

Multivariate Time Series: Recent Additions to the VARMAX Procedure

Xilong Chen and Stefanos Kechagias, SAS Institute Inc.

ABSTRACT

Recent advances in computing technology, monitoring systems, and data collection mechanisms have prompted renewed interest in multivariate time series analysis. In contrast to univariate time series models, which focus on temporal dependencies of individual variables, multivariate time series models also exploit the interrelationships between different series, thus often yielding improved forecasts. This paper focuses on cointegration and long memory, two phenomena that require careful consideration and are observed in time series data sets from several application areas, such as finance, economics, and computer networks. Cointegration of time series implies a long-run equilibrium between the underlying variables, and long memory is a special type of dependence in which the impact of a series' past values on its future values dies out slowly with the increasing lag. Two examples illustrate how you can use the new features of the VARMAX procedure in SAS/ETS® 14.1 and 14.2 to glean important insights and obtain improved forecasts for multivariate time series. One example examines cointegration by using the Granger causality tests and the vector error correction models, which are the techniques frequently applied in the Federal Reserve Board's Comprehensive Capital Analysis and Review (CCAR), and the other example analyzes the long-memory behavior of US inflation rates.

INTRODUCTION

The VARMAX procedure is a well-established and powerful tool for analyzing multivariate time series. Through the large variety of models that it offers, its users can exploit both the temporal dependencies of individual series and the interrelationships between different series to obtain high-quality forecasts. This paper focuses on two new additions to the VARMAX procedure in SAS/ETS 14.1 and 14.2.

The first addition is the enhancement of the vector error correction model (VECM), which you can use to model nonstationary series. Joint modeling of nonstationary time series is known to give rise to spurious relationships when the series are independent. However, in economic theory, time series are often linked by a long-run linear relationship—a phenomenon known as cointegration. In the bivariate case, for example, two nonstationary series, y_{1t} and y_{2t} , are cointegrated if each can become stationary after taking first-order differences and if there exists a linear combination $z_t = \beta_1 y_{1t} + \beta_2 y_{2t}$ that is stationary. The intuition here is that the two series y_{1t} and y_{2t} obey a long-run equilibrium, yet they are allowed to deviate substantially from it in the short run. Note that for a multiple series $y_t = (y_{1t} \ y_{2t} \ \dots \ y_{kt})'$, there might be more than one linearly independent cointegrating relationship between the components of the series.

The second addition is the vector autoregressive fractionally integrated moving average (VARFIMA) model that you can use to model long-range dependent (LRD) time series. These are stationary time series, where the dependence between observations that are increasingly far apart in time dies out slowly. This behavior is often referred to as persistence or long memory, and it occurs in many macroeconomic and financial data, such as inflation and currency exchange rates. In the presence of long-memory behavior, vector autoregressive (VAR) models might not be parsimonious, because they require a large number of parameters in order to capture the dependence structure in the data; hence there is a need for an alternative model.

The rest of the paper is organized as follows. It begins by illustrating how to implement the Granger causality tests for nonstationary time series, which is often the start of the model selection process. Then restricted and unrestricted VECMs are fitted to a data set of coffee prices and their forecasting performance is compared. These techniques are extremely useful in the study and application of macroeconomics and financial econometrics, such as in the Federal Reserve Board's Comprehensive Capital Analysis and Review (CCAR). Following that, the focus shifts to the VARFIMA model, whose forecast performance is compared to that of a VAR model for a data set of two US inflation rates.

ERROR CORRECTION MODELS WITH OR WITHOUT CONSTRAINTS

When you are dealing with cointegrated series, the VECM,

$$\Delta y_t = \alpha(\beta' y_{t-1}) + \sum_{i=2}^p \Phi_i^* \Delta y_{t-i+1} + \varepsilon_t$$

might provide better interpretability than the VAR model, because the VECM captures both the long-run and short-run relationships between the components of the multiple series (the first and second terms, respectively, in the right-hand side of the VECM). Moreover, because Δy_t and $\beta' y_{t-1}$ are stationary, you can view the parameters α , Φ_i^* , $i = 2, \dots, p$, in the VECM as coefficients in a linear regression of stationary time series. For example, when $k = 2$ and $p = 2$, the VECM becomes

$$\begin{pmatrix} \Delta y_{1t} \\ \Delta y_{2t} \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \begin{pmatrix} \beta_1 & \beta_2 \end{pmatrix} \begin{pmatrix} y_{1,t-1} \\ y_{2,t-1} \end{pmatrix} + \begin{pmatrix} \phi_{2,11} & \phi_{2,12} \\ \phi_{2,21} & \phi_{2,22} \end{pmatrix} \begin{pmatrix} \Delta y_{1,t-1} \\ \Delta y_{2,t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}$$

or

$$\begin{pmatrix} \Delta y_{1t} \\ \Delta y_{2t} \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} (\beta_1 y_{1,t-1} + \beta_2 y_{2,t-1}) + \begin{pmatrix} \phi_{2,11} & \phi_{2,12} \\ \phi_{2,21} & \phi_{2,22} \end{pmatrix} \begin{pmatrix} \Delta y_{1,t-1} \\ \Delta y_{2,t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix}$$

This paper focuses on the mean parameters $(\alpha_1, \alpha_2, \beta_1, \beta_2, \phi_{2,11}, \phi_{2,12}, \phi_{2,21}, \phi_{2,22})$ and examines how the short-run relationship depicted in $(\phi_{2,11}, \phi_{2,12}, \phi_{2,21}, \phi_{2,22})$ is adjusted through (α_1, α_2) by the long-run relationship (β_1, β_2) .

A case of particular interest, for example, is when $\alpha_2 = \phi_{2,12} = 0$, where the series y_{1t} does not Granger-cause y_{2t} (the concept of Granger causality is revisited in greater detail later in this section). Is it safe to remove the two parameters $\alpha_2, \phi_{2,12}$ from the model to improve its forecast performance? Before SAS/ETS 14.1, omitting nonsignificant or zero-valued parameters from a VECM was not possible. The good news is that starting with SAS/ETS 14.1, the RESTRICT and TEST statements are supported for the VECMs. So if you think that some parameters should be zeros after testing, you can set constraints directly on them. In fact, besides improving forecast performance, the RESTRICT statement in the VECM might also help you perform hypothesis testing or score new data. This direction is not explored here, because it goes beyond the scope of the paper.

To illustrate the points that have been made so far, consider a data set of monthly coffee prices of the arabica and robusta varieties. The data are from the World Bank's Global Economic Monitor (GEM) commodities (World Bank 2016) and range from January 1976 to September 2016, for a total of 489 monthly observations. The following statements load the data into SAS® and plot the two series:

```
/* monthly data from January 1976 to September 2016 */
data coffee;
  input Arabica0 Robusta0;
  Arabica = 100*Arabica0; /*in unit cent*/
  Robusta = 100*Robusta0; /*in unit cent*/
  date = intnx( 'month', '01jan1976'd, _n_-1 );
  format date date9.;
datalines;
2.047000000000000 1.744476000000000
2.187400000000000 1.822297000000000
...more lines...
3.693620348000000 2.023620698000000
3.886745060000000 2.135835856000000
;

proc sgplot data=coffee;
  series x=date y=Arabica / lineattrs=(pattern=solid);
  series x=date y=Robusta / lineattrs=(pattern=dash);
  yaxis label="Coffee Price"; xaxis label="Date";
run;
```

Figure 1 displays the plot of the arabica and robusta coffee prices.

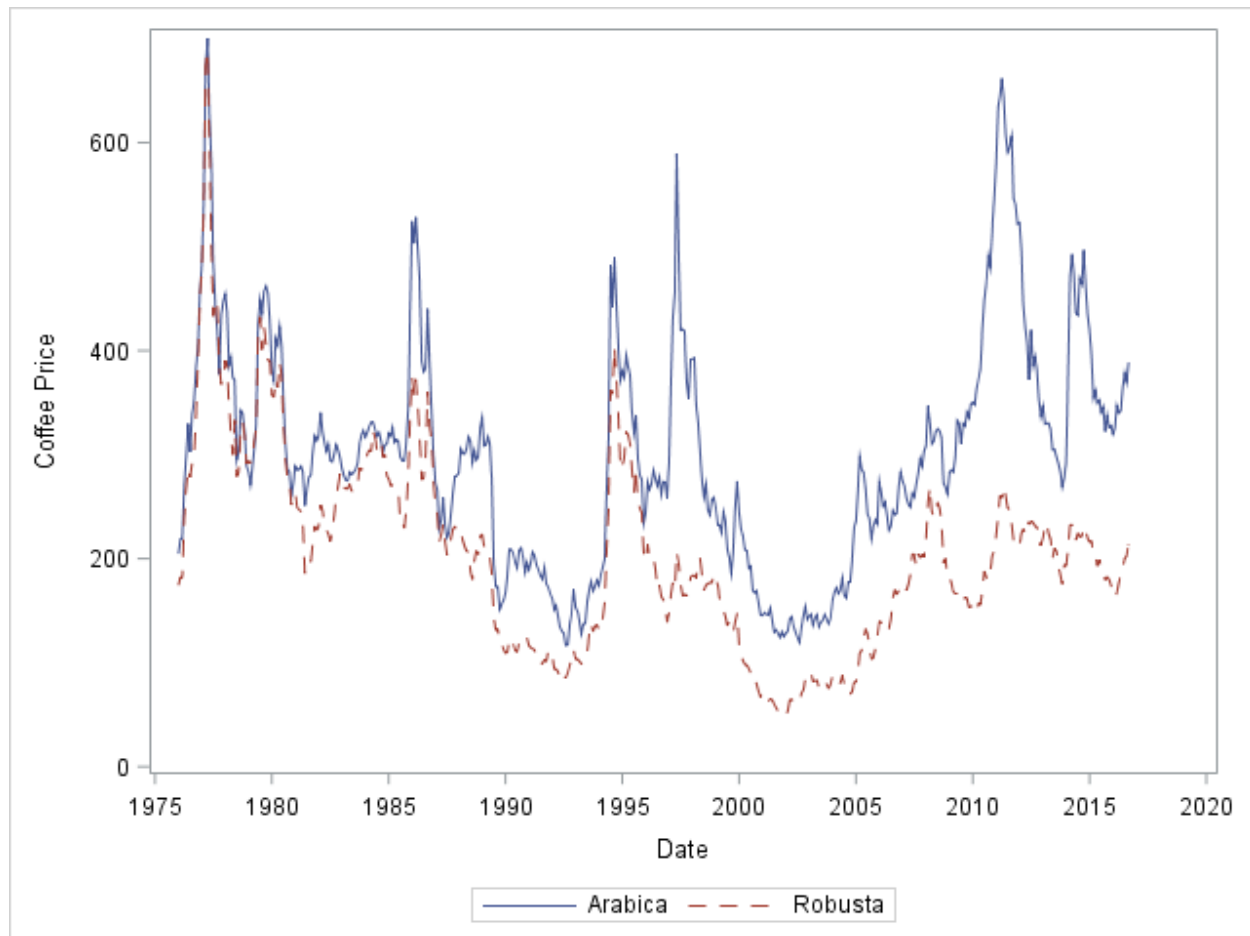


Figure 1. Plot of Arabica and Robusta Coffee Prices

For the initial task of variable selection, you can use the Granger causality test. Recall that if X and Y are two groups of time series, then X is said to Granger-cause Y if past X can help predict Y . In the presence of nonstationarities, testing for Granger causality becomes challenging because the Wald statistic, which you typically use to test for linear restrictions, does not follow the chi-square distribution. This, among other subtle issues related to Granger causality tests, is elegantly discussed in Dave Giles's blog (Giles 2011); Giles then implements the tests in a subset of the robusta-arabica prices data set and concludes that robusta coffee price does not Granger-cause arabica coffee price, whereas arabica price Granger-causes robusta price, at the 10% significance level. The following statements illustrate how you can perform these tests in a straightforward manner by using the VARMAX procedure:

```
proc varmax data=coffee(obs=423);
  model Arabica Robusta / p=7; /* estimate p lags but test (p-1) lags */
  test ar(1:6,1,2) = 0; /* H0: Robusta does not Granger-cause arabica */
  test ar(1:6,2,1) = 0; /* H0: Arabica does not Granger-cause robusta */
run;
```

Table 1 shows the test results. Then, at the 10% significance level, you cannot reject the null hypothesis of no causality from robusta to arabica price, but you can reject the null of no causality from arabica to robusta price.

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	6	8.67	0.1931
2	6	12.54	0.0510

Table 1. Granger Causality Test Results for 423-Month Data

It is important to exercise caution when interpreting these tests, especially when the sample size is small. For example, the following statements repeat the tests by using the prices from only the first 20 years:

```
proc varmax data=coffee(obs=240);
  model Arabica Robusta / p=7; /* estimate p lags but test (p-1) lags */
  test ar(1:6,1,2) = 0; /* H0: Robusta does not Granger-cause arabica */
  test ar(1:6,2,1) = 0; /* H0: Arabica does not Granger-cause robusta */
run;
```

The test results are shown in Table 2. Both null hypotheses are rejected at the 10% significance level.

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	6	21.72	0.0014
2	6	22.80	0.0009

Table 2. Granger Causality Test Results for 240-Month Data

Now, 40 years of data are used to perform the Granger causality tests:

```
proc varmax data=coffee(obs=480);
  model Arabica Robusta / p=7; /* estimate p lags but test (p-1) lags */
  test ar(1:6,1,2) = 0; /* H0: Robusta does not Granger-cause arabica */
  test ar(1:6,2,1) = 0; /* H0: Arabica does not Granger-cause robusta */
run;
```

As shown in Table 3, you should reject the null hypothesis of no causality from robusta to arabica price at the 10% significance level, but you cannot reject the null of no causality from arabica to robusta price—the opposite conclusion from that of using 423-month data in Giles's blog (Giles 2011). The message is clear: you need to be careful in inferring the Granger causality test results when the sample size is finite.

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	6	11.96	0.0628
2	6	10.35	0.1107

Table 3. Granger Causality Test Results for 480-Month Data

Nevertheless, the test results indicate that it is plausible that arabica and robusta prices might Granger-cause each other, so you can model them in the VECM. The following SAS code specifies the VECM on the full arabica and robusta coffee prices data set:

```
proc varmax data=coffee;
  id date interval=month;
  model Arabica Robusta / p=7;
  cointeg rank=1;
run;
```

The four parameters AR1_1_1, AR1_1_2, AR1_2_1, and AR1_2_2 are the elements of $\alpha\beta'$. For now the discussion skips them and focuses instead on the 24 short-run parameters: AR2_1_1, ..., AR7_2_2. As shown in the ODS table “Model Parameter Estimates” (which is omitted here), 15 of the 24 short-run parameters are nonsignificant at the 10% significance level. Can you eliminate all these nonsignificant parameters? To eliminate them, you must make sure that the null hypothesis that all those parameters are zeros cannot be rejected. The following statements perform the test:

```
proc varmax data=coffee;
  id date interval=month;
  model Arabica Robusta / p=7;
  cointeg rank=1;
  test ar(2:6,1,2)=0, ar(4:6,1,1)=0,
       ar(2,2,1)=0,   ar(4:6,2,1)=0, ar(4:6,2,2)=0;
run;
```

The test result is shown in Table 4. The null hypothesis is rejected at the 10% significance level, so you cannot eliminate all these nonsignificant parameters at the same time.

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	15	24.98	0.0502

Table 4. Joint Test Result of Null Hypothesis That 15 Parameters Are Zeros

After some trial and error, it is clear that for a special group of 13 short-run parameters, the null hypothesis that all of them are zeros cannot be rejected:

```
proc varmax data=coffee;
  id date interval=month;
  model Arabica Robusta / p=7;
  cointeg rank=1;
  test ar(2:6,1,2)=0, ar(4:5,2,1)=0, ar(2,2,1)=0,
       ar(4:5,2,2)=0, ar(4:6,1,1)=0;
run;
```

The test result is shown in Table 5, and obviously the null hypothesis at the 10% significance level is acceptable. In fact, the conclusion that all those 13 short-run parameters are zeros is very robust to the change in sample size.

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	13	13.21	0.4319

Table 5. Joint Test Result of Null Hypothesis That 13 Parameters Are Zeros

You can eliminate those 13 short-run parameters by using the RESTRICT statement:

```
proc varmax data=coffee;
  id date interval=month;
  model Arabica Robusta / p=7;
  cointeg rank=1;
  restrict ar(2:6,1,2)=0, ar(4:5,2,1)=0, ar(2,2,1)=0,
          ar(4:5,2,2)=0, ar(4:6,1,1)=0;
run;
```

As shown in the “Model Parameter Estimates” ODS table (which is omitted here), all remaining short-run parameters are significant at the 10% significance level except AR6_2_2. Although AR6_2_2 seems nonsignificant, it cannot be eliminated together with those 13 parameters, because the test rejects the null

hypothesis that all 14 parameters are zeros at the 10% significance level (the code and test result are omitted here).

Eliminating the 13 short-run nonsignificant parameters from the model is likely to improve its forecast performance. To verify this, you can use the root mean squared forecast error (RMSFE) criterion and the rolling-window method. The RMSFE is defined as

$$\text{RMSFE}^{(m,h)} = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_{T+h}^{(m,i)} - y_{T+h}^{(i)})^2}{n}}, \quad h = 1, \dots, H$$

where $y_{T+h}^{(i)}$ is the value of y_{T+h} in the i th window, $\hat{y}_{T+h}^{(m,i)}$ is its h -step-ahead forecast based on model m , and $\text{RMSFE}^{(m,h)}$ is the RMSFE of the h -step-ahead forecast for model m . The smaller the $\text{RMSFE}^{(m,h)}$, the better the forecast ability of model m for the h -step-ahead forecast. In the rolling-window method, there are n windows. In this case, there are $n = 244$ windows. Each window consists of two periods: an in-sample period and an out-of-sample period. The in-sample period is used for estimation, and the out-of-sample period is used to check the forecast ability. Each window has the same sample size, $T + H$, but starts at a different time point. For each window, the in-sample sample size, $T = 240$, and the out-of-sample sample size (that is, the number of forecast horizons), $H = 6$, 20 years of in-sample data are estimated in order to predict the next six months of out-of-sample data. In detail, the in-sample period of the i th window, $i = 1, \dots, 244$, starts $(i - 1)$ months later than January 1976 and ends $(i - 1)$ months later than December 1995. The out-of-sample period of the i th window, $i = 1, \dots, 244$, starts $(i - 1)$ months later than January 1996 and ends $(i - 1)$ months later than June 1996. The details are shown in Table 6.

Window Index	In-Sample Period		Out-of-Sample Period	
	From	To	From	To
1	January 1976	December 1995	January 1996	June 1996
2	February 1976	January 1996	February 1996	July 1996
3	March 1976	February 1996	March 1996	August 1996
...				
244	April 1996	March 2016	April 2016	September 2016

Table 6. In-Sample and Out-of-Sample Periods for Each Window

You can compare the forecast ability of two models: the VECM without constraints on any parameters, and the VECM with the constraint that the 13 short-run nonsignificant parameters are restricted to zeros. For easy reference, the first model is named model o and the second model is named model r ; that is, $m = o$ or $m = r$.

The following SAS macros estimate and forecast model o and model r for each window:

```

/* unrestricted VECM */
%macro oModel(firstObs, lastObs, lead, out);
  proc varmax data=coffee(firstobs=&firstObs. obs=&lastObs.) noprint;
    id date interval=month;
    model Arabica Robusta / p=7;
    cointeg rank=1;
    output out=&out.&firstObs. lead=&lead.;
  run;
%mend;

/* restricted ECM */
%macro rModel(firstObs, lastObs, lead, out);
  proc varmax data=coffee(firstobs=&firstObs. obs=&lastObs.) noprint;
    id date interval=month;
    model Arabica Robusta / p=7;

```

```

        cointeg rank=1;
        restrict ar(2:6,1,2)=0, ar(4:5,2,1)=0, ar(2,2,1)=0,
                ar(4:5,2,2)=0, ar(4:6,1,1)=0;
        output out=&out.&firstObs. lead=&lead.;
run;
%mend;

```

The following SAS macro implements the rolling-window method, which calculates and compares the RMSFEs for the two models:

```

/* rolling-window method */
%macro rollingWindow(windowLenth, firstWindow, lastWindow, lead, offset,
                    modell, model2, out1, out2, dataset, var1, var2);
%do iWindow = &firstWindow. %to &lastWindow.;
/* estimate and forecast models for each window */
%let firstObs=%eval(&offset.+&iWindow.);
%let lastObs =%eval(&offset.+&iWindow.+&windowLenth.-1);

/* estimate and forecast first model */
%&modell.(&firstObs., &lastObs., &lead., &out1.);

/* estimate and forecast second model */
%&model2.(&firstObs., &lastObs., &lead., &out2.);

/* save the forecasts */
proc iml;
    use &out1.&firstObs.;
    read all var{date for1 for2} into o;   close;
    use &out2.&firstObs.;
    read all var{for1 for2} into r;         close;
    use &dataset.;
    read all var{&var1. &var2.} into d;     close;

    rec          = J(&lead.,12,0);
    rec[,1]      = (1:&lead.)`;
    rec[,2]      = o[&windowLenth.+1:&windowLenth.+&lead.,1];
    rec[,3:4]    = d[&lastObs.+1:&lastObs.+&lead.,1:2];
    rec[,5:6]    = o[&windowLenth.+1:&windowLenth.+&lead.,2:3];
    rec[,7:8]    = r[&windowLenth.+1:&windowLenth.+&lead.,1:2];
    rec[,9:10]   = rec[,5:6] - rec[,3:4];
    rec[,11:12]  = rec[,7:8] - rec[,3:4];

    cn = {h date &var1. &var2. of1 of2 rf1 rf2
          ofd1 ofd2 rfd1 rfd2};
    if(&iWindow.=&firstWindow.) then
        create rwcomp from rec[colname=cn];
    else
        edit rwcomp;
    append from rec; close;
quit;
%end;

/* calculate RMSFEs and compare them */
proc sort data=rwcomp; by h; run;

proc iml;
    use rwcomp; read all into a; close;

```

```

rec=J(&lead.,5,0);
n = nrow(a)/&lead.;
do h=1 to &lead.;
  d = a[(h-1)*n+1:h*n,];
  rec[h,1] = h;
  rec[h,2] = sqrt(d[##, 9]/n);
  rec[h,3] = sqrt(d[##,11]/n);
  rec[h,4] = sqrt(d[##,10]/n);
  rec[h,5] = sqrt(d[##,12]/n);
end;

cn={h &out1.Rmsfe&var1. &out2.Rmsfe&var1.
    &out1.Rmsfe&var2. &out2.Rmsfe&var2.};
create rwres from rec[colname=cn]; append from rec; close;
quit;

proc print data=rwres noobs; run;
%mend;

```

The following statement calls the rolling-window macro to compare model o and model r :

```

/* compare the restricted and unrestricted ECMs */
%rollingWindow(240,1,244,6,0,
               oModel,rModel,o,r,coffee,Arabica,Robusta);

```

The final results of the comparison are shown in Table 7.

h	RMSFE for Arabica Coffee Price		RMSFE for Robusta Coffee Price	
	Model o	Model r	Model o	Model r
1	26.1007	24.8141	11.2156	10.5104
2	41.3720	39.3408	17.8706	17.1596
3	53.4722	50.2846	23.2839	22.0512
4	64.5390	61.0917	27.7987	26.3611
5	74.1871	71.1771	32.4952	31.1754
6	81.1872	78.9115	36.4965	35.6295

Table 7. Comparison of RMSFEs between VECMs with and without Constraints

As shown in Table 7, for all six forecast horizons and for both coffee varieties, the RMSFEs for model r are always smaller than the RMSFEs for model o . This means that the forecast ability of the restricted VECM dominates the forecast ability of the unrestricted VECM. Model selection pays off!

Besides the TEST and RESTRICT statements for the VECMs, the VARMAX procedure, starting in SAS/ETS 14.1, also supports the BOUND and INITIAL statements as well as more advanced features, such as the moving average terms (using the Q= option in the MODEL statement) and modeling the ARCH effects (using the GARCH statement) in the VECMs. These new features give you additional flexibility in modeling and forecasting nonstationary time series with cointegration.

LONG-MEMORY MODELING

The most popular model for analyzing a univariate LRD series, y_t , is the autoregressive fractionally integrated moving average (ARFIMA) model. It is defined as

$$\Phi(B)(I - B)^d y_t = \theta(B)\epsilon_t$$

where $\Phi(z)$, $\theta(z)$ are the typical AR and MA polynomials, B is the backshift operator, and d is the so-called long-range-dependent parameter. This parameter takes values in the interval $(-1/2, 1/2)$ and

controls the strength of dependence in the series. More specifically, if $0 < d < 1/2$, then the series is said to exhibit long memory, with values closer to $1/2$ corresponding to stronger dependence. If $d = 0$, the model reduces to the classic ARMA series, which has an exponentially decaying autocorrelation function and is therefore short-range-dependent. Finally, if $d < 0$, the series is said to have a special type of short memory called antipersistence.

The fractional differencing operator $(I - B)^d$ gives rise to the long-memory property when $0 < d < 1/2$, by producing a spectrum that explodes at frequencies near zero. As discussed in the seminal paper by Granger and Joyeux (1980), this operator comes as a third option in the dilemma of whether to difference a series or not. If a series has infinite variance, differencing can be an appropriate first step in building a model. However, even first-order differencing might eliminate important information at low frequencies, thus leading to possibly worse long-run forecasts. Hence, when you analyze stationary series, whose sample autocorrelation function decays slowly, you should consider the ARFIMA class of models.

One way to extend the ARFIMA model in the vector case, focusing for ease of presentation on bivariate series without AR and MA terms, is to write

$$(I - B)^D y_t = \begin{pmatrix} (I - B)^{d_1} y_{1t} \\ (I - B)^{d_2} y_{2t} \end{pmatrix} = \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix} = \epsilon_t$$

where $D = \text{diag}(d_1, d_2)$ and ϵ_t is bivariate white noise with the covariance matrix Σ . The nondiagonal entries of Σ can of course take nonzero values, and hence the model allows for dependence between the two series. Moreover, the LRD parameters, d_1 and d_2 , do not need to be equal, thus allowing for different degrees of memory in the two component series. Finally, short-memory effects can be captured by the full vector ARFIMA (VARFIMA) model,

$$(I - B)^D \Phi(B) y_t = \Theta(B) \epsilon_t$$

where $\Phi(z), \Theta(z)$ are now matrix-valued AR and MA polynomials of orders p and q respectively. This model is also referred to as VARFIMA(p, D, q).

When you are dealing with multiple series whose sample autocorrelation functions decay slowly (or analogously whose periodograms explode at frequencies near zero), a VARFIMA model might have better forecasting performance than a VARMA model, even when the VARMA model includes many AR or MA terms. Moreover, if the component series are correlated, then modeling them jointly is also likely to yield improved forecasts compared to those obtained by fitting separate univariate ARFIMA models.

In what follows, the forecasting performance of a VARFIMA model and a VAR model is compared for a bivariate series of two commonly used inflation measures, the consumer price index (CPI) and the personal consumption expenditure (PCE) price index. The data are collected by the Bureau of Labor Statistics and the Bureau of Economic Analysis, respectively, and consist of seasonally adjusted US inflation rates from January 1982 to November 2016. Although the CPI and the PCE both gauge consumer inflation, they are inherently different measures of prices, and being able to model and forecast their co-movements might be useful to analysts and policymakers.

The following statements load the data into SAS and produce the plots of the inflation rates shown in Figure 2:

```
data cpipe;
  input cpi pce;
  date = intnx( 'month', '01jan1982'd, _n_-1 );
  format date date9.;
datalines;
0.5376344086 0.5530973451
0.4278074866 0.7700770077
...more lines...
0.1940866501 0.0703432752
0.2591624162 0.3233516097
;
```

```

/*plot the series*/
proc sgplot data=cpi;
  series x=date y=cpi / lineattrs=(pattern=solid);
  series x=date y=pce / lineattrs=(pattern=dash);
  yaxis label = "Inflation Rate"; xaxis label = "Date";
run;

```

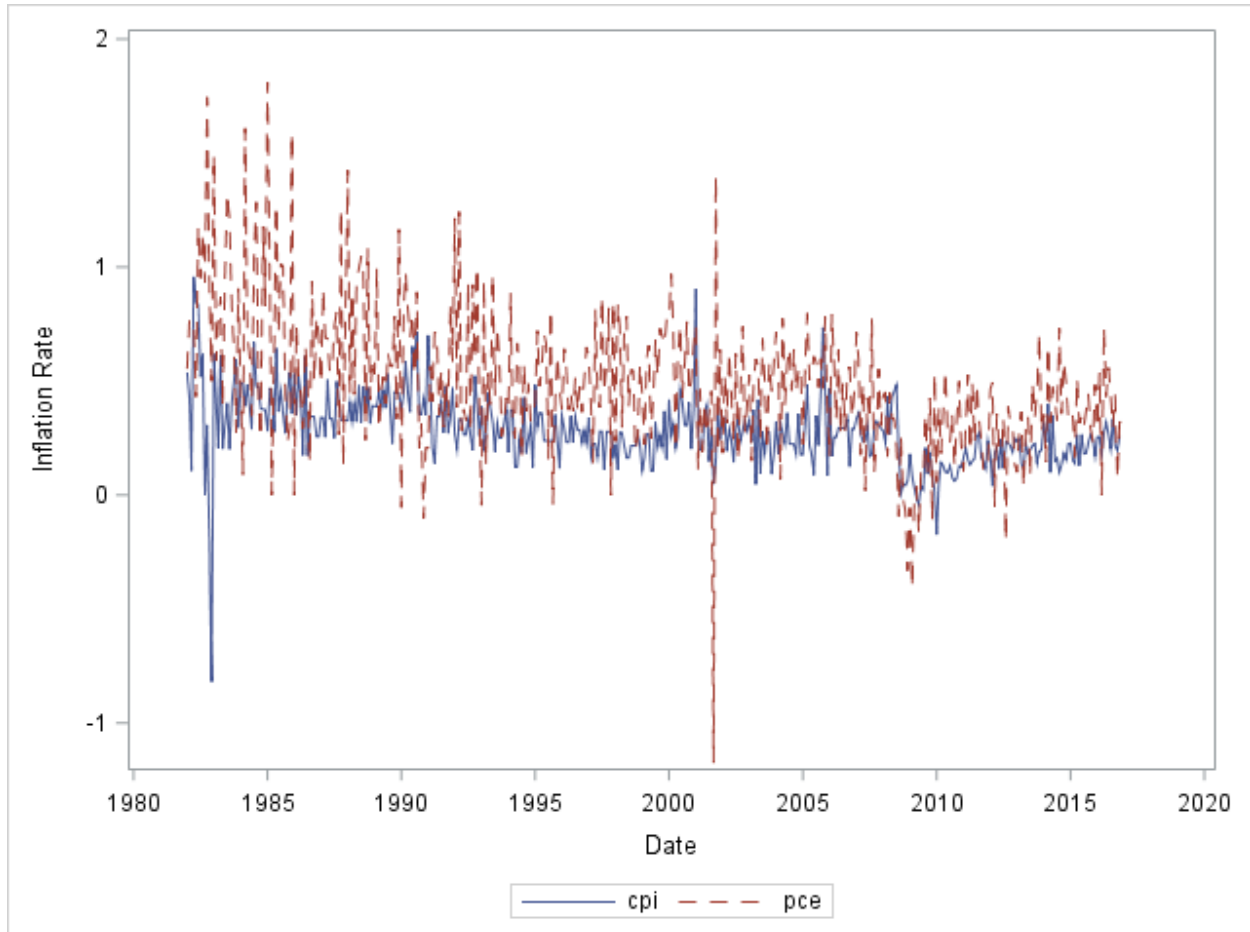


Figure 2. US CPI and PCE Monthly Rates from January 1982 to November 2016

The spectrum $f(\lambda)$ of each component of an LRD series behaves as $c\lambda^{-2d}$ for some $c > 0$ at low frequencies λ . As a result, in the log scale the spectrum follows the linear relation

$$\log f(\lambda) \approx \log c - 2d \log \lambda$$

for λ near zero. This property provides a simple diagnostic tool to detect long memory in a series. In particular, by regressing the values of the log-periodogram of an LRD series on small log-frequencies, you can expect the fitted line's slope to be equal to $-2d$. Hence, by solving for d you can obtain a model-free estimate of the LRD parameter. The following statements implement this procedure for the inflation rates data set, with 125 frequencies used in the regression:

```

%macro plotLogLogPeriodogram(d, v, f, l, m);
  proc spectra data=&d.(firstobs=&f. obs=&l.) out=spectra;
    var &v.;
  run;

```

```

/* convert to log scale */
data logspectra;
  set spectra(firstobs=2);
  /* Fourier frequencies */
  logfreq = log(2*constant('PI')*_N_/(&l.-&f.+1));
  /* log-periodograms */
  logpdg = log(P_01);
  /* select frequencies that will be used in regression */
  wt = (1<=_N_<=&m.);
  keep wt logfreq logpdg;
run;

/* linear regression of first m log-frequencies
   on corresponding log-periodogram values */
proc varmax data=logspectra(obs=&m.) outest=oe noprint;
  model logpdg = logfreq;
run;

/* save slope from regression */
data _null_;
  set oe;
  if _N_=1 then call symput('Slope', put(XL0_1, BEST6.));
run;

/* plot the periodograms in log-log scale */
proc sgplot data=logspectra;
  series x=logfreq y=logpdg / lineattrs=(pattern=solid);
  reg y=logpdg x=logfreq / nomarkers weight=wt
    lineattrs=(pattern=dash thickness=1 color='red');
  inset "Slope = &Slope." / position=topright
    textattrs=(color='red');
  xaxis label='log-frequency';
  yaxis label= "&y log-periodogram";
run;
%mend;

%plotLogLogPeriodogram(cpipce, cpi, 1, 419, 125);
%plotLogLogPeriodogram(cpipce, pce, 1, 419, 125);

```

The results are shown in Figure 3.

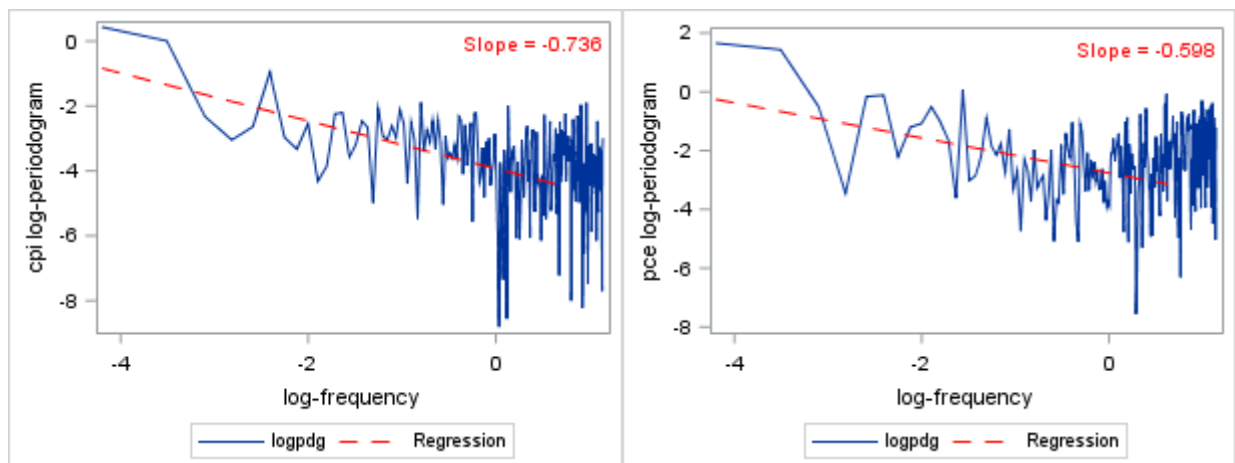


Figure 3. Log-Log Periodogram Plots for CPI and PCE Inflation Rate Series

The resulting LRD parameters are $d_{\text{CPI}} = 0.37$ and $d_{\text{PCE}} = 0.30$, hinting that the two series might have long memory. You should use these values of the LRD parameters as diagnostic tools or as initial points when you estimate the VARFIMA model. For more information, see Beran et al. (2013).

The following statements fit the VARFIMA model (with no AR, MA, and intercept terms) to the inflation data by specifying the FI option in the MODEL statement:

```
proc varmax data = cpipe;
  model cpi pce / noint fi;
run;
```

The model estimates for the LRD parameters and the entries of the covariance matrix are $d_1 = 0.3835$, $d_2 = 0.3199$, $\Sigma_{11} = 0.0194$, $\Sigma_{12} = 0.0073$, and $\Sigma_{22} = 0.0998$, all of which are significant at the 5% level. Because both the CPI and the PCE measure inflation, it is no surprise that the LRD parameter estimates do not differ by much. Note that the VARMAX procedure also supports the INITIAL statement for this model; this statement yields the same results if it is used in combination with the log-log periodogram initial estimates for the inflation rates data set:

```
initial d(1)=0.37, d(2)=0.3;
```

The model selection procedure that is used for the VECM is not repeated here, because the notion of Granger causality becomes more delicate in the presence of long memory. However, the typical information criteria (AIC, SBC, and so on) are supported for this model, and you can use them for model selection. Comparing, for example, a VAR(2) model with a VARFIMA(2, D , 0) model for the inflation rates data set, the latter is preferred with respect to both AIC and SBC. A more thorough and complete model selection study is omitted here, but note that the VARFIMA(2, D , 0) model was also selected by the SBC over the less parsimonious VAR(7) model (which had the smallest information criteria among VAR(p) models).

The following SAS macros estimate and forecast the VARFIMA(2, D , 0) and VAR(2) models:

```
/* VAR(2) model */
%macro arModel(firstObs, lastObs, lead, out);
  proc varmax data=cpipe(firstobs=&firstObs. obs=&lastObs.) noprint;
    id date interval=month;
    model cpi pce / p=2;
    output out=&out.&firstObs. lead=&lead.;
  run;
%mend;

/* VARFIMA(2,D,0) model */
%macro fiModel(firstObs, lastObs, lead, out);
  proc varmax data=cpipe(firstobs=&firstObs. obs=&lastObs.) noprint;
    id date interval=month;
    model cpi pce / p=2 fi;
    output out=&out.&firstObs. lead=&lead.;
  run;
%mend;
```

The following statement calls the rolling-window macro to compare the forecasting performance of the VARFIMA(2, D , 0) and VAR(2) models with 174 windows, each of which consists of a 240-month in-sample period and a 6-month out-of-sample period:

```
/* compare the two models */
%rollingWindow(240,1,174,6,0,
               arModel,fiModel,ar,fi,cpipce,cpi,pce);
```

The results are shown in Table 8.

h	RMSFE for CPI Rates		RMSFE for PCE Rates	
	VAR(2)	VARFIMA(2, D , 0)	VAR(2)	VARFIMA(2, D , 0)
1	0.10353	0.10264	0.22519	0.20095
2	0.10989	0.10602	0.22046	0.19721
3	0.11814	0.10818	0.24355	0.21062
4	0.12134	0.10890	0.24079	0.21401
5	0.12451	0.11051	0.24934	0.21970
6	0.12574	0.11041	0.24979	0.22452

Table 8. Comparison of RMSFEs between VARFIMA(2, D , 0) and VAR(2) Models

The VARFIMA(2, D , 0) model clearly has better forecasting performance (under the RMSFE) than the VAR(2) model for both series and for all forecast horizons.

The TEST and RESTRICT statements that are used for the VECM are also supported for the VARFIMA model starting with SAS/ETS 14.2. These statements are particularly helpful when you need to test the presence of long memory in some or all of the components of a multivariate series.

CONCLUSION

This paper illustrates several new features in SAS/ETS 14.1 and 14.2. In particular, it focuses on the VECM and the VARFIMA model, two popular and powerful tools for analyzing multivariate time series. For VECMs, Granger causality tests as well as parameter estimation with or without constraints are demonstrated. Moreover, the implementation of the rolling-window method under the RMSFE—a common way to compare the forecast ability of models—is illustrated for both models by using the data sets of arabica and robust coffee prices and US inflation rates (CPI and PCE). Finally, although not presented here, the SAS macro that is used to apply the rolling-window method can be easily extended for criteria other than RMSFE.

REFERENCES

- Beran, J., Feng, Y., Ghosh, S., and Kulik, R. (2013). *Long-Memory Processes: Probabilistic Properties and Statistical Methods*. Heidelberg: Springer.
- Giles, D. (2011). "Testing for Granger Causality." Accessed January 16, 2017. <http://davegiles.blogspot.com/2011/04/testing-for-granger-causality.html>.
- Granger, C. W. J., and Joyeux, R. (1980). "An Introduction to Long-Memory Time Series Models and Fractional Differencing." *Journal of Time Series Analysis* 1:15–29.
- World Bank. (2016). GEM Commodities. Accessed January 16, 2017. <http://data.worldbank.org/data-catalog/commodity-price-data>.

ACKNOWLEDGMENTS

The authors would like to thank Jan Chvosta for his valuable comments and suggestions and Ed Huddleston for his thorough editorial review.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Xilong Chen and Stefanos Kechagias
SAS Institute Inc.
Xilong.Chen@sas.com and Stefanos.Kechagias@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.