Paper 243-2009

Model Fitting and Data Analyses in SAS/ETS® Software Using ODS Statistical Graphics

Jan Chvosta, SAS Institute Inc., Cary NC, Mark Little, SAS Institute Inc., Cary NC

ABSTRACT

Graphical visualization of statistical results is increasingly important in all areas of data analysis, and SAS/ETS procedures now provide graphical output for many kinds of econometric and time series models. Correct use of these features helps you to diagnose data patterns and choose appropriate models. In this paper, several SAS/ETS procedures (including the AUTOREG, PANEL, UCM, and SIMILARITY procedures) are used to demonstrate a graphical diagnostic tool for effective modeling of cross-sectional, time-series, and panel data. The emphasis of the paper is on correct model selection and effective use of SAS/ETS software.

OVERVIEW

ODS Statistical Graphics (also called ODS Graphics) is an extension to ODS (Output Delivery System) that is implemented in SAS® 9.2. ODS Graphics enhances the capability of the SAS/ETS software by enabling you to visualize results and find patterns and trends in data that would be otherwise difficult to determine. It summarizes the information into graphs and further enhances the analytical power of the SAS software. This paper demonstrates the use of ODS Graphics on several examples. Each of the examples analyzes simulated time series, panel, or cross-sectional data with known statistical properties.

Graphical output is also available in other SAS/ETS procedures including the MODEL, SYSLIN, ENTROPY, ARIMA, and VARMAX procedures. Even though the syntax and plot availability vary somewhat across procedures, most of the techniques and principles of data analysis with ODS Graphics discussed in this paper can be easily extended to other procedures. For more information about SAS/ETS software and ODS Graphics, see the *SAS/ETS 9.2 User's Guide*.

WHAT IS ODS GRAPHICS?

ODS Graphics enhances the typical tabular output delivered by SAS/ETS procedures. It is a tool that enables you to produce plots to analyze data patterns and choose an appropriate model more easily.

The plots are produced automatically and can be saved in different graphical formats ranging from HTML to RTF. ODS Graphics needs to be enabled before its use. It is enabled by executing the following SAS statement:

```
ods graphics on;
```

ODS Graphics stays enabled through the whole SAS session unless disabled by the following statement:

```
ods graphics off;
```

Once the ODS Graphics is enabled, you can access various plot options. The plot options are usually located in the PROC or MODEL statements. The UCM procedure enables you to specify the PLOTS option for each of the components, estimates, and forecast. If you specify the PLOTS option without specifying any other options, the default series of plots is displayed. You can further control the plot selection by using local options. For example, the following statement displays the autocorrelation function and white noise plots for the AUTOREG procedure:

```
proc autoreg data=a plots(unpackpanel only) = (ACFPlot WN);
```

ANALYZING TIME SERIES DATA WITH THE AUTOREG PROCEDURE

The AUTOREG procedure estimates linear regression models for time series data when the errors are autocorrelated. When time series data are used in regression analysis, often the error term is not independent through time. Instead, the errors are *serially correlated* or *autocorrelated*. If the error term is autocorrelated, the efficiency of ordinary least squares (OLS) parameter estimates is adversely affected and standard error estimates are biased. The autoregressive error model corrects for serial correlation. The AUTOREG procedure can fit autoregressive error models of any order. The autoregressive model has the form:

$$y_t = \mathbf{x}_t'\beta + v_t$$

$$v_t = -\varphi_1 v_{t-1} - \varphi_2 v_{t-2} - \ldots - \varphi_m v_{t-m} + \epsilon_t$$

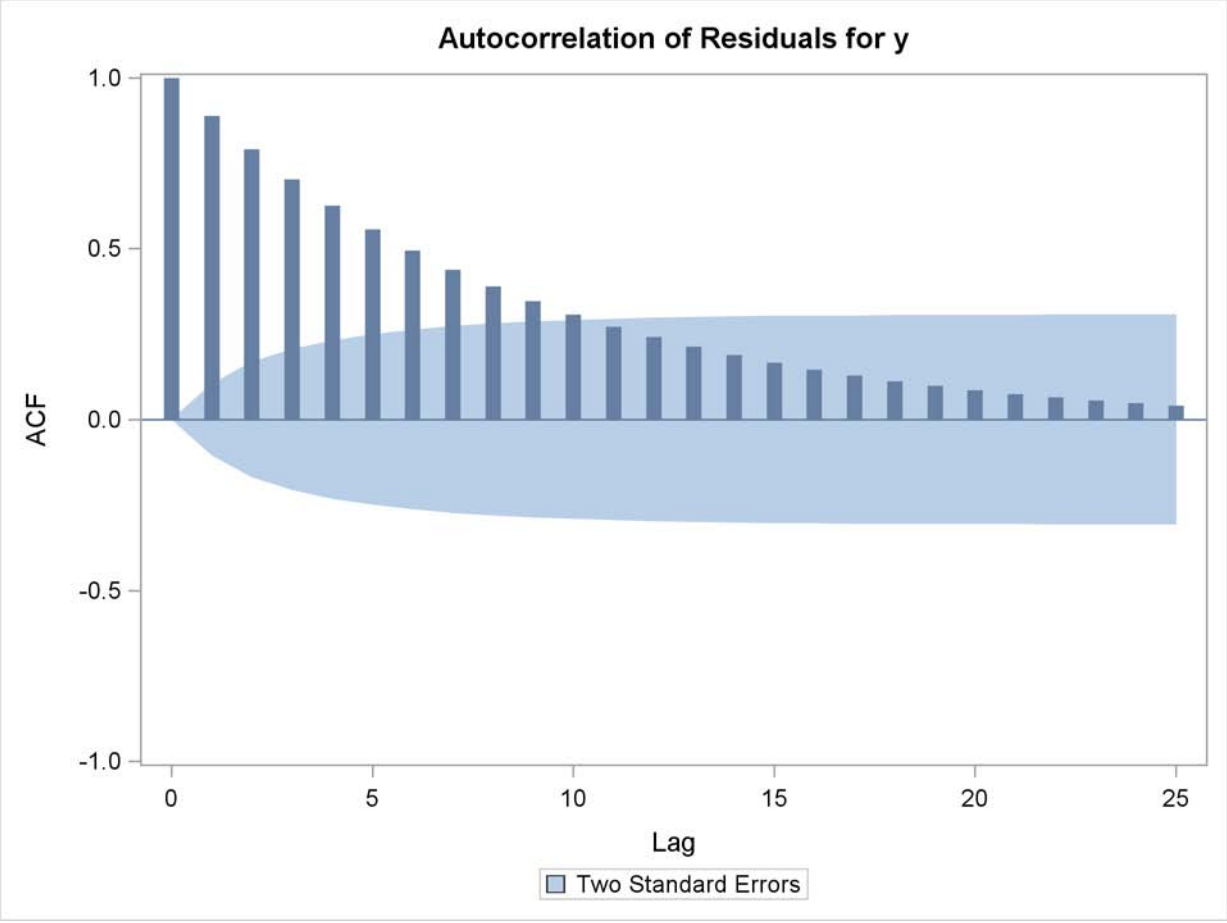
$$\epsilon_t \sim \mathrm{IN}(0, \sigma^2)$$

The following example uses simulated data to analyze different time series patterns. First you simulate an AR(2) process that is nonstationary. It is necessary to test the stationarity condition before proceeding further with analysis. This test can be done by using tests provided by the AUTOREG procedure or by examining the autocovariance (ACF) function. If the ACF function is slowly declining over time, the data might be nonstationary or the unit root might be present. The following statements fit the OLS model to a nonstationary AR(2) series:

```
proc autoreg data=a plots(unpackpanel only) = ACFPlot;
model y = time;
run;
```

The ACF function for the simulated nonstationary series is depicted in Figure 1.

Figure 1 Nonstationary ACF Plot

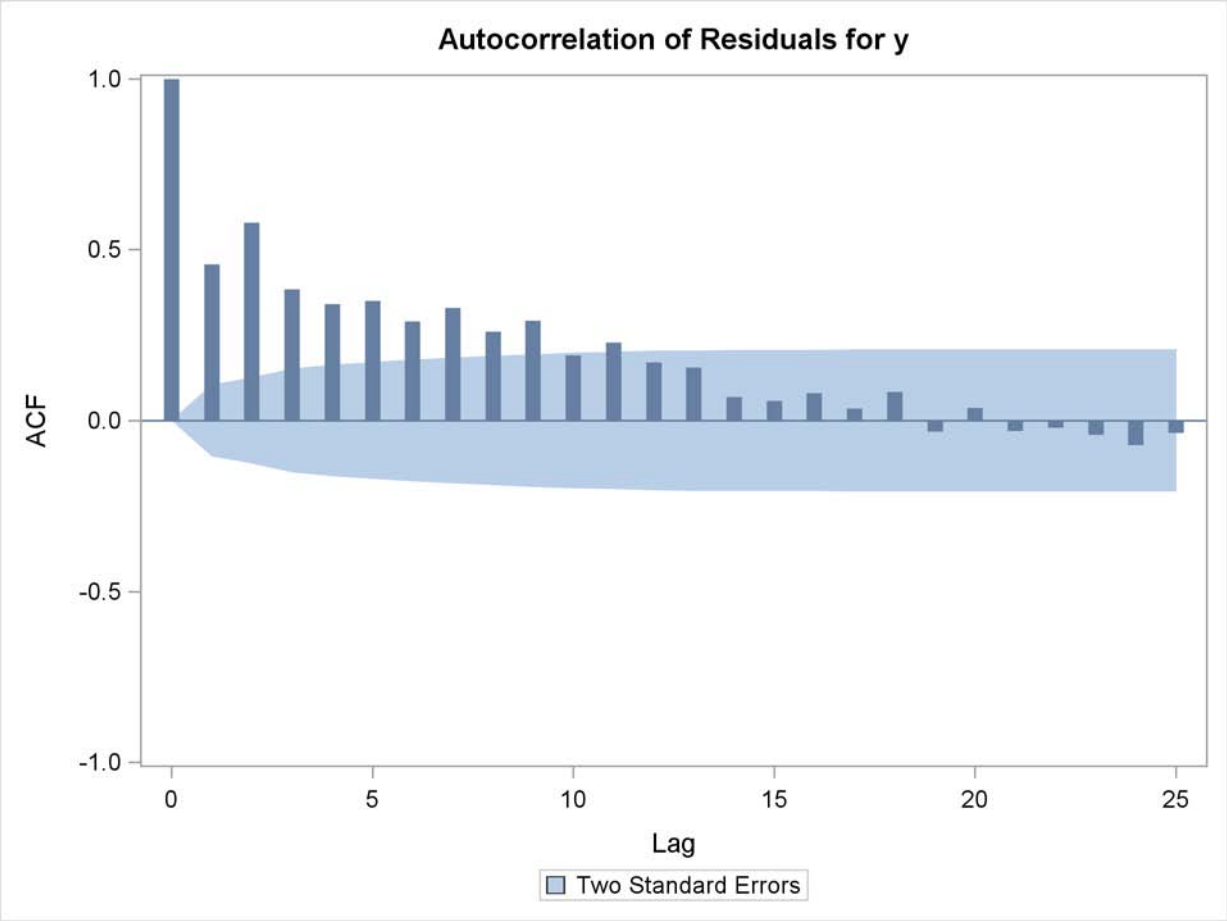


Since the process is nonstationary, it is not too big a surprise that the ACF function is only slowly declining. If you want to proceed any further with analysis of this time series, you would need to difference the series to ensure stationarity.

The following step creates a stationary AR(2) process whose plot is shown in Figure 2. Then you can compare the ACF functions with the nonstationary process presented in Figure 1.

```
proc autoreg data=b plots(unpackpanel only) = ACFPlot;
model y = time;
run;
```

Figure 2 Stationary ACF Plot



In this example, the coefficients for the autoregressive process were selected in such a way that the resulting process is stationary. How do you go about analyzing the data and fitting the most appropriate model? The ACF function depicted in Figure 2 is declining more rapidly than in the nonstationary case. However, it is not obvious from Figure 2 that the process is stationary. If this were not a simulated data set with known properties, additional testing would be needed. Since the emphasis of the paper is on ODS Graphics, these tests are skipped and stationarity is assumed for further analysis. It was already mentioned that OLS regression provides results with biased standard errors, but it might still be a good starting point in determining the appropriate model.

The following statements execute a simple model that uses simulated stationary data. The PLOTS=ALL option requests all available plots. By default, the plots are grouped into two diagnostic panels which are shown in Figure 3 and Figure 4.

```
proc autoreg data=b plots=all;
model y = time;
run;
```

Figure 3 depicts the residuals, actual values, predicted values, Cook's D plot, and Q-Q plot. When you examine the plots in this panel, it is not obvious that the residuals are not independent and identically normally distributed.

Figure 3 Diagnostics Panel 1

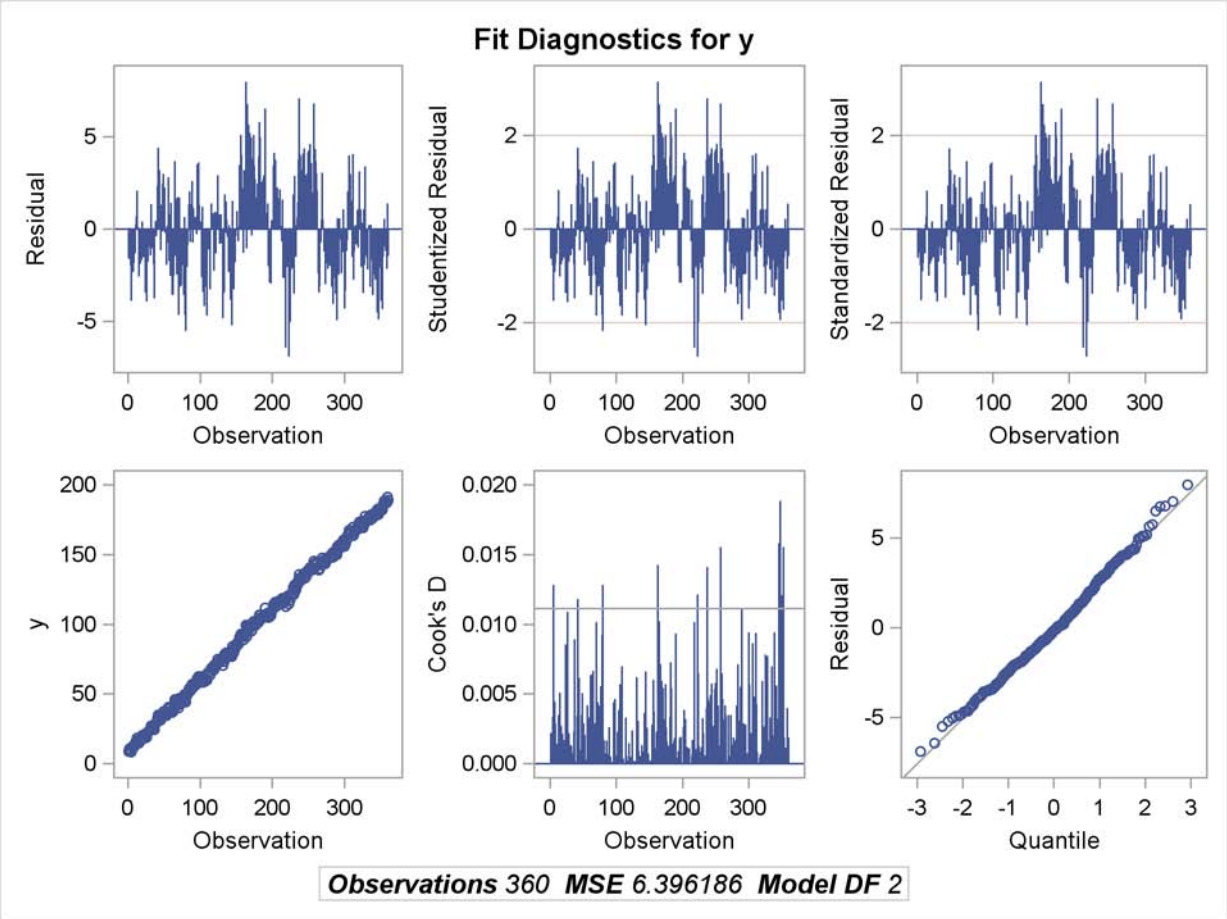
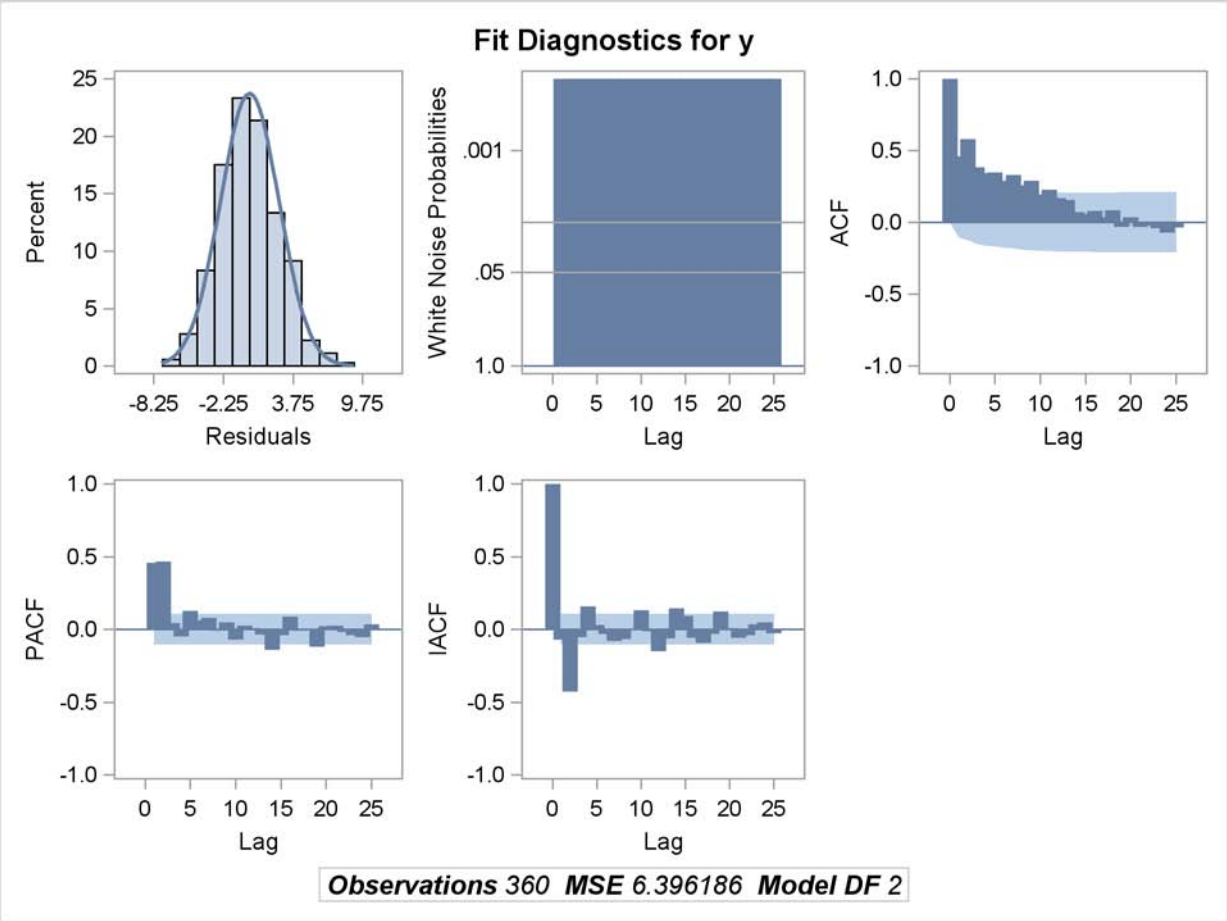


Figure 4 might be more useful in deciding what model to fit. The white noise probabilities are all low (high bars), indicating that the obtained residuals are not white noise and that OLS might not be the best model to fit. This suspicion is further confirmed by examining the partial autocorrelation function (PACF) plot. The two bars outside of the two standard error bend indicate that the correct model to fit would be AR(2).

Figure 4 Diagnostics Panel 2

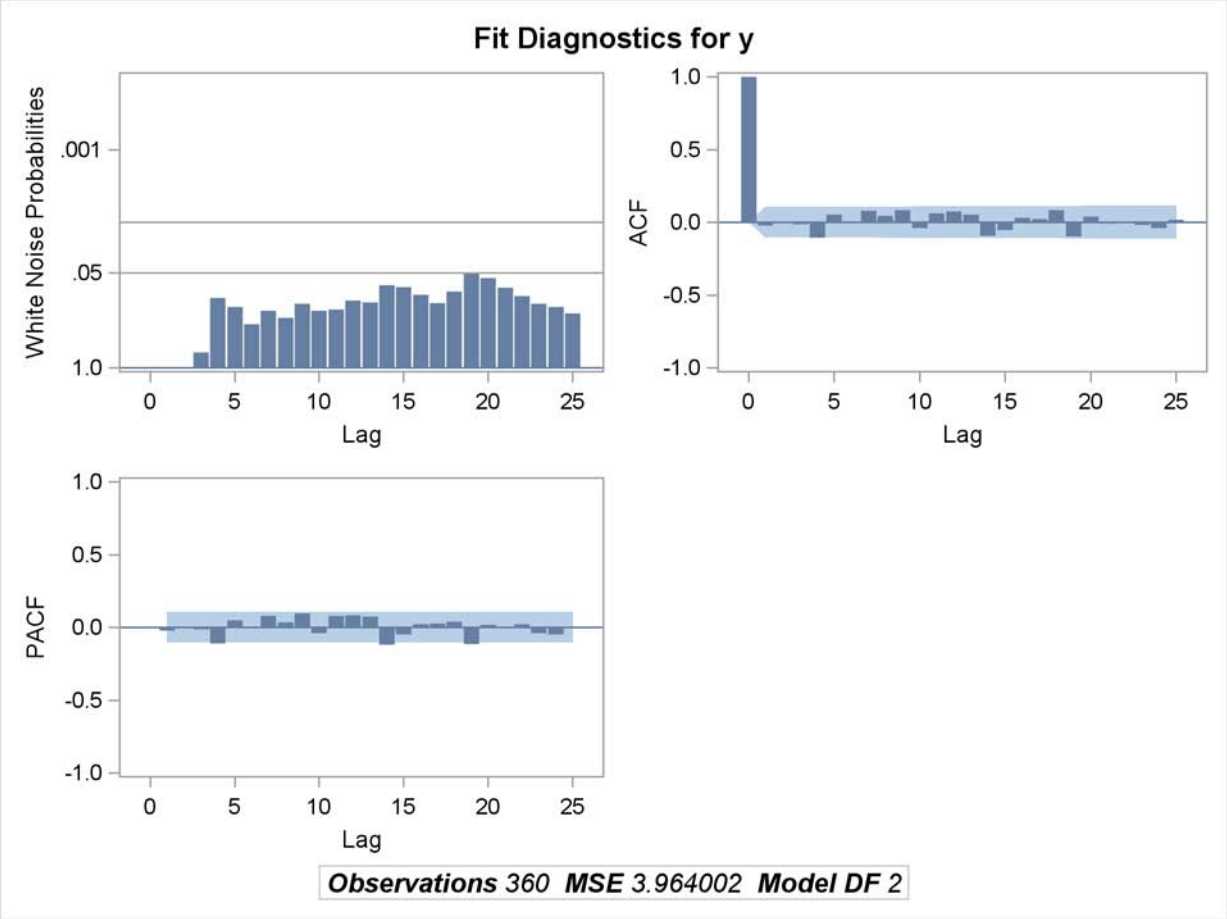


The following statements use the NLAG option in the MODEL statement to specify an AR(2) model. The PLOTS= option again requests plots that help determine whether the AR(2) model is the correct one to fit.

```
proc autoreg data=b plots(only) = (ACF PACF WN);
model y = time /nlag=2;
run;
```

The resulting plots are depicted in Figure 5.

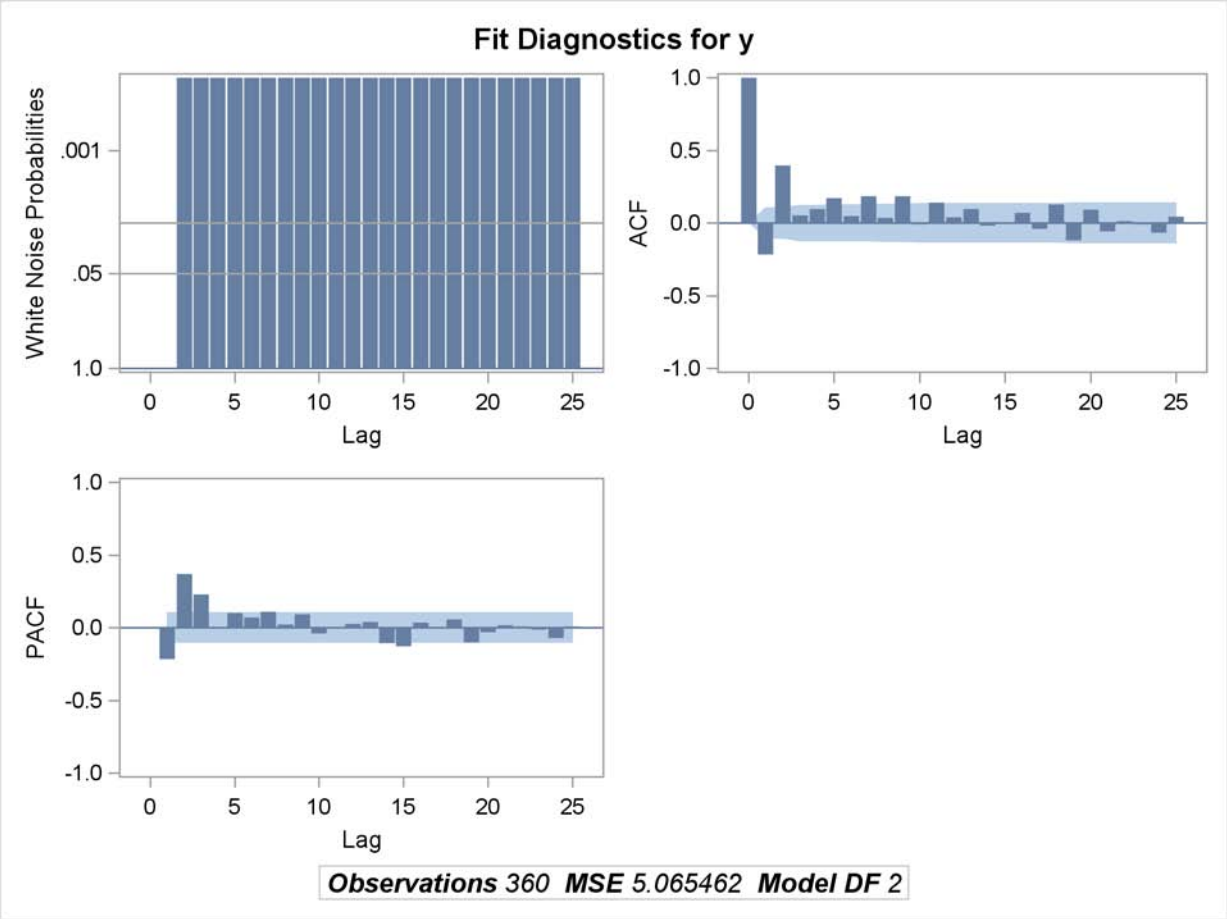
Figure 5 AR(2) Diagnostics Panel 1



The white noise probabilities plot indicates that the resulting residuals are white noise, and the PACF plot indicates that all partial autocorrelations are within the two standard error bends. The AR(2) model was the appropriate one to fit.

Another interesting question is what happens if you fit a lower order lag AR(1) model to AR(2) stationary series. The white noise plot in Figure 6 shows that the autocorrelation at the first lag was removed but that it still remains at higher lags. Analyzing the white noise plot along with ACF and PACF functions where some autocorrelations are outside of the two standard error bends leads to a conclusion that AR(1) is not the right model to fit.

Figure 6 AR(1) Diagnostics Panel 1

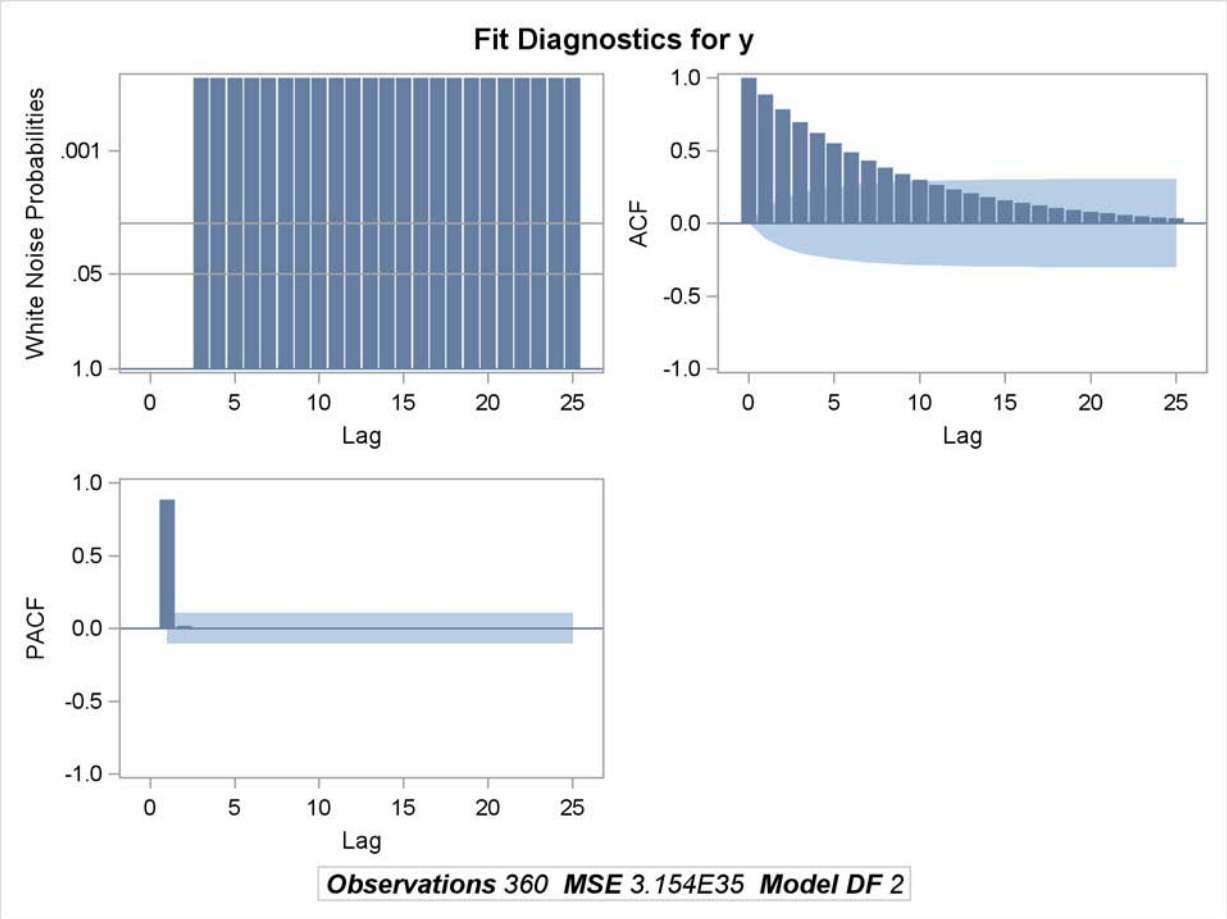


As mentioned earlier, if the data were found to be nonstationary to start with, the data would have to be differenced. The following statements request ACF, PACF, and WN plots for an AR(2) model fitted to nonstationary data without differencing:

```
proc autoreg data=a plots(only) = (ACF PACF WN);
model y = time /nlag=2;
run;
```

When you compare Figure 7 (produced by the nonstationary model) to Figure 5 (created using the stationary model), you notice that obtained residuals for the nonstationary process are not white noise. In the nonstationary case, the ACF function is still only slowly declining. It would be reasonable to conclude that the AR(2) model is not the right one to fit.

Figure 7 Diagnostics Panel 1



ANALYZING TIME SERIES DATA WITH THE UCM PROCEDURE

The UCM procedure analyzes univariate time series data by using the unobserved component model (UCM). The UCM model can control for seasonality, cyclical patterns, and various trends. The following example demonstrates the use of UCM procedure on a data set with monthly airline data AIR. This data set is distributed with SAS/ETS software as a part of the SASHELP library. The **air** variable was transformed by taking a natural logarithm to correct data asymmetry.

In the UCM procedure, plot options can be used in statements to represent the components of the model (IRREGULAR, LEVEL, SLOPE, SEASON) or forecasted series (FORECAST). Plot options in the ESTIMATE statement produce residual diagnostics plots that are somewhat similar to plots produced by the AUTOREG and PANEL procedures; these plot options are not discussed in this section. The following statements estimate a UCM model of **logair** with irregular error, level, slope, and season components. A forecast of **logair** is obtained for 24 periods (months).

```
proc ucm data = air;
 id date interval = month;
 model logair;
 irregular;
 level;
 slope var = 0 noest;
 season length = 12 type=trig plot=smooth;
 estimate back=24 plot=panel;
 forecast back=12 lead=24 print=forecasts plot=forecasts;
run;
```

Smooth seasonal component plots and forecasts for **logair** are presented in Figure 8 and Figure 9.

Figure 8 Smoothed Seasonal Plot

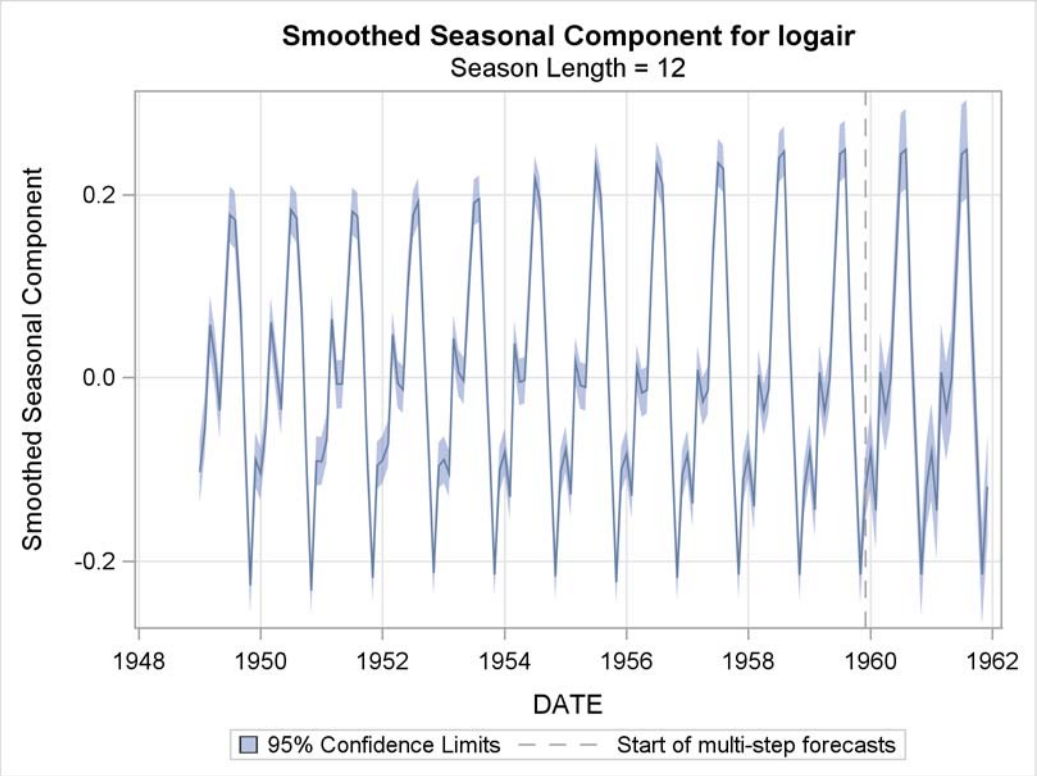
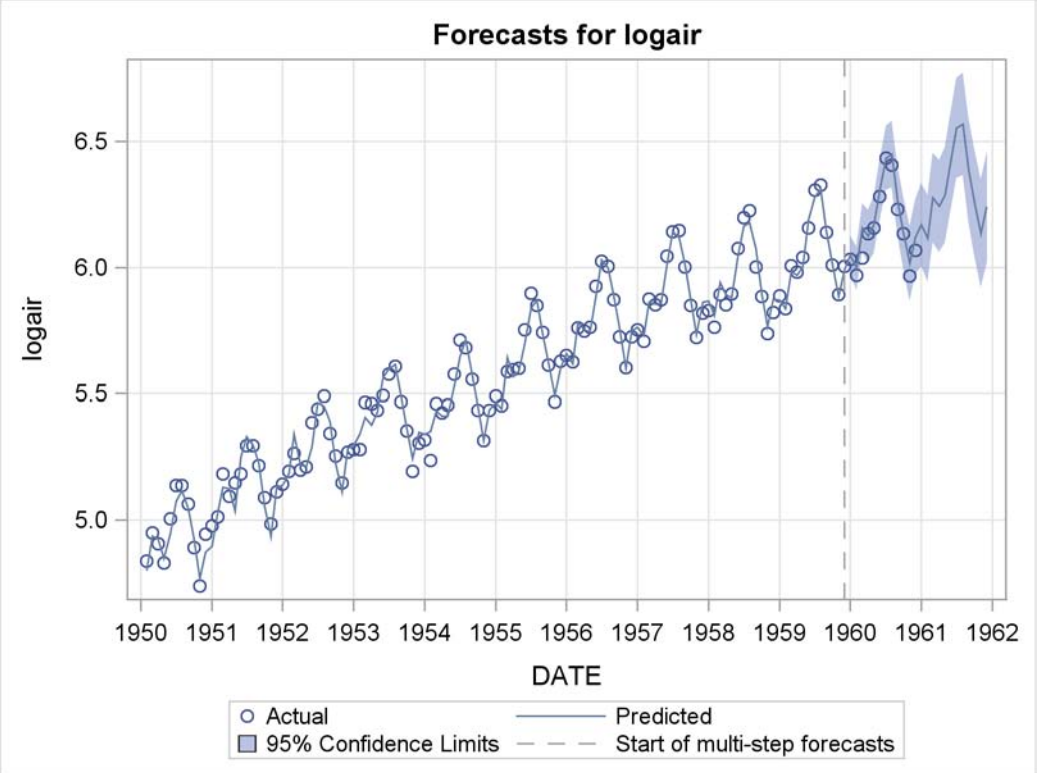


Figure 9 Forecasts Plot



ANALYZING PANEL DATA USING PANEL PROCEDURE

The PANEL procedure analyzes data that combine time series and cross-sectional effects. Typical examples of these models include one-way or two-way fixed- or random-effects models. Panel data include surveys (one subject is questioned multiple times in different years), trade data, or data collected for different firms in multiple years. This paper fits one-way fixed-effects model to simulated data to show the use of ODS Graphics. The following notation represents the one-way fixed-effects model:

$$y_{it} = \sum_{k=1}^{K} x_{itk}\beta_k + u_{it} \qquad i = 1,\ldots N; t = 1,\ldots T_i$$

with the specification of u_{it} as

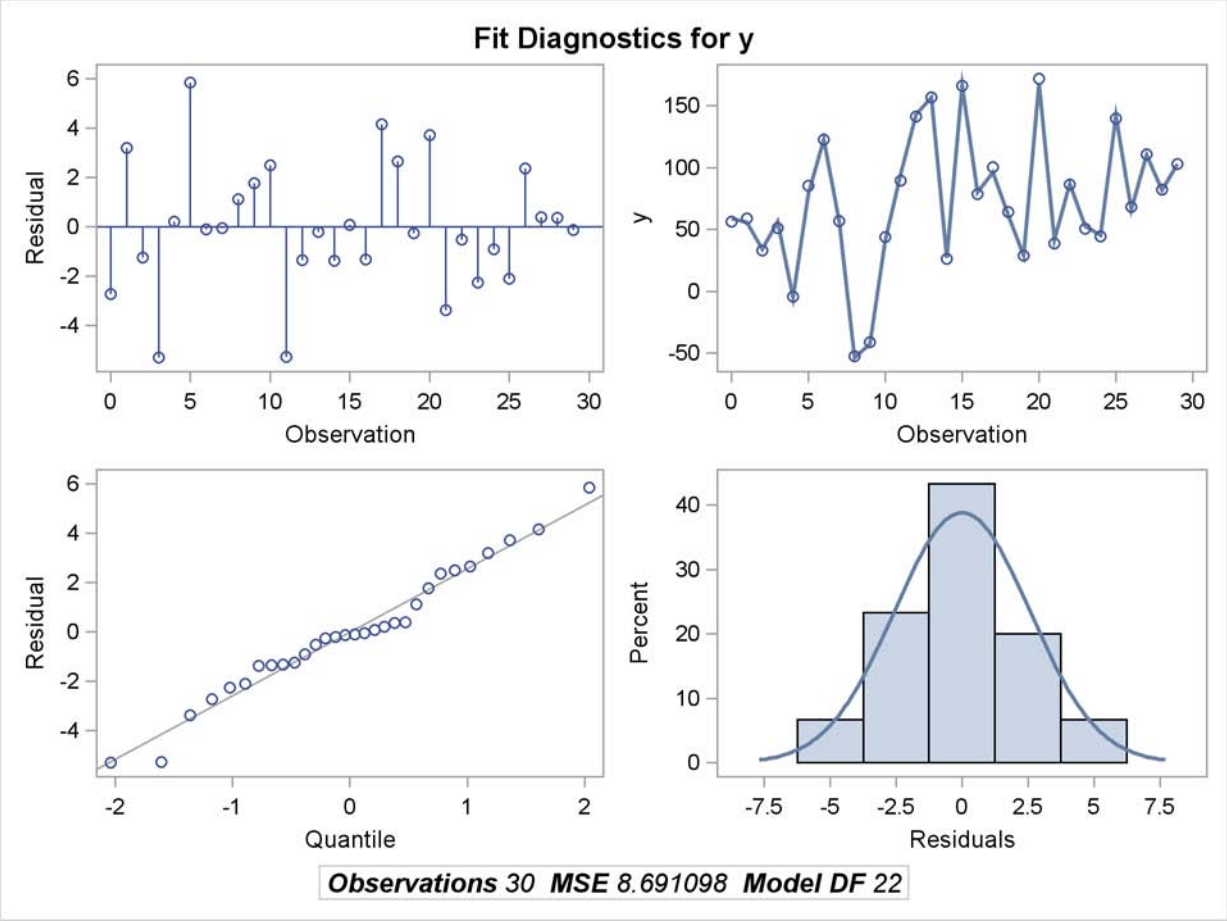
$$u_{it} = \gamma_i + \epsilon_{it}$$

where the γ_is are nonrandom parameters to be estimated.

The following statements estimate a one-way fixed-effects panel model and produce a panel version of all available plots:

```
proc panel data=b;
model  y = x1 x2 x3 /fixone plots=all;
id i t;
run;
```

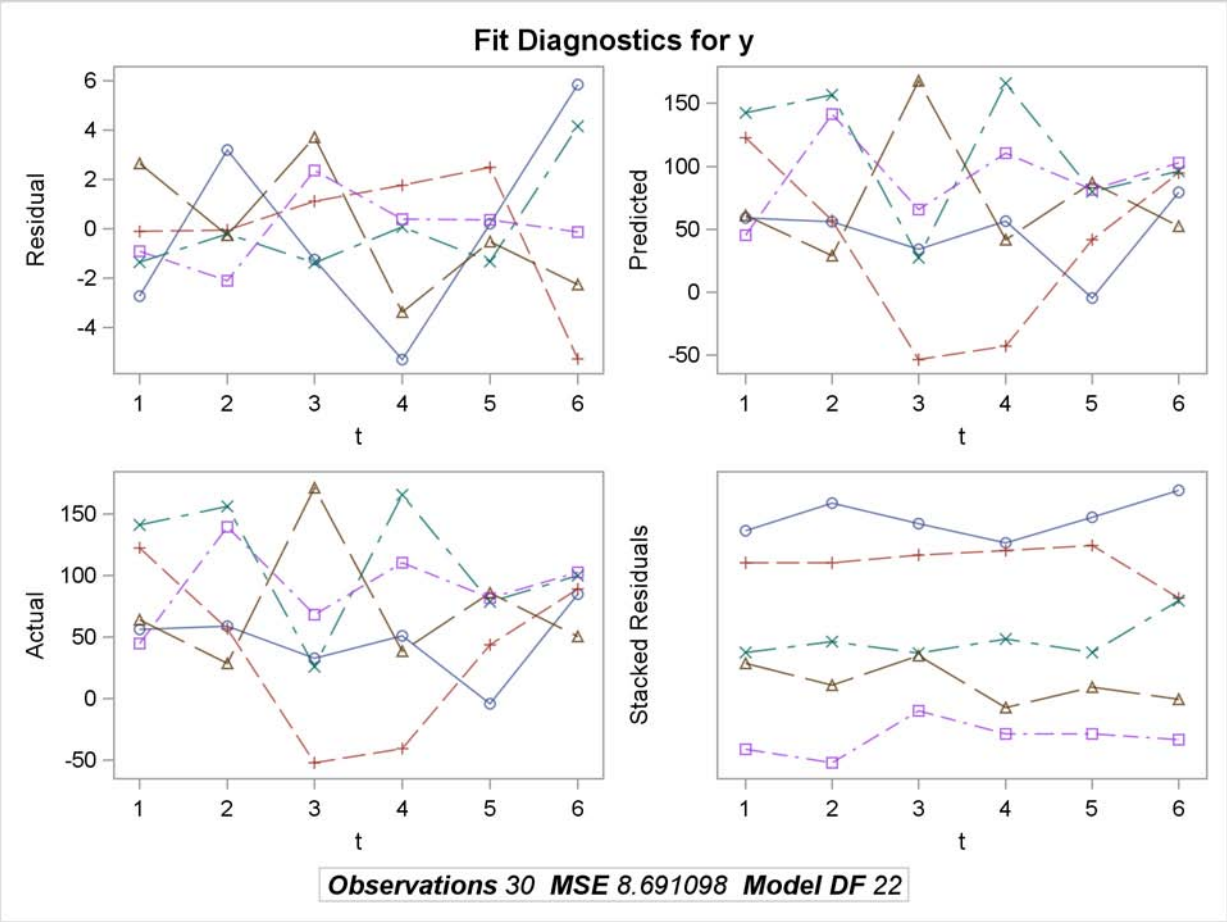
Figure 10 Diagnostics Panel 1



Several diagnostics plots shown in Figure 10 depict the residuals, actual values, predicted values, Cook's D test for normality of residuals, and the histogram of residuals. The histogram of residuals is slightly skewed to the right, but overall it appears that the one-way random-effects model is appropriate to fit.

This is somewhat confirmed by Figure 11, which shows plots of cross sections in time. There are no obvious patterns that would signal nonnormality of residuals and therefore a problem with fitting a model that is not appropriate.

Figure 11 Diagnostics Panel 2

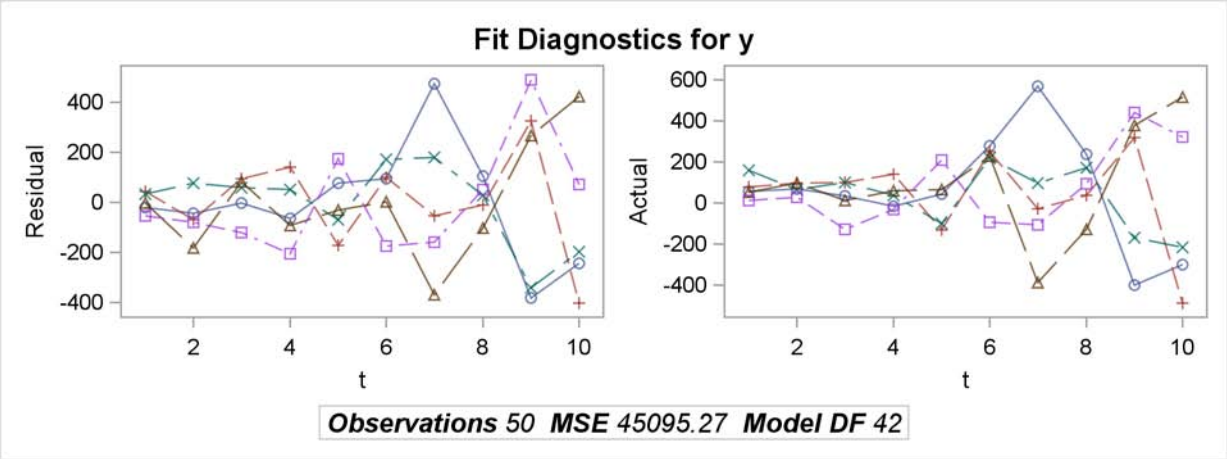


You can enhance the model presented in Figure 10 and Figure 11 by adding four time periods for each cross section and making the error term dependent on the time period.

Since $\mathbf{v_{it}}$ clearly has a trend built into it now, you would expect graphs similar to Figure 11 to capture this trend by showing an increase in variance of residuals with increasing time. The following statements use the UNPACKPANEL option to produce plots of predicted values and residuals:

```
proc panel data=b;
model  y = x1 x2 x3 /fixone plots(only)=(ActSurface ResidSurface);
id i t;
run;
```

Figure 12 Surface Plot of Actual Values and Residuals



The range of residuals and actual values is much lower in the beginning of the series than in its end, indicating that heteroscedasticity could be present.

EXPLORING SIMILARITY MEASURES WITH THE SIMILARITY PROCEDURE

The SIMILARITY procedure analyzes sequentially ordered data and calculates similarity measures by accumulating data into time series format. In order to produce similarity measures, two ordered data sequences are required (input and target). Given the input and target sequence, the similarity measure is a distance between the two. The SIMILARITY procedure offers many options that normalize and transform the series. All resulting series can be stored in output data sets or graphed by using ODS Graphics. This section presents several PROC SIMILARITY ODS Graphics plots that give you insight to the capability of PROC SIMILARITY. To fully understand the options and view examples with additional ODS plots, see Chapter 22, "The SIMILARITY Procedure" *(SAS/ETS User's Guide)*.

The following statements modify the data-generating process used in Figure 2 to demonstrate the use of PROC SIMILARITY. The TARGET statement adds a new series **y2**, which has the same slope but a greater intercept (25) on the time coefficient variable. The sixth observation in the series was altered so that at this point series **y2** peaks in the direction opposite from **y1**. The AR(2) process is used for the error **u** for both **y1** and **y2** series. The series starts on January 15, 2008, and represents monthly observations.

Time series **y1** is plotted along with series **y2** in Figure 13 by using the following statements:

```
proc similarity data=b out=timedata plot= Sequences;
 id date interval=month accumulate=total;
 input y1;
 target y2;
run;
```

To compare how similar the two newly created time series are, you can add PATHS and WARPS options to the PROC SIMILARITY example as follows:

```
proc similarity data=b out=timedata plot= (Sequences Paths Warps);
 id date interval=month;
 input y1/normalize=standard;
 target y2/normalize=standard;
run;
```

Figure 14 represents the path plot for the transformed data. There are many path plots through the distance matrix. The path represented in Figure 14 is a measure that minimizes path cost. Horizontal sections of the path plot represent an extension of the target sequence with respect to the input sequence; vertical movements represent its contraction; and diagonal movements represent a direct mapping between the target and imput series. You can see from Figure 14 that the normalized series differ between points 2 and 7.

Figure 13 Sequence Plot

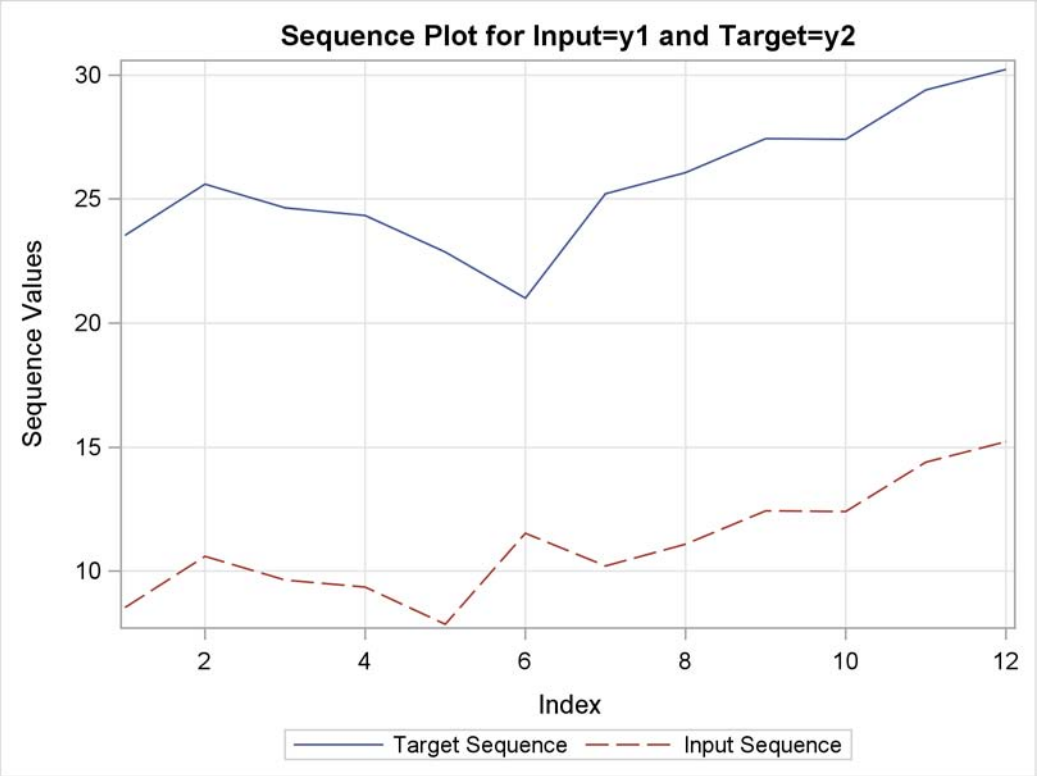
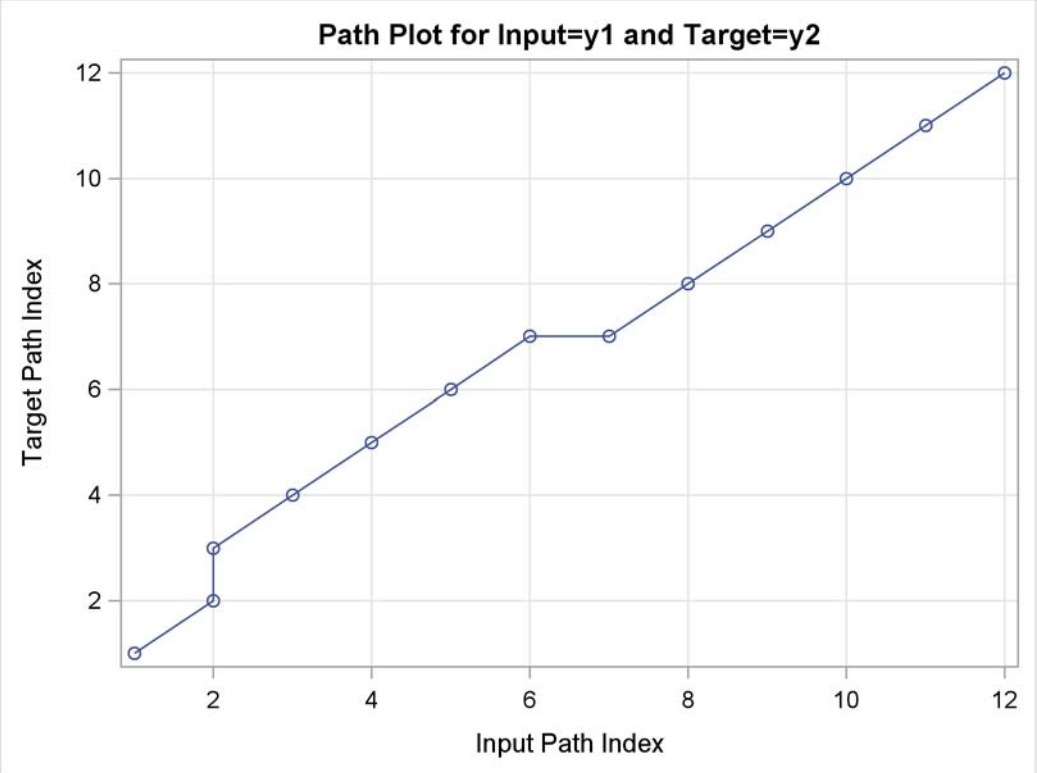
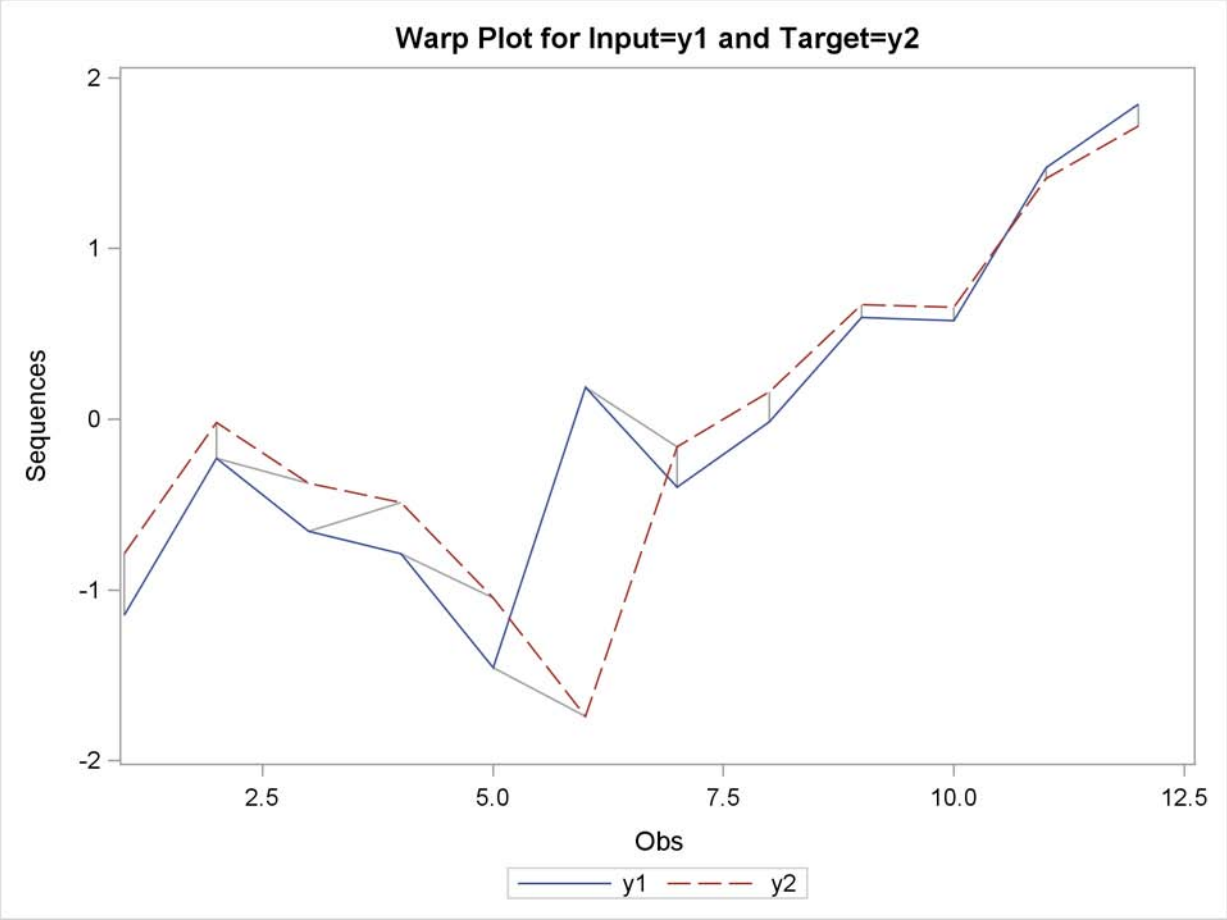


Figure 14 Path Plot



The warp plot in Figure 15 shows the exact mapping between the two series. You can see that points 2 and 3 of the target series **y2** are mapped into point 2 of the input series **y1** (the target series contracts) while point 7 of the target series is mapped into points 6 and 7 of the input series (the target series expands).

Figure 15 Warp Plot



CONCLUSION

This paper reviews the use of ODS Graphics in several SAS/ETS procedures. Graphical output is an important tool that can be easily combined with tabular output in order to produce thorough econometric analysis. The availability of plot options varies across the procedures to accommodate procedure-specific requirements. Plot options are usually specified in the PROC statement; they often contain global options which apply to all plots and local options which are plot specific.

This paper is intended to introduce the use of ODS Graphics in SAS/ETS software; it only touches on this important topic. For more information about ODS Graphics and SAS/ETS procedures, visit the Statistics and Operations Research Web site at `www.support.sas.com/rnd/app/`. The SAS/ETS documentation with many ODS Graphics examples can be also accessed from this web page.