Estimating a VAR

The vector autoregressive model (VAR) is actually simpler to estimate than the VEC model. It is used when there is no cointegration among the variables and it is estimated using time series that have been transformed to their stationary values.

In the example from your book, we have macroeconomic data log of real personal disposable income (denoted as Y) and log of real personal consumption expenditure (denoted as C) for the U.S. economy over the period 1960:1 to 2009:4 that are found in the fred.dta dataset. As in the previous example, the first step is to determine whether the variables are stationary. If they are not, then difference them, checking to make sure that the differences are stationary (i.e., the levels are integrated). Next, test for cointegration. If they are cointegrated, estimate the VEC model. If not, use the differences and lagged differences to estimate a VAR model.

Load the data. In this exercise we’ll be using the fred.dta data.

```
use fred, clear
```

The data are quarterly and begin in 1960q1 and extend to 2009q4. Just as we did in the example above, sequences of quarterly dates:

```
gen date = q(1960q1) + _n - 1
format %tq date
tsset date
```

The first step is to plot the series in order to identify whether constants or trends should be included in the tests of nonstationarity. Both the levels and differences are plotted.

```
tsline lc ly, legend(lab (1 "ln(RPCE)") lab(2 "ln(RPDI)")) ///
   name(l1, replace)
tsline D.lc D.ly, legend(lab (1 "ln(RPCE)") lab(2 "ln(RPDI)")) ///
   name(d1, replace)
```

```

![Graph showing plots of ln(RPCE) and ln(RPDI)](image)

```
The levels series appear to be trending together. The differences show no obvious trend, but the mean of the series appears to be greater than zero, suggesting that a constant be included in the ADF regressions.

The other decision that needs to be made is the number of lagged differences to include in the augmented Dickey-Fuller regressions. The principle to follow is to include just enough so that the residuals of the ADF regression are not autocorrelated. So, start out with a basic regression that contains no lags, estimate the DF regression, then use the LM test discussed earlier to determine whether the residuals are autocorrelated. Add enough lags to eliminate the autocorrelation among residuals. If this strategy is pursued in Stata, then the ADF regressions will have to be explicitly estimated; the `estat bgodfrey` command will not be based on the proper regression if issued after `dfuller`.

The regressions for the ADF tests are

```
qui reg D.lc L.lc L(1/1).D.lc
estat bgodfrey, lags(1 2 3)
qui reg D.lc L.lc L(1/2).D.lc
estat bgodfrey, lags(1 2 3)
qui reg D.lc L.lc L(1/3).D.lc
estat bgodfrey, lags(1 2 3)
```

The test results for the last two regressions appear below.
It is clear that the residuals of the ADF(2) are autocorrelated and those of the ADF(3) are not. The resulting ADF statistic is obtained using:

```
dfuller lc, lags(3)
```

where the indicated number of lags is used.

```
dfuller lc, lags(3)
```

Breusch-Godfrey LM test for autocorrelation

<table>
<thead>
<tr>
<th>lags((p))</th>
<th>chi 2</th>
<th>df</th>
<th>Prob &gt; chi 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.451</td>
<td>1</td>
<td>0.0196</td>
</tr>
<tr>
<td>2</td>
<td>5.518</td>
<td>2</td>
<td>0.0634</td>
</tr>
<tr>
<td>3</td>
<td>7.795</td>
<td>3</td>
<td>0.0504</td>
</tr>
</tbody>
</table>

HD: no serial correlation

```
qui reg L(0/3).D.lc L.lc
di "Lags = 3"
'estat bgodfrey, lags(1 2 3)
```

Breusch-Godfrey LM test for autocorrelation

<table>
<thead>
<tr>
<th>lags((p))</th>
<th>chi 2</th>
<th>df</th>
<th>Prob &gt; chi 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.056</td>
<td>1</td>
<td>0.8122</td>
</tr>
<tr>
<td>2</td>
<td>2.311</td>
<td>2</td>
<td>0.3148</td>
</tr>
<tr>
<td>3</td>
<td>3.499</td>
<td>3</td>
<td>0.3209</td>
</tr>
</tbody>
</table>

HD: no serial correlation

It is clear that the residuals of the ADF(2) are autocorrelated and those of the ADF(3) are not. The resulting ADF statistic is obtained using:

```
dfuller lc, lags(3)
```

where the indicated number of lags is used.

```
dfuller lc, lags(3)
```

Augmented Dickey-Fuller test for unit root

```
forvalues p = 1/3 {
    qui reg L(0/`p').D.ly L.ly
    di "Lags = `p"
    estat bgodfrey, lags(1 2 3)
}
```

It is clear that the residuals of the ADF(2) are autocorrelated and those of the ADF(3) are not. The resulting ADF statistic is obtained using:

```
dfuller lc, lags(3)
```

where the indicated number of lags is used.

```
dfuller lc, lags(3)
```

Breusch-Godfrey LM test for autocorrelation

<table>
<thead>
<tr>
<th>lags((p))</th>
<th>chi 2</th>
<th>df</th>
<th>Prob &gt; chi 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.056</td>
<td>1</td>
<td>0.8122</td>
</tr>
<tr>
<td>2</td>
<td>2.311</td>
<td>2</td>
<td>0.3148</td>
</tr>
<tr>
<td>3</td>
<td>3.499</td>
<td>3</td>
<td>0.3209</td>
</tr>
</tbody>
</table>

HD: no serial correlation

It is clear that the residuals of the ADF(2) are autocorrelated and those of the ADF(3) are not. The resulting ADF statistic is obtained using:

```
dfuller lc, lags(3)
```

where the indicated number of lags is used.

```
dfuller lc, lags(3)
```

Augmented Dickey-Fuller test for unit root

```
forvalues p = 1/3 {
    qui reg L(0/`p').D.ly L.ly
    di "Lags = `p"
    estat bgodfrey, lags(1 2 3)
}
```

As seen earlier in the course, the loop structure is fairly easy to use. It starts with `forvalues p`, where `p` is the counter, and `p` is instructed to increment beginning at 1 and ending at 3. The default increment size is one. Thus `p` will be set at 1, 2 and 3 as the loop runs. The first line must end with a left brace `{`. In the body of the loop are the regression, a display statement to indicate the lag length, and a call to the post-
estimation command **bgodfrey**. The loop has to be closed on a separate line using a right brace `}`. After initializing the counter, `p`, it must be referred to in single quotes (left ` and right `—which as you may recall are different characters on the keyboard). This is as it appears in the first two lines of the loop’s body. The output is:

```
Lags = 1
Breusch-Godfrey LM test for autocorrelation

<table>
<thead>
<tr>
<th>lags</th>
<th>chi2</th>
<th>df</th>
<th>Prob &gt; chi2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.208</td>
<td>1</td>
<td>0.6487</td>
</tr>
<tr>
<td>2</td>
<td>2.853</td>
<td>2</td>
<td>0.2401</td>
</tr>
<tr>
<td>3</td>
<td>2.880</td>
<td>3</td>
<td>0.4105</td>
</tr>
</tbody>
</table>
```

HD: no serial correlation

```
Lags = 2
Breusch-Godfrey LM test for autocorrelation

<table>
<thead>
<tr>
<th>lags</th>
<th>chi2</th>
<th>df</th>
<th>Prob &gt; chi2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.077</td>
<td>1</td>
<td>0.1495</td>
</tr>
<tr>
<td>2</td>
<td>2.539</td>
<td>2</td>
<td>0.2810</td>
</tr>
<tr>
<td>3</td>
<td>2.542</td>
<td>3</td>
<td>0.4677</td>
</tr>
</tbody>
</table>
```

HD: no serial correlation

```
Lags = 3
Breusch-Godfrey LM test for autocorrelation

<table>
<thead>
<tr>
<th>lags</th>
<th>chi2</th>
<th>df</th>
<th>Prob &gt; chi2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.157</td>
<td>1</td>
<td>0.6916</td>
</tr>
<tr>
<td>2</td>
<td>1.271</td>
<td>2</td>
<td>0.5297</td>
</tr>
<tr>
<td>3</td>
<td>2.098</td>
<td>3</td>
<td>0.5523</td>
</tr>
</tbody>
</table>
```

HD: no serial correlation

If nothing else, the output is a lot neater looking since none of the commands are echoed to the screen. For this series, no lagged differences of `ly` are included as regressors (i.e., the regular Dickey-Fuller regression). The Dickey-Fuller test yields:

```
. dfuller ly, lags(0)
```

**Dickey-Fuller test for unit root**

<table>
<thead>
<tr>
<th>Z(t)</th>
<th>-2.741</th>
<th>-3.477</th>
<th>-2.883</th>
<th>-2.573</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>-2.741</td>
<td>-3.477</td>
<td>-2.883</td>
<td>-2.573</td>
</tr>
<tr>
<td>Statistic</td>
<td>1% Critical Value</td>
<td>5% Critical Value</td>
<td>10% Critical Value</td>
<td></td>
</tr>
<tr>
<td>Interpolated Dickey-Fuller</td>
<td>-3.477</td>
<td>-2.883</td>
<td>-2.573</td>
<td></td>
</tr>
</tbody>
</table>

**MacKinnon approximate p-value for Z(t) = 0.0673**

The p-value is less than 10% but larger than 5%. It’s your call as to whether you are convinced that the unit root is rejected.

Recall that the cointegrating relationship can be estimated using least squares.

\[
C_t = \beta_1 + \beta_2 Y_t + v_t
\]

The residuals from this regression are obtained and their changes are regressed on the lagged value

\[
\Delta \hat{e}_t = \gamma \hat{e}_{t-1} + \delta \Delta \hat{e}_{t-1} + v_t
\]
The Stata code for this procedure is:

\begin{verbatim}
reg lc ly
predict ehat, res
reg D.ehat L.ehat D.L.ehat, noconst
di \_b[L.ehat]/\_se[L.ehat]
\end{verbatim}

Note that an intercept term has been included here to capture the component of (log) consumption that is independent of disposable income. The 5% critical value of the test for stationarity in the cointegrating residuals is $-3.37$. Since the unit root t-value of $-2.873$ is greater than $-3.37$, it indicates that the errors are not stationary, and hence that the relationship between $C$ (i.e., $\ln(RPCE)$) and $Y$ (i.e., $\ln(RPDI)$) is spurious—that is, we have no cointegration. In this case, estimate the coefficients of the model using a VAR in differences.

The VAR is simple to estimate in Stata. The easiest route is to use the `varbasic` command. `varbasic` fits a basic vector autoregressive (VAR) model and graphs the impulse-response functions (IRFs) or the forecast-error variance decompositions (FEVDs).

The basic structure of the VAR is found in the equations below:

\[ \Delta y_t = \beta_{11} \Delta y_{t-1} + \beta_{12} \Delta x_{t-1} + \nu_t^{y} \]
\[ \Delta x_t = \beta_{21} \Delta y_{t-1} + \beta_{22} \Delta x_{t-1} + \nu_t^{x} \]

The variables $x_t$ and $y_t$ are nonstationary, but the differences are stationary. Each difference is a linear function of its own lagged differences and of lagged differences of each of the other variables in the system. The equations are linear and least squares can be used to estimate the parameters. The `varbasic` command simplifies this. You need to specify the variables in the system ($\Delta y_t$ and $\Delta x_t$) and the number of lags to include on the right-hand-side of the model. In our example, only 1 lag is included and the syntax to estimate the VAR is:

\begin{verbatim}
varbasic D.lc D.ly, lags(1/1) step(12) nograph
\end{verbatim}

The syntax `lags(1/1)` tells Stata to include lags from the first number to the last, which in this case is lag 1 to lag 1. If your VAR is longer than 1 lag then you’ll change that here. Also added is the `step(12)` option. This option is useful in generating forecasts, a topic covered next. The output from this is:
In light of the fact that longer lags were used in the Dickey-Fuller regressions it is likely that the VAR should also have longer lags. In practice, it would probably be a good idea to test the residuals of the VAR for autocorrelation. The Stata command `varlmar` issued after `varbasic` will perform a LM test of the residuals similar to the ones we performed for autocorrelation.

There is evidence of autocorrelation in the residuals since the p-value at lag 1 is less than 5%.

Stata includes another procedure that makes selecting lag lengths in VAR models very easy. The `varsoc` command reports the final prediction error (FPE), Akaike's information criterion (AIC), Schwarz's Bayesian information criterion (SC), and the Hannan and Quinn information criterion (HQIC) lag-order selection statistics for a series of vector autoregressions. This can be used to find lag lengths for VAR or VEC models of unknown order. For the example above Stata yields:

```
. varbasic D.lc D.ly, lags(1/1) step(12) nograph

Vector autoregression

Sample: 1960q3 - 2009q4  No. of obs  =  198
Log likelihood = 1400.444  AIC           =  -14.0853
FPE           =  2.62e-09  HQC           =  -14.0496
Det(Sigma_ml) =  2.46e-09  SBC           =  -13.9856

Equation         Parms    RMSE    R-sq   chi2    P>chi2
D.lc               3       .006575  0.1205  27.12459  0.0000
D.ly               3       .008562  0.1118  24.92656  0.0000


            | Coef.  | Std. Err. |    z   |  P>|z|    | [95% Conf. Interval] |
D.lc         |        |           |       |       |                   |
|            | l_c   |           |       |       |                   |
|            | LD.   | .2156068  | .0741801 | 2.91  | 0.004 | .0702164  | .3609972 |
|            | l_y   | .1493798  | .0572953 | 2.61  | 0.009 | .0370832  | .2616765 |
|            | _cons | .0052776  | .0007516 | 7.02  | 0.000 | .0038046  | .0067507 |

D.ly         |        |           |       |       |                   |
|            | l_c   |           |       |       |                   |
|            | LD.   | .4754276  | .0965863 | 4.92  | 0.000 | .286122   | .6647332 |
|            | l_y   | -.2171679 | .0746013 | -2.91 | 0.004 | -.3633839 | -.070952  |
|            | _cons | .0060367  | .0009786 | 6.17  | 0.000 | .0041187  | .0079547 |
```
The AIC selects a lag order of 3 while the SC (labeled by Stata, SBIC) chooses 1.