Chapter 9: Series Variables

Overview

Series variables have a number of characteristics that are sufficiently different from other types of variables that they need examining in more detail. Series variables are always at least two-dimensional, although one of the dimensions may be implicit. The most common type of series variable is a *time series*, in which a series of values of some feature or event are recorded over a period of time. The series may consist of only a list of measurements, giving the appearance of a single dimension, but the ordering is by time, which, for a time series, is the implicit variable.

The series values are always measured on one of the scales already discussed, nominal through ratio, and are presented as an ordered list. It is the ordering, the expression of the implied variable, that requires series data to be prepared for mining using techniques in addition to those discussed for nonseries data. Without these additional techniques the miner will not be able to best expose the available information. This is because series variables carry additional information within the ordering that is not exposed by the techniques discussed so far.

Up to this point in the book we have developed precise descriptions of features of nonseries data and various methods for manipulating the identified features to expose information content. This chapter does the same for series data and so has two main tasks:

1. Find unambiguous ways to describe the component features of a series data set so that it can be accurately and completely characterized

2. Find methods for manipulating the unique features of series data to expose the information content to mining tools

Series data has features that require more involvement by the miner in the preparation process than for nonseries data. Where miner involvement is required, fully automated preparation tools cannot be used. The miner just has to be involved in the preparation and exercise judgment and experience. Much of the preparation requires visualizing the data set and manipulating the series features discussed. There are a number of excellent commercial tools for series data visualization and manipulation, so the demonstration software does not include support for these functions. Thus, instead of implementation notes concluding the chapter discussing how the features discussed in the chapter are put into practice, this chapter concludes with a suggested checklist of actions for preparing series data for the miner to use.

9.1 Here There Be Dragons!

Mariners and explorers of old used fanciful and not always adequate maps. In unexplored or unknown territory, the map warned of dragons—terrors of the unknown. So it is when preparing data, for the miner knows at least some of the territory. Many data explorers have passed this way. A road exists. Signposts point the way. Maybe the dragons were chased away, but still be warned. “Danger, quicksand!” Trouble lurks inside series data; the road of data preparation is rocky and uncertain, sometimes ending mired in difficulties. It is all too easy to seriously damage data, render it useless, or worse, create wonderful-looking distortions that are but chimera that melt away when exposed to the bright light of reality. Like all explorers faced with uncertainty, the miner needs to exercise care and experience here more than elsewhere. The road is rough and not always well marked. Unfortunately, the existing signposts, with the best of intentions, can still lead the miner seriously astray. Tread this path with caution!

9.2 Types of Series

Nonseries multivariable measurements are taken without any particular note of their ordering. Ordering is a critical feature of a series. Unless ordered, it’s not a series. One of the variables (called the displacement variable, and described in a moment) is always monotonic—either constantly increasing or constantly decreasing. Whether there is one or several other variables in the series, their measurements are taken at defined points on the range of the monotonic variable. The key ordering feature is the change in the monotonic variable as its values change across part or all of its range.

Time series are by far the most common type of series. Measurements of one variable are taken at different times and ordered such that an earlier measurement always comes before a later measurement. For a time series, time is the *displacement variable*—the measurements of the other variable (or variables) are made as time is “displaced,” or changed. The displacement variable is also called the *index* variable. That is because the points along the displacement variable at which the measurements are taken are called the *index points*.

Dimensions other than time can serve as the displacement dimension. Distance, for instance, can be used. For example, measuring the height of the American continent above sea level at different points on a line extending from the Atlantic to the Pacific produces a distance displacement series.

Since time series are the most common series, where this chapter makes assumptions, a time series will be assumed. The issues and techniques described about time series also apply to any other displacement series. Series, however indexed, share many features in common, and techniques that apply to one type of series usually apply to other types of series. Although the exact nature of the displacement variable may make little difference to the preparation and even, to some degree, the analysis of the series itself, it makes all the

difference to the interpretation of the result!

9.3 Describing Series Data

Series data differs from the forms of data so far discussed mainly in the way in which the data enfolds the information. The main difference is that the ordering of the data carries information. This ordering, naturally, precludes random sampling since random sampling deliberately avoids, and actually destroys, any ordering. Preserving the ordering is the main reason that series data has to be prepared differently from nonseries data.

There is a large difference between preparing data for modeling and actually modeling the data. This book focuses almost entirely on how to prepare the data for modeling, leaving aside almost all of the issues about the actual modeling, insofar as is practical. The same approach will apply to series data. Some of the tools needed to address the data preparation problems may look similar, indeed are similar, to those used to model and glean information and insight into series data. However, they are put to different purposes when preparing data. That said, in order to understand some of the potential problems and how to address them, some precise method of describing a series is needed. A key question is, What are the features of series data?

To answer this question, the chapter will first identify some consistent, recognizable, and useful features of series data. The features described have to be consistent and recognizable as well as useful. The useful features are those that best help the miner in preparing series data for modeling. The miner also needs these same features when modeling. This is not surprising, as finding the best way to expose the features of interest for modeling is the main objective of data preparation.

9.3.1 Constructing a Series

A series is constructed by measuring and recording a feature of an object or event at defined index points on a displacement dimension.

This statement sufficiently identifies a series for mining purposes. It is not a formal definition but a conceptual description, which also includes the following assumptions:

1. The feature or event is recorded as numerical information.

2. The index point information is either recorded, or at least the displacements are defined.

3. The index, if recorded, is recorded numerically.

It is quite possible to record a time series using alpha labels for the nondisplacement dimension, but this is extremely rare. Numerating such alpha values within the series is

possible, although it requires extremely complex methods. While it is very unusual indeed to encounter series with one alpha dimension, it is practically unknown to find a series with an alpha-denominated displacement variable. The displacement dimension has to be at least an ordinal variable (ratio more likely), and these are invariably numerical. Because series with all dimensions numerical are so prevalent, we will focus entirely on those.

It is also quite possible to record multivariable series sharing a common displacement variable, in other words, capturing several features or events at each index mark. An example is collecting figures for sales, backlog, new orders, and inventory level every week. “Time” is the displacement variable for all the measurements, and the index point is weekly. The index point corresponds to the validating event referred to in Chapter 2. There is no reason at all why several features should not be captured at each index, the same as in any nonseries multidimensional data set. However, just as each of the variables can be considered separately from each other during much of the nonseries data preparation process, so too can each series variable in a multidimensional series be considered separately during preparation.

9.3.2 Features of a Series

By its nature a series has some implicit pattern within the ordering. That pattern may repeat itself over a period. Often, time series are thought of by default as repetitive, or cyclic, but there is no reason that any repeating pattern should in fact exist. There is, for example, a continuing debate about whether the stock market exhibits a repetitive pattern or is simply the result of a random walk (touched on later). Enormous effort has been put into detecting any cyclic pattern that may exist, and still the debate continues. There is, nonetheless, a pattern in series data, albeit not necessarily a repeating one. One of the objectives of analyzing series data is to describe that pattern, identify it as recognizable if possible, and find any parts that are repetitive. Preparing series data for modeling, then, must preserve the nature of the pattern that exists. Preparation also includes putting the data into a form in which the desired information is best exposed to a modeling tool. Once again, a warning: this is not always easy!

Before looking at how series data may be prepared, and what problems may be detected and corrected, the focus now turns to finding some way to unambiguously describe the series.

9.3.3 Describing a Series—Fourier

Jean Baptiste Joseph Fourier was not a professional mathematician. Nonetheless, he exerted an influence on mathematicians and scientists of his day second only to that of Sir Isaac Newton. Until Fourier revealed new tools for analyzing data, several scientists lamented that the power of mathematics seemed to be just about exhausted. His insights reinvigorated the field. Such is the power of Fourier’s insight that its impact continues to

reverberate in the modern world today. Indeed, Fourier provided the key insights and methods for developing the tools responsible for building the modern technology that we take for granted.

To be fair, the techniques today brought under the umbrella description of Fourier analysis were not all entirely due to Fourier. He drew on the work of others, and subsequent work enormously extends his original insight. His name remains linked to these techniques, and deservedly so, because he had the key insights from which all else flowed.

One of the tools he devised is a sort of mathematical prism. Newton used a prism to discover that white light consists of component parts (colors). Fourier's prism scatters the information in a series into component parts. It is a truly amazing device that hinges on two insights: waves can be added together, and adding enough simple sine and cosine waves of different frequencies, phases, and amplitudes together is sufficient to create any series shape. *Any* series shape!

When adding waveforms together, several things can be varied. The three key items are

- The frequency, or how many times a waveform repeats its pattern in a given time
- The phase, that is, where the peaks and troughs of a wave occur in relation to peaks and troughs of other waves
- The amplitude, or distance between the highest and lowest values of a wave

Figure 9.1 shows how two waveforms can be added together to produce a third. The frequency simply measures how many waves, or cycles, occur in a given time. The top two waveforms are symmetrical and uniform. It is easy to see where they begin to repeat the previous pattern. The two top waveforms also have different frequencies, which is shown by the identified wavelengths tracing out different lengths on the graph. The lower, composite waveform cannot, just by looking at it, positively be determined to have completed a repeating pattern in the width of the graph.

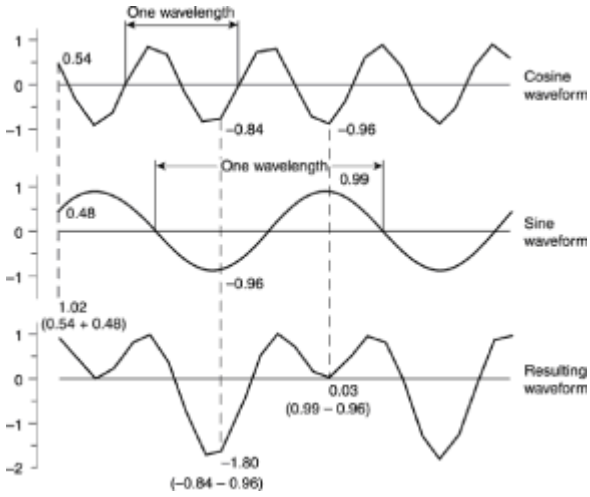

Figure 9.1 The addition of two waveforms shows how to create a new waveform. The values of the consine and sine waveforms are added together and the result plotted. The resulting wave-form may look nothing like the source that created it.

Figure 9.2 shows two waveforms that are both complete cycles, and both are identical in length. The waveforms illustrate the sine and the cosine functions:

$y = \text{sine}(x^{\circ})$

and

$y = \text{cosine}(x^{\circ})$

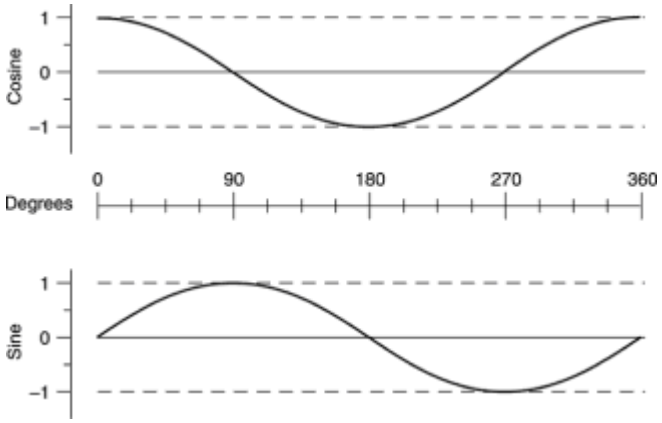

Figure 9.2 Values of the functions “sine” and “cosine” plotted for the number of degrees shown on the central viniculum.

When used in basic trigonometry, both of these functions return specific values for any

number of degrees. They are shown plotted over 360°, the range of a circle. Because 0° represents the same circular position as 360° (both are due north on a compass, for example), this has to represent one complete cycle for sine and cosine waveforms—they begin an identical repetition after that point. Looking at the two waveforms shows that the sine has identical values to the cosine, but occurring 90° later (further to the right on the graph). The sine is an identical waveform to the cosine, shifted 90°. “Shifted” here literally means moved to the right by a distance corresponding to 90°. This shift is called a *phase shift*, and the two waveforms are said to be 90° out of phase with each other.

The two upper images in Figure 9.3 show the effect of changing amplitude. Six sine and cosine waveforms, three of each, are added together. The frequencies of each corresponding waveform in the two upper images are identical. All that has changed is the amplitude of each of the waveforms. This makes a very considerable difference to the resulting waveform shown at the bottom of each image. The lower two images show the amplitudes held constant, but the frequency of each contributing waveform differs. The resulting waveforms—the lower waveform of each frame—show very considerable differences.

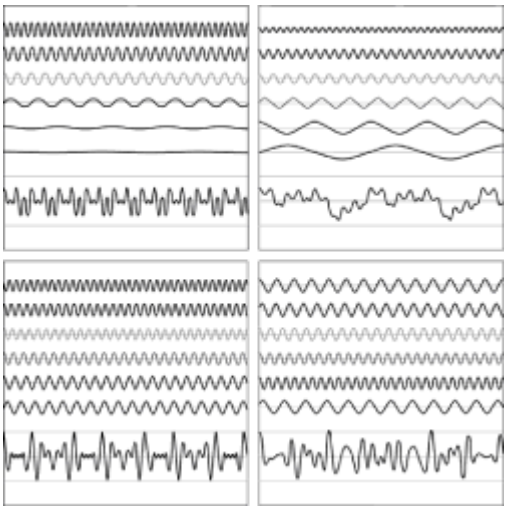

Figure 9.3 These four images show the result of summing six waveforms. In the top two images, the frequencies of the source waveforms are the same; only their amplitude differs. In both of the two lower images, all waveforms have similar amplitude.

It was Fourier’s insight that by combining enough of these two types of waveforms, varying their amplitude, phase, and frequency as needed, any desired resultant waveform can be built. Fourier analysis is the “prism” that takes in a complex waveform and “splits” it into its component parts—just as a crystal prism takes in white light and splits it into the various colors. And just as there is only one rainbow of colors, so too, for any specific input waveform, there is a single “rainbow” of outputs.

A Fourier analysis provides one way of uniquely describing a series. In Figure 9.4, Fourier analysis illustrates the prism effect—splitting a composite waveform into components. Also in Figure 9.4, Fourier synthesis shows how the reverse effect works—reassembling the waveform from components.

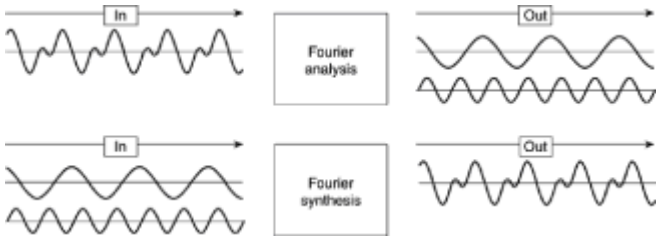

Figure 9.4 Fourier analysis takes in a complex waveform and yields an expression giving the appropriate component sine and cosine expressions, together with their amplitude, frequency, and phase, to re-create the analyzed waveform. Fourier synthesis takes the analyzed expression and yeilds the composite waveform.

9.3.4 Describing a Series—Spectrum

A *spectrum* is normally thought of as an array of colors—"the colors of the spectrum." For light, that exactly describes a spectrum, but an infinite variety of different spectra exist. When sunlight passes through a prism, it breaks into a band showing the array of all colors possible from white light. If, however, a colored light beam passes through a prism, the resulting spectrum does not show all of the possible colors equally brightly. Depending on the exact color of the original colored light beam, all of the possible colors may be present. The brightest intensity of color in the spectrum will correspond to the apparent color of the incoming beam. Because light is a form of energy, the energy of the brightest portion of the spectrum contains the most energy. For light, a spectrum shows the energy distribution of an incoming light beam—brightest color, highest energy.

Fourier analysis also allows a spectrum to be generated. The previous section explained that any shape of waveform can be built out of component parts—various individual sine and cosine waveforms of specific frequency, phase, and amplitude. (Fourier analysis produces information about which [sine/cosine] waveforms are present in the series, which frequencies are present, and how strong [amplitude] each of the component parts is.) Figure 9.5 shows a cosine waveform in the upper image, with the associated "power" spectrum below. The power in this case relates to the most prevalent component waveform. The cosine waveform is "pure" in the sense that it consists of entirely one wavelength and is uniform in amplitude. When producing the spectrum for this waveform, there is a single spike in the spectrum that corresponds to the frequency of the waveform. There are no other spikes, and most of the curve shows zero energy, which is to be

expected from a pure waveform.

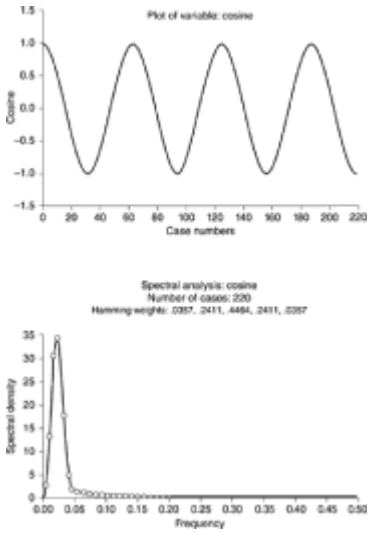

Figure 9.5 A pure cosine waveform of uniform amplitude and fequency (top) and the frequency spectrum for this waveform (bottom).

What happens if there are several frequencies present? Figure 9.6, in the top image, shows a composite waveform created from the six waveforms shown above it. When a spectral analysis is made of the composite waveform, shown in the lower image, there are six spikes. The height of each spike corresponds to the amplitude of each component waveform, and the position along the *x* (horizontal) axis of the graph corresponds to the frequency of the component waveform. The spectrum shows clearly that there are six component frequencies, declining approximately in amplitude from left to right. Inspection of the upper image reveals that the upper six waveforms, the components of the lower composite waveform, increase in frequency from top to bottom, and also decrease in amplitude from top to bottom. So the spectrum accurately reflects the way the analyzed composite waveform was actually constructed. In this example, Figures 9.5 and 9.6 show a spectrum for single and composite waveforms that consist of “clean” components. What does a spectrum look like for a noisy signal?

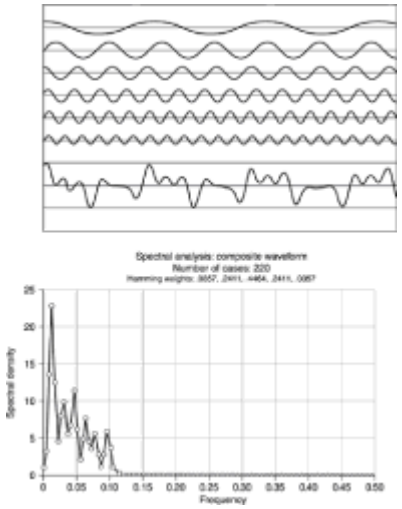

Figure 9.6 Six components of a composite waveform (top) the composite waveform itself is shown as the lowest waveform—and the power spectrum for the composite waveform (bottom).

Figure 9.7 uses the same composite signal built of six components. Considerable noise is added to the waveform. The top-left image graphs the noise. This is a random waveform, varying in value between x1. The bottom-left image shows the power spectrum for the all-noise waveform. The power is distributed fairly evenly along the bottom of the graph. This indicates that there is a fairly equal amount of power present at all wavelengths (frequencies). However, even though this is randomly generated noise, some frequencies, by chance, have more power than others, as shown by the fact that the graphed power spectrum has peaks and valleys. Since this is known to be random noise, the peaks of power are known positively to be spurious in this case. (They are present as shown in the sample of noise generated for the example, but another, identically generated random sample would have minor fluctuations in totally unpredictable places.)

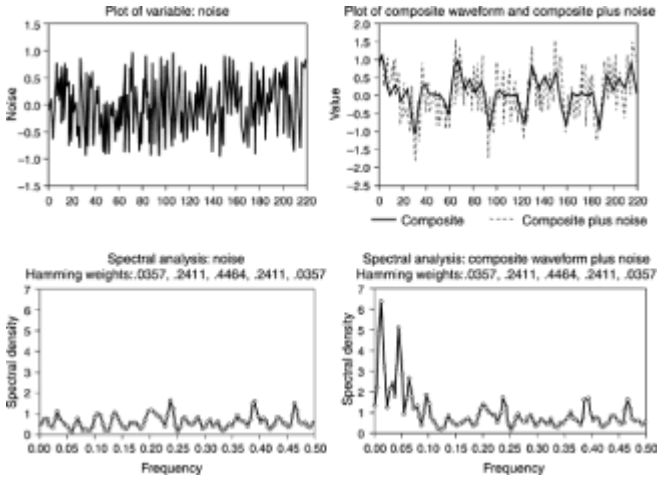

Figure 9.7 Composite signal built of six components showing graph of the random noise (top left), composite waveform with random noise (top right), and two power spectrums (lower left and right).

In any case, the level of power is low relative to the lower-right image of Figure 9.7. The composite waveform, together with the noise added to it, is shown in the upper-right image. The power spectrum in the lower-right image still shows the peaks corresponding to the six component waveforms, but the noise obscures exactly how many there are and precisely where they are located. Adding noise, in other words, "blurred" the original waveform.

So far, all of the waveforms examined have no trend. A *trend* is a noncyclic, monotonically increasing or decreasing component of the waveform. Figure 9.8 shows the composite waveform with an increasing trend in the top image. The bottom image shows the spectrum for such a trended waveform. The power in the trend swamps the detail. The peak at 0 on the *x*-axis is very large compared with the power shown in the other spectra. (Most of the spectral images share a common vertical scale within each figure. The vertical scale of this image has to be much larger than in other images in other figures to show the amplitude of the energy present.) Clearly this causes a problem for the analysis, and dealing with it is discussed shortly.

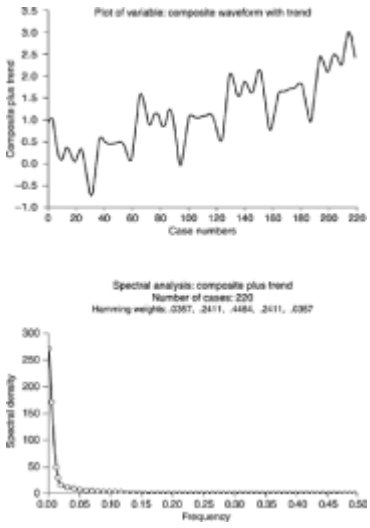

Figure 9.8 Adding a trend to the composite waveform makes it rise overall over time. The rise appears quite modest (top), but the power in the trend component completely swamps any other detail in the power spectrum (bottom).

In these examples, almost all of the waveforms discussed are produced by sine and cosine functions. Distorting noise, not deliberately produced by sine or cosine functions, is added, but the underlying waveforms are regularly cyclical. Figure 9.9 looks at the

spectral analysis of a waveform generated at random. This is the type of shape known as a “random walk.” The random walk shown in the top image starts at 0. From whatever point it is at, it moves from that position either up or down at random, and for a distance of between 0–1 chosen at random, to reach its next position. Each step in distance and direction that the waveform takes, starts wherever the last one ended. (Although removing trend has not been discussed yet, because the random walk shows an apparent trend that causes a spectral analysis problem, the waveform has been detrended before the spectrum was produced. Detrending is discussed later in this chapter.)

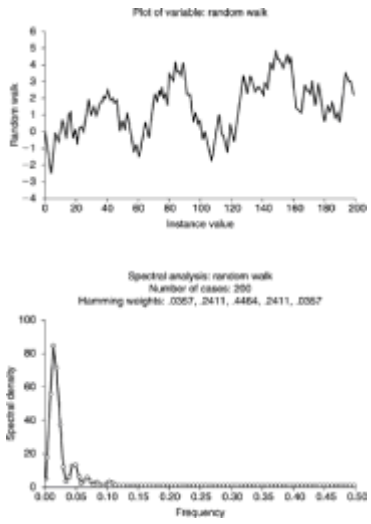

Figure 9.9 A waveform generated by a random selection of the next direction and distance in which to move from every point (top) and a spectral analysis, which shows high energy at some frequencies (bottom).

The spectral analysis in the lower image shows that one frequency predominates in this random walk. Indeed, Fourier analysis describes this waveform as an assemblage of sine and cosine functions. Does this mean that the random walk is predictable? Unfortunately not. Describing an existing waveform, and predicting some future shape for a waveform, are entirely different activities. This waveform is randomly shaped. Random numbers have a distribution. That distribution may be rectangular, normal, or some other shape. It may change over time. It may even be completely unknown. Regardless, it is nonetheless present, and, since a distribution has some structure, even truly random numbers drawn from this distribution carry evidence of the structure of the distribution. This shows up in the spectral analysis. In this case, no prediction at all can be made about the future progress of this particular series. Some inferences about various probabilities can be made, but no valid predictions.

But surely, if inferences can be made, predictions can be made too! As an example of the difference between an *inference* and a *prediction*, observe that each discrete step in the random walk is never more than one unit in each direction. Furthermore, by adding the

assumption that the future will continue to be like the past, we can infer that each step will continue to be less than one unit in each direction. This still does not lead to a prediction of any future value. The best that can be said is that next direction and distance of change will remain less than one unit, positive or negative. We can estimate some probability of any particular value being the next value based on the observed past distribution of step direction and sizes. Even then, unless one knows the causative mechanism that is generating the size and direction of the positive and negative steps, that too can change at any time, invalidating the assumption. Even though there is a structure, the next-step direction, for instance, is no more certain than the outcome of the flip of a fair coin.

How to tell, then, if this is a random walk, or a genuine cyclic pattern? That too is a very fraught question. There are tests for randomness, and some "fingerprints" of randomness are discussed later in this chapter. Even so, it is very easy to "discover" meaningless patterns. Aspects of series data preparation, even as opposed to analysis, depend to some extent on discovering patterns. Meaningless patterns are worse than useless, and as discussed later, they may be positively damaging. Here indeed lurk dragons, chimera, and quicksand!

9.3.5 Describing a Series—Trend, Seasonality, Cycles, Noise

A form of time series analysis known as *classical decomposition* looks at the series as being built from four separate components: trend, seasonality, cycles, and noise. Three of these components turned up during the discussion of spectral analysis. Regarding the components as separate entities helps in dealing with data preparation problems. For instance, in the example used in Figure 9.8, adding a trend component to a cyclical component swamped any other information in the power spectra. In building a description of how power spectra describe waveforms, it was convenient and natural to describe the components in terms of trend, cycles, and noise. Since these components will be manipulated in preparing series data, a slightly closer look at each of them is useful, as well as a look at what "seasonality" might be.

Trend moves in a consistent direction. That is, it is monotonically unidirectional—either never decreasing or never increasing. If never decreasing, it is increasing, and although the rate at which it increases may vary over time, even to 0, it never falls below 0. Similarly, if decreasing, the rate at which it decreases may vary over time, but while it may be flat (fails to actually decrease), it never actually increases. Should the trend line moving in one direction change to the other direction, it is regarded as a different trend, or perhaps as a long period cycle.

Note that what is regarded as a trend over one time period may be a cycle over a different period. For instance, sales, fraud, or some other measure may increase in an upward trend over a year or two. Over 10 years that same trend may be seen to be a part of a larger cyclical pattern such as the business or economic cycle. This piece of the overall pattern, then, would be seen as a trend over 2 years, and a cycle over 10. This distinction

is valid, not arbitrary, as trend components have to be treated differently from cyclical components. So it is true that the difference between finding a trend and finding a cycle may depend entirely on the period that the data covers.

Seasonality reflects the insight that, regardless of any other trend, cycle, or noise influence, certain seasons are inherently different. In time series, seasons are often exactly that—seasons of the year. For example, regardless of economic conditions and other factors, consumers spend more in late December than at other times of the year. Although it appears to be a cyclic effect, it isn't. It is caused by a phenomenon that is local to the season: Christmas. Although Christmas *occurs* cyclically, it is not a cyclic event itself. That is to say, Christmas doesn't wax and wane over the course of the year. What is the level of "Christmasness" in, say, June, July, or August? Is the change in Christmasness part of a cycle during those months? The answer is no, since Christmas is not a cyclical phenomenon but a seasonal one. For instance, to understand how a June sales campaign performed relative to a December campaign, the effect of Christmas occurring in December needs to be removed before attempting to make a fair comparison.

What in a time series are called seasonal effects do occur in other types of displacement series, although they are usually much harder to intuitively understand. Where they do occur, it usually requires a domain expert to identify the seasonalities.

Cycles are fluctuations in the level of the series that have some identifiable repetitive form and structure. Cycles represent the "what goes around, comes around" part of the series. So long as what is going around and coming around can be positively identified, and it reoccurs over some defined period—even if the period itself changes over time—it forms a cycle. Cycles aren't necessarily based on, or thought of as, a collection of sine and cosine waves. That is only the way that Fourier analysis looks at cycles—indeed, at whole waveforms. Very useful, but not necessary. It is only necessary to be able to identify the shape of the repetitive component.

Noise has also been discussed. Chapter 2 looked at some of the sources of noise that in series data can appear in a number of guises. Figure 9.7 shows it as a distortion added to a signal, hiding the underlying structure of the signal. There it is seen as a sort of "fog." Figure 9.9, on the other hand, shows noise as the generating process of a waveform. Seeing through the murk of Figure 9.7 is the problem there. Detecting that it is murk at all, and that it's all murk, is the problem in Figure 9.9!

Noise is the component that is left after the trend, cyclic, and seasonal components have been extracted. It is irregular, for if it were not, it would be characterized as something else. Even so, it has characteristics. Noise comes in different types, colors actually. There is gray noise, pink noise, white noise, and blue noise, to name but a few. The shades are named by analogy with the color spectrum. The color appellation describes, by analogy, the frequency distribution of the noise. Blue light is at the high-frequency end of the color

spectrum. So too, blue noise has a power distribution weighted towards the higher frequencies. White light's energy is evenly distributed at all frequencies—so too with white noise. Just as with blue and white noise, the other noise-shaded frequencies share their relation with the color spectrum.

Noise can be generated from a variety of sources. In the physical world, different processes tend to generate noise with different "signatures"—characteristic frequency distributions. So it is too with noise in behavioral data. Because a noise source may have a characteristic signature, which may be seen by looking at a power spectrum, noise sources can sometimes be identified. If the noise characteristics are known and constant, it can make filtering a waveform out of the noise much easier. (A brief introduction to simple filtering techniques is discussed shortly.)

9.3.6 Describing a Series—Autocorrelation

Correlation measures how values of one variable change as values of another variable change. There are many types of correlation. *Linear correlation* measures the closeness to linear of the between-variable relationship. (Chapter 8 discussed linear relationships in terms of linear regression.) If two variables vary in such a way that the relationship is exactly linear, then, knowing of the linear relationship, and knowing the value of one of the variables, a 100% confident prediction can be made of the value of the other. For instance, the "two times table" has a perfectly correlated, linear relationship between the number to be multiplied and the result. Knowing the value of one variable, "the number to be multiplied," is enough to completely define the value of the other variable, "the result."

Correlation is expressed as a number ranging between +1 and –1. A correlation of x1 indicates perfect predictability. The linear correlation between the two variables in the two times table example is +1. When positive, a linear correlation of ±1 says not only that the two variables are completely linearly related, but also that they move in the same direction. As one becomes more positive, so does the other. A linear correlation of –1 indicates a perfectly predictable relationship, but the values of the variables move in opposite directions—one getting more positive, the other more negative. The "minus two times table" (multiplying any number by –2) would have such a correlation.

Figure 8.5 in the last chapter explained how the equation for a straight line can be interpreted. However, the line on that graph does not join all of the points, which are shown by small circles. The data points cluster about the line, but the fit is certainly not perfect. Knowing one variable's value gives some idea of the value of the other variable, but not an exact idea. Under these circumstances the correlation is less than 1, but since there is some relationship, the correlation is not 0. A correlation of 0 means that knowing the value of one variable tells nothing about the value of the other variable.

As a general guide, until the correlation gets outside the range of between +0.3 to –0.3, any connection is tenuous at best. Not until correlations get to be greater than 0.8, or less

than –0.8, do they begin to indicate a good fit. A difficulty in understanding what correlation says about the “goodness of fit” between two variables is that the relationship between correlation and the goodness of fit itself is not linear! When trying to understand what correlation reveals about the goodness of fit between two variables, perhaps a more useful measure is the amount of “explanatory power” one variable has about the value of another. This explanatory power represents a linear relationship of how well one variable’s value explains another variable’s value. This value, commonly denoted by the symbol r^2 (technically known as the *sample coefficient of determination*), is the square of the correlation. The square of a number between 0 and 1 is *smaller* than the original number. So a correlation of 0.3 represents a r^2 of 0.3 x 0.3 = 0.09, a very small value that indicates little explanatory power indeed. Even a correlation of 0.8 only represents an explanatory power of 0.8 x 0.8 = 0.64. Note that r^2 can never have a negative value since the square of any number is always positive and r^2 ranges in value from 0 to 1. Correlation values are most commonly used and quoted, but it is well to keep in mind that the strength of the connection is perhaps more intuitively represented by the square of the correlation.

Having looked at correlation, autocorrelation follows naturally. *Autocorrelation* literally means correlation with self. When used as a series description, it is a measure of how well one part of the series correlates with another part of the same series some fixed number of steps away. The distance between index points on the series is called the *lag*. An autocorrelation with a lag of one measures the correlation between every point and its immediate neighbor. A lag of two measures the correlation between every point and its neighbor two distant. Figure 9.10 shows how the data points can be placed into the columns of a matrix so that each column has a different lag from the first column. Using this matrix, it is easy to find the linear correlation between each column, each corresponding to a different lag.

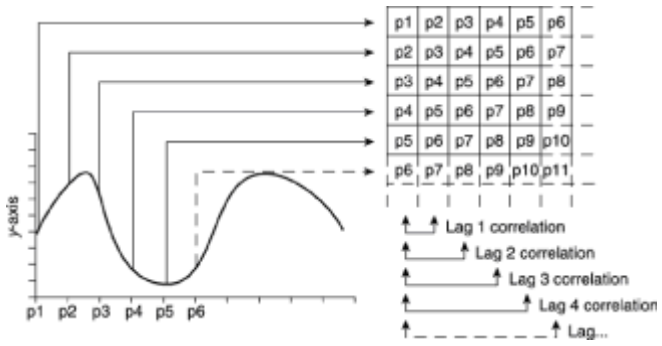

Figure 9.10 Building a matrix to find linear correlation for many lags. Every additional column is one point further lagged from the first column. Finding the correlation between every column gives the data needed to build a correlogram.

The result of building a multiple lag autocorrelation is a correlogram. A *correlogram* measures, and shows in graphical form, the correlation for each of many different lags. This is done by plotting the linear correlation for a lag of one, then for a lag of two, then

three, and so on. Figure 9.11 shows a series of the waveforms used so far, together with their associated correlograms. Comparison between the correlograms and the spectra for the same waveforms shows that different features about the waveforms are emphasized by each descriptive technique.

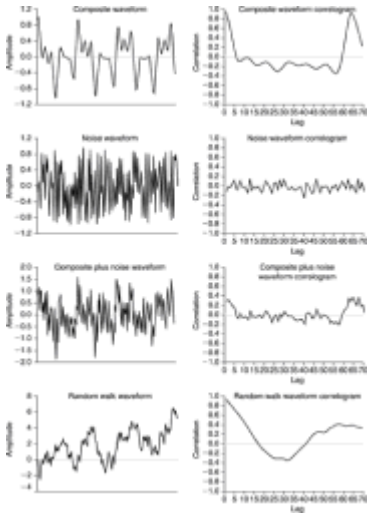

Figure 9.11 Waterforms and their correlograms.

9.4 Modeling Series Data

Given these tools for describing series data, how do they help with preparing the data for modeling? There are two main approaches to modeling series data. One uses extensions to the descriptive tools just discussed, and attempts to understand and explicate the within-variable and between-variable interactions in terms of the series itself. The other approach decomposes the data, using something like a reverse pivot (described in Chapter 4), and uses nonseries tools to further analyze the data.

Both of these approaches try to understand the interactions of the data over time. The first approach tries to understand the data in time, whereas the second tries to understand the effect of time. For instance:

- Approach 1 shows how sales/fraud/claims vary over time.

- Approach 2 shows what the effect of the trend/season/cycle/noise of sales/fraud/claims has on inferences/predictions about backlog/losses/payments.

Deciding which approach, or deciding to use both approaches, does have an impact on data preparation. However, it mainly affects data set assembly, rather than other aspects of preparing series data. In general, the same problems need to be addressed regardless of which approach is used. For instance, and this is discussed more fully later, a trend is a monotonic component of a series. Monotonicity is a problem that needs to be addressed

whether or not it is a series that is being modeled, and regardless of which type of series model is being built.

9.5 Repairing Series Data Problems

Series data shares most of the problems of nonseries data, plus some additional ones of its own. Where the problems are similar, some of the preparation techniques are similar too. Some are not. Even those that are similar may need to be implemented differently. The important point for the miner to remember about series data is maintaining the information contained in the ordering.

9.5.1 Missing Values

While a poorly constructed series can contain empty values, it is very unlikely. Missing values cause a major problem in series data, and it is very unlikely that any series would be constructed to permit empty values. (Chapter 2 discusses the difference between missing and empty values.) In any case, whether missing or empty, series modeling techniques fare even worse with values that are absent than nonseries techniques.

There are two dimensions of a series in which a value could be missing: the feature variable, and the index variable. Index variable problems are addressed next. Here attention will be confined to a value missing in the feature variable.

When replacing a missing value in nonseries data, joint variability is preserved between variables, and a suitable value for the replacement is found using the information that is contained in whatever variable values are present. If the series is a multivariable series, that technique works here too. Instead of multiple regression, a multiple autoregression is used to find a replacement value. The concept represents a straightforward extension of multiple regression, described in Chapter 8, combined with autocorrelation described earlier in this chapter.

Autoregression measures the self-similarity of the waveform across different lags. Just as with single linear or multiple regression, so too can autoregressions be determined using the ratio method (Chapter 8). Using multiple autoregression techniques for replacing missing values can provide a robust estimate for missing values. However, chimerical dragons rear their heads when doing this!

Often, time series, both in physical and in behavioral data sets, tend to have contiguous missing values—that is, runs of values all missing. This can easily happen if the collection mechanism either fails or is intermittent in operation. Then there may occur runs of data with periods of no data in between. Filling in these holes with self-similar patterns from other parts of the series reinforces the apparent self-similarity. The estimated missing values will be smoothed (smoothing is mentioned later) by the replacement process itself since they represent some sort of aggregated estimate of the missing value. Unfortunately

for missing-value replacement, smoothing enhances any regular patterns in the data that are being used to make the estimate. So replacing missing values necessarily enhances a pattern that is discovered elsewhere in the series and used to replace the missing runs of data.

This means that not only is a missing run replaced with an aggregate pattern borrowed from elsewhere, it is an aggregate pattern *enhanced* by smoothing! When the prepared data is later modeled, this enhanced pattern may be, almost certainly will be, "discovered." Depending on the length of the replaced run, it may predominate in a spectral analysis or correlogram, for instance.

One way to ameliorate the problem to some extent is to add noise to the replacement values. A glance at the correlograms in Figure 9.11 shows that the noise added to the composite waveform considerably masks the strength of the correlations at various lags. But how much noise to add? One estimate for additional noise to add is to use the same level that is estimated to exist in the values that are present. And what is that level of noise? That question, although examined later, does not always have an easy answer.

This is a very tough chimerical dragon to slay! The miner needs to look carefully at which patterns are being reinforced by the replacement of missing values, and be very circumspect in deciding later that these are at all meaningful when they are later "discovered" during modeling. It is, unfortunately, quite possible that the training, test, and evaluation data sets all suffer from the same replacement problem, and thus, seem to confirm the "discovery" of the pattern. This chimera can be very persuasive; caution is the watchword!

9.5.2 Outliers

Chapter 2 first mentioned outliers—a few values that lie far from the bulk of the range. Outliers occur in time series too. They come as individual occurrences and, sometimes, in "runs"—clusters of consecutive values of the same order of magnitude as each other, but as a group lying well outside the range of the other values. The miner will need to ask hard questions about why the outliers exist. Are they really significant? What sort of process could account for them? Can they be translated back into the normal range if they are indeed errors?

If no rationale can account for the outliers, with all of the same caveats mentioned for missing values, as a last resort replace the outliers, exactly as for missing values.

9.5.3 Nonuniform Displacement

Usually, although not invariably, displacement steps are spaced uniformly across the indexing dimension—measurements, in other words, taken in regular and uniform increments. Many of the analytical techniques assume this uniformity and won't work well,

or at all, if the displacement is not constant. Since almost all techniques, including spectral analysis and correlogramming, assume uniform displacement between indexes, the values must be adjusted to reflect what they would have been had they been taken with uniform displacement.

Figure 9.12 shows a sine wave sampled at nonuniform displacements. Graphing it as if the displacements were uniform clearly shows jagged distortion. The less uniform the displacement intervals, the worse the situation becomes. Since the jagged waveform appears affected by noise, techniques for removing the noise work well to estimate the original waveform as it would have been if sampled at uniform displacements.

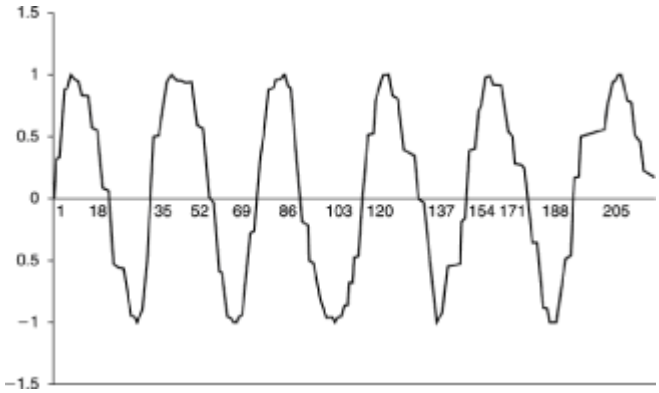

Figure 9.12 A sine wave that is sampled with a nonconstant displacement. When reproduced as if the intervals were constant, the waveform becomes distorted.

9.5.4 Trend

Trend is the monotonically increasing or decreasing component of a waveform. Leaving a trend present in a waveform causes problems for almost all modeling methods. If the trend is nonlinear, then removing the linear component leaves only the nonlinear component of the trend. The nonlinear component then appears cyclic. Figure 9.13 shows how this can be done. In the left image, performing a linear regression linearly approximates the nonlinear trend. Subtracting the linear component from the trend is shown in the right image. This leaves the cyclic part of the trend, but there is no trend remaining—or at least, stated differently, what there is of the trend now appears to be cyclic.

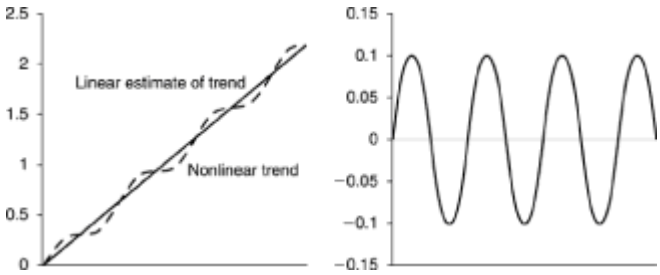

Figure 9.13 A nonlinear trend and a linear estimates found by linear regression (left). Subtracting the linear part of the trend from the nonlinear part turns the nonlinear part of the trend, in the case, into a cyclic representation (right).

Detrending a series is obviously, at least on occasion, an absolute necessity. (Figure 9.8 showed how trend can swamp the spectrum.) None of the modeling techniques discussed can deal with the problem, whether it is called monotonicity or trend. But detrending has dangers. Figure 9.14 shows a cosine waveform. This is a perfectly cyclic waveform that oscillates uniformly about the 0 point. A cosine waveform actually has a completely flat trend component since, if extended far enough, it has a completely symmetrical distribution about the 0 point. Here, as much of the waveform as is available has been linearly detrended. The “discovered” linear trend shown apparently has a downward slope. Subtracting this slope from the “trended” waveform distorts the waveform. Using such a distorted waveform in modeling the data leads to many problems, and bad—frequently useless—models. Detrending nontrended data can do enormous damage. But can the miner avoid it?

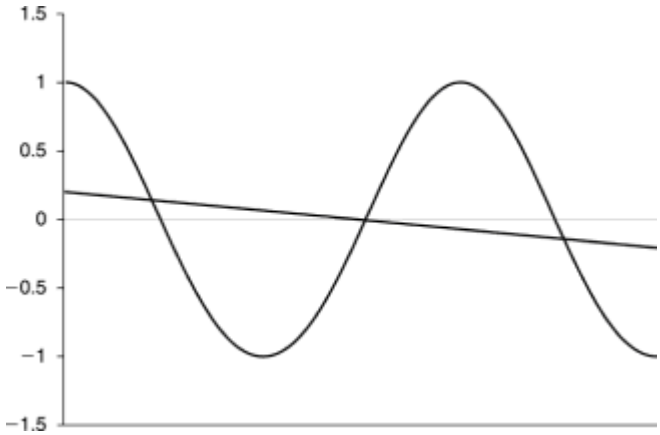

Figure 9.14 A cosine waveform and straight line indicating a trend even though the waveform actually has no trend. This is caused by using less than a complete waveform.

The problem in Figure 9.14 is that only partial cycles were used. In the example, which

was constructed specifically to show the problem, this is obvious. In real-world data it is often very much harder, impossible even, to determine if apparent trend is an artifact of the data or real. There is no substitute for looking at the data in the form of data plots, correlograms, spectra, and so on. Also, as always, the miner should try to determine if there is a rationale for any discovery. Always the miner should ask, “In this case, is there some reason to expect the existence of the discovered trend?” More than that, if there is a positive known reason that precludes the existence of trend, data should not be detrended—even if it appears to have a trend. Look back at the random walk, say, Figure 9.11. This waveform was entirely generated by additive noise processes—accumulated errors. Suppose that this additive error had contaminated the series that otherwise contained no trend. Detrending this type of waveform can make the cyclic information impossible for a modeling tool to discover. Over the long run, such additive noise as is shown averages to 0—that is, over the long haul it is trendless. Including trend as part of the model not only hides the cyclic information, but also adds a nonexistent trend to the predictions. On the other hand, it may be almost impossible to work with a waveform that is not detrended.

Deep quicksand here! The only real answer is to experiment! Survey the data extensively, trended and detrended. If access to surveying software is difficult, at least build multiple models and work extensively with the data, both trended and detrended.

9.6 Tools

Removing trend involves identifying and removing a component of the overall waveform. Doing this makes the remaining waveform more convenient or tractable to handle, or better reveals information of concern for modeling. But removing trend is really just a special case in a set of manipulations for exposing series information. These manipulations—filtering, moving averages, and smoothing—are the miner’s basic series manipulation tools.

9.6.1 Filtering

A *filter* is a device that selectively holds some things back and lets other pass. In the case of series data, the filtering is performed on different components of the overall waveform. Since a complex waveform can be thought of as being constructed from simpler waveforms, each of a separate single frequency, the components can be thought of as those simpler waveforms. A wide array of filters can be constructed. High-pass filters, for example, let only high frequencies through, “holding back” the lower frequencies. Actually the “holding back” is known as attenuation. *Attenuation* means “to make less,” and the lower-frequency amplitude is actually reduced, rather than held back, leaving the higher frequencies more visible. If selected components at different frequencies are attenuated, their amplitude is reduced. Changing the amplitude of component cycles changes the shape of the waveform, as shown in Figure 9.3. By using filters, various parts of the frequency spectrum can be removed from the overall waveform and investigated

separately from the effects of the remaining frequencies.

While it is possible to construct complex mathematical structures to perform the necessary filtering, the purpose behind filtering is easy to understand and to see.

Figure 9.8 showed the spectrum of a trended waveform. Almost all of the power in this spectrum occurs at the lowest frequency, which is 0. With a frequency of 0, the corresponding waveform to that frequency doesn't change. And indeed, that is a linear trend—an unvarying increase or decrease over time. At each uniform displacement, the trend changes by a uniform amount. Removing trend corresponds to low-frequency filtering at the lowest possible frequency—0. If the trend is retained, it is called *low-pass filtering* as the trend (the low-frequency component) is "passed through" the filter. If the trend is removed, it would be called *high-pass filtering* since all frequencies but the lowest are "passed through" the filter.

In addition to the zero frequency component, there are an infinite number of possible low-frequency components that are usefully identified and removed from series data. These components consist of fractional frequencies. Whereas a zero frequency represents a completely unvarying component, a fractional frequency simply represents a fraction of the whole cycle. If the first quarter of a sine wave is present in a composite waveform, for example, that component would rise from 0 to 1, and look like a nonlinear trend.

Some of the more common fractional frequency components include exponential growth curves, logistic function curves, logarithmic curves, and power-law growth curves, as well as the linear trend already discussed. Figure 9.15 illustrates several common trend lines. Where these can be identified, and a suitable underlying generating mechanism proposed, that mechanism can be used to remove the trend. For instance, taking the logarithm of all of the series values for modeling is a common practice for some series data sets. Doing this removes the logarithmic effect of a trend. Where an underlying generating mechanism cannot be suggested, some other technique is needed.

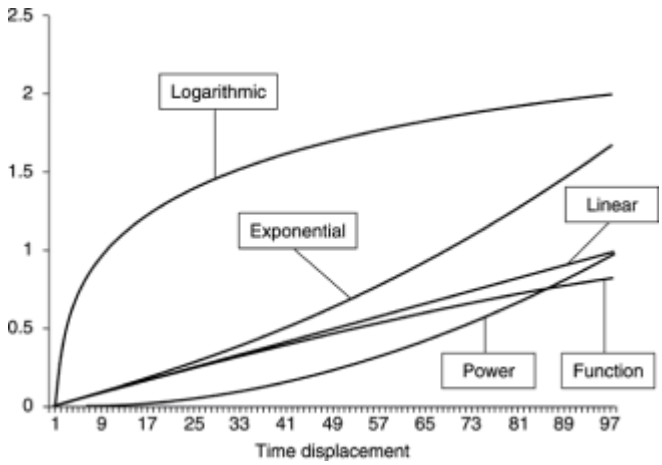

Figure 9.15 Several low-frequency components commonly discovered in a series data that can be beneficially identified and removed.

9.6.2 Moving Averages

Moving averages are used for general-purpose filtering, for both high and low frequencies. Moving averages come in an enormous range and variety. To examine the most straightforward case of a simple moving average, pick some number of samples of the series, say, five. Starting at the fifth position, and moving from there onward through the series, use the average of that position plus the previous four positions instead of the actual value. This simple averaging reduces the variance of the waveform. The longer the period of the average, the more the variance is reduced. With more values in the weighting period, the less effect any single value has on the resulting average.

TABLE 9.1 Log-five SMA

Position	Series value	SMA5	SMA5 range
1	0.1338		
2	0.4622		
3	0.1448	0.2940	1-5
4	0.6538	0.3168	2-6
5	0.0752	0.3067	3-7
6	0.2482	0.3497	4-8
7	0.4114	0.3751	5-9
8	0.3598	0.4673	6-10
9	0.7809		
10	0.5362		

9.1 shows a lag-five *simple moving average (SMA)*. The values are shown in the column "Series value," with the value of the average in the column "SMA5." Each moving average value is the average of the two series values above it, the one series value opposite and the two next series values, making five series values in all. The column "SMA5 range" shows which positions are included in any particular moving average value.

One drawback with SMAs, especially for long period weightings, is that the average cannot begin to be calculated until the number of periods in the weighting has passed. Also, the average value refers to the data point that is at the center of the weighting period. (Table 9.1 plots the average of positions 1–5 in position 3.) With a weighting period of, say, five days, the average can only be known as of two days ago. To know the moving average value for today, two days have to pass.

Another potential drawback is that the contribution of each data point is equal to that of all the other data points in the weighting period. It may be that the more distant past data values are less relevant than more recent ones. This leads to the creation of a *weighted moving average (WMA)*. In such a construction, the data values are weighted so that the more recent ones contribute more to the average value than earlier ones. Weights are chosen for each point in the weighting period such that they sum to 1.

Table 9.2 shows the weights for constructing the lag-five WMA that is shown in Table 9.3. The "v_{-4} indicates that the series value four steps back is used, and the weight "0.066" indicates that the value with that lag is multiplied by the number 0.066, which is the weight. The lag-five WMA's value is calculated by multiplying the last five series values by the appropriate weights.

TABLE 9.2 Weight for calculating a lag-five WMA.

Log	Weight
V_{-4}	0.576766
V_{-3}	0.423234
V_{-2}	0.576766
V_{-1}	0.423234
V_{0}	0.576766

Wt total	1.000

TABLE 9.3 Log-five WMA

Position	Series value	WMA5
1	0.1338	
2	0.4622	
3	0.1448	
4	0.6538	0.2966
5	0.0752	0.2833
6	0.2482	0.3161
7	0.4114	0.3331
8	0.3598	0.4796
9	0.7809	0.5303
10	0.5362	

Table 9.3 shows the actual average values. Because of the weights, it is difficult to "center" a WMA. Here it is shown "centered" one advanced on the lag-five SMA. This is done because the weights favor the most recent values over the past values—so it should be plotted to reflect that weighting.

Exponential moving averages (EMAs) solve the delay problem. Such averages consist of two parts, a "head" and a "tail." The tail value is the previous average value. The head

value is the current data value. The average's value is found by moving the tail some way closer to the head, but not all of the way. A weight is applied to decide how far to move the tail toward the head. With light tail weights, the tail follows the head quite closely, and the average behaves much like a short weighting period simple moving average. With heavier tail weights, the tail moves more slowly, and it behaves somewhat like a longer-period SMA. The head weight and the tail weight taken together must always sum to a value of 1.

No two averages behave in exactly the same way, but for EMAs, obviously the heavier the head weight, the "faster" the EMA value will move—that is to say, the more closely it follows the value of the series. For comparison, the EMA weights shown in Table 9.4 approximate the lag-five SMA.

TABLE 9.4 Head and tail weights to approximate a lag-five SMA.

Head weight	0.576766
Head weight	0.423234

Table 9.5 shows the actual values for the EMA. In this table, position 1 of the EMA is set to the starting value of the series. The formula for determining the present value of the EMA is

$$v_{EMA0} = (v_{s0} \times w_h) + (v_{EMA-1} \times w_t)$$

where

v_{EMA0}	is the value of the current EMA
v_{s0}	is the current series value
w_h	is the head weight
v_{EMA-1}	is the last value of the EMA
w_t	is the tail weight

TABLE 9.5 Values of the EMA

Position	Series value	EMA	Head	Tail
1	0.1338	0.1338		
2	0.4622	0.3232	0.2666	0.0566
3	0.1448	0.2203	0.0835	0.1956
4	0.6538	0.4703	0.3771	0.0613
5	0.0752	0.2424	0.0434	0.2767
6	0.2482	0.2458	0.1432	0.0318
7	0.4114	0.3413	0.2373	0.1051
8	0.3598	0.3519	0.2075	0.1741
9	0.7809	0.5993	0.4504	0.1523
10	0.5362	0.5629	0.3092	0.3305

This formula, with these weights, specifies that the current average value is found by multiplying the current series value by 0.576766, and the last value of the average by 0.423243. The results are added together. The table shows the value of the series, the current EMA, and the head and the tail values.

Figure 9.16 illustrates the moving averages discussed so far, and the effects of changing the way they are constructed. The series itself changes value quite abruptly, and all of the averages change more slowly. The SMA is the slowest to change of the averages shown. The WMA moves similarly to the SMA, but clearly responds more to the recent values, exactly as it is constructed to do.

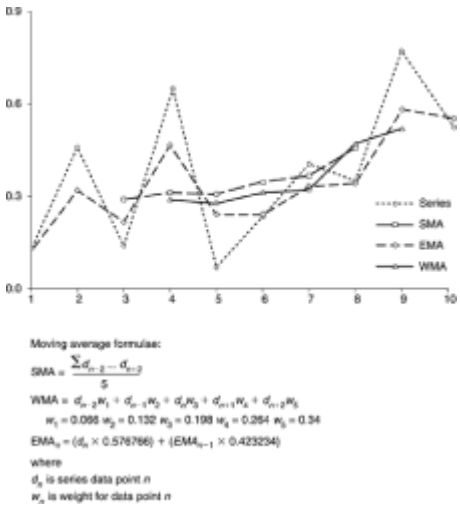

Figure 9.16 Various moving averages and the effects of changing weights showing SMAs, WMAs (weights shown separately), and EMAs (weights included in formula). The graph illustrates the data shown in Tables 9.1, 9.2, and 9.5.

The EMA is the most responsive to the actual series value of the three averages shown. Yet the weights were chosen to make it approximate the lag-five SMA average. Since they seem to behave so differently, in what sense are these two approximately the same? Over a longer series, with this set of weights, the EMA tends to be centered about the value of the lag-five SMA. A series length of 10, as in the examples, is not sufficient to show the effect clearly.

In general, as the lag periods get longer for SMAs and WMAs, or the head weights get lighter (so the tail weights get heavier) for the EMAs, the average reacts more slowly to changes in the series. Slow changes correspond to longer wavelengths, and longer wavelengths are the same as lower frequencies. It is this ability to effectively change the frequency at which the moving average reacts that makes them so useful as filters.

Although specific moving averages are constructed for specific purposes, for the examples that follow later in the chapter, an EMA is the most convenient. The convenience here is that given a data value (head), the immediate EMA past value (tail), and the head and tail weights, then the EMA needs no delay before its value is known. It is also quick and easy to calculate.

Moving averages can be used to separate series data into two frequency domains—above and below the threshold set by the reactive frequency of the moving average. How does this work in practice?

Moving Averages as Filters—Removing Noise

The composite-plus-noise waveform, first shown in Figure 9.7, seems to have a slower

cycle buried in higher-frequency noise. That is, buried in the rapid fluctuations, there appears to be some slower fluctuation. Since this is a waveform built especially for the example, this is in fact the case. However, nonmanufactured signals often show this type of noise pattern too. Discovery of the underlying signal starts by trying to remove some of the noise. Using an EMA, the high frequencies can be separated from the lower frequencies.

High frequencies imply an EMA that moves fast. The speed of reaction of an EMA is set by adjusting its weights. In this case, the head weight is set at 0.44 so that it moves very fast. However, because of the tail weight, it cannot follow the fastest changes in the waveform—and the fastest changes are the highest frequencies. The path of the EMA itself represents the waveform without the higher frequencies. To separate out just the high frequencies, subtract the EMA from the original waveform. The difference is the high-frequency component missing from the EMA trace. Figure 9.17 shows the original waveform, waveform plus noise, EMA, and high frequencies remaining after subtraction. Using an EMA with a head weight of 0.44 better resembles the original signal than the noisy version because it has filtered out the high frequencies. Subtracting the EMA from the noisy signal leaves the high frequencies removed by the EMA (top).

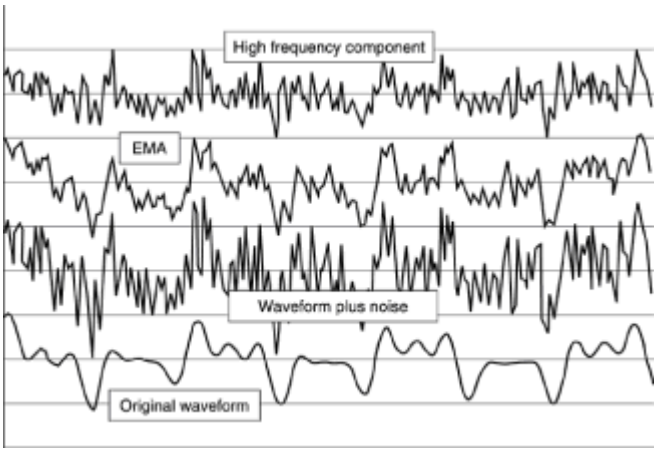

Figure 9.17 The original waveform, waveform plus noise, EMA, and high frequencies remaining after subtraction.

It turns out that with this amount of weighting, the EMA is approximately equivalent to a three-sample SMA (SMA3). An SMA3 has its value centered over position two, the middle position. Doing this for the EMA used in the example recovers the original composite waveform with a correlation of about 0.8127, as compared to the correlation for the signal plus noise of about 0.6.

9.6.3 Smoothing 1—PVM Smoothing

There are many other methods for removing noise from an underlying waveform that do

not use moving averages as such. One of these is *peak-valley-mean (PVM)* smoothing. Using PVM, a *peak* is defined as a value higher than the previous and next values. A *valley* is defined as a value lower than the previous and next values. PVM smoothing uses the mean of the last peak and valley (i.e., (*P* + *V*)/2) as the estimate of the underlying waveform, instead of a moving average. The PVM retains the value of the last peak as the current peak value until a new peak is discovered, and the same is true for the valleys. This is the shortest possible PVM and covers three data points, so it is a lag-three PVM. It should be noted that PVMs with other, larger lags are possible.

Figure 9.18 shows in the upper image the peak, valley, and mean values. The lower image superimposes the recovered waveform on the original complex waveform without any noise added. Once again, as with moving averages, the recovered waveform needs to be centered appropriately. Centering again is at position two of three, halfway along the lag distance, as from there it is always the last and next positions that are being evaluated. The recovery is quite good, a correlation a little better than 0.8145, very similar to the EMA method.

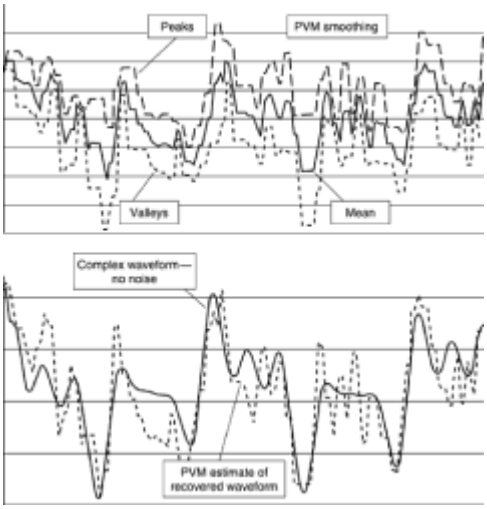

Figure 9.18 PVM smoothing: the peak, valley, and mean values for the composite-plus-noise waveform (top) and the mean estimate superimposed on the actual composite waveform (bottom).

9.6.4 Smoothing 2—Median Smoothing, Resmoothing, and Hanning

Median smoothing uses “windows.” A *window* is a group of some specific number of contiguous data points. It corresponds to the lag distance mentioned before. The only difference between a window and a lag is that the data in a window is manipulated in some way, say, changed in order. A lag implies that the data is not manipulated. As the window moves through the series, the oldest data point is discarded, and a new one is

added. When *median smoothing*, use the median of the values in the window in place of the actual value. A *median* is the value that comes in the middle of a list of values ordered by value. When the window is an even length, use as the median value the average of the two middle values in the list. In many ways, median smoothing is similar to average smoothing except that the median is used instead of the average. Using the median makes the smoothed value less sensitive to extremes in the window since it is always the middle value of the ordered values that is taken. A single extreme value will never appear in the middle of the ordered list, and thus does not affect the median value.

Resmoothing is a technique of smoothing the smoothed values. One form of resmoothing continues until there is no change in the resmoothed waveform. Other resmoothing techniques use a fixed number of resmooths, but vary the window size from smoothing to smoothing.

Hanning is a technique borrowed from computer vision, where it is used for image smoothing. Essentially it is a form of weighted averaging. The window is three long, left in the original order, so it is really a lag. The three data points are multiplied by the weights 0.25, 0.50, 0.25, respectively. The hanning operation removes any final spikes left after smoothing or resmoothing.

There are very many types of resmoothing. A couple of examples of the technique will be briefly examined. The first, called “3R2H,” is a median smooth with a window of three, repeated (the “R” in the name) until no change in the waveform occurs; then a median smoothing with a window length of two; then one hanning operation. When applied to the example waveform, this smoothing has a correlation with the original waveform of about 0.8082.

Another, called “4253H” smoothing, has four median smoothing operations with windows of four, two, five, and three, respectively, followed by a hanning operation. This has a correlation with the original example waveform of about 0.8030. Although not illustrated, both of these smooths produce a waveform that appears to be very similar to that shown in the lower image of Figure 9.18.

Again, although not illustrated, these techniques can be combined in almost any number of ways. Smoothing the PVM waveform and performing the hanning operation, for example, improves the fit with the original slightly to a correlation of about 0.8602.

9.6.5 Extraction

All of these methods remove noise or high-frequency components. Sometimes the high-frequency components are not actually noise, but an integral part of the measurement. If the miner is interested in the slower interactions, the high-frequency component only serves to mask the slower interactions. Extracting the slower interactions can be done in several ways, including moving averages and smoothing. The various

smoothing and filtering operations can be combined in numerous ways, just as smoothing and hanning the PVM smooth shows. Many other filtering methods are also available, some based on very sophisticated mathematics. All are intended to separate information in the waveform into its component parts.

What is extracted by the techniques described here comes in two parts, high and lower frequencies. The first part is the filtered or smoothed part. The remainder forms the second part and is found by subtracting the first part, the filtered waveform, from the original waveform. When further extraction is made on either, or both, of the extracted waveforms, this is called *reextraction*. There seems to be an endless array of smoothing and resmoothing, extraction and reextraction possibilities!

Waveforms can be separated in high-, middle-, and low-frequency components—and then the separated components further separated. Here is where the miner must use judgment. Examination of the extracted waveforms is called for—indeed, it is essential. The object of all filtering and smoothing is to separate waveforms with pattern from noise. The time to stop is when the extraction provides no additional separation. But how does the miner know when to stop?

This is where the spectra and correlograms are very useful. The noise spectrum (Figure 9.7) and correlogram (Figure 9.11) show that noise, at least of the sort shown here, has a fairly uniform spectrum and uniformly low autocorrelation at all lags. There still might be useful information contained in the waveform, but the chance is small. This is a good sign that extra effort will probably be better placed elsewhere. But what of the random walk? Here there is a strong correlation in the correlogram, and the spectrum shows clear peaks. Is there any way to determine that this is random walking?

9.6.6 Differencing

Differencing a waveform provides another powerful way to look at the information it contains. The method takes the difference between each value and some previous value, and analyzes the differences. A lag value determines exactly which previous value is used, the lag having the same meaning as mentioned previously. A lag of one, for instance, takes the difference between a value and the immediately preceding value.

The actual differences tend to appear noisy, and it is often very hard to see any pattern when the difference values are plotted. Figure 9.19 shows the lag-one difference plot for the composite-plus-noise waveform (left). It is hard to see what, if anything, this plot indicates about the regularity and predictability of the waveform! Figure 9.19 also shows the lag-one difference plot for the complex waveform without noise added (right). Here it is easy to see that the differences are regular, but that was easy to see from the waveform itself too—little is learned from the regularity shown.

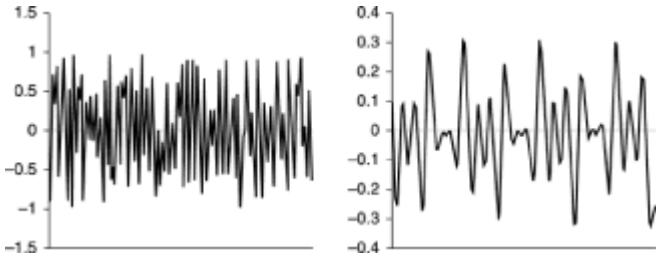

Figure 9.19 Log-one difference plots: composite-plus-noise waveform differences (left) and pattern of differences for the composite waveform without noise (right).

Forward Differencing

Looking at the spectra and correlograms of the lag-one difference plots, however, does reveal information. When first seen, the spectra and correlograms shown in Figure 9.20 look somewhat surprising. It is worth looking back to compare them with the nondifferenced spectra for the same waveforms in Figures 9.6, 9.7, and 9.9, and the nondifferenced correlograms in Figure 9.11.

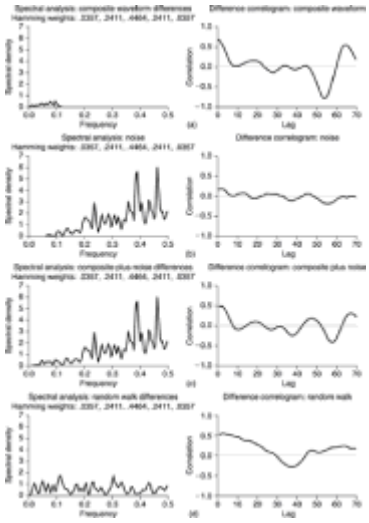

Figure 9.20 Differences spectra and correlograms for various waveforms.

Figure 9.20(a) shows that the differenced composite waveform contains little spectral energy at any of the frequencies shown. What energy exists is in the lower frequencies as before. The correlogram for the same waveform still shows a high correlation, as expected.

In Figure 9.20(b), the noise waveform, the differencing makes a remarkable difference to the power spectrum. High energy at high frequencies—but the correlogram shows little

correlation at any lag.

Although the differenced noise spectrum in Figure 9.20(b) is remarkably changed, it is nothing like the spectrum for the differenced random walk in 9.20(d). Yet both of these waveforms were created from random noise. What is actually going on here?

Randomness Detector?

What is happening that makes the random waveforms produce such different spectra? The noise power spectrum (shown in Figure 9.7) is fairly flat. Differencing it, as shown in Figure 9.20(b), amplified—made larger—the higher frequencies. In fact, the higher the frequency, the more the amplification. At the same time, differencing attenuated—made smaller—the lower frequencies. So differencing serves as a high-pass filter.

What of the random walk? The random walk was actually constructed by taking random noise, in the form of numbers in the range of –1 to +1, and adding them together step by step. When this was differenced, back came the original random noise used to generate it. In other words, creating a walk, or "undifferencing," serves to amplify the low frequencies and attenuate the high frequencies—exactly the opposite of differencing! Building the random walk obviously did something that hid the underlying nature of the random noise used to construct it. When differenced, the building process was undone, and back came a spectrum characteristic of noise. So, to go back to the question, "Is there a way to tell that the random walk is generated by a random process?" the answer is a definite "maybe." Differencing can at least give some clues that the waveform was generated by some process that, at least by this test, looks random.

There is no way to tell from the series itself if the random walk is in fact random. That requires knowing the underlying process in the real world that is actually responsible for producing the series. The numbers used here, for instance, were not actually random, but what is known as pseudo-random. (Genuinely random numbers turn out to be fiendishly difficult to come by!) A computer algorithm was used that has an internal mechanism that produces a string of numbers that pass certain tests for randomness. However, the sequence is actually precisely defined, and not random at all. Nonetheless, it looks random, and lacking an underlying explanation, which may or may not be predictive, it is at least known to have some of the properties of a random number. Simply finding a spectrum indicating possible randomness only serves as a flag that more tests are needed. If it eventually passes enough tests, this indeed serves as a practical definition of randomness. What constitutes "enough" tests depends on the miner and the needs of the application. But nonetheless, the working definition of randomness for a series is simply one that passes all the tests of randomness and has no underlying explanation that shows it to be otherwise.

Reverse Differencing (Summing)

Interestingly, discovering a way to potentially expose random characteristics used the reverse process of differencing. Building the random walk required adding together random distance and direction steps generated by random noise. It turns out that creating any series in a similar way is the equivalent of reverse differencing! (This, of course, is summing—the exact opposite of taking a difference. “Reverse differencing” seems more descriptive.) Without going into details, the power spectrum and correlogram for the reverse-differenced composite-plus-noise waveform is shown in Figure 9.21. The power spectrum shows the low-frequency amplification, high-frequency attenuation that is the opposite effect of forward differencing. The correlogram is interesting as the correlation curve is much stronger altogether when the high-frequency components are attenuated. In this case, the reverse-differenced curve becomes very highly autocorrelated—in other words, highly predictable.

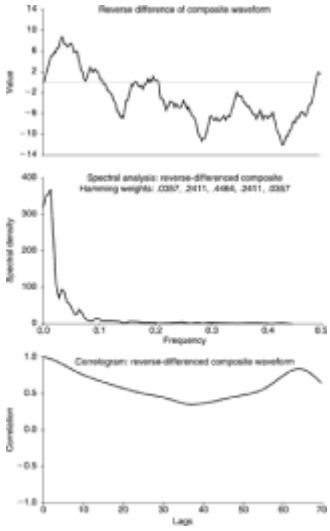

Figure 9.21 Effects of reverse differencing. Low frequencies are enhanced, and high frequencies are attenuated.

Just as differencing can yield insights, so too can summing. Linearly detrending the waveform before the summing operation may help too.

9.7 Other Problems

So far, the problems examined have been specific to series data. The solutions have focused on ways of extracting information from noisy or distorted series data. They have involved extracting a variety of waveforms from the original waveform that emphasize particular aspects of the data useful for modeling. But whatever has been pulled out, or extracted, from the original series, it is still in the form of another series. It is quite possible to look at the distribution of values in such a series exactly as if it were not a series. That is to say, taking care not to actually lose the indexing, the variable can be treated exactly as if it were a nonseries variable. Looking at the series this way allows some of the tools

used for nonseries data to be applied to series data. Can this be done, and where does it help?

9.7.1 Numerating Alpha Values

As mentioned in the introduction to this chapter, numeration of alpha values in a series presents some difficulties. It can be done, but alpha series values are almost never found in practice. On the rare occasions when they do occur, numerating them using the nonseries techniques already discussed, while not providing an optimal numeration, does far better than numeration without any rationale. Random or arbitrary assignment of values to alpha labels is always damaging, and is just as damaging when the data is a series. It is not optimal because the ordering information is not fully used in the numeration. However, using such information involves projecting the alpha values in a nonlinear phase space that is difficult to discover and computationally intense to manipulate. Establishing the nonlinear modes presents problems because they too have to be constructed from the components cycle, season, trend, and noise. Accurately determining those components is not straightforward, as we have seen in this chapter. This enormously compounds the problem of in-series numeration.

The good news is that, with time series in particular, it seems easier to find an appropriate rationale for numerating alpha values from a domain expert than for nonseries data. Reverse pivoting the alphas into a table format, and numerating them there, is a good approach. However, the caveat has to be noted that since alpha numerated series occur so rarely, there is little experience to draw on when preparing them for mining. This makes it difficult to draw any hard and fast general conclusions.

9.7.2 Distribution

As far as distributions are concerned, a series variable has a distribution that exists without reference to the ordering. When looked at in this way, so long as the ordering—that is, the index variable—is not disturbed, the displacement variable can be redistributed in exactly the same manner as a nonseries variable. Chapter 7 discussed the nature of distributions, and reasons and methods for redistributing values. The rationale and methods of redistribution are similar for series data and may be even more applicable in some ways. There are time series methods that require the variables' data to be centered (equally distributed above and below the mean) and normalized. For series data, the distribution should be normalized after removing any trend.

When modeling series data, the series should, if possible, be what is known as stationary. A *stationary* series has no trend and constant variance over the length of the series, so it fluctuates uniformly about a constant level.

Redistribution Modifying Waveform Shape

Redistribution as described in Chapter 7, when applied to series variables' data, goes far toward achieving a stationary series. Any series variable can be redistributed exactly as described for nonseries. However, this is not always an unambiguous blessing! (More dragons.) Whenever the distribution of a variable is altered, the transform required is captured so that it can always be undone. Indeed, the PIE-O has to undo any transformation for any output variables. However, it may be that the exact shape of the waveform is important to the modeling tool. (Only the modeler is in a position to know for sure if this is the case at modeling time.) If so, the redistribution may introduce unwanted distortion. In Figure 9.22, the top-left image shows a histogram of the distribution of values for the sine wave. Redistribution creates a rectangular distribution, shown in the top-right image. But redistribution changes the nature of the shape of the wave! The lower image shows both a sine wave and the wave shape after redistribution. Redistribution is intended to do exactly what is seen here—all of the nonlinearity has been removed. The curved waveform is translated into a linear representation—thus the straight lines. This may or may not cause a problem. However, the miner must be aware of the issue.

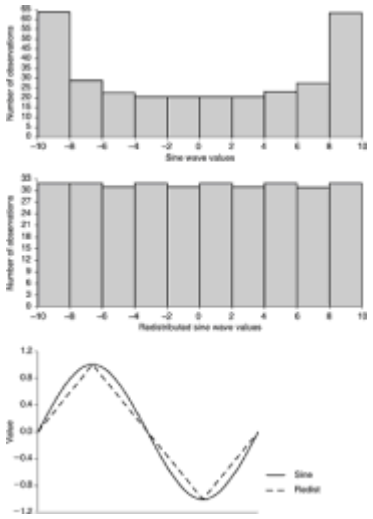

Figure 9.22 Redistributing the distribution linearizes the nonlinear waveform. As the distribution of a pure sine wave is adjusted to be nearer rectangular, so the curves are straightened. If maintaining the wave shape is important, some other transform is required.

Distribution Maintaining Waveform Shape

Redistribution goes a long way toward equalizing the variance. However, some other method is required if the wave shape needs to be retained. If the variance of the series changes as the series progresses, it may be possible to transform the values so that the variance is more constant. Erratic fluctuations of variance over the length of the series cause more problems, but may be helped by a transformation. A "Box-Cox" transformation (named after the people who first described it) may work well. The

transform is fairly simple to apply, and is as follows:

$$v_t = \frac{v_a^{\lambda} - 1}{\lambda}$$

where

v_t is transformed value
v_a is original value
λ is a user-selected value

When the changing variance is adjusted, the distribution still has to be balanced. A second transform accomplishes this. The second transform subtracts the mean of the transformed variable from each transformed value, and divides the result by the standard deviation. The formula for this second transformation is

$$v_s = \frac{v_{t_1} - \overline{v}_{t_1}}{\sigma_{v_{t_1}}}$$

where

v_{t_1} is the value after the first transform
v_s is standardized value
$\overline{v}_{t_1}$ is mean value of variable v_{t_1}
$\sigma_{v_{t_1}}$ is standard deviation of variable v_{t_1}

The index, or displacement, variable should not be redistributed, even if it is of unequal increments.

9.7.3 Normalization

Normalization over the range of 0 to 1 needs no modification. The displacement variable can be normalized using exactly the same techniques (described in Chapter 7) that work for nonseries data.

9.7 Other Problems

So far, the problems examined have been specific to series data. The solutions have focused on ways of extracting information from noisy or distorted series data. They have involved extracting a variety of waveforms from the original waveform that emphasize particular aspects of the data useful for modeling. But whatever has been pulled out, or extracted, from the original series, it is still in the form of another series. It is quite possible to look at the distribution of values in such a series exactly as if it were not a series. That is to say, taking care not to actually lose the indexing, the variable can be treated exactly as if it were a nonseries variable. Looking at the series this way allows some of the tools used for nonseries data to be applied to series data. Can this be done, and where does it help?

9.7.1 Numerating Alpha Values

As mentioned in the introduction to this chapter, numeration of alpha values in a series presents some difficulties. It can be done, but alpha series values are almost never found in practice. On the rare occasions when they do occur, numerating them using the nonseries techniques already discussed, while not providing an optimal numeration, does far better than numeration without any rationale. Random or arbitrary assignment of values to alpha labels is always damaging, and is just as damaging when the data is a series. It is not optimal because the ordering information is not fully used in the numeration. However, using such information involves projecting the alpha values in a nonlinear phase space that is difficult to discover and computationally intense to manipulate. Establishing the nonlinear modes presents problems because they too have to be constructed from the components cycle, season, trend, and noise. Accurately determining those components is not straightforward, as we have seen in this chapter. This enormously compounds the problem of in-series numeration.

The good news is that, with time series in particular, it seems easier to find an appropriate rationale for numerating alpha values from a domain expert than for nonseries data. Reverse pivoting the alphas into a table format, and numerating them there, is a good approach. However, the caveat has to be noted that since alpha numerated series occur so rarely, there is little experience to draw on when preparing them for mining. This makes it difficult to draw any hard and fast general conclusions.

9.7.2 Distribution

As far as distributions are concerned, a series variable has a distribution that exists without reference to the ordering. When looked at in this way, so long as the ordering—that is, the index variable—is not disturbed, the displacement variable can be redistributed in exactly the same manner as a nonseries variable. Chapter 7 discussed the nature of distributions, and reasons and methods for redistributing values. The rationale and methods of redistribution are similar for series data and may be even more applicable in some ways. There are time series methods that require the variables' data to be centered (equally distributed above and below the mean) and normalized. For series data, the distribution should be normalized after removing any trend.

When modeling series data, the series should, if possible, be what is known as stationary. A *stationary* series has no trend and constant variance over the length of the series, so it fluctuates uniformly about a constant level.

Redistribution Modifying Waveform Shape

Redistribution as described in Chapter 7, when applied to series variables' data, goes far toward achieving a stationary series. Any series variable can be redistributed exactly as described for nonseries. However, this is not always an unambiguous blessing! (More dragons.) Whenever the distribution of a variable is altered, the transform required is

captured so that it can always be undone. Indeed, the PIE-O has to undo any transformation for any output variables. However, it may be that the exact shape of the waveform is important to the modeling tool. (Only the modeler is in a position to know for sure if this is the case at modeling time.) If so, the redistribution may introduce unwanted distortion. In Figure 9.22, the top-left image shows a histogram of the distribution of values for the sine wave. Redistribution creates a rectangular distribution, shown in the top-right image. But redistribution changes the nature of the shape of the wave! The lower image shows both a sine wave and the wave shape after redistribution. Redistribution is intended to do exactly what is seen here—all of the nonlinearity has been removed. The curved waveform is translated into a linear representation—thus the straight lines. This may or may not cause a problem. However, the miner must be aware of the issue.

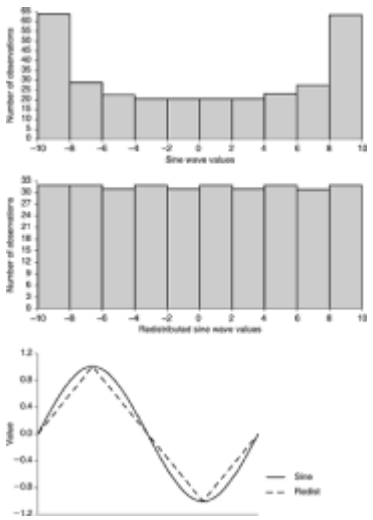

Figure 9.22 Redistributing the distribution linearizes the nonlinear waveform. As the distribution of a pure sine wave is adjusted to be nearer rectangular, so the curves are straightened. If maintaining the wave shape is important, some other transform is required.

Distribution Maintaining Waveform Shape

Redistribution goes a long way toward equalizing the variance. However, some other method is required if the wave shape needs to be retained. If the variance of the series changes as the series progresses, it may be possible to transform the values so that the variance is more constant. Erratic fluctuations of variance over the length of the series cause more problems, but may be helped by a transformation. A “Box-Cox” transformation (named after the people who first described it) may work well. The transform is fairly simple to apply, and is as follows:

$$v_t = \frac{v_a^{\lambda} - 1}{\lambda}$$

where

v_t is transformed value
v_a is original value
λ is a user-selected value

When the changing variance is adjusted, the distribution still has to be balanced. A second transform accomplishes this. The second transform subtracts the mean of the transformed variable from each transformed value, and divides the result by the standard deviation. The formula for this second transformation is

$$v_s = \frac{v_{t_1} - \overline{v}_{t_1}}{\sigma_{v_{t_1}}}$$

where

v_{t_1} is the value after the first transform
v_s is standardized value
$\overline{v}_{t_1}$ is mean value of variable v_{t_1}
$\sigma_{v_{t_1}}$ is standard deviation of variable v_{t_1}

The index, or displacement, variable should not be redistributed, even if it is of unequal increments.

9.7.3 Normalization

Normalization over the range of 0 to 1 needs no modification. The displacement variable can be normalized using exactly the same techniques (described in Chapter 7) that work for nonseries data.

9.8 Preparing Series Data

A lot of ground was covered in this chapter. A brief review will help before pulling all the pieces together and looking at a process for actually preparing series data.

- Series come in various types, of which the most common by far is the time series. All series share a common structure in that the ordering of the measurements carries information that the miner needs to use.

- Series data can be completely described in terms of its four component parts: trend, cycles, seasonality, and noise. Alternatively, series can also be completely described as consisting of sine and cosine waveforms in various numbers and of various amplitudes, phases, and frequencies. Tools to discover the various components include Fourier analysis, power spectra, and correlograms.

- Series data are modeled either to discover the effects of time or to look at how the data

changes in time.

- Series data shares all the problems that nonseries data has, plus several that are unique to series.

  — Missing values require special procedures, and care needs to be taken not to insert a pattern into the missing values by replicating part of a pattern found elsewhere in the series.

  — Nonuniform displacement is dealt with as if it were any other form of noise.

  — Trend needs special handling, exactly as any other monotonic value.

- Various techniques exist for filtering out components of the total waveform. They include, as well as complex mathematical devices for filtering frequencies,

  — Moving averages of various types. A moving average involves using lagged values over the series data points and using all of the lagged values in some way to reestimate the data point value. A large variety of moving average techniques exist, including simple moving averages (SMAs), weighted moving averages (WMAs), and exponential moving averages (EMAs).

  — Smoothing techniques of several types. Smoothing is a windowing technique in which a window of adjustable length selects a particular subseries of data points for manipulation. The window slides over the whole series and manipulates each separate subset of data points to reestimate the window's central data point value. Smoothing techniques include peak-valley-mean (PVM), median smoothing, and Hanning.

  — Resmoothing is a smoothing technique that involves either reapplying the same smoothing technique several times until no change occurs, or applying different window sizes or techniques several times.

- Differencing and reverse differencing (summing) offer alternative ways of looking at high- or low-frequency components of a waveform. Differencing and summing also transform waveforms in ways that may give clues to underlying randomness.

- The series data alone cannot ever be positively determined to contain a random component, although additional tests can raise the confidence level that detected noise is randomly generated. Only a rationale or causal explanation external to the data can confirm random noise generation.

- Components of a waveform can be separated out from the original waveform using one or several of the above techniques. These components are themselves series that