1 Introduction

This report details an investigation of VAR (Vector Autoregressive) models and VECM (Vector Error Correction Models) diagnostics in Stata. We broadly define a diagnostic as a procedure performed on data that yields easily interpretable evidence for or against the fit of a particular model to that data.

With this definition of diagnostics, we are including things like the overall F-test for linear regression and individual wald-tests for the coefficients in a logistic regression. These simple tests are very important, and also familiar enough that discussing them is not very interesting. We will ignore these simple types of diagnostics and focus on the others.

These other diagnostics are often afterthoughts once a model is fit. An example of such a diagnostic is the rendering of a normal quantile-quantile plot of the standardized residuals after a linear regression. This gives important information about the normality of the errors, but is often done after looking at the $R^2$ and overall F-test values.

These other diagnostics yield important information about whether the data meets the more general assumptions of a model. Coefficient values are very specific information about a model, while the overall lag order and cointegration rank of the model are very general.

We focus on the more general diagnostics that Dr. Lütkepohl finds important in [Lütkepohl(2007)]. His text has been invaluable in this project. We will use examples and simulations within Stata to show and corroborate Lütkepohl’s statement of the nature of each diagnostic.

1.1 VAR model

We define a Vector Autoregressive (VAR) process of order $p$ as

$$y_t = v + A_1 y_{t-1} + \ldots + A_p y_{t-p} + u_t \quad t \in \mathbb{Z}$$

where

- $y_t$ is a $k \times 1$ random vector
- The $A_i$ are $k \times k$ fixed coefficient matrices
- $v$ is a $k \times 1$ fixed vectors of intercept terms
- $u_t$ is a $k \times 1$ white noise process

The random vectors $u_t$ are a white noise process iff
**VAR & VECM Diagnostics**

- \( Eu_t = 0 \)
- \( Eu_t u'_t = \Sigma_u \) (nonsingular)
- \( Eu_t u'_s = 0 \) if \( s \neq t \)

We will put two further conditions on our definition of a VAR \((p)\) process. These improve the utility of the process in statistical modeling.

The first condition is that the \( u_t \) are multivariate normal. This makes the \( y_t \) multivariate normal, so we call the \( y_t \) a gaussian process. The second condition is that \( y_t \) is a **stable** process. The stability condition is satisfied when for every complex \( z \), \(|I - A_1 z - \ldots - A_p z^p| \neq 0|\).

### 1.2 VECM model

When the stability condition is not satisfied, we may still analyze the process.

Let \( y_t \) be a \( k \times 1 \) VAR \((p)\) process. We say \( y_t \) is **integrated** of order \( d \), or \( y_t \sim I(d) \) if \( \Delta^d y_t \) is stable but \( \Delta^{d-1} y_t \) is not.

The sequence \( y_t \sim I(d) \) is **cointegrated** if there exists a fixed \( k \times 1 \) vector \( \beta \) such that \( \beta^T y_t \sim I(e) \) where \( e < d \). For convenience we will adapt the shorthand notation \( y_t \sim CI(d) \).

We define a Vector Error Correction Model (VECM) of order \( p \) as

\[
\Delta y_t = \Pi y_{t-1} + \Gamma_1 \Delta y_{t-1} + \ldots + \Delta_{p-1} y_{t-p+1} + u_t \quad t \in \mathbb{Z}
\]

where

- \( y_t \) is a \( k \times 1 \) random vector
- Aside from stability, \( y_t \) sequence is a VAR \((p)\) process
- \( y_t \sim CI(1) \)
- \( \Pi, \Gamma_1, \ldots, \Gamma_k \) are \( k \times k \) fixed coefficient matrices.
- \( u_t \) is a \( k \times 1 \) white noise process.

As before, we further stipulate that \( u_t \) is gaussian.

The matrix \( \Pi \) has rank \( r \leq k \) and \( \Pi = \alpha \beta^T \). The \( k \times r \) \( \alpha \) matrix is called the loading matrix. The \( r \times k \) \( \beta \) matrix is the called the cointegration matrix.

The columns of \( \beta \), \( \beta_i \), are such that \( \beta_i^T y_t \) is stable. The columns are cointegrating vectors. When we find the rank of cointegration for the VECM \( y_t \), we are finding the rank of \( \Pi \), the number of cointegrating vectors \( \beta_i \).

### 1.3 Stata

Stata is a popular and powerful statistical software package. Its time series facilities are very easy to use and extensive. It implements much of what Lütkepohl found important and many of the documentation examples use the data from his text.
He uses one dataset extensively, the West-German investment/income/consumption data. This dataset contains quarterly macro-economic data. Stata uses the data in the documentation for its VAR fit command \texttt{var}. We fit the suggested VAR(2) model (section 3.2.3 in [Lütkepohl(2007)]) on the first differences of the logarithms of each variable. The results on page 78 of [Lütkepohl(2007)] are matched.

.webuse lutkepohl2, clear  
(Quarterly SA West German macro data, Bil DM, from Lutkepohl 1993 Table E.1)  
.tsset qtr  
    time variable: qtr, 1960q1 to 1982q4  
    delta: 1 quarter  
.var dln_inv dln_inc dln_consump if qtr<=tq(1978q4),dfk

Vector autoregression

Sample: 1960q4 - 1978q4 No. of obs = 73
Log likelihood = 606.307 AIC = -16.03581
FPE = 2.18e-11 HQIC = -15.77323
Det(Sigma_ml) = 1.23e-11 SBIC = -15.37691
EquationParmsRMSE R-sqchi2 P>chi2
dln_inv7 .0461480.1286 9.736909 0.1362
dln_inc7 .0117190.1142 8.508289 0.2032
dln_consump7 .0094450.2513 22.15096 0.0011

|            | Coef. | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|------------|-------|-----------|-------|-------|----------------------|
| dln_inv    |       |           |       |       |                      |
| L1.        | -.3196318 | .1254564 | -2.55 | 0.011 | -.5655218 -.0737419 |
| L2.        | -.1605508 | .1249066 | -1.29 | 0.199 | -.4053633 .0842616 |
| dln_inc    |       |           |       |       |                      |
| L1.        | .1459851 | .5456664 | 0.27  | 0.789 | -.9235013 1.215472 |
| L2.        | .1146009 | .5345709 | 0.21  | 0.830 | -.9331388 1.162341 |
| dln_consump|       |           |       |       |                      |
| L1.        | .9612288 | .6643086 | 1.45  | 0.148 | -.3407922 2.26325  |
| L2.        | .9344001 | .6650949 | 1.40  | 0.160 | -.369162 2.237962  |
| _cons      | -.0167221 | .0172264 | -0.97 | 0.332 | -.0504852 .0170409 |

|            | Coef. | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|------------|-------|-----------|-------|-------|----------------------|
| dln_inc    |       |           |       |       |                      |
| L1.        | .0439309 | .0318592 | 1.38  | 0.168 | -.018512 1.063739  |
| L2.        | .0500302 | .0317196 | 1.58  | 0.115 | -.0121391 1.121995 |
| dln_inc    |       |           |       |       |                      |
| L1.        | -.1527311 | .1385702 | -1.10 | 0.270 | -.4243237 .1188615 |
| L2.        | .0191634 | .1357525 | 0.14  | 0.888 | -.2469067 .2852334 |
| dln_consump|       |           |       |       |                      |
| L1.        | .2884992 | .168699  | 1.71  | 0.087 | -.0421448 .6191431 |
| L2.        | -.0102   | .1688987 | -0.06 | 0.952 | -.3412354 .3208353 |
| _cons      | .0157672 | .0043746 | 3.60  | 0.000 | .0071932 .0243412 |

|            | Coef. | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|------------|-------|-----------|-------|-------|----------------------|
| dln_consump|       |           |       |       |                      |
| L1.        | -.002423 | .0256763 | -0.09 | 0.925 | -.0527476 .0479016 |
| L2.        | .0333806 | .0255638 | 1.33  | 0.185 | -.0162235 .0839847 |
| dln_consump|       |           |       |       |                      |
| L1.        | .2248134 | .1116778 | 2.01  | 0.044 | .005929 .4436978 |
| L2.        | .3549135 | .1094069 | 3.24  | 0.001 | .1404798 .5693471 |
| _cons      | .0129258 | .0035256 | 3.67  | 0.000 | .0060157 .0198358 |

We will use this dataset as an example for the VAR diagnostics that we discuss. The diagnostics will be explored through simulation and examples. The simulations required the creation of...
several Stata programs. Among other functions, these programs create realizations of VAR processes. They are listed in the appendix.

Many of the presented results are asymptotic in nature. If not stated otherwise, the limiting results apply as the length of the time series goes to infinity.

Now we are finished with the preliminaries and will begin discussing diagnostics. We begin with VAR diagnostics.

2 VAR diagnostics

2.1 Lag Order Selection

Determining the lag order of a VAR($p$) process $y_t$ is finding $p$ such that $A_i = 0$ for all $i > p$ in the model. So we are finding the index of the most lagged value of $y_t$ that should contribute to the current value.

There are two suggested approaches to choosing lag order. We may use a likelihood ratio test to verify the lag order. We can also use information criteria to choose the lag order that is most pragmatic.

2.1.1 Likelihood Ratio Test

Consider the first option. First find an upper bound on the lag order, $M$.

Consider the sequence of tests

$$H_0^i : A_{M-i+1} = 0 \quad \text{vs.} \quad H_1^i : A_{M-i+1} \neq 0 \quad H_{ij}^0, j = 1, \ldots, i - 1$$

To perform the likelihood ratio test, we run this sequence from $i = 1$ to $M$. When $H_0^i$ is the first hypothesis rejected, we choose the lag order to be $M - i + 1$. Let $\hat{\Sigma}_u(m)$ be the mle of $\Sigma_u$ when we fit a VAR($m$) model to $y_t$ and the $y_t$ process has observed length $T$. To test $H_0^i$, we use the simplified likelihood ratio statistic

$$\lambda_{LR}(i) = T \left[ \ln |\hat{\Sigma}_u (M - i)| - \ln |\hat{\Sigma}_u (M - i + 1)| \right]$$

Under the assumption that $H_0^i$ and all previous null hypotheses are true,

$$\lambda_{LR}(i) \xrightarrow{d} \chi^2(k^2)$$

Rejection of lower indices of $H_0^i$ implies rejection of higher indices. So the type 1 error rate of the individual tests differ as the sequence progresses. In [Paulsen and Tjostheim(1985)], it was shown that $\lambda_{LR}(m)$ and $\lambda_{LR}(j)$ are asymptotically independent given $H_0^1, \ldots, H_0^m$, $m \neq j$ and $m, j < i$. Let $\gamma_i$ be significance level used for the test $i$ in the sequence. The asymptotic probability of Type I error for the test $j$ in the sequence is

$$\epsilon_j = 1 - (1 - \gamma_1) \ldots \ast (1 - \gamma_j)$$

So the actual Type I error rate explodes as the sequence of null hypotheses gets long.

We will demonstrate this diagnostic via an example. We use the German bank data. The LR column of the \texttt{varsoc} command output provides the likelihood ratio test values. Using $M=4$ as our upper bound, we reject $H_0^3$ at level .05 since $\chi^2_{3,.95} = 16.92$. 

. varsoc, maxlag(4) lutstats
Selection-order criteria (lutstats)
Sample: 1961q2 - 1978q4
Number of obs = 71

<table>
<thead>
<tr>
<th>lag</th>
<th>LL</th>
<th>LR</th>
<th>df</th>
<th>p</th>
<th>FPE</th>
<th>AIC</th>
<th>HQIC</th>
<th>SBIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>564.784</td>
<td>2.7e-11</td>
<td>24.423</td>
<td>24.423*</td>
<td>24.423*</td>
<td>24.2102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>576.409</td>
<td>23.249</td>
<td>9</td>
<td>0.006</td>
<td>24.497</td>
<td>24.3829</td>
<td>24.005</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>588.859</td>
<td>24.901*</td>
<td>9</td>
<td>0.003</td>
<td>24.5942*</td>
<td>24.3661</td>
<td>23.5472</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>591.237</td>
<td>4.7566</td>
<td>9</td>
<td>0.855</td>
<td>24.4076</td>
<td>24.0655</td>
<td>23.5472</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>598.457</td>
<td>14.438</td>
<td>9</td>
<td>0.108</td>
<td>2.9e-11</td>
<td>24.3575</td>
<td>23.9012</td>
<td>23.2102</td>
</tr>
</tbody>
</table>

Endogenous: dln_inv dln_inc dln_consump
Exogenous: _cons

The details of this example are found on page 146 of [Lütkepohl(2007)].

### 2.1.2 Information Criteria

In linear regression, we do not always pick the model with the lowest RSS. Redundant predictors bloat the variance of estimates, and counterbalance the extra predictive power they bring to the model. We have a similar situation in lag order selection.

More lags mean more parameters to estimate. This leads to less biased but more variant predictions. We try to find a lag order estimate that does not sacrifice the precision of the model for accuracy. Using Information criteria as lag order selection values will help us.

The final prediction error (FPE) criterion measures the mean square error (MSE) of the 1-step ahead forecast

\[
\Sigma_y(1) = \frac{T + km + 1}{T}
\]

We adjust \( \hat{\Sigma}_u(m) \) to make it unbiased and then take the determinant.

\[
FPE(m) = \left[ \frac{T + km + 1}{T - km - 1} \right]^k |\hat{\Sigma}_u(m)|
\]

Akaike’s Information Criterion [Akaike(1974)] may also be used for lag order selection.

\[
AIC(m) = \ln |\hat{\Sigma}_u(m)| + \frac{2}{T} (#\text{free parameters})
\]

\[
= \ln |\hat{\Sigma}_u(m)| + \frac{2mk^2}{T}
\]

This may be interpreted as usual, (- Log likelihood of parameters) + (penalty term for # parameters).

There are two additional criteria of interest.

The Hannan and Quinn criterion [E.J. and Quinn(1979)] is similar to AIC, but uses a larger penalization for extra parameters

\[
HQ(m) = \ln |\hat{\Sigma}_u(m)| + \frac{2 \ln (\ln T)}{T} (#\text{free parameters})
\]

\[
= \ln |\hat{\Sigma}_u(m)| + \frac{2 \ln (\ln T)}{T} mk^2
\]
Schwarz used Bayesian arguments to derive the SC criterion in \cite{Schwarz(1978)}.

\[ SC(m) = \ln \left| \tilde{\Sigma}_u(m) \right| + \ln \frac{T}{T} \text{ (#free parameters)} \]

\[ = \ln \left| \tilde{\Sigma}_u(m) \right| + \ln \frac{T}{T} mk^2 \]

The different information criteria are related. Both HQ and SC are strong consistent for the true lag order. AIC and FP asymptotically overestimate the true lag order with positive probability. But in both small and large samples they may produce better forecasts.

Stata provides each of the information criteria through \texttt{varsoc} with the \texttt{lutstats} option. In the last section, we showed the information criteria for the German bank data. This may be compared with the table on page 148 of \cite{Lütkepohl(2007)}.

A rudimentary examination of the literature reveals that little work has been published detailing the use of the corrected AIC information criteria for VAR lag order selection. It has been used for lag order selection in univariate time series though. In Hurvich and Tsai’s original article \cite{Hurvich and Tsai(1989)} it was used to select the order of an AR model. This AICC measure is obtained by AIC with an additional penalty term. A precise reformulation of the additional penalty term is necessary for the implementation of AICC in VAR lag order selection. Not having access to this reformulation, we will not use the AICC measure here.

It will be instructive to test the asymptotic performance of the information criteria in choosing the actual lag order. Using the author’s \texttt{VARgen} command we simulate draws from a simple bivariate VAR(2) process with the following parameters.

\[
\begin{align*}
v &= \begin{pmatrix} .02 \\ .03 \end{pmatrix} \\
A_1 &= \begin{pmatrix} .5 & .1 \\ .4 & .5 \end{pmatrix} \\
A_2 &= \begin{pmatrix} 0 & 0 \\ .25 & 0 \end{pmatrix} \\
\Sigma_u &= \begin{pmatrix} .09 & 0 \\ 0 & .04 \end{pmatrix}
\end{align*}
\]

For each draw, we calculate the information criteria for lag orders 1-6 and record the optimal order for each criteria. We report the percentage of correct choices for each information criteria over 500 realizations of the VAR(2) process for each of the time lengths 30, 60, 100, 200, 500, 1000.

<table>
<thead>
<tr>
<th>T</th>
<th>FPE</th>
<th>AIC</th>
<th>HQ</th>
<th>SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>.324</td>
<td>.312</td>
<td>.292</td>
<td>.152</td>
</tr>
<tr>
<td>60</td>
<td>.474</td>
<td>.478</td>
<td>.382</td>
<td>.15</td>
</tr>
<tr>
<td>100</td>
<td>.694</td>
<td>.692</td>
<td>.558</td>
<td>.262</td>
</tr>
<tr>
<td>200</td>
<td>.862</td>
<td>.862</td>
<td>.864</td>
<td>.582</td>
</tr>
<tr>
<td>500</td>
<td>.882</td>
<td>.882</td>
<td>.998</td>
<td>.98</td>
</tr>
<tr>
<td>1000</td>
<td>.886</td>
<td>.886</td>
<td>.994</td>
<td>1</td>
</tr>
</tbody>
</table>

As expected, we find poor performance for each information criterion in the smaller sample sizes. It improves for each criterion as the sample size increases. The positive probability of error by AIC and FPE as the sample size approaches infinity is evident. Note that this simulation only checked that the information criteria selected the actual lag order, not the pragmatic lag order.
2.2 Whiteness of Residuals

The $u_t$ error process of a VAR($p$) model is unknown. We estimate it with the observed residuals $\hat{u}_t$. Using $\hat{u}_t$, there are two tests commonly used for testing the autocorrelation restriction of the $u_t$. Tests of the other whiteness properties are possible as well, but we will not focus on these.

2.2.1 Portmanteau Test

The Portmanteau test jointly tests the significance of all error autocorrelations up to a set order, $h$. Define $R_i$ as the autocorrelation matrix for lag $i$ among the errors. The Portmanteau test evaluates

$$H_0 : R_h = (R_1, \ldots, R_h) = 0 \quad vs. \quad H_1 : R_h \neq 0$$

The matrix $\hat{R}_i$ is the estimated autocorrelation of lag $i$. Similarly, $\hat{R}_u$ is the estimated correlation matrix of $u_t$.

The test statistic for Portmanteau test is

$$Q_h = T \sum_{i=1}^{h} tr \left( \hat{R}_i^T \hat{R}_u^{-1} \hat{R}_i \hat{R}_u^{-1} \right)$$

For large $T$ and $h$, under $H_0$

$$Q_h \approx \chi^2 (k^2 (h - p))$$

Lütkepohl actually suggests a modification of $Q_h$, $\overline{Q}_h$ be used.

$$\overline{Q}_h = T^2 \sum_{i=1}^{h} \frac{1}{T-i} tr \left( \hat{R}_i^T \hat{R}_u^{-1} \hat{R}_i \hat{R}_u^{-1} \right)$$

Stata implements the Portmanteau test through the user written command `wntstmvq`. This command is documented in [Sperling and Baum(2001)]. The statistic used in `wntstmvq` is a slight modification of the one mentioned in [Lütkepohl(2007)]. It uses a different scaling constant than $T^2$. Both of these modifications to the original $Q_h$ result in more appropriate values for small samples. Neither changes the asymptotic distribution of the test statistic as well. The conclusion we reach (accept no autocorrelation at the .05 level) is still the same as page 171 in [Lütkepohl(2007)]. Note that the $d$ variables are the residuals from the model and $h = 12$.

```
. wntstmvq d1 d2 d3, lags(12) varlags(2)
Multivariate Ljung-Box statistic (3 variables, 12 lags): 96.7943
Prob > chi2(90) = 0.2934
```

2.2.2 Lagrange Multiplier Test

Assume a VAR model for the error

$$u_t = D_1 u_{t-1} + \ldots + D_h u_{t-h} + v_t$$
To test autocorrelation up to order $h$ in $u_t$, we test

$$H_0 : D_1 = \ldots = D_h = 0 \quad vs. \quad H_1 : D_j \neq 0 \text{ for at least one } j < h$$

We use the lagrange multiplier method to perform the test. This method is very useful for finding optimal estimates under constraint conditions. Lütkepohl suggests a particular test called the Breusch-Godfrey test. Stata implements a different lagrange multiplier test, which we will discuss later.

Under $H_0$ we only need to estimate the regular VAR model ($u_t = v_t$). So the constrained case estimates are simple.

To determine the test statistic we begin with the auxiliary regression model

$$\hat{U} = BZ + D\hat{U} + E$$

where

$$\hat{U} = [\hat{u}_1 \ldots \hat{u}_T]$$
$$Z_t = [1^T \quad y_t^T \ldots y_{t-p+1}^T]^T$$
$$Z = [Z_0 \ldots Z_{T-1}]$$
$$D = [D_1 \ldots D_h]$$

Define $F_i$ such that

$$\hat{U}F_i\hat{U}^T = \sum_{t=i+1}^T \hat{u}_t\hat{u}_{t-i}^T$$

then

$$F = [F_1 \ldots F_h]$$
$$\hat{U} = (I \otimes \hat{U}) F^T$$

This yields the least squares estimate of $D$

$$\hat{D} = \hat{U}\hat{U}^T \left( \hat{U}\hat{U}^T - \hat{U}Z^T (ZZ^T)^{-1} Z\hat{U}^T \right)^{-1}$$

The standard $\chi^2$ test statistic for testing whether $D = 0$ (no autocorrelation) is

$$\lambda_{LM}(h) = vec(\hat{D})^T \left( \left[ \hat{U}\hat{U}^T - \hat{U}Z^T (ZZ^T)^{-1} Z\hat{U}^T \right] \otimes \hat{\Sigma}_u \right) vec(\hat{D})$$

Under $H_0$,

$$\lambda_{LM}(h) \overset{d}{\rightarrow} \chi^2(hk^2)$$

Stata implements a lagrange multiplier test through the \texttt{varlmar} command. Rather than using Breusch-Godfrey, they use a test advocated in [Johansen(1995)].
Let $\hat{\Sigma}_s$ be the MLE of $\Sigma_u$ when the residuals of the original model (up to lag $s$) are added to the model as exogenous variables. This means that they are treated as independent of the errors in estimation. Denote the number of estimated coefficients in the new VAR by $d$.

The lagrange multiplier statistic from [Johansen(1995)] is given by

$$
\lambda_{LMJ}(s) = (T - d - .5) \ln \left( \frac{|\hat{\Sigma}_u|}{|\hat{\Sigma}_s|} \right)
$$

We demonstrate the `varlmar` command using the German bank data.

```
. varlmar, mlag(4)

Lagrange-multiplier test

<table>
<thead>
<tr>
<th>lag</th>
<th>ch12</th>
<th>df</th>
<th>Prob &gt; ch12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.5871</td>
<td>9</td>
<td>0.78043</td>
</tr>
<tr>
<td>2</td>
<td>6.3189</td>
<td>9</td>
<td>0.70763</td>
</tr>
<tr>
<td>3</td>
<td>8.4022</td>
<td>9</td>
<td>0.49418</td>
</tr>
<tr>
<td>4</td>
<td>11.8742</td>
<td>9</td>
<td>0.22049</td>
</tr>
</tbody>
</table>
```

H0: no autocorrelation at lag order

We arrive at matching conclusions to those on page 174 of [Lütkepohl(2007)].

Under the null hypothesis,

$$
\lambda_{LMJ}(s) \xrightarrow{d} \chi^2(k^2)
$$

It will be instructive to check the asymptotic distributions of $\lambda_{LMJ}(s)$ and Stata's version of $Q_h$. We do this by simulation, using the VAR(2) model from section 2.1.2 and following a similar strategy to the simulation in that section.

We will generate 500 realizations from length 30, 60, 100, and 500 sequences of the model. The test statistics for autocorrelations up to order 3 will be computed with each realization. Then chi-squared quantile plots will be constructed using these realizations. The results are given in the following figure. As expected, the realizations conform more to the appropriate $\chi^2$ distribution as the sample size increases.
Now we look at order ten for the Portmanteau test. The increased order causes greater conformity to the anticipated $\chi^2$ distribution.
2.3 Normality of Residuals

Lütkepohl suggests using the multivariate generalization of the Jarque-Bera test [Jarque and Bera(1987)] on \( \hat{u}_t \) to test the multivariate normality of the \( u_t \).

This tests the skewness and kurtosis properties of the \( u_t \) against those of a multivariate normal distribution of the appropriate dimension.

It is possible that the first four moments of the \( u_t \) match the multivariate normal moments, and the \( u_t \) are still not normally distributed. It is hoped that most of the normal properties desired by the model fitter in the \( u_t \) are met by these four moments.

This situation has an analog in linear regression. We assume that the errors are independent, but we can only test whether they are correlated. In linear regression, it is adequate to test the correlation of the residuals. If they are uncorrelated, that is enough independence for getting the variance calculations correct. We do not worry about the other forms of dependence.

Our formulation of the Jarque-Bera test uses a mean adjusted form of the VAR(p) model, but it applies to our general form in 1.2.

\[
\hat{u}_t = (y_t - \bar{y}) - \hat{A}_1 (y_{t-1} - \bar{y}) - \ldots - \hat{A}_p (y_{t-p} - \bar{y})
\]

\[
\hat{\Sigma}_u = \frac{1}{T-kp-1} \sum_{t=1}^{T} \hat{u}_t\hat{u}_t^T
\]

\[
\hat{P}\hat{P}^T = \hat{\Sigma}_u \ni \hat{P} \to \Sigma_u^{1/2} \text{ a.s.}
\]

Now we define the standardized residuals and their sample moments.

\[
\hat{w}_t = \hat{P}^{-1}\hat{u}_t
\]

\[
\hat{b}_1 = (\hat{b}_{11} \ldots \hat{b}_{k1}) \ni \hat{b}_{i1} = \frac{1}{T} \sum_{t=1}^{T} \hat{w}_t^{3i}
\]

\[
\hat{b}_2 = (\hat{b}_{12} \ldots \hat{b}_{k2}) \ni \hat{b}_{i2} = \frac{1}{T} \sum_{t=1}^{T} \hat{w}_t^{4i}
\]

Finally our test statistics are

\[
\lambda_s = T\hat{b}_1^T\hat{b}_1/6
\]

\[
\lambda_k = T \left( \hat{b}_2 - 31 \right)^T \left( \hat{b}_2 - 31 \right) / 24
\]

\[
\lambda_{sk} = \lambda_s + \lambda_k
\]

The third and fourth moments of \( u_t \) should be 0 and 31.

Under the third moment assumption

\[
\lambda_s \xrightarrow{d} \chi^2(k)
\]

Under the fourth moment assumption

\[
\lambda_k \xrightarrow{d} \chi^2(k)
\]
Under both assumptions

$$\lambda_{sk} \rightarrow^d \chi^2(2k)$$

So all three test statistics may be used to test the multivariate normality of $u_t$. Stata implements these tests using `varlnorm`. We shall demonstrate the tests using the German bank data. Stata gives the univariate test statistics for each variable as well. These results match those on page 181 of [Lütkepohl(2007)].

```
. varnorm

Jarque-Bera test

<table>
<thead>
<tr>
<th>Equation</th>
<th>chi2</th>
<th>df</th>
<th>Prob &gt; chi2</th>
</tr>
</thead>
<tbody>
<tr>
<td>dln_inv</td>
<td>2.821</td>
<td>2</td>
<td>0.24397</td>
</tr>
<tr>
<td>dln_inc</td>
<td>3.450</td>
<td>2</td>
<td>0.17817</td>
</tr>
<tr>
<td>dln_consump</td>
<td>1.566</td>
<td>2</td>
<td>0.45702</td>
</tr>
<tr>
<td>ALL</td>
<td>7.838</td>
<td>6</td>
<td>0.25025</td>
</tr>
</tbody>
</table>
```

Skewness test

```
<table>
<thead>
<tr>
<th>Equation</th>
<th>Skewness</th>
<th>chi2</th>
<th>df</th>
<th>Prob &gt; chi2</th>
</tr>
</thead>
<tbody>
<tr>
<td>dln_inv</td>
<td>-0.11935</td>
<td>0.173</td>
<td>1</td>
<td>0.67718</td>
</tr>
<tr>
<td>dln_inc</td>
<td>0.38316</td>
<td>1.786</td>
<td>1</td>
<td>0.18139</td>
</tr>
<tr>
<td>dln_consump</td>
<td>-0.31275</td>
<td>1.190</td>
<td>1</td>
<td>0.27532</td>
</tr>
<tr>
<td>ALL</td>
<td>3.150</td>
<td></td>
<td>3</td>
<td>0.36913</td>
</tr>
</tbody>
</table>
```

Kurtosis test

```
<table>
<thead>
<tr>
<th>Equation</th>
<th>Kurtosis</th>
<th>chi2</th>
<th>df</th>
<th>Prob &gt; chi2</th>
</tr>
</thead>
<tbody>
<tr>
<td>dln_inv</td>
<td>3.9331</td>
<td>2.648</td>
<td>1</td>
<td>0.10367</td>
</tr>
<tr>
<td>dln_inc</td>
<td>3.7396</td>
<td>1.664</td>
<td>1</td>
<td>0.19710</td>
</tr>
<tr>
<td>dln_consump</td>
<td>2.6484</td>
<td>0.376</td>
<td>1</td>
<td>0.53973</td>
</tr>
<tr>
<td>ALL</td>
<td>4.688</td>
<td></td>
<td>3</td>
<td>0.19613</td>
</tr>
</tbody>
</table>
```

dfk estimator used in computations

We will perform simulations to examine the distribution of the Jarque-Bera test statistics at sequence lengths 30, 60, 100, and 500. As before we will use 500 realizations of each sequence length and use the VAR(2) model introduced in section 2.1.2. The results are as expected, increasing the sequence length makes the statistics more conformable to the appropriate $\chi^2$ distributions.
2.4 Stability Test

Checking that a VAR(p) process is stable is fairly straightforward. We merely find all the roots of $|I - A_1 z - \ldots - A_p z^p|$, plugging in the estimates of the $A_i$. If none of the roots are real, we have met the stability condition. Equivalently, we may check that the eigenvalues of the following matrix are less than one. If they are then the VAR is stable.

The stability check matrix is

$$ A = \begin{pmatrix} A_1 & A_2 & \ldots & A_{p-1} & A_p \\ I & 0 & \ldots & 0 & 0 \\ 0 & I & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & I & 0 \end{pmatrix} $$

Stata uses the `varstable` command to perform this task. We demonstrate using the German bank data.

```
    . varstable
    Eigenvalue stability condition

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Modulus</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5704708</td>
<td>.570471</td>
</tr>
<tr>
<td>-.3905477 + .3890671i</td>
<td>.581272</td>
</tr>
<tr>
<td>-.3905477 - .3890671i</td>
<td>.581272</td>
</tr>
<tr>
<td>-.077255 + .4856136i</td>
<td>.49172</td>
</tr>
<tr>
<td>-.077255 - .4856136i</td>
<td>.49172</td>
</tr>
<tr>
<td>-.3711977</td>
<td>.371198</td>
</tr>
</tbody>
</table>
```

All the eigenvalues lie inside the unit circle. VAR satisfies stability condition.
3 VECM diagnostics

Many VECM diagnostics are very similar to those of VAR. We will not discuss or investigate these overlapping diagnostics. This leaves us discussing one diagnostic.

3.1 Rank of Cointegration

When we find the rank of cointegration, we are finding the rank of $\Pi$, the number of cointegrating vectors $\beta_i$.

For brevity, we will investigate testing the rank under the simple model with no deterministic terms (no intercept, linear trend in $t$, etc.). We originally specified our VECM models in this way.

Let $l(r)$ be the maximum of the likelihood of the VECM model under cointegration rank $r_i$ (similar to profile likelihood).

$$
\lambda_{LR}(r_0, r_1) = 2 \left[ \ln l(r_0) - \ln l(r_1) \right]
= -T \sum_{i=r_0+1}^{r_1} \ln (1 - \lambda_i)
$$

The $\lambda_i$ are the ordered eigenvalues of a particular (and complicated) matrix used in the ML estimation of the VECM. It would require too much space to develop this matrix here. Details can be found in [Lütkepohl(2007)].

Under $H_0: \text{rank}(\Pi) = r_0$, the asymptotic distribution of $\lambda_{LR}(r_0, r_1)$ is not $\chi^2$ or any familiar distribution.

Two cases were considered in [Johansen(1995)]

$$
H_0 : \text{rank}(\Pi) = r_0 \text{ vs. } r_0 < \text{rank}(\pi) = k
\quad H_0 : \text{rank}(\Pi) = r_0 \text{ vs. } \text{rank}(\Pi) = r_0 + 1
$$

The test statistic for the first $\lambda_{LR}(r_0, k)$ is called the trace statistic. The test statistic for the second $\lambda_{LR}(r_0, r_0 + 1)$ is called the maximum eigenvalue statistic.

Let $W$ be a $k - r_0$ dimension standard Wiener process. The Wiener process is defined in appendix C section 8 of [Lütkepohl(2007)]

$$
D = \left( \int_0^1 W dW^T \right)^T \left( \int_0^1 W W^T ds \right)^{-1} \left( \int_0^1 W dW^T \right)
$$

Johansen found that under $H_0 : \text{rank}(\Pi) = r_0$

$$
\lambda_{LR}(r_0, k) \xrightarrow{d} \text{tr}(D)
\quad \lambda_{LR}(r_0, r_0 + 1) \xrightarrow{d} \lambda_{\text{max}}(D)
$$

Stata supports cointegration rank testing through the vecrank command. We will demonstrate this command by an example. We use a non-Lütkepohl dataset for this purpose. The balance2 dataset contains quarterly data on the natural logs of aggregate consumption, investment, and GDP in the US from
the first quarter of 1959 to the fourth quarter of 1982. The balanced growth hypothesis in economics [King et al.(1991)King, Plosser, Stock, and Watson] would imply that we find two cointegrating equations among the three variables. Unlike the var diagnostic commands, we do not need to fit a VECM before we use vecrank.

```
webuse balance2, clear
(macro data for VECM/balance study)
vecrank y i c, max trend(none) lags(5)
```

**Johansen tests for cointegration**

<table>
<thead>
<tr>
<th>Trend: none</th>
<th>Number of obs = 91</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample: 1960q2 - 1982q4</td>
<td>Lags = 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>maximum</th>
<th>rank</th>
<th>parms</th>
<th>LL</th>
<th>eigenvalue</th>
<th>trace statistic value</th>
<th>critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>36</td>
<td>1224.3691</td>
<td>48.0021</td>
<td>24.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>41</td>
<td>1236.9105</td>
<td>0.24091</td>
<td>12.53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>44</td>
<td>1245.8999</td>
<td>0.17928</td>
<td>3.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>45</td>
<td>1248.3702</td>
<td>0.05284</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
vecrank y i c, max trend(constant) lags(5)
```

**Johansen tests for cointegration**

<table>
<thead>
<tr>
<th>Trend: constant</th>
<th>Number of obs = 91</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample: 1960q2 - 1982q4</td>
<td>Lags = 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>maximum</th>
<th>rank</th>
<th>parms</th>
<th>LL</th>
<th>eigenvalue</th>
<th>trace statistic value</th>
<th>critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>39</td>
<td>1231.1041</td>
<td>25.0826</td>
<td>17.89</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>44</td>
<td>1245.3882</td>
<td>0.26943</td>
<td>11.44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>47</td>
<td>1252.5055</td>
<td>0.14480</td>
<td>3.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>48</td>
<td>1254.1787</td>
<td>0.03611</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We reject the hypothesis that the cointegrating matrix Π is of less than full rank at the .05 level.

Not having a trend in our VECM may be naive. If we were to allow for a constant trend, and adjust our definition of the trace and maximum eigenvalue statistics, we would obtain the theoretically expected result. Both statistics fall below the critical point for the test of cointegration rank 2, so we do not reject $H_0: \text{rank}(\Pi) = 2$.

```
vecrank y i c, max trend(constant) lags(5)
```

**Johansen tests for cointegration**

<table>
<thead>
<tr>
<th>Trend: constant</th>
<th>Number of obs = 91</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample: 1960q2 - 1982q4</td>
<td>Lags = 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>maximum</th>
<th>rank</th>
<th>parms</th>
<th>LL</th>
<th>eigenvalue</th>
<th>trace statistic value</th>
<th>critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>39</td>
<td>1231.1041</td>
<td>28.5682</td>
<td>20.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>44</td>
<td>1245.3882</td>
<td>0.26943</td>
<td>14.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>47</td>
<td>1252.5055</td>
<td>0.14480</td>
<td>3.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>48</td>
<td>1254.1787</td>
<td>0.03611</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4 Conclusion

This report detailed the theory and application of VAR and VECM diagnostics. The reader should be able to apply these diagnostics in Stata and correctly interpret the results. The reader should also be able to choose which of these diagnostics is appropriate to use with his or her data.

References


5 Appendix

Several programs were used to create this report. Stata allows for the creation of .ado files which are automatically loaded programs. Two of these were used.

The first .ado file allowed for the creation of gaussian white noise processes. It was called `errorGen.ado`.

```
//for normal error generation, uncorrelated errors
// T is sequence length
// covmat is covariance matrix for error
program errorGen
syntax newvarlist , t(integer) covmat(string)
drawnorm ‘varlist’, clear cov(‘covmat’) n(‘t’)
end
```
The other .ado file used errorGen to create VAR processes. It was called VARgen.ado

//for normal VAR process, uncorrelated errors
//t is the length of the process
//s is the number of burnoff observations
// (actually s+1 burnoff observations, -s, ..., 0
//errcov is the covariance matrix of the errors
//p is the lag order
//Alist is a stringlist containing the names of p coefficient matrices
//v is the mean vector

program VARgen
syntax , t(integer) s(integer) errcov(string) p(integer) alist(string) v(string)
quietly {
    clear
    local tot = `t' + `s' + 1
    set obs `tot'

    //dimension of process
    local k = colsof(`errcov')

    //generate errors
    errorGen er1-er`k', covmat(`errcov') t(`tot')

    //organize alist
    tokenize "`alist'"
    forvalues i = 1/`p' {
        matrix A'`i' = "`i'"
    }

    // create VAR variates
    forvalues i = 1/`k' {
        gen y'`i' = .
    }

    //initialize first p values
    //do more complicated setup as Lutkepohl mentions, but not necessarily suggests
    //if necessary
    //for now, try starting initial values as the process mean
    matrix procmean = I(`k')
    forvalues i = 1/`p' {
        matrix procmean = procmean - A'`i'
    }
    matrix procmean = inv(procmean)*`v'
    forvalues i = 1/`k' {
        replace y'`i' = procmean[`i',1] if _n <= `p'
    }

    //now do recursion to fill in rest.
    local i = `p'+1
    while `i'<`tot' {
        //matrix with variables in columns, observations rows
mkmat y1-y'k' if _n <= i & _n >= i-p, matrix(tmpdatamat)
//now make observations columns
matrix tmpdatamat = tmpdatamat'
matrix actval = y'
forvalues j = 1/p {
    matrix actval = actval + A'j'*tmpdatamat[1..k',('p'-'j'+1)]
}
forvalues g = 1/p {
    replace y'g' = actval['g',1] + er'g' if _n == i
}
local i = i + 1
}
end

The rest of the program files were .do files. These are analysis files that do not specifically define named programs, as .ado files do.

The first .do file used was VARnu.do. It created the non-simulation output in the report.

version 10.0
set more off
clear all
set seed 123456789

sjlog using vv1p3, replace
webuse lutkepohl2, clear
tsset qtr
var dln_inv dln_inc dln_consump if qtr<=tq(1978q4),dfk
sjlog close, replace

sjlog using vv2p1p1, replace
varsoc, maxlag(4) lutstats
sjlog close, replace

predict d1, equation(#1) residuals
predict d2, equation(#2) residuals
predict d3, equation(#3) residuals
sjlog using vv2p2p1, replace
wntstmvq d1 d2 d3, lags(12) varlags(2)
sjlog close, replace

sjlog using vv2p2p2, replace
varlmar, mlag(4)
sjlog close, replace

sjlog using vv2p3, replace
varnorm
sjlog close, replace
VAR & VECM Diagnostics

sjlog using vv2p4, replace
varstable
sjlog close, replace

sjlog using vv3p1, replace
webuse balance2, clear
vecrank y i c, max trend(none) lags(5)
sjlog close, replace

sjlog using vv3p1_2, replace
vecrank y i c, max trend(constant) lags(5)
sjlog close, replace

The remaining .do files created the simulation results. The first of these was sect2p1p2.do. It created the information criteria table in section 2.1.2.

version 10.0
set more off
clear all

matrix v = (.02\ .03)
matrix A1 = (.5,.1\ .4,.5)
matrix A2 = (0,0\ .25,0)
matrix sigu = (.09,0\ 0,.04)

postfile lagOrd T FPE AIC HQ SC using lagorder , replace

docal tll "30 60 100 200 500 1000"
foreach tlen of local tll {
    forvalues si = 1/500 {
        if (mod('si',10)==0) {
            di 'tlen' " " 'si'
        }
        VARgen ,t('tlen') s(100) errcov(sigu) p(2) alist("A1 A2") v(v)
gen time = _n
qui
tset time
qui var y1 y2 if _n > 101, dfk
qui varsoc, maxlag(6) lutstats
matrix ta = r(stats)
matrix ta = ta[1..7,6..9]
forvalues i = 1/4 {
    scalar x'i' = min(ta[1,'i'],ta[2,'i'],ta[3,'i'],ta[4,'i'],ta[5,'i'],ta[6,'i'],ta[7,'i'])
    forvalues k = 1/7 {
        if (ta['k','i'] == x'i') {
            scalar x'i' = 'k'-1
            break
        }
    }
}
}
The next do file was sect2p2p2_sim.do. It created figures 1 and 2 in section 2.2.2.

version 10.0
set more off
clear all
matrix v = (.02\.03)
matrix A1 = (.5,.1\.4,.5)
matrix A2 = (0,0\.25,0)
matrix sigu = (.09,0\0,.04)

postfile white sim T port lms using acwhite , replace
local acorder = 3
local tll "30 60 100 500"
foreach tlen of local tll {  
    forvalues si = 1/500 {  
        if (mod(`si',10) == 0) {  
            di `tlen' " " `si'
        }  
        VARgen ,t(`tlen') s(100) erccov(sigu) p(2) alist("A1 A2") v(v)
        gen time = _n
        qui tsset time
        qui var y1 y2 if _n > 101, dfk
        qui varlmar, mlag(`acorder')
        matrix td = r(lm)
        local varlret = td['`acorder',2]
        qui predict d1 , equation(#1) residuals
        qui predict d2 , equation(#2) residuals
        qui wntstmvq d1 d2, lags(`acorder') varlags(2)
        qui post white (`si') (`tlen') (r(stat)) (`varlret')
    }
}
postclose white
use acwhite, clear

local tll "30 60 100 500"
foreach tlen of local tll {  
    local def = 2*2*(`acorder'-2)
VAR & VECM Diagnostics

```stata
qchi port if T == 'tlen', df('def') title("T = 'tlen'") name(port'tlen') nodraw
local def = 2*2
qchi lms if T == 'tlen', df('def') title("T = 'tlen'") name(lms'tlen') nodraw
}

graph combine port30 port60 port100 port500, title("Portmanteau")
subtitle("Order 'acorder'") name(a) cols(4) nodraw
graph combine lms30 lms60 lms100 lms500, title("Lagrange Multiplier")
subtitle("Order 'acorder'") name(b) cols(4) nodraw
graph combine a b, rows(2)

graph save white1.gph, replace
clear all

matrix v = (.02\.03)
matrix A1 = (.5,.1\.4,.5)
matrix A2 = (0,0\.25,0)
matrix sigu = (.09,0\0,.04)

postfile white sim T port lms using acwhite , replace
local acorder = 10
local tll "30 60 100 500"
foreach tlen of local tll {
    foreach si of local si {1/500 {
        if (mod('si',10) == 0) {
            di 'tlen' " " 'si'

        }
        VARgen ,t('tlen') s(100) errcov(sigu) p(2) alist("A1 A2") v(v)
gen time = _n
qui tsset time
qui var y1 y2 if _n > 101,dfk
qui varlmar, mlag('acorder')
matrix td = r(lm)
local varlret = td['acorder',2]
qui predict d1 , equation(#1) residuals
qui predict d2 , equation(#2) residuals
qui wntstmvq d1 d2, lags('acorder') varlags(2)
qui post white ('si') ('tlen') (r(stat)) ('varlret')
    }
}
}
postclose white

local acorder = 10
use acwhite, clear
local tll "30 60 100 500"
foreach tlen of local tll {
    local def = 2*2*('acorder'-2)
    qchi port if T == 'tlen', df('def') title("T = 'tlen'") name(port'tlen') nodraw
}
```

```
VAR & VECM Diagnostics

The final .do file was sect2p3.sim.do. It created figure 3 in section 2.3.

version 10.0
set more off
clear all
matrix v = (.02\ .03)
matrix A1 = (.5,.1| .4,.5)
matrix A2 = (0,0| .25,0)
matrix sigu = (.09,0\0,.04)

postfile norm sim T kurt skew jb using normjb , replace
local acorder = 3
local tll "30 60 100 500"
foreach tlen of local tll {
    forvalues si = 1/500 {
        if (mod('si',10) == 0) {
            di 'tlen' " " 'si'
        }
        VARgen ,t('tlen') s(100) errcov(sigu) p(2) alist("A1 A2") v(v)
        gen time = _n
        qui tsset time
        qui var y1 y2 if _n > 101, dfk
        qui varnorm
        matrix rkurt = r(kurtosis)
        matrix rskew = r(skewness)
        matrix rjb = r(jb)
        local rkurt = rkurt[3,2]
        local rskew = rskew[3,2]
        local rjb = rjb[3,1]
        qui post norm ('si') ('tlen') ('rkurt') ('rskew') ('rjb')
    }
}
postclose norm
use normjb, clear

local tll "30 60 100 500"
foreach tlen of local tll {
    qchi kurt if T == 'tlen', df(2) title("T = 'tlen'") name(kurt'tlen') nodraw
    qchi skew if T == 'tlen', df(2) title("T = 'tlen'") name(skew'tlen') nodraw
    qchi jb if T == 'tlen', df(4) title("T = 'tlen'") name(jb'tlen') nodraw
}

graph combine kurt30 kurt60 kurt100 kurt500, subtitle("Kurtosis Statistic") name(a) cols(4) nodraw
graph combine skew30 skew60 skew100 skew500, subtitle("Skewness Statistic") name(b) cols(4) nodraw
graph combine jb30 jb60 jb100 jb500, subtitle("Overall Statistic") name(c) cols(4) nodraw
graph combine a b c, title("Jarque-Berra") rows(3)
graph save jb.gph, replace