Chapter 10
Time Series Analysis and Control Examples

Chapter Table of Contents

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 **TIM**e **S**eries **A**nalysis and **C**ontrol 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 **TIM**e **S**eries **A**nalysis and **C**ontrol (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 (Ω) consisting of sets of member models (ω), you can describe the uncertainty of Ω using a prior distribution of ω. The member model ω is directly related to model parameters. Let the prior probability density function be $p(\omega)$. When you observe the data $\mathbf{y}$ that is generated from the model Ω, the data distribution is described as $p(Y|\omega)$ given the unknown ω 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 TSBAYSEA subroutine is a typical example of Bayesian modeling. The following subroutines are supported:

TSBAYSEA	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 TSBAYSEA subroutine provides a way of handling Bayesian seasonal adjustment, and it can be an alternative to the SAS/ETS X-11 procedure. The TSBAYSEA 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.

```
proc iml;
   y = { 2.430 2.506 2.767 2.940 3.169 3.450 3.594 3.774 3.695 3.411
         2.718 1.991 2.265 2.446 2.612 3.359 3.429 3.533 3.261 2.612
         2.179 1.653 1.832 2.328 2.737 3.014 3.328 3.404 2.981 2.557
         2.576 2.352 2.556 2.864 3.214 3.435 3.458 3.326 2.835 2.476
         2.373 2.389 2.742 3.210 3.520 3.828 3.628 2.837 2.406 2.675
         2.554 2.894 3.202 3.224 3.352 3.154 2.878 2.476 2.303 2.360
         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
         2.880 3.115 3.540 3.845 3.800 3.579 3.264 2.538 2.582 2.907
         3.142 3.433 3.580 3.490 3.475 3.579 2.829 1.909 1.903 2.033
         2.360 2.601 3.054 3.386 3.553 3.468 3.187 2.723 2.686 2.821
         3.000 3.201 3.424 3.531 };
   call tsunimar(arcoef,ev,nar,aic) data=y opt={-1 1} print=1
        maxlag=20;
```

You can also invoke the TSUNIMAR call as follows:

```
call tsunimar(arcoef,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.

```
ORDER  INNOVATION VARIANCE
  M           V(M)             AIC(M)
  0          0.31607294   -108.26753229
  1          0.11481982   -201.45277331
  2          0.04847420   -280.51201122
  3          0.04828185   -278.88576251
  4          0.04656506   -280.28905616
  5          0.04615922   -279.11190502
  6          0.04511943   -279.25356641
  7          0.04312403   -281.50543541
  8          0.04201118   -281.96304075
  9          0.04128036   -281.61262868
 10          0.03829179   -286.67686828
 11          0.03318558   -298.13013264
 12          0.03255171   -297.94298716
 13          0.03247784   -296.15655602
 14          0.03237083   -294.46677874
 15          0.03234790   -292.53337704
 16          0.03187416   -291.92021487
 17          0.03183282   -290.04220196
 18          0.03126946   -289.72064823
 19          0.03087893   -288.90203735
 20          0.02998019   -289.67854830
```

Figure 10.1. Minimum AIC Table - I

```
                          AIC(M)-AICMIN (truncated at 40.0)
                      0        10        20        30        40
 M   AIC(M)-AICMIN    +---------+---------+---------+---------+
 0    189.862600      |                                       .
 1     96.677359      |                                       .
 2     17.618121      |                 *                     |
 3     19.244370      |                  *                    |
 4     17.841076      |                 *                     |
 5     19.018228      |                  *                    |
 6     18.876566      |                  *                    |
 7     16.624697      |                *                      |
 8     16.167092      |               *                       |
 9     16.517504      |                *                      |
10     11.453264      |          *                            |
11             0      *                                       |
12      0.187145      *                                       |
13      1.973577      | *                                     |
14      3.663354      |   *                                   |
15      5.596756      |     *                                 |
16      6.209918      |     *                                 |
17      8.087931      |       *                               |
18      8.409484      |       *                               |
19      9.228095      |        *                              |
20      8.451584      |       *                               |
                      +---------+---------+---------+---------+
```

Figure 10.2. Minimum AIC Table - II

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.

```
..........................M  A  I  C  E.........................
.                                                              .
.                                                              .
.                                                              .
.     M     AR Coefficients: AR(M)                             .
.                                                              .
.     1     1.181322                                           .
.     2    -0.551571                                           .
.     3     0.231372                                           .
.     4    -0.178040                                           .
.     5     0.019874                                           .
.     6    -0.062573                                           .
.     7     0.028569                                           .
.     8    -0.050710                                           .
.     9     0.199896                                           .
.    10     0.161819                                           .
.    11    -0.339086                                           .
.                                                              .
.                                                              .
.      AIC =   -298.1301326                                    .
.      Innovation Variance =    0.033186                       .
.                                                              .
.                                                              .
.            INPUT DATA   START =    21  FINISH =   114        .
................................................................
```

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);
```

```
..........................M  A  I  C  E.........................
.                                                              .
.                                                              .
.                                                              .
.     M     AR Coefficients: AR(M)                             .
.                                                              .
.     1     1.149416                                           .
.     2    -0.533719                                           .
.     3     0.276312                                           .
.     4    -0.326420                                           .
.     5     0.169336                                           .
.     6    -0.164108                                           .
.     7     0.073123                                           .
.     8    -0.030428                                           .
.     9     0.151227                                           .
.    10     0.192808                                           .
.    11    -0.340200                                           .
.                                                              .
.                                                              .
.      AIC =   -318.7984105                                    .
.      Innovation Variance =    0.036563                       .
.                                                              .
.                                                              .
.            INPUT DATA   START =    12  FINISH =   114        .
................................................................
```

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.

```
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.

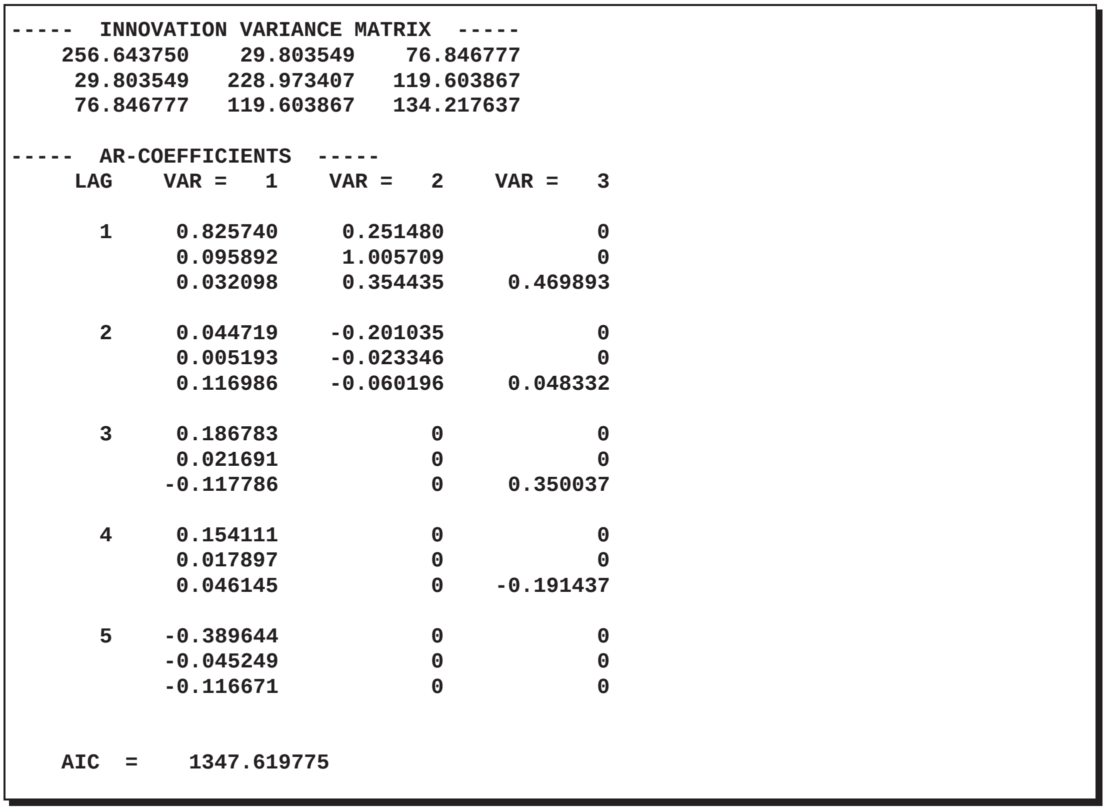
```
ORDER  INNOVATION VARIANCE
  M         LOG(|V(M)|)        AIC(M)
  0         25.98001095   2136.36089828
  1         15.70406486   1311.73331883
  2         15.48896746   1312.09533158
  3         15.18567834   1305.22562428
  4         14.96865183   1305.42944974
  5         14.74838535   1305.36759889
  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


                          AIC(M)-AICMIN (truncated at 40.0)
                       0        10        20        30        40
  M   AIC(M)-AICMIN    +---------+---------+---------+---------+
  0    831.135274      |                                       .
  1      6.507695      |      *                                |
  2      6.869707      |      *                                |
  3             0      *                                       |
  4      0.203825      *                                       |
  5      0.141975      *                                       |
  6      6.195240      |     *                                 |
  7     19.859523      |                   *                   |
  8     24.423369      |                       *               |
  9     24.209069      |                       *               |
 10     16.784212      |                *                      |
                       +---------+---------+---------+---------+
```

Figure 10.5. VAR Model Selection

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.

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



```
-----  INNOVATION VARIANCE MATRIX  -----
    256.643750    29.803549    76.846777
     29.803549   228.973407   119.603867
     76.846777   119.603867   134.217637

-----  AR-COEFFICIENTS  -----
     LAG    VAR =   1    VAR =   2    VAR =   3

       1     0.825740     0.251480            0
             0.095892     1.005709            0
             0.032098     0.354435     0.469893

       2     0.044719    -0.201035            0
             0.005193    -0.023346            0
             0.116986    -0.060196     0.048332

       3     0.186783            0            0
             0.021691            0            0
            -0.117786            0     0.350037

       4     0.154111            0            0
             0.017897            0            0
             0.046145            0    -0.191437

       5    -0.389644            0            0
            -0.045249            0            0
            -0.116671            0            0


    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:

$$(T - lag)\log(|\hat{\Sigma}|) + 2(p \times n^2 + n \times \textit{intercept})$$

where p is the order of the n-variate VAR process, and *intercept*=1 if the intercept is specified; otherwise, *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.

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

```
          AIC       NAR

     1403.0762     3


              ARCOEF

  4.8245814  5.3559216  17.066894
  0.8855926  0.3401741  -0.014398
  0.1684523  1.0502619   0.107064
  0.0891034  0.4591573   0.4473672
 -0.059195  -0.298777    0.1629818
  0.1128625 -0.044039   -0.088186
  0.1684932 -0.025847   -0.025671
  0.0637227 -0.196504    0.0695746
 -0.226559   0.0532467  -0.099808
 -0.303697  -0.139022    0.2576405
```

Figure 10.7. AIC from Instantaneous Response Model

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

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

```
     AIC       NAR

  1397.0762     3


               ARCOEF

  4.8245814  5.3559216  17.066894
  0.8855926  0.3401741  -0.014398
  0.1684523  1.0502619   0.107064
  0.0891034  0.4591573   0.4473672
 -0.059195  -0.298777    0.1629818
  0.1128625 -0.044039   -0.088186
  0.1684932 -0.025847   -0.025671
  0.0637227 -0.196504    0.0695746
 -0.226559   0.0532467  -0.099808
 -0.303697  -0.139022    0.2576405
```

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;
```

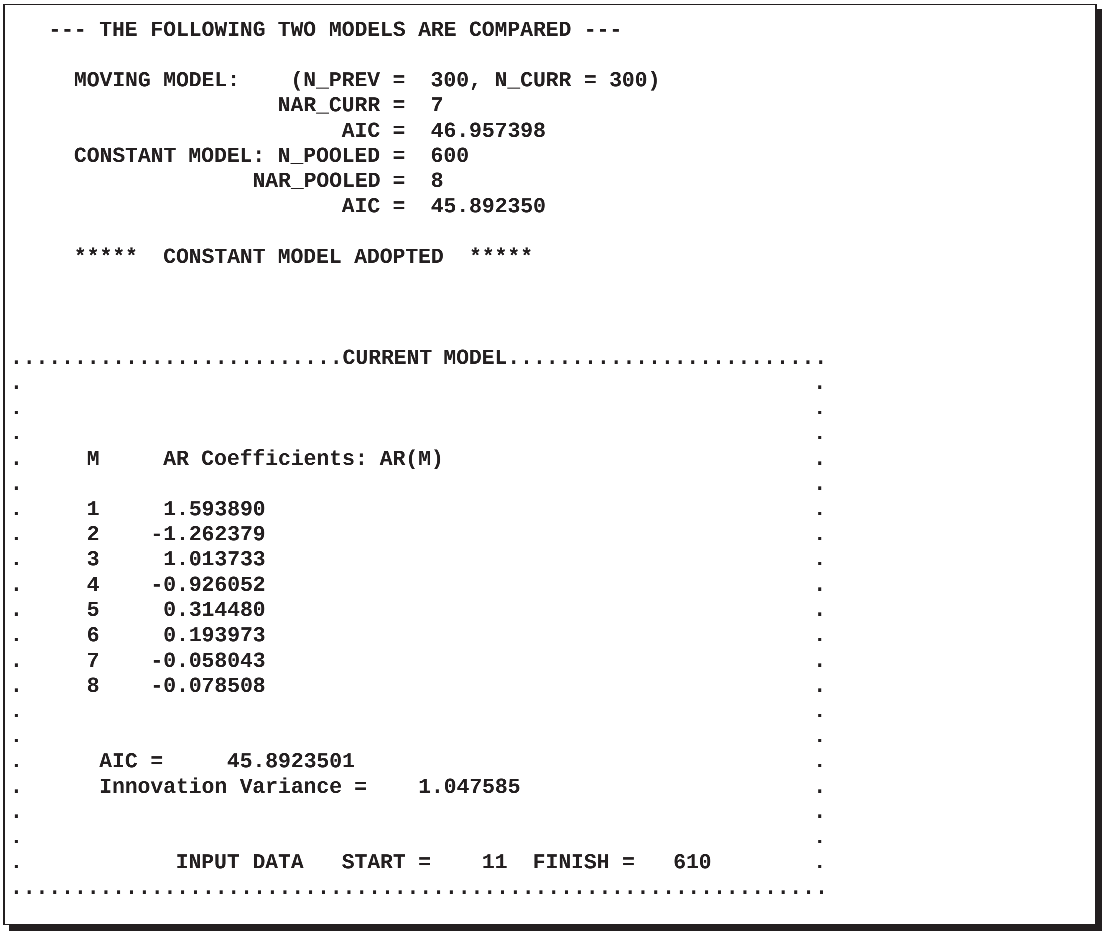
Estimation results are displayed with the graphs of power spectrum $(\log 10(f_{YY}(g)))$, where $f_{YY}(g)$ is a rational spectral density function. See the “Spectral Analysis” section on page 261. As the first block and the second block do not have any sizable difference, the pooled model (AIC=45.892) 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.........................
.                                                              .
.                                                              .
.                                                              .
.     M     AR 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

```
        POWER SPECTRAL DENSITY
20.00+
     |
     |
     |
     |
     |  XXXX
     XXX    XX    XXX
     |        XXXX
     |               X
     |
10.00+
     |                X
     |
     |                 X
     |
     |                  X               XX
     |                                    X
     |                   X             X
     |
     |                    X           X    X
    0+                     X
     |                      X        X      X
     |                       XX    XX
     |                         XXXX          X
     |
     |                                        X
     |                                         X
     |
     |                                          X
     |                                           X
-10.0+                                            X
     |                                             XX
     |                                               XX
     |                                                 XX
     |                                                   XXX
     |                                                      XXXXXX
     |
     |
     |
     |
-20.0+-----------+-----------+-----------+-----------+-----------+
    0.0         0.1         0.2         0.3         0.4         0.5

                               FREQUENCY
```

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        .
................................................................
```

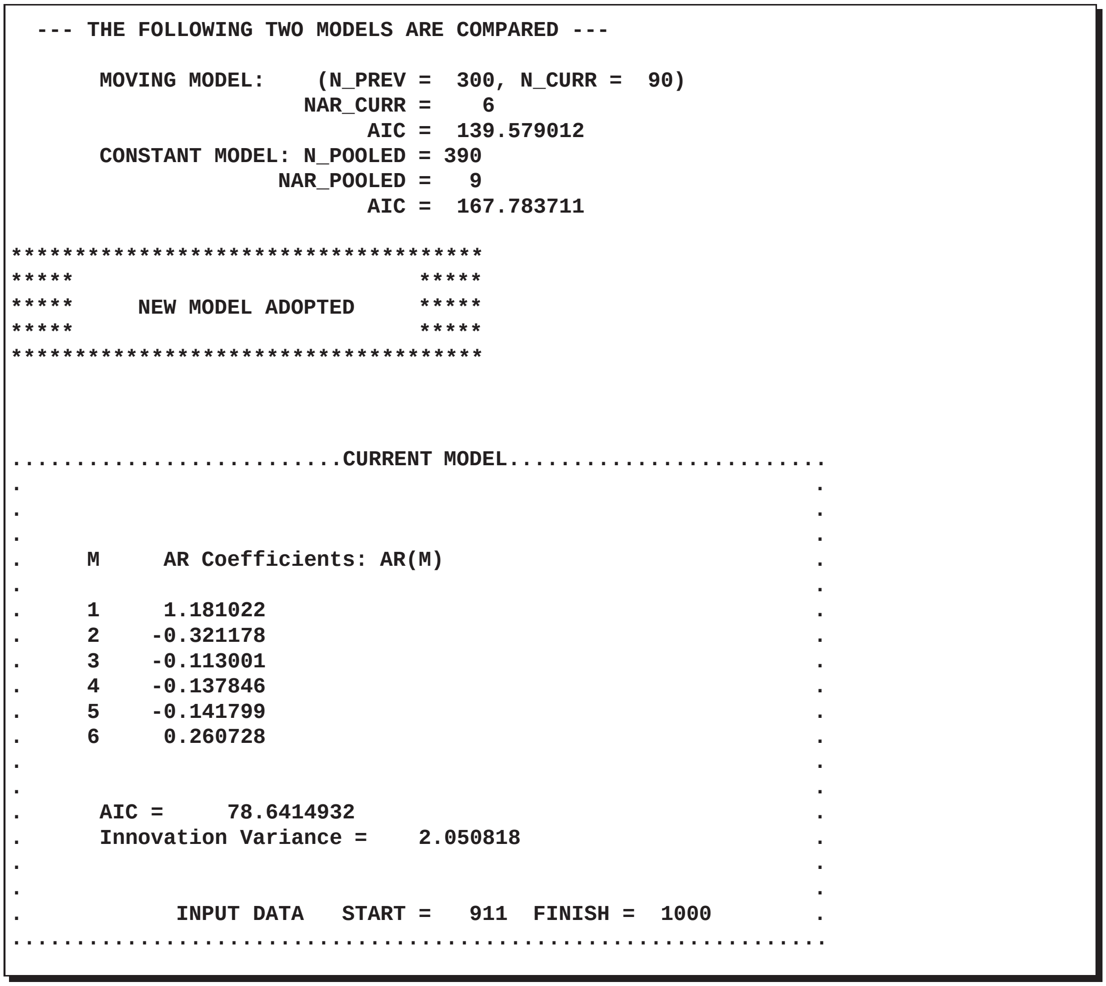
Figure 10.12. Locally Stationary Model for Third Block

```
 POWER SPECTRAL DENSITY
20.00+             X
     |            X
     |           X
     |              X
     |        XXX
     |   XXXXX
     | XX
    XX              X
     |
     |
10.00+                X
     |
     |
     |                 X
     |
     |                  X
     |                                   X
     |                   X              X
     |                                    X
     |                    X            X
    0+                     X          X    X
     |                      X
     |                       X      XX      X
     |                        XXXXXX
     |                                       X
     |
     |                                        X
     |
     |                                         X
     |                                          X
-10.0+                                           X
     |                                            XX
     |                                              XX       XXXXX
     |                                                XXXXXXX
     |
     |
     |
     |
     |
     |
-20.0+-----------+-----------+-----------+-----------+-----------+
    0.0         0.1         0.2         0.3         0.4         0.5

                               FREQUENCY
```

Figure 10.13. Power Spectrum 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 COMPARED ---

       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         .
.................................................................
```

Figure 10.14. Locally Stationary Model for Last Block

```
   POWER SPECTRAL DENSITY
30.00+
     |
     |
     |
     |
     |           X
     |
     |            X
     |
     |
20.00+          X
     |
     |
     |         X   X
     |
     |        X
     XXX     X
     |  XXXXX       X
     |
     |
10.00+               X
     |
     |                X
     |
     |                 X
     |
     |                  X
     |                   X
     |                    X
     |                     XX
    0+                       XX      XXXXXX
     |                         XXXXXX      XX
     |                                       XX
     |                                         XX               XX
     |                                           XX         XXXX
     |                                             XXXXXXXXX
     |
     |
     |
     |
-10.0+-----------+-----------+-----------+-----------+-----------+
    0.0         0.1         0.2         0.3         0.4         0.5

                               FREQUENCY
```

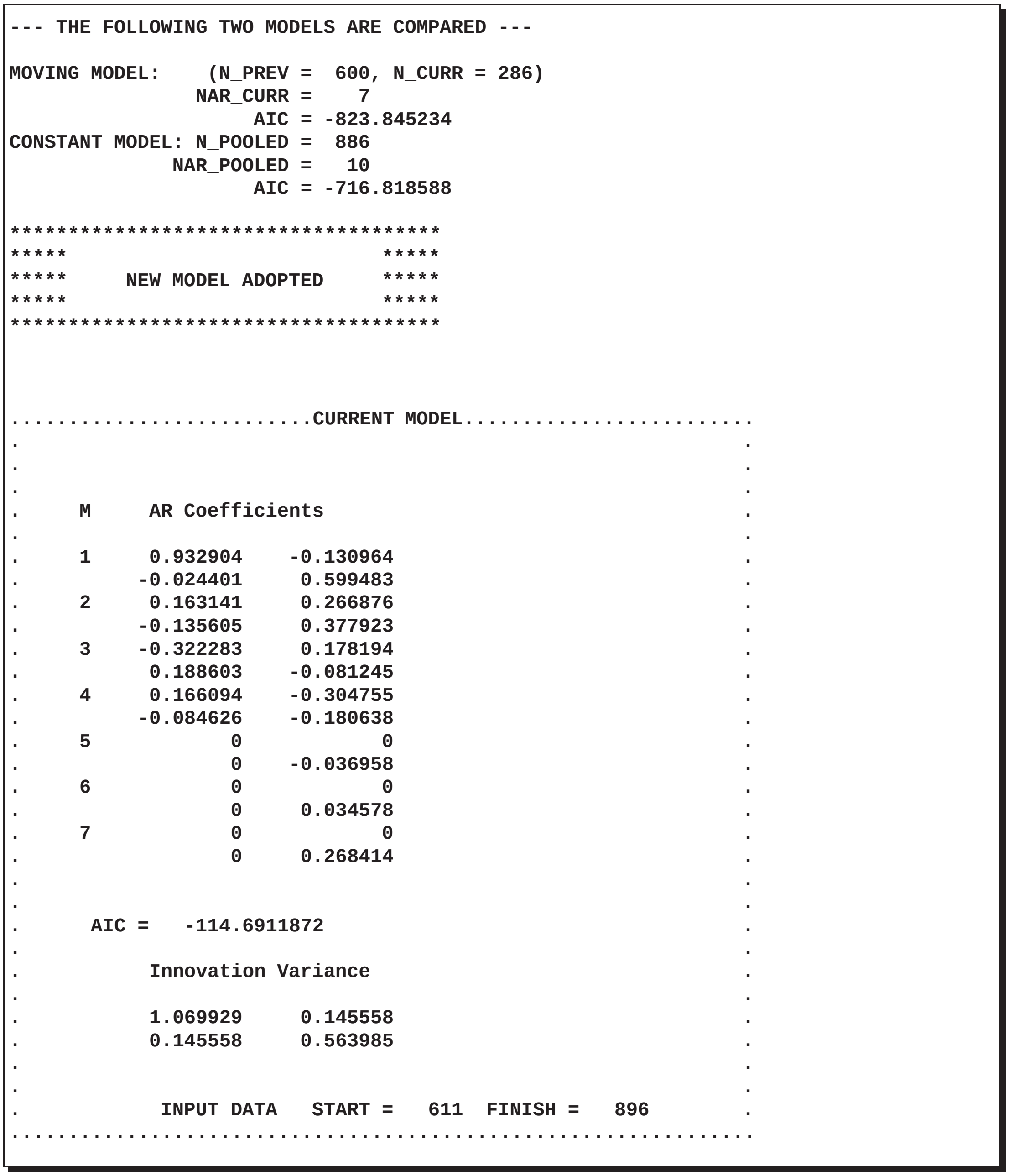
Figure 10.15. Power Spectrum for Last Block

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.

```
proc iml;
   rudder = {. . . data lines omitted . . .};
   yawing = {. . . data lines omitted . . .};

   y = rudder‘ || yawing‘;
   c = {0.01795 0.02419};
  /*-- calibration of data --*/
```

```
y = y # (c @ j(n,1,1));
mdel = -1;
lspan = 300; /* local span of data */
maice = 1;
call tsmlomar(arcoef,ev,nar,aic,first,last) data=y maxlag=10
      opt = (mdel || lspan || maice) print=1;
```



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

MOVING MODEL:    (N_PREV =  600, N_CURR = 286)
                NAR_CURR =    7
                     AIC = -823.845234
CONSTANT MODEL: N_POOLED =  886
              NAR_POOLED =   10
                     AIC = -716.818588

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



..........................CURRENT MODEL.........................
.                                                              .
.                                                              .
.                                                              .
.     M     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                              .
.                                                              .
.                                                              .
.      AIC =   -114.6911872                                    .
.                                                              .
.           Innovation Variance                                .
.                                                              .
.           1.069929     0.145558                              .
.           0.145558     0.563985                              .
.                                                              .
.                                                              .
.            INPUT DATA   START =   611  FINISH =   896        .
................................................................
```

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.

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

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

$$\begin{aligned} y_t &= \mathbf{H}\mathbf{z}_t + \epsilon_t \\ \mathbf{z}_t &= \mathbf{F}\mathbf{z}_{t-1} + \mathbf{w}_t \end{aligned}$$

where

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \\ \mathbf{F} &= \begin{bmatrix} 2 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & \alpha_1 & \alpha_2 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ \mathbf{z}_t &= (T_t, T_{t-1}, u_t, u_{t-1})' \\ \mathbf{w}_t &= (w_{1t}, 0, w_{2t}, 0)' \sim \left(0, \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_2^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) \end{aligned}$$

The parameters of this state space model are σ_1^2, σ_2^2, α_1, and α_2.

```
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
         263.7 297.8 337.1 361.3 355.2 312.6 309.9 323.7
         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 };
   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


I   TAU2(I)       AR(I)      PARCOR(I)
1     2.915075     1.397374     0.876163
2   113.957712    -0.594879    -0.594879
```

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.

```
    Y            CMP

487.7  514.01142  -26.94343
497.2  532.62744  -32.48673
529.8  552.02402  -24.46593
551    571.90122  -20.15113
581.1  592.31944  -10.58647
617.8  613.21855    5.2504354
658.1  634.43665   20.799209
675.2  655.70431   22.161597
706.6  677.2125    27.927978
724.7  698.72364   25.957961
       720.23478   19.659202
       741.74592   12.029403
       763.25707    5.1147232
       784.76821   -0.00886
       806.27935   -3.055023
```

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.

$$\begin{aligned} T_t &= 2T_{t-1} - T_{t-2} + w_{1t} \\ S_t &= -S_{t-1} - \cdots - S_{t-11} + w_{2t} \end{aligned}$$

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.

```
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
```

```
    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;
```

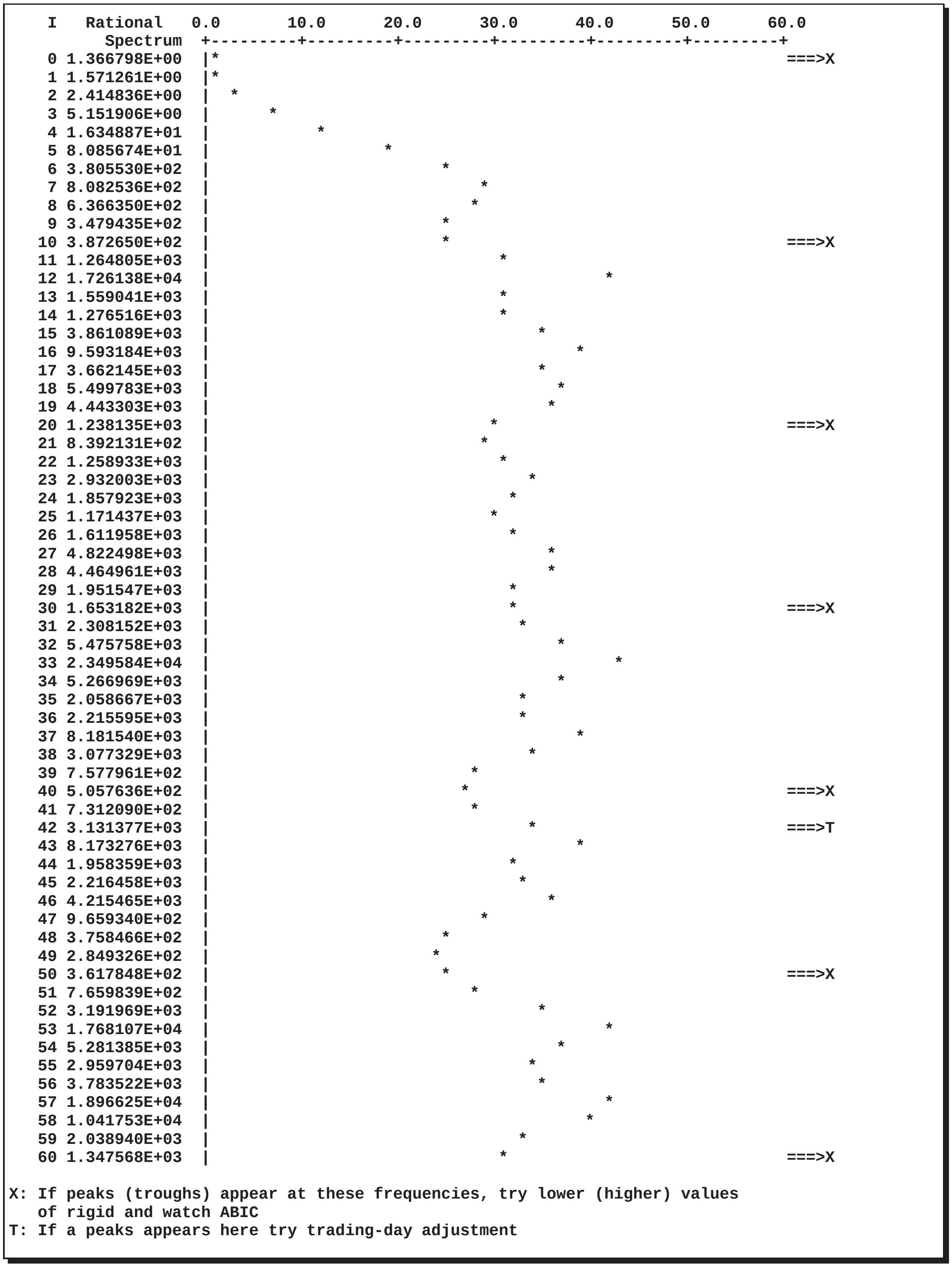
```
                Seasonal Component
 576.866752   612.796066   324.020037  -198.760111
-572.556158   493.248873   218.901469  -126.976886
-223.927593  -440.622170  -345.477541  -339.527540
 567.417780   649.108143   315.457702  -195.764740
-567.242588   503.917031   226.829019  -142.216380
-209.010499  -511.275202  -344.187789  -365.761124
 647.626707   686.576003   324.601881  -242.421270
-582.439797   516.512576   248.795247  -160.227108
-212.583209  -538.237178  -364.306967  -416.965872
 749.318446   705.520212   361.245687  -273.971547
-617.748290   506.336574   239.146930  -132.685481
-254.706508  -510.461942  -348.035057  -391.992877
 711.125340   748.595903   367.983922  -290.532690
-700.824658   519.764643   242.638512   -73.786428
-288.809493  -509.321443  -302.485088  -397.322723
 650.134120   800.460271   395.841362  -340.552541
-719.314201   553.049123   201.955997   -54.527951
-295.332122  -487.701411  -266.216231  -440.347213
 650.770701   800.937334   396.198661  -340.285229
-719.114602   553.197644   202.065816   -54.447682
-295.274714  -487.662081  -266.191701  -440.335439
***  Last   12 Values Are Forecasted  ***
```

Figure 10.19. Seasonal Component Estimates and Forecasts

```
 Adjusted = Data - Seasonal - Trading_Day_Comp - OCF

  4870.133248  4799.203934  4890.979963  4895.760111
  4916.556158  4932.751127  4954.098531  4983.976886
  4881.927593  4910.622170  4613.477541  4455.527540
  4107.582220  4195.891857  4196.542298  4369.764740
  4366.242588  4343.082969  4323.170981  4350.216380
  4374.010499  4274.275202  4400.187789  4423.761124
  4360.373293  4453.423997  4430.398119  4543.421270
  4726.439797  4863.487424  5011.204753  5045.227108
  5414.583209  5582.237178  6049.306967  6522.965872
  7430.681554  7603.479788  7997.754313  8093.971547
  8240.748290  8062.663426  7969.853070  7828.685481
  7776.706508  7754.461942  7579.035057  7586.992877
  7462.874660  7284.404097  7157.016078  7180.532690
  7004.824658  7135.235357  7334.361488  7395.786428
  7314.809493  7342.321443  7397.485088  7419.322723
  7197.865880  7308.539729  7160.158638  6908.552541
  6870.314201  6899.950877  6739.044003  6811.527951
  6732.332122  6708.701411  6612.216231  6320.347213
```

Figure 10.20. Seasonally Adjusted Series



```
    I   Rational   0.0       10.0      20.0      30.0      40.0      50.0      60.0
          Spectrum  +---------+---------+---------+---------+---------+---------+
    0 1.366798E+00  |*                                                           ===>X
    1 1.571261E+00  |*
    2 2.414836E+00  |  *
    3 5.151906E+00  |      *
    4 1.634887E+01  |           *
    5 8.085674E+01  |                  *
    6 3.805530E+02  |                        *
    7 8.082536E+02  |                            *
    8 6.366350E+02  |                           *
    9 3.479435E+02  |                        *
   10 3.872650E+02  |                        *                                   ===>X
   11 1.264805E+03  |                              *
   12 1.726138E+04  |                                         *
   13 1.559041E+03  |                              *
   14 1.276516E+03  |                              *
   15 3.861089E+03  |                                  *
   16 9.593184E+03  |                                      *
   17 3.662145E+03  |                                  *
   18 5.499783E+03  |                                    *
   19 4.443303E+03  |                                   *
   20 1.238135E+03  |                             *                              ===>X
   21 8.392131E+02  |                            *
   22 1.258933E+03  |                              *
   23 2.932003E+03  |                                 *
   24 1.857923E+03  |                               *
   25 1.171437E+03  |                             *
   26 1.611958E+03  |                               *
   27 4.822498E+03  |                                   *
   28 4.464961E+03  |                                   *
   29 1.951547E+03  |                               *
   30 1.653182E+03  |                               *                            ===>X
   31 2.308152E+03  |                                *
   32 5.475758E+03  |                                    *
   33 2.349584E+04  |                                          *
   34 5.266969E+03  |                                    *
   35 2.058667E+03  |                                *
   36 2.215595E+03  |                                *
   37 8.181540E+03  |                                      *
   38 3.077329E+03  |                                 *
   39 7.577961E+02  |                           *
   40 5.057636E+02  |                          *                                 ===>X
   41 7.312090E+02  |                           *
   42 3.131377E+03  |                                 *                          ===>T
   43 8.173276E+03  |                                      *
   44 1.958359E+03  |                               *
   45 2.216458E+03  |                                *
   46 4.215465E+03  |                                   *
   47 9.659340E+02  |                            *
   48 3.758466E+02  |                        *
   49 2.849326E+02  |                       *
   50 3.617848E+02  |                        *                                   ===>X
   51 7.659839E+02  |                           *
   52 3.191969E+03  |                                  *
   53 1.768107E+04  |                                         *
   54 5.281385E+03  |                                    *
   55 2.959704E+03  |                                 *
   56 3.783522E+03  |                                  *
   57 1.896625E+04  |                                         *
   58 1.041753E+04  |                                       *
   59 2.038940E+03  |                                *
   60 1.347568E+03  |                              *                             ===>X

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
```

Figure 10.21. Spectrum of Irregular Component

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

$$\mathbf{y}_t + \sum_{i=1}^{p} \mathbf{A}_i \mathbf{y}_{t-i} = \epsilon_t + \sum_{i=1}^{q} \mathbf{M}_i \epsilon_{t-i}$$

Then the COEF matrix is constructed by stacking matrices $\mathbf{A}_1, \ldots, \mathbf{A}_p, \mathbf{M}_1, \ldots, \mathbf{M}_q$.

```
proc iml;
   c = { 264 235 239 239 275 277 274 334 334 306
         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 = { 690 690 688 690 694 702 702 702 700 702
         702 694 708 702 702 708 700 700 702 694
         698 694 700 702 700 702 708 708 710 704
         704 700 700 694 702 694 710 710 710 708 };
   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    .079386   -5.4382,
          -.39983    .94448    .027938   -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 };
```

```
nar = 2; nma = 1;
call tspred(forecast,impulse,mse,y,coef,nar,nma,ev,
            5,nrow(y),-1);
```

```
   OBSERVED       PREDICTED
 Y1    Y2        P1       P2

 264   690   269.950  700.750
 235   690   256.764  691.925
 239   688   239.996  693.467
 239   690   242.320  690.951
 275   694   247.169  693.214
 277   702   279.024  696.157
 274   702   284.041  700.449
 334   702   286.890  701.580
 334   700   321.798  699.851
 306   702   330.355  702.383
 308   702   297.239  700.421
 309   694   302.651  701.928
 295   708   294.570  696.261
 271   702   283.254  703.936
 277   702   269.600  703.110
 221   708   270.349  701.557
 223   700   231.523  705.438
 227   700   233.856  701.785
 215   702   234.883  700.185
 223   694   229.156  701.837
 241   698   235.054  697.060
 250   694   249.288  698.181
 270   700   257.644  696.665
 303   702   272.549  699.281
 311   700   301.947  701.667
 307   702   306.422  700.708
 322   708   304.120  701.204
 335   708   311.590  704.654
 335   710   320.570  706.389
 334   704   315.127  706.439
 309   704   311.083  703.735
 262   700   288.159  702.801
 228   700   251.352  700.805
 191   694   226.749  700.247
 188   702   199.775  696.570
 215   694   202.305  700.242
 215   710   222.951  696.451
 249   710   226.553  704.483
 291   710   259.927  707.610
 296   708   291.446  707.861
             293.899  707.430
             293.477  706.933
             292.564  706.190
             290.313  705.384
             286.559  704.618
```

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

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

you can specify the following polynomial equation:

$$z^p - \sum_{i=1}^{p} \alpha_i z^{p-i} = 0$$

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 θ_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.

```
proc iml;
   y = { 2.430 2.506 2.767 2.940 3.169 3.450 3.594 3.774 3.695 3.411
         2.718 1.991 2.265 2.446 2.612 3.359 3.429 3.533 3.261 2.612
         2.179 1.653 1.832 2.328 2.737 3.014 3.328 3.404 2.981 2.557
         2.576 2.352 2.556 2.864 3.214 3.435 3.458 3.326 2.835 2.476
         2.373 2.389 2.742 3.210 3.520 3.828 3.628 2.837 2.406 2.675
         2.554 2.894 3.202 3.224 3.352 3.154 2.878 2.476 2.303 2.360
         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
         2.880 3.115 3.540 3.845 3.800 3.579 3.264 2.538 2.582 2.907
         3.142 3.433 3.580 3.490 3.475 3.579 2.829 1.909 1.903 2.033
         2.360 2.601 3.054 3.386 3.553 3.468 3.187 2.723 2.686 2.821
         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).

```
..........................M  A  I  C  E.........................
.                                                               .
.                                                               .
.                                                               .
.     M     AR Coefficients: AR(M)                              .
.                                                               .
.     1     1.300307                                            .
.     2    -0.723280                                            .
.     3     0.242193                                            .
.     4    -0.378757                                            .
.     5     0.137727                                            .
.                                                               .
.                                                               .
.      AIC =   -318.6137704                                     .
.      Innovation Variance =    0.049055                        .
.                                                               .
.                                                               .
.            INPUT DATA   START =     6  FINISH =   114         .
.................................................................
```

Figure 10.23. Minimum AIC AR Estimation

```
I     Real    Imaginary    MOD(Z)   ATAN(I/R)    Degree

1   -0.29755    0.55991    0.6341    2.0593     117.9869
2   -0.29755   -0.55991    0.6341   -2.0593    -117.9869
3    0.40529          0    0.4053         0            0
4    0.74505    0.53866    0.9194    0.6260      35.8660
5    0.74505   -0.53866    0.9194   -0.6260     -35.8660
```

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(ar_cx) matin=root nar=nar qcoef=1
                         nma=0 print=1;
```

```
I       AR(real)       AR(imag)

1        1.30031              0
2       -0.72328    1.11022E-16
3        0.24219    1.66533E-16
4       -0.37876    -2.7756E-17
5        0.13773              0
```

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) φ 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

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

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

$$B(\varphi, f) = \mathrm{E}_Z \log f(Z) - \mathrm{E}_Z \log \varphi(Z)$$

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 $\mathrm{E}_Z \log f(Z)$, the closer the function f is to the true PDF φ. Given the data $\mathbf{y} = (y_1, \ldots, y_T)'$ that has the same distribution as the random variable Z, let the likelihood function of the parameter vector θ 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 $\mathrm{E}_Z \log f(Z)$ by using the Bayesian predictive likelihood. The AIC is the bias-corrected estimate of $-2T\mathrm{E}_Z \log f(Z|\hat{\theta})$, where $\hat{\theta}$ is the maximum likelihood estimate.

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

Let $\theta = (\theta_1, \ldots, \theta_K)'$ be a $K \times 1$ parameter vector that is contained in the parameter space Θ_K. Given the data $\mathbf{y}$, 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 θ^0 is contained in Θ_K. Let $\hat{\theta}_K$ be a maximum likelihood estimate. The maximum of the log likelihood function is denoted as $\ell(\hat{\theta}_K) = \max_{\theta \in \Theta_K} \ell(\theta)$. The expected log likelihood function is defined by

$$\ell^*(\theta) = T\mathrm{E}_Z \log f(Z|\theta)$$

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

$$\ell^*(\theta) \overset{A}{=} \ell^*(\theta^0) + T(\theta - \theta^0)'\mathrm{E}_Z \frac{\partial \log f(Z|\theta^0)}{\partial \theta} - \frac{T}{2}(\theta - \theta^0)' I(\theta^0)(\theta - \theta^0)$$

where $I(\theta^0)$ is the information matrix and $\overset{A}{=}$ stands for asymptotic equality. Note that $\frac{\partial \log f(z|\theta^0)}{\partial \theta} = 0$ since $\log f(z|\theta)$ is maximized at θ^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

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

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, $\ell^*(K) = \mathrm{E}_Y \ell^*(\hat{\theta}_K)$, 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

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

Since $\ell(\hat{\theta}_K)$ is the maximum log likelihood function, $\left. \frac{\partial \ell(\theta)}{\partial \theta} \right|_{\hat{\theta}_K} = 0$. Note that $\mathrm{plim} \left[-\frac{1}{T} \left. \frac{\partial^2 \ell(\theta)}{\partial \theta \partial \theta'} \right|_{\hat{\theta}_K} \right] = I(\theta^0)$ if the maximum likelihood estimator $\hat{\theta}_K$ is a consistent estimator of θ. Replacing θ with the true parameter θ^0 and taking expectations with respect to the random variable Y,

$$\mathrm{E}_Y \ell(\theta^0) \overset{A}{=} \mathrm{E}_Y \ell(\hat{\theta}_K) - \frac{K}{2}$$

Consider the following relationship:

$$\begin{aligned}\ell^*(\theta^0) &= T\mathrm{E}_Z \log f(Z|\theta^0) \\ &= \mathrm{E}_Y \sum_{t=1}^{T} \log f(Y_t|\theta^0) \\ &= \mathrm{E}_Y \ell(\theta^0)\end{aligned}$$

From the previous derivation,

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

Therefore,

$$\ell^*(K) \overset{A}{=} \mathrm{E}_Y \ell(\hat{\theta}_K) - K$$

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

$$\ell^*(K) \overset{A}{=} \ell(\hat{\theta}_K) - K$$

Consequently, the AIC is defined as an asymptotically unbiased estimator of $-2(\textit{mean expected log likelihood})$

$$\mathrm{AIC}(K) = -2\ell(\hat{\theta}_K) + 2K$$

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 ϵ_t is an iid random variable with zero mean and positive variance σ^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 λ 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 ∇^k denotes the kth order difference operator. The value of λ 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 kth order stochastically perturbed difference equation.

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

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.

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

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

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

The values of hyperparameters, τ_1^2 and τ_2^2, refer to a measure of uncertainty of prior information. For example, the large value of τ_1^2 implies a relatively smooth trend component. The ratio $\frac{\tau_i^2}{\sigma^2}$ $(i = 1, 2)$ 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 recursive computation are discussed in the section, "State Space and Kalman Filter Method".

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_αth 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 α_i and β_j.

The seasonally adjusted series is obtained by subtracting the estimated seasonal component from the original series. Usually, the error term ϵ_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 $N_t = T_t + \epsilon_t$ 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 λ_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

$$\begin{aligned} \phi_j(B) &= 1 - \phi_1 B - \cdots - \phi_j B^{p_j} \\ \theta_i(B) &= 1 - \theta_1 B - \cdots - \theta_i B^{q_i} \end{aligned}$$

and $\tilde{y}_t = y_t - \mathrm{E}(Y_t)$ 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$(p,d,q)(P,D,Q)_s$ 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 ϕ, 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

$$T_{i-1} = \sum_{j=1}^{i-1} t_j < t \leq T_i = \sum_{j=1}^{i} t_j, \qquad \text{for } i = 1, \ldots, k$$

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

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

Given α_j^i, $j = 0, \ldots, p_i$, the maximum of the log likelihood function is attained at

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

The concentrated log likelihood function is given by

$$\ell^* = -\frac{T}{2}[1 + \log(2\pi)] - \frac{1}{2} \sum_{i=1}^{k} t_i \log(\hat{\sigma}_i^2)$$

Therefore, the maximum likelihood estimates, $\hat{\alpha}_j^i$ and $\hat{\sigma}_i^2$, are obtained by minimizing the following local SSE:

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

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

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

where *intercept* = 1 if the intercept term (α_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

$$\begin{aligned} \mathrm{AIC}_1 &= T \log(\sigma_1^2) + 2(p_1 + \textit{intercept} + 1) \\ \mathrm{AIC}_2 &= T_1 \log(\sigma_2^2) + 2(p_2 + \textit{intercept} + 1) \end{aligned}$$

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

$$\mathrm{AIC}_J = \mathrm{AIC}_1 + \mathrm{AIC}_2$$

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

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

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

Decision

- If $\mathrm{AIC}_J < \mathrm{AIC}_A$, switch to the new model, since there is a change in the structure of the time series.
- If $\mathrm{AIC}_J \geq \mathrm{AIC}_A$, 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 α_{it} are assumed to change gradually with time. The following simple stochastic difference equation constraint is imposed on each coefficient:

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

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:

$$\sum_{t=1}^{T} \left(y_t - \sum_{i=1}^{m} \alpha_{it} y_{t-i} \right)^2 + \tau^2 \sum_{t=1}^{T} \sum_{i=1}^{m} \left(\nabla^k \alpha_{it} \right)^2 + \lambda^2 \sum_{t=1}^{T} \sum_{i=1}^{m} i^2 \alpha_{it}^2 + \nu^2 \sum_{t=1}^{T} \sum_{i=1}^{m} \alpha_{it}^2$$

where τ^2, λ^2, and ν^2 are hyperparameters of the prior distribution.

Using a state space representation, the model is

$$\begin{aligned}
\mathbf{x}_t &= \mathbf{F}\mathbf{x}_{t-1} + \mathbf{G}\mathbf{w}_t \\
y_t &= \mathbf{H}_t\mathbf{x}_t + \epsilon_t
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{x}_t &= (\alpha_{1t}, \ldots, \alpha_{mt}, \ldots, \alpha_{1,t-k+1}, \ldots, \alpha_{m,t-k+1})' \\
\mathbf{H}_t &= (y_{t-1}, \ldots, y_{t-m}, \ldots, 0, \ldots, 0) \\
\mathbf{w}_t &= (w_{1t}, \ldots, w_{mt})' \\
k &= 1 : \mathbf{F} = \mathbf{I}_m \quad \mathbf{G} = \mathbf{I}_m \\
k &= 2 : \mathbf{F} = \begin{bmatrix} 2\mathbf{I}_m & -\mathbf{I}_m \\ \mathbf{I}_m & 0 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \mathbf{I}_m \\ 0 \end{bmatrix} \\
k &= 3 : \mathbf{F} = \begin{bmatrix} 3\mathbf{I}_m & -3\mathbf{I}_m & \mathbf{I}_m \\ \mathbf{I}_m & 0 & 0 \\ 0 & \mathbf{I}_m & 0 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \mathbf{I}_m \\ 0 \\ 0 \end{bmatrix} \\
\begin{bmatrix} \mathbf{w}_t \\ \epsilon_t \end{bmatrix} &\sim N\left(\mathbf{0}, \begin{bmatrix} \tau^2\mathbf{I} & 0 \\ 0 & \sigma^2 \end{bmatrix}\right)
\end{aligned}$$

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

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

Using the $\mathbf{LDL}'$ factorization method, the error covariance is decomposed as

$$\Sigma = \mathbf{LDL}'$$

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 $\mathbf{D}$. 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.

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

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

$$\Psi_m = \frac{\partial \mathbf{y}_{t+m}}{\partial \epsilon_t'}$$

The lagged response on the one-unit increase in the orthogonalized disturbances ϵ_t^* is denoted

$$\frac{\partial \mathbf{y}_{t+m}}{\partial \epsilon_{jt}^*} = \frac{\partial \mathrm{E}(\mathbf{y}_{t+m} | y_{jt}, y_{j-1,t}, \ldots, \mathbf{X}_t)}{\partial y_{jt}} = \Psi_m \mathbf{L}_j$$

where $\mathbf{L}_j$ is the jth column of the unit triangular matrix $\mathbf{L}$ and $\mathbf{X}_t = [\mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-p}]$. 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

$$\mathrm{E}(\mathbf{y}_{t+m} - \mathbf{y}_{t+m|t})(\mathbf{y}_{t+m} - \mathbf{y}_{t+m|t})' = \Sigma + \Psi_1 \Sigma \Psi_1' + \cdots + \Psi_{m-1} \Sigma \Psi_{m-1}'$$

Note that $\epsilon_t = \mathbf{L}\epsilon_t^*$. Then the covariance matrix of ϵ_t is decomposed

$$\Sigma = \sum_{i=1}^{n} \mathbf{L}_i \mathbf{L}_i' d_{ii}$$

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

$$\sum_{i=1}^{n} d_{ii} \left[\mathbf{L}_i \mathbf{L}_i' + \Psi_1 \mathbf{L}_i \mathbf{L}_i' \Psi_1' + \cdots + \Psi_{m-1} \mathbf{L}_i \mathbf{L}_i' \Psi_{m-1}' \right]$$

Therefore, the contribution of the ith orthogonalized innovation to the MSE is

$$\mathbf{V}_i = d_{ii} \left[\mathbf{L}_i \mathbf{L}_i' + \Psi_1 \mathbf{L}_i \mathbf{L}_i' \Psi_1' + \cdots + \Psi_{m-1} \mathbf{L}_i \mathbf{L}_i' \Psi_{m-1}' \right]$$

The ith forecast error variance decomposition is obtained from diagonal elements of the matrix $\mathbf{V}_i$.

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 $\mathrm{E}y_t = \mathrm{E}y_{t+d}$ and $\mathrm{E}y_s y_t = \mathrm{E}y_{s+d}y_{t+d}$. 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 ϵ_t is uncorrelated with mean zero and $\mathrm{E}\epsilon_t^2 = \sigma_t^2$, $p_t = p_{t+d}$, $\sigma_t^2 = \sigma_{t+d}^2$, 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, $\mathbf{x}_t = (x_{1t}, \ldots, x_{dt})'$, is autoregressive of order p, where $p = \max_j \mathrm{int}((p_j - j)/d) + 1$. 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

$$C_{YY}(k) = \mathrm{E}(Y_{t+k} Y_t)$$

where $\mathrm{E}Y_t = 0$. 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

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

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 $(\phi(B) = 1 - \phi_1 B)$ 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 $(\phi(B) = 1 - \phi_1 B - \phi_2 B^2)$ equals

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

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 $C_{YY}(k)$.

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

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 ϵ_t is a white noise with mean zero and constant variance σ^2.

The autocovariance function of white noise ϵ_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

$$\begin{aligned} C_{\epsilon\epsilon}(k) &= \sum_{m=0}^{p}\sum_{n=0}^{p} \phi_m\phi_n \int_{-1/2}^{1/2} \exp(2\pi ig(k - m + n))f_{YY}(g)dg \\ &= \int_{-1/2}^{1/2} \exp(2\pi igk) \left| 1 - \sum_{m=1}^{p} \phi_m \exp(-2\pi igm) \right|^2 f_{YY}(g)dg \end{aligned}$$

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,

$$f_{\epsilon\epsilon}(g) = \left|1 - \sum_{m=1}^{p} \phi_m \exp(-2\pi i g m)\right|^2 f_{YY}(g)$$

Since $f_{\epsilon\epsilon}(g) = \sigma^2$, the rational spectrum of Y_t is

$$f_{YY}(g) = \frac{\sigma^2}{\left|1 - \sum_{m=1}^{p} \phi_m \exp(-2\pi i g m)\right|^2}$$

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 ϵ_t is assumed to be Gaussian error disturbances. Combining $\mathbf{X}$ and $\mathbf{y}$ as

$$\mathbf{Z} = [\mathbf{X} \vdots \mathbf{y}]$$

the $\mathbf{Z}$ matrix can be decomposed as

$$\mathbf{Z} = \mathbf{QU} = \mathbf{Q} \begin{bmatrix} \mathbf{R} & \mathbf{w}_1 \\ \mathbf{0} & \mathbf{w}_2 \end{bmatrix}$$

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

$$\mathbf{Ra} = \mathbf{w}_1$$

The unbiased residual variance estimate is

$$\hat{\sigma}^2 = \frac{1}{T-p} \sum_{i=p+2}^{T-p} w_i^2 = \frac{w_{p+2}^2}{T-p}$$

and

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

In practice, least squares estimation does not require the orthogonal matrix $\mathbf{Q}$. The TIMSAC subroutines compute the upper triangular matrix without computing the matrix $\mathbf{Q}$.

Bayesian Constrained Least Squares

Consider the additive time series model

$$y_t = T_t + S_t + \epsilon_t, \qquad \epsilon_t \sim N(0, \sigma^2)$$

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

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

Using matrix notation,

$$(\mathbf{y} - \mathbf{Ma})'(\mathbf{y} - \mathbf{Ma}) + (\mathbf{a} - \mathbf{a}_0)'\mathbf{D}'\mathbf{D}(\mathbf{a} - \mathbf{a}_0)$$

where $\mathbf{M} = [\mathbf{I}_T \vdots \mathbf{I}_T]$, $\mathbf{y} = (y_1, \ldots, y_T)'$, and $\mathbf{a}_0$ is the initial guess of $\mathbf{a}$. The matrix $\mathbf{D}$ is a $3T \times 2T$ control matrix in which structure varies according to the order of differencing in trend and season.

$$\mathbf{D} = d \begin{bmatrix} \mathbf{E}_m & \mathbf{0} \\ z\mathbf{F} & \mathbf{0} \\ \mathbf{0} & s\mathbf{G}_k \end{bmatrix}$$

where

$$
\begin{aligned}
\mathbf{E}_m &= \mathbf{C}_m \otimes \mathbf{I}_L, \qquad m = 1, 2, 3 \\
\mathbf{F} &= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 1 & \cdots & 1 & 1 \end{bmatrix}_{T \times T} \\
\mathbf{G}_1 &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{T \times T} \\
\mathbf{G}_2 &= \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix}_{T \times T} \\
\mathbf{G}_3 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ -3 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 3 & -3 & 1 & 0 & 0 & \cdots & 0 \\ -1 & 3 & -3 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 3 & -3 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 3 & -3 & 1 \end{bmatrix}_{T \times T}
\end{aligned}
$$

The $n \times n$ matrix $\mathbf{C}_m$ has the same structure as the matrix $\mathbf{G}_m$, and $\mathbf{I}_L$ is the $L \times L$ identity matrix. The solution of the constrained least squares method is equivalent to that of maximizing the following function

$$
L(\mathbf{a}) = \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{Ma})'(\mathbf{y} - \mathbf{Ma})\right\} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{a} - \mathbf{a}_0)'\mathbf{D}'\mathbf{D}(\mathbf{a} - \mathbf{a}_0)\right\}
$$

Therefore, the PDF of the data $\mathbf{y}$ is

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

The prior PDF of the parameter vector $\mathbf{a}$ is

$$
\pi(\mathbf{a}|\mathbf{D}, \sigma^2, \mathbf{a}_0) = \left(\frac{1}{2\pi}\right)^{T} \left(\frac{1}{\sigma}\right)^{2T} |\mathbf{D}'\mathbf{D}| \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{a} - \mathbf{a}_0)'\mathbf{D}'\mathbf{D}(\mathbf{a} - \mathbf{a}_0)\right\}
$$

When the constant d is known, the estimate $\hat{\mathbf{a}}$ of $\mathbf{a}$ is the mean of the posterior distribution, where the posterior PDF of the parameter $\mathbf{a}$ is proportional to the function $L(\mathbf{a})$. It is obvious that $\hat{\mathbf{a}}$ is the minimizer of $\|\mathbf{g}(\mathbf{a}|d)\|^2 = (\tilde{\mathbf{y}} - \tilde{\mathbf{D}}\mathbf{a})'(\tilde{\mathbf{y}} - \tilde{\mathbf{D}}\mathbf{a})$, where

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{D}\mathbf{a}_0 \end{bmatrix}$$

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

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

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

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:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{F}\mathbf{x}_{t-1} + \mathbf{G}\mathbf{w}_t \\ y_t &= \mathbf{H}_t\mathbf{x}_t + \epsilon_t \end{aligned}$$

where $\epsilon_t \sim N(0, \sigma^2)$ and $\mathbf{w}_t \sim N(\mathbf{0}, \mathbf{Q})$. If the observations, $(y_1, \ldots, y_T)$, and the initial conditions, $\mathbf{x}_{0|0}$ and $\mathbf{P}_{0|0}$, are available, the one-step predictor $(\mathbf{x}_{t|t-1})$ of the state vector $\mathbf{x}_t$ and its mean square error (MSE) matrix $\mathbf{P}_{t|t-1}$ are written as

$$\mathbf{x}_{t|t-1} = \mathbf{F}\mathbf{x}_{t-1|t-1}$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}\mathbf{P}_{t-1|t-1}\mathbf{F}' + \mathbf{G}\mathbf{Q}\mathbf{G}'$$

Using the current observation, the filtered value of $\mathbf{x}_t$ and its variance $\mathbf{P}_{t|t}$ are updated.

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{K}_t e_t$$

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1}$$

where $e_t = y_t - \mathbf{H}_t\mathbf{x}_{t|t-1}$ and $\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t'[\mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t' + \sigma^2\mathbf{I}]^{-1}$. 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 $v_{t|t-1}$ is the conditional variance of the one-step prediction error e_t.

Consider the additive time series decomposition

$$y_t = T_t + S_t + TD_t + u_t + \mathbf{x}_t'\beta_t + \epsilon_t$$

where $\mathbf{x}_t$ is a $(K \times 1)$ regressor vector and β_t is a $(K \times 1)$ time-varying coefficient vector. Each component has the following constraints:

$$\begin{aligned}
\nabla^k T_t &= w_{1t}, \qquad w_{1t} \sim N(0,\tau_1^2) \\
\nabla_L^m S_t &= w_{2t}, \qquad w_{2t} \sim N(0,\tau_2^2) \\
u_t &= \sum_{i=1}^{p} \alpha_i u_{t-i} + w_{3t}, \qquad w_{3t} \sim N(0,\tau_3^2) \\
\beta_{jt} &= \beta_{j,t-1} + w_{3+j,t}, \qquad w_{3+j,t} \sim N(0,\tau_{3+j}^2), \quad j = 1, \cdots, K \\
\sum_{i=1}^{7} \gamma_{it} TD_t(i) &= \sum_{i=1}^{6} \gamma_{it}(TD_t(i) - TD_t(7)) \\
\gamma_{it} &= \gamma_{i,t-1}
\end{aligned}$$

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

$$\begin{aligned}
T_t &= 3T_{t-1} - 3T_{t-2} + T_{t-3} + w_{1t} \\
\sum_{i=0}^{11} S_{t-i} &= w_{2t} \\
u_t &= \sum_{i=1}^{3} \alpha_i u_{t-i} + w_{3t}
\end{aligned}$$

The state vector is defined as

$$\mathbf{x}_t = (T_t, T_{t-1}, T_{t-2}, S_t, \ldots, S_{t-11}, u_t, u_{t-1}, u_{t-2}, \gamma_{1t}, \ldots, \gamma_{6t})'$$

The matrix $\mathbf{F}$ is

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & 0 & 0 & 0 \\ 0 & \mathbf{F}_2 & 0 & 0 \\ 0 & 0 & \mathbf{F}_3 & 0 \\ 0 & 0 & 0 & \mathbf{F}_4 \end{bmatrix}$$

where

$$\mathbf{F}_1 = \begin{bmatrix} 3 & -3 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{F}_2 = \begin{bmatrix} -\mathbf{1}' & -1 \\ \mathbf{I}_{10} & 0 \end{bmatrix}$$

$$\mathbf{F}_3 = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{F}_4 = \mathbf{I}_6$$

$$\mathbf{1}' = (1, 1, \ldots, 1)$$

The matrix G can be denoted as

$$G = \begin{bmatrix} \mathbf{g}_1 & 0 & 0 \\ 0 & \mathbf{g}_2 & 0 \\ 0 & 0 & \mathbf{g}_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where

$$\mathbf{g}_1 = \mathbf{g}_3 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}'$$

$$\mathbf{g}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}'$$

Finally, the matrix $\mathbf{H}_t$ is time-varying,

$$\mathbf{H}_t = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \mathbf{h}_t' \end{bmatrix}$$

where

$$\begin{aligned} \mathbf{h}_t &= \begin{bmatrix} D_t(1) & D_t(2) & D_t(3) & D_t(4) & D_t(5) & D_t(6) \end{bmatrix}' \\ D_t(i) &= TD_t(i) - TD_t(7), \qquad i = 1, \ldots, 6 \end{aligned}$$

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, MISSING=0, MISSING=1, or MISSING=2. When the MISSING=0 (default) option is specified, the first contiguous observations with no missing values are used. The 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 $\mathbf{g}(\mathbf{a}|d)$ that correspond to the missing observations. The vector $\mathbf{g}(\mathbf{a}|d)$ 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, TSROOT, 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

$$\mathbf{y}_t = \mathbf{A}_0 + \mathbf{A}_1\mathbf{y}_{t-1} + \mathbf{A}_2\mathbf{y}_{t-2} + \mathbf{A}_3\mathbf{y}_{t-3} + \epsilon_t$$

The matrix ARCOEF contains the AR coefficients ($\mathbf{A}_1$,$\mathbf{A}_2$, and $\mathbf{A}_3$), and the matrix EV contains error covariance estimates. An intercept vector $\mathbf{A}_0$ 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   304 751  653   307 763  668
 317 766  679  314 779  686   306 808  697   304 785  688
 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 $\mathbf{L}^{-1}$ and diagonal matrix $\mathbf{D}_t$, 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

                          295.21042 190.94664 59.361516
                                  1         0         0
                           -0.02239         1         0
                          -0.256341 -0.500803         1
```

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 $\mathbf{A}_i = \mathbf{L}\mathbf{A}_i^*(i = 0, 1, 2, 3)$, where $\mathbf{A}_i^*$ is a coefficient matrix of the instantaneous model. For example, you can verify this result using the first lag coefficient matrix $(\mathbf{A}_1)$.

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

Output 10.1.2. VAR Estimates

```
                      -----  AR-COEFFICIENTS  -----
          LAG    VAR =   1    VAR =   2    VAR =   3

            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
```

Output 10.1.3. Instantaneous Response Model Estimates

```
                -----  AR  -----

       0.885593     0.340174    -0.014398
       0.148624     1.042645     0.107386
      -0.222272    -0.154018     0.397440

      -0.059195    -0.298777     0.162982
       0.114188    -0.037349    -0.091835
       0.127145     0.072796    -0.023287

       0.063723    -0.196504     0.069575
      -0.227986     0.057646    -0.101366
      -0.206570    -0.115316     0.289790
```

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=h 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.

```
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,

$$\frac{\partial \mathbf{y}_{t+2}}{\partial \epsilon_t'} = \left[\begin{array}{ccc} 0.781100 & 0.353140 & 0.180211 \\ 0.448501 & 1.165474 & 0.069731 \\ 0.364611 & 0.692111 & 0.222342 \end{array} \right]$$

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)

```
                           IMPULSE

                    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
```

In addition, you can compute the lagged response on the one-unit increase in the orthogonalized disturbances ϵ_t^*.

$$\frac{\partial \mathbf{y}_{t+m}}{\partial \epsilon_{jt}^*} = \frac{\partial \mathrm{E}(\mathbf{y}_{t+m}|y_{jt}, y_{j-1,t}, \ldots, \mathbf{X}_t)}{\partial y_{jt}} = \Psi_m \mathbf{L}_j$$

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 $\mathbf{L}^{-1}$.

```
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

```
                           ORTH_IMP

                    1         0          0
            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
            0.8979858  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
```

You can verify the result for the case of

$$\frac{\partial \mathbf{y}_{t+2}}{\partial \epsilon^*_{2t}} = \frac{\partial \mathrm{E}(\mathbf{y}_{t+2}|y_{2t}, y_{1t}, \ldots, \mathbf{X}_t)}{\partial y_{2t}} = \Psi_2 \mathbf{L}_2$$

using the simple computation

$$\begin{bmatrix} 0.443390 \\ 1.200395 \\ 0.803461 \end{bmatrix} = \begin{bmatrix} 0.781100 & 0.353140 & 0.180211 \\ 0.448501 & 1.165474 & 0.069731 \\ 0.364611 & 0.692111 & 0.222342 \end{bmatrix} \begin{bmatrix} 0.000000 \\ 1.000000 \\ 0.500803 \end{bmatrix}$$

The contribution of the ith orthogonalized innovation to the mean square error matrix of the 10-step forecast is computed using the formula

$$d_{ii}[\mathbf{L}_i\mathbf{L}_i' + \Psi_1\mathbf{L}_i\mathbf{L}_i'\Psi_1' + \cdots + \Psi_9\mathbf{L}_i\mathbf{L}_i'\Psi_9']$$

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).

```
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

```
                 CONTRIB                          VAR

               1879.3774  238.08543  46.247569  2163.7104
               935.54383  3636.8824  1.5546701  4573.9809
               452.67794  1916.1676  97.660432  2466.506
```

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 ith 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 (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.

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

Output 10.1.7. Innovation Account

```
                                ACCOUNT

                      86.859008  11.003572  2.1374196
                      20.453602  79.512409  0.0339894
                      18.353004  77.687531  3.9594646
```

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

$$\mathbf{y}_t = \mathbf{b}_t + \mathbf{H}_t\mathbf{z}_t + \epsilon_t$$

where $\mathbf{b}_t$ is an $N_y \times 1$ vector, $\mathbf{H}_t$ is an $N_y \times N_z$ matrix, the sequence of observation noise ϵ_t is independent, $\mathbf{z}_t$ is an $N_z \times 1$ state vector, and $\mathbf{y}_t$ 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.

$$\mathbf{z}_{t+1} = \mathbf{a}_t + \mathbf{F}_t\mathbf{z}_t + \eta_t$$

where $\mathbf{a}_t$ is an $N_z \times 1$ vector, $\mathbf{F}_t$ is an $N_z \times N_z$ transition matrix, and the sequence of transition noise η_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

$$\mathbf{z}_t = \mathbf{a}_t + \mathbf{F}_t\mathbf{z}_{t-1} + \eta_t$$

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, $E(\eta_t\epsilon_t') = 0$. It is assumed that

$$\begin{aligned} E(\eta_t\eta_s') &= \mathbf{V}_t\delta_{ts} \\ E(\epsilon_t\epsilon_s') &= \mathbf{R}_t\delta_{ts} \\ E(\eta_t\epsilon_s') &= \mathbf{G}_t\delta_{ts} \end{aligned}$$

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

$$\begin{aligned}
\mathbf{y}_t &= \mathbf{X}_t\beta + \mathbf{H}_t\mathbf{z}_t + \epsilon_t \\
\mathbf{z}_{t+1} &= \mathbf{W}_t\beta + \mathbf{F}_t\mathbf{z}_t + \eta_t \\
\mathbf{z}_0 &= \mathbf{a} + \mathbf{A}\delta \\
\beta &= \mathbf{b} + \mathbf{B}\delta
\end{aligned}$$

where δ is a random variable with a mean of μ 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 δ 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 μ_t is a trend component and ϵ_t is a white noise error with $\epsilon_t \sim (0, \sigma_\epsilon^2)$. 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:

$$\begin{aligned}
\mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_{1t} \\
\beta_t &= \beta_{t-1} + \eta_{2t}
\end{aligned}$$

where η_{1t} and η_{2t} are independent white noise disturbances with $\eta_{1t} \sim (0, \sigma^2_{\eta_1})$ and $\eta_{2t} \sim (0, \sigma^2_{\eta_2})$.

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

$$\begin{aligned} y_t &= \mathbf{H}\mathbf{z}_t + \epsilon_t \\ \mathbf{z}_t &= \mathbf{F}\mathbf{z}_{t-1} + \eta_t \end{aligned}$$

where

$$\begin{aligned} \mathbf{H} &= (1, 0) \\ \mathbf{F} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \\ \mathbf{z}_t &= (\mu_t, \beta_t)' \\ \eta_t &= (\eta_{1t}, \eta_{2t})' \\ \mathrm{Var}\left(\begin{bmatrix} \epsilon_t \\ \eta'_t \end{bmatrix}\right) &= \begin{bmatrix} \sigma^2_\epsilon & 0 & 0 \\ 0 & \sigma^2_{\eta 1} & 0 \\ 0 & 0 & \sigma^2_{\eta 2} \end{bmatrix} \end{aligned}$$

When the observation noise ϵ_t is normally distributed, the average log likelihood function of the SSM is

$$\begin{aligned} \ell &= \frac{1}{T}\sum_{t=1}^{T} \ell_t \\ \ell_t &= -\frac{N_y}{2}\log(2\pi) - \frac{1}{2}\log(|\mathbf{C}_t|) - \frac{1}{2}\hat{\epsilon}'_t\mathbf{C}_t^{-1}\hat{\epsilon}_t \end{aligned}$$

where $\mathbf{C}_t$ is the mean square error matrix of the prediction error $\hat{\epsilon}_t$, such that $\mathbf{C}_t = \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}' + \mathbf{R}_t$.

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=10^6I. 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=10^6I) 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=10I). The data are the annual real GNP for the years 1909 to 1969.

```
title 'Likelihood Evaluation of SSM';
title2 'DATA: Annual Real GNP 1909-1969';
data gnp;
   input y @@;
```

```
datalines;
116.8 120.1 123.2 130.2 131.4 125.6 124.5 134.3
135.2 151.8 146.4 139.0 127.8 147.0 165.9 165.5
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
263.7 297.8 337.1 361.3 355.2 312.6 309.9 323.7
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),nrow(f),.);
   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);
```

```
      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;
```

Output 10.2.1. Average Log Likelihood of SSM

```
                   Likelihood Evaluation of SSM
                 DATA: Annual Real GNP 1909-1969

                   No initial values are given

                               LOGL

                            -26314.66

                      Initial values are given


                               LOGL

                            -91884.41
```

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

```
                       Y       PRED                FILT

                   116.8          0           0  116.78832           0
                   120.1  116.78832           0  120.09967   3.3106857
                   123.2  123.41035   3.3106857  123.22338   3.1938303
                   130.2  126.41721   3.1938303  129.59203   4.8825531
                   131.4  134.47459   4.8825531  131.93806   3.5758561
                   125.6  135.51391   3.5758561  127.36247  -0.610017
                   124.5  126.75246  -0.610017  124.90123  -1.560708
                   134.3  123.34052  -1.560708  132.34754   3.0651076
                   135.2  135.41265   3.0651076  135.23788   2.9753526
                   151.8  138.21324   2.9753526  149.37947   8.7100967
                   146.4  158.08957   8.7100967  148.48254   3.7761324
                   139    152.25867   3.7761324  141.36208  -1.82012
                   127.8  139.5419   -1.82012    129.89187  -6.776195
                   147    123.11568  -6.776195   142.74492   3.3049584
                   165.9  146.04988   3.3049584  162.36363  11.683345
                   165.5  174.04698  11.683345  167.02267   8.075817
```

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:

$$\begin{aligned}\mathbf{y}_t &= \mathbf{H}\mathbf{z}_t + \epsilon_t \\ \mathbf{z}_t &= \mathbf{F}\mathbf{z}_{t-1} + \eta_t\end{aligned}$$

where $\mathbf{H}$ is a 2×2 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 μ and covariance Σ. For estimation, the Σ matrix is fixed as

$$\begin{bmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{bmatrix}$$

while the mean vector μ 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

$$\begin{aligned}\mathbf{F}^{i+1} &= \mathbf{S}_t(1)[\mathbf{S}_{t-1}(0)]^{-1} \\ \mathbf{V}^{i+1} &= \frac{1}{T}\left(\mathbf{S}_t(0) - \mathbf{S}_t(0)[\mathbf{S}_{t-1}(0)]^{-1}\mathbf{S}'_t(1)\right) \\ \mathbf{R}^{i+1} &= \frac{1}{T}\sum_{t=1}^{T}\left[(\mathbf{y}_t - \mathbf{H}\mathbf{z}_{t|T})(\mathbf{y}_t - \mathbf{H}\mathbf{z}_{t|T})' + \mathbf{H}\mathbf{P}_{t|T}\mathbf{H}'\right] \\ \mu^{i+1} &= \mathbf{z}_{0|T}\end{aligned}$$

where the index i represents the current iteration number, and

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

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

$$\mathbf{P}_{t-1,t-2|T} = \mathbf{P}_{t-1|t-1}\mathbf{P}^{*\prime}_{t-2} + \mathbf{P}^{*}_{t-1}(\mathbf{P}_{t,t-1|T} - \mathbf{F}\mathbf{P}_{t-1|t-1})\mathbf{P}^{*\prime}_{t-2}$$

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

$$\mathbf{P}_{T,T-1|T} = \left[\mathbf{I} - \mathbf{P}_{t|t-1}\mathbf{H}'(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}' + \mathbf{R})\mathbf{H}\right]\mathbf{F}\mathbf{P}_{T-1|T-1}$$

Note that the initial value of the state vector is updated for each iteration

$$\begin{aligned}\mathbf{z}_{1|0} &= \mathbf{F}\mu^i \\ \mathbf{P}_{1|0} &= \mathbf{F}^i\Sigma\mathbf{F}^{i\prime} + \mathbf{V}^i\end{aligned}$$

The objective function value is computed as -2ℓ in the IML module LIK. The log-likelihood function is written

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

where $\mathbf{C}_t = \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}' + \mathbf{R}$.

The iteration history is shown in Output 10.3.1. As shown in Output 10.3.2, the eigenvalues of $\mathbf{F}$ 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 $\mathbf{y}_t$ are computed using the KALCVF call.

```
call kalcvf(pred,vpred,filt,vfilt,y,15,a,f,b,h,var,z0,vz0);
```

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

```
title 'SSM Estimation using EM Algorithm';
data one;
   input y1 y2 @@;
datalines;
   /*. . . data lines omitted . . .*/
;

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};
```

```
/*-- 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_ii1 = (i(nz)-kt*h)*f*p_t1_t1;
    st0 = vsm[(t-1)*nz+1:t*nz,] + sm[t,]‘*sm[t,];
```

```
   st1 = p_t_ii1 + sm[t,]‘*sm[t-1,];
   st00 = p_t_0 + z_t_0 * z_t_0‘;
   cov = (y[t,]‘ - h*sm[t,]‘) * (y[t,]‘ - h*sm[t,]‘)‘ +
         h*vsm[(t-1)*nz+1:t*nz,]*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*jt2‘ + jt1*(p_t_ii1 - f*p_i1_i1)*jt2‘;
      p_t_ii1 = p_t_i1i2;
      temp = vsm[(i-2)*nz+1:(i-1)*nz,];
      sm1 = sm[i-1,]‘;
      st0 = st0 + ( temp + sm1 * sm1‘ );
      if ( i > 2 ) then
         st1 = st1 + ( p_t_ii1 + sm1 * sm[i-2,]);
      else st1 = st1 + ( p_t_ii1 + sm1 * z_t_0‘);
      st00 = st00 + ( temp + sm1 * sm1‘ );
      cov = cov + ( h * temp * h‘ +
            (y[i-1,]‘ - h * sm1)*(y[i-1,]‘ - h * sm1)‘ );
   end;

   /*-- 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*log 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;
```

```
      /*-- 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

```
                     SSM Estimation using EM Algorithm

      Iter -2*log L      F11      F12      F21      F22    MYU11    MYU22

    1.0000 -154.010   1.0000   0.0000   0.0000   1.0000   0.0000   0.0000
    2.0000 -237.962   0.7952  -0.6473   0.3263   0.5143   0.0530   0.0840
    3.0000 -238.083   0.7967  -0.6514   0.3259   0.5142   0.1372   0.0977
    4.0000 -238.126   0.7966  -0.6517   0.3259   0.5139   0.1853   0.1159
    5.0000 -238.143   0.7964  -0.6519   0.3257   0.5138   0.2143   0.1304
    6.0000 -238.151   0.7963  -0.6520   0.3255   0.5136   0.2324   0.1405
    7.0000 -238.153   0.7962  -0.6520   0.3254   0.5135   0.2438   0.1473
    8.0000 -238.155   0.7962  -0.6521   0.3253   0.5135   0.2511   0.1518
    9.0000 -238.155   0.7962  -0.6521   0.3253   0.5134   0.2558   0.1546
   10.0000 -238.155   0.7961  -0.6521   0.3253   0.5134   0.2588   0.1565
```

Output 10.3.2. Eigenvalues of F Matrix

```
                Real       Imag       MOD

             0.6547534   0.438317  0.7879237
             0.6547534  -0.438317  0.7879237
```

Output 10.3.3. Multistep Prediction

```
             n-Step    Z1_T_n     Z2_T_n     SE_Z1     SE_Z2

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

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.

```
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).

```
   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 kalcvf_p kaldff_p;
```

Output 10.4.1. Diffuse Kalman Filtering

```
                         Diffuse Kalman Filtering

                     KALCVF_P            KALDFF_P

                            0         0         0         0
                     1.441911  0.961274 1.1214871 0.9612746
                    -0.882128 -0.267663 -0.882138 -0.267667
                    -0.723156 -0.527704 -0.723158 -0.527706
                    1.2964969  0.871659 1.2964968 0.8716585
                    -0.035692 0.1379633 -0.035692 0.1379633
                    -2.698135 -1.967344 -2.698135 -1.967344
                    -5.010039 -4.158022 -5.010039 -4.158022
                    -9.048134 -7.719107 -9.048134 -7.719107
                    -8.993153 -8.508513 -8.993153 -8.508513
                    -11.16619 -10.44119 -11.16619 -10.44119
                    -10.42932 -10.34166 -10.42932 -10.34166
                    -8.331091 -8.822777 -8.331091 -8.822777
                    -9.578258 -9.450848 -9.578258 -9.450848
                    -6.526855 -7.241927 -6.526855 -7.241927
                    -5.218651 -5.813854 -5.218651 -5.813854
                     -5.01855 -5.291777  -5.01855 -5.291777
                      -6.5699 -6.284522   -6.5699 -6.284522
                    -4.613301 -4.995434 -4.613301 -4.995434
                    -5.057926  -5.09007 -5.057926  -5.09007
```

The likelihood function for the diffuse Kalman filter under the finite initial covariance matrix Σ_δ is written

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

where $\mathbf{y}^(\#)$ is the dimension of the matrix $(\mathbf{y}_1', \cdots, \mathbf{y}_T')'$. The likelihood function for the diffuse Kalman filter under the diffuse initial covariance matrix $(\Sigma_\delta \to \infty)$ is computed as $\lambda(\mathbf{y}) - \frac{1}{2}\log(|\mathbf{S}|)$, where the $\mathbf{S}$ matrix is the upper $N_\delta \times N_\delta$ matrix of $\mathbf{Q}_t$. 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.

```
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

```
               Log L         -11.42547

               Diffuse Log L -9.457596
```

References

Afifi, A.A. and Elashoff, R.M. (1967), “Missing Observations in Multivariate Statistics II. Point Estimation in Simple Linear Regression,” *Journal of the American Statistical Association*, 62, 10−29.

Akaike, H. (1974), “A New Look at Statistical Model Identification,” *IEEE Transactions on Automatic Control*, 19, 716−723.

Akaike, H. (1977), “On Entropy Maximization Principle,” in *Applications of Statistics*, ed. P.R. Krishnaiah, Amsterdam: North-Holland Publishing Co., 27−41.

Akaike, H. (1978a), “A Bayesian Analysis of the Minimum AIC Procedure,” *Ann. Inst. Statist. Math.*, 30, 9−14.

Akaike, H. (1978b), “Time Series Analysis and Control through Parametric Models,” in *Applied Time Series Analysis*, ed. D.F. Findley, New York: Academic Press, 1−23.

Akaike, H. (1979), “A Bayesian Extension of the Minimum AIC Procedure of Autoregressive Model Fitting,” *Biometrika*, 66, 237−242.

Akaike, H. (1980a), “Likelihood and the Bayes Procedure,” *Bay Statistics*, eds. J.M. Bernardo, M.H. DeGroot, D.V. Lindley, and M. Smith, Valencia, Spain: University Press.

Akaike, H. (1980b), “Seasonal Adjustment by a Bayesian Modeling,” *Journal of Time Series Analysis*, 1, 1−13.

Akaike, H. (1981), “Likelihood of a Model and Information Criteria,” *Journal of Econometrics*, 16, 3−14.

Akaike, H. (1986), “The Selection of Smoothness Priors for Distributed Lag Estimation,” in *Bayesian Inference and Decision Techniques*, ed. P. Goel and A. Zellner, Elsevier Science Publishers, 109−118.

Akaike, H. and Ishiguro, M. (1980), “Trend Estimation with Missing Observations,” *Ann. Inst. Statist. Math.*, 32, 481−488.

Akaike, H. and Nakagawa, T. (1988), *Statistical Analysis and Control of Dynamic Systems*, Tokyo: KTK Scientific Publishers.

Anderson, T.W. (1971), *The Statistical Analysis of Time Series*, New York: John Wiley & Sons, Inc.

Brockwell, P.J. and Davis, R.A. (1991), *Time Series: Theory and Methods*, Second Edition, New York: Springer-Verlag.

Doan, T.; Litterman, R.; and Sims, C. (1984), “Forecasting and Conditional Projection using Realistic Prior Distributions,” *Econometric Review*, 3, 1−100.

Gersch, W. and Kitagawa, G. (1983), “The Prediction of Time Series with Trends and Seasonalities,” *Journal of Business and Economic Statistics*, 1, 253−264.

Harvey, A.C. (1989), *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge: Cambridge University Press.

Ishiguro, M. (1984), “Computationally Efficient Implementation of a Bayesian Seasonal Adjustment Procedure,” *Journal of Time Series Analysis*, 5, 245−253.

Ishiguro, M. (1987), “TIMSAC-84: A New Time Series Analysis and Control Package,” *Proceedings of American Statistical Association: Business and Economic Section*, 33−42.

Jones, R.H. and Brelsford, W.M. (1967), “Time Series with Periodic Structure,” *Biometrika*, 54, 403−408.

Kitagawa, G. (1981), “A Nonstationary Time Series Model and its Fitting by a Recursive Filter,” *Journal of Time Series Analysis*, 2, 103−116.

Kitagawa, G. (1983), “Changing Spectrum Estimation,” *Journal of Sound and Vibration*, 89, 433−445.

Kitagawa, G. and Akaike, H. (1978), “A Procedure for the Modeling of Non-Stationary Time Series,” *Ann. Inst. Statist. Math.*, 30, 351−363.

Kitagawa, G. and Akaike, H. (1981), “On TIMSAC-78,” in *Applied Time Series Analysis II*, ed. D.F. Findley, New York: Academic Press, 499−547.

Kitagawa, G. and Akaike, H. (1982), “A Quasi Bayesian Approach to Outlier Detection,” *Ann. Inst. Statist. Math.*, 34, 389−398.

Kitagawa, G. and Gersch, W. (1984), “A Smoothness Priors-State Space Modeling of Time Series with Trend and Seasonality,” *Journal of the American Statistical Association*, 79, 378−389.

Kitagawa, G. and Gersch, W. (1985a), “A Smoothness Priors Time-Varying AR Coefficient Modeling of Nonstationary Covariance Time Series,” *IEEE Transactions on Automatic Control*, 30, 48−56.

Kitagawa, G. and Gersch, W. (1985b), “A Smoothness Priors Long AR Model Method for Spectral Estimation,” *IEEE Transactions on Automatic Control*, 30, 57−65.

Lutkepohl, H. (1991), *Introduction to Multiple Time Series Analysis*, Berlin: Springer-Verlag.

Pagano, M. (1978), “On Periodic and Multiple Autoregressions,” *The Annals of Statistics*, 6, 1310−1317.

Sakamoto, Y.; Ishiguro, M.; and Kitagawa, G. (1986), *Akaike Information Criterion Statistics*, Tokyo: KTK Scientific Publishers.

Shiller, R.J. (1973), “A Distributed Lag Estimator Derived from Smoothness Priors,” *Econometrica*, 41, 775−788.

Tamura, Y. H. (1986), “An Approach to the Nonstationary Process Analysis,” *Ann. Inst. Statist. Math.*, 39, 227−241.

Wei, W.W.S. (1990), *Time Series Analysis: Univariate and Multivariate Methods*, Redwood: Addison-Wesley.

Whittaker, E.T. (1923), “On a New Method of Graduation,” *Proceedings of the Edinborough Mathematical Society*, 41, 63−75.

Whittaker, E.T. and Robinson, G. (1944), *Calculus of Observation*, Fourth Edition, London: Blackie & Son Limited.

Zellner, A. (1971), *An Introduction to Bayesian Inference in Econometrics*, New York: John Wiley & Sons, Inc.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/IML User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 846 pp.

SAS/IML User's Guide, Version 8

ISBN 1–58025–553–1

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

The Institute is a private company devoted to the support and further development of its software and related services.