Chapter 10
Time Series Analysis and Control Examples

Chapter Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERVIEW</td>
<td>223</td>
</tr>
<tr>
<td>TIME SERIES ANALYSIS AND CONTROL SUBROUTINES</td>
<td>223</td>
</tr>
<tr>
<td>Getting Started</td>
<td>225</td>
</tr>
<tr>
<td>Syntax</td>
<td>249</td>
</tr>
<tr>
<td>Details</td>
<td>250</td>
</tr>
<tr>
<td>Example 10.1 VAR Estimation and Variance Decomposition</td>
<td>273</td>
</tr>
<tr>
<td>KALMAN FILTER SUBROUTINES</td>
<td>279</td>
</tr>
<tr>
<td>Example 10.2 Kalman Filtering: Likelihood Function Evaluation</td>
<td>280</td>
</tr>
<tr>
<td>Example 10.3 Kalman Filtering: Estimating an SSM Using the EM Algorithm</td>
<td>283</td>
</tr>
<tr>
<td>Example 10.4 Diffuse Kalman Filtering</td>
<td>288</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>291</td>
</tr>
</tbody>
</table>
Chapter 10
Time Series Analysis and Control
Examples

Overview

This chapter describes SAS/IML subroutines related to time series analysis including the TIMSAC subroutines adapted from parts of the TIMe Series Analysis and Control package developed by the Institute of Statistical Mathematics (ISM) in Japan, and routines for Kalman filtering and smoothing.

Time Series Analysis and Control Subroutines

This section describes a collection of SAS/IML subroutines for time series analysis. These subroutines are an adaptation of parts of the TIMe Series Analysis and Control (TIMSAC) package developed by the Institute of Statistical Mathematics (ISM) in Japan.

Selected routines from the TIMSAC package from ISM were converted by SAS Institute staff into SAS/IML routines under an agreement between SAS Institute and ISM. Credit for authorship of these TIMSAC SAS/IML routines goes to ISM, which has agreed to make them available to SAS users without charge.

There are four packages of TIMSAC programs. See the section “ISM TIMSAC Packages” on page 270 for more information on the TIMSAC package produced by ISM. Since these SAS/IML time series analysis subroutines are adapted from the corresponding FORTRAN subroutines in the TIMSAC package produced by ISM, they are collectively referred to as “the TIMSAC subroutines” in this chapter.

The subroutines analyze and forecast univariate and multivariate time series data. The nonstationary time series and seasonal adjustment models can also be analyzed by using the Interactive Matrix Language TIMSAC subroutines. These subroutines contain the Bayesian modeling of seasonal adjustment and changing spectrum estimation.

Discrete time series modeling has been widely used to analyze dynamic systems in economics, engineering, and statistics. The Box-Jenkins and Box-Tiao approaches are classical examples of unified time series analysis through identification, estimation, and forecasting (or control). The ARIMA procedure in the SAS/ETS product uses these approaches. Bayesian methods are being increasingly applied despite the controversial issues involved in choosing a prior distribution.

The fundamental idea of the Bayesian method is that uncertainties can be explained by probabilities. If there is a class model \( \Omega \) consisting of sets of member models \( \omega \), you can describe the uncertainty of \( \Omega \) using a prior distribution of \( \omega \). The mem-
ber model $\omega$ is directly related to model parameters. Let the prior probability density function be $p(\omega)$. When you observe the data $y$ that is generated from the model $\Omega$, the data distribution is described as $p(Y|\omega)$ given the unknown $\omega$ with a prior probability density $p(\omega)$, where the function $p(Y|\omega)$ is the usual likelihood function. Then the posterior distribution is the updated prior distribution given the sample information. The posterior probability density function is proportional to observed likelihood function $\times$ prior density function.

The IML TIMSAC subroutines contain various time series analysis and Bayesian models. Most of the subroutines are based on the minimum Akaike Information Criterion (AIC) or on the minimum ABIC method to determine the best model among alternative models. The TSBA YSEA subroutine is a typical example of Bayesian modeling. The following subroutines are supported:

- **TSBA YSEA**: Bayesian seasonal adjustment modeling
- **TSDECOMP**: time series decomposition analysis
- **TSMLOCAR**: locally stationary univariate AR model fitting
- **TSMLOMAR**: locally stationary multivariate AR model fitting
- **TSMULMAR**: multivariate AR model fitting
- **TSPERARS**: periodic AR model fitting
- **TSPRED**: ARMA model forecasting and forecast error variance
- **TSROOT**: polynomial roots or ARMA coefficients computation
- **TSTVCAR**: time-varying coefficient AR model estimation
- **TSUNIMAR**: univariate AR model fitting

For univariate and multivariate autoregressive model estimation, the least squares method is used. The least squares estimate is an approximate maximum likelihood estimate if error disturbances are assumed to be Gaussian. The least squares computation is performed by using the Householder transformation method. See the section, "Least Squares and Householder Transformation", for details.

The TSUNIMAR and TSMULMAR subroutines estimate the autoregressive models and select the appropriate AR order automatically by using the minimum AIC method. The TSMLOCAR and TSMLOMAR subroutines analyze the nonstationary time series data. The Bayesian time-varying AR coefficient model (TSTVCAR) offers another nonstationary time series analysis method. The state space and Kalman filter method is systematically applied to the smoothness priors models (TSDECOMP and TSTVCAR), which have stochastically perturbed difference equation constraints. The TSBA YSEA subroutine provides a way of handling Bayesian seasonal adjustment, and it can be an alternative to the SAS/ETS X-11 procedure. The TSBA YSEA subroutine employs the smoothness priors idea through constrained least squares estimation, while the TSDECOMP and TSTVCAR subroutines estimate the smoothness tradeoff parameters using the state space model and Kalman filter recursive computation. The TSPRED subroutine computes the one-step or multi-step predicted values of the ARMA time series model. In addition, the TSPRED subroutine computes forecast error variances and impulse response functions. The TSROOT subroutine
computes the AR and MA coefficients given the characteristic roots of the polynomial equation and the characteristic roots for the AR or MA model.

## Getting Started

### Minimum AIC Model Selection

The time series model is automatically selected using the AIC. The TSUNIMAR call estimates the univariate autoregressive model and computes the AIC. You need to specify the maximum lag or order of the AR process with the MAXLAG= option or put the maximum lag as the sixth argument of the TSUNIMAR call.

```sas
proc iml;
         2.179 1.653 1.832 2.328 2.737 3.014 3.328 3.404 2.981 2.557
         2.671 2.867 3.310 3.449 3.646 3.400 2.590 1.863 1.581 1.690
         1.771 2.274 2.576 3.111 3.605 3.543 2.769 2.021 2.185 2.588
         3.000 3.201 3.424 3.531 };
   call tsunimar(arcoeff, ev, nar, aic) data=y opt={-1 1} print=1
            maxlag=20;
```

You can also invoke the TSUNIMAR call as follows:

```sas
   call tsunimar(arcoeff, ev, nar, aic, y, 20, {-1 1}, ,1);
```

The optional arguments can be omitted. In this example, the argument MISSING is omitted, and thus the default option (MISSING=0) is used. The summary table of the minimum AIC method is displayed in Figure 10.1 and Figure 10.2. The final estimates are given in Figure 10.3.
Chapter 10. Time Series Analysis and Control Examples

<table>
<thead>
<tr>
<th>ORDER</th>
<th>INNOVATION VARIANCE</th>
<th>AIC(M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.31607294</td>
<td>-108.26753229</td>
</tr>
<tr>
<td>1</td>
<td>0.11481982</td>
<td>-201.45277331</td>
</tr>
<tr>
<td>2</td>
<td>0.04847420</td>
<td>-280.51201122</td>
</tr>
<tr>
<td>3</td>
<td>0.04828185</td>
<td>-278.88576251</td>
</tr>
<tr>
<td>4</td>
<td>0.04656506</td>
<td>-280.28905616</td>
</tr>
<tr>
<td>5</td>
<td>0.04615922</td>
<td>-279.1190502</td>
</tr>
<tr>
<td>6</td>
<td>0.04511943</td>
<td>-279.25356641</td>
</tr>
<tr>
<td>7</td>
<td>0.04312403</td>
<td>-281.50543541</td>
</tr>
<tr>
<td>8</td>
<td>0.04201118</td>
<td>-281.96304075</td>
</tr>
<tr>
<td>9</td>
<td>0.04128036</td>
<td>-281.61262868</td>
</tr>
<tr>
<td>10</td>
<td>0.03829179</td>
<td>-286.67686828</td>
</tr>
<tr>
<td>11</td>
<td>0.03318558</td>
<td>-298.13013264</td>
</tr>
<tr>
<td>12</td>
<td>0.03255171</td>
<td>-297.94298716</td>
</tr>
<tr>
<td>13</td>
<td>0.03247784</td>
<td>-296.15655602</td>
</tr>
<tr>
<td>14</td>
<td>0.03237083</td>
<td>-294.46677874</td>
</tr>
<tr>
<td>15</td>
<td>0.03234790</td>
<td>-292.5337704</td>
</tr>
<tr>
<td>16</td>
<td>0.03187416</td>
<td>-291.9201487</td>
</tr>
<tr>
<td>17</td>
<td>0.03183282</td>
<td>-290.0420196</td>
</tr>
<tr>
<td>18</td>
<td>0.03126946</td>
<td>-289.72064823</td>
</tr>
<tr>
<td>19</td>
<td>0.03087893</td>
<td>-288.90203735</td>
</tr>
<tr>
<td>20</td>
<td>0.02998019</td>
<td>-289.67854830</td>
</tr>
</tbody>
</table>

**Figure 10.1.** Minimum AIC Table - I

The minimum AIC order is selected as 11. Then the coefficients are estimated as in Figure 10.3. Note that the first 20 observations are used as presample values.
Figure 10.3. Minimum AIC Estimation

You can estimate the AR(11) model directly by specifying OPT={-10} and using the first 11 observations as presample values. The AR(11) estimates shown in Figure 10.4 are different from the minimum AIC estimates in Figure 10.3 because the samples are slightly different.

call tsunimar(arcoef,ev,nar,aic,y,11,{-1 0},,1);

Figure 10.4. AR(11) Estimation
The minimum AIC procedure can also be applied to the vector autoregressive (VAR) model using the TSMULMAR subroutine. See the section “Multivariate Time Series Analysis” on page 259 for details. Three variables are used as input. The maximum lag is specified as 10.

```sas
data one;
  input invest income consum @@;
datalines;
  ... data lines omitted ...
;
proc iml;
  use one;
  read all into y var{invest income consum};
  mdel = 1;
  maice = 2;
  misw = 0; /* instantaneous modeling ? */
  opt = mdel || maice || misw;
  maxlag = 10;
  miss = 0;
  print = 1;
  call tsmulmar(arcoef,ev,nar,aic,y,maxlag,opt,miss,print);
```

The VAR(3) model minimizes the AIC and was selected as an appropriate model (see Figure 10.5). However, AICs of the VAR(4) and VAR(5) models show little difference from VAR(3). You can also choose VAR(4) or VAR(5) as an appropriate model in the context of minimum AIC since this AIC difference is much less than 1.

```
ORDER INNOVATION VARIANCE
 M   LOG(|V(M)|)  AIC(M)
 0  25.98001095  2136.36089828
 1  15.70406486  1311.73331883
 2  15.48896746  1312.09533158
 3  15.18567834  1305.22562428
 4  14.96865183  1305.42944974
 5  14.74838535  1305.36759889
 6  14.60269347  1311.42086432
 7  14.54981887  1325.08514729
 8  14.38596333  1329.64899297
 9  14.16383772  1329.43469312
10  13.85377849  1322.00983656
```

Figure 10.5.  VAR Model Selection

<table>
<thead>
<tr>
<th>M</th>
<th>AIC(M)-AICMIN (truncated at 40.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>831.135274</td>
</tr>
<tr>
<td>1</td>
<td>-7.507695</td>
</tr>
<tr>
<td>2</td>
<td>-6.869707</td>
</tr>
<tr>
<td>3</td>
<td>0.203825</td>
</tr>
<tr>
<td>4</td>
<td>0.141975</td>
</tr>
<tr>
<td>5</td>
<td>6.195240</td>
</tr>
<tr>
<td>6</td>
<td>19.859523</td>
</tr>
<tr>
<td>7</td>
<td>24.423369</td>
</tr>
<tr>
<td>8</td>
<td>24.209069</td>
</tr>
<tr>
<td>9</td>
<td>16.784212</td>
</tr>
</tbody>
</table>

SAS OnlineDoc™: Version 8
The TSMULMAR subroutine estimates the instantaneous response model with diagonal error variance. See the section “Multivariate Time Series Analysis” on page 259 for details on the instantaneous response model. Therefore, it is possible to select the minimum AIC model independently for each equation. The best model is selected by specifying MAXLAG=5.

```sas
   call tsmulmar(arcoef, ev, nar, aic) data=y maxlag=5
    opt={1 1 0} print=1;
```

![Table: Innovation Variance Matrix](image)

![Table: AR-Coefficients](image)

AIC = 1347.619775

**Figure 10.6.** Model Selection via Instantaneous Response Model

You can print the intermediate results of the minimum AIC procedure using the PRINT=2 option.

Note that the AIC value depends on the MAXLAG=lag option and the number of parameters estimated. The minimum AIC VAR estimation procedure (MAICE=2) uses the following AIC formula:

\[
(T - \text{lag}) \log(|\hat{\Sigma}|) + 2(p \times n^2 + n \times \text{intercept})
\]

where \(p\) is the order of the \(n\)-variate VAR process, and \(\text{intercept}=1\) if the intercept is specified; otherwise, \(\text{intercept}=0\). When you use the MAICE=1 or MAICE=0 option, AIC is computed as the sum of AIC for each response equation. Therefore, there is an AIC difference of \(n(n - 1)\) since the instantaneous response model contains the additional \(n(n - 1)/2\) response variables as regressors.
The following code estimates the instantaneous response model. The results are shown in Figure 10.7.

```sas
call tsmulmar(arcoef, ev, nar, aic) data=y maxlag=3
   opt={1 0 0};
print aic nar;
print arcoef;
```

<table>
<thead>
<tr>
<th>AIC</th>
<th>NAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1403.0762</td>
<td>3</td>
</tr>
</tbody>
</table>

**ARCOEF**

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.8245814</td>
<td>5.3559216</td>
<td>17.066894</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8855926</td>
<td>0.3401741</td>
<td>-0.014398</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1684523</td>
<td>1.0502619</td>
<td>0.107064</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0891034</td>
<td>0.4591573</td>
<td>0.4473672</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.059195</td>
<td>-0.298777</td>
<td>0.1629818</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1128625</td>
<td>-0.044039</td>
<td>-0.088186</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1684932</td>
<td>-0.025847</td>
<td>-0.025671</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0637227</td>
<td>-0.196504</td>
<td>0.0695746</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.226559</td>
<td>0.0532467</td>
<td>-0.099808</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.303697</td>
<td>-0.139022</td>
<td>0.2576405</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 10.7.** AIC from Instantaneous Response Model

The following code estimates the VAR model. The results are shown in Figure 10.8.

```sas
call tsmulmar(arcoef, ev, nar, aic) data=y maxlag=3
   opt={1 2 0};
print aic nar;
print arcoef;
```

<table>
<thead>
<tr>
<th>AIC</th>
<th>NAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1397.0762</td>
<td>3</td>
</tr>
</tbody>
</table>

**ARCOEF**

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.8245814</td>
<td>5.3559216</td>
<td>17.066894</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8855926</td>
<td>0.3401741</td>
<td>-0.014398</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1684523</td>
<td>1.0502619</td>
<td>0.107064</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0891034</td>
<td>0.4591573</td>
<td>0.4473672</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.059195</td>
<td>-0.298777</td>
<td>0.1629818</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1128625</td>
<td>-0.044039</td>
<td>-0.088186</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1684932</td>
<td>-0.025847</td>
<td>-0.025671</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0637227</td>
<td>-0.196504</td>
<td>0.0695746</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.226559</td>
<td>0.0532467</td>
<td>-0.099808</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.303697</td>
<td>-0.139022</td>
<td>0.2576405</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 10.8.** AIC from VAR Model

The AIC computed from the instantaneous response model is greater than that obtained from the VAR model estimation by 6. There is a discrepancy between Figure 10.8 and Figure 10.5 because different observations are used for estimation.
Nonstationary Data Analysis

This example shows how to manage nonstationary data using TIMSAC calls. In practice, time series are considered to be stationary when the expected values of first and second moments of the series do not change over time. This weak or covariance stationarity can be modeled using the TSMLOCAR, TSMLOMAR, TSDECOMP, and TSTVCAR subroutines.

First, the locally stationary model is estimated. The whole series (1000 observations) is divided into three blocks of size 300 and one block of size 90, and the minimum AIC procedure is applied to each block of the data set. See the “Nonstationary Time Series” section on page 256 for more details.

```
data one;
  input y @@;
datalines;
... data lines omitted ...
;
proc iml;
  use one;
  read all var{y};

  mdel = -1;
  lspan = 300; /* local span of data */
  maice = 1;
  opt = mdel || lspan || maice;
  call tsmlocar(arcoef,ev,nar,aic,first,last)
    data=y maxlag=10 opt=opt print=2;
```

Estimation results are displayed with the graphs of power spectrum ($\log(10(f_{YY}(g)))$), where $f_{YY}(g)$ is a rational spectral density function. See the “Spectral Analysis” section on page 261. As the first block and the second block do not have any sizable difference, the pooled model (AIC=45.892) is selected instead of the moving model (AIC=46.957) in Figure 10.10. However, you can notice a slight change in the shape of the spectrum of the third block of the data (observations 611 through 910). See Figure 10.11 on page 234 and Figure 10.13 on page 236 for comparison. The moving model is selected since the AIC (106.830) of the moving model is smaller than that of the pooled model (108.867).
INITIAL LOCAL MODEL: N_CURR = 300
    NAR_CURR = 8
    AIC = 37.583203

..........................CURRENT MODEL..........................

. MAR Coefficients: AR(M) .

1 1.605717 .
2 -1.245350 .
3 1.014847 .
4 -0.931554 .
5 0.394230 .
6 -0.004344 .
7 0.111608 .
8 -0.124992 .

AIC = 37.5832030 .
Innovation Variance = 1.067455 .

INPUT DATA START = 11 FINISH = 310 .

Figure 10.9.  Locally Stationary Model for First Block
--- THE FOLLOWING TWO MODELS ARE COMPARED ---

**MOVING MODEL:**  
(N_PREV = 300, N_CURR = 300)  
NAR_CURR = 7  
AIC = 46.957398

**CONSTANT MODEL:**  
N_POOLED = 600  
NAR_POOLED = 8  
AIC = 45.892350

***** CONSTANT MODEL ADOPTED *****

CURRENT MODEL

M  | AR Coefficients: AR(M)  
--- | ---  
1  | 1.593890  
2  | -1.262379  
3  | 1.013733  
4  | -0.926052  
5  | 0.314480  
6  | 0.193973  
7  | -0.058043  
8  | -0.078508

AIC = 45.8923501  
Innovation Variance = 1.047585

INPUT DATA START = 11  FINISH = 610

Figure 10.10.  Locally Stationary Model Comparison
Figure 10.11. Power Spectrum for First and Second Blocks
--- THE FOLLOWING TWO MODELS ARE COMPARED ---

MOVING MODEL: (N_PREV = 600, N_CURR = 300)
   NAR_CURR = 7
   AIC = 106.829869
CONSTANT MODEL: N_POOLED = 900
   NAR_POOLED = 8
   AIC = 108.867091

*************************************
***** ****
***** NEW MODEL ADOPTED *****
***** ****
*************************************

..................CURRENT MODEL..................

M   AR Coefficients: AR(M)

  1  1.648544
  2 -1.201812
  3  0.674933
  4 -0.567576
  5 -0.018924
  6  0.516627
  7 -0.283410

AIC = 60.9375188
Innovation Variance = 1.161592

INPUT DATA START = 611 FINISH = 910

-----------------------------------

Figure 10.12.  Locally Stationary Model for Third Block
Finally, the moving model is selected since there is a structural change in the last block of data (observations 911 through 1000). The final estimates are stored in variables ARCOEF, EV, NAR, AIC, FIRST, and LAST. The final estimates and spectrum are given in Figure 10.14 and Figure 10.15, respectively. The power spectrum of the final model (Figure 10.15) is significantly different from that of the first and second blocks (see Figure 10.11).
--- THE FOLLOWING TWO MODELS ARE COMPARSED ---

MOVING MODEL: (N_PREV = 300, N_CURR = 90)
   NAR_CURR = 6
   AIC = 139.579012

CONSTANT MODEL: N_POOLED = 390
   NAR_POOLED = 9
   AIC = 167.783711

*************************************
***** *****
***** NEW MODEL ADOPTED *****
***** *****
*************************************

..................................CURRENT MODEL..................................


M   AR Coefficients: AR(M)
  1   1.181022
  2  -0.321178
  3  -0.113001
  4  -0.137846
  5  -0.141799
  6   0.260728

AIC = 78.6414932
Innovation Variance = 2.050818

INPUT DATA  START = 911  FINISH = 1000

..................................CURRENT MODEL..................................


Figure 10.14. Locally Stationary Model for Last Block
Figure 10.15. Power Spectrum for Last Block

The multivariate analysis for locally stationary data is a straightforward extension of the univariate analysis. The bivariate locally stationary VAR models are estimated. The selected model is the VAR(7) process with some zero coefficients over the last block of data. There seems to be a structural difference between observations from 11 to 610 and those from 611 to 896.

```sas
proc iml;
    rudder = {. . . data lines omitted . . .};
    yawing = {. . . data lines omitted . . .};
    y = rudder' || yawing';
    c = {0.01795 0.02419};
    /*-- calibration of data --*/
```
\[ y_t = y \# (c @ j(n,1,1)); \]
\[ \text{mdel} = -1; \]
\[ \text{lspan} = 300; /* \text{local span of data} */ \]
\[ \text{maice} = 1; \]
\[ \text{call tsmlomar(arcoef,ev,nar,aic,first,last) data=y maxlag=10} \]
\[ \text{opt = (mdel || lspan || maice) print=1; } \]

--- THE FOLLOWING TWO MODELS ARE COMPARED ---

**MOVING MODEL:** (N_PREV = 600, N_CURR = 286)
- \( \text{AIC} = -823.845234 \)
- \( \text{NAR_CURR} = 7 \)

**CONSTANT MODEL:** N_POOLED = 886
- \( \text{NAR_POOLED} = 10 \)
- \( \text{AIC} = -716.818588 \)

****************************
***** NEW MODEL ADOPTED *****
******
****************************

..........................CURRENT MODEL.........................

\[ \begin{array}{cccc}
M & \text{AR Coefficients} \\
1 & 0.932904 & -0.130964 \\
   & -0.024401 & 0.599483 \\
2 & 0.163141 & 0.266876 \\
   & -0.135605 & 0.377923 \\
3 & -0.322283 & 0.178194 \\
   & 0.188603 & -0.081245 \\
4 & 0.166094 & -0.304755 \\
   & -0.084626 & -0.180638 \\
5 & 0 & 0 \\
   & 0 & -0.036958 \\
6 & 0 & 0 \\
   & 0 & 0.034578 \\
7 & 0 & 0 \\
   & 0 & 0.268414 \\
\end{array} \]

\[ \text{AIC} = -114.6911872 \]

\[ \text{Innovation Variance} \]
\[ \begin{array}{cc}
1.069929 & 0.145558 \\
0.145558 & 0.563985 \\
\end{array} \]

\[ \text{INPUT DATA START = 611 FINISH = 896} \]

--- THE FOLLOWING TWO MODELS ARE COMPARED ---

**MOVING MODEL:** (N_PREV = 600, N_CURR = 286)
- \( \text{AIC} = -823.845234 \)
- \( \text{NAR_CURR} = 7 \)

**CONSTANT MODEL:** N_POOLED = 886
- \( \text{NAR_POOLED} = 10 \)
- \( \text{AIC} = -716.818588 \)

****************************
***** NEW MODEL ADOPTED *****
******
****************************

Figure 10.16. Locally Stationary VAR Model Analysis

Consider the time series decomposition

\[ y_t = T_t + S_t + u_t + \epsilon_t \]
where $T_t$ and $S_t$ are trend and seasonal components, respectively, and $u_t$ is a stationary AR(p) process. The annual real GNP series is analyzed under second difference stochastic constraints on the trend component and the stationary AR(2) process.

$$
T_t = 2T_{t-1} - T_{t-2} + w_{1t}
$$
$$
u_t = \alpha_1 u_{t-1} + \alpha_2 u_{t-2} + w_{2t}
$$

The seasonal component is ignored if you specify SORDER=0. Therefore, the following state space model is estimated:

$$
y_t = H z_t + \epsilon_t
$$
$$
z_t = F z_{t-1} + w_t
$$

where

$$
H = \begin{bmatrix}
1 & 0 & 1 & 0
\end{bmatrix}
$$
$$
F = \begin{bmatrix}
2 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & \alpha_1 & \alpha_2 \\
0 & 0 & 1 & 0
\end{bmatrix}
$$

$$
z_t = (T_t, T_{t-1}, u_t, u_{t-1})'
$$
$$
w_t = (w_{1t}, w_{2t}, 0)' \sim \begin{pmatrix}
\sigma_1^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & \sigma_2^2 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
$$

The parameters of this state space model are $\sigma_1^2$, $\sigma_2^2$, $\alpha_1$, and $\alpha_2$.

```sas
proc iml;
    y = { 116.8 120.1 123.2 130.2 131.4 125.6 124.5 134.3
          135.2 151.8 146.4 139.0 127.8 147.0 165.9 165.5
          179.4 190.0 189.8 190.9 203.6 183.5 169.3 144.2
          141.5 154.3 169.5 193.0 203.2 192.9 209.4 227.2
          263.7 297.8 337.1 361.3 355.2 312.6 309.9 327.7
          324.1 355.3 383.4 395.1 412.8 406.0 438.0 446.1
          452.5 447.3 475.9 487.7 497.2 529.8 551.0 581.1
          617.8 658.1 675.2 706.6 724.7 }
    y = y'; /* convert to column vector */
    mdel = 0;
    trade = 0;
    tvreg = 0;
    year = 0;
    period = 0;
    log = 0;
    maxit = 100;
    update = .; /* use default update method */
```
line = .; /* use default line search method */
sigmax = 0; /* no upper bound for variances */
back = 100;
opt = mdel || trade || year || period || log || maxit ||
    update || line || sigmax || back;
call tsdecomp(cmp,coef,aic) data=y order=2 sorder=0 nar=2
    npred=5 opt=opt icmp={1 3} print=1;

The estimated parameters are printed when you specify the PRINT= option. In Figure
10.17, the estimated variances are printed under the title of TAU2(I), showing that
\( \hat{\sigma}_1^2 = 2.915 \) and \( \hat{\sigma}_2^2 = 113.9577 \). The AR coefficient estimates are \( \hat{\alpha}_1 = 1.397 \) and
\( \hat{\alpha}_2 = -0.595 \). These estimates are also stored in the output matrix COEF.

<<< Final Estimates >>>

--- PARAMETER VECTOR ---

1.607426E-01 6.283837E+00 8.761628E-01 -5.94879E-01

--- GRADIENT ---

3.385021E-04 5.760929E-06 3.029534E-04 -1.18396E-04

LIKELIHOOD = -249.937193 SIG2 = 18.135052
AIC = 509.874385

<table>
<thead>
<tr>
<th>I</th>
<th>TAU2(I)</th>
<th>AR(I)</th>
<th>PARCOR(I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.915075</td>
<td>1.397374</td>
<td>0.876163</td>
</tr>
<tr>
<td>2</td>
<td>113.957712</td>
<td>-0.594879</td>
<td>-0.594879</td>
</tr>
</tbody>
</table>

Figure 10.17. Nonstationary Time Series and State Space Modeling

The trend and stationary AR components are estimated using the smoothing method,
and out-of-sample forecasts are computed using a Kalman filter prediction algorithm.
The trend and AR components are stored in the matrix CMP since the ICMP={1 3}
option is specified. The last 10 observations of the original series Y and the last 15
observations of two components are shown in Figure 10.18. Note that the first column
of CMP is the trend component and the second column is the AR component. The
last 5 observations of the CMP matrix are out-of-sample forecasts.
Chapter 10. Time Series Analysis and Control Examples

<table>
<thead>
<tr>
<th>Y</th>
<th>CMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>487.7</td>
<td>514.01142</td>
</tr>
<tr>
<td>497.2</td>
<td>532.62744</td>
</tr>
<tr>
<td>529.8</td>
<td>552.02402</td>
</tr>
<tr>
<td>551</td>
<td>571.90122</td>
</tr>
<tr>
<td>581.1</td>
<td>592.31944</td>
</tr>
<tr>
<td>617.8</td>
<td>613.21855</td>
</tr>
<tr>
<td>658.1</td>
<td>634.43665</td>
</tr>
<tr>
<td>675.2</td>
<td>655.70431</td>
</tr>
<tr>
<td>706.6</td>
<td>677.2125</td>
</tr>
<tr>
<td>724.7</td>
<td>698.72364</td>
</tr>
<tr>
<td>720.2</td>
<td>700.72364</td>
</tr>
<tr>
<td>741.7</td>
<td>741.72364</td>
</tr>
<tr>
<td>763.2</td>
<td>763.22364</td>
</tr>
<tr>
<td>784.7</td>
<td>784.72364</td>
</tr>
<tr>
<td>806.2</td>
<td>806.22364</td>
</tr>
</tbody>
</table>

Figure 10.18. Smoothed and Predicted Values of Two Components

**Seasonal Adjustment**

Consider the simple time series decomposition

\[ y_t = T_t + S_t + \epsilon_t \]

The TSBAYSEA subroutine computes seasonally adjusted series by estimating the seasonal component. The seasonally adjusted series is computed as \( y_t^* = y_t - \hat{S}_t \). The details of the adjustment procedure are given in the section “Bayesian Seasonal Adjustment” on page 254.

The monthly labor force series (1972–1978) are analyzed. You do not need to specify the options vector if you want to use the default options. However, you should change OPT[2] when the data frequency is not monthly (OPT[2]=12). The NPRED= option produces the multistep forecasts for the trend and seasonal components. The stochastic constraints are specified as ORDER=2 and SORDER=1.

\[
T_t = 2T_{t-1} - T_{t-2} + w_{1t} \\
S_t = -S_{t-1} - \cdots - S_{t-11} + w_{2t}
\]

The seasonal components are shown in Figure 10.19 on page 243, and the adjusted series are shown in Figure 10.20 on page 243. The estimated spectral density function of the irregular series \( \hat{\epsilon}_t \) is shown in Figure 10.21 on page 244.

```sas
proc iml;
    y =
        { 5447  5412  5215  4697  4344  5426
          5173  4857  4658  4470  4268  4116
          4675  4845  4512  4174  3799  4847
          4550  4208  4165  3763  4056  4058
          5008  5140  4755  4301  4144  5380
          5260  4885  5202  5044  5685  6106
          8180  8309  8359  7820  7623  8569
          8209  7696  7522  7244  7231  7195
          8174  8033  7525  6890  6304  7655
        ;
```
Getting Started

7577 7322 7026 6833 7095 7022
7848 8109 7556 6568 6151 7453
6941 6757 6437 6221 6346 5880

\[
y = y';
\]

```
call tsbaysea(trend, season, series, adj, abic)
data=y order=2 sorder=1 npred=12 print=2;
print trend season series adj abic;
```

### Seasonal Component

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>576.866752</td>
<td>612.796066</td>
</tr>
<tr>
<td>-572.556158</td>
<td>493.248873</td>
</tr>
<tr>
<td>-223.927593</td>
<td>-440.622170</td>
</tr>
<tr>
<td>567.417780</td>
<td>649.108143</td>
</tr>
<tr>
<td>-209.010499</td>
<td>-511.275202</td>
</tr>
<tr>
<td>647.626707</td>
<td>686.576003</td>
</tr>
<tr>
<td>-582.439797</td>
<td>516.512576</td>
</tr>
<tr>
<td>-212.853209</td>
<td>-538.237178</td>
</tr>
<tr>
<td>749.318446</td>
<td>705.520212</td>
</tr>
<tr>
<td>-617.748290</td>
<td>506.336574</td>
</tr>
<tr>
<td>254.706508</td>
<td>510.461942</td>
</tr>
<tr>
<td>711.125340</td>
<td>748.595903</td>
</tr>
<tr>
<td>-700.824658</td>
<td>519.764643</td>
</tr>
<tr>
<td>-288.809493</td>
<td>-509.321443</td>
</tr>
<tr>
<td>650.134120</td>
<td>800.460271</td>
</tr>
<tr>
<td>-719.314201</td>
<td>553.049123</td>
</tr>
<tr>
<td>-295.332122</td>
<td>-487.701411</td>
</tr>
<tr>
<td>650.770701</td>
<td>800.937334</td>
</tr>
<tr>
<td>-719.114602</td>
<td>553.197644</td>
</tr>
<tr>
<td>-295.274714</td>
<td>-487.662081</td>
</tr>
</tbody>
</table>

*** Last 12 Values Are Forecasted ***

---

**Figure 10.19.** Seasonal Component Estimates and Forecasts

<table>
<thead>
<tr>
<th>Adjusted = Data - Seasonal - Trading_Day_Comp - OCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>4870.133248</td>
</tr>
<tr>
<td>4916.556158</td>
</tr>
<tr>
<td>4881.927593</td>
</tr>
<tr>
<td>4107.582220</td>
</tr>
<tr>
<td>4366.242588</td>
</tr>
<tr>
<td>4374.010499</td>
</tr>
<tr>
<td>4360.373293</td>
</tr>
<tr>
<td>4726.439797</td>
</tr>
<tr>
<td>5414.583209</td>
</tr>
<tr>
<td>7430.681554</td>
</tr>
<tr>
<td>8240.748290</td>
</tr>
<tr>
<td>7776.706508</td>
</tr>
<tr>
<td>7462.874660</td>
</tr>
<tr>
<td>7004.824658</td>
</tr>
<tr>
<td>7314.809493</td>
</tr>
<tr>
<td>7197.865880</td>
</tr>
<tr>
<td>6870.314201</td>
</tr>
<tr>
<td>6732.332122</td>
</tr>
</tbody>
</table>

---

**Figure 10.20.** Seasonally Adjusted Series
Figure 10.21.  Spectrum of Irregular Component

X: If peaks (troughs) appear at these frequencies, try lower (higher) values of rigid and watch ABIC
T: If a peaks appears here try trading-day adjustment
Miscellaneous Time Series Analysis Tools

The forecast values of multivariate time series are computed using the TSPRED call. In this example, the multistep ahead forecasts are produced from the VARMA(2,1) estimates. Since the VARMA model is estimated using the mean deleted series, you should specify the CONSTANT=1 option. You need to provide the original series instead of the mean deleted series to get the correct predictions. The forecast variance MSE and the impulse response function IMPULSE are also produced.

The VARMA\((p,q)\) model is written

\[
y_t + \sum_{i=1}^{p} A_i y_{t-i} = \epsilon_t + \sum_{i=1}^{q} M_i \epsilon_{t-i}
\]

Then the COEF matrix is constructed by stacking matrices \(A_1, \ldots, A_p, M_1, \ldots, M_q\).

```sas
proc iml;
    c = { 264 235 239 275 277 274 334 334 306
          308 309 295 271 277 221 223 227 215 223
          241 250 270 303 311 307 322 335 335 334
          309 262 228 191 188 215 215 249 291 296
    };
    f = { 690 690 688 690 702 702 702 700 700 702
          702 694 708 702 708 700 700 702 694
          698 694 700 702 700 708 708 710 704
          704 700 700 694 704 710 710 710 708
    };
    t = { 1152 1288 1288 1288 1368 1456 1656 1496 1744 1464
          1560 1376 1336 1336 1296 1296 1280 1264 1280 1272
          1344 1328 1352 1480 1472 1600 1512 1456 1368 1280
          1224 1112 1112 1048 1176 1064 1168 1280 1336 1248
    };
    p = { 254.14 253.12 251.85 250.41 249.09 249.19 249.52 250.19
          248.74 248.41 249.95 250.64 250.87 250.94 250.96 251.33
          251.18 251.05 251.00 250.99 250.79 250.44 250.12 250.19
          249.77 250.27 250.74 250.90 252.21 253.68 254.47 254.80
          254.92 254.96 254.96 254.96 254.96 254.54 253.21 252.08
    };
    y = c' || f' || t' || p';
    ar = { .82028 -.97167 .07938 6.4382,
            -.39983 .94448 .02793 -1.7477,
            -.42278 -2.3314 1.4682 -70.996,
            .031038 -.019231 -.0004904 1.3677,
            -.029811 .89262 -.047579 4.7873,
            .31476 .0061959 -.012221 1.4921,
            .3813 2.7182 -.52993 67.711,
            -.020818 .01764 .00037981 -.38154
    };
    ma = { .083035 -1.0509 .055898 -3.9778,
            -.40452 .36876 .026369 -.81146,
            .062379 -2.6506 .80784 -76.952,
            .03273 -.031555 -.00019776 -.025205
    };
    coef = ar // ma;
    ev = { 188.55 6.8082 42.385 .042942,
            6.8082 32.169 37.995 -.062341,
            42.385 37.995 5138.8 -.10757,
            .042942 -.062341 -.10757 .34313
    );
SAS OnlineDoc™: Version 8
nar = 2; nma = 1;
call tspred(forecast, impulse, mse, y, coef, nar, nma, ev,
5, nrow(y), -1);

<table>
<thead>
<tr>
<th>OBSERVED</th>
<th>PREDICTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>Y2</td>
</tr>
<tr>
<td>264</td>
<td>690</td>
</tr>
<tr>
<td>235</td>
<td>690</td>
</tr>
<tr>
<td>239</td>
<td>688</td>
</tr>
<tr>
<td>239</td>
<td>690</td>
</tr>
<tr>
<td>275</td>
<td>694</td>
</tr>
<tr>
<td>277</td>
<td>702</td>
</tr>
<tr>
<td>274</td>
<td>702</td>
</tr>
<tr>
<td>334</td>
<td>702</td>
</tr>
<tr>
<td>334</td>
<td>700</td>
</tr>
<tr>
<td>306</td>
<td>702</td>
</tr>
<tr>
<td>308</td>
<td>702</td>
</tr>
<tr>
<td>309</td>
<td>694</td>
</tr>
<tr>
<td>295</td>
<td>708</td>
</tr>
<tr>
<td>271</td>
<td>702</td>
</tr>
<tr>
<td>277</td>
<td>702</td>
</tr>
<tr>
<td>221</td>
<td>708</td>
</tr>
<tr>
<td>223</td>
<td>700</td>
</tr>
<tr>
<td>227</td>
<td>700</td>
</tr>
<tr>
<td>215</td>
<td>702</td>
</tr>
<tr>
<td>223</td>
<td>694</td>
</tr>
<tr>
<td>241</td>
<td>698</td>
</tr>
<tr>
<td>250</td>
<td>694</td>
</tr>
<tr>
<td>270</td>
<td>700</td>
</tr>
<tr>
<td>303</td>
<td>702</td>
</tr>
<tr>
<td>311</td>
<td>700</td>
</tr>
<tr>
<td>307</td>
<td>702</td>
</tr>
<tr>
<td>322</td>
<td>708</td>
</tr>
<tr>
<td>335</td>
<td>708</td>
</tr>
<tr>
<td>335</td>
<td>710</td>
</tr>
<tr>
<td>334</td>
<td>704</td>
</tr>
<tr>
<td>309</td>
<td>704</td>
</tr>
<tr>
<td>262</td>
<td>700</td>
</tr>
<tr>
<td>228</td>
<td>700</td>
</tr>
<tr>
<td>191</td>
<td>694</td>
</tr>
<tr>
<td>188</td>
<td>702</td>
</tr>
<tr>
<td>215</td>
<td>694</td>
</tr>
<tr>
<td>215</td>
<td>710</td>
</tr>
<tr>
<td>249</td>
<td>710</td>
</tr>
<tr>
<td>291</td>
<td>710</td>
</tr>
<tr>
<td>296</td>
<td>708</td>
</tr>
</tbody>
</table>

Figure 10.22. Multivariate ARMA Prediction

The first 40 forecasts are one-step predictions. The last observation is the five-step forecast values of variables C and F. You can construct the confidence interval for these forecasts using the mean square error matrix, MSE. See the “Multivariate Time Series Analysis” section on page 259 for more details on impulse response functions and the mean square error matrix.
The TSROOT call computes the polynomial roots of the AR and MA equations. When the AR\(p\) process is written

\[ y_t = \sum_{i=1}^{p} c_i y_{t-i} + \epsilon_t \]

you can specify the following polynomial equation:

\[ z^p - \sum_{i=1}^{p} c_i z^{p-i} = 0 \]

When all \(p\) roots of the preceding equation are inside the unit circle, the AR\(p\) process is stationary. The MA\(q\) process is invertible if the following polynomial equation has all roots inside the unit circle:

\[ z^q + \sum_{i=1}^{q} \theta_i z^{q-i} = 0 \]

where \(\theta_i\) are the MA coefficients. For example, the best AR model is selected and estimated by the TSUNIMAR subroutine (see Figure 10.23). You can obtain the roots of the preceding equation by calling the TSROOT call. Since the TSROOT call can handle the complex AR or MA coefficients, note that you should add zero imaginary coefficients for the second column of the MATIN matrix for real coefficients.

```sas
proc iml;
          2.718 1.991 2.265 2.446 2.612 2.359 2.429 2.533 2.619 2.612
          2.179 1.653 1.832 2.328 2.737 3.014 3.328 3.402 3.981 2.557
          2.671 2.867 3.310 3.449 3.646 3.400 2.590 1.863 1.581 1.690
          1.771 2.274 2.576 3.111 3.605 3.543 2.769 2.021 2.185 2.588
          3.000 3.201 3.424 3.531 };
    call tsunimar(ar,v,nar,aic) data=y maxlag=5
        opt=((-1 1)) print=1;
    /*-- set up complex coefficient matrix --*/
    ar_cx = ar || j(nrow(ar),1,0);
    call tsroot(root) matin=ar_cx nar=nar
        nma=0 print=1;
```

In Figure 10.24, the roots and their lengths from the origin are shown. The roots are also stored in the matrix ROOT. All roots are within the unit circle, while the mod values of the fourth and fifth roots appear to be sizable (0.9194).
Figure 10.23. Minimum AIC AR Estimation

<table>
<thead>
<tr>
<th>I</th>
<th>Real</th>
<th>Imaginary</th>
<th>MOD(Z)</th>
<th>ATAN(I/R)</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.29755</td>
<td>0.55991</td>
<td>0.6341</td>
<td>2.0593</td>
<td>117.9869</td>
</tr>
<tr>
<td>2</td>
<td>-0.29755</td>
<td>-0.55991</td>
<td>0.6341</td>
<td>-2.0593</td>
<td>-117.9869</td>
</tr>
<tr>
<td>3</td>
<td>0.40529</td>
<td>0</td>
<td>0.4053</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0.74505</td>
<td>0.53866</td>
<td>0.9194</td>
<td>0.6260</td>
<td>35.8660</td>
</tr>
<tr>
<td>5</td>
<td>0.74505</td>
<td>-0.53866</td>
<td>0.9194</td>
<td>-0.6260</td>
<td>-35.8660</td>
</tr>
</tbody>
</table>

Figure 10.24. Roots of AR Characteristic Polynomial Equation

The TSROOT call can also recover the polynomial coefficients if the roots are given as an input. You should specify the QCOEF=1 option when you want to compute the polynomial coefficients instead of polynomial roots. You can compare the result with the preceding output of the TSUNIMAR call.

```
call tsroot(arCx) matin=root nar=nar qcoef=1
  nma=0 print=1;
```

<table>
<thead>
<tr>
<th>I</th>
<th>AR(real)</th>
<th>AR(imag)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.30031</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>-0.72328</td>
<td>1.11022E-16</td>
</tr>
<tr>
<td>3</td>
<td>0.24219</td>
<td>1.66533E-16</td>
</tr>
<tr>
<td>4</td>
<td>-0.37876</td>
<td>-2.7756E-17</td>
</tr>
<tr>
<td>5</td>
<td>0.13773</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 10.25. Polynomial Coefficients
Syntax

For details on the syntax of TIMSAC subroutines, see the individual entries in the reference library in Chapter 17, “Language Reference.”

TIMSAC routines are controlled by the following statements:

CALL TSBAYSEA(trend, season, series, adjust, abic, data
   <,order, sorder, rigid, npred, opt, cntl, print>);

CALL TSDECOMP(comp, est, aic, data, <,xdata, order, sorder,
   nar, npred, init, opt, icmp, print>);

CALL TSMLOCAR(arcoef, ev, nar, aic, start, finish, data
   <,maxlag, opt, missing, print>);

CALL TSMLOMAR(arcoef, ev, nar, aic, start, finish, data
   <,maxlag, opt, missing, print>);

CALL TSMULMAR(arcoef, ev, nar, aic, data
   <,maxlag, opt, missing, print>);

CALL TSPEARS(arcoef, ev, nar, aic, data
   <,maxlag, opt, missing, print>);

CALL TSPRED(forecast, impulse, mse, data, coef, nar, nma
   <,ev, npred, start, constant>);

CALL TSROOT(matout, matin, nar, nma, <,qcoef, print>);

CALL TSTVCAR(arcoef, variance, est, aic, data
   <,nar, init, opt, outlier, print>);

CALL TSUNIMAR(arcoef, ev, nar, aic, data
   <,maxlag, opt, missing, print>);
Details

This section presents an introductory description of the important topics that are directly related to TIMSAC IML subroutines. The computational details, including algorithms, are described in the “Computational Details” section on page 264. A detailed explanation of each subroutine is not given; instead, basic ideas and common methodologies for all subroutines are described first and are followed by more technical details. Finally, missing values are discussed in the section “Missing Values” on page 269.

Minimum AIC Procedure

The AIC statistic is widely used to select the best model among alternative parametric models. The minimum AIC model selection procedure can be interpreted as a maximization of the expected entropy (Akaike 1981). The entropy of a true probability density function (PDF) $\varphi$ with respect to the fitted PDF $f$ is written as

$$B(\varphi, f) = -I(\varphi, f)$$

where $I(\varphi, f)$ is a Kullback-Leibler information measure, which is defined as

$$I(\varphi, f) = \int \left[ \log \frac{\varphi(z)}{f(z)} \right] \varphi(z) dz$$

where the random variable $Z$ is assumed to be continuous. Therefore,

$$B(\varphi, f) = E_Z \log f(Z) - E_Z \log \varphi(Z)$$

where $B(\varphi, f) \leq 0$ and $E_Z$ denotes the expectation concerning the random variable $Z$. $B(\varphi, f) = 0$ if and only if $\varphi = f$ (a.s.). The larger the quantity $E_Z \log f(Z)$, the closer the function $f$ is to the true PDF $\varphi$. Given the data $y = (y_1, \ldots, y_T)'$ that has the same distribution as the random variable $Z$, let the likelihood function of the parameter vector $\theta$ be $\prod_{t=1}^{T} f(y_t|\theta)$. Then the average of the log likelihood function $\frac{1}{T} \sum_{t=1}^{T} \log f(y_t|\theta)$ is an estimate of the expected value of $\log f(Z)$. Akaike (1981) derived the alternative estimate of $E_Z \log f(Z)$ by using the Bayesian predictive likelihood. The AIC is the bias-corrected estimate of $-2E_Z \log f(Z|\theta)$, where $\theta$ is the maximum likelihood estimate.

$$\text{AIC} = -2(\text{maximum log likelihood}) + 2(\text{number of free parameters})$$

Let $\theta = (\theta_1, \ldots, \theta_K)'$ be a $K \times 1$ parameter vector that is contained in the parameter space $\Theta_K$. Given the data $y$, the log likelihood function is

$$\ell(\theta) = \sum_{t=1}^{T} \log f(y_t|\theta)$$

Suppose the probability density function $f(y|\theta)$ has the true PDF $\varphi(y) = f(y|\theta^0)$, where the true parameter vector $\theta^0$ is contained in $\Theta_K$. Let $\hat{\theta}_K$ be a maximum
likelihood estimate. The maximum of the log likelihood function is denoted as 
\( \ell(\theta_K) = \max_{\theta \in \Theta_K} \ell(\theta) \). The expected log likelihood function is defined by

\[ \ell^*(\theta) = T E_Z \log f(Z|\theta) \]

The Taylor series expansion of the expected log likelihood function around the true parameter \( \theta^0 \) gives the following asymptotic relationship:

\[ \ell^*(\theta) \overset{A}{=} \ell^*(\theta^0) + T(\theta - \theta^0)' E_Z \frac{\partial \log f(Z|\theta^0)}{\partial \theta} - \frac{T}{2} (\theta - \theta^0)' I(\theta^0)(\theta - \theta^0) \]

where \( I(\theta^0) \) is the information matrix and \( A \) stands for asymptotic equality. Note that \( \frac{\partial \log f(z|\theta^0)}{\partial \theta} = 0 \) since \( \log f(z|\theta) \) is maximized at \( \theta^0 \). By substituting \( \hat{\theta}_K \), the expected log likelihood function can be written as

\[ \ell^*(\hat{\theta}_K) \overset{A}{=} \ell^*(\theta^0) - \frac{T}{2} (\hat{\theta}_K - \theta^0)' I(\theta^0)(\hat{\theta}_K - \theta^0) \]

The maximum likelihood estimator is asymptotically normally distributed under the regularity conditions

\[ \sqrt{T} I(\theta^0)^{1/2}(\hat{\theta}_K - \theta^0) \overset{d}{\rightarrow} N(0,I_K) \]

Therefore,

\[ T(\hat{\theta}_K - \theta^0)' I(\theta^0)(\hat{\theta}_K - \theta^0) \overset{a}{\sim} \chi^2_K \]

The mean expected log likelihood function, \( \ell^*(K) = E_Y \ell^*(\hat{\theta}_K) \), becomes

\[ \ell^*(K) \overset{A}{=} \ell^*(\theta^0) - \frac{K}{2} \]

When the Taylor series expansion of the log likelihood function around \( \hat{\theta}_K \) is used, the log likelihood function \( \ell(\theta) \) is written

\[ \ell(\theta) \overset{A}{=} \ell(\hat{\theta}_K) + (\theta - \hat{\theta}_K)' \frac{\partial \ell(\hat{\theta}_K)}{\partial \theta} \bigg|_{\hat{\theta}_K} + \frac{1}{2} (\theta - \hat{\theta}_K)' \left( \frac{\partial^2 \ell(\hat{\theta}_K)}{\partial \theta \partial \theta'} \bigg|_{\hat{\theta}_K} \right) (\theta - \hat{\theta}_K) \]

Since \( \ell(\hat{\theta}_K) \) is the maximum log likelihood function, \( \frac{\partial \ell(\hat{\theta}_K)}{\partial \theta} \bigg|_{\hat{\theta}_K} = 0 \). Note that

\[ \text{plim} \left[ -\frac{1}{T} \frac{\partial^2 \ell(\theta)}{\partial \theta \partial \theta'} \bigg|_{\hat{\theta}_K} \right] = I(\theta^0) \]

if the maximum likelihood estimator \( \hat{\theta}_K \) is a consistent estimator of \( \theta \). Replacing \( \theta \) with the true parameter \( \theta^0 \) and taking expectations with respect to the random variable \( Y \),

\[ E_Y \ell(\theta^0) \overset{A}{=} E_Y \ell(\hat{\theta}_K) - \frac{K}{2} \]
Consider the following relationship:

\[
\ell^*(\theta^0) = T \mathbb{E}_Z \log f(Z|\theta^0) = \mathbb{E}_Y \sum_{t=1}^{T} \log f(Y_t|\theta^0) = \mathbb{E}_Y \ell(\theta^0)
\]

From the previous derivation,

\[
\ell^*(K) \triangleq \ell^*(\theta^0) - \frac{K}{2}
\]

Therefore,

\[
\ell^*(K) \triangleq \mathbb{E}_Y \ell(\hat{\theta}_K) - K
\]

The natural estimator for \(\mathbb{E}_Y \ell(\hat{\theta}_K)\) is \(\hat{\ell}(\hat{\theta}_K)\). Using this estimator, you can write the mean expected log likelihood function as

\[
\ell^*(K) \triangleq \hat{\ell}(\hat{\theta}_K) - K
\]

Consequently, the AIC is defined as an asymptotically unbiased estimator of

\[ -2(\text{mean expected log likelihood}) \]

\[
AIC(K) = -2\hat{\ell}(\hat{\theta}_K) + 2K
\]

In practice, the previous asymptotic result is expected to be valid in finite samples if the number of free parameters does not exceed \(2\sqrt{T}\) and the upper bound of the number of free parameters is \(\frac{T}{2}\). It is worth noting that the amount of AIC is not meaningful in itself, since this value is not the Kullback-Leibler information measure. The difference of AIC values can be used to select the model. The difference of the two AIC values is considered insignificant if it is far less than 1. It is possible to find a better model when the minimum AIC model contains many free parameters.

**Smoothness Priors Modeling**

Consider the time series \(y_t\):

\[
y_t = f(t) + \epsilon_t
\]

where \(f(t)\) is an unknown smooth function and \(\epsilon_t\) is an iid random variable with zero mean and positive variance \(\sigma^2\). Whittaker (1923) provides the solution, which balances a tradeoff between closeness to the data and the \(kth\) order difference equation. For a fixed value of \(\lambda\) and \(k\), the solution \(\hat{f}\) satisfies

\[
\min_{f} \sum_{t=1}^{T} \left\{ [y_t - f(t)]^2 + \lambda^2 [\nabla^k f(t)]^2 \right\}
\]
where $\nabla^k$ denotes the $k$th order difference operator. The value of $\lambda$ can be viewed as the smoothness tradeoff measure. Akaike (1980a) proposed the Bayesian posterior PDF to solve this problem.

$$\ell(f) = \exp \left\{ -\frac{1}{2\sigma^2} \sum_{t=1}^{T} [y_t - f(t)]^2 \right\} \exp \left\{ -\frac{\lambda^2}{2\sigma^2} \sum_{t=1}^{T} [\nabla^k f(t)]^2 \right\}$$

Therefore, the solution can be obtained when the function $\ell(f)$ is maximized.

Assume that time series is decomposed as follows:

$$y_t = T_t + S_t + \epsilon_t$$

where $T_t$ denotes the trend component and $S_t$ is the seasonal component. The trend component follows the $k$th order stochastically perturbed difference equation.

$$\nabla^k T_t = w_{1t}, \quad w_{1t} \sim N(0, \tau_1^2)$$

For example, the polynomial trend component for $k = 2$ is written as

$$T_t = 2T_{t-1} - T_{t-2} + w_{1t}$$

To accommodate regular seasonal effects, the stochastic seasonal relationship is used.

$$\sum_{i=0}^{L-1} S_{t-i} = w_{2t}, \quad w_{2t} \sim N(0, \tau_2^2)$$

where $L$ is the number of seasons within a period. In the context of Whittaker and Akaike, the smoothness priors problem can be solved by the maximization of

$$\ell(f) = \exp \left\{ -\frac{1}{2\sigma^2} \sum_{t=1}^{T} (y_t - T_t - S_t)^2 \right\} \exp \left\{ -\frac{\tau_1^2}{2\sigma^2} \sum_{t=1}^{T} (\nabla^1 T_t)^2 \right\} \times \exp \left\{ -\frac{\tau_2^2}{2\sigma^2} \sum_{t=1}^{T} \left( \sum_{i=0}^{L-1} S_{t-i} \right)^2 \right\}$$

The values of hyperparameters, $\tau_1^2$ and $\tau_2^2$, refer to a measure of uncertainty of prior information. For example, the large value of $\tau_1^2$ implies a relatively smooth trend component. The ratio $\frac{\tau_1^2}{\tau_2^2}$ ($i = 1, 2$) can be considered as a signal-to-noise ratio.

Kitagawa and Gersch (1984) use the Kalman filter recursive computation for the likelihood of the tradeoff parameters. The hyperparameters are estimated by combining the grid search and optimization method. The state space model and Kalman filter
Bayesian Seasonal Adjustment

Seasonal phenomena are frequently observed in many economic and business time series. For example, consumption expenditure might have strong seasonal variations because of Christmas spending. The seasonal phenomena are repeatedly observed after a regular period of time. The number of seasons within a period is defined as the smallest time span for this repetitive observation. Monthly consumption expenditure shows a strong increase during the Christmas season, with 12 seasons per period.

There are three major approaches to seasonal time series: the regression model, the moving average model, and the seasonal ARIMA model.

Regression Model

Let the trend component be \( T_t = \sum_{i=1}^{m_\alpha} \alpha_i U_{it} \) and the seasonal component be \( S_t = \sum_{j=1}^{m_\beta} \beta_j V_{jt} \). Then the additive time series can be written as the regression model

\[
y_t = \sum_{i=1}^{m_\alpha} \alpha_i U_{it} + \sum_{j=1}^{m_\beta} \beta_j V_{jt} + \epsilon_t
\]

In practice, the trend component can be written as the \( m_\alpha \)th order polynomial, such as

\[
T_t = \sum_{i=0}^{m_\alpha} \alpha_i t^i
\]

The seasonal component can be approximated by the seasonal dummies \((D_{jt})\)

\[
S_t = \sum_{j=1}^{L-1} \beta_j D_{jt}
\]

where \( L \) is the number of seasons within a period. The least squares method is applied to estimate parameters \( \alpha_i \) and \( \beta_j \).

The seasonally adjusted series is obtained by subtracting the estimated seasonal component from the original series. Usually, the error term \( \epsilon_t \) is assumed to be white noise, while sometimes the autocorrelation of the regression residuals needs to be allowed. However, the regression method is not robust to the regression function type, especially at the beginning and end of the series.

Moving Average Model

If you assume that the annual sum of a seasonal time series has small seasonal fluctuations, the nonseasonal component \( \hat{N}_t = T_t + \epsilon_t \) can be estimated by using the moving average method.

\[
\hat{N}_t = \sum_{i=-m}^{m} \lambda_i y_{t-i}
\]
where \( m \) is the positive integer and \( \lambda_i \) is the symmetric constant such that \( \lambda_i = \lambda_{-i} \) and \( \sum_{i=-m}^{m} \lambda_i = 1 \).

When the data are not available, either an asymmetric moving average is used, or the forecast data is augmented to use the symmetric weight. The X-11 procedure is a complex modification of this moving average method.

### Seasonal ARIMA Model

The regression and moving average approaches assume that the seasonal component is deterministic and independent of other nonseasonal components. The time series approach is used to handle the stochastic trend and seasonal components.

The general ARIMA model can be written

\[
\prod_{j=1}^{m} \phi_j(B) \prod_{i=1}^{k} (1 - B^{s_i})^{d_i} \tilde{y}_t = \theta_0 + \prod_{i=1}^{q} \theta_i(B) \epsilon_t
\]

where \( B \) is the backshift operator and

\[
\phi_j(B) = 1 - \phi_1 B - \cdots - \phi_j B^{p_j} \quad \theta_i(B) = 1 - \theta_1 B - \cdots - \theta_i B^{q_i}
\]

and \( \tilde{y}_t = y_t - E(Y_t) \) if \( d_i = 0 \); otherwise, \( \tilde{y}_t = y_t \). The power of \( B \), \( s_i \), can be considered as a seasonal factor. Specifically, the Box-Jenkins multiplicative seasonal ARIMA model is written as

\[
\phi_p(B) \Phi_P(B^s) (1 - B)^d (1 - B^s)^D \tilde{y}_t = \theta_q(B) \Theta_Q(B^s) \epsilon_t
\]

ARIMA modeling is appropriate for particular time series and requires burdensome computation.

The TSBAYSEA subroutine combines the simple characteristics of the regression approach and time series modeling. The TSBAYSEA and X-11 procedures use the model-based seasonal adjustment. The symmetric weights of the standard X-11 option can be approximated by using the integrated MA form

\[
(1 - B)(1 - B^{12}) y_t = \theta(B) \epsilon_t
\]

With a fixed value \( \phi \), the TSBAYSEA subroutine is approximated as

\[
(1 - \phi B)(1 - B)(1 - B^{12}) y_t = \theta(B) \epsilon_t
\]

The subroutine is flexible enough to handle trading day or leap year effects, the shift of the base observation, and missing values. The TSBAYSEA-type modeling approach has some advantages: it clearly defines the statistical model of the time series;
modification of the basic model can be an efficient method of choosing a particular procedure for the seasonal adjustment of a given time series; and the use of the concept of the likelihood provides a minimum AIC model selection approach.

**Nonstationary Time Series**

The subroutines TSMLOCAR, TSMLOMAR, and TSTVCAR are used to analyze nonstationary time series models. The AIC statistic is extensively used to analyze the locally stationary model.

**Locally Stationary AR Model**

When the time series is nonstationary, the TSMLOCAR (univariate) and TSMLOMAR (multivariate) subroutines can be employed. The whole span of the series is divided into locally stationary blocks of data, and then the TSMLOCAR and TSMLOMAR subroutines estimate a stationary AR model by using the least squares method on this stationary block. The homogeneity of two different blocks of data is tested using the AIC.

Given a set of data \( \{y_1, \ldots, y_T\} \), the data can be divided into \( k \) blocks of sizes \( t_1, \ldots, t_k \), where \( t_1 + \cdots + t_k = T \), and \( k \) and \( t_i \) are unknown. The locally stationary model is fitted to the data

\[
y_t = \alpha_0^i + \sum_{j=1}^{p_i} \alpha_j^i y_{t-j} + \epsilon_t^i
\]

where

\[
T_{i-1} = \sum_{j=1}^{i-1} t_j < T_i = \sum_{j=1}^{i} t_j, \quad \text{for } i = 1, \ldots, k
\]

where \( \epsilon_t^i \) is a Gaussian white noise with \( \text{E} \epsilon_t^i = 0 \) and \( \text{E}(\epsilon_t^i)^2 = \sigma_t^2 \). Therefore, the log likelihood function of the locally stationary series is

\[
\ell = -\frac{1}{2} \sum_{i=1}^{k} \left[ t_i \log(2\pi \sigma_t^2) + \frac{1}{\sigma_t^2} \sum_{t=T_{i-1}+1}^{T_i} \left( y_t - \alpha_0^i - \sum_{j=1}^{p_i} \alpha_j^i y_{t-j} \right)^2 \right]
\]

Given \( \alpha_j^i, j = 0, \ldots, p_i \), the maximum of the log likelihood function is attained at

\[
\hat{\sigma}_t^2 = \frac{1}{t_i} \sum_{t=T_{i-1}+1}^{T_i} \left( y_t - \alpha_0^i - \sum_{j=1}^{p_i} \alpha_j^i y_{t-j} \right)^2
\]

The concentrated log likelihood function is given by

\[
\ell^* = -\frac{T}{2} \left[ 1 + \log(2\pi) \right] - \frac{1}{2} \sum_{i=1}^{k} t_i \log(\hat{\sigma}_t^2)
\]
Therefore, the maximum likelihood estimates, $\hat{\alpha}_i^j$ and $\hat{\sigma}_i^2$, are obtained by minimizing the following local SSE:

$$SSE = \sum_{t=T_{i-1}+1}^{T_i} \left( \frac{y_t - \hat{\alpha}_0^i - \sum_{j=1}^{p_i} \hat{\alpha}_j^i y_{t-j}}{\hat{\sigma}_i^2} \right)^2$$

The least squares estimation of the stationary model is explained in the section, "Least Squares and Householder Transformation".

The AIC for the locally stationary model over the pooled data is written as

$$\sum_{i=1}^{k} t_i \log(\hat{\sigma}_i^2) + 2 \sum_{i=1}^{k} (p_i + intercept + 1)$$

where intercept = 1 if the intercept term ($\alpha_0^i$) is estimated; otherwise, intercept = 0. The number of stationary blocks ($k$), the size of each block ($t_i$), and the order of the locally stationary model is determined by the AIC. Consider the autoregressive model fitted over the block of data, $\{y_1, \ldots, y_T\}$, and let this model $M_1$ be an AR($p_1$) process. When additional data, $\{y_{T+1}, \ldots, y_{T+T_1}\}$, are available, a new model $M_2$, an AR($p_2$) process, is fitted over this new data set, assuming that these data are independent of the previous data. Then AICs for models $M_1$ and $M_2$ are defined as

$$AIC_1 = T_1 \log(\hat{\sigma}_1^2) + 2(p_1 + intercept + 1)$$
$$AIC_2 = T_1 \log(\hat{\sigma}_2^2) + 2(p_2 + intercept + 1)$$

The joint model AIC for $M_1$ and $M_2$ is obtained by summation

$$AIC_J = AIC_1 + AIC_2$$

When the two data sets are pooled and estimated over the pooled data set, $\{y_1, \ldots, y_{T+T_1}\}$, the AIC of the pooled model is

$$AIC_A = (T + T_1) \log(\hat{\sigma}_A^2) + 2(p_A + intercept + 1)$$

where $\sigma_A^2$ is the pooled error variance and $p_A$ is the order chosen to fit the pooled data set.

**Decision**

- If $AIC_J < AIC_A$, switch to the new model, since there is a change in the structure of the time series.
- If $AIC_J \geq AIC_A$, pool the two data sets, since two data sets are considered to be homogeneous.
If new observations are available, repeat the preceding steps to determine the homogeneity of the data. The basic idea of locally stationary AR modeling is that, if the structure of the time series is not changed, you should use the additional information to improve the model fitting, but you need to follow the new structure of the time series if there is any change.

**Time-Varying AR Coefficient Model**

Another approach to nonstationary time series, especially those that are nonstationary in the covariance, is time-varying AR coefficient modeling. When the time series is nonstationary in the covariance, the problem in modeling this series is related to an efficient parameterization. It is possible for a Bayesian approach to estimate the model with a large number of implicit parameters of the complex structure by using a relatively small number of hyperparameters.

The TSTVCAR subroutine uses smoothness priors by imposing stochastically perturbed difference equation constraints on each AR coefficient and frequency response function. The variance of each AR coefficient distribution constitutes a hyperparameter included in the state space model. The likelihood of these hyperparameters is computed by the Kalman filter recursive algorithm.

The time-varying AR coefficient model is written

\[
y_t = \sum_{i=1}^{m} \alpha_{it} y_{t-i} + \epsilon_t
\]

where time-varying coefficients \( \alpha_{it} \) are assumed to change gradually with time. The following simple stochastic difference equation constraint is imposed on each coefficient:

\[
\nabla^k \alpha_{it} = w_{it}, \quad w_{it} \sim N(0, \tau^2), \quad i = 1, \ldots, m
\]

The frequency response function of the AR process is written

\[
A(f) = 1 - \sum_{j=1}^{m} \alpha_{jt} \exp(-2\pi j i f)
\]

The smoothness of this function can be measured by the \( kth \) derivative smoothness constraint,

\[
R_k = \int_{-1/2}^{1/2} \left| \frac{d^k A(f)}{df^k} \right|^2 df = (2\pi)^{2k} \sum_{j=1}^{m} j^{2k} \alpha_{jt}^2
\]

Then the TSTVCAR call imposes zero and second derivative smoothness constraints. The time-varying AR coefficients are the solution of the following constrained least squares:

\[
\sum_{t=1}^{T} \left( y_t - \sum_{i=1}^{m} \alpha_{it} y_{t-i} \right)^2 + \tau^2 \sum_{t=1}^{T} \sum_{i=1}^{m} \left( \nabla^k \alpha_{it} \right)^2 + \lambda^2 \sum_{t=1}^{T} \sum_{i=1}^{m} \alpha_{it}^2 + \nu^2 \sum_{t=1}^{T} \sum_{i=1}^{m} \alpha_{it}^2
\]
where $\tau^2$, $\lambda^2$, and $\nu^2$ are hyperparameters of the prior distribution.

Using a state space representation, the model is

\[
\begin{align*}
\mathbf{x}_t &= \mathbf{F}\mathbf{x}_{t-1} + \mathbf{G}\mathbf{w}_t \\
y_t &= \mathbf{H}\mathbf{x}_t + \epsilon_t
\end{align*}
\]

where

\[
\begin{align*}
\mathbf{x}_t &= (\alpha_{1,t}, \ldots, \alpha_{mt}, \ldots, \alpha_{1,t-k+1}, \ldots, \alpha_{m,t-k+1})' \\
\mathbf{H}_t &= (y_{t-1}, \ldots, y_{t-m}, \ldots, 0, \ldots, 0) \\
\mathbf{w}_t &= (w_{1t}, \ldots, w_{mt})' \\
k &= 1 : \mathbf{F} = \mathbf{I}_m \quad \mathbf{G} = \mathbf{I}_m \\
k &= 2 : \mathbf{F} = \begin{bmatrix} 2\mathbf{I}_m & -\mathbf{I}_m \\ \mathbf{I}_m & 0 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \mathbf{I}_m \\ 0 \end{bmatrix} \\
k &= 3 : \mathbf{F} = \begin{bmatrix} 3\mathbf{I}_m & -3\mathbf{I}_m & \mathbf{I}_m \\ \mathbf{I}_m & 0 & 0 \\ 0 & \mathbf{I}_m & 0 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \mathbf{I}_m \\ 0 \\ 0 \end{bmatrix}
\end{align*}
\]

\[
\begin{bmatrix} \mathbf{w}_t \\ \epsilon_t \end{bmatrix} \sim N\left(0, \begin{bmatrix} \tau^2\mathbf{I} & 0 \\ 0 & \sigma^2 \end{bmatrix} \right)
\]

The computation of the likelihood function is straightforward. See the section, "State Space and Kalman Filter Method" for the computation method.

**Multivariate Time Series Analysis**

The subroutines TSMULMAR, TSMLOMAR, and TSPRED analyze multivariate time series. The periodic AR model, TSPEARS, can also be estimated by using a vector AR procedure, since the periodic AR series can be represented as the covariance-stationary vector autoregressive model.

The stationary vector AR model is estimated and the order of the model (or each variable) is automatically determined by the minimum AIC procedure. The stationary vector AR model is written

\[
\begin{align*}
\mathbf{y}_t &= \mathbf{A}_0 + \mathbf{A}_1\mathbf{y}_{t-1} + \cdots + \mathbf{A}_p\mathbf{y}_{t-p} + \epsilon_t \\
\epsilon_t &\sim N(0, \Sigma)
\end{align*}
\]

Using the LDL' factorization method, the error covariance is decomposed as

\[
\Sigma = \mathbf{L}\mathbf{D}\mathbf{L}'
\]

where $\mathbf{L}$ is a unit lower triangular matrix and $\mathbf{D}$ is a diagonal matrix. Then the instantaneous response model is defined as

\[
\mathbf{C}\mathbf{y}_t = \mathbf{A}_0^* + \mathbf{A}_1^*\mathbf{y}_{t-1} + \cdots + \mathbf{A}_p^*\mathbf{y}_{t-p} + \epsilon_t^*
\]

where $\mathbf{C} = \mathbf{L}^{-1}$, $\mathbf{A}_i^* = \mathbf{L}^{-1}\mathbf{A}_i$ for $i = 0, 1, \ldots, p$, and $\epsilon_t^* = \mathbf{L}^{-1}\epsilon_t$. Each equation of the instantaneous response model can be estimated independently, since its
error covariance matrix has a diagonal covariance matrix $D$. Maximum likelihood estimates are obtained through the least squares method when the disturbances are normally distributed and the presample values are fixed.

The TSMULMAR call estimates the instantaneous response model. The VAR coefficients are computed using the relationship between the VAR and instantaneous models.

The general VARMA model can be transformed as an infinite order MA process under certain conditions.

$$y_t = \mu + \epsilon_t + \sum_{m=1}^{\infty} \Psi_m \epsilon_{t-m}$$

In the context of the VAR($p$) model, the coefficient $\Psi_m$ can be interpreted as the $m$-lagged response of a unit increase in the disturbances at time $t$.

$$\Psi_m = \frac{\partial y_{t+m}}{\partial \epsilon_t^j}$$

The lagged response on the one-unit increase in the orthogonalized disturbances $\epsilon_t^j$ is denoted

$$\frac{\partial y_{t+m}}{\partial \epsilon_t^j} = \frac{\partial \mathbb{E}(y_{t+m}|y_{jt}, y_{j-1,t}, \ldots, X_t)}{\partial y_{jt}} = \Psi_m L_j$$

where $L_j$ is the $j$th column of the unit triangular matrix $L$ and $X_t = [y_{t-1}, \ldots, y_{t-p}]$. When you estimate the VAR model using the TSMULMAR call, it is easy to compute this impulse response function.

The MSE of the $m$-step prediction is computed as

$$\mathbb{E}(y_{t+m} - y_{t+m}|t)(y_{t+m} - y_{t+m}|t)' = \Sigma + \Psi_1 \Sigma \Psi_1' + \cdots + \Psi_{m-1} \Sigma \Psi_{m-1}'$$

Note that $\epsilon_t = L \epsilon_t^\alpha$. Then the covariance matrix of $\epsilon_t$ is decomposed

$$\Sigma = \sum_{i=1}^{n} L_i L_i' d_{ii}$$

where $d_{ii}$ is the $i$th diagonal element of the matrix $D$ and $n$ is the number of variables. The MSE matrix can be written

$$\sum_{i=1}^{n} d_{ii} [L_i L_i' + \Psi_1 L_i L_i' \Psi_1' + \cdots + \Psi_{m-1} L_i L_i' \Psi_{m-1}']$$

Therefore, the contribution of the $i$th orthogonalized innovation to the MSE is

$$V_i = d_{ii} [L_i L_i' + \Psi_1 L_i L_i' \Psi_1' + \cdots + \Psi_{m-1} L_i L_i' \Psi_{m-1}']$$
The $ith$ forecast error variance decomposition is obtained from diagonal elements of the matrix $V_i$.

The nonstationary multivariate series can be analyzed by the TSMLOMAR subroutine. The estimation and model identification procedure is analogous to the univariate nonstationary procedure, which is explained in the “Nonstationary Time Series” section on page 256.

A time series $y_t$ is periodically correlated with period $d$ if $E_y t = E_y t+d$ and $E_y s y_t = E_y s+d y_t+d$. Let $y_t$ be autoregressive of period $d$ with AR orders $(p_1, \ldots, p_d)$, that is,

\[
y_t = \sum_{j=1}^{p_t} \alpha_{jt} y_{t-j} + \epsilon_t
\]

where $\epsilon_t$ is uncorrelated with mean zero and $E\epsilon_t^2 = \sigma_t^2$, $p_t = p_{t+d}$, $\sigma_t^2 = \sigma_{t+d}^2$, and $\alpha_{jt} = \alpha_{j+t+d}(j = 1, \ldots, p_t)$. Define the new variable such that $x_{jt} = y_{j+d(t-1)}$. The vector series, $x_t = (x_{1t}, \ldots, x_{dt})'$, is autoregressive of order $p$, where $p = \max_j \text{int}((p_j - j)/d) + 1$. The TSPEARS subroutine estimates the periodic autoregressive model using minimum AIC vector AR modeling.

The TSPRED subroutine computes the one-step or multistep forecast of the multivariate ARMA model if the ARMA parameter estimates are provided. In addition, the subroutine TSPRED produces the (intermediate and permanent) impulse response function and performs forecast error variance decomposition for the vector AR model.

**Spectral Analysis**

The autocovariance function of the random variable $Y_t$ is defined as

\[
C_{YY}(k) = E(Y_{t+k}Y_t)
\]

where $EY_t = 0$. When the real valued process $Y_t$ is stationary and its autocovariance is absolutely summable, the population spectral density function is obtained using the Fourier transform of the autocovariance function

\[
f(g) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} C_{YY}(k) \exp(-igk) \quad -\pi \leq g \leq \pi
\]

where $i = \sqrt{-1}$ and $C_{YY}(k)$ is the autocovariance function such that $\sum_{k=-\infty}^{\infty} |C_{YY}(k)| < \infty$.

Consider the autocovariance generating function

\[
\gamma(z) = \sum_{k=-\infty}^{\infty} C_{YY}(k)z^k
\]

where $C_{YY}(k) = C_{YY}(-k)$ and $z$ is a complex scalar. The spectral density function can be represented as

\[
f(g) = \frac{1}{2\pi} \gamma(\exp(-ig))
\]
The stationary ARMA($p, q$) process is denoted:

$$\phi(B)y_t = \theta(B)\epsilon_t \quad \epsilon_t \sim (0, \sigma^2)$$

where $\phi(B)$ and $\theta(B)$ do not have common roots. Note that the autocovariance generating function of the linear process $y_t = \psi(B)\epsilon_t$ is given by

$$\gamma(B) = \sigma^2\psi(B)\psi(B^{-1})$$

For the ARMA($p, q$) process, $\psi(B) = \frac{\theta(B)}{\phi(B)}$. Therefore, the spectral density function of the stationary ARMA($p, q$) process becomes

$$f(g) = \frac{\sigma^2}{2\pi} \left| \frac{\theta(\exp(-ig))\theta(\exp(ig))}{\phi(\exp(-ig))\phi(\exp(ig))} \right|^2$$

The spectral density function of a white noise is a constant.

$$f(g) = \frac{\sigma^2}{2\pi}$$

The spectral density function of the AR(1) process ($\phi(B) = 1 - \phi_1B$) is given by

$$f(g) = \frac{\sigma^2}{2\pi(1 - \phi_1\cos(g) + \phi_1^2)}$$

The spectrum of the AR(1) process has its minimum at $g = 0$ and its maximum at $g = \pm \pi$ if $\phi_1 < 0$, while the spectral density function attains its maximum at $g = 0$ and its minimum at $g = \pm \pi$, if $\phi_1 > 0$. When the series is positively autocorrelated, its spectral density function is dominated by low frequencies. It is interesting to observe that the spectrum approaches $\frac{\sigma^2}{4\pi} \frac{1}{1 - \cos(g)}$ as $\phi_1 \to 1$. This relationship shows that the series is difference-stationary if its spectral density function has a remarkable peak near 0.

The spectrum of AR(2) process ($\phi(B) = 1 - \phi_1B - \phi_2B^2$) equals

$$f(g) = \frac{\sigma^2}{2\pi} \left\{ \frac{1}{-4\phi_2 \left[ \cos(g) + \frac{\phi_1(1-\phi_1)}{4\phi_2} \right]^2 + \frac{(1+\phi_1)^2(4\phi_2+\phi_1^2)}{4\phi_2}} \right\}$$

Refer to Anderson (1971) for details of the characteristics of this spectral density function of the AR(2) process.

In practice, the population spectral density function cannot be computed. There are many ways of computing the sample spectral density function. The TSBAYSEA and TSMLOCAR calls compute the power spectrum using AR coefficients and the white noise variance.
The power spectral density function of $Y_t$ is derived using the Fourier transformation of $C_{YY}(k)$.

$$f_{YY}(g) = \sum_{k=-\infty}^{\infty} \exp(-2\pi igk)C_{YY}(k), \quad \frac{1}{2} \leq g \leq \frac{1}{2}$$

where $i = \sqrt{-1}$ and $g$ denotes frequency. The autocovariance function can also be written as

$$C_{YY}(k) = \int_{-1/2}^{1/2} \exp(2\pi igk)f_{YY}(g)dg$$

Consider the following stationary AR($p$) process:

$$y_t - \sum_{i=1}^{p} \phi_i y_{t-i} = \epsilon_t$$

where $\epsilon_t$ is a white noise with mean zero and constant variance $\sigma^2$. The autocovariance function of white noise $\epsilon_t$ equals

$$C_{\epsilon\epsilon}(k) = \delta_{k0}\sigma^2$$

where $\delta_{k0} = 1$ if $k = 0$; otherwise, $\delta_{k0} = 0$. Therefore, the power spectral density of the white noise is $f_{\epsilon\epsilon}(g) = \sigma^2$, $-\frac{1}{2} \leq g \leq \frac{1}{2}$. Note that, with $\phi_0 = -1$,

$$C_{\epsilon\epsilon}(k) = \sum_{m=0}^{p} \sum_{n=0}^{p} \phi_m \phi_n C_{YY}(k - m + n)$$

Using the following autocovariance function of $Y_t$,

$$C_{YY}(k) = \int_{-1/2}^{1/2} \exp(2\pi igk)f_{YY}(g)dg$$

the autocovariance function of the white noise is denoted as

$$C_{\epsilon\epsilon}(k) = \sum_{m=0}^{p} \sum_{n=0}^{p} \phi_m \phi_n \int_{-1/2}^{1/2} \exp(2\pi ig(k - m + n))f_{YY}(g)dg$$

$$= \int_{-1/2}^{1/2} \exp(2\pi igk) \left| 1 - \sum_{m=1}^{p} \phi_m \exp(-2\pi igm) \right|^2 f_{YY}(g)dg$$

On the other hand, another formula of the $C_{\epsilon\epsilon}(k)$ gives

$$C_{\epsilon\epsilon}(k) = \int_{-1/2}^{1/2} \exp(2\pi igk)f_{\epsilon\epsilon}(g)dg$$
Therefore,
\[
\hat{f}(\omega) = \left| 1 - \sum_{m=1}^{p} \phi_m \exp(-2\pi i \omega m) \right|^2 \hat{f}_{YY}(\omega)
\]

Since \( \hat{f}(\omega) = \sigma^2 \), the rational spectrum of \( Y_t \) is
\[
\hat{f}_{YY}(\omega) = \frac{\sigma^2}{\left| 1 - \sum_{m=1}^{p} \phi_m \exp(-2\pi i \omega m) \right|^2}
\]

To compute the power spectrum, estimated values of white noise variance \( \hat{\sigma}^2 \) and AR coefficients \( \hat{\phi}_m \) are used. The order of the AR process can be determined by using the minimum AIC procedure.

**Computational Details**

**Least Squares and Householder Transformation**

Consider the univariate AR(\( p \)) process
\[
y_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i} + \epsilon_t
\]

Define the design matrix \( \mathbf{X} \).
\[
\mathbf{X} = \begin{bmatrix}
1 & y_p & \cdots & y_1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & y_{T-1} & \cdots & y_{T-p}
\end{bmatrix}
\]

Let \( \mathbf{y} = (y_{p+1}, \ldots, y_n)' \). The least squares estimate, \( \hat{\mathbf{a}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \), is the approximation to the maximum likelihood estimate of \( \mathbf{a} = (\alpha_0, \alpha_1, \ldots, \alpha_p) \) if \( \epsilon_t \) is assumed to be Gaussian error disturbances. Combining \( \mathbf{X} \) and \( \mathbf{y} \) as
\[
\mathbf{Z} = [\mathbf{X}:\mathbf{y}]
\]

the \( \mathbf{Z} \) matrix can be decomposed as
\[
\mathbf{Z} = \mathbf{Q}\mathbf{U} = \mathbf{Q} \begin{bmatrix}
\mathbf{R} & \mathbf{w}_1 \\
\mathbf{0} & \mathbf{w}_2
\end{bmatrix}
\]

where \( \mathbf{Q} \) is an orthogonal matrix and \( \mathbf{R} \) is an upper triangular matrix, \( \mathbf{w}_1 = (w_1, \ldots, w_{p+1})' \), and \( \mathbf{w}_2 = (w_{p+2}, 0, \ldots, 0)' \).

\[
\mathbf{Q}'\mathbf{y} = \begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_{T-p}
\end{bmatrix}
\]
The least squares estimate using Householder transformation is computed by solving the linear system

\[ Ra = w_1 \]

The unbiased residual variance estimate is

\[ \hat{\sigma}^2 = \frac{1}{T - p} \sum_{i=p+2}^{T-p} u_i^2 = \frac{u_{p+2}^2}{T - p} \]

and

\[ \text{AIC} = (T - p) \log(\hat{\sigma}^2) + 2(p + 1) \]

In practice, least squares estimation does not require the orthogonal matrix \( Q \). The TIMSAC subroutines compute the upper triangular matrix without computing the matrix \( Q \).

**Bayesian Constrained Least Squares**

Consider the additive time series model

\[ y_t = T_t + S_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2) \]

Practically, it is not possible to estimate parameters \( a = (T_1, \ldots, T_T, S_1, \ldots, S_T)' \), since the number of parameters exceeds the number of available observations. Let \( \nabla^m_L \) denote the seasonal difference operator with \( L \) seasons and degree of \( m \); that is, \( \nabla^m_L = (1 - B^L)^m \). Suppose that \( T = L \ast n \). Some constraints on the trend and seasonal components need to be imposed such that the sum of squares of \( \nabla^k T_t, \nabla^m_L S_t \), and \( (\sum_{i=0}^{L-1} S_{t-i}) \) is small. The constrained least squares estimates are obtained by minimizing

\[ \sum_{t=1}^{T} \left\{ (y_t - T_t - S_t)^2 + d^2 \left[ s^2 (\nabla^k T_t)^2 + (\nabla^m_L S_t)^2 + z^2 (S_t + \cdots + S_{t-L+1})^2 \right] \right\} \]

Using matrix notation,

\[ (y - Ma)'(y - Ma) + (a - a_0)'D'D(a - a_0) \]

where \( M = [I_T : I_T], y = (y_1, \ldots, y_T)' \), and \( a_0 \) is the initial guess of \( a \). The matrix \( D \) is a \( 3T \times 2T \) control matrix in which structure varies according to the order of differencing in trend and season.

\[
D = d \begin{bmatrix} E_m & 0 \\ zF & 0 \\ 0 & sG_k \end{bmatrix}
\]
where

\[
E_m = C_m \otimes I_L, \quad m = 1, 2, 3
\]

\[
F = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
1 & 1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
1 & \cdots & 1 & 1
\end{bmatrix}^{T \times T}
\]

\[
G_1 = \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
-1 & 1 & 0 & \cdots & 0 \\
0 & -1 & 1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & -1 & 1
\end{bmatrix}^{T \times T}
\]

\[
G_2 = \begin{bmatrix}
1 & 0 & 0 & 0 & \cdots & 0 \\
-2 & 1 & 0 & 0 & \cdots & 0 \\
1 & -2 & 1 & 0 & \cdots & 0 \\
0 & 1 & -2 & 1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & 1 & -2 & 1
\end{bmatrix}^{T \times T}
\]

\[
G_3 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
-3 & 1 & 0 & 0 & 0 & \cdots & 0 \\
3 & -3 & 1 & 0 & 0 & \cdots & 0 \\
-1 & 3 & -3 & 1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & -1 & 3 & -3 & 1
\end{bmatrix}^{T \times T}
\]

The \( n \times n \) matrix \( C_m \) has the same structure as the matrix \( G_m \), and \( I_L \) is the \( L \times L \) identity matrix. The solution of the constrained least squares method is equivalent to that of maximizing the following function

\[
L(a) = \exp \left\{ -\frac{1}{2\sigma^2}(y - Ma)'(y - Ma) \right\} \exp \left\{ -\frac{1}{2\sigma^2}(a - a_0)'D'D(a - a_0) \right\}
\]

Therefore, the PDF of the data \( y \) is

\[
f(y|\sigma^2, a) = \left( \frac{1}{2\pi} \right)^{T/2} \left( \frac{1}{\sigma} \right)^T \exp \left\{ -\frac{1}{2\sigma^2}(y - Ma)'(y - Ma) \right\}
\]

The prior PDF of the parameter vector \( a \) is

\[
\pi(a|D, \sigma^2, a_0) = \left( \frac{1}{2\pi} \right)^T \left( \frac{1}{\sigma} \right)^{2T} |D'D| \exp \left\{ -\frac{1}{2\sigma^2}(a - a_0)'D'D(a - a_0) \right\}
\]
When the constant $d$ is known, the estimate $\hat{a}$ of $a$ is the mean of the posterior distribution, where the posterior PDF of the parameter $a$ is proportional to the function $L(a)$. It is obvious that $\hat{a}$ is the minimizer of $\|g(a|d)\|^2 = (\hat{y} - D\hat{a})'(\hat{y} - D\hat{a})$,

$$\hat{y} = \begin{bmatrix} y \\ Da_0 \end{bmatrix}$$

$$D = \begin{bmatrix} M \\ D \end{bmatrix}$$

The value of $d$ is determined by the minimum ABIC procedure. The ABIC is defined as

$$ABIC = T \log \left( \frac{1}{T} \|g(a|d)\|^2 \right) + 2 \{ \log[\det(D'D + M'M)] - \log[\det(D'D)] \}$$

**State Space and Kalman Filter Method**

In this section, the mathematical formulas for state space modeling are introduced. The Kalman filter algorithms are derived from the state space model. As an example, the state space model of the TSDECOMP subroutine is formulated.

Define the following state space model:

$$x_t = Fx_{t-1} + Gw_t$$

$$y_t = H_t x_t + \epsilon_t$$

where $\epsilon_t \sim N(0, \sigma^2)$ and $w_t \sim N(0, Q)$. If the observations, $(y_1, \ldots, y_T)$, and the initial conditions, $x_0$ and $P_0$, are available, the one-step predictor $(x_{t|t-1})$ of the state vector $x_t$ and its mean square error (MSE) matrix $P_{t|t-1}$ are written as

$$x_{t|t-1} = Fx_{t-1|t-1}$$

$$P_{t|t-1} = FP_{t-1|t-1}F' + GQG'$$

Using the current observation, the filtered value of $x_t$ and its variance $P_{t|t}$ are updated.

$$x_{t|t} = x_{t|t-1} + K_t \epsilon_t$$

$$P_{t|t} = (I - K_t H_t)P_{t|t-1}$$

where $\epsilon_t = y_t - H_t x_{t|t-1}$ and $K_t = P_{t|t-1}H'_t (H_t P_{t|t-1} H'_t + \sigma^2 I)^{-1}$. The log-likelihood function is computed as

$$\ell = -\frac{1}{2} \sum_{t=1}^{T} \log(2\pi v_{t|t-1}) - \sum_{t=1}^{T} \frac{e_t^2}{2v_{t|t-1}}$$
where \(v_{d_{t-1}}\) is the conditional variance of the one-step prediction error \(\epsilon_t\).

Consider the additive time series decomposition

\[
y_t = T_t + S_t + TD_t + u_t + x_t' \beta_t + \epsilon_t
\]

where \(x_t\) is a \((K \times 1)\) regressor vector and \(\beta_t\) is a \((K \times 1)\) time-varying coefficient vector. Each component has the following constraints:

\[
\nabla^k T_t = w_{1t}, \quad w_{1t} \sim N(0, \tau_1^2) \\
\nabla^m S_t = w_{2t}, \quad w_{2t} \sim N(0, \tau_2^2) \\
\quad u_t = \sum_{i=1}^{p} \alpha_i u_{t-i} + w_{3t}, \quad w_{3t} \sim N(0, \tau_3^2) \\
\quad \beta_{jt} = \beta_{j,t-1} + w_{3+j,t}, \quad w_{3+j,t} \sim N(0, \tau_{3+j}^2), \quad j = 1, \ldots, K \\
\quad \sum_{i=1}^{7} \gamma_{it} TD_t(i) = \sum_{i=1}^{6} \gamma_{it} (TD_t(i) - TD_t(7)) \\
\gamma_{it} = \gamma_{i,t-1}
\]

where \(\nabla^k = (1 - B)^k\) and \(\nabla^m = (1 - B^L)^m\). The AR component \(u_t\) is assumed to be stationary. The trading day component \(TD_t(i)\) represents the number of the \(i\)th day of the week in time \(t\). If \(k = 3, p = 3, m = 1,\) and \(L = 12\) (monthly data),

\[
T_t = 3T_{t-1} - 3T_{t-2} + T_{t-3} + w_{1t} \\
\sum_{i=0}^{11} S_{t-i} = w_{2t} \\
\quad u_t = \sum_{i=1}^{3} \alpha_i u_{t-i} + w_{3t}
\]

The state vector is defined as

\[
x_t = (T_t, T_{t-1}, T_{t-2}, S_t, \ldots, S_{t-11}, u_t, u_{t-1}, u_{t-2}, \gamma_{1t}, \ldots, \gamma_{6t})'
\]

The matrix \(F\) is

\[
F = \begin{bmatrix} 
F_1 & 0 & 0 & 0 \\
0 & F_2 & 0 & 0 \\
0 & 0 & F_3 & 0 \\
0 & 0 & 0 & F_4 
\end{bmatrix}
\]

where

\[
F_1 = \begin{bmatrix} 
3 & -3 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 
\end{bmatrix}
\]
The matrix \( G \) can be denoted as

\[
G = \begin{bmatrix}
g_1 & 0 & 0 \\
0 & g_2 & 0 \\
0 & 0 & g_3 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{bmatrix}
\]

where

\[
g_1 = g_3 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}'
\]

\[
g_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}'
\]

Finally, the matrix \( H_t \) is time-varying,

\[
H_t = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & h_t' \end{bmatrix}
\]

where

\[
h_t = \begin{bmatrix} D_t(1) & D_t(2) & D_t(3) & D_t(4) & D_t(5) & D_t(6) \end{bmatrix}'
\]

\[
D_t(i) = TD_t(i) - TD_t(7), \quad i = 1, \ldots, 6
\]

**Missing Values**

The TIMSAC subroutines skip any missing values at the beginning of the data set. When the univariate and multivariate AR models are estimated via least squares (TSMLOCAR, TSMLOMAR, TSUNIMAR, TSMULMAR, and TSPEARS), there are three options available; that is, \texttt{MISSING=0}, \texttt{MISSING=1}, or \texttt{MISSING=2}. When the \texttt{MISSING=0} (default) option is specified, the first contiguous observations with no missing values are used. The \texttt{MISSING=1} option specifies that only
nonmissing observations should be used by ignoring the observations with missing values. If the MISSING=2 option is specified, the missing values are filled with the sample mean. The least squares estimator with the MISSING=2 option is biased in general.

The BAYSEA subroutine assumes the same prior distribution of the trend and seasonal components that correspond to the missing observations. A modification is made to skip the components of the vector $g(a_jd)$ that correspond to the missing observations. The vector $g(a_jd)$ is defined in the section, "Bayesian Constrained Least Squares". In addition, the TSBAYSEA subroutine considers outliers as missing values. The TSDECOMP and TSTVCAR subroutines skip the Kalman filter updating equation when the current observation is missing.

ISM TIMSAC Packages

A description of each TIMSAC package follows. Each description includes a list of the programs provided in the TIMSAC version.

TIMSAC-72

analyzes and controls the feedback systems (for example, cement kiln process). Univariate- and multivariate-AR models are employed in this original TIMSAC package. The final prediction error (FPE) criterion is used for model selection.

- AUSPEC estimates the power spectrum by the Blackman-Tukey procedure.
- AUTCOR computes autocovariance and autocorrelation.
- DECONV computes the impulse response function.
- FFTCOR computes autocorrelation and crosscorrelation via the fast Fourier transform.
- FPEAUT computes AR coefficients and FPE for the univariate AR model.
- FPEC computes AR coefficients and FPE for the control system or multivariate AR model.
- MULCOR computes multiple covariance and correlation.
- MULNOS computes relative power contribution.
- MULRSP estimates the rational spectrum for multivariate data.
- MULSPE estimates the cross spectrum by Blackman-Tukey procedure.
- OPTDES performs optimal controller design.
- OPTSIM performs optimal controller simulation.
- RASPEC estimates the rational spectrum for univariate data.
- SGLFRE computes the frequency response function.
- WNOISE performs white noise simulation.

TIMSAC-74

estimates and forecasts the univariate and multivariate ARMA models by fitting the canonical Markovian model. A locally stationary autoregressive model is also analyzed. Akaike’s information criterion (AIC) is used for model selection.
- AUTARM performs automatic univariate ARMA model fitting.
- BISPEC computes bispectrum.
- CANARM performs univariate canonical correlation analysis.
- CANOCA performs multivariate canonical correlation analysis.
- COVGEN computes the covariance from gain function.
- FRDPLY plots the frequency response function.
- MARKOV performs automatic multivariate ARMA model fitting.
- NONST estimates the locally stationary AR model.
- PRDCTR performs ARMA model prediction.
- PWDPLY plots the power spectrum.
- SIMCON performs optimal controller design and simulation.
- THIRMO computes the third-order moment.

**TIMSAC-78**

uses the Householder transformation to estimate the time series models. This package also contains Bayesian modeling and the exact maximum likelihood estimation of the ARMA model. Minimum AIC or Akaike Bayesian Information Criterion (ABIC) modeling is extensively used.

- BLOCAR estimates the locally stationary univariate AR model using the Bayesian method.
- BLOMAR estimates the locally stationary multivariate AR model using the Bayesian method.
- BSUBST estimates the univariate subset regression model using the Bayesian method.
- EXSAR estimates the univariate AR model using the exact maximum likelihood method.
- MLOCAR estimates the locally stationary univariate AR model using the minimum AIC method.
- MLOMAR estimates the locally stationary multivariate AR model using the minimum AIC method.
- MULBAR estimates the multivariate AR model using the Bayesian method.
- MULMAR estimates the multivariate AR model using the minimum AIC method.
- NADCON performs noise adaptive control.
- PERARS estimates the periodic AR model using the minimum AIC method.
- UNIBAR estimates the univariate AR model using the Bayesian method.
- UNIMAR estimates the univariate AR model using the minimum AIC method.
- XSARMA estimates the univariate ARMA model using the exact maximum likelihood method.
In addition, the following test subroutines are available: TSSBST, TSWIND, TS-ROOT, TSTIMS, and TSCANC.

**TIMSAC-84**

contains the Bayesian time series modeling procedure, the point process data analysis, and the seasonal adjustment procedure.

- ADAR estimates the amplitude dependent AR model.
- BAYSEA performs Bayesian seasonal adjustments.
- BAYTAP performs Bayesian tidal analysis.
- DECOMP performs time series decomposition analysis using state space modeling.
- EPTREN estimates intensity rates of either the exponential polynomial or exponential Fourier series of the nonstationary Poisson process model.
- LINLIN estimates linear intensity models of the self-exciting point process with another process input and with cyclic and trend components.
- LINSIM performs simulation of the point process estimated by the subroutine LINLIN.
- LOCCAR estimates the locally constant AR model.
- MULCON performs simulation, control, and prediction of the multivariate AR model.
- NONSPA performs nonstationary spectrum analysis using the minimum Bayesian AIC procedure.
- PGRAPH performs graphical analysis for point process data.
- PTSPEC computes periodograms of point process data with significant bands.
- SIMBVH performs simulation of bivariate Hawkes’ mutually exciting point process.
- SNDE estimates the stochastic nonlinear differential equation model.
- TVCAR estimates the time-varying AR coefficient model using state space modeling.

Refer to Kitagawa and Akaike (1981) and Ishiguro (1987) for more information about TIMSAC programs.
Example 10.1. VAR Estimation and Variance Decomposition

In this example, a VAR model is estimated and forecast. The VAR(3) model is estimated using investment, durable consumption, and consumption expenditures. The data are found in the appendix to Lutkepohl (1991). The stationary VAR(3) process is specified as

\[ y_t = A_0 + A_1 y_{t-1} + A_2 y_{t-2} + A_3 y_{t-3} + \epsilon_t \]

The matrix ARCOEF contains the AR coefficients \((A_1, A_2, \text{ and } A_3)\), and the matrix EV contains error covariance estimates. An intercept vector \(A_0\) is included in the first row of the matrix ARCOEF if OPT[1]=1 is specified.

data one;
input invest income consum @@;
datalines;
180 451 415 179 465 421 185 485 434 192 493 448
211 509 459 202 520 458 207 521 479 214 540 487
231 548 497 229 558 510 234 574 516 237 583 525
206 591 529 250 599 538 259 610 546 263 627 555
264 642 574 280 653 574 282 660 586 292 694 602
286 709 617 302 734 639 234 574 516 237 583 525
317 766 679 314 779 686 306 808 697 307 763 668
292 794 704 275 799 699 273 799 709 301 812 715
280 837 724 289 853 746 303 876 758 322 897 779
315 922 798 339 949 816 364 979 837 371 988 858
375 1025 881 432 1063 905 453 1104 934 460 1131 968
475 1137 983 496 1178 1013 494 1211 1034 498 1256 1064
526 1290 1101 519 1314 1102 516 1346 1145 531 1385 1173
573 1416 1216 551 1436 1229 538 1462 1242 532 1493 1267
558 1516 1295 524 1557 1317 525 1613 1355 519 1642 1371
526 1690 1402 510 1759 1452 519 1756 1485 538 1780 1516
549 1807 1549 570 1831 1567 559 1873 1588 584 1897 1631
611 1910 1650 597 1943 1685 603 1976 1722 619 2018 1752
635 2040 1774 658 2070 1807 675 2121 1831 700 2132 1842
692 2199 1890 759 2253 1958 782 2276 1948 816 2318 1994
844 2369 2061 830 2423 2056 853 2457 2102 852 2470 2121
833 2521 2145 860 2545 2164 870 2580 2206 830 2620 2225
801 2639 2235 824 2618 2237 831 2628 2250 830 2651 2271
;
proc iml;
use one;
read all into y var(invest income consum);
mdel = 1;
maice = 0;
misw = 0; /*-- instantaneous modeling ? --*/
call tsmulmar(arcoef,ev,nar,aic) data=y maxlag=3
   opt=(mdel || maice || misw) print=1;

To obtain the unit triangular matrix \(L^{-1}\) and diagonal matrix \(D_t\), you need to estimate the instantaneous response model. When you specify the OPT[3]=1 option, the first row of the output matrix EV contains error variances of the instantaneous response model, while the unit triangular matrix is in the second through the fifth rows. See Output 10.1.1 on page 274.
misw = 1;
call tsmulmar(arcoef,ev,nar,aic) data=y maxlag=3
   opt=(mdel || maice || misw) print=1;
print ev;

Output 10.1.1. Error Variance and Unit Triangular Matrix

\[
EV = \begin{bmatrix}
295.21042 & 190.94664 & 59.361516 \\
1 & 0 & 0 \\
-0.02239 & 1 & 0 \\
-0.256341 & -0.500803 & 1
\end{bmatrix}
\]

In Output 10.1.2 on page 274 and Output 10.1.3 on page 275, you can see the relationship between the instantaneous response model and the VAR model. The VAR coefficients are computed as \( A_i = L A_i^* (i = 0, 1, 2, 3) \), where \( A_i^* \) is a coefficient matrix of the instantaneous model. For example, you can verify this result using the first lag coefficient matrix \( A_1 \).

\[
\begin{bmatrix}
0.886 & 0.340 & -0.014 \\
0.168 & 1.050 & 0.107 \\
0.089 & 0.459 & 0.447
\end{bmatrix}
= \begin{bmatrix}
1.000 & 0 & 0 \\
-0.022 & 1.000 & 0 \\
-0.256 & -0.501 & 1.000
\end{bmatrix}^{-1}
\begin{bmatrix}
0.886 & 0.340 & -0.014 \\
0.149 & 1.043 & 0.107 \\
-0.222 & -0.154 & 0.397
\end{bmatrix}
\]

Output 10.1.2. VAR Estimates

\[
\begin{array}{ccc|ccc}
\text{LAG} & \text{VAR = 1} & \text{VAR = 2} & \text{VAR = 3} \\
\hline
1 & 0.885593 & 0.340174 & -0.014398 \\
& 0.168452 & 1.050262 & 0.107064 \\
& 0.089103 & 0.459157 & 0.447367 \\
2 & -0.059195 & -0.298777 & 0.162982 \\
& 0.112862 & -0.044039 & -0.088186 \\
& 0.168493 & -0.025847 & -0.025671 \\
3 & 0.063723 & -0.196504 & 0.069575 \\
& -0.226559 & 0.053247 & -0.099808 \\
& -0.303697 & -0.139022 & 0.257641
\end{array}
\]
Example 10.1. VAR Estimation and Variance Decomposition

Output 10.1.3. Instantaneous Response Model Estimates

<table>
<thead>
<tr>
<th>-----</th>
<th>AR</th>
<th>-----</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.885593</td>
<td>0.340174</td>
<td>-0.014398</td>
</tr>
<tr>
<td>0.148624</td>
<td>1.042645</td>
<td>0.107386</td>
</tr>
<tr>
<td>-0.222272</td>
<td>-0.154018</td>
<td>0.397440</td>
</tr>
<tr>
<td>-0.059195</td>
<td>-0.298777</td>
<td>0.162982</td>
</tr>
<tr>
<td>0.114188</td>
<td>-0.037349</td>
<td>-0.091835</td>
</tr>
<tr>
<td>0.127145</td>
<td>0.072796</td>
<td>-0.023287</td>
</tr>
<tr>
<td>0.063723</td>
<td>-0.196504</td>
<td>0.069575</td>
</tr>
<tr>
<td>-0.227986</td>
<td>0.057646</td>
<td>-0.101366</td>
</tr>
<tr>
<td>-0.206570</td>
<td>-0.115316</td>
<td>0.289790</td>
</tr>
</tbody>
</table>

When the VAR estimates are available, you can forecast the future values using the TSPRED call. As a default, the one-step predictions are produced until the START= point is reached. The NPRED= option specifies how far you want to predict. The prediction error covariance matrix MSE contains h mean square error matrices. The output matrix IMPULSE contains the estimate of the coefficients (Ψ_i) of the infinite MA process. The following IML code estimates the VAR(3) model and performs 10-step-ahead prediction.

```sas
mdel = 1;
maice = 0;
misw = 0;
call tsmulmar(arcoef,ev,nar,aic) data=y maxlag=3
   opt=(mdel || maice || misw);
call tspred(forecast,impulse,mse,y,arcoef,nar,0,ev)
   npred=10 start=nrow(y) constant=mdel;
print impulse;
```

The lagged effects of a unit increase in the error disturbances are included in the matrix IMPULSE. For example,

$$\frac{\partial y_{t+2}}{\partial \epsilon_t} = \begin{bmatrix} 0.781100 & 0.353140 & 0.180211 \\ 0.448501 & 1.165474 & 0.069731 \\ 0.364611 & 0.692111 & 0.222342 \end{bmatrix}$$

Output 10.1.4 on page 276 displays the first 15 rows of the matrix IMPULSE.
Output 10.1.4.  Moving Average Coefficients: MA(0)—MA(4)

\[
\begin{array}{cccc}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0.8855926 & 0.3401741 & -0.014398 \\
0.1684523 & 1.0502619 & 0.107064 \\
0.0891034 & 0.4591573 & 0.4473672 \\
0.7810999 & 0.3531397 & 0.1802109 \\
0.4485013 & 1.1654737 & 0.0697311 \\
0.3646106 & 0.6921108 & 0.2223425 \\
0.8145483 & 0.243637 & 0.2914643 \\
0.4997732 & 1.3625363 & 0.018202 \\
0.2775237 & 0.7555914 & 0.3885065 \\
0.7960884 & 0.2593068 & 0.260239 \\
0.5275069 & 1.4134792 & 0.0335483 \\
0.267452 & 0.8659426 & 0.3190203 \\
\end{array}
\]

In addition, you can compute the lagged response on the one-unit increase in the orthogonalized disturbances \( \epsilon_t \).

\[
\frac{\partial y_{t+m}}{\partial \epsilon_{jt}} = \frac{\partial E(y_{t+m}|y_{jt}, y_{j-1,t}, \ldots, X_t)}{\partial y_{jt}} = \psi_m L_j
\]

When the error matrix EV is obtained from the instantaneous response model, you need to convert the matrix IMPULSE. The first 15 rows of the matrix ORTH_IMP are shown in Output 10.1.5 on page 276. Note that the matrix constructed from the last three rows of EV become the matrix \( L^{-1} \).

```
call tsmulmar(arcoef,ev,nar,aic) data=y maxlag=3
opt={1 0 1};
lmtx = inv(ev[2:nrow(ev),]);
orth_imp = impulse * lmtx;
print orth_imp;
```

Output 10.1.5.  Transformed Moving Average Coefficients

\[
\begin{array}{cccc}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0.0223902 & 1 & 0 \\
0.267554 & 0.5008031 & 1 \\
0.889357 & 0.3329638 & -0.014398 \\
0.2206132 & 1.1038799 & 0.107064 \\
0.219079 & 0.6832001 & 0.4473672 \\
0.8372229 & 0.4433899 & 0.1802109 \\
0.4932533 & 1.2003953 & 0.0697311 \\
0.4395957 & 0.8034606 & 0.2223425 \\
0.8979658 & 0.3896033 & 0.2914643 \\
0.5254106 & 1.3534206 & -0.018202 \\
0.398388 & 0.9501566 & 0.3885065 \\
0.8715223 & 0.3896353 & 0.260239 \\
0.5681309 & 1.4302804 & 0.0335483 \\
0.3721958 & 1.025709 & 0.3190203 \\
\end{array}
\]
Example 10.1. VAR Estimation and Variance Decomposition

You can verify the result for the case of

\[
\frac{\partial y_{t+2}}{\partial \varepsilon_{2t}} = \frac{\partial \mathbb{E}(y_{t+2}|y_{2t}, y_{1t}, \ldots, X_t)}{\partial y_{2t}} = \Psi_2 L_2
\]

using the simple computation

\[
\begin{bmatrix}
0.443390 \\
1.200395 \\
0.803461
\end{bmatrix} =
\begin{bmatrix}
0.781100 & 0.353140 & 0.180211 \\
0.448501 & 1.165474 & 0.069731 \\
0.364611 & 0.692111 & 0.222342
\end{bmatrix}
\]

The contribution of the \(i\)th orthogonalized innovation to the mean square error matrix of the 10-step forecast is computed using the formula

\[
d_{ii} = L_i L_i' + \Psi_i L_i L_i' \Psi_i + \cdots + \Psi_9 L_i L_i' \Psi_9
\]

In Output 10.1.6 on page 277, diagonal elements of each decomposed MSE matrix are displayed as the matrix CONTRIB as well as those of the MSE matrix (VAR).

```sas
mse1 = j(3,3,0);
mse2 = j(3,3,0);
mse3 = j(3,3,0);
do i = 1 to 10;
  psi = impulse[(i-1)*3+1:3*i,];
  mse1 = mse1 + psi*lmtx[,1]*lmtx[,1]'*psi';
  mse2 = mse2 + psi*lmtx[,2]*lmtx[,2]'*psi';
  mse3 = mse3 + psi*lmtx[,3]*lmtx[,3]'*psi';
end;
mse1 = ev[1,1]*mse1;
mse2 = ev[1,2]*mse2;
mse3 = ev[1,3]*mse3;
contrib = vecdiag(mse1) || vecdiag(mse2) || vecdiag(mse3);
var = vecdiag(mse[28:30,]);
print contrib var;
```

**Output 10.1.6.** Orthogonal Innovation Contribution

<table>
<thead>
<tr>
<th>CONTRIB</th>
<th>VAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1879.3774</td>
<td>238.08543</td>
</tr>
<tr>
<td>935.54383</td>
<td>3636.8824</td>
</tr>
<tr>
<td>452.67794</td>
<td>1916.1676</td>
</tr>
</tbody>
</table>

The investment innovation contribution to its own variable is 1879.3774, and the income innovation contribution to the consumption expenditure is 1916.1676. It is easy to understand the contribution of innovations in the \(i\)th variable to MSE when you compute the innovation account. In Output 10.1.7 on page 278, innovations in the first variable (investment) explain 20.45% of the error variance of the second variable.
variable (income), while the innovations in the second variable explain 79.5% of its own error variance. It is straightforward to construct the general multistep forecast error variance decomposition.

```sas
account = contrib * 100 / (var@j(1,3,1));
print account;
```

**Output 10.1.7.** Innovation Account

<table>
<thead>
<tr>
<th>ACCOUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>86.859008</td>
</tr>
<tr>
<td>20.453602</td>
</tr>
<tr>
<td>18.353004</td>
</tr>
<tr>
<td>11.003572</td>
</tr>
<tr>
<td>79.512409</td>
</tr>
<tr>
<td>77.687531</td>
</tr>
<tr>
<td>2.1374196</td>
</tr>
<tr>
<td>0.0339894</td>
</tr>
<tr>
<td>3.9594646</td>
</tr>
</tbody>
</table>
Kalman Filter Subroutines

This section describes a collection of Kalman filtering and smoothing subroutines for time series analysis; immediately following are three examples using Kalman filtering subroutines. The state space model is a method for analyzing a wide range of time series models. When the time series is represented by the state space model (SSM), the Kalman filter is used for filtering, prediction, and smoothing of the state vector. The state space model is composed of the measurement and transition equations. The measurement (or observation) equation can be written

\[ y_t = b_t + H_t z_t + \epsilon_t \]

where \( b_t \) is an \( N_y \times 1 \) vector, \( H_t \) is an \( N_y \times N_z \) matrix, the sequence of observation noise \( \epsilon_t \) is independent, \( z_t \) is an \( N_z \times 1 \) state vector, and \( y_t \) is an \( N_y \times 1 \) observed vector.

The transition (or state) equation is denoted as a first-order Markov process of the state vector.

\[ z_{t+1} = a_t + F_t z_t + \eta_t \]

where \( a_t \) is an \( N_z \times 1 \) vector, \( F_t \) is an \( N_z \times N_z \) transition matrix, and the sequence of transition noise \( \eta_t \) is independent. This equation is often called a shifted transition equation because the state vector is shifted forward one time period. The transition equation can also be denoted using an alternative specification

\[ z_t = a_t + F_t z_{t-1} + \eta_t \]

There is no real difference between the shifted transition equation and this alternative equation if the observation noise and transition equation noise are uncorrelated, that is, \( E(\eta_t \epsilon_t^t) = 0 \). It is assumed that

\[
\begin{align*}
E(\eta_t \eta_t^t) &= V_t \delta_{ts} \\
E(\epsilon_t \epsilon_t^t) &= R_t \delta_{ts} \\
E(\eta_t \epsilon_t^t) &= G_t \delta_{ts}
\end{align*}
\]

where

\[ \delta_{ts} = \begin{cases} 
1 & \text{if } t = s \\ 
0 & \text{if } t \neq s 
\end{cases} \]

De Jong (1991a) proposed a diffuse Kalman filter that can handle an arbitrarily large initial state covariance matrix. The diffuse initial state assumption is reasonable if
you encounter the case of parameter uncertainty or SSM nonstationarity. The SSM of the diffuse Kalman filter is written

\[
\begin{align*}
\mathbf{y}_t &= \mathbf{X}_t \beta + \mathbf{H}_t \mathbf{zt} + \epsilon_t \\
\mathbf{zt} + 1 &= \mathbf{W}_t \beta + \mathbf{F}_t \mathbf{zt} + \eta_t \\
\mathbf{z}_0 &= \mathbf{a} + \mathbf{A} \delta \\
\beta &= b + B \delta
\end{align*}
\]

where \( \delta \) is a random variable with a mean of \( \mu \) and a variance of \( \sigma^2 \Sigma \). When \( \Sigma \to \infty \), the SSM is said to be diffuse.

The following IML Kalman filter calls are supported:

- KALCVF performs covariance filtering and prediction
- KALCVS performs fixed-interval smoothing
- KALDFF performs diffuse covariance filtering and prediction
- KALDFS performs diffuse fixed-interval smoothing

The KALCVF call computes the one-step prediction \( \mathbf{z}_{t+1|t} \) and the filtered estimate \( \mathbf{z}_{t|t} \), together with their covariance matrices \( \mathbf{P}_{t+1|t} \) and \( \mathbf{P}_{t|t} \), using forward recursions. You can obtain the \( k \)-step prediction \( \mathbf{z}_{t+k|t} \) and its covariance matrix \( \mathbf{P}_{t+k|t} \) with the KALCVF call. The KALCVS call uses backward recursions to compute the smoothed estimate \( \mathbf{z}_{t|T} \) and its covariance matrix \( \mathbf{P}_{t|T} \) when there are \( T \) observations in the complete data.

The KALDFF call produces one-step prediction of the state and the unobserved random vector \( \delta \) as well as their covariance matrices. The KALDFS call computes the smoothed estimate \( \mathbf{z}_{t|T} \) and its covariance matrix \( \mathbf{P}_{t|T} \).

See Chapter 17, "Language Reference," for more information about Kalman filtering subroutines.

**Example 10.2. Kalman Filtering: Likelihood Function Evaluation**

In this example, the log likelihood function of the SSM is computed using prediction error decomposition. The annual real GNP series, \( y_t \), can be decomposed as

\[
y_t = \mu_t + \epsilon_t
\]

where \( \mu_t \) is a trend component and \( \epsilon_t \) is a white noise error with \( \epsilon_t \sim (0, \sigma^2) \). Refer to Nelson and Plosser (1982) for more details on these data. The trend component is assumed to be generated from the following stochastic equations:

\[
\begin{align*}
\mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_1 t \\
\beta_t &= \beta_{t-1} + \eta_2 t
\end{align*}
\]
where $\eta_{1t}$ and $\eta_{2t}$ are independent white noise disturbances with $\eta_{1t} \sim (0, \sigma_{\eta_1}^2)$ and $\eta_{2t} \sim (0, \sigma_{\eta_2}^2)$.

It is straightforward to construct the SSM of the real GNP series.

$$y_t = H z_t + \epsilon_t$$
$$z_t = F z_{t-1} + \eta_t$$

where

$$H = (1, 0)$$
$$F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$
$$z_t = (\mu_t, \beta_t)'$$
$$\eta_t = (\eta_{1t}, \eta_{2t})'$$

$$\text{Var} \left( \begin{bmatrix} \epsilon_t \\ \eta_{lt} \end{bmatrix} \right) = \begin{bmatrix} \sigma_{\epsilon}^2 & 0 & 0 \\ 0 & \sigma_{\eta_1}^2 & 0 \\ 0 & 0 & \sigma_{\eta_2}^2 \end{bmatrix}$$

When the observation noise $\epsilon_t$ is normally distributed, the average log likelihood function of the SSM is

$$\ell = \frac{1}{T} \sum_{t=1}^{T} \ell_t$$
$$\ell_t = -\frac{N_y}{2} \log(2\pi) - \frac{1}{2} \log(|C_t|) - \frac{1}{2} \hat{\epsilon}_t' C_t^{-1} \hat{\epsilon}_t$$

where $C_t$ is the mean square error matrix of the prediction error $\hat{\epsilon}_t$, such that $C_t = HP_{[t-1]}H' + R_t$.

The LIK module computes the average log likelihood function. First, the average log likelihood function is computed using the default initial values: $Z0=0$ and $VZ0=10^6 I$.

The second call of module LIK produces the average log likelihood function with the given initial conditions: $Z0=0$ and $VZ0=10^{-3} I$. You can notice a sizable difference between the uncertain initial condition ($VZ0=10^6 I$) and the almost deterministic initial condition ($VZ0=10^{-3} I$) in Output 10.2.1.

Finally, the first 15 observations of one-step predictions, filtered values, and real GNP series are produced under the moderate initial condition ($VZ0=101 I$). The data are the annual real GNP for the years 1909 to 1969.

```sas
title 'Likelihood Evaluation of SSM';
title2 'DATA: Annual Real GNP 1909-1969';
data gnp;
  input y @@;
```
Chapter 10. Time Series Analysis and Control Examples

datalines;
116.8 120.1 123.2 130.2 131.4 125.6 124.5 134.3
135.2 151.8 146.4 139.0 127.8 165.9 165.5
179.4 190.0 189.8 190.9 203.6 169.3 144.2
141.5 154.3 169.5 193.0 203.2 192.9 209.4 227.2
263.7 297.8 337.1 361.3 355.2 312.6 309.9 323.7
324.1 355.3 383.4 395.1 412.8 406.0 438.0 446.1
452.5 447.3 475.9 487.7 497.2 529.8 551.0 581.1
617.8 658.1 675.2 706.6 724.7
;

proc iml;
start lik(y,a,b,f,h,var,z0,vz0);
   nz = nrow(f);
   n = nrow(y);
   k = ncol(y);
   const = k*log(8*atan(1));
   if ( sum(z0 = .) | sum(vz0 = .) ) then
      call kalcvf(pred,vpred,filt,vfilt,y,0,a,f,b,h,var);
   else
      call kalcvf(pred,vpred,filt,vfilt,y,0,a,f,b,h,var,z0,vz0);
   et = y - pred*h';
   sum1 = 0;
   sum2 = 0;
   do i = 1 to n;
      vpred_i = vpred[(i-1)*nz+1:i*nz,];
      et_i = et[i,];
      ft = h*vpred_i*h' + var[nz+1:nz+k,nz+1:nz+k];
      sum1 = sum1 + log(det(ft));
      sum2 = sum2 + et_i*inv(ft)*et_i';
   end;
   return(-const-.5*(sum1+sum2)/n);
finish;

start main;
use gnp;
read all var {y};
   f = {1 1, 0 1};
   h = {1 0};
   a = j(nrow(f),1,0);
   b = j(nrow(h),1,0);
   var = diag(j(1,nrow(f)+ncol(y),1e-3));
   /*-- initial values are computed --*/
   z0 = j(1,nrow(f),.);
   vz0 = j(nrow(f),1e-3);;
   logl = lik(y,a,b,f,h,var,z0,vz0);
   print 'No initial values are given', logl;
   /*-- initial values are given --*/
   z0 = j(1,nrow(f),0);
   vz0 = 1e-3#i(nrow(f));
   logl = lik(y,a,b,f,h,var,z0,vz0);
   print 'Initial values are given', logl;
   z0 = j(1,nrow(f),0);
print 'No initial values are given', logl;
print 'Initial values are given', logl;
finish;
Example 10.3. Kalman Filtering: Estimating an SSM

\begin{verbatim}
vz0 = 10*i(nrow(f));
call kalcvf(pred,vpred,filt,vfilt,y,1,a,f,b,h,var,z0,vz0);
print y pred filt;
finish;
run;
\end{verbatim}

Output 10.2.1. Average Log Likelihood of SSM

<table>
<thead>
<tr>
<th>Likelihood Evaluation of SSM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DATA:</strong> Annual Real GNP 1909-1969</td>
</tr>
</tbody>
</table>

No initial values are given

\begin{verbatim}
LOGL
-26314.66
\end{verbatim}

Initial values are given

\begin{verbatim}
LOGL
-91884.41
\end{verbatim}

Output 10.2.2 shows the observed data, the predicted state vectors, and the filtered state vectors for the first 16 observations.

Output 10.2.2. Filtering and One-Step Prediction

<table>
<thead>
<tr>
<th>Y</th>
<th>PRED</th>
<th>FILT</th>
</tr>
</thead>
<tbody>
<tr>
<td>116.8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120.1</td>
<td>116.78832</td>
<td>0</td>
</tr>
<tr>
<td>123.2</td>
<td>123.41035</td>
<td>3.3106857</td>
</tr>
<tr>
<td>130.2</td>
<td>126.41721</td>
<td>3.1938303</td>
</tr>
<tr>
<td>131.4</td>
<td>134.47459</td>
<td>4.8825531</td>
</tr>
<tr>
<td>125.6</td>
<td>135.51391</td>
<td>3.5758561</td>
</tr>
<tr>
<td>124.5</td>
<td>126.75246</td>
<td>-0.610017</td>
</tr>
<tr>
<td>134.3</td>
<td>123.34052</td>
<td>-1.560708</td>
</tr>
<tr>
<td>135.2</td>
<td>135.41265</td>
<td>3.0651076</td>
</tr>
<tr>
<td>151.8</td>
<td>138.21324</td>
<td>2.9753526</td>
</tr>
<tr>
<td>146.4</td>
<td>158.08957</td>
<td>8.7100967</td>
</tr>
<tr>
<td>139</td>
<td>152.25867</td>
<td>3.7761324</td>
</tr>
<tr>
<td>127.8</td>
<td>139.5419</td>
<td>-1.82012</td>
</tr>
<tr>
<td>147</td>
<td>123.11568</td>
<td>-6.776195</td>
</tr>
<tr>
<td>165.9</td>
<td>146.04988</td>
<td>3.3049584</td>
</tr>
<tr>
<td>165.5</td>
<td>174.04698</td>
<td>11.683345</td>
</tr>
</tbody>
</table>

Example 10.3. Kalman Filtering: Estimating an SSM Using the EM Algorithm

This example estimates the normal SSM of the mink-muskrat data using the EM algorithm. The mink-muskrat series are detrended. Refer to Harvey (1989) for details of this data set. Since this EM algorithm uses filtering and smoothing, you can learn
how to use the KALCVF and KALCVS calls to analyze the data. Consider the bivariate SSM:

\[
\begin{align*}
\mathbf{y}_t &= \mathbf{Hz}_t + \epsilon_t \\
\mathbf{z}_t &= \mathbf{Fz}_{t-1} + \eta_t
\end{align*}
\]

where \( \mathbf{H} \) is a \( 2 \times 2 \) identity matrix, the observation noise has a time invariant covariance matrix \( \mathbf{R} \), and the covariance matrix of the transition equation is also assumed to be time invariant. The initial state \( \mathbf{z}_0 \) has mean \( \mu \) and covariance \( \Sigma \). For estimation, the \( \Sigma \) matrix is fixed as

\[
\begin{bmatrix}
0.1 & 0.0 \\
0.0 & 0.1
\end{bmatrix}
\]

while the mean vector \( \mu \) is updated by the smoothing procedure such that \( \hat{\mu} = \mathbf{z}_{0|T} \). Note that this estimation requires an extra smoothing step since the usual smoothing procedure does not produce \( \mathbf{z}_{T|0} \).

The EM algorithm maximizes the expected log likelihood function given the current parameter estimates. In practice, the log likelihood function of the normal SSM is evaluated while the parameters are updated using the M-step of the EM maximization

\[
\begin{align*}
\mathbf{F}^{i+1} &= \mathbf{S}_t(1)[\mathbf{S}_{t-1}(0)]^{-1} \\
\mathbf{V}^{i+1} &= \frac{1}{T} (\mathbf{S}_t(0) - \mathbf{S}_t(0)[\mathbf{S}_{t-1}(0)]^{-1}\mathbf{S}_t'(1)) \\
\mathbf{R}^{i+1} &= \frac{1}{T} \sum_{t=1}^T [(\mathbf{y}_t - \mathbf{Hz}_{t|T})(\mathbf{y}_t - \mathbf{Hz}_{t|T})' + \mathbf{HP}_{t|T}\mathbf{H}'] \\
\mu^{i+1} &= \mathbf{z}_{0|T}
\end{align*}
\]

where the index \( i \) represents the current iteration number, and

\[
\begin{align*}
\mathbf{S}_t(0) &= \sum_{t=1}^T (\mathbf{P}_{t|T} + \mathbf{z}_t|T\mathbf{z}_t'|T), \\
\mathbf{S}_t(1) &= \sum_{t=1}^T (\mathbf{P}_{t,t-1|T} + \mathbf{z}_t|T\mathbf{z}_{t-1}|T)
\end{align*}
\]

It is necessary to compute the value of \( \mathbf{P}_{t,t-1|T} \) recursively such that

\[
\mathbf{P}_{t-1,t-2|T} = \mathbf{P}_{t-1,t-1|T}\mathbf{P}^{t-1}_{t-2} + \mathbf{P}^{t-1}_{t-1}(\mathbf{P}_{t,t-1|T} - \mathbf{F}\mathbf{P}_{t-1|t-1})\mathbf{P}^{t-1}_{t-2}
\]

where \( \mathbf{P}^{t-1}_{t} = \mathbf{P}_{t|T}\mathbf{F}'\mathbf{P}^{-1}_{t+1|t} \) and the initial value \( \mathbf{P}_{T,T-1|T} \) is derived using the formula

\[
\mathbf{P}_{T,T-1|T} = [\mathbf{I} - \mathbf{P}_{t|T-1}\mathbf{H}'(\mathbf{HP}_{t|T-1}\mathbf{H}' + \mathbf{R})\mathbf{H}] \mathbf{F}\mathbf{P}_{T-1|T-1}
\]
Note that the initial value of the state vector is updated for each iteration

\[
\begin{align*}
\mathbf{z}_{1|0} &= F \mu^i \\
\mathbf{P}_{1|0} &= F^i \Sigma F^i + \mathbf{V}^i
\end{align*}
\]

The objective function value is computed as \(-2\ell\) in the IML module LIK. The log-likelihood function is written

\[
\ell = -\frac{1}{2} \sum_{t=1}^{T} \log(|C_t|) - \frac{1}{2} \sum_{t=1}^{T} (\mathbf{y}_t - \mathbf{H}\mathbf{z}_{t|t-1})C_t^{-1}(\mathbf{y}_t - \mathbf{H}\mathbf{z}_{t|t-1})'
\]

where \(C_t = \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}' + \mathbf{R}\).

The iteration history is shown in Output 10.3.1. As shown in Output 10.3.2, the eigenvalues of \(F\) are within the unit circle, which indicates that the SSM is stationary. However, the muskrat series (Y1) is reported to be difference stationary. The estimated parameters are almost identical to those of the VAR(1) estimates. Refer to Harvey (1989, p. 469). Finally, multistep forecasts of \(y_t\) are computed using the KALCVF call.

\[
call \texttt{kalcvf}(\texttt{pred}, \texttt{vpred}, \texttt{filt}, \texttt{vfilt}, \texttt{y}, 15, \texttt{a}, \texttt{f}, \texttt{b}, \texttt{h}, \texttt{var}, \texttt{z0}, \texttt{vz0});
\]

The predicted values of the state vector \(\mathbf{z}_t\) and their standard errors are shown in Output 10.3.3.

```sas
proc iml;
start lik(y,pred,vpred,h,rt);
    n = nrow(y);
    nz = ncol(h);
    et = y - pred*h';
    sum1 = 0;
    sum2 = 0;
    do i = 1 to n;
        vpred_i = vpred[(i-1)*nz+1:i*nz,];
        et_i = et[i,];
        ft = h*vpred_i*h' + rt;
        sum1 = sum1 + log(det(ft));
        sum2 = sum2 + et_i*inv(ft)*et_i';
    end;
    return(sum1+sum2);
finish;

start main;
    use one;
    read all into y var {y1 y2};
```

SAS OnlineDoc™: Version 8
Chapter 10. Time Series Analysis and Control Examples

/*-- mean adjust series --*/
t = nrow(y);
ny = ncol(y);
nz = ny;
f = i(nz);
h = i(ny);

/*-- observation noise variance is diagonal --*/
rt = 1e-5*i(ny);

/*-- transition noise variance --*/
vt = .1*i(nz);
a = j(nz,1,0);
b = j(ny,1,0);
myu = j(nz,1,0);
sigma = .1*i(nz);
converge = 0;
do iter = 1 to 100 while( converge = 0 );

/*--- construct big cov matrix ---*/
var = ( vt || j(nz,ny,0) ) //
( j(ny,nz,0) || rt );

/*--- initial values are changed ---*/
z0 = myu' * f';
vz0 = f * sigma * f' + vt;

/*--- filtering to get one-step prediction and filtered value ---*/
call kalcvf(pred,vpred,filt,vfilt,y,0,a,f,b,h,var,z0,vz0);

/*--- smoothing using one-step prediction values ---*/
call kalcvs(sm,vsm,y,a,f,b,h,var,pred,vpred);

/*--- compute likelihood values ---*/
logl = lik(y,pred,vpred,h,rt);

/*--- store old parameters and function values ---*/
myu0 = myu;
f0 = f;
vt0 = vt;
rt0 = rt;
logl0 = logl;
itermat = itermat // ( iter || logl0 || shape(f0,1) || myu0' );

/*-- obtain P*(t) to get P_T_0 and Z_T_0 ---*/
/*-- these values are not usually needed ---*/
/*-- See Harvey (1989 p154) or Shumway (1988, p177) ---*/
jt1 = sigma * f' * inv(vpred[1:nz]);
p_t_0 = sigma + jt1*(vsm[1:nz] - vpred[1:nz])*jt1';
z_t_0 = myu + jt1*(sm[1,]' - pred[1,]');
p_t1_t = vpred[(t-1)*nz+1:t*nz];
p_t1_t1 = vfilt[(t-2)*nz+1:(t-1)*nz];
kt = p_t1_t*h'*inv(h*p_t1_t*h'+rt);

/*-- obtain P_T_TT1. See Shumway (1988, p180) ---*/
p_t_t11 = (i(nz)-kt*h)*f*p_t1_t1;
st0 = vsm[(t-1)*nz+1:t*nz] + sm[t,]'*sm[t,];
Example 10.3. Kalman Filtering: Estimating an SSM

\[\begin{align*}
st_1 &= p_{t_{i1}} + sm[t,] \cdot sm[t-1,]; \\
st_{00} &= p_{t_0} + z_{t_0} \cdot z_{t_0}'; \\
cov &= (y[t,] - h*sm[t,]) \cdot (y[t,] - h*sm[t,])' + \\
& \quad h*sm[t-1,]*h'; \\
do i = t to 2 by -1; \\
p_{i1_{i1}} &= vfilt[(i-2)*nz+1:(i-1)*nz,]; \\
p_{i1_i} &= vpred[(i-1)*nz+1:i*nz,]; \\
jt1 &= p_{i1_{i1}} * f' * inv(p_{i1_i}); \\
p_{i1_i} &= vpred[(i-2)*nz+1:(i-1)*nz,]; \\
if ( i > 2 ) then \\
p_{i2_i2} &= vfilt[(i-3)*nz+1:(i-2)*nz,]; \\
else \\
p_{i2_i2} &= sigma; \\
jt2 &= p_{i2_i2} * f' * inv(p_{i1_i}); \\
p_{t_{i1i2}} &= p_{i1_{i1}} \cdot jt2' + jt1*(p_{t_{i1}} - f*p_{i1_{i1}})*jt2'; \\
p_{t_{i1i2}} &= p_{i1_{i1}} \cdot jt2' + jt1*(p_{t_{i1}} - f*p_{i1_{i1}})*jt2'; \\
end;
\end{align*}\]

/*-- M-step: update the parameters --*/

myu = z_{t_0};

f = st1 * inv(st00);

vt = (st0 - st1 * inv(st00) * st1')/t;

rt = cov / t;

/*-- check convergence --*/

if ( max(abs((myu - myu0)/(myu0+1e-6))) < 1e-2 & 
max(abs((f - f0)/(f0+1e-6))) < 1e-2 & 
max(abs((vt - vt0)/(vt0+1e-6))) < 1e-2 & 
max(abs((rt - rt0)/(rt0+1e-6))) < 1e-2 & 
abs((logl-logl0)/(logl0+1e-6)) < 1e-3 ) then 
converge = 1;

end;

reset noname;

colnm = {'Iter' '2*2log L' 'F11' 'F12' 'F21' 'F22' 
'MYU11' 'MYU22'};

print itermat[colname=colnm format=8.4];

eval = teigval(f0);

colnm = {'Real' 'Imag' 'MOD'};

eval = eval || sqrt((eval#eval),+);

print eval[colname=colnm];

var = ( vt || j(nz,ny,0) ) // 
( j(ny,nz,0) || rt );

/*-- initial values are changed --*/

z0 = myu' * f';

vz0 = f * sigma * f' + vt;

free itermat;
Chapter 10. Time Series Analysis and Control Examples

```sas
/*-- multistep prediction --*/
call kalcvf(pred,vpred,filt,vfilt,y,15,a,f,b,h,var,z0,vz0);
do i = 1 to 15;
   itermat = itermat // ( i || pred[t+i,] ||
      sqrt(vecdiag(vpred[(t+i-1)*nz+1:(t+i)*nz,])))`);
end;
colnm = {'n-Step' 'Z1_T_n' 'Z2_T_n' 'SE_Z1' 'SE_Z2'};
print itermat[colname=colnm];
finish;
run;
```

Output 10.3.1. Iteration History

<table>
<thead>
<tr>
<th>Iter</th>
<th>-2*log L</th>
<th>F11</th>
<th>F12</th>
<th>F21</th>
<th>F22</th>
<th>MYU11</th>
<th>MYU22</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>-154.010</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>2.0000</td>
<td>-237.962</td>
<td>0.7952</td>
<td>-0.6473</td>
<td>0.3263</td>
<td>0.5143</td>
<td>0.0530</td>
<td>0.0840</td>
</tr>
<tr>
<td>3.0000</td>
<td>-238.083</td>
<td>0.7967</td>
<td>-0.6514</td>
<td>0.3259</td>
<td>0.5142</td>
<td>0.1372</td>
<td>0.0977</td>
</tr>
<tr>
<td>4.0000</td>
<td>-238.126</td>
<td>0.7966</td>
<td>-0.6517</td>
<td>0.3259</td>
<td>0.5139</td>
<td>0.1853</td>
<td>0.1159</td>
</tr>
<tr>
<td>5.0000</td>
<td>-238.143</td>
<td>0.7964</td>
<td>-0.6519</td>
<td>0.3257</td>
<td>0.5138</td>
<td>0.2143</td>
<td>0.1304</td>
</tr>
<tr>
<td>6.0000</td>
<td>-238.151</td>
<td>0.7963</td>
<td>-0.6520</td>
<td>0.3255</td>
<td>0.5136</td>
<td>0.2324</td>
<td>0.1405</td>
</tr>
<tr>
<td>7.0000</td>
<td>-238.153</td>
<td>0.7962</td>
<td>-0.6520</td>
<td>0.3254</td>
<td>0.5135</td>
<td>0.2438</td>
<td>0.1473</td>
</tr>
<tr>
<td>8.0000</td>
<td>-238.155</td>
<td>0.7962</td>
<td>-0.6521</td>
<td>0.3253</td>
<td>0.5135</td>
<td>0.2511</td>
<td>0.1518</td>
</tr>
<tr>
<td>9.0000</td>
<td>-238.155</td>
<td>0.7962</td>
<td>-0.6521</td>
<td>0.3253</td>
<td>0.5134</td>
<td>0.2558</td>
<td>0.1546</td>
</tr>
<tr>
<td>10.0000</td>
<td>-238.155</td>
<td>0.7961</td>
<td>-0.6521</td>
<td>0.3253</td>
<td>0.5134</td>
<td>0.2588</td>
<td>0.1565</td>
</tr>
</tbody>
</table>

Output 10.3.2. Eigenvalues of F Matrix

<table>
<thead>
<tr>
<th>Real</th>
<th>Imag</th>
<th>MOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6547534</td>
<td>0.438317</td>
<td>0.7879237</td>
</tr>
<tr>
<td>0.6547534</td>
<td>-0.438317</td>
<td>0.7879237</td>
</tr>
</tbody>
</table>

Output 10.3.3. Multistep Prediction

<table>
<thead>
<tr>
<th>n-Step</th>
<th>Z1_T_n</th>
<th>Z2_T_n</th>
<th>SE_Z1</th>
<th>SE_Z2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.055792</td>
<td>-0.587049</td>
<td>0.2437666</td>
<td>0.237074</td>
</tr>
<tr>
<td>2</td>
<td>0.3384325</td>
<td>-0.319505</td>
<td>0.3140478</td>
<td>0.290662</td>
</tr>
<tr>
<td>3</td>
<td>0.4778022</td>
<td>-0.053949</td>
<td>0.3669731</td>
<td>0.3104052</td>
</tr>
<tr>
<td>4</td>
<td>0.4155731</td>
<td>0.1276996</td>
<td>0.4021048</td>
<td>0.3218256</td>
</tr>
<tr>
<td>5</td>
<td>0.2475671</td>
<td>0.2007098</td>
<td>0.419699</td>
<td>0.3319293</td>
</tr>
<tr>
<td>6</td>
<td>0.0661993</td>
<td>0.1835492</td>
<td>0.4268943</td>
<td>0.3396153</td>
</tr>
<tr>
<td>7</td>
<td>-0.067001</td>
<td>0.1157541</td>
<td>0.430752</td>
<td>0.3438409</td>
</tr>
<tr>
<td>8</td>
<td>-0.128831</td>
<td>0.0376316</td>
<td>0.4341532</td>
<td>0.3456312</td>
</tr>
<tr>
<td>9</td>
<td>-0.127107</td>
<td>-0.022581</td>
<td>0.4369411</td>
<td>0.3465325</td>
</tr>
<tr>
<td>10</td>
<td>-0.086466</td>
<td>-0.052931</td>
<td>0.4385978</td>
<td>0.3473038</td>
</tr>
<tr>
<td>11</td>
<td>-0.034319</td>
<td>-0.055293</td>
<td>0.4393282</td>
<td>0.3479612</td>
</tr>
<tr>
<td>12</td>
<td>0.0087379</td>
<td>-0.039546</td>
<td>0.4396666</td>
<td>0.3483717</td>
</tr>
<tr>
<td>13</td>
<td>0.0327466</td>
<td>-0.017459</td>
<td>0.439936</td>
<td>0.3485586</td>
</tr>
<tr>
<td>14</td>
<td>0.0374564</td>
<td>0.0016876</td>
<td>0.4401753</td>
<td>0.3486415</td>
</tr>
<tr>
<td>15</td>
<td>0.0287193</td>
<td>0.0130482</td>
<td>0.440335</td>
<td>0.3487034</td>
</tr>
</tbody>
</table>

SAS OnlineDoc™: Version 8
Example 10.4. Diffuse Kalman Filtering

The nonstationary SSM is simulated to analyze the diffuse Kalman filter call KALDFF. The transition equation is generated using the following formula:

\[
\begin{bmatrix}
  z_{1t} \\
  z_{2t}
\end{bmatrix}
= \begin{bmatrix}
  1.5 & -0.5 \\
  1.0 & 0.0
\end{bmatrix}
\begin{bmatrix}
  z_{1t-1} \\
  z_{2t-1}
\end{bmatrix}
+ \begin{bmatrix}
  \eta_{1t} \\
  0
\end{bmatrix}
\]

where \( \eta_{1t} \sim N(0,1) \). The transition equation is nonstationary since the transition matrix \( F \) has one unit root.

```sas
proc iml;
  z_1 = 0; z_2 = 0;
  do i = 1 to 30;
    z = 1.5*z_1 - .5*z_2 + rannor(1234567);
    z_2 = z_1;
    z_1 = z;
    x = z + .8*rannor(1234578);
    if (i>10) then y = y // x;
  end;
```

The KALDFF and KALCVF calls produce one-step prediction, and the result shows that two predictions coincide after the fifth observation (Output 10.4.1).

```sas
t = nrow(y);
h = { 1 0 }; f = { 1.5 - .5, 1 0 }; rt = .64;
vt = diag({1 0}); ny = nrow(h); nz = ncol(h); nb = nz; nd = nz;
a = j(nz,1,0); b = j(ny,1,0);
int = j(ny+nz,nb,0);
coef = f // h;
var = ( vt || j(nz,ny,0) ) // ( j(ny,nz,0) || rt );
intd = j(nz+nb,1,0); coefd = i(nz) // j(nb,nd,0);
at = j(t*nz,nd+1,0);
mt = j(t*nz,nz,0);
qt = j(t*(nd+1),nd+1,0);
n0 = -1;
call kaldff(kaldff_p,dvpred,initial,s2,y,0,int,
           coef,var,intd,coefd,n0,at,mt,qt);
call kalcvf(kalcvf_p,vpred,filt,vfilt,y,0,a,f,b,h,var);
print kaldff_p kaldff_p;
```
Output 10.4.1. Diffuse Kalman Filtering

The likelihood function for the diffuse Kalman filter under the finite initial covariance matrix $\Sigma_\delta$ is written

$$
\lambda(y) = -\frac{1}{2} [y^\# \log(\sigma^2) + \sum_{t=1}^{T} \log(|D_t|)]
$$

where $y^{(#)}$ is the dimension of the matrix $(y_1', \ldots, y_T')'$. The likelihood function for the diffuse Kalman filter under the diffuse initial covariance matrix ($\Sigma_\delta \to \infty$) is computed as $\lambda(y) - \frac{1}{2} \log(|S|)$, where the $S$ matrix is the upper $N_\delta \times N_\delta$ matrix of $Q_t$. See the section “KALDFF Call” on page 583 for more details on matrix notation. Output 10.4.2 on page 290 displays the log likelihood and the diffuse log likelihood.

```sas
/*
   d = 0;
   do i = 1 to t;
      dt = h*mt[(i-1)*nz+1:i*nz,]*h' + rt;
      d = d + log(det(dt));
   end;
   s = qt[(t-1)*(nd+1)+1:t*(nd+1)-1,1:nd];
   log_l = -(t*log(s2) + d)/2;
   dff_logl = log_l - log(det(s))/2;
   print log_l dff_logl;
/*
```

**Output 10.4.2. Diffuse Likelihood Function**

<table>
<thead>
<tr>
<th>Log L</th>
<th>Diffuse Log L</th>
</tr>
</thead>
<tbody>
<tr>
<td>-11.42547</td>
<td>-9.457596</td>
</tr>
</tbody>
</table>
References


