Bayesian Autoregressive and Time-Varying Coefficients Time Series Models

Overview

The MCMC procedure in SAS/STAT 14.1 was enhanced with the ability to access lead and lagged values for random variables that are indexed. Two types of random variables in PROC MCMC are indexed: the response (MODEL statement) is indexed by observations, and the random effect (RANDOM statement) is indexed by the SUBJECT= option variable. This new capability extends the range of models that you can fit by using PROC MCMC. For example, you can now fit certain dynamic linear models that were previously inaccessible. In other cases, model specification in PROC MCMC is greatly simplified. For example, to fit autoregressive time series models before this enhancement, you were required to preprocess the data and manually generate variables that contain the lagged values of the response variable. Also, there was no built-in means to account for the initial states of the lagged variables. With the new enhancement, autoregressive time series models no longer require you to preprocess the data, and you can easily specify starting values or prior distributions for the unobserved initial states. What follows are two examples that demonstrate the use of this enhancement to PROC MCMC. The first example fits a fourth-order autoregressive model (AR(4)). The data in the example are simulated in order to avoid the issue of model identification. The second example fits a dynamic linear model with time-varying coefficients to UK coal consumption data, inspired by examples from Congdon (2003) and Harvey (1989). The first example demonstrates the use of lagged values in the MODEL statement, and the second example demonstrates the use of lagged values in the RANDOM statement. Both examples also demonstrate how to produce forecasts by using the PREDDIST statement.

The SAS source code for this example is available as an attachment in a text file. In Adobe Acrobat, right-click the icon and select **Save Embedded File to Disk**. You can also double-click the icon to open the file.

source code

Accessing Lag and Lead Variables In PROC MCMC

Two types of random variables in PROC MCMC are indexed: the response (MODEL statement) is indexed by observations, and the random effect (RANDOM statement) is indexed by the SUBJECT= option variable. As the procedure steps through the input data set, the response or the random-effects symbols are filled with values of the current observation or the random-effects parameters in the current subject. Often you might want to access lag or lead variables across an index. For example, the likelihood function for y_i can depend on y_{i-1} in an autoregressive model, or the prior distribution for μ_j can depend on μ_k, where $k \neq j$, in a dynamic linear model. In these situations, you can use the following rules to construct symbols to access values from other observations or subjects:

rv.*Li*: the *i*th lag of the variable rv. This looks back to the past.

rv.*Ni*: the *i*th lead of the variable rv. This looks forward to the future.

The construction is allowed for random variables that are associated with an index, either a response variable or a random-effects variable. You concatenate the variable name, a dot, either the letter *L* (for "lag") or the letter *N* (for "next"), and a lag or a lead number. PROC MCMC resolves these variables according to the indices that are associated with the random variable, with respect to the current observation.

For example, the following RANDOM statement specifies a first-order Markov dependence in the random effect mu that is indexed by the subject variable time:

```
random mu ~ normal(mu.l1,sd=1) subject=time;
```

This corresponds to the prior

$$\mu_t \sim \text{normal}(\mu_{t-1}, \text{sd=1})$$

At each observation, PROC MCMC fills in the symbol mu with the random-effects parameter μ_t that belongs to the current cluster t. To fill in the symbol mu.l1, the procedure looks back and finds a lag-1 random-effects parameter, μ_{t-1}, from the last cluster t–1. As the procedure moves forward in the input data set, these two symbols are constantly updated, as appropriate.

When the index is out of range, such as t–1 when t is 1, PROC MCMC fills in the missing state from the ICOND= option in either the MODEL or RANDOM statement. The following example illustrates how PROC MCMC fills in the values of these lag and lead variables as it steps through the data set.

Assume that the random effect mu has five levels, indexed by $\text{sub} = \{\text{a}, \text{b}, \text{c}, \text{d}, \text{e}\}$. The model contains two lag variables and one lead variable (mu.l1, mu.l2, and mu.n2):

```
mn = (mu.l1 + mu.l2 + mu.n2) / 3
random mu ~ normal(mn, sd=1) subject=time icond=(alpha beta gamma kappa);
```

In this setup, instead of a list of five random-effects parameters that the variable mu can be assigned values to, you now have a list of nine variables for the variables mu, mu.l1, mu.l2, and mu.n2. The list lines up in the following order:

$$\text{alpha}, \text{beta}, \mu_\text{a}, \mu_\text{b}, \mu_\text{c}, \mu_\text{d}, \mu_\text{e}, \text{gamma}, \text{kappa}$$

PROC MCMC finds relevant symbol values according to this list, as the procedure steps through different subject cluster. The process is illustrated in Table 1.

Table 1 Processing Lag and Lead Variables

sub	mu	mu.l2	mu.l1	mu.n2
a	μ_a	alpha	beta	μ_c
b	μ_b	beta	μ_a	μ_d
c	μ_c	μ_a	μ_b	μ_e
d	μ_d	μ_b	μ_c	gamma
e	μ_e	μ_c	μ_d	kappa

For observations in cluster a, PROC MCMC sets the random-effects variable mu to μ_{a}, looks back two lags and fills in mu.l2 with alpha, looks back one lag and fills in mu.l1 with beta, and looks forward two leads and fills in mu.n2 with μ_{c}. As the procedure moves to observations in cluster b, mu becomes μ_{b}, mu.l2 becomes beta, mu.l1 becomes μ_{a}, and mu.n2 becomes μ_{d}. For observations in the last cluster, cluster e, mu becomes μ_{e}, mu.l2 becomes μ_{c}, mu.l1 becomes μ_{d}, and mu.n2 is filled with kappa.

Autoregressive Model of Order *p* (AR(*p*))

The observational equation for an AR(p) model is

$$y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \epsilon_t \qquad \text{for } t = 1, \ldots, T$$

where c is a constant, $\phi_1, \ldots, \phi_p$ are the unknown autoregressive parameters, and $\epsilon_t \sim N(0, \sigma^2)$. The response variable y is assumed to be observed at equally spaced intervals that are indexed by t.

If you condition on the unobserved initial states, $y_0, \ldots, y_{-p+1}$, then the likelihood for y_t is

$$p(y_t | y_{t-1}, \ldots, y_{t-p}, \boldsymbol{\phi}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_t - c - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p})^2}{2\sigma^2} \right)$$

Including prior distributions for $y_0, \ldots, y_{-p+1}$ in the model specification leads to what is known as a full likelihood model (Congdon 2003).

To perform a Bayesian analysis, you assign prior distributions to the unknown quantities $c, \phi_1, \ldots, \phi_p, \sigma^2, y_0, \ldots, y_{1-p}$. In the following example, the following prior distributions are assigned to these quantities:

$$\begin{aligned}
c &\sim \text{normal}(\mu_c, \sigma_c^2) \\
\phi_j &\sim \text{normal}(\mu_{\phi_i}, \sigma_{\phi_i}^2) \quad \text{for} \quad j = 1, \ldots, p \\
\sigma^2 &\sim \text{igamma}(v, w) \\
y_k &\sim \text{normal}(\mu_{y_k}, \sigma_{y_k}^2) \quad \text{for} \quad k = 0, \ldots, 1 - p
\end{aligned}$$

Example 1: Simulated AR(4) Model

The following example fits an AR(4) model to a simulated time series with $c = 0, \phi_1 = 0.8, \phi_2 = -0.64, \phi_3 = 0.512, \phi_4 = -0.4096, \sigma^2 = 4$, and a sample size of 500. The statements that generate the data are not shown but are included in the sample SAS program file that accompanies this example. A copy of the response variable Y is generated and named X, and then the last 20 observations of Y are set to missing.

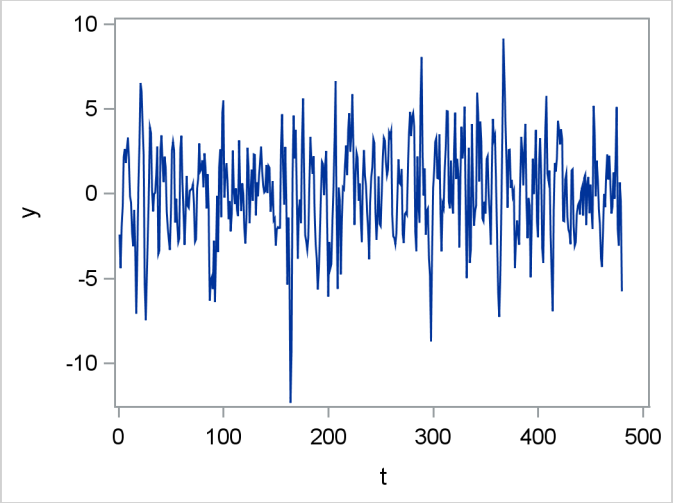
PROC MCMC automatically imputes values for the missing observations, thus producing an out-of-sample forecast for the response variable. Keeping a copy of the original data stored in X provides the means to assess the accuracy of the out-of-sample forecast.

The following statements generate a plot of the simulated time series Y by using the SGPLOT procedure:

```
ods graphics on;
proc sgplot data=ar4;
  series x=t y=y;
run;
```

Figure 1 shows the plot of the time series variable Y.

Figure 1 Simulated AR(4) Time Series



The following statements specify the AR(4) model by using PROC MCMC:

```
proc mcmc data=ar4 nmc=100000 seed=100 nthreads=8 propcov=quanew;
  parms phi_1 phi_2 phi_3 phi_4;
  parms sigma2 1;
  parms Y_0 Y_1 Y_2 Y_3;
  prior phi_:~normal(0,var=1000);
  prior sigma2 ~ igamma(shape = 3/10, scale = 10/3);
  prior Y_: ~ n(0, var=1000 );
  mu=phi_1*y.l1 + phi_2*y.l2 + phi_3*y.l3 + phi_4*y.l4;
  model y~normal(mu, var=sigma2)
        icond=(Y_3 Y_2 Y_1 Y_0);
  preddist outpred=AR4outpred statistics=brief;
  ods output PredSumInt=AR4PredSumInt;
run;
```

In the PROC MCMC statement, the NMC= option specifies the number of MCMC iterations, excluding the burn-in iterations; the NTHREADS= option specifies the number of threads to use; and the PROPCOV= option specifies the method to use in constructing the initial covariance matrix for the Metropolis-Hastings algorithm. The QUANEW method finds numerically approximated covariance matrices at the optimum of

the posterior density function with respect to all continuous parameters.

The first PARMS statement declares the parameters ϕ_1, ϕ_2, ϕ_3, and ϕ_4 and places them in a single parameter block. The second PARMS statement declares the parameter σ^2 and places it in a separate parameter block. The third PARMS statement specifies that the unobserved initial conditions y_0, y_{-1}, y_{-2}, and y_{-3} be treated as model parameters and places them in a separate parameter block.

The first PRIOR statement specifies normal prior distributions with a mean equal to 0 and a variance equal to 1,000 for the parameters ϕ_1, ϕ_2, ϕ_3, and ϕ_4. The second PRIOR statement specifies an inverse gamma prior distribution with shape equal to 3/10 and a scale equal to 10/3 for the parameter σ^2. The third PRIOR statement specifies normal prior distributions with a mean equal to 0 and a variance equal to 1,000 for the initial condition parameters y_0, y_{t-1}, y_{t-2}, and y_{t-3}.

The statement that follows is included strictly for convenience. It specifies μ, which is the mean of the response variable's distribution. The quantities y.l1, y.l2, y.l3, and y.l4 represent the first, second, third, and fourth lag of the response variable, respectively.

The MODEL statement specifies that the response variable has a normal distribution with mean μ and variance σ^2. The ICOND= option specifies the initial states of the lag variables for the response variable when the observation indices are out of the range.

In the PREDDIST statement, the OUTPRED= option creates a data set, AR4outpred. that contains random samples from the posterior predictive distribution of the response variable. The STATISTICS=BRIEF option species that posterior means, standard deviations, and the $100(1-\alpha)\%$ equal-tail intervals be computed. By default, the number of simulated predicted values is the same as the number of simulations that are specified in the NMC= option in the PROC MCMC statement. However, you can specify a different number of simulations by specifying the NSIM= option in the PREDDIST statement. Because the last 20 observations of the response variable are set to missing, PROC MCMC treats those observations as random variables and imputes their values in the predictive distribution. The result is an out-of-sample forecast for the response variable.

The ODS OUTPUT statement specifies that the information from the prediction summaries and intervals table be saved in the data set AR4PredSumInt.

Output 1 displays the posterior summaries and intervals table for the model. The 95% HPD intervals for the parameters ϕ_1, ϕ_2, ϕ_3, ϕ_4, and σ^2 all contain the true population values, and the means of their respective posterior distributions provide fairly accurate point estimates.

Output 1 Posterior Summaries and Intervals

The MCMC Procedure

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
phi_1	100000	0.7947	0.0522	0.6945	0.8980
phi_2	100000	-0.5210	0.0630	-0.6467	-0.3987
phi_3	100000	0.4331	0.0628	0.3100	0.5560
phi_4	100000	-0.3674	0.0514	-0.4663	-0.2662
sigma2	100000	4.5998	0.4842	3.7601	5.5631
Y_0	100000	0.9900	5.7539	-9.6331	12.9945
Y_1	100000	0.2228	8.7826	-17.2451	17.5142
Y_2	100000	4.8682	8.9749	-13.4494	22.2866
Y_3	100000	13.6260	9.6851	-5.5431	32.6751

Output 2 shows the effective sample sizes table. The autocorrelation times are high for all the parameters, and the effective sample sizes are small relative to the number of Monte Carlo samples. This evidence indicates slow mixing. However, the trace plots (not shown) provide no evidence that the Markov chain failed to converge.

Output 2 Effective Sample Sizes

The MCMC Procedure

Effective Sample Sizes

Parameter	ESS	Autocorrelation Time	Efficiency
phi_1	6136.4	16.2963	0.0614
phi_2	7259.9	13.7743	0.0726
phi_3	5961.9	16.7731	0.0596
phi_4	4849.7	20.6198	0.0485
sigma2	12298.8	8.1309	0.1230
Y_0	7265.8	13.7631	0.0727
Y_1	6758.8	14.7956	0.0676
Y_2	6487.0	15.4154	0.0649
Y_3	6319.7	15.8236	0.0632

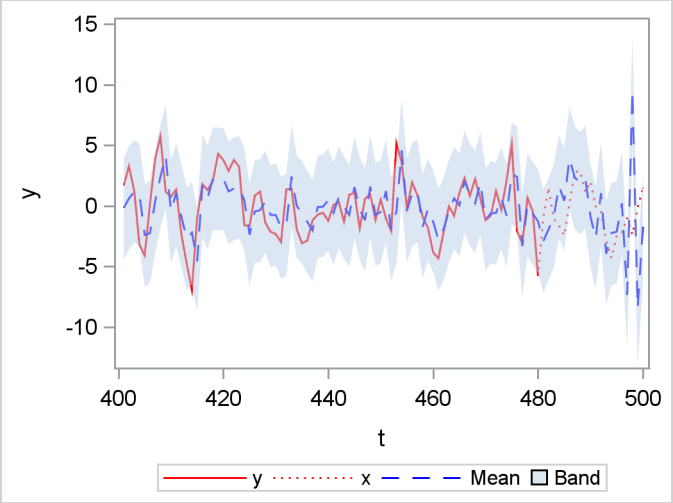
The following statements merge the original data set Ar4 with the ODS OUTPUT data set AR4PredSumInt and then use PROC SGPLOT to plot the original in-sample series, the in-sample prediction, the out-of-sample forecast, and a 95% HPD interval around the forecast. Only the last 100 observations are used in the plot to make it easier to visually assess the model fit and the accuracy of the forecast.

```
data ar4forecast;
  merge ar4 AR4PredSumInt;
run;
```

```
proc sgplot data=ar4forecast(where=(t>400));
  series x=t y=y / lineattrs=(color=red);
  series x=t y=x / lineattrs=(color=red pattern=dot);
  series x=t y=mean / lineattrs=(color=blue pattern=dash);
  band x=t upper=hpdupper lower=hpdlower / transparency=.5;
run;
```

Figure 2 shows the resulting plots. The solid red line represents the original in-sample response variable; the dotted red line represents the out-of-sample values of the response variable; the dashed blue line represents the mean of the predictive distribution; and the light blue shaded area represents the 95% HPD interval. As is typical of linear autoregressive models, the forecasts tend to lag at turning points, and the out-of-sample forecast accuracy diminishes as you forecast further into the future. Because the data are simulated with known population parameter values, model identification is not an issue. Thus, models such as this one provide a useful qualitative benchmark with which to compare the performance of models of "real world" data, where model identification is uncertain.

Figure 2 Forecast of Simulated AR(4) Time Series



Dynamic Linear Model with Time-Varying Coefficients

Dynamic linear models with time-varying coefficients enable you to model stochastic shifts in regression parameters. This is accomplished by using random-effects models that specify time dependence between successive parameter values in the form of smoothness priors. For example, the following describes a smoothing model that appears in Congdon (2003, p. 207) for a time series with seasonal effects and a secular trend:

$$
\begin{aligned}
y_t &\sim N(x_t, \theta_4) && \text{Observational model} \\
x_t &= \mu_t + s_t && \text{Underlying trend after allowing for seasonal effects} \\
\mu_t &\sim N(\mu_{t-1} + \alpha_{t-1}, \theta_1) && \text{Evolution in mean} \\
\alpha_{t-1} &\sim N(\phi\alpha_{t-2}, \theta_2) && \text{Increments in mean} \\
s_t &\sim N(-s_{t-1} - s_{t-2} - s_{t-3}, \theta_3) && \text{Seasonality}
\end{aligned}
$$

The first equation states that the response variable is normally distributed and the mean of the distribution is indexed by time, implying that the response variable is nonstationary. The second equation states that the mean of the response variable consists of a trend plus seasonal effects. The third equation states that the trend follows a random walk plus drift. The fourth equation states that the drift term follows a first-order autoregressive process. The fifth equation states that the seasonal effect is a normally distributed random shock whose mean equals the negated sum of the three previous shocks.

To perform a Bayesian analysis, you assign prior distributions to the unknown quantities α_0 and μ_0, which are the initial states of α_t and μ_t, respectively; s_1, s_2, and s_3, which are the initial states of s_t; θ_1, θ_2, θ_3, and θ_4, which are the unknown variances of μ_t, α_t, s_t, and y_t, respectively; ϕ, which is the first-order autoregressive parameter that modifies α_{t-2} in the mean of α_{t-1}; and θ_ϕ, which is the variance of α_{t-1}. The following prior distributions are assigned to these quantities in the example that follows:

$$
\begin{aligned}
\alpha_0 &\sim \text{normal}(\mu_\alpha, \theta_2) \\
\mu_0 &\sim \text{normal}(\mu_0, \sigma^2) \\
s_q &\sim \text{normal}(\mu_{s_q}, \theta_3) \quad \text{for} \quad q = 1, \ldots, 3 \\
\theta_j &\sim \text{IG}(\text{shape}=a_j, \text{scale}=b_j) \quad \text{for} \quad j = 1, \ldots, 4 \\
\phi &\sim \text{normal}(\mu_\phi, \theta_\phi) \\
\theta_\phi &\sim \text{IG}(\text{shape}=a_\phi, \text{scale}=b_\phi)
\end{aligned}
$$

Example 2: UK Coal Consumption

This example fits a time-varying coefficients model to observed quarterly coal consumption in the United Kingdom for the years 1960–1986. It is based on an example that is published in Congdon (2003, p. 207), which in turn is based on a model and data that are published in Harvey (1989). The following statements create the data set UKcoal, which contains the variables Coal, C, Year, Quarter, and T. The variable Coal measures quarterly coal consumption. C is a variance-stabilizing transformation of Coal; specifically, it is the natural logarithm of Coal divided by 1,000. The variables Year and Quarter represent the years and quarters of the observations. The variable T indexes the observations.

The following statements plot the original coal consumption series and the transformed series:

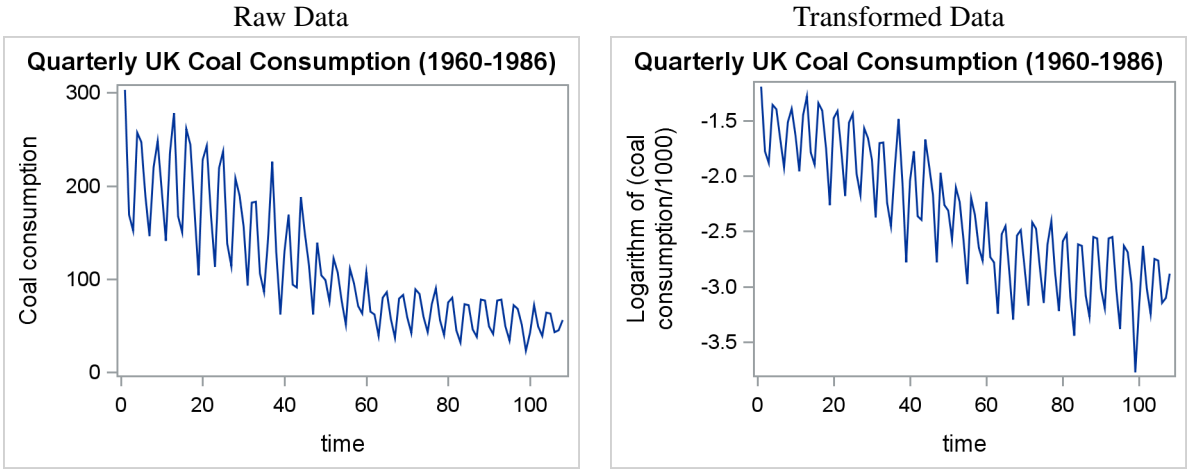
```
proc sgplot data=UKcoal;
  title "Quarterly UK Coal Consumption (1960-1986)";
  series y=coal x=t;
```

```
  yaxis label="Coal consumption";
  xaxis label="time";
run;
title;


proc sgplot data=UKcoal;
  title "Quarterly UK Coal Consumption (1960–1986)";
  series y=c x=t;
  yaxis label="Logarithm of (coal consumption/1000)";
  xaxis label="time";
run;
title;
```

Figure 3 displays the original coal consumption time series on the left and the transformed coal consumption time series on the right. It appears that the logarithmic transformation has successfully made the variance of the series more homogeneous.

Figure 3 Quarterly UK Coal Consumption (1960–1986)



The following DATA step creates the variable Z, which is just a copy of C. After Z is created, Coal is set to missing for the years 1985–1986 to enable you to assess the accuracy of the model's out-of-sample forecast.

```
data UKcoal;
  set UKcoal;
  z=c;
  if year>1984 then c=.;
run;
```

The following statements use PROC MCMC to fit the time-varying coefficients model:

```
proc mcmc data=UKcoal nmc=100000 seed=123456 outpost=posterior propcov=quanew;
  parms alpha0;
  parms mu0;
  parms s0 s1 s2;
  parms theta1;
  parms theta2;
  parms theta3;
```

```
  parms theta4;
  parms theta_phi;
  parms phi;
  prior phi~normal(0,var=exp(theta_phi));
  prior alpha0~normal(0,var=theta2);
  prior mu0~normal(0,var=100);
  prior s:~normal(0,var=theta3);
  prior theta:~igamma(shape = 3/10, scale = 10/3);
  random alpha~normal(phi*alpha.l1,var=exp(theta2)) subject=t icond=(alpha0);
  random s~normal(-s.l1-s.l2-s.l3,var=exp(theta3)) subject=quarter icond=(s2 s1 s0);
  random mu~normal(mu.l1 + alpha.l1,var=exp(theta1)) subject=t icond=(mu0);
  x=mu + s;
  model c~normal(x,var=exp(theta4));
  preddist outpred=TVCoutpred statistics=brief;
  ods output PredSumInt=TVCPredSumInt;
run;
```

In the PROC MCMC statement, the NMC= option specifies the number of MCMC iterations, excluding the burn-in iterations; the PROPCOV= option specifies the method to use in constructing the initial covariance matrix for the Metropolis-Hastings algorithm. The QUANEW method finds numerically approximated covariance matrices at the optimum of the posterior density function with respect to all continuous parameters.

The nine PARMS statements declare the model parameters. The three seasonal parameters s_0, s_1, and s_2 are placed in the same block; experimentation with this model and these data showed that putting all the other parameters in separate blocks improved the mixing.

The first PRIOR statement specifies a normal prior distribution for the parameter ϕ, with mean equal to 0 and variance equal to $\exp(\theta_\phi)$. Experimentation with this model and these data showed that sampling the parameter θ_ϕ on the logarithmic scale significantly improved mixing. The second PRIOR statement specifies a normal prior distribution for the parameter α_0, with mean equal to 0 and variance equal to the parameter θ_2; α_0 represents the initial state of the first-order autoregressive drift component. The third PRIOR statement specifies a normal prior distribution for the parameter μ_0, with mean equal to 0 and variance equal to 1,000; μ_0 represents the initial state for μ_t. The fourth PRIOR statement specifies normal prior distributions for s_0, s_1, and s_2, with means equal to 0 and variances equal to the parameter θ_3; s_0, s_1, and s_2 represent the initial states for the seasonal random effect s_t. The fifth PRIOR statement specifies inverse gamma distributions for the variance parameters θ_1, θ_2, θ_3, θ_4, and θ_ϕ. All these distributions have shape and scale parameters specified to be 3/10 and 10/3, respectively.

The first RANDOM statement specifies that the random effect α_t has a normal prior distribution, with a mean equal to the product of ϕ and the first lag of α_t and a variance equal to $\exp(\theta_2)$. Experimentation with this model and these data showed that sampling θ_2 on the logarithmic scale significantly improved mixing. The SUBJECT= option specifies that the random effect α_t is indexed by the variable T. The ICOND= option specifies that the initial state of α_t is represented by the parameter α_0.

The second RANDOM statement specifies that the seasonal random effect s_t has a normal prior distribution, with a mean equal to $-(s_{t-1} + s_{t-2} + s_{t-3})$ and a variance equal to $\exp(\theta_3)$. Experimentation with this model and these data showed that sampling θ_3 on the logarithmic scale significantly improved mixing. The SUBJECT= option specifies that the random effect s_t is indexed by the variable Quarter. The ICOND= option specifies that the initial states of s_t are represented by the parameters s_0, s_1, and s_3.

The third RANDOM statement specifies that the trend random effect μ_t has a normal prior distribution, with a mean equal to $\mu_{t-1} + \alpha_{t-1}$ and a variance equal to $\exp(\theta_1)$. Experimentation with this model and these data showed that sampling θ_1 on the logarithmic scale significantly improved mixing. The SUBJECT=

option specifies that the random effect μ_t is indexed by the variable T. The ICOND= option specifies that the initial state of μ_t is represented by the parameter μ_0.

The next statement is included primarily for convenience. It specifies that x_t, the mean of the response variable's distribution, is equal to $\mu_t + s_t$.

The MODEL statement specifies that the response variable has a normal distribution, with a mean equal to x_t and a variance equal to $\exp(\theta_4)$. Experimentation with this model and these data showed that sampling θ_4 on the logarithmic scale significantly improved mixing.

The PREDDIST statement specifies that the random samples from the predictive distribution be saved in the data set TVCoutpred. The STATISTICS=BRIEF option specifies that the posterior means, standard deviations, and $100(1-\alpha)\%$ equal-tail intervals be computed. The NSIM= option is omitted, so the number of iterations defaults to the number that is specified in the NMC= option in the PROC MCMC statement.

The ODS OUTPUT statement specifies that the information from the prediction summaries and intervals table be saved in the data set TVCPredSumInt.

Output 3 shows the posterior summaries and intervals table for the time-varying coefficients model, and Output 4 shows the effective sample sizes. The autocorrelation times are relatively large, indicating slow mixing. This is the reason that the number of MCMC iterations was set to the relatively large value of 100,000.

Output 3 Posterior Summaries and Intervals

The MCMC Procedure

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
alpha0	100000	-0.00141	0.6895	-1.3397	1.3774
mu0	100000	-1.4996	1.8070	-4.8613	2.2364
s0	100000	-0.0202	1.0511	-2.1555	2.0876
s1	100000	-0.0896	1.0811	-2.2344	2.1274
s2	100000	-0.0536	1.1248	-2.3323	2.1594
theta1	100000	0.4690	0.1156	0.2620	0.7038
theta2	100000	0.4917	0.1223	0.2660	0.7339
theta3	100000	1.5701	0.7386	0.4389	3.0281
theta4	100000	0.4234	0.1012	0.2390	0.6230
theta_phi	100000	2.7582	1.6956	0.5189	6.1682
phi	100000	-0.3280	0.1954	-0.7009	0.0557

Output 4 Effective Sample Size

The MCMC Procedure

Effective Sample Sizes

Parameter	ESS	Autocorrelation Time	Efficiency
alpha0	8829.0	11.3263	0.0883
mu0	2635.0	37.9511	0.0263
s0	2256.6	44.3139	0.0226
s1	8318.4	12.0215	0.0832
s2	7972.2	12.5436	0.0797
theta1	10565.8	9.4645	0.1057
theta2	5391.2	18.5489	0.0539
theta3	1696.3	58.9535	0.0170
theta4	9771.8	10.2336	0.0977
theta_phi	10643.7	9.3952	0.1064
phi	2136.4	46.8074	0.0214

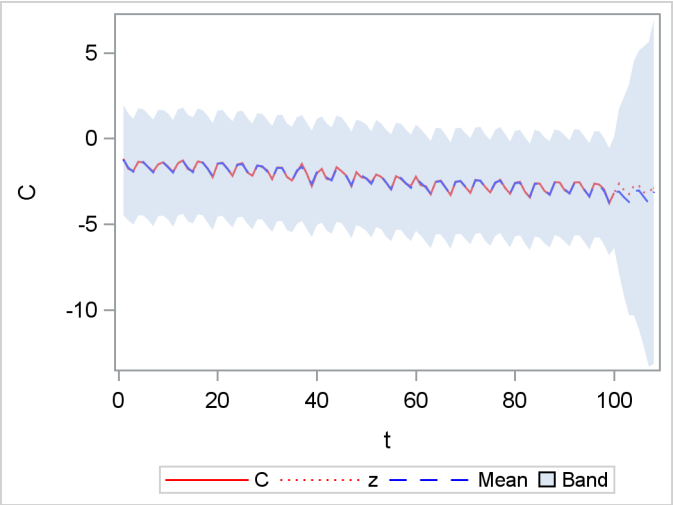
he following statements merge the original data set, UKcoal, with the ODS OUTPUT data set TVCPredSumInt and then use PROC SGPLOT to plot the original in-sample series, the in-sample prediction, the out-of-sample forecast, and a 95% HPD interval around the forecast:

```
data forecast;
  merge UKcoal TVCPredSumInt;
run;

proc sgplot data=forecast;
  series x=t y=c / lineattrs=(color=red);
  series x=t y=z / lineattrs=(color=red pattern=dot);
  series x=t y=mean / lineattrs=(color=blue pattern=dash);
  band x=t upper=hpdupper lower=hpdlower / transparency=.5;
run;
```

Figure 4 shows the resulting plots. The solid red line represents the original in-sample response variable; the dotted red line represents the out-of-sample values of the response variable; the dashed blue line represents the mean of the predictive distribution; and the light blue shaded area represents the 95% HPD interval. The in-sample model prediction appears to fit the data. The out-of-sample forecast likewise appears to perform well. The 95% HPD interval around the out-of-sample forecast grows very quickly, indicating, not unexpectedly, that the further out of sample you forecast, the greater the uncertainty of the forecast.

Figure 4 Forecast of Quarterly UK Coal Consumption (1960–1986)



References

Congdon, P. (2003). *Applied Bayesian Modeling*. New York: John Wiley & Sons.

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