INTRODUCTION to TIME SERIES ANALYSIS in SAS

This document provides thumbnail sketches of several important procedures and approaches to time series analysis in SAS.

Time series = a sequence of observations taken on a variable at successive points in time.

Objectives of time series analysis:

1. To forecast/predict future values of the time series
2. To understand the structure of the time series (how it depends on time, itself, and other time series variables)

Four important features often found in time series:

1. Trend
2. Seasonality
3. Autocorrelation
4. Heteroscedasticity

Autocorrelation = correlation between a time series and its lags (interpreted as the ability of past values of the series to affect the present or the future of the series) (also called *serial correlation*)

Ex: Time series: 2 5 3 8 9 (A)
Lag 1 of time series: . 2 5 3 8 (B)
Lag 2 of time series: . . 2 5 3 (C)
Autocorrelation function ACF at lag 1 = Corr(A,B) = 0.389; ACF lag 2 = Corr(A,C) = 0.645

NOTATION for illustrations to follow below: Suppose Y is a (response) time series, X is a (predictor) time series, t is time in quarters (1,2,3,4,5,6,7,8,9...), n is the number of time periods, S is season (1,2,3,4,1,2,3,4,1,2,...), ε is a residual error,. Where convenient to indicate that the series is a function of time, I write Y_t , X_t , and ε_t .

Detecting Autocorrelation

1. Autocorrelation function. ACF of Y at lag k is $Corr(Y_t, Y_{t-k})$, k = 1,2,3.... Can be obtained in a couple of different ways:
(1) by creating lags in DATA step and running through PROC CORR.

```
Ex: data example;
      input y;
      lag1 = lag1(y);
      lag2 = lag2(y);
      lag3 = lag3(y);
    cards; ...
    proc corr;
      var y lag1 lag2 lag3;
```

(2) Can also be obtained from PROC AUTOREG.

```
Ex: proc autoreg data = example;
      model y = / nlags=5;
```
(produces autocorrelation at lags 1 - 5)

2. Durbin-Watson statistic: Measures autocorrelation in regression residuals. Obtained from PROC REG by specifying the `dw` option in the `model` statement.

```
Ex: proc reg data=example;
      model y = x / dw;
```

Also, is printed automatically as part of the standard PROC AUTOREG output.

```
Ex: proc autoreg data=example;
      model y = x ;
```

3. Graphical: Plot time series and draw horizontal line at the mean of *Y*. If the time series crosses the mean line infrequently, then there is positive autocorrelation. If there are many crossings of the mean line, then there is negative autocorrelation. If the autocorrelation is really 0, then the expected number of crossings of the mean line (actually, median line) is approximately $n/2$.

Autocorrelation can be induced by presence of trend, seasonality, model misspecification. So putting terms in the model to capture trend, seasonality, and missing model terms sometimes makes autocorrelation vanish.

Regression as a Portmanteau Method for Time Series Modeling

What is wrong with using regression for modeling time series?

- Perhaps nothing. The test is whether the regression residuals satisfy the regression assumptions: linearity, homoscedasticity, independence, and (if necessary) normality.
- Often, time series analyzed by regression suffer from autocorrelated residuals. In the real world, positive autocorrelation seems to occur much more frequently than negative.
- Positively autocorrelated residuals make standard regression tests seem more significant than they should be and confidence intervals too narrow; negatively autocorrelated residuals do the reverse.
- In some time series regression models, if the residuals are correlated with their own lags (autocorrelated) or correlated with the *X* predictors, the parameter estimates may be substantially biased, and the bias may not decline no matter how many more data you get.

These problems can often be fixed by creative use of regression.

To use regression methods on time series data, first plot the data over time. Study the plot for evidence of trend and seasonality. Use numerical tests for autocorrelation, if not apparent from the plot. Add predictors to the regression model that mirror the type of trend, seasonality, autocorrelation, and heteroscedasticity that you see in the data. The predictors that you need to do this are often suggested by visual inspection of the plot of the original data. The goal is to build those features into the estimated model so that they do not show up in the residuals. In order for

the regression model to be valid, the residuals should be independent and identically distributed. The residuals cannot satisfy these conditions if any features remain in the residuals: Since *residual = Y data – estimate*, then a feature present in the *Y* data can be filtered out if that feature is also built into the estimate. After all of the systematic features in the *Y* data have also been captured and put into the estimate, then what will remain of the residuals is just random noise – i.e., independent and identically distributed data. Your goal in modeling is to reduce the data to noise, which by definition has no further structure that can be exploited.

- Trend can be dealt with by using functions of time and/or lags of *Y* as predictors.
- Seasonality can be dealt with by using seasonal indicators and/or lags of *Y* as predictors or by treating SEASON as a CLASS (categorical) variable.
- Autocorrelation can be dealt with by using lags of the response variable *Y* as predictors – and often by fixing other problems that seem unrelated to autocorrelation.
- Heteroscedasticity can often be dealt with by converting *Y* to log(*Y*) or by converting *Y* to percentage changes in *Y*. Also (more advanced) by dividing all variables by a variable with which the magnitudes of the residuals are proportional.
- Run the regression and diagnose how well the regression assumptions are met, make adjustments, and repeat.
- Be alert to the common possibility that the same problem can trigger multiple problems – and that fixing the one problem may solve the other problems. E.g., nonlinearity often triggers heteroscedasticity, autocorrelation, and non-normality in the residuals.

[The above is meant as an incomplete thumbnail sketch of a potentially complex modeling process -- but it may get you started.]

Here is a sample SAS program that models quarterly data as a function of linear trend (*T*), with four categorical seasonal (dummy) variables, and dependence on the preceding two quarters.

```
Ex: data example;
      input Y;
      T = _n_;
      SEASON = mod(_n_,4);
      lag1 = lag1(Y);
      lag2 = lag2(Y);
    cards; ...
    proc plot data=example;
      plot Y * T = SEASON;
    proc glm data=example;
      classes SEASON;
      model Y = T SEASON lag1 lag2 / solution;
      output out=example p=predy r=residy;
    proc plot data=example;
      plot residy * T  residy * predy;
    proc autoreg data=example;
      model residy = / nlags=2;
    proc univariate data=example normal;
      var residy;
```

Some simple time series models (in each case, the errors ε_t must satisfy the regression assumptions – zero mean i.i.d. normal errors):

1. Random sample:

$$Y_t = \varepsilon_t$$

2. Deterministic time trend:

$$Y_t = \alpha + \beta T + \varepsilon_t$$

$$Y_t = \alpha + \beta_1 T + \beta_2 T^2 + \varepsilon_t$$

3. Random Walk:

$$Y_t = Y_{t-1} + \varepsilon_t$$

4. First-order autoregressive process:

$Y_t = \alpha + \beta Y_{t-1} + \varepsilon_t$ or equivalently: $Y_t = \alpha + u_t$ and $u_t = \rho u_{t-1} + \varepsilon_t$

5. Seasonal means model:

$$Y_t = SEASON_i + \varepsilon_t$$

where $SEASON_i$ means that *SEASON* has a different value for each season (*i*) – a one-way analysis of variance model, modeled by a set of dummy (0-1) variables for *SEASON*.

6. First-order moving average model (unweighted):

$$Y_t = \varepsilon_t - \varepsilon_{t-1}$$

This is equivalent to $Y_1 = \varepsilon_1$, $Y_1 + Y_2 = \varepsilon_2$, $Y_1 + Y_2 + Y_3 = \varepsilon_3$, etc.

7. First-order moving average model (weighted):

$$Y_t = \varepsilon_t - \beta \varepsilon_{t-1}$$

8. Second-order autoregressive process:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \varepsilon_t$$

9. Second-order moving average process:

$$Y_t = \varepsilon_t - \beta_1 \varepsilon_{t-1} - \beta_2 \varepsilon_{t-2}$$

Combination models are possible and common:

Ex: $Y_t = \alpha + \beta_1 T + SEASON_i + \beta_2 Y_{t-1} + \varepsilon_t$

To be useful for forecasting/predicting, the right-hand-side of the equation must not contain any variables that are unknown for the time for which the forecast must be made.

Important SAS PROCs for Time Series

FORECAST, AUTOREG, X11, ARIMA, STATESPACE, SPECTRA, SYSLIN
(all in SAS/ETS).

PROC FORECAST

Forecasts univariate time series efficiently, combining three models. Does not use other time series (like X_t) as predictors.

The three models that are used:

1. Trend - for long term deterministic trend: constant, linear, or quadratic
2. Autoregressive - for short term fluctuations
3. Seasonal - for regular seasonal fluctuations

PROC FORECAST uses by default a stepwise autoregressive procedure to select the number of lags in the autoregressive part of the model. It can also use exponential smoothing and Winters methods. The seasonal part can be additive or multiplicative.

```
Ex: proc forecast data=example out=newexamp interval=qtr
method=stepar trend=3 lead=4 seasons=qtr;
      var Y;
```

This fits a quadratic trend (`trend=3`) to quarterly (`interval=qtr`) *Y*, including seasonal components (`seasons=qtr`), and computes the residuals; then fits an autoregressive process to the residuals, selecting those that are significant (`method=stepAR`); then forecasts the next four quarters (`lead=4`), storing the forecasts in WORK.NEWEXAMP (`out=newexamp`).

PROC AUTOREG

Fits models of the form $Y_t = \alpha + \beta_1 X_{1t} + \beta_2 X_{2t} + \beta_3 X_{3t} + \cdots + \beta_k X_{kt} + u_t$, where u_t is an autoregressive process, i.e., $u_t = \varepsilon_t - \alpha_1 u_{t-1} - \alpha_2 u_{t-2} - \cdots - \alpha_m u_{t-m}$. By default, parameter estimation uses the Yule-Walker equations, but may also use maximum likelihood or unconditional least squares,

Examples:

```
  1. proc autoreg data=example;
       model Y = ;
```

-- produces ACF of Y_t.

```
  2. proc autoreg data=example;
       model Y = / nlag=1;
```

-- fits first-order autoregressive process to *Y* by Yule-Walker method. Includes random sample and random walk as possible special cases.

```
  3. proc autoreg data=example;
       model Y = / nlag=1 method=ml;
```

-- same as example (2) but fits by maximum likelihood.

```
  4. proc autoreg data=example;
       model Y = X / nlag=2;
```

-- fits $Y_t = \alpha + \beta_1 X_t + u_t$, where u_t is second-order autoregressive, by Yule-Walker method.

PROC X11

Seasonally adjusts monthly or quarterly time series with the Bureau of Census X11 Seasonal Adjustment program. Corresponds to classical decomposition of time series into trend, cyclical, seasonal, and irregular components still taught in many basic business statistics books.

```
Ex: proc x11 data=example yraheadout;
      quarterly start='85Q1' end='94Q1';
      output out=forexamp d10=d10 d11=d11;
```

Seasonally adjusts quarterly data (`quarterly`) from first quarter of 1985 through first quarter of 1994 (`start='85Q1' end='94Q1'`); adds forecasts of the next year (`yraheadout`), i.e., 1994 Q2-1995 Q1, to WORK.FOREXAMP and variables d10 (final seasonal factors) and d11 (final seasonally adjusted series) to WORK.EXAMP.

PROC ARIMA

ARIMA = Autoregressive Integrated Moving Average models.
Sophisticated time series models made popular by George Box and Gwilym Jenkins. Also called "Box-Jenkins" models. Requires advanced statistical knowledge to use well.
Includes autoregressive (AR) and moving average (MA) models as special cases. May have both AR and MA types in the same model, combined with differencing (I) and seasonality.

PROC STATESPACE

In state space models, the observable time series *Y* is regarded as composed of separate components like trend, seasonal, regression terms, and error. The components themselves are not directly observable, but are modeled as separate time series, and the whole is estimated together.

PROC SYSLIN

SYSLIN = SYStems of LINear equations. Often a time-dependent phenomenon requires more than one regression equation to describe its behavior. For example, the market for sugar may require one equation to describe price and another equation to describe demand as functions of sets of relevant predictors. Moreover, the two equations may be interconnected because price depends upon demand and demand depends upon price. Special methods are required to estimate the two equations simultaneously.

PROC SPECTRA

Performs spectral analysis, an alternative approach to time series analysis through the *frequency domain*, rather than the *time domain*. All of the preceding methodologies try to understand time series by analyzing the data values as sequences in time. Spectral analysis tries to understand time series by analyzing the Fourier transform of the time series for important frequencies that recur in the data. Understanding spectral analysis requires advanced mathematics.