SAS for Forecasting Time Series –

Part 1: Overview of Time Series & Simple Models: Autoregression

Charlie Hallahan

November 18, 2009

SAS for Forecasting Time Series”,
2nd edition by Brocklebank & Dickey, 2003

Outline*

The topics covered in the text are:

1. Overview of Time Series
2. Simple Models: Autoregression
3. The General ARIMA Model
4. The ARIMA Model: Introductory Applications
5. The ARIMA Model: Special Applications
6. State Space Modeling
7. Spectral Analysis
8. Data Mining and Forecasting

*These notes are based on the SAS Books by Users text "**SAS for Forecasting Time Series", 2nd edition by Brocklebank & Dickey, 2003**

Simple Models: Regression*

Three features of time series data that must be dealt with:

1. seasonality

2. trend

3. autocorrelation

Simple Models: Regression

Multiple regression model: $Y_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{2t} + \varepsilon_t$

Usual assumptions on ε_t : $\varepsilon_t \sim iid\ N\left(0, \sigma^2\right)$

With time series data, the *iid* assumption is commonly violated

Simple Models: Regression

The **Durbin - Watson** statistic tests for 1^{st} order autocorrelation, i.e., whether $\rho = Corr\left(\varepsilon_t, \varepsilon_{t\text{-}1}\right) = 0.$

$$DW = d = \sum_{t=2}^{T}\left(\hat{\varepsilon}_t - \hat{\varepsilon}_{t\text{-}1}\right)^2 / \sum_{t=1}^{T}\hat{\varepsilon}_t^2 \approx 2\left(1 - \hat{\rho}\right)$$

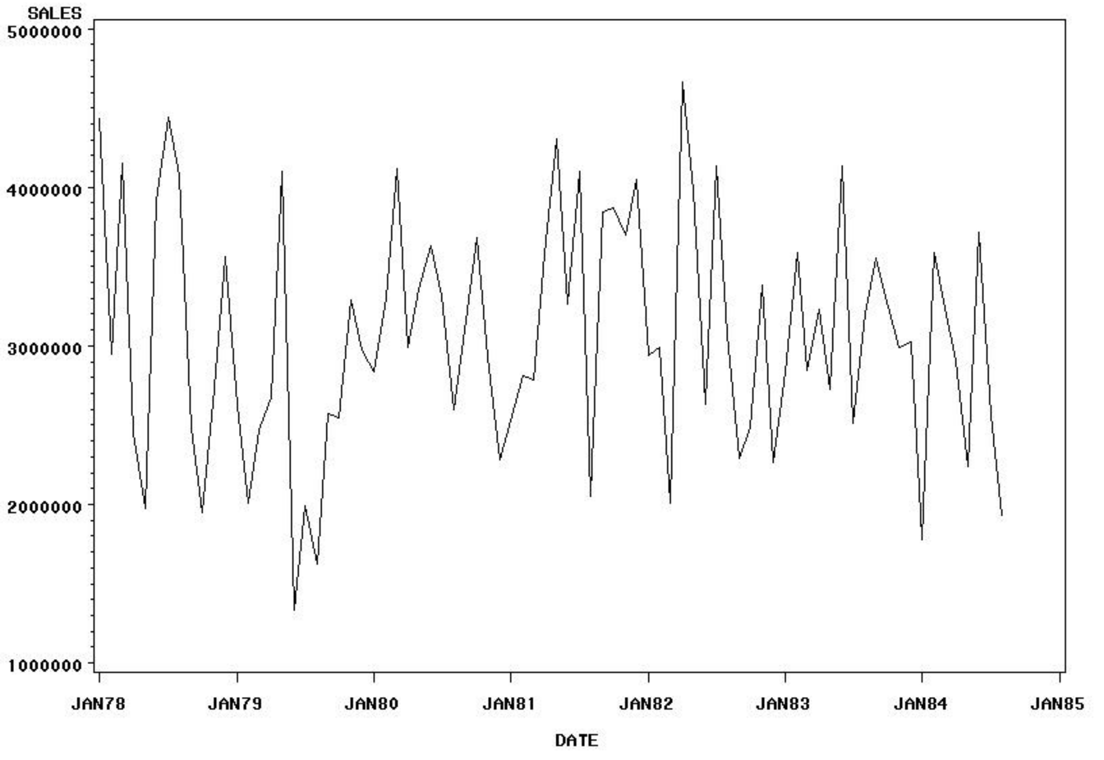
The significance of d depends on the number of regressors (k) in the model and the number of observations (n).

There also so-called *indeterminate* intervals such that if d falls into these intervals, no conclusion can be drawn.

Simple Models: Regression

Let Y_t = sales in month t, X_{1t} = expenditures in month t and X_{2t} = competitors sales in month t.

Plot of Monthly Sales Versus Date



Simple Models: Regression

```
title "Predicting Sales Using Advertising";
title2 "Expenditures and Competitors' Sales";
proc reg data=ffc.sales;
      model sales = adv comp / dw;
      output out=out1 p=p r=r;
run;
```

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	2700165	373957	7.22	<.0001
ADV	1	10.17968	1.91705	5.31	<.0001
COMP	1	-0.60561	0.08465	-7.15	<.0001

Durbin-Watson D	**1.394**
Number of Observations	80
1st Order Autocorrelation	**0.283**

Simple Models: Regression

In this example, $k = 3$ and $n = 80$. Given k and n, the critical value can still depend on the values of the explanatory variables.

To test for positive autocorrelation, **Durbin & Watson** state that the critical value must be between $d_L = 1.59$ and $d_U = 1.69$ (at the *5%* significance level).

Since $d = 1.349 < d_L$, we can conclude that the error term ε_t does exhibit **positive autocorrelation**.

The **DW test** only tests for 1st order autocorrelation.

The **DW test** is invalid if the lagged dependent variable is included as an explanatory variable.

An **omitted variable** which is itself autocorrelated can cause a significant DW *d*.

PROC AUTOREG can test for higher order autocorrelation and also provide **p-values**.

Simple Models: Regression

PROC AUTOREG incorporates a computer-intensive method of **Durbin & Watson** to calculate **exact p-values** for the test statistic ***d***.

We'll 1st estimate a misspecified model leaving out an important variable.

Using generated data with true values of parameters = 1.0, 2.0, and 3.0.

The **omitted variable** is ***X2*** which is autocorrelated (and, hence, gets absorbed into the error term)..

```
title "Significant DW statistic with omitted variable";
proc autoreg data=ffc.sales2;
      model y = x1 / dwprob;
run;
```

Simple Models: Regression

Dependent Variable y

Ordinary Least Squares Estimates

SSE	952.399237	DFE	78
MSE	12.21025	Root MSE	3.49432
SBC	433.950833	AIC	429.18678
Regress R-Square	1.0000	Total R-Square	1.0000
Durbin-Watson	**1.2073**	**Pr < DW**	**0.0001**
Pr > DW	**0.9999**		

NOTE: Pr<DW is the p-value for testing positive autocorrelation, and Pr>DW is the p-value for testing negative autocorrelation.

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	**0.6293**	0.3927	1.60	0.1131
x1	1	**2.0000**	0.0000409	48870.3	<.0001

Simple Models: Regression

```
title "Insignificant DW statistic when omitted variable included";
proc autoreg data=ffc.sales2;
       model y = x1 x2/ dwprob;
run;
```

Dependent Variable y

Ordinary Least Squares Estimates

SSE	94.972708	DFE	77
MSE	1.23341	Root MSE	1.11059
SBC	253.90128	AIC	246.7552
Regress R-Square	1.0000	Total R-Square	1.0000
Durbin-Watson	**2.2225**	**Pr < DW**	**0.8295**
Pr > DW	**0.1705**		

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	**0.9042**	0.1253	7.22	<.0001
x1	1	**2.0000**	0.0000130	153761	<.0001
x2	1	**3.0209**	0.1146	26.37	<.0001

Highly Regular Seasonality

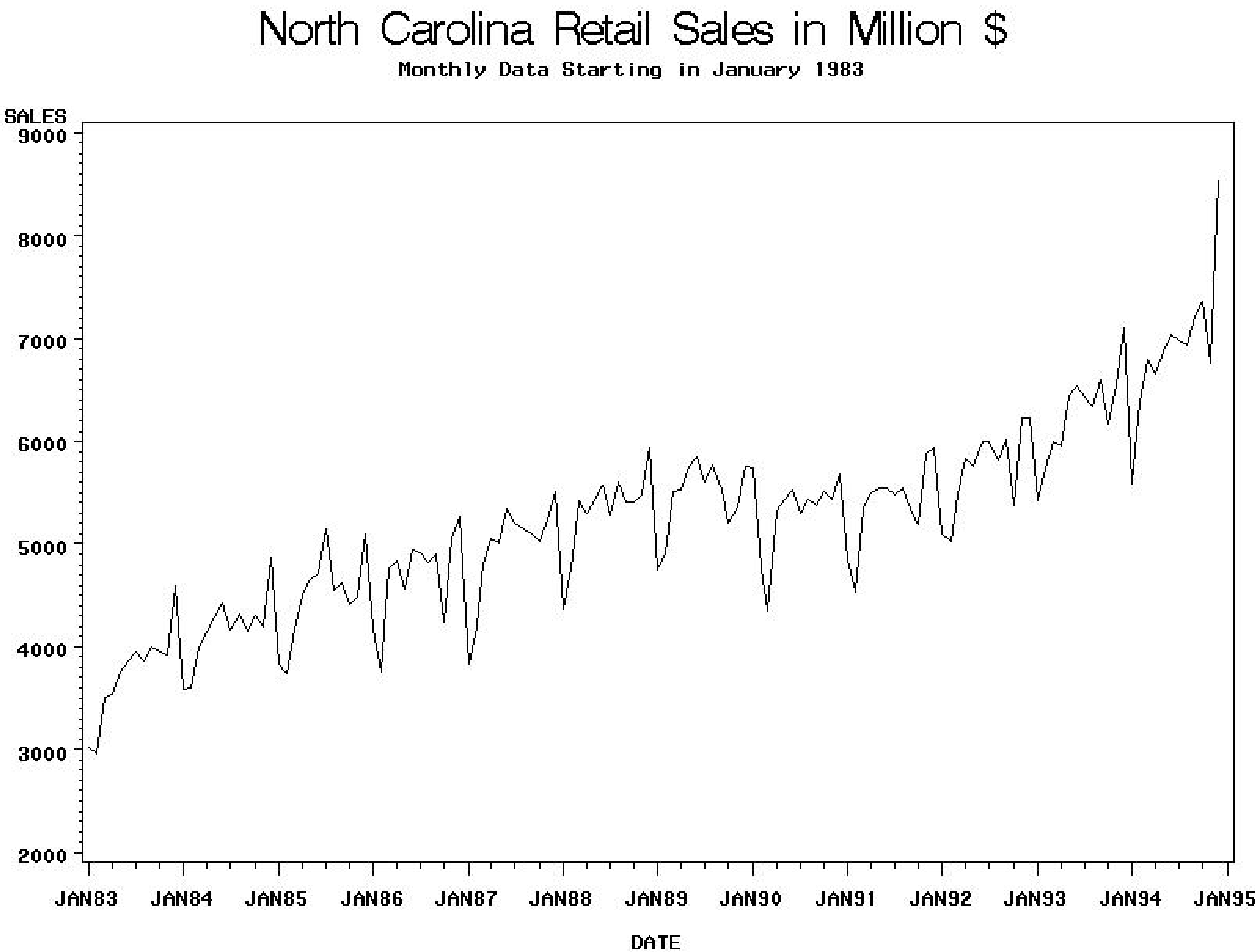
When the data exhibits a regular seasonal pattern, simple dummy variables for each season may be sufficient to model the seasonal pattern.

The example dataset is quarterly retail sales changes for North Carolina.

Since a plot of the data shows a potential quadratic trend, the terms $\boldsymbol{t}$ and $\boldsymbol{t^2}$ (where $t = 1,2,3,\ldots$) will be added to the initial model.

```
 Obs     DATE        CHANGE     S1     S2     S3     T1       T2

  1     83Q1            .       1      0      0      1        1
  2     83Q2       1678.41      0      1      0      2        4
  3     83Q3        633.24      0      0      1      3        9
  4     83Q4        662.35      0      0      0      4       16
  5     84Q1      -1283.59      1      0      0      5       25
  (output omitted)
 47     94Q3        543.61      0      0      1     47     2209
 48     94Q4       1526.95      0      0      0     48     2304
```

Highly Regular Seasonality



Highly Regular Seasonality

```
proc autoreg data=ffc.ncsales;
       model change = t1 t2 s1 s2 s3 / dwprob;
run;
```

```
Dependent Variable    CHANGE

                 Ordinary Least Squares Estimates

SSE                     5290127.6    DFE                          41
MSE                        129028    Root MSE              359.20398
SBC                    703.147758    AIC                   692.046872
Regress R-Square           0.9221    Total R-Square           0.9221
Durbin-Watson              2.3770    Pr < DW                  0.8608
```

No evidence of autocorrelation in this example.

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	679.4273	200.1247	3.40	0.0015
T1	1	-44.9929	16.4428	-2.74	0.0091
T2	1	0.9915	0.3196	3.10	0.0035
S1	1	-1726	150.3312	-11.48	<.0001
S2	1	1504	146.8483	10.24	<.0001
S3	1	-221.2871	146.6958	-1.51	0.1391

Highly Regular Seasonality

With no evidence of autocorrelation in this example, the interpretation is the same as if the estimation was done with **PROC REG**.

Change = 679.4 – 44.99 t + 0.99 t^2 for quarter = 4 (base reference quarter)

Change = 679.4 – 1725.83 - 44.99 t + 0.99 t2 for quarter = 1

The parameters for the seasonal dummies represent shifts in the quadratic function.

To forecast just provide values for the explanatory variables (completely determined in this example) and set the values for the dependent variable, Change, equal to missing.

To save these forecasts, include an **`output`** statement

```
output out=forecasts pm=f_change;
```

The keyword **`pm`** means "predicted mean" – applicable in this example.

Regression with Transformed Data

Often data is analyzed in transformed form, the **log** transformation being one of the most common (*log(y)* is a member of the **Box - Cox** family of power transformations)

If $Y_t = \beta_0 \left(\beta_1^{x_t} \right) \varepsilon_t$ then $\log\left(Y_t\right) = \log\left(\beta_0\right) + \log\left(\beta_1\right) x_t + \log\left(\varepsilon_t\right)$.

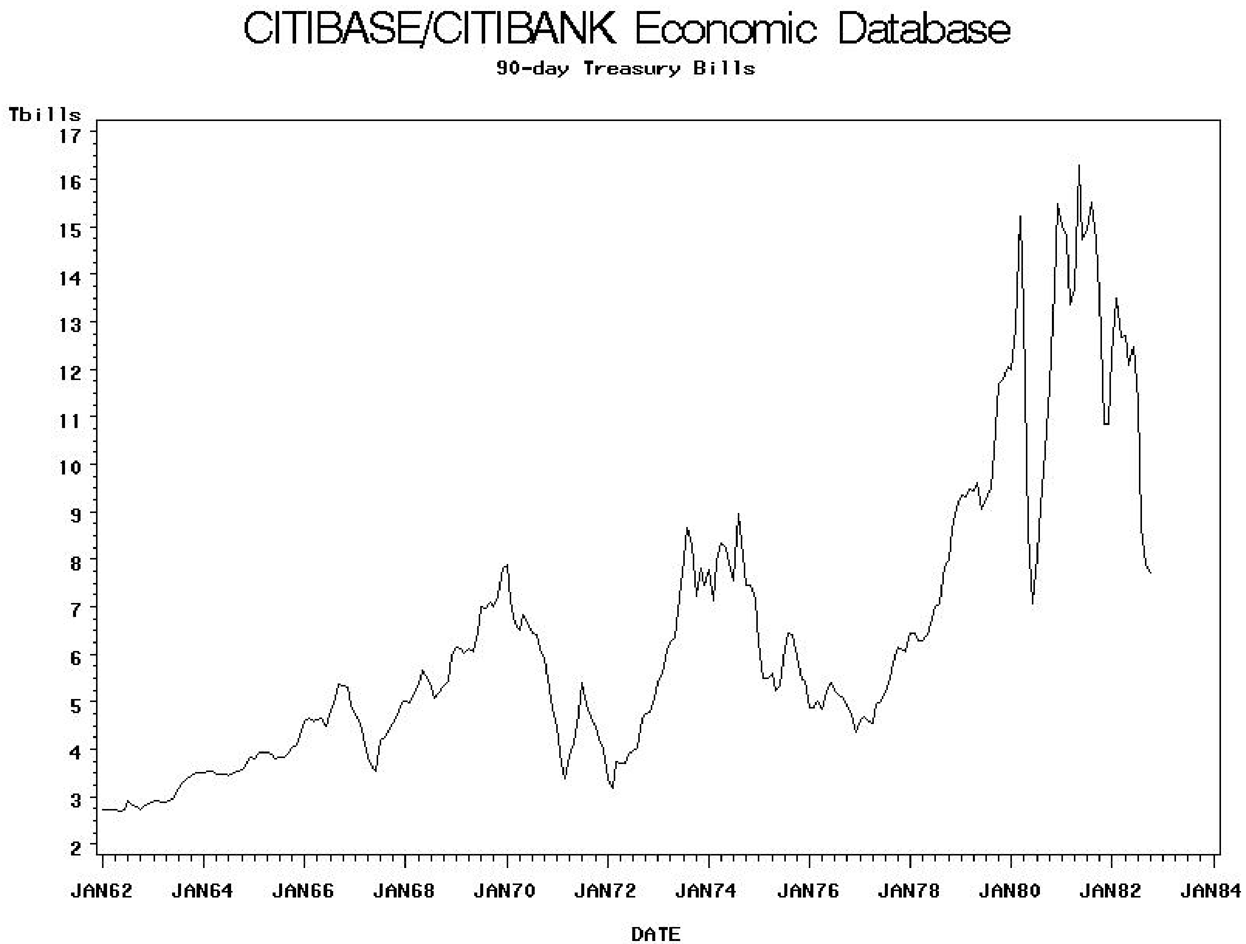
If $\eta_t = \log\left(\varepsilon_t\right)$ satisfies the standard regression assumptions, then we can just apply OLS regression.

Predictions of $\log\left(Y_t\right)$ and confidence limits can be easily transformed into the Y_t-scale.

For example, a 95% confidence interval $-1.13 < \log\left(Y_{n+t}\right) < 2.7 \Rightarrow e^{-1.13} < Y_{n+t} < e^{2.7}$.

Regression with Transformed Data

As an example, we'll look at 90-day treasury bill rates from Jan 1962 – Oct 1982.



Regression with Transformed Data

Note how logging the data has stabilized the variance (somewhat).

Regression with Transformed Data

```
* Add obs to forecast;
data ffc.tbills2;
       set ffc.tbills end=eof;
       time+1;
       output;
       if eof then
              do i = 1 to 24;
                     L_Tbills = .;
                     time+1;
                     date=intnx('month',date,1);
                     output;
              end;
       drop i;
run;
```

Regression with Transformed Data

```
title "CITIBASE/CITIBANK Economic Database";
title2 "Regression with transformed data";
proc reg data=ffc.tbills2;
      id date;
      model L_Tbills = time / dw p cli;
run;
```

```
              CITIBASE/CITIBANK Economic Database
                 Regression with transformed data

                       The REG Procedure
                         Model: MODEL1
                 Dependent Variable: L_Tbills

   Number of Observations Read                         274
   Number of Observations Used                         250
   Number of Observations with Missing Values           24
```

Regression with Transformed Data

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	32.68570	32.68570	540.63	<.0001
Error	248	14.99365	0.06046		
Corrected Total	249	47.67935			

Root MSE	0.24588	R-Square	0.6855
Dependent Mean	1.74783	Adj R-Sq	0.6843
Coeff Var	14.06788		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	1.11904	0.03120	35.87	<.0001
time	1	0.00501	0.00021548	23.25	<.0001

Durbin-Watson D	**0.090**
Number of Observations	250
1st Order Autocorrelation	**0.951**

Regression with Transformed Data

Output Statistics

Obs	DATE	Dependent Variable	Predicted Value	Std Error Mean Predict	95% CL Predict		Residual
1	JAN62	1.0006	1.1240	0.0310	0.6359	1.6122	-0.1234
2	FEB62	1.0043	1.1291	0.0308	0.6410	1.6171	-0.1248
3	MAR62	1.0006	1.1341	0.0306	0.6460	1.6221	-0.1334
4	APR62	1.0043	1.1391	0.0305	0.6511	1.6271	-0.1348
5	MAY62	0.9858	1.1441	0.0303	0.6562	1.6320	-0.1583
6	JUN62	1.0043	1.1491	0.0301	0.6612	1.6370	-0.1448
7	JUL62	1.0716	1.1541	0.0299	0.6663	1.6420	-0.0825
8	AUG62	1.0367	1.1591	0.0297	0.6713	1.6469	-0.1224
9	SEP62	1.0225	1.1641	0.0295	0.6764	1.6519	-0.1417
:							
:							
270	JUN84	.	2.4718	0.0348	1.9827	2.9609	.
271	JUL84	.	2.4768	0.0350	1.9877	2.9660	.
272	AUG84	.	2.4818	0.0352	1.9926	2.9711	.
273	SEP84	.	2.4868	0.0354	1.9976	2.9761	.
274	OCT84	.	2.4919	0.0356	2.0025	2.9812	

Regression with Transformed Data

To get confidence intervals for the original parameters:

for example, $1.119 - (1.96)(0.0312) < \log(\beta_0) < 1.119 + (1.96)(0.0312)$
or: $2.880 < \beta_0 < 3.225$ is a 95% confidence interval for β_0.

Similarly, $1.0046 < \beta_1 < 1.0054$ is a 95% confidence interval for β_1.
Thus, the **growth rate** of Treasury bills is estimated to range from 0.46% to 0.54% per time period.

The forecast for October 1984 can be obtained from:
$2.0025 < 2.4919 = \log(\text{Tbill}_{\text{Oct84}}) < 2.9812 \Rightarrow 7.41 < 12.08 = \text{Tbill}_{\text{Oct84}} < 19.71$

Note that while the original confidence intervals are symmetric, the exponentiated intervals are not, i.e., 12.08 is not the midpoint of [7.41, 19.71].

Regression with Transformed Data

Note $DW = 0.090$. However, since $n = 250$ is beyond the range of the Durbin-Watson tables, a normal approximation for the distribution of $\hat{\rho}$ is used.

Since $\hat{\rho} = 0.951$, $n^{1/2}\hat{\rho}\left(1-\hat{\rho}^2\right)^{1/2} = 48.63$ which is highly significant implying there is non-zero autocorrelation in the residuals.

The next section discusses how to handle autocorrelated residuals.

Suppose $X = \log(y)$ and X is normal with mean μ_X and variance σ_X^2.
Then $y = e^X$ has median e^{μ_X} and mean $\mathrm{e}^{\left(\mu_X + 1/2\sigma_X^2\right)}$. For this reason some authors suggest adding half the error variance to a log scale forecast prior to exponentiation. **Brocklebank & Dickey** prefer to simply exponentiate the log scale forecast and use the median as a more reasonable forecast for the highly skewed distribution of *y*.

Simple Models: AutoRegression*

Autoregression means that the series Y_t is regressed on its own past values, Y_{t-1}, Y_{t-2}, ….

For example,

(1) $Y_t - \mu = \rho(Y_{t-1} - \mu) + \varepsilon_t$ where $\varepsilon_t \sim$ iid $N(0,\sigma^2)$.

ε_t is also known as **white noise**.

*These notes are based on the SAS Books by Users text "**SAS for Forecasting Time Series", 2nd edition by Brocklebank & Dickey, 2003**

Simple Models: AutoRegression

Assuming equation (1) holds for all time periods, we can write $Y_{t\text{-}1} - \mu = \rho\left(Y_{t\text{-}2} - \mu\right) + \varepsilon_{t-1},$ etc and substituting into equation (1) yields $Y_t - \mu = \varepsilon_t + \rho\varepsilon_{t-1} + \rho^2\left(Y_{t\text{-}2} - \mu\right).$

Continuing like this yields:

(2) $\quad Y_t - \mu = \varepsilon_t + \rho\varepsilon_{t-1} + \rho^2\varepsilon_{t-2} + ... + \rho^{t-1}\varepsilon_1 + \rho^t\left(Y_0 - \mu\right)$

Assuming $\left|\rho\right| < 1 \Rightarrow$ the effect of the series values before data was collected (Y_0, for example) is minimal.

It follows from (1) that $\quad Var(Y_t) \;=\; \rho^2 Var\left(Y_{t-1}\right) + \sigma^2.$

Assuming $Var(Y_t)$ is constant, it follows that $Var(Y_t) = \sigma^2 / \left(1 - \rho^2\right)$

Statistical Background

Wold representation of Y_t in equation (1): $Y_t = \mu + \sum_{j=0}^{\infty} \rho^j \varepsilon_{t-j}$

$$\operatorname{cov}\left(Y_t, Y_{t-j}\right) = \gamma\left(j\right) = \rho^{|j|}\sigma^2 / \left(1 - \rho^2\right) = \rho^{|j|} Var(Y_t)$$

$$\therefore \operatorname{corr}\left(Y_t, Y_{t-j}\right) = \rho^{|j|}.$$

We can check if an observed series Y_t satisfies equation (1) by plotting $\operatorname{corr}\left(Y_t, Y_{t-j}\right)$ and seeing if it behaves like $\rho^{|j|}$ for some $-1<\rho<1$.

Forecasting

Given that Y_t satisfies equation (1), so $Y_{n+1} = \mu + \rho\left(Y_n - \mu\right) + \varepsilon_{n+1}$

we can forecast Y_{n+1} as follows: $\hat{Y}_{n+1} = \mu + \rho\left(Y_n - \mu\right)$ where μ and ρ are replaced by estimates $\hat{\mu}$ and $\hat{\rho}$.

The 1-step-ahead forecast error is $Y_{n+1} - \hat{Y}_{n+1} = \varepsilon_{n+1}$

Since $Y_{n+2} = \mu + \rho\left(Y_{n+1} - \mu\right) + \varepsilon_{n+2}$, we have $\hat{Y}_{n+2} = \hat{\mu} + \hat{\rho}\left(\hat{Y}_{n+1} - \hat{\mu}\right) = \hat{\mu} + \hat{\rho}^2\left(Y_n - \hat{\mu}\right)$.

The 2-step-ahead forecast error is $Y_{n+2} - \hat{Y}_{n+2} = \varepsilon_{n+1} + \rho\varepsilon_{n+2}$.

In general, $\hat{Y}_{n+L} = \mu + \rho^L\left(Y_n - \mu\right)$ with the L-step-ahead forecast error:

$Y_{n+L} - \hat{Y}_{n+L} = \varepsilon_{n+L} + \rho\varepsilon_{n+L-1} + \ldots + \rho^{L-1}\varepsilon_{n+1}$.

Forecasting

Forecasting strategy :

1. Examine estimates of $\gamma(j)$ to see if they decrease exponentially.
2. If so, assume model (1) holds and estimate μ and ρ.
3. Calculate the forecast $\hat{Y}_{n+L} = \mu + \rho^L \left(Y_n - \hat{\mu}\right)$ and the forecast error variance $\sigma^2 \left(1 + \rho^2 + \rho^4 ... + \rho^{2L-2}\right)$

Forecasting

From the expansion $Y_{n+L} - \mu = \left\{ \varepsilon_{n+L} + \rho\varepsilon_{n+L-1} + \ldots + \rho^{L-1}\varepsilon_{n+1} \right\} + \rho^{L} \left\{ \varepsilon_{n} + \rho\varepsilon_{n-1} + \ldots \right\}$

and $Y_n - \mu = \varepsilon_n + \rho\varepsilon_{n-1} + \ldots$, it follows that

1. The best (minimum prediction error variance) prediction of Y_{n+L} is $\mu + \rho^{L} \left(Y_n - \mu \right)$

2. The prediction error is $\varepsilon_{n+L} + \rho\varepsilon_{n+L-1} + \ldots + \rho^{L-1}\varepsilon_{n+1}$ with prediction error variance = $\sigma^2 \left[1 + \rho^2 + \rho^4 \ldots + \rho^{2L-2} \right]$

3. The effect of shocks, $\varepsilon_{L\text{-}j}$, that happened a long time ago (j large) has little impact on Y_{n+L} if $|\rho|<1$.

Forecasting

Historic residuals, $Y_t - \hat{Y}_t = \hat{\varepsilon}_t$, can be used to estimate the error variance σ^2, and calculate prediction intervals.

A whole class of models known as **ARMA** models will be shown to have the property that Y_{n+L} can be decomposed into a prediction that is a function of current and past $Y's$ plus a prediction error that is a linear combination of future shocks ($\varepsilon's$).

The coefficients in these expressions are functions of the model parameters, like ρ, which can be estimated from the data.

Forecasting with PROC ARIMA

200 observations are generated such that $Y_t = 100 + 0.8\left(Y_{t\text{-}1} - 100\right) + 20 * \varepsilon_t$ where $\varepsilon_t \sim N\left(0,1\right)$.

Y_t is autoregressive with ρ=0.8, $\sigma^2 = 400$ and $\sqrt{\sigma^2 / \left(1-\rho^2\right)} = 33.3$

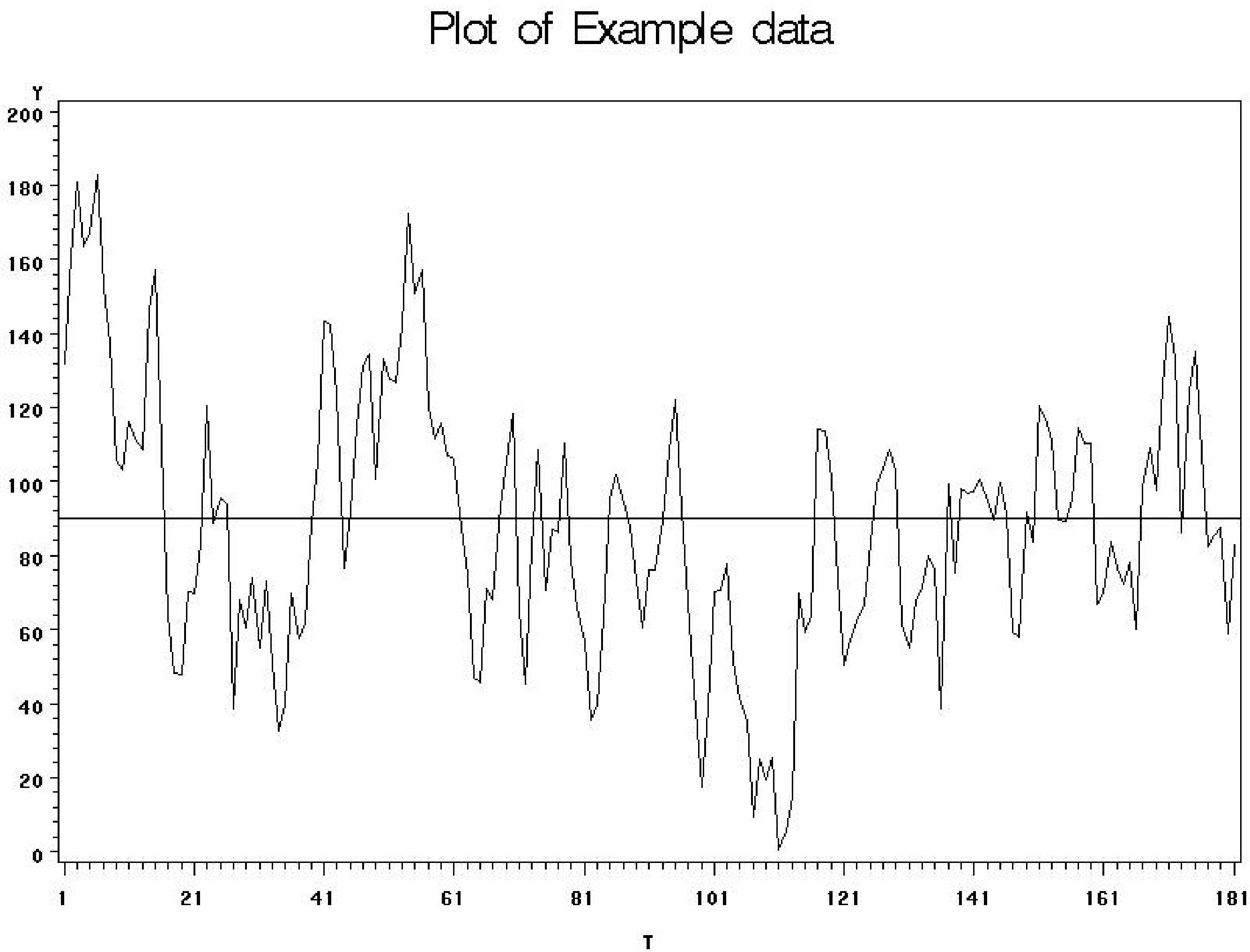
```
The MEANS Procedure

Analysis Variable : Y
```

N	Mean	Std Dev	Minimum	Maximum
200	90.0906434	34.7067449	0.7232617	183.0666353

Note, the theoretical values are Mean = 100 and Std Dev = 33.3

Forecasting with PROC ARIMA

Note how the series tends to linger on one side of its mean value (**positive autocorrelation**)



Forecasting with PROC ARIMA

```
proc arima data=ffc.example;
     identify var=y center;
     estimate p=1 noconstant;
     forecast lead=5;
run;
```

The *center* option says to use the series mean $\bar{Y}_t$ to estimate μ.

Forecasting with PROC ARIMA

```
                                 The ARIMA Procedure

                                Name of Variable = Y

                         Mean of Working Series           0
                         Standard Deviation        34.61987
                         Number of Observations         200


                                  Autocorrelations

Lag    Covariance    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1      Std Error

  0      1198.535        1.00000    |                    |********************|              0
  1       955.584        0.79729    |                .   |****************    |       0.070711
  2       708.551        0.59118    |               .    |************        |       0.106568
  3       524.036        0.43723    |              .     |*********           |       0.121868
  4       402.374        0.33572    |              .     |*******             |       0.129474
  5       308.942        0.25777    |              .     |*****               |       0.133755

    (output omitted)

 20    -82.867591        -.06914    |             .     *|     .              |       0.147720
 21      -140.527        -.11725    |             .    **|     .              |       0.147882
 22      -113.545        -.09474    |             .    **|     .              |       0.148346
 23    -88.683505        -.07399    |             .     *|     .              |       0.148648
 24    -50.803423        -.04239    |             .     *|     .              |       0.148832

                             "." marks two standard errors
```

Forecasting with PROC ARIMA

$\gamma(j)$ is in the column labeled **Covariance** and j in the column labeled **Lag**.

Dividing each covariance $\gamma(j)$ by the variance $\gamma(0)$ gives the values in the **Correlation** column, $\rho(j)$.

Note that the correlations decline at an *exponential rate* of about 0.8.
In fact, the estimate of the 1st-order autocorrelation is $\hat{\rho}(1) = 0.79729$.

Forecasting with PROC ARIMA

Conditional Least Squares Estimation

Parameter	**Estimate**	Standard Error	**t Value**	Approx Pr > |t|	Lag
AR1,1	**0.80575**	0.04261	**18.91**	<.0001	1

Variance Estimate	**430.7275**
Std Error Estimate	**20.75397**
AIC	1781.668
SBC	1784.966
Number of Residuals	200

* AIC and SBC do not include log determinant.

The *estimate* statement produces $\hat{\rho}$=0.80575 with a *t*-value of 18.91 (highly significant).

The estimate of σ^2 is 430.7275 with a standard error estimate of 20.75397 (very close to the theoretical value of 20).

Forecasting with PROC ARIMA

```
                Model for variable Y

  Data have been centered by subtracting the value    90.09064

                No mean term in this model.

                   Autoregressive Factors

                Factor 1:  1 - 0.80575 B**(1)

                  Forecasts for variable Y
```

Obs	Forecast	Std Error	95% Confidence Limits	
201	**130.5036**	**20.7540**	89.8265	171.1806
202	**122.6533**	**26.6528**	70.4149	174.8918
203	116.3280	29.8651	57.7936	174.8625
204	111.2314	31.7772	48.9492	173.5136
205	107.1248	32.9593	42.5257	171.7239

Note that $\hat{Y}_{201} = 90.091 + 0.80575(140.246 - 90.091) = 130.503$ where $\hat{Y}_{200} = 140.246.$

with a **forecast standard error** of $\left(430.73\right)^{1/2} = 20.754.$

$\hat{Y}_{202} = 90.091 + 0.80575^2(140.246 - 90.091) = 122.653$ with a **forecast standard error**

of $\left(430.73\left(1 + 0.80575^2\right)\right)^{1/2} = 26.6528.$

Forecasting with PROC ARIMA

The coefficients are estimated by the **least squares (LS) method**, i.e., in this case

$$0.80575 = \sum_{t=2}^{200}\left(Y_t - \overline{Y}\right)\left(Y_{t\text{-}1} - \overline{Y}\right) / \sum_{t=2}^{200}\left(Y_{t\text{-}1} - \overline{Y}\right)^2$$

Alternative estimation methods include **maximum - likelihood (ML)** and **unconditional least squares (ULS).** These methods are discussed below in the context of the AR(1) model.

ML : Idea is to find values of the parameters that maximize the joint probability density function viewed as a function of the unknown parameters. The **joint pdf** is the product of the pdf for each observation if the random variables in the joint pdf are independent. While the series $\{Y_t\}_{t=1}^{T}$ are not independent, the error terms $\{\varepsilon_t\}_{t=1}^{T}$ are assumed to be independent.

Forecasting with PROC ARIMA

ML : The first observation is treated differently than the rest. $Y_1 \sim N\left(\mu, \sigma^2 / \left(1-\rho^2\right)\right)$

and its pdf is $\dfrac{\sqrt{1-\rho^2}}{\sqrt{2\pi\sigma^2}} \exp\left(-\dfrac{\left(Y_1-\mu\right)^2\left(1-\rho^2\right)}{2\sigma^2}\right)$.

For observations 2 to 200, we use the independent variables $\varepsilon_t = Y_t - \rho Y_{t-1}$ and

$\varepsilon_t \sim iid\ N\left(\mu - \rho\mu, \sigma^2\right)$ with pdf $\dfrac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\dfrac{\left[\left(Y_t-\mu\right)-\rho\left(Y_{t-1}-\mu\right)\right]^2}{2\sigma^2}\right)$

and since $Y_1, \varepsilon_2, \varepsilon_3, \ldots, \varepsilon_n$ are independent, their joint pdf is:

$$\frac{\sqrt{1-\rho^2}}{\left(\sqrt{2\pi\sigma^2}\right)^n} \exp\left(\frac{\left(1-\rho^2\right)\left(Y_1-\mu\right)^2 + \left[\left(Y_2-\mu\right)-\rho\left(Y_1-\mu\right)\right]^2 + \ldots + \left[\left(Y_n-\mu\right)-\rho\left(Y_{n-1}-\mu\right)\right]^2}{2\sigma^2}\right)$$

Forecasting with PROC ARIMA

ML : Substituting the observed data y_i for Y_i and maximizing the resulting function whose arguments are μ, ρ, and σ^2 gives the maximum likelihood estimates $\hat{\mu}$, $\hat{\rho}$, and $\hat{\sigma}^2$ $\hat{\sigma}^2 = USS / n$ where USS (**unconditional sum of squares**) equals

$$USS = \left(1-\hat{\rho}^2\right)\left(y_1-\hat{\mu}\right)^2+\left[\left(y_2-\hat{\mu}\right)-\hat{\rho}\left(y_1-\hat{\mu}\right)\right]^2+\ldots+\left[\left(y_n-\hat{\mu}\right)-\hat{\rho}\left(y_{n-1}-\hat{\mu}\right)\right]^2$$

Substituting the above expression for σ^2 into the likelihood function on the previous page results in a function with only two unknowns, μ and ρ, and is called the **concentrated likelihood**.

CLS : The **conditional least squares** method minimizes a slightly different objective function:

$$\left(Y_1-\mu\right)^2+\left[\left(Y_2-\mu\right)-\rho\left(Y_1-\mu\right)\right]^2+\ldots+\left[\left(Y_n-\mu\right)-\rho\left(Y_{n-1}-\mu\right)\right]^2$$

Substituting $\overline{Y}$ for μ leaves only ρ to be estimated.

Forecasting with PROC ARIMA

`Method = CLS` is the default. Use **`Method = ML`** for maximum likelihood and **`Method = ULS`** for unconditional least squares.

Different estimates are produced with each method. For example, trying each method on the current model:

```
title "Conditional Least Squares (default)";
proc arima data=ffc.example;
      identify var=y center noprint;
      estimate p=1 noconstant method=cls printall;
run;
quit;
```

The option **`center`** subtracts the sample mean from *Y* while the option **`noconstant`** suppresses estimation of a constant term.

The option **`printall`** shows the iteration history and illustrates that different objective functions are used with each method.

Forecasting with PROC ARIMA

Conditional Least Squares Estimation

Iteration	SSE	AR1,1	Lambda	R Crit
0	85732	0.79729	0.00001	1
1	85715	0.80575	1E-6	0.014065
2	85715	0.80575	1E-7	1.403E-7

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
AR1,1	**0.80575**	0.04261	18.91	<.0001	1

Unconditional Least Squares Estimation

Iteration	SSE	AR1,1	Lambda	R Crit
0	84572	0.80575	0.00001	1
1	84564	0.81166	1E-6	0.009962
2	84564	0.81178	1E-7	0.000209

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
AR1,1	**0.81178**	0.04203	19.32	<.0001	1

Forecasting with PROC ARIMA

Maximum Likelihood Estimation

Iter	Loglike	AR1,1	Lambda	R Crit
0	-889.01577	0.80575	0.00001	1
1	-889.01479	0.80761	1E-6	0.003131
2	-889.01479	0.80760	1E-7	9.101E-6

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
AR1,1	**0.80760**	0.04192	19.26	<.0001	1

Both **`ULS`** and **`ML`** do a preliminary **`CLS`** estimation to get starting values.

Forecasting with PROC ARIMA

The previous model is autoregressive of order 1, i.e., AR(1). Higher order models are common.

For example, an AR(2) model would be: $Y_t - \mu = \alpha_1 \left(Y_{t\text{-}1} - \mu\right) + \alpha_2 \left(Y_{t\text{-}2} - \mu\right) + \varepsilon_t$

Examination of the autocorrelation plot of a time series helps to determine the order of an autoregressive process.

```
proc arima data=mydata;
      identify var = Y;
run;
```

The **backshift notation** is very useful in manipulating AR models.

Backshift Notation B for Time Series

The backshift operator B is defined as follows:

$$B\left(Y_t\right) = Y_{t\text{-}1},\; B^2\left(Y_t\right) = B\left(Y_{t-1}\right) = Y_{t\text{-}2}, \quad \text{etc.}$$

So $B^n\left(Y_t\right) = Y_{t\text{-}n}$. The AR(1) process $Y_t = 0.8Y_{t-1} + \varepsilon_t$ can be written $\left(1\text{-}0.8B\right)Y_t = \varepsilon_t$ or $Y_t = \left(1\text{-}0.8B\right)^{-1}\varepsilon_t$.

Using the result $\left(1 - x\right)^{-1} = 1 + x + x^2 + x^3 + \dots$ for $|x| < 1$, we can write

$$Y_t = \left(1 + 0.8B + 0.8^2 B^2 + 0.8^3 B^3 + \dots\right)\varepsilon_t = \varepsilon_t + 0.8\varepsilon_{t-1} + 0.64\varepsilon_{t-2} + \dots$$

Backshift Notation B for Time Series

Using the algebra of partial fraction expansions, it can be shown that higher order AR models can also be expressed as MA(∞) models.

For example, consider the AR(2) model: $Y_t = 1.70Y_{t\text{-}1} - 0.72Y_{t\text{-}2} + \varepsilon_t$

So $Y_t = \dfrac{1}{\left(1+1.70B-0.72B^2\right)}\varepsilon_t = \left(\dfrac{9}{\left(1-0.9B\right)} - \dfrac{8}{\left(1-0.8B\right)}\right)\varepsilon_t = \sum_{j=1}^{\infty} W_j \varepsilon_{t-j}$

where $W_j = 9\left(0.9^j\right) - 8\left(0.8^j\right)$.

Yule-Walker Equations for Covariances

The **Yule - Walker equations** are a set of recursive equations that can be used to solve for either the autocovariances, $\gamma(j)$, or autocorrelations, $\rho(j)$.

The principle can be illustrated with an AR(2) model (without loss of generality, we'll assume the data has been centered): (1) $Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \varepsilon_t$

The basic idea is to multiply both sides of equation (1) by Y_{t-j}, take expected values, and note that $\mathrm{E}\left(Y_{t-j}\varepsilon_t\right)=0$ for $j \geq 1$.

$$j=0\text{: } \mathrm{E}\left(Y_tY_t\right) = \alpha_1 E\left(Y_tY_{t-1}\right) + \alpha_2 E\left(Y_tY_{t-2}\right) + E\left(Y_t\varepsilon_t\right) \Rightarrow \gamma(0) = \alpha_1\gamma(0) + \alpha_2\gamma(1) + \sigma^2$$
$$j=1\text{: } \mathrm{E}\left(Y_tY_{t-1}\right) = \alpha_1 E\left(Y_{t-1}Y_{t-1}\right) + \alpha_2 E\left(Y_{t-2}Y_{t-1}\right) + E\left(Y_{t-1}\varepsilon_t\right) \Rightarrow \gamma(1) = \alpha_1\gamma(0) + \alpha_2\gamma(1)$$
$$j \geq 2\text{: } \mathrm{E}\left(Y_tY_{t-j}\right) = \alpha_1 E\left(Y_{t-1}Y_{t-j}\right) + \alpha_2 E\left(Y_{t-2}Y_{t-j}\right) + E\left(Y_{t-j}\varepsilon_t\right) \Rightarrow \gamma(j) = \alpha_1\gamma(j-1) + \alpha_2\gamma(j-2)$$

Given σ^2 we can then solve the three equations in three unknowns $\gamma(0), \gamma(1)$ and $\gamma(2)$:

$$\gamma(0) = \alpha_1\gamma(1) + \alpha_2\gamma(2) + \sigma^2$$
$$\gamma(1) = \alpha_1\gamma(0) + \alpha_2\gamma(1)$$
$$\gamma(2) = \alpha_1\gamma(1) + \alpha_2\gamma(2)$$

Since $\rho(j) = \gamma(j)/\gamma(0)$, we can also use the Yule-Walker equations to solve for the $\rho(j)$.

Yule-Walker Equations for Covariances

The general AR(p) model can be written:

$$Y_t - \mu = \alpha_1\left(Y_{t\text{-}1} - \mu\right) + \alpha_2\left(Y_{t\text{-}2} - \mu\right) + \ldots + \alpha_p\left(Y_{t\text{-}p} - \mu\right) + \varepsilon_t$$

or equivalently: $\left(1 - \alpha_1 B - \alpha_2 B^2 - \ldots - \alpha_p B^p\right)\left(Y_t - \mu\right) = \varepsilon_t$

The polynomial $\left(1 - \alpha_1 x - \alpha_2 x^2 - \ldots - \alpha_p x^p\right)$ is called the **characteristic polynomial**.

If the roots of the characteristic polynomial are inside the unit circle (i.e., have modulus less than 1), then the infinite series $\left(1 - \alpha_1 x - \alpha_2 x^2 - \ldots - \alpha_p x^p\right)^{-1}$ will converge and the coefficients in the expansion $\left(1 - \alpha_1 B - \alpha_2 B^2 - \ldots - \alpha_p B^p\right)^{-1} \varepsilon_t$ will (eventually) decrease to zero.

In this case, the AR(p) series is said to be **stationary**.

Fitting an AR Model in PROC REG

Autoregressive models, unlike Moving Average models discussed in the next section, can be estimated by simple OLS methods.

One disadvantage of using **PROC REG** is that lags have to be created by the user before doing the estimation.

The example fits a fourth order autoregressive model to the stocks of silver at the New York Commodity Exchange in 1000 troy weight ounces from December 1976 through May 1981.

Note how the following code creates lagged values without using the **`lag`** function.

Fitting an AR Model in PROC REG

```
DATA ffc.SILVER;
   TITLE 'MONTH END STOCKS OF SILVER';
   INPUT SILVER @@;
   T=_N_;
   RETAIN DATE '01DEC76'D LSILVER1-LSILVER4;
   DATE=INTNX('MONTH',DATE,1);
   FORMAT DATE MONYY.;
   OUTPUT;
   LSILVER4=LSILVER3;
   LSILVER3=LSILVER2;
   LSILVER2=LSILVER1;
   LSILVER1=SILVER;
   CARDS;
 846 827 799 768 719 652 580 546 500 493 530 548 565 572 632 645 674
 693 706 661 648 604 647 684 700 723 741 734 708 728 737 729 678 651
 627 582 521 519 496 501 555 541 485 476 515 606 694 788 761 794 836
 846
;
RUN;
```

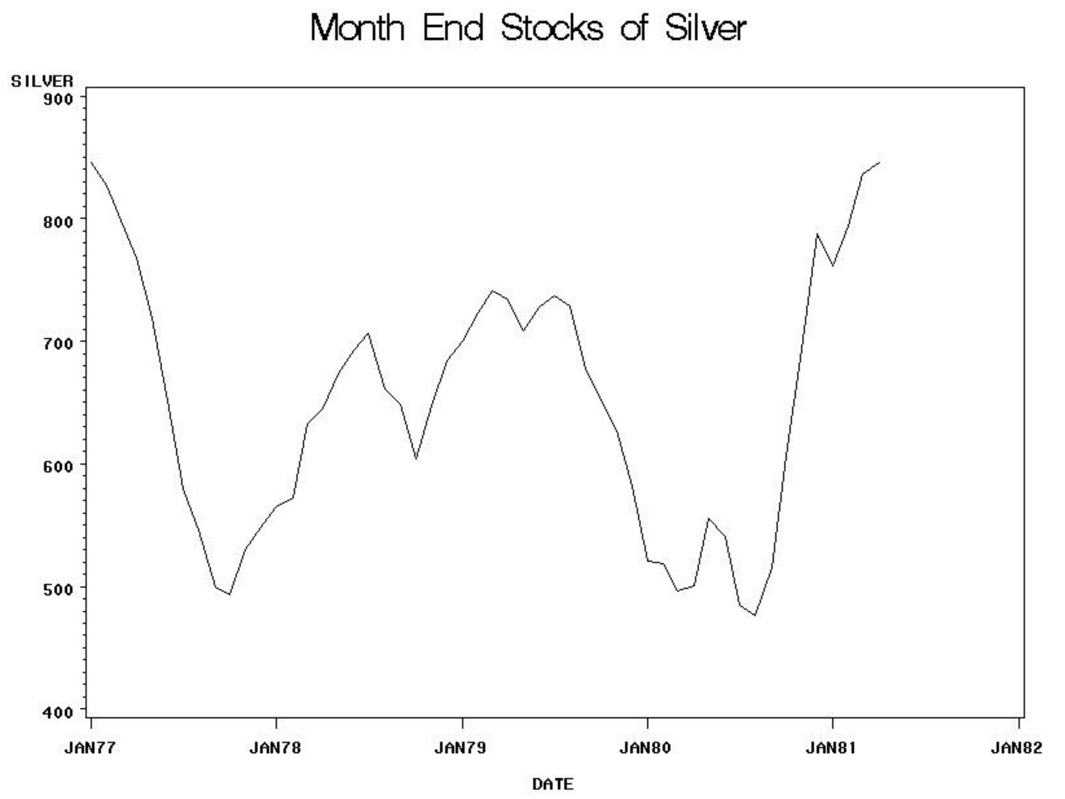
Fitting an AR Model in PROC REG

```
proc print data=ffc.silver(obs=10);
run;
```

Obs	SILVER	T	DATE	LSILVER1	LSILVER2	LSILVER3	LSILVER4
1	846	1	JAN77	.	.	.	.
2	827	2	FEB77	846	.	.	.
3	799	3	MAR77	827	846	.	.
4	768	4	APR77	799	827	846	.
5	719	5	MAY77	768	799	827	846
6	652	6	JUN77	719	768	799	827
7	580	7	JUL77	652	719	768	799
8	546	8	AUG77	580	652	719	768
9	500	9	SEP77	546	580	652	719
10	493	10	OCT77	500	546	580	652

Fitting an AR Model in PROC REG

```
title "Month End Stocks of Silver";
symbol1 i=join;
axis1 order=('01jan77'd to '01jan82'd by year);
axis2 order=(400 to 900 by 100);
proc gplot data=ffc.silver;
        plot silver*date / haxis = axis1 vaxis = axis2 ;
run;
```



Fitting an AR Model in PROC REG

1st fit an **AR(4)** model:

```
proc reg data=ffc.silver;
      model silver = lsilver1 - lsilver4 / ss1;
run;
```

```
Dependent Variable: SILVER

Number of Observations Read                         52
Number of Observations Used                         48
Number of Observations with Missing Values           4
```

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Type I SS
Intercept	1	102.84126	37.85904	2.72	0.0095	19470543
LSILVER1	1	1.38589	0.15156	9.14	<.0001	387295
LSILVER2	1	-0.44231	0.26078	-1.70	0.0971	28472
LSILVER3	**1**	**0.00921**	**0.26137**	**0.04**	**0.9720**	**1061.93530**
LSILVER4	**1**	**-0.11236**	**0.15185**	**-0.74**	**0.4633**	**599.56290**

The partial output above shows that lags 3 and 4 are insignificant. This can be verified with a F-test.

Fitting an AR Model in PROC REG

Refit the model as an **AR(2)** model:

```
proc reg data=ffc.silver;
      model silver = lsilver1 - lsilver2 / ss1;
run;
```

```
                         Dependent Variable: SILVER

             Number of Observations Read                        52
             Number of Observations Used                        50
             Number of Observations with Missing Values          2
```

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	**t Value**	**Pr > |t|**	Type I SS
Intercept	1	77.95372	30.21038	**2.58**	**0.0131**	20657021
LSILVER1	1	1.49087	0.11589	**12.86**	**<.0001**	428317
LSILVER2	1	-0.61144	0.11543	**-5.30**	**<.0001**	29136

Lags one and two are highly significant.

Fitting an AR Model in PROC REG

The final model is: $Y_t = 77.9537 + 1.4909Y_{t\text{-}1} - 0.6114Y_{t\text{-}2} + \varepsilon_t$

or equivalently, $Y_t - 647 = 1.4909\left(Y_{t-1} - 647\right) - 0.6114\left(Y_{t-2} - 647\right) + \varepsilon_t$

The associated characteristic polynomial is $f(x) = 0.61x^2 - 1.49x + 1.$

Note that $f(1) = 0.12 \approx 0$, so the data is close to being non-stationary.