Paper 080-30

Mining Transactional and Time Series Data

Michael Leonard, SAS Institute Inc., Cary, NC
Brenda Wolfe, SAS Institute Inc., Cary, NC

ABSTRACT

Web sites and transactional databases collect large amounts of time-stamped data related to an organization's suppliers and/or customers over time. Mining these time-stamped data can help business leaders make better decisions by listening to their suppliers or customers via their transactions collected over time. A business can have many suppliers and/or customers and may have a set of transactions associated with each one. However, the size of each set of transactions may be quite large, making it difficult to perform many traditional data-mining tasks. This paper proposes techniques for large-scale reduction of time-stamped data using time series analysis, seasonal decomposition, and automatic time series model selection. After data reduction, traditional data mining techniques can then be applied to the reduced data along with other profile data. This paper demonstrates these techniques using SAS/ETS®, SAS/STAT®, Enterprise Miner^{TM}, and SAS® High-Performance Forecasting software.

INTRODUCTION

Businesses often want to discover knowledge from their time-stamped data stored in their transactional or time series databases. Web sites, point-of-sale (POS) systems, call centers, and inventory systems are examples of transactional databases. A skilled analyst can analyze a single set of transactions or a single time series by using various time series analysis and statistical techniques, good software based on proven statistical theory, and sound judgment based on his or her knowledge and experience. Generating large numbers of analyses and/or frequently generating analyses requires some degree of simplification and automation. Common problems that a business faces are

- No skilled analyst is available for detailed analysis.
- Frequent updates to the analyses are required.
- Many sets of transactions must be analyzed.
- Each set of transactions is large.
- Time-stamped transactional data must be converted to time series data.
- Statistical models are not known and must be discovered.

This paper presents ways to help solve these problems by proposing techniques for large-scale reduction of time-stamp data for subsequent data mining. For the data miner, this paper provides a brief background on transactional data analysis, time series analysis, seasonal decomposition, time series models, and automatic time series model selection. For the time series analyst, this paper provides a brief background on distance and similarity measures, as well as traditional data mining tasks (cluster analysis and decision tree analysis). Additionally, this paper discusses large-scale visualization related to transactional and time series data.

SCOPE

This paper focuses on mining (discrete) time-stamped data generated by business and economic activity. The techniques described in this paper are less applicable to voice, image, multimedia, medical, and scientific data or continuous-time data.

BACKGROUND

This section provides a brief theoretical background on large-scale reduction of time-stamp data. It is intended to provide the analyst with motivation, orientation, and references. An introductory discussion of time series analysis and forecasting can be found in Makridakis, Wheelwright, and Hyndman (1997), Brockwell and Davis (1996), and Chatfield (2000). A more detailed discussion of time series analysis and forecasting can be found in Box, Jenkins, and Reinsel (1994), Hamilton (1994), Fuller (1995), and Harvey (1994). An introductory discussion of data mining can be found in Barry and Linoff (1997). A more detailed discussion of data mining can be found in Han and Kamber (2001). A detailed discussion of data preparation for data mining can be found in Pyle (1999).

TRANSACTIONAL DATA

Transactional data are time-stamped data collected over time at no particular frequency. Some examples of transactional data are

- Internet data
- Point of Sales (POS) data
- Inventory data

- Call Center data
- Trading data

Figure 1 illustrates an example of raw transactional data (only the first year is shown). Figure 2 illustrates monthly intervals.

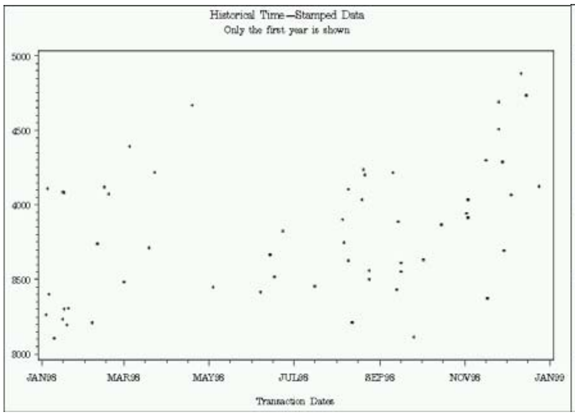


Figure 1: Transactional Series

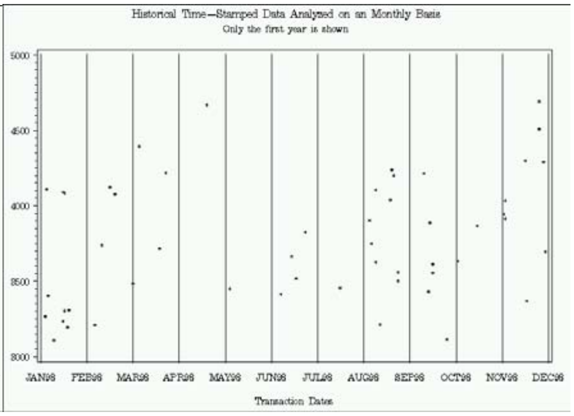


Figure 2: Monthly Intervals

TRANSACTIONAL (TREND AND SEASONAL) ANALYSIS

Businesses often want to analyze transactional data for trends and seasonal variation. To analyze transactional data for trends and seasonality, statistics must be computed for each time period and season of concern. The frequency and the season may vary with the business problem. For example, various statistics can be computed on each time period and season, such as

- Web site visits by hour and by hour of day
- Sales per month and by month of year
- Inventory draws per week and by week of month
- Calls per day and by day of week
- Trades per weekday and by weekday of week

Figure 2 illustrates the “binning” of transactional data based on a monthly frequency (only the first year is shown). Figure 3 illustrates the seasonal statistics (totals) for each month. Figure 4 illustrates the trend statistics (totals) for a single month (March).

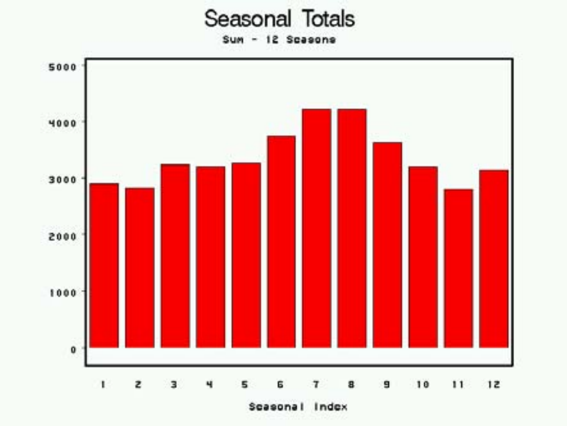


Figure 3

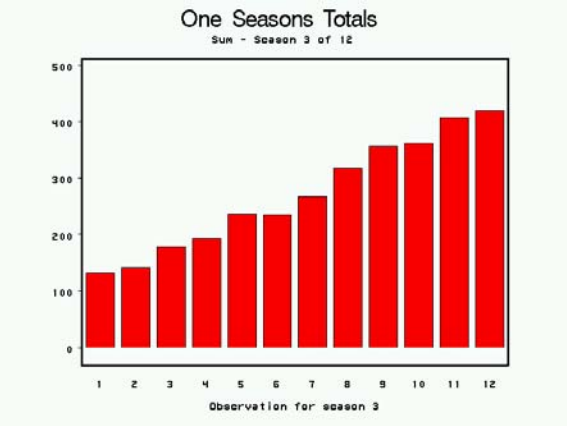


Figure 4

Trend and seasonal statistical analysis of time-stamped data can help reduce the information contained in a single set of transactions to a small set of statistics.

TIME SERIES DATA

Time series data are time-stamped data collected over time at a particular frequency. Some examples of time series data are

- Web site visits per hour

- Sales per month
- Inventory draws per week
- Calls per day
- Trades per weekday

As can be seen, the frequency associated with the time series varies with the problem at hand. The frequency or time interval may be hourly, daily, weekly, monthly, quarterly, yearly, or many other variants of the basic time intervals. The choice of frequency is an important modeling decision.

Associated with each time series is a seasonal cycle or seasonality. For example, the length of seasonality for a monthly time series is usually assumed to be 12 because there are 12 months in a year. Likewise, the seasonality of a daily time series is usually assumed to be 7. The usual seasonality assumption may not always hold. For example, if a particular business's seasonal cycle is 14 days long, the seasonality is 14, not 7.

Time series that consist of mostly zero values (or a single value) are called interrupted or intermittent time series. These time series are mainly constant valued except for relatively few occasions. Figure 5 illustrates time series data. Figure 6 illustrates intermittent time series data.

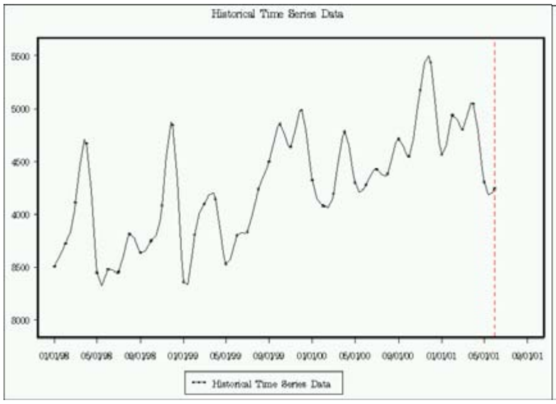


Figure 5: Time Series

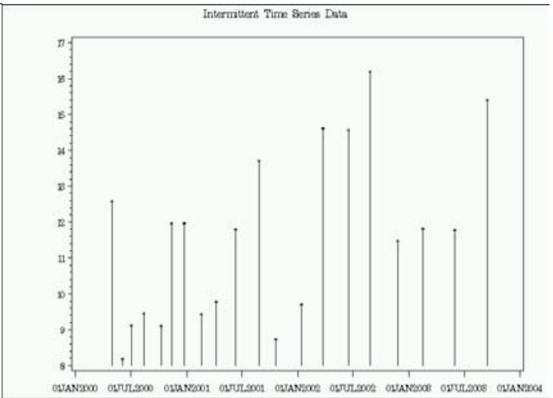


Figure 6: Intermittent Time Series

ACCUMULATING TRANSACTIONAL DATA

The accumulation of time-stamped data into time series data is based on a particular frequency. For example, time-stamped data can be accumulated to form hourly, daily, weekly, monthly, or yearly time series. Additionally, the method for accumulating the transactions within each time period is based on a particular statistic. For example, the sum, mean, median, minimum, maximum, standard deviation, and other statistics can be used to accumulate the transactions within a particular time period.

There may be no data recorded for certain time periods, resulting in missing values in the accumulated time series. These missing values can represent unknown values and thus are left missing, or they can represent no activity, in which case they should be set to zero or some other appropriate value.

Figure 7 illustrates an example of accumulating transactional data in monthly "bins" based on the average value (only the first year is shown). Figure 8 illustrates the resulting time series (multiple years shown).

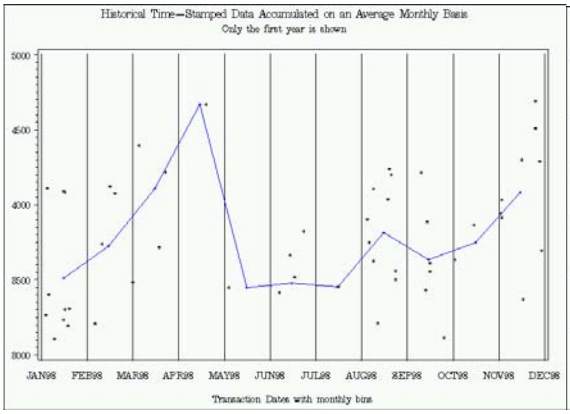


Figure 7: One Year of Transactions

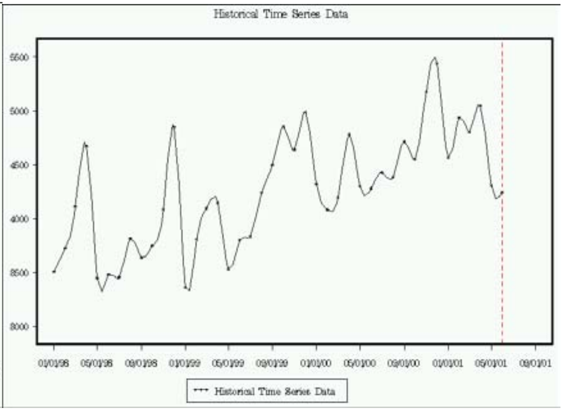


Figure 8: Accumulated Time Series

For a large set of transactions, accumulating the transaction to form a time series often results in a data reduction. Once the transactional data is accumulated to form a time series, time domain and frequency domain analysis of the accumulated time series can further reduce the data.

FREQUENCY DOMAIN ANALYSIS

Typically, frequency domain analysis is employed to decompose a time series into several orthonormal periodic components. Computing the frequency domain properties of the time series can help reduce the amount of information that must be analyzed. Some frequency domain analysis techniques include the following:

- Periodogram Analysis
- Spectral Density Analysis
- White Noise Tests
- Wavelet Analysis
- and many others

Figure 9 and Figure 10 illustrate the periodograms associated with two time series with period 12 (monthly) indicated by the vertical line.

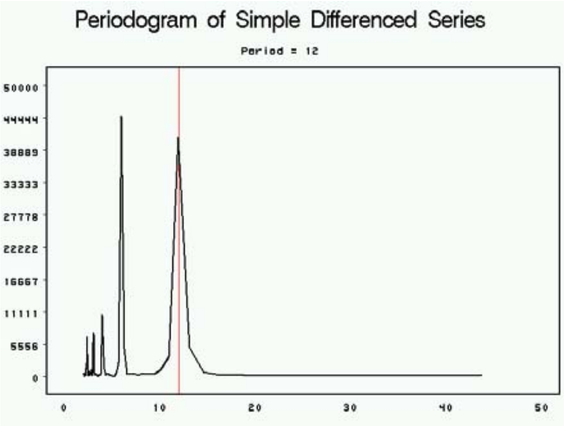


Figure 9: Peak at Period 12

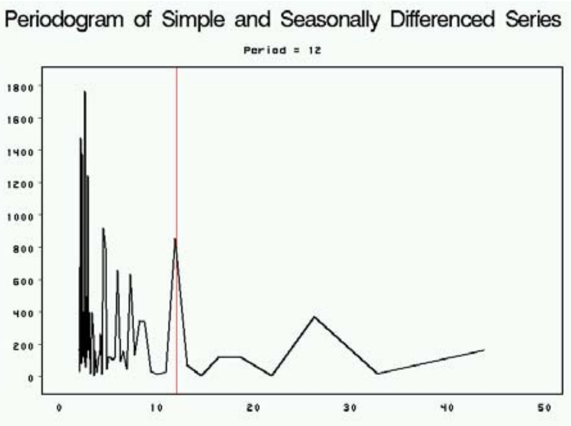


Figure 10: No Peak at Period 12

Frequency domain analysis can reduce a single time series to a small set of frequency domain statistics.

TIME DOMAIN ANALYSIS

Typically, time domain analysis is employed to understand the relationship between the current observation of a time series and previous observations (time lags). Computing the time domain properties of the time series can help reduce the amount of information that must be analyzed. Some time domain analysis techniques include the following:

- Autocorrelation/partial autocorrelation/inverse autocorrelation analysis
- Extended sample autocorrelation analysis
- Smallest canonical correlation analysis
- Stationarity analysis (nonseasonal and seasonal)
- and many others

Figure 11 illustrates the autocorrelations and white noise probabilities associated with a monthly series. For monthly series, time lags that are multiples of 12 (1, 12, 24) are of particular interest.

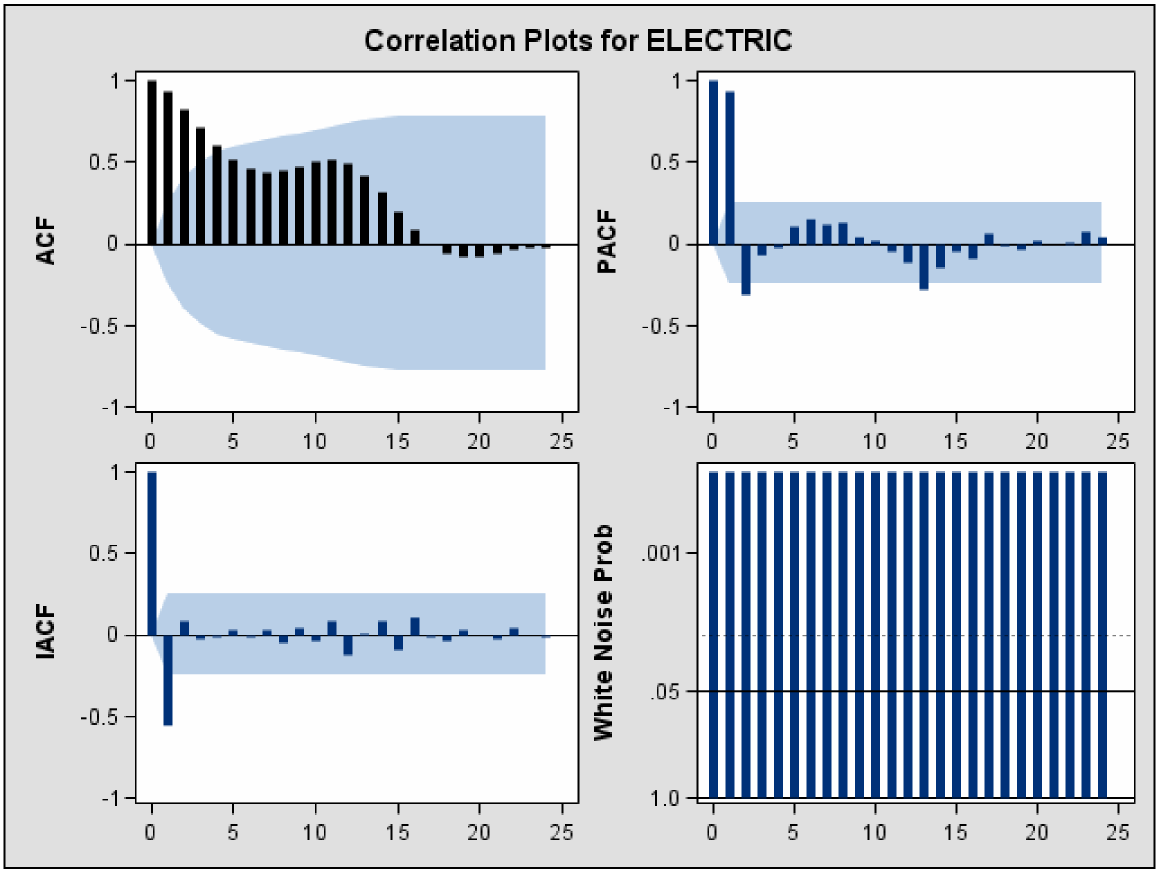


Figure 11: Autocorrelation Analysis

Table 1 illustrates an example of the autocorrelations for several time series (A, B, C, D, …). The table shows the standardized autocorrelations (autocorrelation divided by standard error) for four lags of interest.

Series	Lag 1	Lag2	Lag 3	Lag 12
A	6.28444	4.49058	4.20230	1.98727
B	0.46690	1.74200	1.04061	-1.15256
C	-3.49831	-1.48204	0.86102	0.80821
D	-0.56481	-0.72690	-2.07045	7.17706
…	…		…	…

Table 1: Autocorrelation Table

Time domain analysis can reduce a single time series to a small set of time domain statistics.

SEASONAL DECOMPOSITION ANALYSIS

Time series often have an (unobserved) seasonal component that causes the time series to fluctuate with the changing seasons. For example, ice cream sales are higher in the summer months than in the winter months. In general, seasonal decomposition techniques decompose the original time series $\left(O_t\right)$ into seasonal $\left(S_t\right)$, trend $\left(T_t\right)$, cycle $\left(C_t\right)$, and irregular $\left(I_t\right)$ components. Computing the seasonally decomposed properties of the time series can help reduce the amount of information that must be analyzed.

There are four commonly used seasonal decomposition techniques:

Additive $$O_t = TC_t + S_t + I_t$$

Multiplicative $$O_t = TC_tS_tI_t$$

Log-Additive $\log(O_t) = TC_t + S_t + I_t$

Pseudo-Additive $O_t = TC_t(S_t I_t - 1)$

The trend-cycle component (TC_t) can be further decomposed into trend (T_t) and cycle (C_t) components using fixed-parameter filtering techniques.

Figure 12 illustrates the seasonal decomposition of a time series.

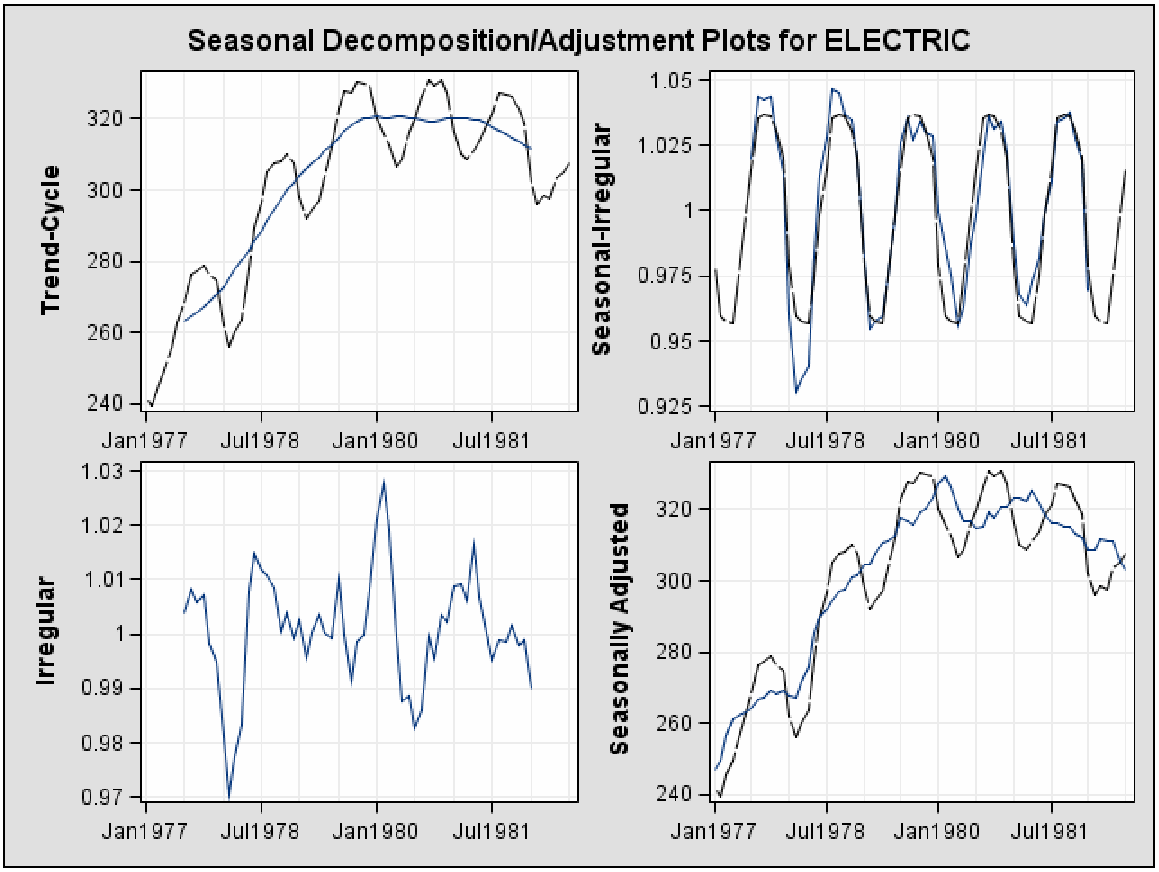


Figure 12: Seasonal Decomposition

Table 2 illustrates an example of multiplicative seasonal decomposition applied to several monthly time series (A, B, C, D, …). The table shows the seasonal component of each series.

Series	Season 1	Season 2	…	Season 12
A	1.00506	0.99529	…	1.00542
B	0.96774	0.98312	…	0.97052
C	1.02036	0.99774		0.99691
D	1.02036	0.99774	…	1.01803
…	…	…	…	…

Table 2: Seasonal Decomposition Table

Seasonal decomposition can reduce a single time series to a small set of (future) seasonal indices (whose size is related to the seasonality) or time-varying trend statistics.

TIME SERIES MODELS

Time series models are used to describe the underlying data-generating process of a time series. Applying a time series model to a time series can help reduce the amount of information that must be analyzed. Additionally, after fitting the time series model to the time series data, the fitted model can be used to determine departures (outliers) from the (assumed) data-generating process or forecast function components (future trend, seasonal or cycle estimates). Some time series modeling techniques include the following:

- ARIMA
- GARCH
- State space or unobserved component models
- Exponential Smoothing Models
- Intermittent (Interrupted) Models
- and many others

Figure 13 illustrates a time series model and its predictions. Figure 14 illustrates the final smoothed seasonal component.

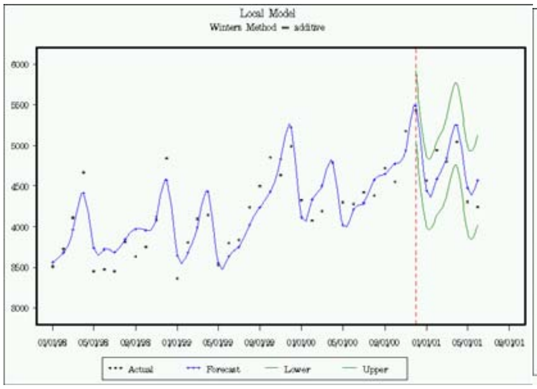


Figure 13: Forecasts

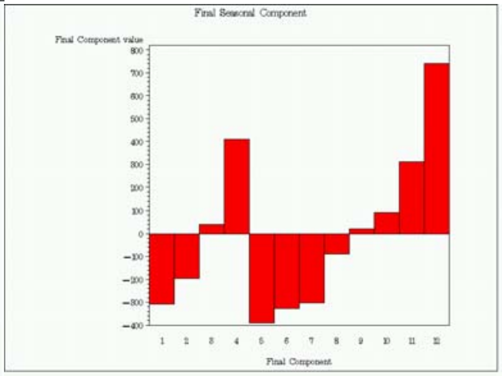


Figure 14: Smoothed Seasonal Component

Table 3 illustrates an example of a single time series model applied to several time series (A, B, C, D, …). The table shows the final estimated components associated with each model.

Series	Level	Trend	Season 1	Season 2	…	Season 12
A	3406	34	101	56	…	43
B	1023	40	33	26	…	34
C	2202	-2	99	1004		103
D	90	-10	200	302	…	344
…	…	…	…	…	…	…

Table 3: Model Components

Time series modeling can reduce a single time series to a small set of modeling parameters, final components (level, slope, season, and/or cycle), or departures from the assumed data-generating process.

Intermittent time series must be modeled differently from nonintermittent time series. Intermittent models decompose the time series into two parts: the interval series and the size series. The interval series measure the number of time periods between departures. The size series measures the magnitude of the departures. After this decomposition, each part is modeled and forecast independently. The interval forecast predicts when the next departure will occur. The size forecast predicts the magnitude of the next departure. After the interval and size predictions are computed, they are combined (predicted magnitude divided by predicted number of periods for the next departure) to produce a forecast for the average departure from the constant value for the next time period.

The intermittent data illustrated in Figure 15 were modeled using an intermittent time series model. Figure 15 and Figure 16 illustrate the actual values and predicted values of the demand size and interval series, respectively. Figure 17 illustrates the predicted average stocking levels, and Figure 18 illustrates the stocking levels and stocking errors.

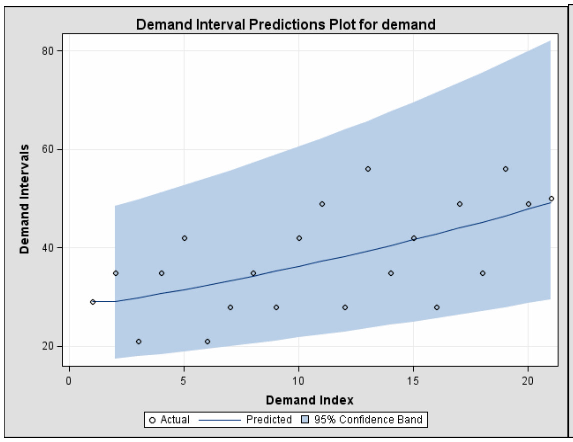


Figure 15: Demand Interval

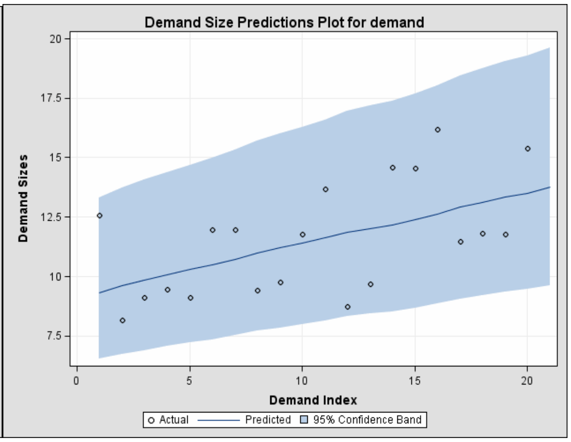


Figure 16: Demand Size

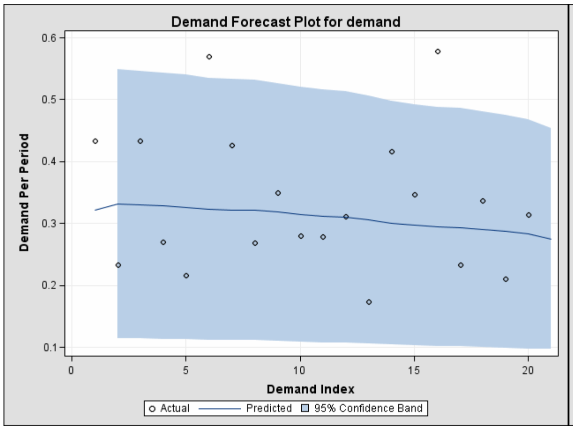


Figure 17: Predictions

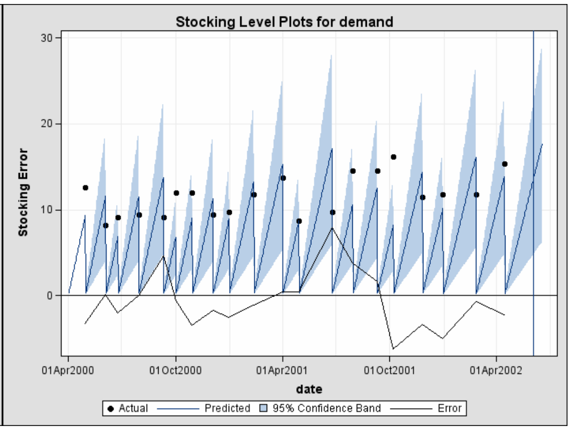


Figure 18: Stocking

Table 4 illustrates an example of an intermittent time series model applied to several time series (A, B, C, D, …). The table shows the final estimated parameters and final smoothed components associated with each series.

Series	Interval Parameter	Interval Component	Size Parameter	Size Component
A	0.18743	4.57633	0.23512	10.2274
B	0.23412	3.49991	0.72839	26.4523
C	0.44023	6.57234	0.03451	15.5677
D	0.80091	3.03413	0.44812	35.4531
…	…	…	…	…

Table 4: Intermittent Demand Model Parameters and Components

Intermittent time series modeling can reduce a single time series to a small set of modeling parameters, final components (demand size, demand interval, average stocking level), or departures from the assumed data-generating process. These models are especially useful for determining the expected time and expected amount of next purchase.

TIME SERIES MONITORING

Fitted time series models can be used to forecast time series. These forecasts can be used to predict future observations as well as to monitor more recent observations for anomalies using holdout sample analysis. For example, after fitting a time series model to the time series data with a holdout sample excluded, the fitted model can be used to forecast within the holdout region. Actual values in the holdout sample that are significantly different from the forecasts could be considered anomalies. The following statistics can be used in holdout sample analysis:

- Performance Statistics (RMSE, MAPE, etc.)
- Prediction Errors (e.g., absolute prediction errors that are three times prediction standard errors)
- Confidence Limits (e.g., actual values outside confidence limits)
- and many others

Figure 19 illustrates a time series model fitted with a six-observation holdout sample excluded. Figure 20 illustrates the actual values and forecasts in the holdout sample. Notice the two anomalies as indicated by the confidence bands.

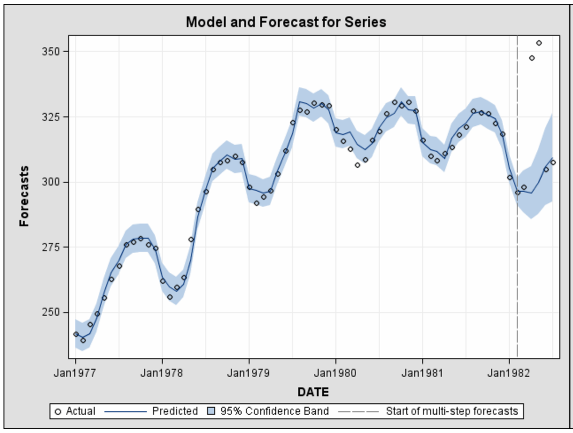


Figure 19: Holdout Sample Fit

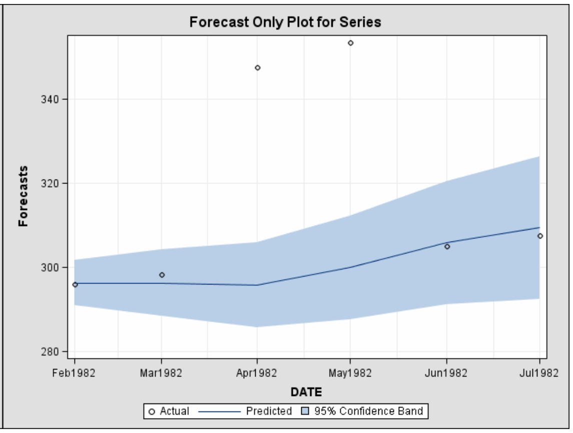


Figure 20: Holdout Sample Forecast

Table 5 illustrates an example of a holdout sample analysis applied to several time series (A, B, C, D, …). The table shows the mean absolute percentage error (MAPE) and the number of actual values outside the confidence band (OUTSIDE) associated with the holdout sample.

Series	MAPE	OUTSIDE
A	10%	0
B	11%	0
C	2202%	5
D	90%	2
…	…	…

Table 5: Holdout Sample Analysis

Time series monitoring can reduce a single time series to a small set of holdout sample statistics. These statistics can be used to determine the presence of anomalies.

AUTOMATIC TIME SERIES MODEL SELECTION

Often, the time series model associated with a time series is not known. This problem is further complicated by the fact that there may be many time series, and no one model explains the data-generating process for all series. Automatic model selection is a technique that selects an appropriate time series model for a given time series. For each time series, a list of candidate models can be chosen based on the time series characteristics (e.g., seasonality, trend, etc.), and an appropriate time series model can be selected from the list of candidate models using a model selection criterion.

Table 6 illustrates an example of automatic model selection for numerous time series (A, B, C, D, …).

Series	Transform	Model Specification	Level Parameter	Trend Parameter	Damping Parameter	Season Parameter
A	None	Local Level	0.2	.		.
B	Log	Local Trend	0.1	0.2	0.9	0.5
C	None	Local Seasonal	0.2	.	.	0.3
D	None	Local Trend	0.4	0.5	0.95	.
…	…	…	…	…	…	...

Table 6: Model Selection Table

Automatic model selection can reduce a single time series to a model specification. The selected model specification can then be fit to the data as described in the previous section.

MULTIVARIATE TIME DOMAIN ANALYSIS

Typically, multivariate time domain analysis is employed to understand the relationships between time series. Computing the time domain properties between two time series can help reduce the amount of information that must be analyzed. Some time domain analysis techniques include the following:

- Cross (Auto) Correlation Analysis
- Partial AR Analysis
- Partial Canonical Correlation Analysis
- and many others

Figure 21 and Figure 22 illustrate the cross-correlations between two monthly series. For monthly series, time lags that are multiples of 12 (…, -24, -12, -1, 0, 1, 12, 24, …) are of particular interest.

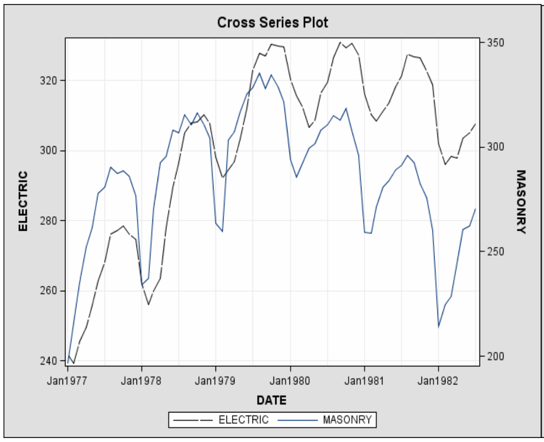


Figure 21: Cross Series

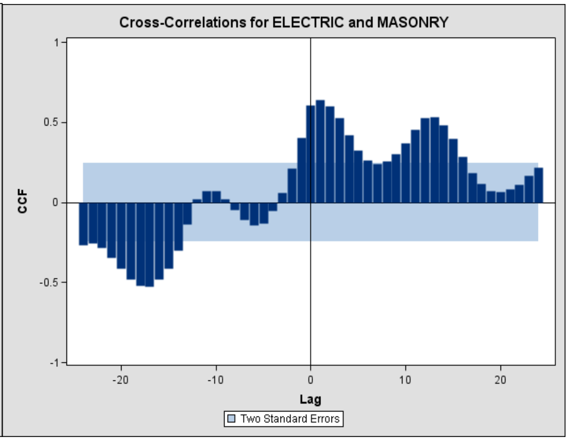


Figure 22: Cross-Correlations

Table 7 illustrates an example of the cross-correlations between a single cross variable to several time series (A, B, C, D, …). The table shows the standardized cross-correlations (cross-correlation divided by standard error) for five lags of interest.

Series	Lag -12	Lag -1	Lag 0	Lag 1	Lag 12
A	-2.92897	1.72270	2.03825	2.31180	5.52200
B	2.58354	5.43210	5.83505	6.22264	9.28630
C	-4.02907	-6.02496	-6.17823	-6.3727	-8.61054
D	-4.06014	-5.80548	-5.91039	-6.05920	-8.02902
…	…	…	…	…	…

Table 7: Cross-Correlations Table

Multivariate time domain analysis can reduce a time series to a small set of multivariate time domain statistics.

CAUSAL TIME SERIES MODELS

Causal time series models are used to describe the underlying data-generating process of a dependent time series using input (predictor or explanatory) variables. Applying a causal time series model to a time series can help reduce the amount of information that must be analyzed, similar to time series models as described above. Additionally, after fitting the causal time series model to the time series data, the fitted model can be used to determine the influence of the input variables. Some causal time series modeling techniques include the following:

- ARIMAX
- ARIMAX-GARCH
- State Space or Unobserved Component Models
- Nonlinear Multivariate Time Series Models
- and many others

For example, when there many multivariate time series that represent the quantity sold and price associated with demand for a product or service, it is often desirable to determine the influence of price on demand.

Figure 23 and Figure 24 illustrate the influence associated with the causal variable, PRICE, on the dependent variable, QUANTITY, for one multivariate time series.

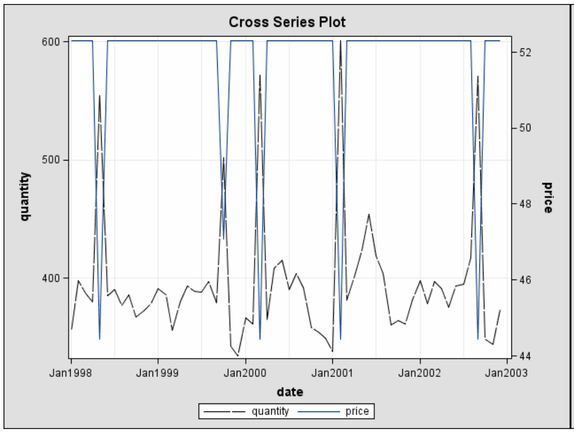


Figure 23: Cross Series

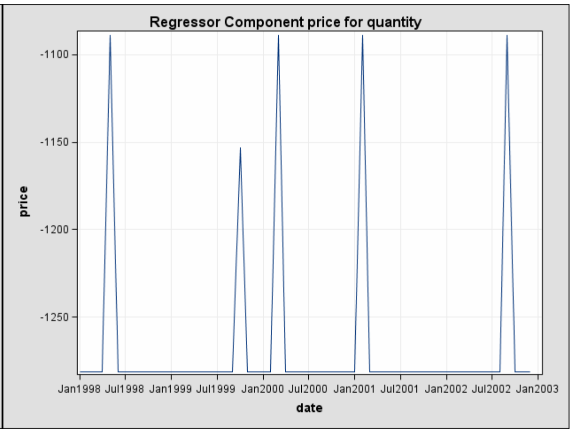


Figure 24: Causal Effect

Table 8 shows the parameter estimates for this causal factor associated with each series.

Series	Parameter	Estimate	Standard Error	T-Value	P-Value
A	Price	-23.2264	1.52663	-15.2141	<0.0001
B	Price	-10.9646	0.72782	-15.0649	<0.0001
C	Price	-11.6618	2.21650	-5.2614	<0.0001
D	Price	-8.3573	1.35316	-6.1761	<0.0001
…		…	…	…	…

Table 8: Causal Time Series Model Parameters Estimates

Causal time series modeling can reduce a single time series to a small set of parameter estimates. These models are especially useful for determining the influence of causal factors. For example, the quantity sold can be modeled with the sales price as the causal variable.

INTERVENTION TIME SERIES MODELS

Unlike causal time series models, intervention time series models are used to describe departures from the underlying data-generating process of a dependent time series using indicator (dummy) variables that indicate when an event occurs in time. Applying an intervention time series model to a time series can help reduce the amount of information that must be analyzed, similar to causal time series models as described above. Additionally, after fitting the intervention model to the time series data, the fitted model can be used to determine the influence of the events or intervention effect. Some intervention time series modeling techniques include the following:

- Transfer Function Models
- State Space or Unobserved Component Models
- Nonlinear Multivariate Time Series Models
- and many others

For example, when there are many univariate and/or multivariate time series that represent the demand for a product or service, it is often desirable to determine the influence of a calendar on demand.

Figure 25 illustrates the intervention effect associated with the calendar event CHRISTMAS.

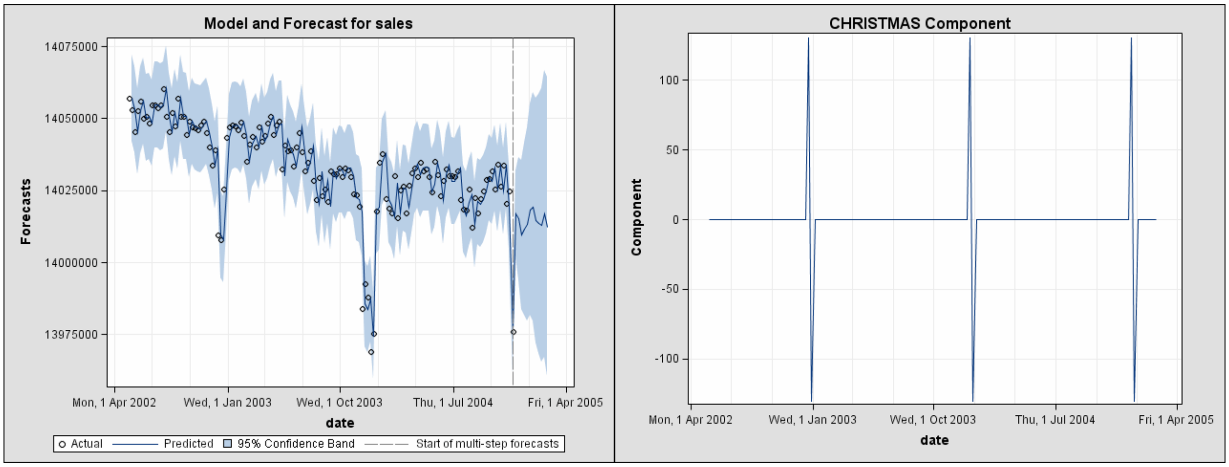


Figure 25: Intervention Effect

Table 9 shows the parameter estimates for the intervention effect associated with each series.

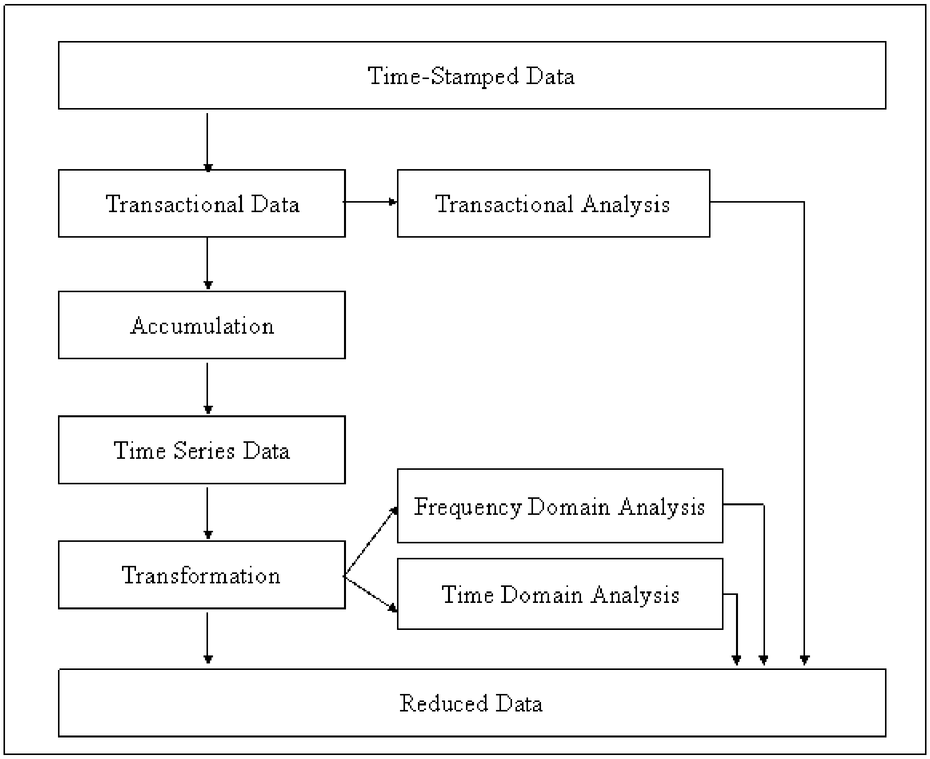
Series	Parameter	Estimate	Standard Error	T-Value	P-Value
A	Christmas	0.18743	4.57633	0.23512	10.2274
B	Christmas	0.23412	3.49991	0.72839	26.4523
C	Christmas	0.44023	6.57234	0.03451	15.5677
D	Christmas	0.80091	3.03413	0.44812	35.4531
…		…	…	…	…

Table 9: Intervention Time Series Model Parameters Estimates

Intervention time series modeling can reduce a single time series to a small set of parameter estimates. These models are especially useful for determining the effect of calendar events. For example, the quantity sold can be modeled with CHRISTMAS as an event.

DIMENSION REDUCTION

Traditional data mining techniques include clustering, classification, decision trees, and others. These analytical techniques are typically applied to large data sets whose observation vectors are relatively small in dimension when compared to the length of a transaction series or time series. In order to effectively apply these data mining techniques to a large number of series, the dimension of each series must be reduced to a small number of statistics that capture their descriptive properties. As described above, various transactional and time series analysis techniques (possibly in combination) can be used to capture these descriptive properties for each time series. The diagram below summarizes the dimension reduction process of transactional and time series data.



Many transactional and time series databases store the data in longitudal form, whereas many data mining software packages require the data to be in coordinate form. As described above, dimension reduction extracts important features of the longitudinal dimension of the series and stores the reduced sequence in coordinate form of fixed dimension. Assume that there are N series with lengths $\{T_1,\dots,T_N\}$.

In longitudinal form, each variable (or column) represents a single series, and each variable observation (or row) represents the series value recorded at a particular time. Notice that the length of each series, T_i, can vary.

$Y_i = \{y_{i,t}\}_{t=1}^{T_i}$ for $i = 1,..,N$

where Y_i is $(T_i \times 1)$. This form is convenient for time series analysis but less desirable for data mining.

In coordinate form, each observation (or row) represents a single reduced sequence, and each variable (or column) represents the reduced sequence value. Notice that the length of each reduced sequence, M, is fixed.

$R_i = \{r_{i,m}\}_{m=1}^{M}$ for $i = 1,..,N$

where R_i is $(1 \times M)$. This form is convenient for data mining but less desirable for time series analysis.

To reduce a single series, a univariate reduction transformation is needed that maps the varying longitudinal dimension to the fixed coordinate dimension.

$R_i = F_i[Y_i]$ for $i = 1,..,N$

where R_i is $(1 \times M)$, Y_i is $(T_i \times 1)$, and $F_i[]$ is the reduction transformation (e.g., seasonal decomposition).

For multivariate series reduction, more than one series is reduced to a single reduction sequence. The bivariate case is illustrated.

$R_i = F_i[Y_i, X_i]$ for $i = 1,..,N$

where R_i is $(1 \times M)$, Y_i is $(T_i \times 1)$, X_i is $(T_i \times 1)$, and $F_i[]$ is the reduction transformation (e.g., cross-correlations).

In the above discussion, the reduction transformation, $F_i[]$, is indexed by the series index, $i = 1,..,N$, but typically it does not vary and further discussion assumes it to be the same, that is, $F[] = F_i[]$.

Table 10 and Table 11 illustrate an example of the above formulas in tabular form. '.' refers to a missing value, and '…' refers to continuation.

Series Table—Longitudinal Form—each series of varying dimension			
Time Index	**Y_1**	**…**	**Y_N**
1	$y_{1,1}$	$\cdots$	$y_{N,1}$
2	$y_{1,2}$	$\cdots$	$y_{N,2}$
…	…	…	…
T_1	y_{1,T_1}	$\cdots$	y_{N,T_1}
…	.	…	…
T_N	.	$\cdots$	y_{N,T_N}

Table 10: Series Table

Reduced Table—Coordinate Form—each sequence of fixed dimension			
Series Index/Reduced Row	**$r_{.,1}$**	**…**	**$r_{.,M}$**
1 **R_1**	$r_{1,1}$	$\cdots$	$r_{1,M}$
…		…	
N **R_N**	$r_{N,1}$	$\cdots$	$r_{N,M}$

Table 11: Reduced Table

Dimension reduction transforms the series table $(T \times N)$ to the reduced table $(N \times M)$ where $T = \max\{T_1,...,T_N\}$ and where typically $M < T$. The number of series, N, can be quite large; therefore, even a simple reduction transform requires the manipulation of a large amount of data. Hence, it is important to get the data in the proper format to avoid the post-processing of large data sets.

Time series analysts often like to analyze the reduced table set in longitudinal form, whereas data miners often like to analyze the reduced data set in coordinate form. Transposing a large table from longitudinal form to coordinate form and vice-versa form can be computationally expensive.

COMBINING OTHER CATEGORICAL DATA

As described above, transactional and time series analysis can reduce a single transactional or time series to a relatively small number of descriptive statistics. The reduced data can be combined or merged with other categorical data (e.g., age, gender, income, etc.).

For example, suppose that the rather large transaction history of a single customer is reduced to a small number of statistics using seasonal decomposition; the seasonal indices can be combined with the customer's income and gender. This combined data set can then be analyzed using both the seasonal indices and categorical data. Such an analysis would be difficult or impossible if the (high-dimension) series data was used directly.

For example, the seasonal decomposition reduction illustrated in Table 2 can be combined with income and gender information to form a new table, illustrated in Table 12.

Series	Season 1	Season 2	…	Season 12	Income	Gender
A	1.00506	0.99529	…	1.00542	100,357	Male
B	0.96774	0.98312	…	0.97052	45,299	Female
C	1.02036	0.99774		0.99691	85,190	Female
D	1.02036	0.99774	…	1.01803	10,140	Male
…	…	…	…	…		…

Table 12: Combined Table

DISTANCE AND SIMILARITY MEASURES

Clustering algorithms require a definition of how to compute the distances, dissimilarity, or similarity between two data vectors. These definitions must obey the axioms of a metric space:

$$\begin{array}{ll} & d(a,a)=0 \\ & d(a,b)=0 \Rightarrow a=b \\ d: M \times M \Rightarrow \Re & d(a,b) \geq 0 \\ a,b,c \in M & d(a,c) \leq d(a,b)+d(b,c) \end{array}$$

In this paper, distance measures refer to metrics applied to the reduced (coordinate) data, and similarity measures refer to metrics applied to series (longitudinal) data.

DISTANCE MEASURES

A wide variety of distance measures can be applied to coordinate data. Data are classified to various levels of measurement, including nominal, ordinal, interval, and ratio levels, and data can be either numeric or character. Each measure accepts certain levels of measurement. For instance, the Euclidean distance accepts ratio, interval, and ordinal levels, while the Gower's similarity accepts all levels. Typically, these measures require that each observation vector has the same length, but some allow for missing data. Some of these measures do not strictly follow the axioms of a metric space. The documentation for the DISTANCE procedure of SAS/STAT software describes many types of distance measures that can be applied to coordinate data.

SIMILIARITY MEASURES

Given two ordered numeric sequences (input and target), such as two time series, a similarity measure is a metric that measures the distance between the input and target sequences while taking into account the ordering. Similarity measures can be computed between input sequence and target sequence as well as similarity measures that "slide" the target sequence with respect to the input sequence. The "slides" can be by observation index (sliding-sequence similarity measures) or by seasonal index (seasonal-sliding-sequence similarity measures).

In computing the similarity measure between two time series, tasks are needed for transforming time series, normalizing sequences, scaling sequences, and computing metrics or measures.

For transforming the input and target time series, functional (log, square root, logistics, and Box-Cox) and differencing (simple and seasonal) transformations and others can be used.

For normalizing the input and target sequences, standard and absolute normalization and others can be used.
For scaling the input sequence to the target sequence, standard and absolute scaling and others can be used.
For computing metrics, fixed and dynamic time-warped metrics for squared and absolute deviations and others can be used. Sankoff and Kruskal (1983) discuss time warping.

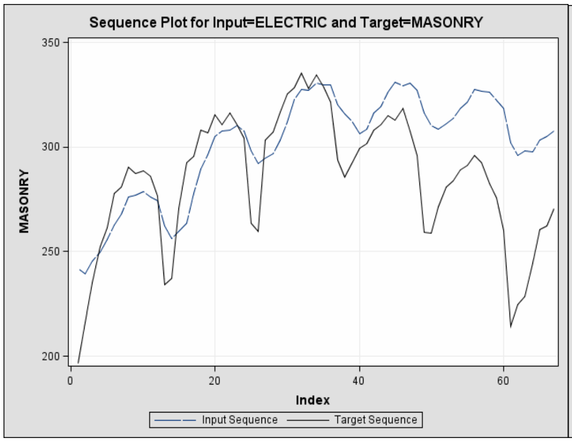
Similarity measures can be used to compare a single target sequence to many other input sequences. This situation arises in time series search and retrieval or ranking. For example, given a single target sequence, you can find input sequences that are "similar" or "close" to the target. Search and retrieval and analogies are important for new product forecasting and analogous time series forecasting. These techniques are useful when a large number of historical time series are available.

Similarity measures can be used to compare a single input sequence to several other representative target sequences. This situation arises in time series classification. For example, given a single input sequence, you can classify the input sequence by finding the "most similar" or "closest" target sequence. This analysis can be repeated to classify large numbers of input sequences.

Similarity measures can be computed between several sequences to form a similarity matrix. For example, given *N* sequences, an (*N* x *N*) symmetric matrix can be constructed whose *ij*th element contains the similarity measure between the *i*th and *j*th sequence. Clustering techniques can then be applied to the similarity matrix. This situation arises in time series clustering.

Sliding similarity measures (observational or seasonal index) can be used to compare a single target sequence to subsequences of many other input sequences on a sliding basis. This situation arises in time series analogies. For example, given a single target series, you can find the times in the history of the input sequence that are similar while preserving the seasonal indices.

The following example illustrates computing similarity measures between a target and input sequence using dynamic time warping. Dynamic time warping expands and compresses the time axis using a dynamic program that minimizes a cost function (e.g., sum of the squared or absolute distances). Figure 26 illustrates the two series (input and target). Figure 27 illustrates the warping path where horizontal movements represent expansion and vertical movements represent compression. Figure 28 illustrates the warping plot that maps the input and target observations. Figure 29 illustrates the distances between the two warped sequences. The similarity measure would be the sum of the squared or absolute distance.



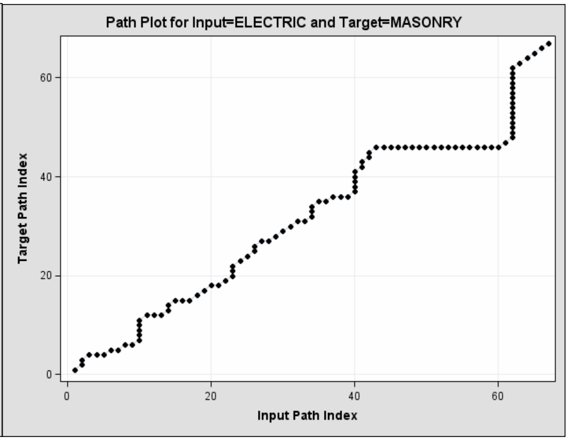
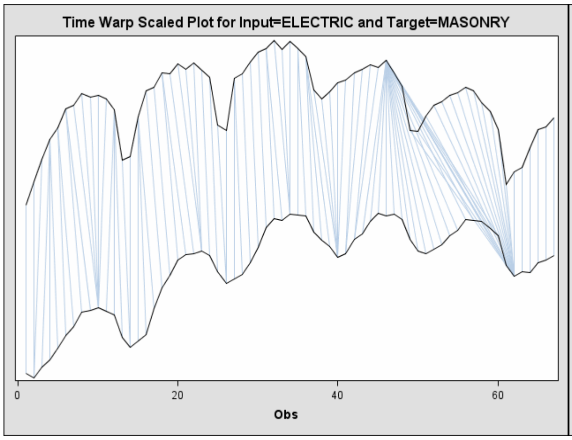


Figure 26: Input and Target Plot

Figure 27: Warping Path



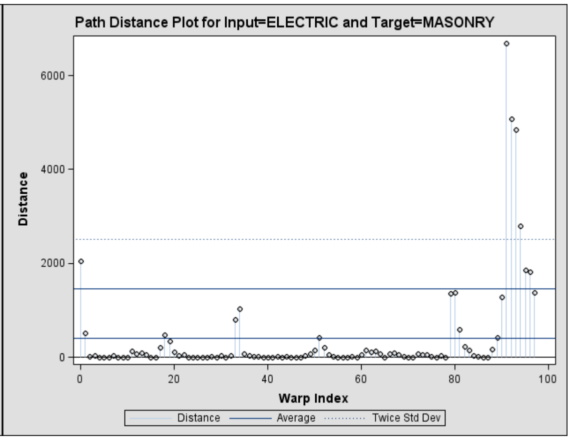


Figure 28: Scaled Warp Plot

Figure 29: Path Distances

DISTANCE AND SIMLARITY MATRICES

A distance matrix is constructed by computing the distance measures between each data set row, R_i.

$$d_{i,j} = d(R_i, R_j) \text{ for } 1 \le i, j \le N$$

$D_i = \{d_{i,j}\}_{j=1}^{N}$ represents the distances related to the *i*th row.

The computational order of distance measures are $O(M)$ and distance matrices are $O(N^2M)$.

When distance measures are applied to the reduced data, the distances between two series are computed indirectly.

$$d_{i,j} = d\left(F[Y_i], F[Y_j]\right) \text{ for } 1 \le i, j \le N$$

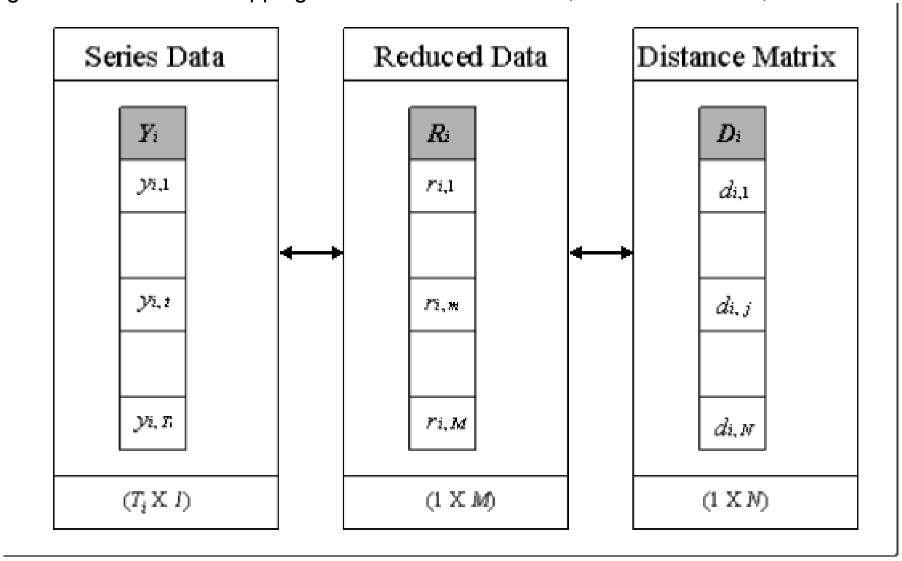
A similarity matrix is constructed by computing the similarity measures between each series.

$$d_{i,j} = Sim(Y_i, Y_j) \text{ for } 1 \le i, j \le N$$

$D_i = \{d_{i,j}\}_{j=1}^{N}$ represents the similarity measures related to the *i*th series

There is a wide variety of similarity measures that can be applied to the series (longitudinal) data. Depending on the method, the computational order of similarity measures vary from $O(T)$ to $O(T^2)$), and distance matrices vary from $O(N^2T)$ to $O(N^2T^2)$. Similarity measures are computationally more expensive than distance measures. Due to the axioms above, only the elements below the diagonal need to be computed.

The following diagram illustrates the mapping between the series data, the reduced data, and the distance matrix.



DATA MINING

Once the transactional series has been accumulated into a time series format, the time series has been reduced, and/or the distance matrix has been computed from the reduced data, data mining techniques can be applied. With an ever-growing number of data mining techniques, this paper focuses on three: sampling, cluster analysis, and decision trees.

SAMPLING

Transactional and time series databases often contain a large number of series. To efficiently explore, model, and assess the model results, random samples must be extracted from the database. The sample should be large enough to contain the significant information, yet small enough to process. For some data mining techniques, such as neural networks, more than one sample is needed (e.g., training, testing, and validation samples). Once a model has been assessed or evaluated for effectiveness, the model can then be applied to the entire database.

There are various ways to sample the data. For reduced (coordinate) data, observation sampling is more appropriate. For series (longitudinal) data, longitudinal sampling is more appropriate.

OBSERVATION SAMPLING

Assume that there are N observations in the reduced (coordinate) data. Typically, for coordinate data, a random sample would be selected from the N observations; that is, N_{SAMPLE} random integers would be selected between one and N without replacement. The sample data would then be created from the observations whose observation index corresponds to one of the N_{SAMPLE} randomly selected integers. The sample data dimensions are $(N_{SAMPLE} \times M)$ with $R_{SAMPLE} \subseteq \{R_1, ..., R_N\}$.

LONGITUDINAL SAMPLING

Assume that there are N series with lengths $\{T_1, ..., T_N\}$. Then, the total number of observations is $NOBS_{TOTAL} = T_1 + ... + T_N$. Using observation sampling, a random sample would be selected from the $NOBS_{TOTAL}$ observations; that is, $NOBS_{SAMPLE}$ random integers would be selected between one and $NOBS_{TOTAL}$ without replacement. The sample data would then be created from the observations whose observation index corresponds to one of the $NOBS_{SAMPLE}$ randomly selected integers. However, for series data, observation sampling is inadequate because the ability to exploit the relationship between observations within a single series is lost.

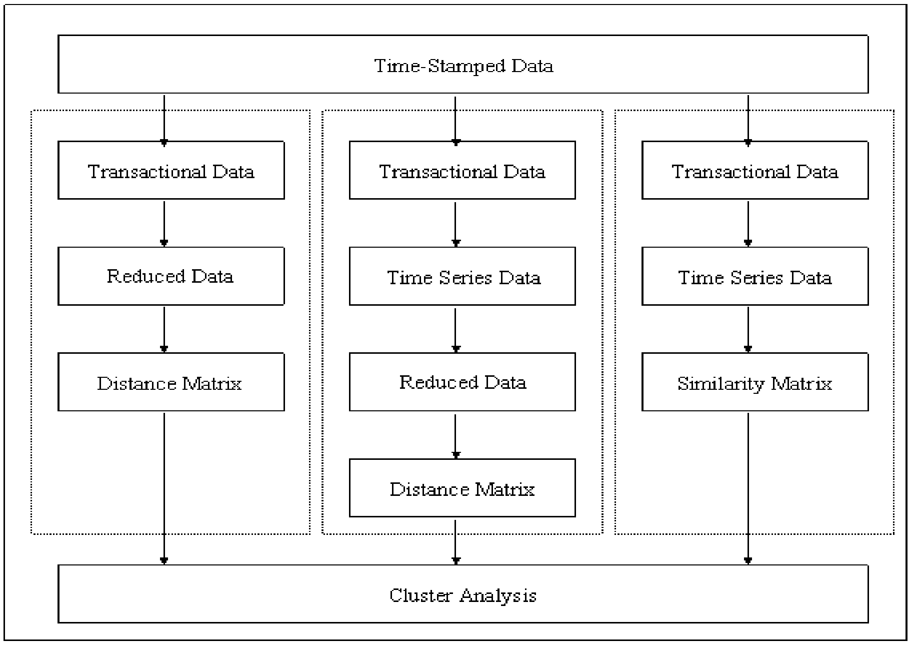
For series (longitudinal) data, a random sample should be selected from the N series; that is, N_{SAMPLE} random numbers would be selected between one and N without replacement. The sample data would then be created from the series whose series index corresponds to one of the N_{SAMPLE} randomly selected integers. For series data, longitudinal sampling is more appropriate; and the sample data dimensions are $(N_{SAMPLE} \times T)$ with $Y_{SAMPLE} \subseteq \{Y_1, ..., Y_N\}$. For multivariate series analysis, all of the covariate series must be randomly selected jointly.

CLUSTER ANALYSIS

The purpose of cluster analysis is to place objects into groups or clusters suggested by the data, not defined a priori, such that objects in a given cluster tend to be similar to each other in some sense, and objects in different clusters tend to be dissimilar. The "Introduction to Clustering Procedures" chapter of the *SAS/STAT User's Guide* describes a variety of clustering algorithms. Depending on the algorithm, coordinate data, distance data, or a correlation or covariance matrix can be used in the cluster analysis.

Given the reduced (coordinate) data, clustering analysis groups the rows of the reduced data. Since each reduced matrix row, R_i, uniquely maps to one series, Y_i, the series are indirectly clustered. Likewise, given a distance matrix, clustering analysis groups the rows of the distance matrix. Since each distance matrix row, D_i, uniquely maps to one series, Y_i, the series are indirectly clustered. Since time series data may vary in dimension and/or the series length may be large, cluster analysis is usually not applied directly to time series data.

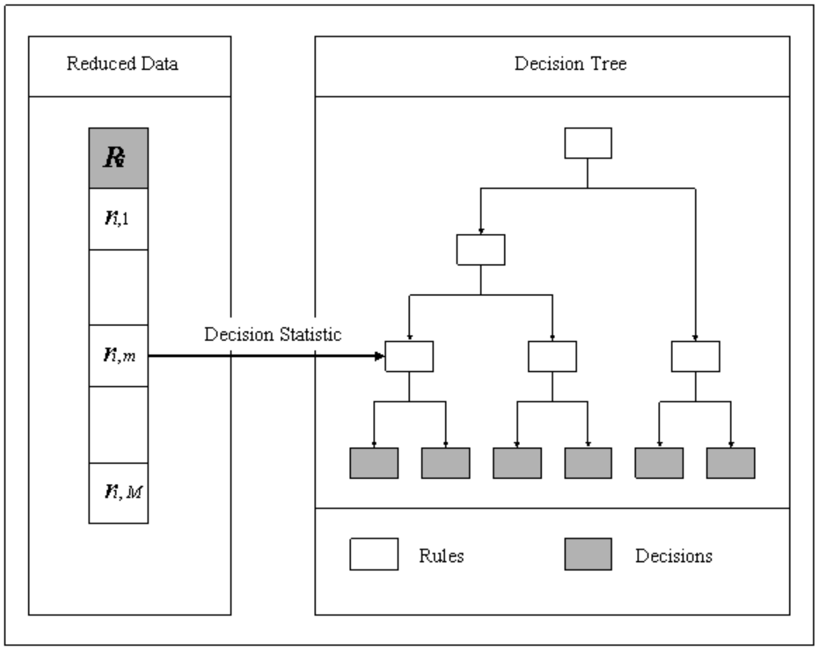
The diagram below illustrates the data flow associated with cluster analysis.



DECISION TREE ANALYSIS

Decision (classification) trees use decision rules in a hierarchical fashion to make a decision or classification. A node in the decision tree represents a decision rule. A leaf in the decision tree represents one possible decision. Each decision node requires a node statistic for which the decision is based. The SAS Enterprise Miner software documentation describes decision trees more fully.

The diagram below illustrates how the reduced data can provide one or more node statistics used in the decision tree.



LARGE-SCALE VISUALIZATION

Typically, transactional and time series are plotted using a two-dimensional plot, with time being recorded on the horizontal axis and the series values being recorded on the vertical axis.

Figure 30 and Figure 31 illustrate commonly used time plots. The figure on the left plots 67 monthly observations and the figure on the right plots 40,000 hourly observations. Notice the loss of detail in the hourly plot.

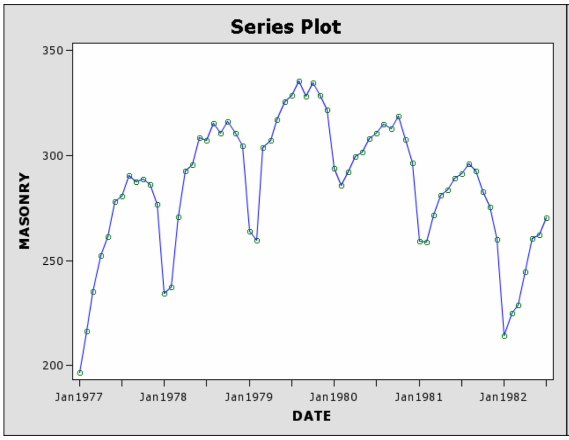


Figure 30: Monthly Observations

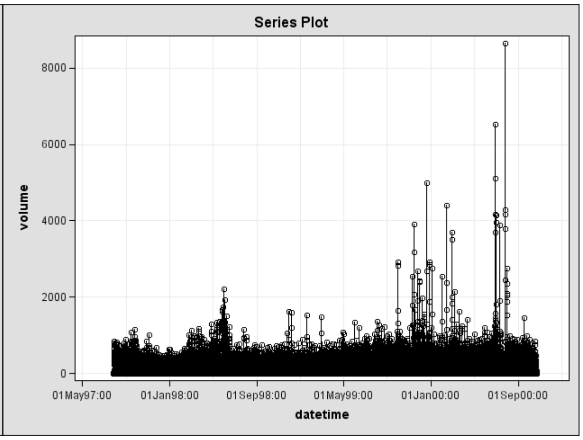


Figure 31: Hourly Observations

A large transaction and/or time series database can contain many such series. Therefore, the analyst must view many graphs in order to gain insight into the underlying processes. Additionally, each series may be very long due to high frequency or long history, making the commonly used series plot ineffective due to loss of detail. Large-scale visualization techniques are needed to overcome these difficulties.

Large-scale visualization aids the analyst in knowledge discovery, model identification, model validation, and visual outlier detection. Examples of transaction and/or time series databases that are difficult to visualize by more simplistic graphical techniques are

- databases that contain several (say less than 50) moderately sized series (e.g., sales by product line, sales by category)
- databases that contain numerous (say more than 50) moderately sized series (e.g., sales data by SKU and location)
- databases that contain numerous (say more than 50) short series (e.g., new products or products with short lifecycles)
- databases that contain relatively high frequency and/or very long series (e.g., utility load and call center data)

The following discussion summarizes some techniques for large-scale visualization of series data. In these discussions, the terms *reduced time series, reduction sequence,* or *reduced form* refer to the information obtained from the time series using the reduction techniques described above (e.g., transactional, time domain, frequency domain, or similarity analysis).

VECTOR SERIES PLOTS

Several moderately sized series can be simultaneously plotted and viewed using vector series plots. In a vector series plot, each series is plotted with a common time axis or by observation index. The series values are represented in the usual way for a two-dimensional plot. Each series can be optionally scaled and/or stacked. Scaling preserves the profile of each series but sacrifices the magnitude. Stacking allows for the convenient comparison on numerous scaled series by sacrificing detail. Vector series plots are particularly useful in knowledge discovery and evaluating the effectiveness of series clustering and/or classification techniques.

Figure 32 and Figure 33 illustrate examples of vector series plots. The vector series plot on the left is randomly plotted. The vector series plot on the right is based on clustering.

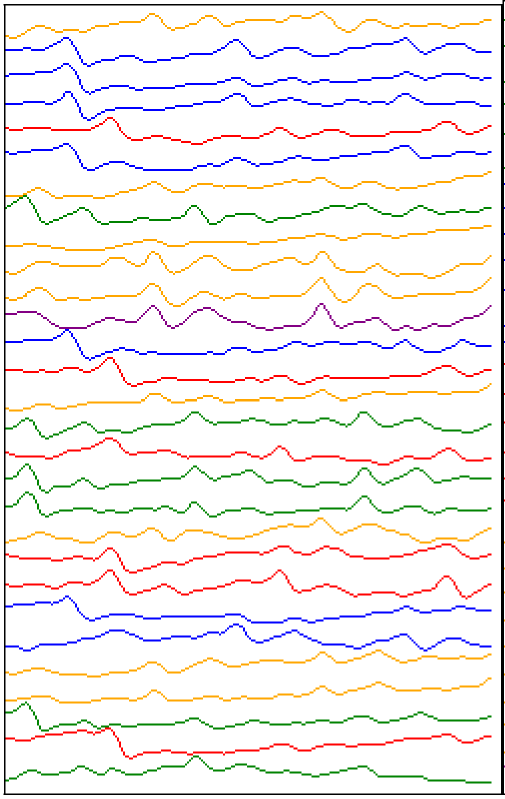


Figure 32: Unclustered Vector Series Plot

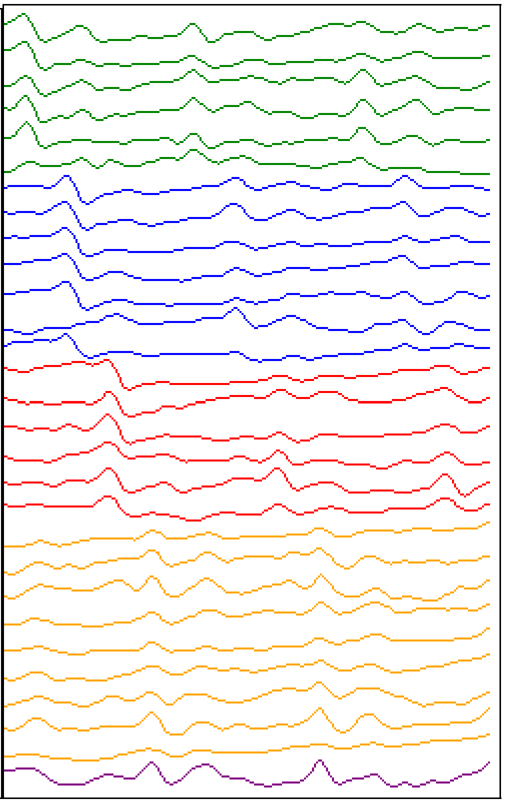


Figure 33: Clustered Vector Series Plot

THUMBNAIL PLOTS

Numerous moderately sized series can be separately plotted using thumbnail plots. In a thumbnail plot, each series is separately plotted with little or no detail; only the series profile can be discerned from these plots. The analyst can simultaneously view several of these plots by scrolling through the browser, making these plots more scalable than a vector series plot. Additionally, these thumbnail plots can be viewed in parallel with their corresponding reduced sequence plot. Thumbnail plots are particularly useful in knowledge discovery, evaluating the effectiveness of series clustering and/or classification techniques, mode residual autocorrelation analysis, and visual outlier detection.

Figure 34 illustrates examples of thumbnail series plots. The time series is plotted on the left and its seasonal component is plotted on the right.

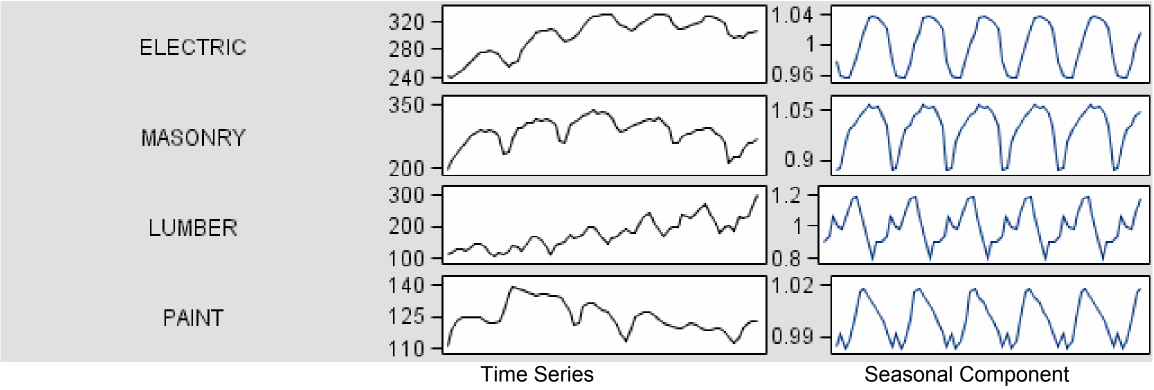


Figure 34: Thumbnail Time Series and Seasonal Component Plots

Figure 35 illustrates examples of thumbnail residual plots. The residual series is plotted on the left and its autocorrelation is plotted on the right.

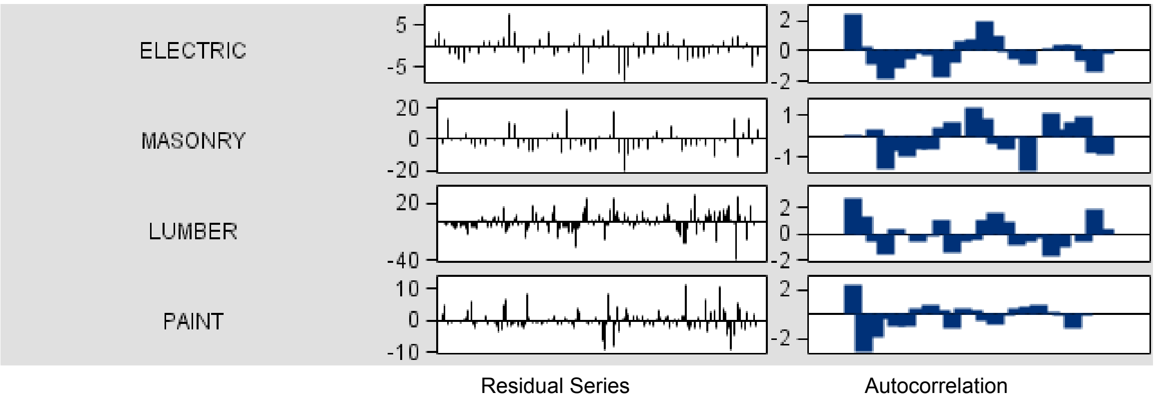
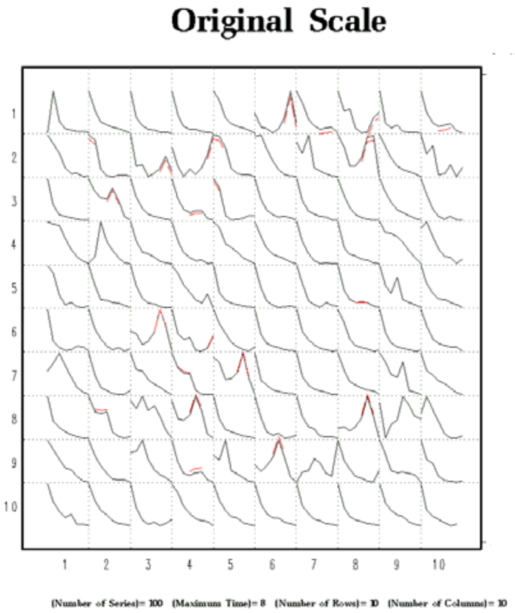


Figure 35: Thumbnail Residual Series and Autocorrelation Plots

LONGITUDINAL PLOTS

Numerous short series can be separately plotted and viewed simultaneously using longitudinal plots. In a longitudinal plot, each series is scaled and separately plotted with little or no detail; only the series profile can be discerned from these plots. However, each series plot is laid out in a matrix enabling the analyst to view many of these plots simultaneously. Additionally, these longitudinal plots can be viewed with calendar events for detecting calendar effects. Longitudinal plots are particularly useful in knowledge discovery and model identification associated with growth curves, diffusion models, and short life-cycle models.

Figure 36 and Figure 37 illustrate examples of longitudinal plots for 100 series of length 8. The original data is plotted on the left, and the log-transformed data is plotted on the right. Holidays are indicated in red (this may not be obvious in black and white). Notice that the log transformed data is close to linear.



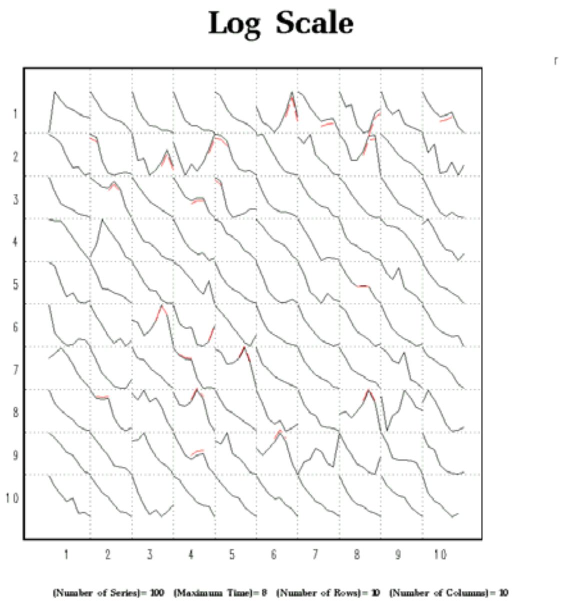


Figure 36: Longitudinal Plot

Figure 37: Log Transformed Longitudinal Plot

HEAT MAPS

Very large numbers of transactional or time series data can be viewed simultaneously in reduced form using heat maps. In a heat map, a matrix of colored squares is rendered with each square representing the relative magnitude of a single reduced sequence value. The squares can be as small as a single pixel, very large numbers of reduced sequences to be viewed simultaneously. For each series, the reduced sequence is plotted in a single column (or row) of the heat map matrix. A single row (or column) represents a single reduced sequence value by sequence index across all series.

Additionally, these heat maps can be viewed in parallel with a dendrogram (phenogram or tree diagram). A dendrogram displays the results of a cluster analysis in a hierarchical tree. The level of the tree represents the distances between the clusters based on a particular distance measure or statistic (e.g., Euclidean). When used with series clustering, the dendrogram divides a set of series into hierarchical clusters using the reduced sequences based on the desired distance measures.

Heat maps are particularly useful in knowledge discovery. When combined with a dendrogram, heat maps are especially useful in interpreting the results of cluster analysis.

Figure 38 illustrates an example of a heat map in parallel with a dendrogram.

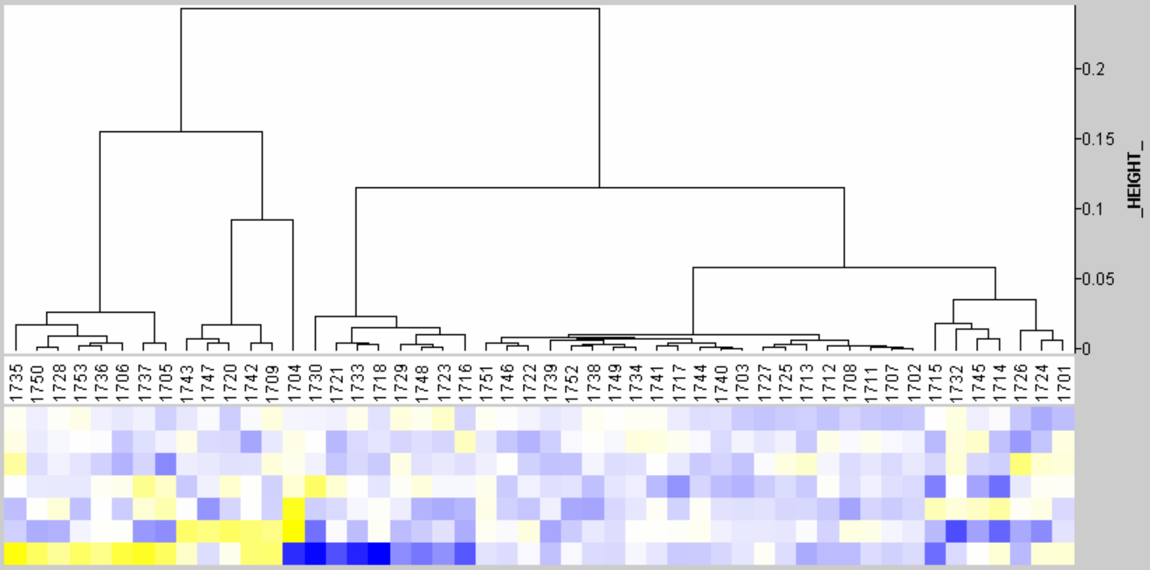


Figure 38: Heat Map and Dendrogram in Parallel

TIME SPIRALS PLOTS

For high-frequency series, plotting a series on a typical two-dimensional time plot can result in a significant loss of detail. Time spirals avoid this loss of detail by graphing the series on a spiral. In a time spiral, time typically progresses from the center of the spiral outward. Each turn of the spiral represents one cycle of time (e.g., 24 hours, one week). The series values are typically represented on the spiral by color for continuous responses and by symbols for discrete responses. Dependent (target) and independent (input) variables can be graphed simultaneously by using concentric spirals.

Figure 39, Figure 40, Figure 41, and Figure 42 illustrate examples of time spiral plots.

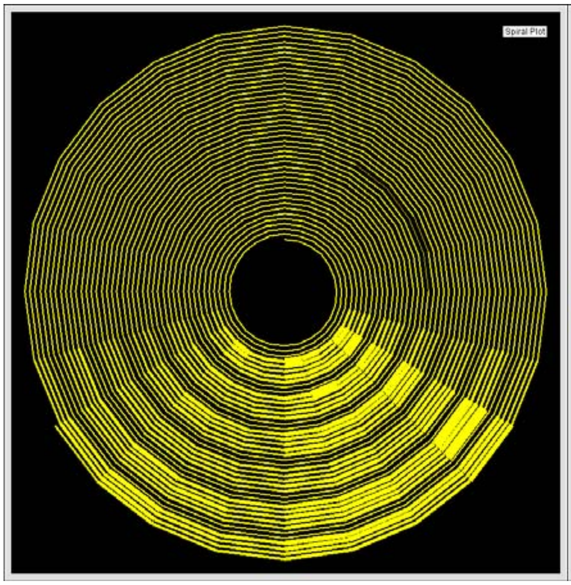


Figure 39: Time Spiral Plot of Dense Data

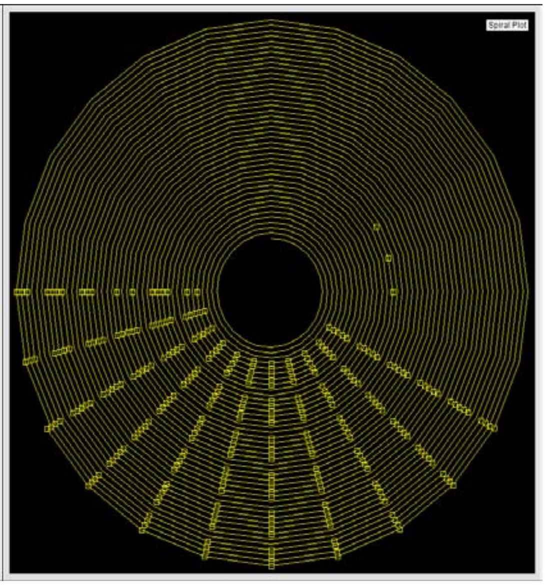


Figure 40: Time Spiral Plot of Sparse Data

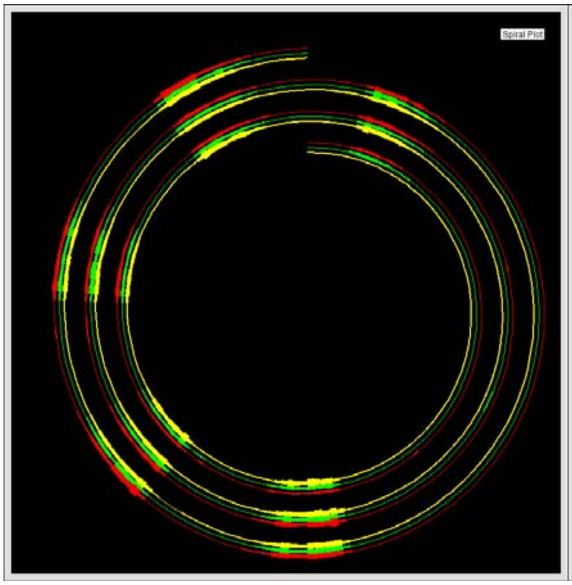


Figure 41: Concentric Spiral Plot of Dense Data

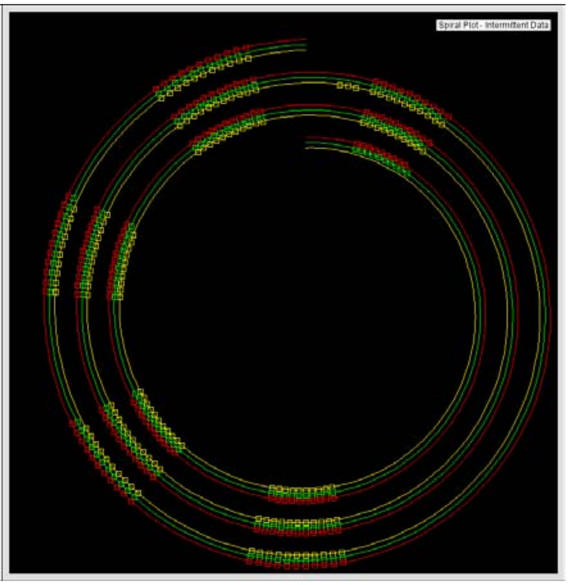


Figure 42: Concentric Spiral Plot of Sparse Data

INTERACTIVE GRAPHICS

For a large number of series, interactive graphical analysis is useful for knowledge discovery.

Figure 43 displays the seasonal decomposition of numerous time series. Each panel plots a particular component for each of the time series. After selecting one time series, each of the components of the selected series is highlighted.

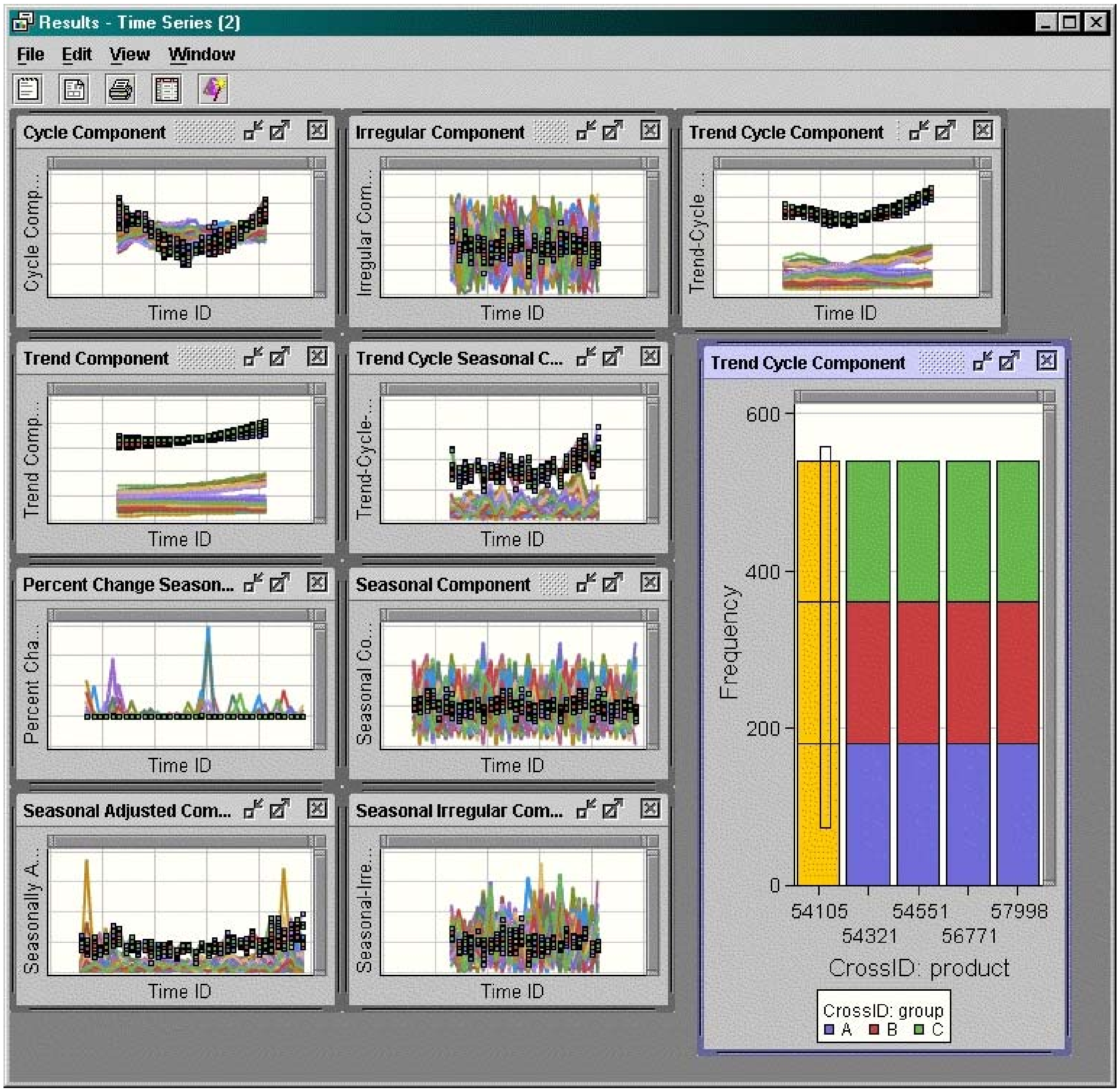


Figure 43: Interactive Graphics

CONCLUSION

Mining transactional and time series data can help business leaders make better decisions by listening to their suppliers or customers via their transactions collected over time. A single set of transactions or a single time series can contain large amounts of information. A business can have many suppliers and/or customers. The volume of information associated with each of the suppliers and customers makes data mining tasks difficult. The approach outlined above relies on reducing transactional data into its trend and seasonal properties, and time series data using time series analysis, seasonal decomposition, time series models, and automatic time series model selection. This information reduction may help make traditional data mining tasks such as cluster analysis and decision trees more tractable.

REFERENCES

Barry, M. J. and Linoff, G. S. (1997), *Data Mining Techniques: For Marketing, Sales, and Customer Support*, New York: John Wiley & Sons, Inc.

Box, G. E. P, Jenkins, G. M., and Reinsel, G. C. (1994), *Time Series Analysis: Forecasting and Control*, Englewood Cliffs, NJ: Prentice Hall, Inc.

Brockwell, P. J. and Davis, R. A. (1996), *Introduction to Time Series and Forecasting,* New York: Springer-Verlag.

Chatfield, C. (2000), *Time Series Models,* Boca Raton, FL: Chapman & Hall/CRC.

Fuller, W. A. (1995), *Introduction to Statistical Time Series*, New York: John Wiley & Sons, Inc.

Hamilton, J. D. (1994), *Time Series Analysis*, Princeton, NJ: Princeton University Press.

Han, J. and Kamber, M. (2001), *Data Mining: Concepts and Techniques,* San Francisco: Morgan Kaufmann Publishers.

Harvey, A. C. (1994), *Time Series Models*, Cambridge, MA: MIT Press.

Leonard, M. J. (2002), *Large Scale Automatic Forecasting: Millions of Forecasts,* International Symposium of Forecasting.

Makridakis, S. G., Wheelwright, S. C., and Hyndman, R. J. (1997), *Forecasting: Methods and Applications*, New York: John Wiley & Sons, Inc.

Pyle, D. (1999), *Data Preparation for Data Mining*, San Francisco: Morgan Kaufman Publishers.

Sankoff, D. and Kruskal, J. B. (1983), *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison,* Stanford, CA: CSLI Publications.

Weber, M., Alexa, M., and Mueller, W. (2001), “Visualizing Time-Series on Spirals,” *Proceedings of the IEEE InfoVis Symposium,* Los Alamitos, CA: IEEE Press.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Michael Leonard
SAS Institute
SAS Campus Drive
Cary, NC 27513

(919) 531-6967
michael.leonard@sas.com
www.sas.com