega and bandwidth bw.
#
#  INPUT: m = integer scalar (or vector) specifying the number of terms
#             in the approximation.
#         omega = an real scalar between -0.5 and 0.5 containing the
#                 frequency at which to center the filter.  The default
#                 value is omega = 0.25.
#         bw = a real scalar between 0 and 1 containing the bandwidth
#              of the filter.  The default value is bw = 0.1.
#         npts = an integer scalar indicating how many points at which
#                to evaulate the transfer functions (on frequencies from
#                0 to 1).  The default is npts = 200.  The plot will be
#                based on [npts/2]+1 (on frequencies from 0 to 0.5).
#
#  VALUE: The function band creates side-by-side plots of a transfer
#         function, with approximations using unweighted Fourier
#         coefficients (on the left) and using Parzen weighted
#         Fourier coefficients (on the right).  The function also
#         returns a list containing the following objects.  For
#         M = max(m),
#         M = an integer scalar containing the maximum value of m.
#         beta = a real vector of length M containing the unweighted
#                Fourier coefficients of the transfer function.
#         z = a real vector of length [npts/2]+1 containing the
#             value of the frequency transfer function using the
#             unweighted Fourier coefficients (beta).
#         betap = a real vector of length M containing the Parzen
#                 weighted Fourier coefficients of the frequency
#                 transfer function.
#         zp = a real vector of length [npts/2]+1 containing the
#              value of the frequency transfer function using Parzen
#              weighted Fourier coefficients (betap).
#         freqs = a vector of length [npts/2] + 1 containing the
#                 frequencies in [0, 0.5] for z and zp.
#
#  NOTE: This function is NOT part of Timeslab in R.
#-----------------------------------------------------------------------
```

```
{
  if(abs(omega) >= 0.5)
    stop("\n The value of omega must be between -0.5 and 0.5.\n")
  if(bw < 0 || bw > 1)
    stop("\n The value of the bandwidth must be between 0 and 1.\n")
  if(max(floor(m) - m) != 0 || min(m) < 4)
    stop("\n All values of m must be integers greater than 4.\n")

  m <- sort(m)
  par(mfrow = c(length(m), 2), oma = rep(0, length = 4),
      mar = c(5, 4, 2, 2) + 0.1, cex.main = 1)

  for(i in 1:length(m)) {
    M     <- m[i]
    zero  <- rep(0, length = (npts - M - 1))
    fr    <- seq(0, 0.5, length.out = length(z))

    beta  <- c(4*bw, 2*cos(2*pi*(1:M)*omega)*sin(2*pi*(1:M)*bw)/(pi*(1:M)))
    z1    <- Re(fft(c(beta, zero)))[1:(floor(npts/2)+1)]
    z2    <- Re(fft(c(beta, zero), inverse = TRUE))[1:(floor(npts/2)+1)]
    z     <- z1 + z2 - 4*bw

    betap <- beta*parz(M)
    z1    <- Re(fft(c(betap, zero)))[1:(floor(npts/2)+1)]
    z2    <- Re(fft(c(betap, zero), inverse = TRUE))[1:(floor(npts/2)+1)]
    zp    <- z1 + z2 - 4*bw

    plot(fr, z, type = "l", col = "blue",
         xlim = c(0, 0.5), ylim = c(min(z, zp), max(z, zp)),
         xlab = expression(omega), ylab = expression(h*(omega)),
         main = paste("No Weights, M =", M))
    lines(c(0, 0.5), c(0, 0))
    lines(rep(omega - bw, length = 2), c(0, 1))
    lines(c(omega - bw, omega + bw), rep(1, length = 2))
    lines(rep(omega + bw, length = 2), c(0, 1))

    plot(fr, zp, type = "l", col = "blue",
         xlim = c(0, 0.5), ylim = c(min(z, zp), max(z, zp)),
         xlab = expression(omega), ylab = expression(h*(omega)),
         main = paste("Parzen Weights, M =", M))
    lines(c(0, 0.5), c(0, 0))
    lines(rep(omega - bw, length = 2), c(0, 1))
    lines(c(omega - bw, omega + bw), rep(1, length = 2))
    lines(rep(omega + bw, length = 2), c(0, 1))
  }
  M <- max(m)
  return(list(beta = beta, z = z, betap = betap, zp = zp, freqs = fr))
}
#
#
#
parz <- function(M)
#------------------------------------------------------------------------
#  R function to compute the Parzen weight function for indices
#  0, 1, ..., M (on frequencies 0, 1/M,..., (M-1)/M, 1).
#
#  INPUT: M = an integer scalar specifying the number of indices at
#             which to compute the Parzen weights.
#
#  VALUE: The parz function returns a real vector of length M
#         containing the value of the Parzen weight function for indices
#         0, 1, ..., M (on frequencies 0, 1/M, ..., (M-1)/M, 1).
#------------------------------------------------------------------------
{
  u  <- (0:M) / M
  z1 <- u[which(u <= 0.5)]
  z2 <- u[which(u > 0.5)]
  z1 <- 1 - 6*z1*(z1 - z1*z1)
  z2 <- 2*(1 - z2)^3

  return(c(z1,z2))
}
```

4.7 A More Advanced Approach

As you may expect, the results in this chapter have a more mathematical basis than provided earlier. In this section, we will provide a more formal statement of many of those results. We will begin by providing a matrix representation of the autocovariance function of a second-order stationary time series. We then expand our explanation of the relationships between the autocovariance function and the spectral density function and why a series can be represented using a spectral distribution in the "Spectral Representation Theorem." The last theorem in this section is the "Univariate Prediction Theorem", which gives mathematical expressions for time series predictors as well as the formal definition and a recursion for finding the partial autocorrelation function.

We begin by providing an extremely useful matrix representation of the autocovariance function of a second-order stationary time series. To begin, we let $\boldsymbol{X}_n = (X_1, \ldots, X_n)^T$ represent a finite realization from a covariance stationary time series $\{X_t, t \in \mathcal{Z}\}$ having autocovariance function R_v, then the covariance matrix of $\boldsymbol{X}_n$ can be written in a matrix having the special form

$$\operatorname{Var}(\boldsymbol{X}_n) = \begin{bmatrix} R_0 & R_1 & R_2 & \cdots & \cdots & R_{n-1} \\ R_1 & R_0 & R_1 & \ddots & & \vdots \\ R_2 & R_1 & R_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & R_2 \\ \vdots & & \ddots & \ddots & \ddots & R_1 \\ R_{n-1} & \cdots & \cdots & R_2 & R_1 & R_0 \end{bmatrix},$$

where the diagonal rows of dots indicate that each element of a particular diagonal is the same; that is, the $(j,k)^{\text{th}}$ element of the matrix is $R_{|j-k|}$. Such matrices are said to be **symmetric Toeplitz** and they have been extensively studied (see Grenander & Szegö (1958) and Example 4.4 for example). An $n \times n$ symmetric Toeplitz matrix has only n distinct elements, namely the elements $R_0, \ldots, R_{n-1}$ that are in its first row. We will use the notation

$$\boldsymbol{\Gamma}_n = \operatorname{Toepl}(R_0, \ldots, R_{n-1})$$

for such a matrix. Sometimes we will consider the correlation matrix of $\boldsymbol{X}_n$; that is

$$\boldsymbol{\Phi}_n = \operatorname{Toepl}(1, \rho_1, \ldots, \rho_{n-1}).$$

Timeslab in R has a function (`toepl`) that will form a Toeplitz matrix (see Section 4.5).

Example 4.4 *There are situations in time series analysis where a knowledge of the eigenvalues of an $n \times n$ covariance matrix is useful (see Part (g) of Theorem A.6.2). In this example, we illustrate the fact that these eigenvalues are related to the values of the spectral density function at the n natural frequencies $(j-1)/n$ for $j = 1, 2, \ldots, n$.*

Let R and f be the autocovariance and spectral density functions of a covariance stationary time series and let

$$\lambda_{n,1} \le \lambda_{n,2} \le \cdots \le \lambda_{n,n}$$

be the eigenvalues of $\boldsymbol{\Gamma}_n = Toepl(R_0, R_1, \ldots, R_{n-1})$ arranged in nonincreasing order. Suppose that m is the largest number such that $f(\omega) \ge m$ for all ω and M is the smallest number such that $f(\omega) \le M$ for all ω. Suppose that $M < \infty$ and let h be any continuous function defined on the interval $[m, M]$. Let

$$f_{(1)} \ge f_{(2)} \ge \cdots \ge f_{(n)}$$

be the values of f (arranged in nonincreasing order) at the frequencies $(j-1)/n$ for $j = 1, \ldots, n$. Then Grenander & Szegö (1958) show that, for a continuous function $h(\cdot)$,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{v=1}^{n} \Big[h(\lambda_{n,v}) - h(f_{(v)}) \Big] = 0,$$

and

$$\lim_{n\to\infty}\frac{1}{n}\sum_{v=1}^{n}h(\lambda_{n,v}) = \lim_{n\to\infty}\frac{1}{n}\sum_{v=1}^{n}h(f_{(v)}) = \int_0^1 h(f(\omega))\,d\omega.$$

Thus if h is the identity function, we have that the eigenvalues and the sorted values of the spectral density are essentially the same for large n. Further, the integral of any continuous function of the spectral density f (for example $h(x) = x^2$ or $h(x) = \log(x)$) is also related to the eigenvalues of $\mathbf{\Gamma}_n$.

The function `ar.eigen` *will create and list the eigenvalues and sorted values of the spectral density for a user-specified AR(p) process.*

```
ar.eigen <- function(alpha, n = length(alpha)+1, rvar = 1)
#-----------------------------------------------------------------------
#  An R function to compute the n eigenvalues of covariance matrix and
#  sorted spectral density function of an AR(p) process with
#  coefficients as specified by alpha.
#
#  INPUT: alpha = a real vector containing the coefficients of the AR
#                 process
#         n = an integer scalar containing the number of eigenvalues
#             and spectral density values to compute.  The value
#             of n must be at least length(alpha) + 1.  The default
#             value is n = length(alpha) + 1.  (Note that the maximum
#             lag will be n - 1, and the dimension of the Toeplitz
#             matrix will by n x n.)
#         rvar = a real scalar containing the variance of the errors.
#                The default value is rvar = 1.
#
#  VALUES: The function ar.eigen returns a list containing the following
#          objects.
#          eigenvals = a vector of length n containing the eigenvalues
#                      of the AR(p) process if the process is
#                      covariance stationary.  If not, eigenvals is NULL.
#          sorted.f = a vector of length n containing the sorted values
#                     of the spectral density function if the process
#                     is second-order stationary.  If the process is
#                     not stationary, sorted.f = NULL.
#          error = an integer indicator.  If error = 0, then the
#                  process is second-order stationary.  Otherwise
#                  error = 1.
#-----------------------------------------------------------------------
{
  if(n < length(alpha)) n <- length(alpha) + 1
  m <- n - 1
  cc <- arcorr(alpha, m, rvar)
  error <- cc$error
  if(error == 0) {
    R0 <- cc$var
    R  <- cc$corr * R0
    evals <- eigen(toepl(R, R0), only.values = TRUE,
      symmetric = TRUE)$values

    f1 <- arsp(alpha, rvar, n)
    f  <- sort(c(f1[1:(length(f1)-1)], f1[1:(length(f1)-1)]),
      decreasing = TRUE)

  }
  if(error != 0) {
    evals <- NULL
    f     <- NULL
  }

  return(list(eigenvals = evals, sorted.f = f, error = cc$error))

}
```

To illustrate the use of this function, Figure 4.5 gives the result of using it for the autoregressive process of order 2 process having coefficients 0.3 and 0.9. (See Figure 4.2 for a plot of the log of the spectral density of this process and Section 5.4 for more information about autoregressive processes.) The function find the eigenvalues of the 40×40 covariance matrix of the process and the spectral density function of the process across the entire [0, 1] interval of frequencies. Notice the close agreement between the eigenvalues and the spectral density function except for the eight largest values. If the number of values computed is increased, to 300 say, the agreement would be even closer for all values.

The next result we provide in this section is an extremely important theorem in time series analysis. It summarizes some of the basic mathematical facts about the autocovariance function R, and introduces two functions that are mathematical equivalents of R, but which are also important in their own right in both theory and in practice. Some of the mathematics in the theorem is quite advanced, and are summarized in Appendix A.2. The interested reader can also consult Bloomfield (1976). A discussion of the practical implications of this theorem was given in Section 4.2.1. A partial proof is provided in Appendix B.3.

Theorem 4.2 Two Spectral Representations.
Let $\{R_v, v \in \mathcal{Z}\}$ be the autocovariance function of the covariance stationary time series X. Then

(a) $R_v = R_{-v}$ for all $v \in \mathcal{Z}$.

```
> ar.eigen(alpha, 40)

Eigenvalues of AR Autocovariance Matrix:
 [1] 60.7497879 60.5895712 21.6894598 21.5073789  8.9790412  8.8375070
 [7]  4.6575636  4.5378611  2.8297989  2.7212789  1.9125721  1.8113270
[13]  1.3929530  1.2985945  1.0711199  0.9861476  0.8567598  0.7874230
[19]  0.7034641  0.6589503  0.5866981  0.5730327  0.5124422  0.4972994
[25]  0.4674119  0.4422248  0.4212451  0.4173799  0.4017665  0.3935355
[31]  0.3695892  0.3288085  0.2968276  0.2717827  0.2522049  0.2370204
[37]  0.2254539  0.2169515  0.2111276  0.2077289

Values of f sorted in descending order:
 [1] 102.4277369 102.4277369  10.9301519  10.9301519  10.0000000  10.0000000
 [7]   3.0822589   3.0822589   2.7290077   2.7290077   1.4843529   1.4843529
[13]   1.2562044   1.2562044   0.9141618   0.9141618   0.7355452   0.7355452
[19]   0.6519112   0.6519112   0.5148625   0.5148625   0.4965626   0.4965626
[25]   0.4401358   0.4401358   0.4022628   0.4022628   0.3695353   0.3695353
[31]   0.2959885   0.2959885   0.2516508   0.2516508   0.2252032   0.2252032
[37]   0.2110655   0.2110655   0.2066116   0.2066116
```

Figure 4.5: An example of the relationship between the eigenvalues of a Toeplitz matrix and the corresponding spectral density function.

(b) R_v is a semipositive definite sequence; that is, for any positive integer n and any real numbers $a_1, \ldots, a_n$, not all zero, and any n time points $t_1, \ldots, t_n$, we have

$$S = \sum_{j=1}^{n}\sum_{k=1}^{n} a_j a_k R_{t_j - t_k} \geq 0.$$

(c) Spectral Representation of R. *There exists a function $F(\omega)$, $\omega \in [0, 1]$, called the* **spectral distribution function of** X *such that*

$$R_v = \int_0^1 \cos(2\pi v\omega)\ dF(\omega), \quad v \in \mathcal{Z},$$

where the integral is a Lebesgue-Stieltjes integral (see Appendix A.2.2 and Theorem A.2.4). Further, $F(0) = 0$, $F(1) = R_0$, and at all continuity points of F, $F(\omega) = R_0 - F(1-\omega)$.

(d) Spectral Representation of X. *There exists two uncorrelated stochastic processes $C = \{C(\omega), \omega \in [0, 1]\}$ and $S = \{S(\omega), \omega \in [0, 1]\}$ having stationary uncorrelated increments (see Appendix A.2.3) such that X_t can be written as the stochastic integral*

$$X_t = \int_0^1 \cos(2\pi t\omega)\ dC(\omega) + \int_0^1 \sin(2\pi t\omega)\ dS(\omega),$$

and for $0 \leq \omega_1 \leq \omega_1 \leq 1$,

$$Var(C(\omega_2) - C(\omega_1)) = Var(S(\omega_2) - S(\omega_1)) = F(\omega_2) - F(\omega_1)$$

(e) *If* R *is absolutely summable, that is* $\sum_{v=-\infty}^{\infty} |R_v| < \infty$, *then* F *is absolutely continuous; that is there exists a function* $f(\omega)$, $\omega \in [0, 1]$, *symmetric about* $\omega = 0.5$ *called the* **spectral density function of** X *such that*

$$F(\omega) = \int_0^\omega f(x)\, dx, \quad f(\omega) = \frac{dF(\omega)}{d\omega}, \quad \textit{and} \quad R_v = \int_0^1 f(\omega) \cos(2\pi v\omega)\, d\omega, \quad v \in \mathcal{Z}.$$

Furthermore f *is continuous and*

$$f(\omega) = \sum_{v=-\infty}^{\infty} R_v \cos(2\pi v\omega), \quad \omega \in [0, 1].$$

(f) *The equations relating* R *and* f *can also be written as*

$$\begin{aligned} R_v &= \int_0^1 f(\omega) e^{2\pi i v\omega}\, d\omega \\ f(\omega) &= \sum_{v=-\infty}^{\infty} R_v e^{-2\pi i v\omega} \\ &= R_0 + 2\sum_{v=1}^{\infty} R_v \cos(2\pi v\omega). \end{aligned}$$

(g) *If* $f(\omega) \geq \lambda > 0$ *for each* $\omega \in [0, 1]$, *then* R *is positive definite; that is, the* $\geq$ *in part (b) can be replaced by strict inequality* ($>$).

The last result in this section is the Univariate Prediction Theorem, discussed in Section 4.4. Before stating it, we introduce a special matrix called the "permutation matrix."

Definition 4.11 Permutation Matrix.
The $n \times n$ *matrix* $\boldsymbol{P}_n$ *consisting of all zeros except for ones on the main reverse diagonal is called the* **permutation matrix of order** n.

We use the permutation matrix because many of the linear combinations that we use are of the form

$$\ell = a_1 x_n + a_2 x_{n-1} + \cdots + a_n x_1 = \boldsymbol{a}^T \boldsymbol{P}_n \boldsymbol{x},$$

where $\boldsymbol{a} = (a_1, \ldots, a_n)^T$ and $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, since pre (post) multiplication of a matrix by the permutation matrix reverses the order of the rows (columns) of that matrix.

There are two other important features of the permutation matrix. First $\boldsymbol{P}_n^2 = \boldsymbol{I}_n$, the $n \times n$ identity matrix. Secondly, if $\boldsymbol{\Gamma}_n$ is the $n \times n$ symmetric Toeplitz matrix (see Section 4.7), then

$$\boldsymbol{\Gamma}_n = \boldsymbol{P}_n \boldsymbol{\Gamma}_n \boldsymbol{P}_n, \quad \text{and} \quad \boldsymbol{\Gamma}_n^{-1} = \boldsymbol{P}_n \boldsymbol{\Gamma}_n^{-1} \boldsymbol{P}_n.$$

These relationships are the topic of Exercise T4.9. With these facts in mind, we now present some of the basic facts about univariate time series prediction.

Theorem 4.3 Univariate Prediction.
Let X be a zero mean time series. Then

(a) the best unbiased predictor and its error variance are given by

$$\begin{aligned}\tilde{X}_{nh} &= E[X_{n+h} \,|\, X_1, \ldots, X_n] \\ \tilde{\sigma}^2_{nh} &= Var(X_{n+h} \,|\, X_1, \ldots, X_n)\,.\end{aligned}$$

(b) If X is covariance stationary with autocovariance function R, then the best unbiased linear predictor and its error variance are given by

$$\hat{X}_{nh} = \boldsymbol{\lambda}^T_{nh}\boldsymbol{P}_n\boldsymbol{X}_n$$
$$\hat{\sigma}^2_{nh} = R_0 - \boldsymbol{r}^T_{nh}\boldsymbol{\Gamma}^{-1}_n\boldsymbol{r}_{nh},$$

where $\boldsymbol{X}_n = (X_1, \ldots, X_n)^T$, and the prediction coefficients

$$\boldsymbol{\lambda}_{nh} = (\lambda_{nh,1}, \ldots, \lambda_{nh,n})^T$$

satisfy the prediction normal equations

$$\boldsymbol{\Gamma}_n\boldsymbol{\lambda}_{nh} = \boldsymbol{r}_{nh},$$

where $\boldsymbol{\Gamma}_n = Toepl(R_0, R_1, \ldots, R_{n-1})$ and $\boldsymbol{r}_{nh} = (R_h, \ldots, R_{h+n-1})^T$. Further, these predictors and prediction error variances can be found using conditional means and conditional variances as in part (a) but for a Gaussian time series have the same autocovariance function as X.

(c) Let $\lambda_{n,j}$ and $\hat{\sigma}^2_n$ denote the coefficients and prediction error variances for the best unbiased linear one step ahead predictor. Then the $\lambda_{n,j}$ and $\hat{\sigma}^2_n$ satisfy Levinson's recursion:

$$\begin{aligned}\lambda_{j+1,j+1} &= \frac{R_{j+1} - \sum_{k=1}^{j}\lambda_{j,k}R_{j+1-k}}{\hat{\sigma}^2_j} \\ \lambda_{j+1,k} &= \lambda_{j,k} - \lambda_{j+1,j+1}\lambda_{j,j+1-k}, \quad k = 1, \ldots, j \\ \hat{\sigma}^2_{j+1} &= \hat{\sigma}^2_j\,(1 - \lambda_{j+1,j+1})\,,\end{aligned}$$

with $\lambda_{1,1} = \rho_1$ and $\hat{\sigma}^2_0 = R_0$. Further, for $k > 1, \lambda_{k,k}$ is equal to the correlation between the errors in predicting X_t from $X_{t+1}, \ldots, X_{t+k-1}$ and predicting X_{t+k} from $X_{t+1}, \ldots, X_{t+k-1}$. Thus $\lambda_{k,k}$ is called the **partial autocorrelation coefficient of lag** *k.*

(d) If X is covariance stationary and

$$\lim_{n\to\infty} \hat{\sigma}^2_{n1} = \hat{\sigma}^2_\infty > 0,$$

and X is said to be **purely nondeterministic**, *and*

(i) *There exists a white noise time series ε having variance σ^2_∞ and an infinite sequence of constants $\gamma_0 = 1, \gamma_1, \gamma_2, \ldots$, such that, as a limit in mean square*

$$X_t = \sum_{k=1}^{\infty} \gamma_k \varepsilon_{t-k}.$$

This is called the **infinite order moving average representation of** X*, and the εs are called the* **innovations** *of the process.*

(ii) *The* **infinite memory predictor** X_{nh} *exists and is given by, as a limit in mean square*

$$X_{nh} = \sum_{k=h}^{\infty} \gamma_k \varepsilon_{n+h-k},$$

while

$$\sigma^2_{nh} = \sigma^2_\infty \sum_{k=1}^{h-1} \gamma_k^2.$$

Further, for $v \geq 0$,

$$\sigma_{hv} = Cov(X_{nh},\, X_{n,h+v}) = \sigma^2_\infty \sum_{j=h}^{\infty} \gamma_h \gamma_{j+v} = R_v - \sigma^2_\infty \sum_{j=0}^{h-1} \gamma_j \gamma_{j+v}.$$

(iii) *A sufficient condition for X to be purely nondeterministic is that X have a spectral density function f satisfying*

$$S = \int_0^1 \log\left[f(\omega)\right]\, d\omega > -\infty,$$

in which case

$$\sigma^2_\infty = e^S.$$

(e) Wold Decomposition. *Any covariance stationary time series X can be written as*

$$X_t = U_t + V_t$$

where U and V are uncorrelated with each other, U is purely nondeterministic and V is purely deterministic.

4.8 Exercises

4.8.1 Theoretical Exercises

T4.1 A time series X is said to be **strictly stationary** if for any positive integer n and any $n+1$ elements $t_1, \ldots, t_n$ and h of the index set T, the joint distributions of

$$X_{t_1}, \ldots, X_{t_n} \quad \text{and} \quad X_{t_1+h}, \ldots, X_{t_n+h}$$

are the same. Show (a) that a strictly stationary series is covariance stationary, and (b) that a covariance stationary process is strictly stationary. (*Hint*: See Appendix A.6.)

T4.2 Show that the sum and difference of two uncorrelated covariance stationary time series is also covariance stationary.

T4.3 We have defined the spectral density function on the interval [0, 1]. Many authors define it on the interval $[\pi, \pi]$ as in the following. Let R be the autocovariance function of a covariance stationary time series X and suppose that the spectral density function of X exists. Show that we can write

$$R_v = \int_{-\pi}^{\pi} f(\lambda) e^{iv\lambda}\, d\lambda, \quad v \in \mathcal{Z},$$

where

$$f(\lambda) = \frac{1}{2\pi} \sum_{v=-\infty}^{\infty} R_v e^{-iv\lambda}, \quad \lambda \in [-\pi, \pi].$$

T4.4 Prove part (c) of the Univariate Filter Theorem.

T4.5 Show that the following are true for integers j and k.

$$\begin{aligned} \min(j,k) &= \frac{(j+k) - |j-k|}{2} \\ \max(j,k) &= \frac{(j+k) + |j-k|}{2} \end{aligned}$$

T4.6 Show that the Dirichlet kernel

$$D_M(\omega) = \sum_{j=-M}^{M} e^{2\pi i j \omega}$$

discussed in Section 4.3.2 can be written as

$$D_M(\omega) = \frac{\sin(\{M+0.5\}2\pi\omega)}{\sin(\pi\omega)}.$$

(*Hint:* Use a geometric series.)

T4.7 Verify that the expressions for the coefficients of the bandpass filter example in Section 4.3.2 are correct.

T4.8 Let $\{X_n, n = 1, 2, \ldots\}$ be a sequence of random variables. Use the Continuous Function Theorem (Theorem A.6.6) to show that if $X_n \sim AN(\mu, b_n\mu^2)$ where $b_n \to 0$, then $\log X_n \sim AN(\log\mu, c)$ where c does not depend on μ.

T4.9 Show that if $\boldsymbol{P}_n$ is the $n \times n$ permutation matrix, and $\boldsymbol{\Gamma}_n$ is an $n \times n$ symmetric Toeplitz matrix, then

$$\boldsymbol{\Gamma}_n = \boldsymbol{P}_n \boldsymbol{\Gamma}_n \boldsymbol{P}_n, \quad \text{and} \quad \boldsymbol{\Gamma}_n^{-1} = \boldsymbol{P}_n \boldsymbol{\Gamma}_n^{-1} \boldsymbol{P}_n.$$

(*Hint:* Use the fact that $\boldsymbol{P}_n^2 = \boldsymbol{I}_n$.)

T4.10 Use the Univariate Filter Theorem (Theorem 4.1) to find the autocorrelation function and spectral density function for the linear filters in Exercise C4.4. The models that result from that filter are known as moving average models. We will learn more about moving average models in Section 5.3.

T4.11 For $t = 1, 2, \ldots, n$, let $X_t = a + bt + \varepsilon_t$, where $\varepsilon \sim \text{WN}(\sigma^2)$.

(a) Find an expression for the mean function of X_t.

(b) Is X_t covariance stationary?

(c) Can you find an expression for the autocorrelation function of X_t? If so, give it a try.

(d) Can you ind an expression for the spectral density function of X_t? if so, give it a try.

Now, let $Y_t = X_t - X_{t-1}$. Repeat parts (a) through (d) above for the first difference. What is the effect of taking the first difference of a linear trend on the mean function, the second-order stationarity of the process, the autocorrelation function and spectral density function?

4.8.2 Computational Exercises

C4.1 Write a function that will create the $n \times n$ permutation matrix.

C4.2 Write a function that will generate a specified number `nreps` of Gaussian white noise series, each of length `n`, with variance `sig2`. The function you write should do the following.

(1) For each of the series compute the sample autocorrelation function for `m` lags. Superimpose all correlograms on the same set of axes along with a reference line at 0.

(2) Compute and return the mean value of the sample correlogram across all series at each lag.

Use the definition and properties of a white noise series, along with the graph and numerical output from your function to answer these questions for sample sizes `n` $=$ 30, 75, 250. In answering the questions, let `m` $= 25$ and `nreps` $=$ `20`.

(a) What behavior do you expect to observe for the collection of sample correlograms? Around what value do you think they'll be centered? Why?

(b) Describe what you observe in the graph in terms of the overall behavior of all the sample correlograms. Did they behave as you expected.

(c) As the length of the series `n` increases, do the values of $\hat{\rho}_v$ come closer to the true value of ρ_v? Is the variability between the $\hat{\rho}_v$ remaining the same with increasing `n`, or changing? How is it changing?

(d) If you know the definition of an *consistent estimator*, answer this: "For each value of v, does it appear that $\hat{\rho}_v$ is a consistent estimator of ρ_v?

C4.3 Write a function that will generate a specified number `nreps` of Gaussian white noise series, each of length `n`, with variance `sig2`. The function you write should do the following.

(1) For each of the series compute the periodogram. Superimpose all periodograms on the same set of axes, along with a reference line representing the value of the variance `sig2` of the series.

(2) Compute and return the mean value of the (raw) periodogram across all series at each natural frequency.

Use the definition and properties of a white noise series, along with the graph and numerical output from your function to answer these questions for sample sizes `n` $= 30, 75, 250$. In answering the questions `nreps` $=$ `20`.

(a) What behavior do you expect to observe for the collection of periodograms? Around what value do you think they'll be centered? Why?

(b) Describe what you observe in the graph in terms of the overall behavior of all the periodograms. Did they behave as you expected.

(c) As the length of the series `n` increases, do the values of $\hat{f}(\omega_j)$ come closer to the true value of $f(\omega_j)$? Is the variability between the $\hat{f}(\omega)$ remaining the same with increasing `n`, or changing? How is it changing?

(d) If you know the definition of an *consistent estimator*, answer this: "For each value of ω_j, does it appear that $\hat{f}(\omega_j)$ is a consistent estimator of $f(\omega_j)$?

C4.4 Write a function that will create a filtered version of white noise using the `filt` function. The lead coefficient `beta0` should always equal 1. For the filtered version, the function should create a 2×2 array of plots containing graphs of the descriptive statistics as seen in Figure 2.9 from Example 2.7. Create these graphs for the following sets of coefficients:

(a) `beta` $=$ `0.9`

(b) `beta` $= -$`0.9`

(c) `beta` $=$ `c(0, 0, 0, 0.8)`

(d) `beta` $=$ `c(`$-$`0.70, `$-$`0.10, 0.60)`

Describe the behavior of each of the descriptive statistics for each set of coefficients for `n = 200` and `m = 30`. If you completed Exercise T4.10, do the behaviors match the expressions you obtained in that exercise? The structure that results from creating a linear filter of white noise is known as a "moving average model." We will learn more about moving average models in Section 5.3.

C4.5 Write an *R* function that accomplishes the tasks listed immediately below.

(1) Generate a series from a linear trend (see Section 3.1.2). Allow the user to specify the intercept `a`, slope `b`, error variance `sig2`, and the length of the series `n`.

(2) Compute the first difference of the linear trend.

(3) For the original series and the first difference, compute the sample autocorrelation function and the periodogram.

(4) Create a 3 × 2 plotting array. Plot the original series, its sample autocorrelation function, and periodogram in the left column of plots, and the corresponding plots for the first difference in the right column.

Use the result from your function to answer the questions that follow. For an intercept of `a = 1`, a slope of `b = 2`, an error variance of `sig2 = 0.25` and a series length of `n = 200`, run the function. What is the result of taking the first difference on the series, the sample autocorrelation function, and the periodogram.