Analyzing Business and Economic Data with SAS/ETS® Software

Mark Little, SAS Institute Inc., Cary NC

ABSTRACT

This paper discusses analyzing business and economic data with SAS/ETS software. The paper surveys some of the different kinds of data analysis problems found with business and economic data, and the need for special data analysis methods for these kinds of problems. The paper discusses the tools SAS/ETS software provides to meet these needs.

INTRODUCTION

What are some of the data analysis needs involving business and economic data? Here are the topics this paper addresses:

1. Need for forecasting the future, or making predictions from past trends. This one seems obvious; it's hard to make plans unless you have some idea what the future will be like.

2. Need for determining the relation between variables when effects take place over time. This need applies to most policy analysis problems and to questions like "How effective is our advertising program?", or "What effect do changes in speed limits have on traffic fatalities?"

3. Need to determine simultaneous relationships. This need applies to any analysis involving supply and demand interactions.

4. Need to account for seasonal regularities and the need for seasonal adjustment. How can you track the trend from month to month when the data normally varies with the time of year?

5. Need to work with data of different periodicities. Some of your data are monthly, some are quarterly, and some are available only annually. What to do?

6. Need to access financial and economic data bases. Vast quantities of economic and financial data are available from many different data vendors. However, most data suppliers deliver data in their own, often complex formats. If you purchase an economic or financial data base, how can you read the data into SAS without a lot of programming effort?

7. Need for time value of money calculations to compare financing alternatives. You need to borrow money and many kinds of loans are available. How do you decide which is the best deal?

8. Need for financial reports and spread sheet calculations. Balance sheets, income statements, and other accounting reports need to be generated.

9. Need to model complex dynamic systems. The most sophisticated approach to forecasting or predicting the effect of policy changes is to build a full scale simulation model of the system. Corporate financial planning models and macro-econometric models of the economy are two examples of simulation models.

The common feature of all these data analysis needs is the role of TIME. Business and economic data are normally time series data.

The data may be for a succession of months or quarters or years or days. Or the data may be financial measures for particular accounting periods. Or the data may concern the present value of a security. Time is a key feature of all such data.

Ordinary data analysis methods do not take time into account, and are often not appropriate for analyzing business and economic data. Traditional statistical methods assume that observations are independent. But the future depends on the past, and so the independence assumptions of traditional statistical methods are almost never true of business and economic data. To analyze business and economic data, we need analytical techniques that take into account time dependencies.

FORECASTING

For example, consider the problem of using regression analysis to fit and then forecast a time trend. Linear regression is a very common data analysis method, and extrapolating a trend line is a simple and commonly used forecasting technique.

The following graph shows an example of a time series, and the ordinary regression time trend line fit to the series. Notice that when the series is above (or below) the trend line, it tends to remain above (below) the trend for several periods. This pattern is an example of autocorrelation. Autocorrelation is a very common pattern seen in business and economic data.

Ordinary regression analysis is based on several statistical assumptions. One key assumption is that the residuals are uncorrelated. However, with time series data, the residuals usually are correlated over time. So if you use ordinary regression for time series data, the assumptions on which the regression method is based will usually be violated.

This has three important consequences. First, statistical tests of the significance of the parameters and the confidence limits for the predicted values are not correct. Second, the estimate of the trend line is not as accurate as it could be if the autocorrelation were taken into account. Third, the residuals contain information that could be used to improve the forecasts.

The solution to this problem is to augment the regression model with an autoregressive model for the random error and thus account for the autocorrelation of the residuals. The AUTOREG procedure is for regression when the errors are autocorrelated.
Here is an example of using the AUTOREG procedure to forecast the time series shown in the previous graph.

```
proc autoreg;
  model y = time / nlag=2;
  output out=pred predict=yhat;
run;
```

Here is the PROC AUTOREG output showing the parameter estimates. Notice that the regression results include coefficients for the lagged residuals, labeled AR(1) and AR(2), in addition to the intercept and slope coefficients.

```
Variable      DF    B Value    Std Error    t Ratio    Approx Prob
Intercept      1    3.77492645  0.773468  4.909 0.0001
TIME           1    0.61014022  0.06115  9.961 0.0002
AR(1)          1    -0.02313670  0.231159 -0.979 0.1340
AR(2)          1    -0.09072334  0.232306 1.549 0.1397
```

Here is a plot of the PROC AUTOREG predicted values. Note that the AUTOREG procedure not only produces the trend forecast; it outputs predicted values that reflect the pattern of recent departures from trend, and thus tracks the data much more closely than the simple regression line does.

The FORECAST procedure can also produce forecasts using this time trend with autoregressive errors model. In addition, the

```
proc forecast out=pred outfull lead=4 interval=month;
  id date;
  var y;
run;
```

This idea of using the the pattern of correlations of a variable with its past can be generalized. The autoregressive integrated moving average, or ARIMA, model predicts future values of a variable from its past values and from past errors, and possibly from values of other variables. ARIMA models are also often called Box-Jenkins models.

The ARIMA procedure provides complete support for ARIMA modeling and forecasting. PROC ARIMA is designed to support the identification, estimation and diagnostic checking, and forecasting stages of model development recommended by Box and Jenkins.

Here is an example of using PROC ARIMA.

```
proc arima;
  identify var=y(1);
  estimate p=2;
  forecast out=pred lead=4
    id=date interval=month;
run;
```

Here is the part of the PROC ARIMA output showing the parameter estimates.

```
Proc ARIMA Results for ARIMA(1,1,0) Model

ARIMA Procedure

```

The FORECAST procedure can also produce forecasts using this time trend with autoregressive errors model. In addition, the
Here is the PROC ARIMA forecast of the example time series.

![PROC ARIMA Prediction](chart)

ARIMA or Box-Jenkins models are widely used by more knowledgeable forecasters and time series data analysts because this class of models is very powerful, and incorporates many simpler forecasting models as special cases. For example, the exponential smoothing model can be fit as an ARIMA model. The autoregressive models used by PROC AUTOREG or PROC FORECAST can also be fit by PROC ARIMA.

An even more general kind of time series model is the statespace model. Statespace models are used to jointly forecast several time series that interact with each other. Statespace models include multivariate ARIMA models as a special case.

The STATESPACE procedure is used to fit and forecast statespace models. By default, PROC STATESPACE automatically selects and fits the best model for the time series.

Here is an example of using PROC STATESPACE.

```sas
proc statespace out=pred lead=4
   id=date interval=month;
var y1(1) y2(1);
run;
```

Here is part of the PROC STATESPACE output, showing the parameter estimates for the bivariate statespace model that the procedure selected for the two series.

<table>
<thead>
<tr>
<th>Parameter Estimates</th>
<th>Std. Err.</th>
<th>t Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>F(1,1)</td>
<td>0.31688</td>
<td>0.09255</td>
</tr>
<tr>
<td>F(1,2)</td>
<td>-0.88863</td>
<td>0.11331</td>
</tr>
<tr>
<td>F(2,1)</td>
<td>0.64057</td>
<td>0.07358</td>
</tr>
<tr>
<td>F(3,2)</td>
<td>-0.63129</td>
<td>0.22354</td>
</tr>
</tbody>
</table>

In addition to SAS procedures, SAS/ETS software also includes a Forecast menu system, developed with SAS/AF software. This menu system uses PROC ARIMA and PROC FORECAST to do time series forecasts. It makes it easy to do time series forecasting without typing procedure statements, and provides many features for graphical data exploration and graphical comparisons of forecasting models and forecasts.

The Forecast menu system makes it easy to use all kinds of ARIMA models and intervention models for forecasting. The Forecast menu system also has a feature that automatically selects the best forecasting model for a series.

![Forecast Menu System](chart)

TIME SERIES REGRESSION AND TIME SERIES ANALYSIS

We’ve discussed some time series analysis methods as tools for forecasting. The other major use for time series tools is to estimate and test relationships between variables that evolve over time.

Ordinary regression analysis usually breaks down when time enters the picture, as explained for the example of using time trend regression for forecasting.

When the purpose is not forecasting but analyzing relationships, the statistical problems caused by autocorrelated residuals are of even greater concern. When using regression models, PROC AUTOREG or a similar tool is needed whenever autocorrelation is an issue, which is often the case when analyzing business and economic data.

PROC ARIMA can also be used to analyze relationships between time series variables. You can specify independent variables as inputs to an ARIMA model for the dependent variable. When independent variables are used in ARIMA model, the models are sometimes called ARIMAX models. Using PROC ARIMA with input series, you can estimate relationships and produce forecasts that make use of both the information in past values of the series and the information contained in independent variables.

PROC ARIMA supports ARIMA models with input variables and general transfer functions. When input variables are used in the simplest form, the result is a regression model with ARMA errors. Complex transfer functions can be used to produce various kinds of distributed lag regression models.

One special kind of ARIMA model with inputs is an intervention model. Often events take place that affect the pattern of a time series. For example, an increase in tax rates can make the sales of a product drop, or an advertising campaign might increase sales. An intervention variable has values that flag the time before and after an event.

The effect of an intervention can be hard to see, since the effect may take time to be realized, and may be obscured by the other patterns in the series. The solution is to incorporate the intervention
variable as an input to an ARIMA model for the series, and thus analyze jointly the effects of past values, past errors, and the intervention.

Here is an example of using PROC ARIMA with an intervention variable and an independent input to the model.

```plaintext
proc arima;
identify var=y crosscorr=tax;
estimate p=1 q=1 input=tax;
run;
```

Here is the PROC ARIMA output showing the estimated effect of the tax rate change intervention:

```
ARIMA Procedure
Maximum Likelihood Estimation
Parameter Estimate Std Error t Ratio Lag Variable Shift
AR(1) 0.056203 0.17704 0.314 0 SALES 0
AR(2) -0.40778 0.24864 -1.64 0 TAX 0
Constant 2.64321002
Variance Estimate = 0.25367794
Std Error Estimate = 0.50366451
AIC = 52.72492
SBC = 55.87045
Number of Residuals = 21
```

Here is a plot of the PROC ARIMA predictions for the TAX intervention model.

When the effect of an independent variable occurs gradually, the relationship is often represented with a distributed lag model. Various kinds of distributed lag relationships can be fit with transfer function models by PROC ARIMA. Another kind of distributed lag model is the polynomial distributed lag, or POL, model.

The PDLREG procedure provides most of the features of PROC AUTOREG, and also allows some independent variables to enter the model as polynomial distributed lag processes.

Here is an example of using PROC PDLREG to regress Y on the current and past 5 values of X, with the lag distribution constrained to a cubic polynomial.

```plaintext
proc pdlreg;
model y = x(5,3);
run;
```

Here is the PDLREG output showing the estimates of the parameters of the cubic polynomial lag distribution curve:

```
PDLREG Procedure
Dependent Variable = Y
```

```
Ordinary Least Squares Estimates
```

```
<table>
<thead>
<tr>
<th>Variable</th>
<th>DF</th>
<th>Value</th>
<th>Std Error</th>
<th>t Ratio</th>
<th>Approx Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1</td>
<td>5.32791078</td>
<td>0.89886</td>
<td>57.925</td>
<td>0.0001</td>
</tr>
<tr>
<td>x(1)</td>
<td>1</td>
<td>0.16033012</td>
<td>0.01313</td>
<td>12.969</td>
<td>0.0001</td>
</tr>
<tr>
<td>x(2)</td>
<td>1</td>
<td>0.36521095</td>
<td>0.13566</td>
<td>2.692</td>
<td>0.0089</td>
</tr>
<tr>
<td>x(3)</td>
<td>1</td>
<td>-0.00112645</td>
<td>0.19138</td>
<td>-0.006</td>
<td>0.995</td>
</tr>
<tr>
<td>x(4)</td>
<td>1</td>
<td>0.18009</td>
<td>0.12112</td>
<td>1.49</td>
<td>0.1453</td>
</tr>
<tr>
<td>x(5)</td>
<td>1</td>
<td>0.39080</td>
<td>0.097</td>
<td>4.01</td>
<td>0.0001</td>
</tr>
<tr>
<td>x(6)</td>
<td>1</td>
<td>0.48319</td>
<td>0.080</td>
<td>6.03</td>
<td>0.0001</td>
</tr>
<tr>
<td>x(7)</td>
<td>1</td>
<td>0.45645</td>
<td>0.080</td>
<td>5.67</td>
<td>0.0001</td>
</tr>
<tr>
<td>x(8)</td>
<td>1</td>
<td>0.30912</td>
<td>0.098</td>
<td>3.17</td>
<td>0.0089</td>
</tr>
<tr>
<td>x(9)</td>
<td>1</td>
<td>0.04172</td>
<td>0.125</td>
<td>0.34</td>
<td>0.7382</td>
</tr>
</tbody>
</table>

```

Here is a plot of the PROC PDLREG predictions for the TAX intervention model.

Sometimes you have time series data that is replicated across geographic or other cross sectional units. An example is sales of a product over time for each of several sales territories. The production cost of a product over time at each of several factories is another example.

With this kind of data, there are often correlations over time for each unit, correlations between units, and differences between units. The TSCSREG procedure provides statistical methods for analyzing regression models for time series-cross sectional data of this sort.
Here is an example of using the TSCSREG procedure to regress Y on X, with yearly data series replicated by state.

```plaintext
proc tscsreg;
  id state year;
  model y = x;
run;
```

Here is what the TSCSREG output looks like.

---

**TSCSREG Procedure**<br />Preliminary Method Estimation<br />Dependent Variable: Y<br />Model Description<br />Estimation Method: PFEFFER<br />Number of Cross Sections: 4<br />Time Series Length: 26<br />

Variance Component Estimates<br />
<table>
<thead>
<tr>
<th>Source</th>
<th>MSE</th>
<th>DF</th>
<th>Root MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSE</td>
<td>18.3211</td>
<td>102.53</td>
<td>0.816273</td>
</tr>
<tr>
<td>MSE</td>
<td>0.161854</td>
<td>0.816273</td>
<td></td>
</tr>
<tr>
<td>Variance Component for Cross Sections</td>
<td>4.342931</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance Component for Time Series</td>
<td>0.007322</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variance Component for Error</td>
<td>0.765655</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameter Estimates<br />
| Variable | DF | Parameter Estimate | Standard Error | t for H0: Parameter = 0 | Prob > |t|<br />
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>1</td>
<td>2.710624</td>
<td>98.190323</td>
<td>0.129</td>
<td>0.8984</td>
</tr>
<tr>
<td>PRICE</td>
<td>1</td>
<td>4.643002</td>
<td>13.353797</td>
<td>0.332</td>
<td>0.6094</td>
</tr>
<tr>
<td>COST</td>
<td>1</td>
<td>-1.975592</td>
<td>0.066743</td>
<td>-29.600</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

---

**SYSTEMS REGRESSION**

We've discussed some of the data analysis problems that arise when the past affects the present and the future. Problems also arise when the present affects the present in a reciprocal or simultaneous way.

Economic data often involve simultaneous relationships. For example, market price and quantity sold are determined by the interaction of supply and demand, by not by either supply or demand separately.

Ordinary statistical methods assume that independent variables affect dependent variables, but dependent variables do not affect the independent variables. That's why we call them independent variables. If this assumption is not true, then ordinary statistical methods don't work right.

For example, if you use ordinary regression analysis to estimate the demand function for a product, the results will be wrong. Price and quantity are simultaneously determined. Neither is an independent variable. The answer is to use special methods of regression analysis that are designed to allow dependent regressors.

The SYSLIN procedure fits and analyzes simultaneous systems of regression models and does regression analysis for models with dependent regressors. PROC SYSLIN supports a variety of methods designed for this simultaneous equation regression problem, such as two- and three-stage least squares, and full and limited information maximum likelihood.

The dependent regressor or simultaneous equation problem also influences the computation of predicted values or forecasts. The SIMLIN procedure compliments the SYSLIN procedure, and is used to forecast or simulate jointly dependent variables.

PROC SYSLIN takes the estimated regression equations produced by PROC SYSLIN and computes predicted values that solve the simultaneous system.

Here is an example of using PROC SYSLIN.

```plaintext
proc syslin fiml outest=est;
endogenous price quantity;
supply:
  model quantity price cost;
demand:
  model quantity price income;
run;
```

Here is part of the PROC SYSLIN output showing the reduced form coefficients.

---

**SIMLIN Procedure**

Inverse Coefficient Matrix for Endogenous Variables

<table>
<thead>
<tr>
<th>Price</th>
<th>Quantity</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.4904</td>
<td>0.8046</td>
<td></td>
</tr>
</tbody>
</table>

Reduced Form for Exogenous Variables

<table>
<thead>
<tr>
<th>Cost</th>
<th>Income</th>
<th>Intercept</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.1734</td>
<td>1.2704</td>
<td>3.2734</td>
</tr>
</tbody>
</table>

PROC SYSLIN also performs multiplier analysis, which determines the sensitivity of the dependent variables to changes in the inde-
pendent variables, taking into account the simultaneous interrelationships between the dependent variables, and also taking into account dynamic or lagged relationships between the dependent variables.

SEASONAL ADJUSTMENT

Business and economic data often have a regular seasonal pattern. A variable may be consistently higher or lower at particular times of the year. Daily series may be higher on particular days of the week, and hourly data higher at particular times of the day. All such periodically recurring patterns are called seasonality.

Seasonality presents two kinds of data analysis problems. First, the seasonal pattern must be taken into account when modeling or forecasting the variable. There are several approaches to doing this.

PROC FORECAST provides the Winters and additive Winters methods for forecasting seasonal series. These methods are like exponential smoothing, but incorporate exponentially smoothed estimates of seasonal factors.

For regression models, seasonal dummy variables can be used. For example, indicator variables for January, February, March, etc., can be added as regressors when fitting a regression model to a monthly series with PROC AUTOREG.

High order autoregressive error models can also be used to capture seasonal differences. A more advanced method is to use seasonal differencing or seasonal autoregressive or moving average operators in an ARIMA model. PROC ARIMA has features for seasonal ARIMA models.

The second data analysis problem posed by seasonal data is the need for seasonal adjustment. In this case, you don't want to model or forecast the series, you just want to see the trend in the data with the seasonal pattern removed.

It can be very hard to perceive the trend from month to month or quarter to quarter when the data have a strong seasonal pattern. This is why official statistics on many economic variables like GDP, unemployment, and retail sales are reported on a seasonally adjusted basis.

What does it mean when the U.S. government reports that retail sales were so many billion dollars, "seasonally adjusted"? It means that the actual retail sales series was transformed by an algorithm called the Census Xll method, which uses various moving average operations to filter out the seasonal pattern.

The X11 procedure performs seasonal adjustment using the official X11 algorithm developed by the U.S. Census Bureau. PROC X11 decomposes a series into trend, seasonal, and irregular components. PROC X11 can output seasonally adjusted series, seasonal factors, and many series representing many intermediate calculations involved in the X11 algorithm.

Here is an example of using PROC X11 to seasonally adjust a series SALES.

Here is part of the PROC X11 printed output.

The following graph shows a seasonal time series and the corresponding seasonally adjusted series produced by PROC X11.

An improved version of the X11 method, call the X11-ARIMA method, was developed by Statistics Canada, a Canadian government agency. PROC X11 also supports the X11-ARIMA seasonal adjustment method.

TIME SERIES INTERPOLATION

One of the problems caused by the time series nature of business and economic data is that different series are measured at different frequencies. You may want to model the relation between two variables, but one variables is available as a monthly series while the other is only available as a quarterly series. Or a variable may be available as a quarterly series, but you need the variable as an annual series, or vice versa.

The EXPAND procedure converts time series from one sampling frequency to another. PROC EXPAND fits cubic spline curves to the data, and then interpolates new values from the curve. For ex-

```plaintext
proc x11;
  monthly date=date additive;
  var sales;
  output out=adj dll=adjsales;
  tables dll;
run;
```
ample, you can interpolate monthly series from quarterly values, or aggregate monthly values to form a quarterly series.

PROC EXPAND can also be used to interpolate missing values in time series without changing the series frequency.

Here is an example of using PROC EXPAND to interpolate from quarterly to monthly:

```sas
proc expand data=qtr1y out=monthly
  from=qtr to=month;
  id date;
  convert Xi
run;
```

Another issue of concern when analyzing time series data, is what part of the time periods do the values represent. For example, if we have a monthly series, do the numbers represent the value as of the end of each month, or at the beginning or middle of the month? Or do they represent a total or average value over the whole month?

PROC EXPAND allows you to specify whether series measure beginning, middle, or end of period values, or period totals or averages. PROC EXPAND takes this information into account when interpolating values of the series. Moreover, PROC EXPAND allows you to convert between these cases. For example, you can interpolate average values for each month from end-of-month input values, or vise versa.

Here is an example of using PROC EXPAND to interpolate end of period estimates from monthly average values:

```sas
proc expand out=out from=month;
  id date;
  convert x / observed=(average,end);
run;
```

ACCESSING ECONOMIC AND FINANCIAL DATA BASES

One obvious problem in data analysis is the mechanics of just collecting the data in a usable form. In the case of business and economic data, there are many vendors supplying data bases of many different economic and financial time series.

These include government agencies, such as the BEA and BLS in the United States, international agencies such as the International Monetary Fund, commercial data vendors such as Citicorp, Haver Analytics, and Compustat, and organizations like the CRSP, the Center for Research in Security Prices.

Making use of these data sources can be challenging. Each vendor has its own format for encoding the data. In some cases these formats are complex, and much programming may be required to extract the data.

The new DATASOURCE procedure makes it easy to extract time series data from files supplied by the vendors listed previously. PROC DATASOURCE reads vendor supplied data files and outputs the extracted data in SAS data sets. PROC DATASOURCE also provides features for selecting specific subsets from the data files, which are often very large.

Here is an example of using PROC DATASOURCE to read CITIBASE data:

```sas
proc datasource infile=citifile
  filetype=citibase
  interval=month
  out=citimon;
  range from 1980;
run;
```

The following data vendors are currently supported by PROC DATASOURCE:

- CITIBASE
- Haver Analytics
- Compustat
- BEA
- BLS
- IMF
- CRSP

LOAN ANALYSIS

One important way that time enters into the analysis of business and economic data is through the time value of money. The LOAN procedure analyzes installment loans, and has special features for mortgage loans.

PROC LOAN can analyze fixed rate, adjustable rate, buydown rate, and balloon payment loans. In addition, PROC LOAN can compare different loan contracts and report which is the most advantageous.

Here is an example of using PROC LOAN to compare financing $100,000 with either a fixed rate loan or an adjustable rate loan, assuming a worst case scenario for the rate adjustments:

```sas
proc loan;
  fixed amount=100000 rate=5.75
def=190;
  arm amount=100000 rate=5.75
def=180 caps=(1,5)
  compare;
run;
```
Here is the PROC LOAN analysis for the adjustable rate loan.

## LOAN Procedure
### Adjustable Rate Loan: Worst Case Analysis Summary

<table>
<thead>
<tr>
<th>Loan No. 2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dwonpayment:</td>
<td>0.00</td>
<td>Principal Amount:</td>
<td>100000.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initialization:</td>
<td>0.20</td>
<td>Points:</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Interest:</td>
<td>100299.97</td>
<td>Nominal Rate:</td>
<td>7.60%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Payments:</td>
<td>200939.97</td>
<td>Effective Rate:</td>
<td>7.60%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pay Interval:</td>
<td>Monthly</td>
<td>Compounding:</td>
<td>Monthly</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of payments:</td>
<td>180</td>
<td>No. of compoundings:</td>
<td>180</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

List of Rates and Payments for Loan No. 2

<table>
<thead>
<tr>
<th>Period</th>
<th>Nominal Rate</th>
<th>Effective Rate</th>
<th>Payment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.40%</td>
<td>7.60%</td>
<td>922.34</td>
</tr>
<tr>
<td>13</td>
<td>6.40%</td>
<td>6.73%</td>
<td>975.80</td>
</tr>
<tr>
<td>25</td>
<td>5.60%</td>
<td>5.82%</td>
<td>1025.75</td>
</tr>
<tr>
<td>49</td>
<td>4.60%</td>
<td>4.95%</td>
<td>1075.79</td>
</tr>
<tr>
<td>61</td>
<td>3.60%</td>
<td>3.91%</td>
<td>1125.73</td>
</tr>
</tbody>
</table>

Here is the comparison of the two loans, based on a worst case analysis of the adjustable loan.

## LOAN Procedure
### Loan Comparison Report

Analysis through Payent Number 180

<table>
<thead>
<tr>
<th>Loan Label</th>
<th>Outstanding Payment</th>
<th>Paid Rate</th>
<th>True Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loan No. 1</td>
<td>0.00</td>
<td>999.01</td>
<td>19900.56</td>
</tr>
<tr>
<td>Loan No. 2</td>
<td>0.00</td>
<td>1175.18</td>
<td>102639.87</td>
</tr>
</tbody>
</table>

Note: "Loan No. 1" is the best alternative based on true rate analysis through period 180.

## FINANCIAL REPORT WRITING

Business data gives rise to the need for calculations and reports showing financial condition over time. Examples are balance sheets and income statements displaying financial results for the current and recent accounting periods.

The COMPUTAB procedure is useful for calculations and report generation of this kind. PROC COMPUTAB combines the computational features of a programmable spread sheet with a report writer for tabular reports.

The SAS system includes the SAS/CALC product for spread sheet applications, and the REPORT procedure for report generation. These products are interactive and more powerful for their intended applications than the older COMPUTAB procedure. However, PROC COMPUTAB has the advantage of combining report writing and spread sheet-like computational features.

## MODELING COMPLEX DYNAMIC SYSTEMS

We've talked about forecasting and analyzing relationships between variables using linear regression and time series models, and we discussed simultaneous dependencies. These methods can be generalized further.

The most sophisticated approach to forecasting or to predicting the effect of policy changes is to build a full scale simulation model. Simulation models can be developed for any complex system. When business and economic data are involved, the model is often of a company or of a whole economic system. Simulation models of a company are often called financial planning models.

The MODEL procedure allows you to specify, estimate, and solve complex simulation models. Model equations can be nonlinear, and are specified using programming statements written in the SAS DATA step language. The system of nonlinear equations can be simultaneous, and can contain dynamic or lagged relationships. You can also include time series processes like ARMA errors in model equations.

PROC MODEL has features for regression analysis of nonlinear simultaneous equation systems. These systems regression methods, such as nonlinear two- and three-stage least squares, are used to estimate unknown parameters in the model equations.

Here is an example of using PROC MODEL for a simple nonlinear regression. The model equation is written using SAS programming statements. TheFIT statement performs the nonlinear regression analysis. The SOLVE statement solves the nonlinear equation to compute predicted values of X as a function of Y.

```sas
proc model;
  Y = a * exp( b * X );
  fit Y;
  solve x / out=pred;
run;
```

Here are the PROC MODEL estimates for the nonlinear regression problem.

![MODEL Procedure - OLS Estimation](image)

Nonlinear OLS Summary of Residual Errors

| Y | 2 | 19 | 7.6103 | 0.460054 | 0.63289 | 0.7071 | 0.6917 |

Nonlinear OLS Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Std Err</th>
<th>Ratio</th>
<th>Prob&gt;</th>
<th>T</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>4.270858</td>
<td>0.65770</td>
<td>6.43</td>
<td>0.0002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>0.057470</td>
<td>0.008904</td>
<td>6.35</td>
<td>0.0002</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Number of Observations | 21 | Objective | 0.3624 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing</td>
<td>5</td>
<td>Objective*</td>
<td>7.4103</td>
</tr>
</tbody>
</table>

PROC MODEL also has features for solving the simultaneous system of nonlinear equations to compute predicted values, and simulating the dynamic behavior of the model predictions over time.

Simulation models are especially useful for policy analysis. Simple extrapolation methods are often very effective for forecasting, but to predict the result of a change in a policy variable, you need a model that takes into account complex interrelationships between dependent variables and across time.
So far, we have talked about using SAS/ETS procedures for analyzing business and economic data. To some extent, the features of SAS/ETS are also available through menu driven interfaces. We've already mentioned the new SAS/ETS Forecast menu system.

Several other features of SAS/ETS software can be used through menus in the SAS/ASSIST product. Indeed, several SAS/ASSIST features rely on SAS/ETS procedures.

SAS/ASSIST applications using SAS/ETS procedures include Loan Analysis (PROC LOAN), Regression with correction for autocorrelation (PROC AUTOREG), Seasonal adjustment (PROC X11), and Convert frequency of time series data (PROC EXPAND).

In summary, business and economic data is usually time series data. To analyze business and economic data, you need data analysis methods that take into account dependencies across time and simultaneous dependencies within time periods.

Using traditional methods like ordinary regression analysis for data which does not satisfy the independence assumptions required by ordinary statistical methods produces incorrect results.

SAS/ETS software provides a wide range of tools designed for the analysis of this kind of data. Some of the SAS/ETS procedures we discussed are:

- AUTOREG, FORECAST, ARIMA, STATESPACE, PDLREG, and TSCSREG for time series analysis and forecasting
- SYSLIN and SIMLIN for linear regression with dependent regressors and simultaneous systems
- X11 for seasonal adjustment
- EXPAND for interpolating time series to different frequencies
- DATASOURCE for accessing financial and economic data bases
- LOAN for loan analysis
- COMPUTAB for report generation and spread sheet calculations
- MODEL for nonlinear systems regression and large scale simulation models

SAS/ETS tools are applicable to any time series data, but the pervasive role of time in business and economic data makes SAS/ETS procedures especially useful in this field.