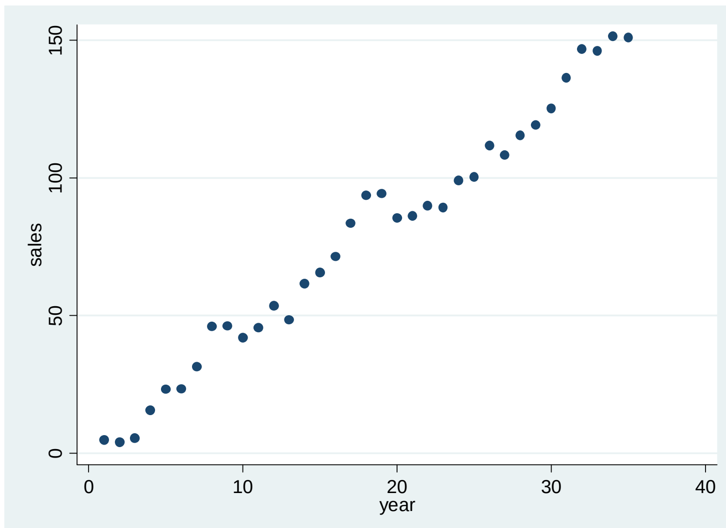
MA397 – Regression-Based Methods for Time Series

Moving Average Smoother

For this exercise we will be using the *sales.dta* dataset found on the course webpage at http://www.colby.edu/personal/l/lobrien/ma397.html.

Recall that the data consist of sales information over a 35-year period. The plot of the data are below:



We see a clear seasonal trend. There may be a seasonal component as well, but it is less clear. We first fit the model using OLS and calculate the Durbin-Watson statistic to see if there is residual autocorrelation (make sure to set the data as time series first).

```
. tsset year
        time variable:  year, 1 to 35
                delta:  1 unit

. regress sales year

      Source |       SS       df       MS              Number of obs =      35
-------------+------------------------------           F(  1,    33) = 1615.72
       Model |  65875.2068     1  65875.2068           Prob > F      =  0.0000
    Residual |  1345.45362    33  40.7713217           R-squared     =  0.9800
-------------+------------------------------           Adj R-squared =  0.9794
       Total |  67220.6604    34  1977.07825           Root MSE      =  6.3852

------------------------------------------------------------------------------
       sales |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        year |    4.29563   .1068669    40.20   0.000     4.078208    4.513053
       _cons |   .4015129   2.205708     0.18   0.857    -4.086034     4.88906
------------------------------------------------------------------------------

. estat dwatson

Durbin-Watson d-statistic(  2,    35) =  .8207266
```

We see that the Durbin-Watson statistic is 0.8207266, which is less than the lower cutoff of 1.40 indicating positive autocorrelation. So let's fit a first-order autoregressive model to the data.

To fit autoregressive, moving average, or autoregressive-moving average models in Stata go to **Statistics > Times series > ARIMA and ARMAX models**. A wide variety of models can be fit via this option. For a simple first-order AR model, enter the dependent variable (sales) and the independent variables (year) in the appropriate boxes. Enter the autoregressive order of 1 (leave the difference and moving average orders at 0. Click okay to obtain:

```
. arima sales year, arima(1,0,0)

(setting optimization to BHHH)
Iteration 0:   log likelihood = -106.26769
Iteration 1:   log likelihood =  -106.2573
Iteration 2:   log likelihood = -106.25527
Iteration 3:   log likelihood = -106.25519
Iteration 4:   log likelihood = -106.25501
(switching optimization to BFGS)
Iteration 5:   log likelihood = -106.25321
Iteration 6:   log likelihood = -106.25321

ARIMA regression

Sample:  1 - 35                                 Number of obs      =        35
                                                Wald chi2(2)       =    346.62
Log likelihood = -106.2532                      Prob > chi2        =    0.0000

------------------------------------------------------------------------------
             |                 OPG
       sales |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
sales        |
        year |   4.295973   .2378623    18.06   0.000     3.829772    4.762175
       _cons |   .4060005   4.988835     0.08   0.935    -9.371937    10.18394
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .5735942   .1496151     3.83   0.000     .2803541    .8668344
-------------+----------------------------------------------------------------
      /sigma |   5.008673   .7330814     6.83   0.000      3.57186    6.445486
------------------------------------------------------------------------------
```

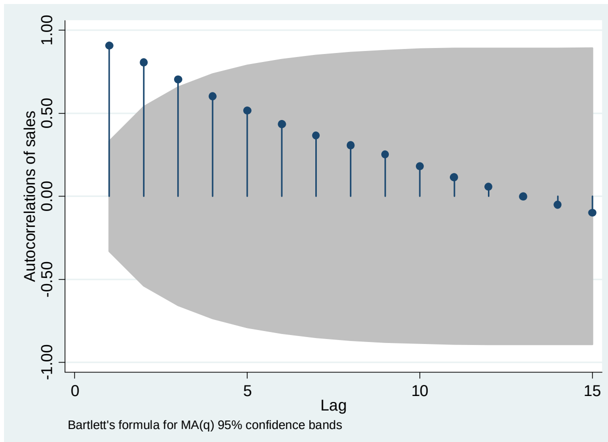
Note the difference in the standard errors of the regression coefficients (they were underestimated with OLS), but that the OLS estimates were unbiased. The coefficient for the ARMA output is phi (0.574 in this case). The "sigma" value is the root MSE for the model. The other statistics given in the textbook printout are not available since the model was estimated using maximum likelihood rather than the modified OLS procedure.

Stata has many interesting commands for time series data. For example, to calculate the autocorrelations using the command:

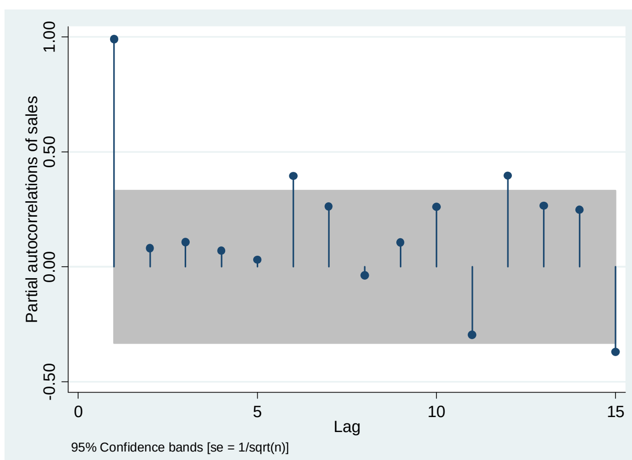
```
. corrgram sales

                                          -1       0       1 -1       0       1
 LAG       AC       PAC      Q     Prob>Q  [Autocorrelation]  [Partial Autocor]
-------------------------------------------------------------------------------
1        0.9077   0.9914   31.384  0.0000          |-------           |-------
2        0.8058   0.0809   56.867  0.0000          |------            |
3        0.7036   0.1081   76.901  0.0000          |-----             |
4        0.6032   0.0707   92.103  0.0000          |----              |
5        0.5143   0.0313   103.52  0.0000          |----              |
6        0.4351   0.3940   111.97  0.0000          |---               |---
```

```
7        0.3656   0.2616   118.16  0.0000          |--                  |--
8        0.3072  -0.0359   122.68  0.0000          |--                  |
9        0.2508   0.1065   125.82  0.0000          |--                  |
10       0.1805   0.2607    127.5  0.0000          |-                   |--
11       0.1152  -0.2948   128.22  0.0000          |                  --|
12       0.0566   0.3962    128.4  0.0000          |                    |---
13      -0.0011   0.2653    128.4  0.0000          |                    |--
14      -0.0504   0.2493   128.56  0.0000          |                    |-
15      -0.1000  -0.3711   129.21  0.0000          |                  --|
```

We can see this graphically with a correlogram by giving the command: `ac sales`



A partial autocorrelation plot can be obtained by giving the command: `pac sales`



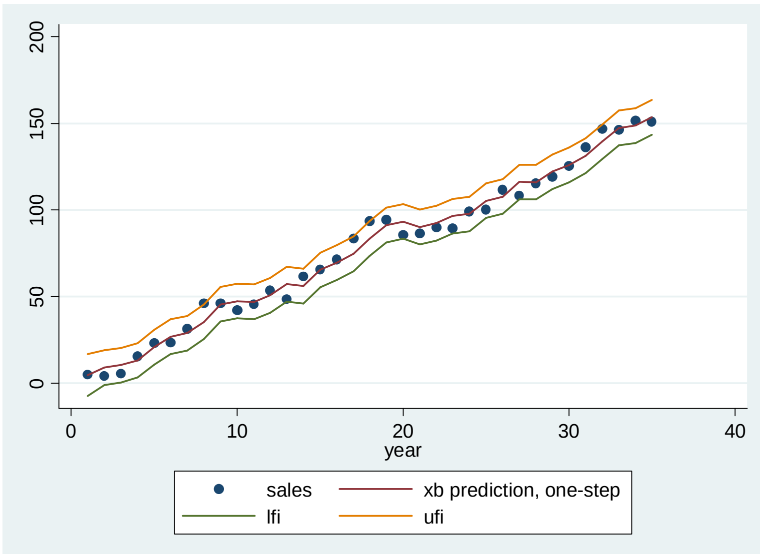
From these plots it looks as if our first-order AR model is a good fit. We may now want to forecast using it. Forecasting in Stata is straightforward after fitting a time series model. However, forecasting beyond the end of the x-data range is not possible. To get the point estimates simply use the “predict” command.

```
. predict yhat
(option xb assumed; predicted values)
```

If we want to generate confidence intervals around these, we need to first estimate the MSE of the observations.

```
. predict mse, mse

. gen lfi = yhat-2*sqrt(mse)

. gen ufi=yhat+2*sqrt(mse)
```

The variables *lfi* and *ufi* now contain the confidence bands around the forecasted values.



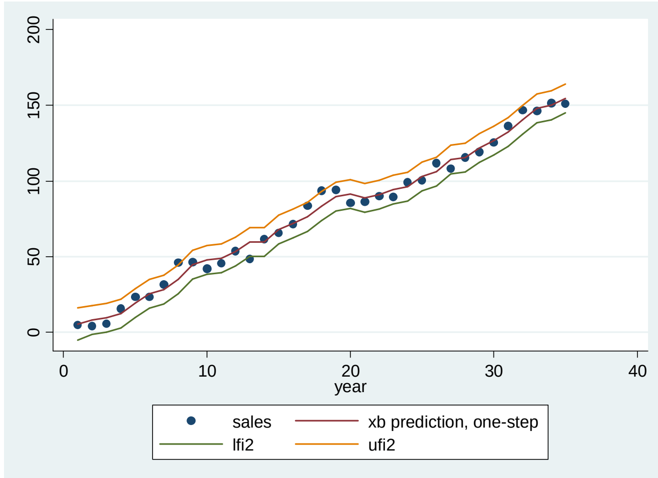
We can try to estimate seasonal variation noticing that there seems to be a 10-year cycle in the data. Try to fit sine and cosine functions and rerun the model:

```
. gen sine=sin((_pi/10)*year)

. gen cosine=cos((_pi/10)*year)

. arima sales year sine cosine, arima(1,0,0)

(setting optimization to BHHH)
Iteration 0:   log likelihood = -104.43414
Iteration 1:   log likelihood = -104.42012
Iteration 2:   log likelihood =  -104.4149
Iteration 3:   log likelihood = -104.41449
Iteration 4:   log likelihood = -104.41442
(switching optimization to BFGS)
Iteration 5:   log likelihood = -104.41442
```

```
ARIMA regression

Sample:  1 - 35                                 Number of obs      =        35
                                                Wald chi2(4)       =    581.77
Log likelihood = -104.4144                      Prob > chi2        =    0.0000

------------------------------------------------------------------------------
             |                 OPG
       sales |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
sales        |
        year |   4.195335   .1850473    22.67   0.000     3.832649    4.558021
        sine |  -4.362489   2.234392    -1.95   0.051    -8.741816    .0168381
      cosine |   .1228667   1.974232     0.06   0.950    -3.746556    3.992289
       _cons |   2.468488   3.579696     0.69   0.490    -4.547589    9.484564
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .4631711    .189805     2.44   0.015     .0911601    .8351822
-------------+----------------------------------------------------------------
      /sigma |   4.763073   .6386057     7.46   0.000     3.511429    6.014717
------------------------------------------------------------------------------

. arima sales year sine, arima(1,0,0)

(setting optimization to BHHH)
Iteration 0:   log likelihood = -104.43257
Iteration 1:   log likelihood =  -104.4219
Iteration 2:   log likelihood = -104.41995
Iteration 3:   log likelihood = -104.41949
Iteration 4:   log likelihood = -104.41902
(switching optimization to BFGS)
Iteration 5:   log likelihood =  -104.4162
Iteration 6:   log likelihood = -104.41619
Iteration 7:   log likelihood = -104.41618

ARIMA regression

Sample:  1 - 35                                 Number of obs      =        35
                                                Wald chi2(3)       =    575.80
Log likelihood = -104.4162                      Prob > chi2        =    0.0000

------------------------------------------------------------------------------
             |                 OPG
       sales |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
sales        |
        year |   4.193308   .1850032    22.67   0.000     3.830708    4.555907
        sine |  -4.363483   2.224331    -1.96   0.050    -8.723091   -.0038739
       _cons |   2.493923   3.523166     0.71   0.479    -4.411356    9.399202
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |    .463647   .1857311     2.50   0.013     .0996207    .8276732
-------------+----------------------------------------------------------------
      /sigma |   4.763267    .636693     7.48   0.000     3.515372    6.011163
------------------------------------------------------------------------------
```

We see that the sine function may provide better forecasts. We can compare the forecasts by generating new values and examining the plot,

```
. predict yhat2
(option xb assumed; predicted values)

. predict mse2, mse

. gen lfi2 = yhat2-2*sqrt(mse2)

. gen ufi2=yhat2+2*sqrt(mse2)
```



The smaller root MSE for the seasonally-adjusted model indicates a slightly better fit, but the difference appear to be small.