The RATSletter

Volume 11, No. 1 **December, 1998**

Estima

RATS 4.3 for Macintosh

The wait is finally over! Version 4.3 of RATS for Macintosh will begin shipping in February. There are two flavors to choose from: **MacRATS '020** for 68020, 68030, and 68040 systems, and **MacRATS PPC** for PowerMac systems. The standard MacRATS product (for 68000 systems) has been discontinued.

The big news here is, of course, the availability of a native PowerMac version of RATS, which provides exceptional performance on PowerMac systems. However, both products also include many improvements and new features that were previously available only in our UNIX and PC versions (see page 4 for a brief summary). Both packages also include a new version of the RATSDATA utility program.

Upgrade to Power Mac Version for just $100!

If you have a PowerMac, you'll want to upgrade to the new MacRATS PPC. This native PowerMac application runs *very* quickly—depending on the specific task, it runs anywhere from 2.5 to 10 times faster than standard MacRATS 4.0 on a given PowerMac system. Users with Version 4.0 of either MacRATS or MacRATS '020 can upgrade to MacRATS PPC for just $100 (plus shipping outside the US). Users with Version 3 of MacRATS can upgrade for just $170.

Free Updates to MacRATS '020 4.3

Licensed users of MacRATS 4.0 and MacRATS '020 4.0 are entitled to a free update to MacRATS '020 Version 4.3. You can order by phone, fax, mail, e-mail, or via our web site. When ordering, please supply your name, address, phone number, and RATS serial number. The free update offer will be available until the release of Version 5 later this year.

Note that MacRATS '020 is designed for 68020, 68030, and 68040 systems with the appropriate math co-processor. It will run on PowerMac systems, but only if you install a floating point emulator such as SoftFPU or PowerFPU. Even with an FPU emulator, the program will run much slower than MacRATS PPC on such systems, so we strongly encourage PowerMac users to upgrade to MacRATS PPC.

CATS Cointegration Analysis for Macintosh

MacRATS users who update to 4.3 can also take advantage of our popular CATS cointegration analysis package. CATS is a powerful, menu-driven procedure written in the RATS language by Soren Johansen, Katarina Juselius, and Henrik Hansen. It can handle a variety of different models and hypothesis tests, and includes a detailed user's manual.

The price for a single-user CATS license is just $100. Please contact Estima for quotes on multi-user licenses.

Markov Switching Models

Markov switching models for time series have become popular lately, largely as a result of the work of James Hamilton (See Chapter 22 of his book *Time Series Analysis* from Princeton University Press). Some of these models can be estimated in RATS using the **MAXIMIZE** instruction.

To start, let's look at a simple Markov chain. Suppose that there are two regimes. The probability of moving from state 1 to state 2 is `p12`, and for moving from state 2 to state 1 is `p21`, both unknown. In this simple case, the regime is known at each time period. Let the state be shown by the dummy variable `REGIME` which will be 1 for state 1 and 0 for state 2.

This is a simple enough situation that the maximum likelihood estimate of `p12` is just the fraction of the times that the system is in state 1 and moves to state 2. However, we'll illustrate how to set this up with **MAXIMIZE**, as more complex examples require only minor modifications. Complete examples will be available on the "Procedures and Examples" page of our web site, at www.estima.com.

```
nonlin p12 p21
frml markov = $
 rp1 = p21*(1-regime{1})+(1-p12)*regime{1},$
 rp2 = (1-p21)*(1-regime{1})+p12*regime{1},$
 pstar = rp1/(rp1+rp2), $
 %if(regime,log(pstar), log(1-pstar))
```

The `MARKOV` formula is then estimated using **MAXIMIZE**. Note that `pstar` is the probability of being in state 1. In this case, it is trivially either `p21` or `1-p12`. In the full examples, `pstar` is put into a data series.

Now we will look at a situation where the regime is independent across time and unknown, but where an observable variable y is governed by different processes in the two regimes:

In regime 1, $y(t) = \text{a01} + \text{a11} * x_1(t) + u_1(t)$

In regime 2, $y(t) = \text{a02} + \text{a12} * x_2(t) + u_2(t)$

(Continued on Page 4)

Contents

Gibbs Sampler and Other Bayesian Techniques

Gibbs Sampler

RATS was at the forefront in the use of modern Bayesian techniques in econometrics. Perhaps the first widely used application of computationally intensive Bayesian methods in the profession was the Monte Carlo integration procedure for computing posterior moments of impulse responses.

The Gibbs sampler[1] is one of several techniques developed recently to deal with posterior distributions which not long ago were considered to be intractable. Monte Carlo integration for impulse responses is able to work well despite the large number of parameters in the underlying vector autoregression because, with the help of a convenient choice of prior, it is fairly easy to make draws from the posterior distribution. Unfortunately, there are very few multivariate distributions for which this is true. Even in a simple linear regression model, all it takes is a slight deviation from "convenience" in the prior to produce a tangled mess in the posterior distribution, making direct draws with the basic toolkit of random uniforms and Normals impossible.

The Gibbs sampler can be brought into play when the parameters can be partitioned such that, although an unconditional draw can't be obtained directly, each partition can be drawn conditional on the parameters outside its partition. The standard result is that if we sequentially draw from the conditional distributions, the resulting draws are, in the limit, from the unconditional distribution. Because this is a limit result, it is common for practitioners to ignore a certain number of early draws (called the "burn-in") which might not be representative of the unconditional distribution.

The most common blocking for the Gibbs sampler is between the regression parameters and the variance or precision. For instance, in the Normal linear model

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{u},\ \mathbf{u}\,|\mathbf{X} \sim N\left(0, h^{-1}\mathbf{I}\right)$$

with prior $\beta \sim N\left(\beta_p, H^{-1}\right)$ and $\nu\sigma^2 h$ distributed chi-square with ν degrees of freedom, the posterior density function is proportional to:

$$h^{\frac{T+\nu}{2}-1} e^{\frac{-h}{2}\left(\mathbf{y}-\mathbf{X}\beta\right)'\left(\mathbf{y}-\mathbf{X}\beta\right)} e^{\frac{-1}{2}\left(\beta-\beta_p\right)H\left(\beta-\beta_p\right)'} e^{\frac{-1}{2}\nu\sigma^2 h}$$

then, conditional on h, $\mathbf{y}$ and $\mathbf{X}$, β is Normal with precision $\mathbf{H}^* = \mathbf{H} + h\mathbf{X}'\mathbf{X}$ and mean $\mathbf{H}^{*-1}\left(\mathbf{H}\beta_p + h\mathbf{X}'\mathbf{X}\mathbf{b}\right)$ where $\mathbf{b}$ is the OLS estimate, and, conditional on β, $\mathbf{y}$ and $\mathbf{X}$,

$$\left(\sigma^2\nu + \left(\mathbf{y}-\mathbf{X}\beta\right)'\left(\mathbf{y}-\mathbf{X}\beta\right)\right)h \ \sim\ \chi^2\left(T+\nu\right)$$

Both of the conditional densities are from families from which draws can be generated easily. To implement this efficiently in RATS, we need to rewrite pieces of this and pump some information out of an OLS regression:

1. $\mathbf{X}'\mathbf{X}\mathbf{b}$ and $\mathbf{H}\beta_p$ are fixed, and can be calculated in advance.
2. $\left(\mathbf{y}-\mathbf{X}\beta\right)'\left(\mathbf{y}-\mathbf{X}\beta\right) = \left(\mathbf{y}-\mathbf{X}\mathbf{b}\right)'\left(\mathbf{y}-\mathbf{X}\mathbf{b}\right) + \left(\mathbf{b}-\beta\right)\mathbf{X}'\mathbf{X}\left(\mathbf{b}-\beta\right) = \mathbf{\%RSS} + \left(\mathbf{b}-\beta\right)\mathbf{X}'\mathbf{X}\left(\mathbf{b}-\beta\right)$

The following define the prior:

- `s2` is the scale parameter and `nu` the degrees of freedom of the prior precision.
- `bprior` and `hprior` are the prior mean and precision for the coefficients.

After the **LINREG**

```
compute rssols  = %rss
compute betaols = %beta
compute xxols   = inv(%xx)
compute xyols   = xxols*betaols
compute xyprior = hprior*bprior
compute sdof    = nu + %nobs
compute beta    = betaols
dec vect u(%nreg)
```

A single pass of the Gibbs sampler for this is

```
* Draw h conditional on current beta
compute center = nu*s2+rssols + $
                 %qform(xxols,beta-betaols)
compute hu = 2.0*%rangamma(sdof*.5)/center
* Draw beta conditional on h
compute hbeta =hu*xxols+hprior
compute vbeta =inv(hbeta)
compute ubeta =%decomp(vbeta)*(u=%ran(1.0))
compute beta  =vbeta*(hu*xyols+xybeta)+ubeta
```

Rejection Method

The Gibbs sampler still requires an ability to generate draws from the conditional distributions. In our example above, this wasn't difficult, because it required only draws from Normal and gamma distributions. It is sometimes necessary to generate draws from a distribution which can't be derived as simple functions of these elementary distributions. One method which can be used in many univariate cases is the Rejection method.

The Rejection (or Acceptance-Rejection) method is actually used within RATS to generate draws from the Normal and gamma distributions. This method requires little more than the ability to compute the density function itself.

We'll look first at a trivial example. Suppose that you need draws from a Normal(μ,σ^2) truncated to the interval [a,b]. An obvious way to do this is to draw Normal(μ,σ^2) deviates, then reject any that fall outside [a,b]. The accepted draws would have the desired distribution, but this process will be fairly inefficient if the probability of the Normal draw falling in [a,b] is fairly low.

(Continued on Page 4)

New Features in MacRATS 4.3

MacRATS 4.3 includes many new features previously available only on the PC and UNIX platforms. These include:

- New instructions for fitting neural network models, and an instruction for solving linear and quadratic programming problems.
- Many graphics improvements, including the ability to place text strings anywhere on a graph, more placement options for keys, more labeling options on **SPGRAPH**, simplified handling of two-scale graphs, and more.
- New instructions including **USERMENU** and **INFOBOX**, which expand the ability to write interactive, menu-driven, programs in RATS. These features are used extensively in the CATS cointegration package, which is now available for Macintosh systems.
- More than a dozen new functions, including functions for extracting elements from an array, extracting characters from a string, and computing inverse Normal and chi-squared distributions.

Year 2000 Update

As previously noted, RATS versions 4.0 and later are fully Year 2000 compliant. Be aware, however, that RATS does assume that 2-digit years refer to the 1900's. Thus, you must be sure to use 4-digit year references in any programs *and* data sets that involve data extending before 1900 or after 1999.

Also, please be aware that the Holiday adjustments available as an option in the EZ-X11 seasonal adjustment program are *not* Year 2000 compliant. These adjustments are based on tables used in the original Census X11 program, and are only valid through 1999. The X11 add-on module available for the UNIX and Windows versions of RATS has been updated to be Year 2000 compliant (contact Estima if you want to verify that your version of the X11 module is Y2K compliant). Stay tuned for news on an EZ-X11 update.

Order Via the Web

The on-line order form on our web site provides a fast and easy way to order RATS, CATS, and other Estima products. We've recently upgraded the site to allow you to order up to 5 new products in a single order. You can also include one update/upgrade order on the same form.

The system runs on a secure web server to keep transactions safe. And, if your browser supports JavaScript (most do), the order system will automatically compute and display information such as update prices and shipping costs. To order, just go to www.estima.com and click on "Order via the Web".

RATS Version 4.3 Prices

Contact Estima for prices on UNIX and Linux versions, network licenses, upgrade paths not shown below, or any other questions.

Update/Upgrade:	To:	Price:
MacRATS '020 ver 4.0	MacRATS PPC 4.3	$100.00
MacRATS '020 ver 4.0	MacRATS '020 4.3	No Charge
MacRATS ver. 4.0	MacRATS PPC 4.3	$100.00
MacRATS ver. 4.0	MacRATS '020 4.3	No Charge
MacRATS '020 ver 3.xx	MacRATS PPC 4.3	$170.00
MacRATS '020 ver 3.xx	MacRATS '020 4.3	$70.00
MacRATS ver 3.xx	MacRATS PPC 4.3	$170.00
MacRATS ver 3.xx	MacRATS '020 4.3	$70.00
WinRATS-32 4.2x	WinRATS-32 4.3	$25.00
WinRATS Plus 4.2x	WinRATS-32 4.3	$50.00
WinRATS Plus 4.2x	WinRATS Plus 4.3	$25.00
WinRATS 4.2x	WinRATS-32 4.3	$100.00
WinRATS 4.2x	WinRATS 4.3	$25.00
RATS386 4.x	WinRATS-32 4.3	$80.00
RATS386 4.x	WinRATS Plus 4.3	$50.00
RATS386 4.2x	RATS386 4.3	$25.00
RATS386 4.0,4.1	RATS386 4.3	$40.00
PC RATS 4.x	WinRATS-32 4.3	$200.00
PC RATS 4.x	WinRATS Plus 4.3	$150.00
PC RATS 4.x	WinRATS 4.3	$100.00
PC RATS 4.x	RATS386 4.3	$120.00

Prices for New Copies of RATS and CATS

MacRATS PPC	$500.00	WinRATS-32	$500.00
MacRATS '020	$400.00	WinRATS Plus	$450.00
RATS386	$420.00	WinRATS	$400.00
CATS (PC or Mac)	$100.00		

Walter Enders' RATS Handbook and Textbook

Walt Enders' popular *Applied Economic Time Series* textbook and the accompanying RATS Handbook are still available directly from Estima. The handbook is $30 ($27.50 when ordered with another Estima product). The textbook is $65.

Tax and Shipping Costs

Illinois customers add 8.5% sales tax. Prices include UPS Ground shipping. Additional charges apply for 2nd Day and Next Day Air shipments, and shipments outside the US.

Dial-Up Bulletin Board is History

Our aging BBS host system recently suffered a catastrophic failure. With the advent of the Web and internet e-mail, traffic on the BBS had been practically nonexistent for many months. And so, with a touch of melancholy, we've decided not to bring the BBS up on another system. Thanks to all who made use of the BBS over the years, and we hope that you find our web site and the RATS mailing list to be suitable replacements.

Markov (continued from page 1)

Assume that, conditional on x_1 and x_2, u_1 and u_2 are distributed Normal$\left(0,\sigma^2\right)$ independently both contemporaneously and across time.

Let p be the (unknown) unconditional probability of the process being in state 1 at any time period. The likelihood comes from the formula

$$f\left(y \mid parameters\right) = p * f\left(y \mid parameters, \text{ state 1}\right) + \left(1-p\right) * f\left(y \mid parameters, \text{ state 2}\right)$$

Bayes' formula gives the post-data probability of being in state 1 as

$$\frac{p * f\left(y \mid parameters, \text{ state 1}\right)}{f\left(y \mid parameters\right)}$$

A (set of) formulas to estimate this model is:

```
nonlin p a01 a02 a11 a12 sigma
frml reg1 = y-a01-a11*x1
frml reg2 = y-a02-a12*x2
frml mixing = $
    f1=%density(reg1{0}/sigma)/sigma , $
    f2=%density(reg2{0}/sigma)/sigma , $
    rp1=f1*p, rp2=f2*(1-p) , $
    pstar=rp1/(rp1+rp2) , log(rp1+rp2)
```

We now combine the two types of models. The state may shift from period to period following the Markov model from the first example. If p* is our estimate of the probability of being in state 1 based upon data through t, then the (pre-data) probability of state 1 in period t+1 is

$$\text{p21}\left(1-p^*\right)+\left(1-\text{p12}\right)p^*$$

The set up for this model is:

```
nonlin p12 p21 a01 a02 a11 a12 sigma
frml markov = $
 f1  = %density(reg1{0}/sigma)/sigma, $
 f2  = %density(reg2{0}/sigma)/sigma, $
 rp1 = f1*(p21*(1-pstar{1})+ $
         (1-p12)*pstar{1}), $
 rp2 = f2*((1-p21)*(1-pstar{1})+ $
         p12*pstar{1}), $
 pstar = rp1/(rp1+rp2) , $
 log(rp1+rp2)
```

It's possible to extend these to three or four states with some help from matrix operations. The worked example in Hamilton's book expands out to 32 states and is beyond the realistic capabilities of the current version of RATS. We're expecting a future release to add some new features to cure this.

Visit Us at the ASSA Show

We invite you to visit us in booth number 158 at this year's ASSA show in New York. We will be demonstrating the new versions of MacRATS, along with many other products. We hope you'll stop by and say hello.

Gibbs Sampler (continued from page 2)

An alternative is to draw a random number x from Uniform(a,b). Also draw a random number z from U(0,1). Compare z with the ratio between $f_N\left(x \mid \mu,\sigma^2\right)$ and the maximum that $f_N\left(\bullet \mid \mu,\sigma^2\right)$ achieves on [a,b]. If z is less, accept x; if not, reject it. This trims away the uniform draws to match the shape of the Normal. This will be more efficient if the density function is fairly constant on [a,b], so that the comparison ratio is close to one and almost all draws are accepted.

In general, in order to generate draws from a distribution with (bounded) density f, it is necessary to have a comparison distribution with density g from which draws can be generated more easily and which is non-zero wherever f is non-zero. In fact, we don't even need a complete density function for f: its kernel is enough. In our first "method" for drawing from the truncated Normal, g was the same as f, just not truncated. In the second, g is $I_{[a,b]}$ (we can also work with just the kernel of g).

A key value which we can compute in advance of making our draws is $c = 1/\max\left(f/g\right)$ where the max is computed over the support of f. In method one, this is just 1. In method two, it is the reciprocal of the maximum that f_N achieves on [a,b]. Given the draw x from the density g, accept with probability $c\,f(x)/g(x)$.

In method one, $f(x)/g(x)$ is 1 on [a,b] and 0 outside of it, allowing us to avoid the need for a second draw. For the second, this works out to $f_N\left(x|\ldots\right)$ over the max, just as we indicated.

What is most important for this to be done efficiently is that g have "tails" at least as fat as the target density f. For instance, the Cauchy is a convenient comparison distribution for many unimodal distributions on the real line. When we draw from the Cauchy, we get a small percentage of very extreme draws. However, if f is a thin-tailed distribution, f/g will be very small out there, and the extreme draws will be rejected. In other words, we thin out the Cauchy tails by rejecting most tail draws. If, however, f/g is quite large in the tails, the only way we can "thicken" the tails is by accepting the tail draws and rejecting most of the draws near the mode.

Rejection Method Example

We cited above a couple of simple ways to draw from a truncated Normal. Each method has intervals over which it would be highly inefficient. And the use of a Uniform as the comparison function isn't even possible unless the interval is truncated at both ends. We show here how to draw truncated Normals using a truncated logistic as the comparison distribution.

A logistic with location parameter a and scale parameter b has distribution function

(Continued on Page 5)

Gibbs Sampler (continued from page 4)

$$G(x;a,b) = 1.0 \Big/ \left(1+e^{-(x-a)/b}\right)$$

and density function

$$g(x;a,b) = 1+e^{-(x-a)/b} \Big/ b\left(1+e^{-(x-a)/b}\right)^2$$

It has a shape very similar to the Normal, but has an invertible distribution function. This is particularly convenient here because drawing a truncated version of it requires only drawing a truncated Uniform and inverting. If we're looking for efficient draws, what choice should we make for the parameter *b*? A result in Geweke[2] shows that the optimal choice is the one which minimizes the maximum ratio between the Normal density and logistic. This is $b = .648$, which produces a maximum ratio at $x = 1.0$.

The core of the rejection method is extremely simple to program and is just a few lines long. Most of the work is the "human time" needed to figure out a good comparison function and to calculate the factor *c*. As with any such exercise, you need to decide how much to substitute raw computing power for ingenuity. With the speed of today's computers, a rejection algorithm which you can put into operation in a few minutes, but which rejects 90% of draws, is likely to get your work done faster than a finely tuned algorithm which rejects 10% of draws but takes a day's effort to get ready.

If you can't determine the value of *c* using calculus, the RATS function **FIND** can come in handy. For instance, to calculate *c* for the Normal versus logistic, we did the following:

```
nonlin xm
compute sfac=.647
compute xm=0.0,value=0.0
find max value
 compute value = -.5*xm**2+xm/sfac $
   +log(sfac)+2*log(1+exp(-xm/sfac))
end find
disp value
```

This finds the maximum of $\log(f/g)$. By varying `sfac`, we were able to find the minimizing value for the logistic scale parameter.

Importance Sampling

An idea similar to the rejection method is known as importance sampling. Both use a convenient density which (one hopes) has a similar shape to the true density. In importance sampling, this density is called the importance function. The difference is that the rejection method generates draws from the true density by rejecting some, while importance sampling uses all the draws generated but weights the draws when computing sample moments. If *f* is the true density and *g* is the importance function, importance sampling is based upon the fact that

$$\begin{aligned} E(h(x)) &= \int h(x) f(x)\, dx \\ &= \int h(x)\left(f(x)/g(x)\right) g(x)\, dx \\ &= \int h(x)\, w(x)\, g(x)\, dx \end{aligned}$$

where the weight function $w(x) = f(x)/g(x)$.

If $x_1,...,X_N$ are draws from the density *g*, the Monte Carlo estimate of $E(h(x))$ is

$$\frac{(1/N)\sum h(x_i) w(x_i)}{(1/N)\sum w(x_i)}$$

In cases where the function(s) whose moments are being computed are very complicated functions of the distribution from which you are drawing, importance sampling has the drawback that you must go through that computation for every draw, no matter how low its weight.

For instance, when we were first developing the Monte Carlo integration procedure for impulse responses (see example 10.1 in the RATS manual), we used importance sampling for the VAR coefficients. For each draw, it is necessary to compute impulse responses and tabulate the first and second moments. That's quite a bit of work when many of the draws had weights less than .10.

---

[1] Gelfand and Smith (1990). "Sampling-Based Approaches to Calculating Marginal Densities", *JASA*, pp. 398-409

[2] Geweke (1992). "Priors for Macroeconomic Time Series and Their Application", Institute for Empirical Macroeconomics Discussion Paper No. 64, Federal Reserve Bank of Minneapolis.

Basic Random Draw Toolkit

For many distributions, you can take draws without resorting to the relatively complex techniques described above. Here, we describe the very useful inverse and direct methods, and techniques for taking draws from multivariate distributions.

Inverse Method

Random variables with an invertible cumulative distribution function are easy to draw by just applying the inverse to a uniform (0,1) draw. For instance, an exponential with parameter (λ) has distribution function $1-e^{(-x/\lambda)}$, $x \geq 0$. If *y* is a U(0,1) draw, then solve $y = 1-e^{(-x/\lambda)}$ to get the draw $x = -\lambda \log(1-y)$. Note that $(1-y)$ can be replaced with *y* because $(1-y)$ is also U(0,1).

The following code generates a vector of 100 draws from an exponential with mean `L`:

```
DEC VECT DRAWS(100)
EWISE DRAWS(I)=-L*LOG(%UNIFORM(0,1))
```

(Continued on Page 6)

Random Draw Tookit (continued from page 5)

and this generates Cauchy deviates:

```
EWISE DRAWS(I)=%TAN(%PI/2.0*%UNIFORM(-1.0,1.0))
```

Direct Method

Many random variables can be obtained by some function of Normal, uniform and gamma random variables, as shown below.

Chi-Squared with NU degrees of freedom:

```
EWISE DRAWS(I) = 2.0*%RANGAMMA(0.5*NU)
```

For *t* (ratio of Normal to chi-square with NU degrees of freedom divided by its degrees of freedom):

```
EWISE DRAWS(I) = 0.5*NU*%RAN(1.0)/ $
                 %RANGAMMA(0.5*NU)
```

Beta(α,β), using the convention that the kernel is $p^{(\alpha-1)}(1-p)^{(\beta-1)}$:

```
EWISE DRAWS(I) = 1.0/(1+%RANGAMMA(B)/ $
                 %RANGAMMA(A))
```

Note that if you try to write this as

```
%RANGAMMA(A)/(%RANGAMMA(A)+%RANGAMMA(B))
```

you'll be in for a bit of a shock, because the two values of %RANGAMMA(A) will be different (RATS takes a separate draw for each function call).

Multivariate Distributions

To draw a multivariate Normal with mean vector **X** and covariance matrix **S**, it is necessary first to get a factor of the **S** matrix: a matrix **P** such that $\mathbf{PP}' = \mathbf{S}$. Any such factor will do. The simplest one to obtain with RATS is the Choleski factor, which you can compute using the function %DECOMP.

The following code draws a single vector (UDRAW) from this distribution. If, as is typical, the actual draws would be inside a loop, the first three instruction lines can be outside of it:

```
DEC RECT P
DEC VECT XDRAW(%ROWS(S)) UDRAW(%ROWS(S))
*
COMPUTE P=%DECOMP(S)
COMPUTE UDRAW=X+P*(UDRAW=%RAN(1.0))
```

A multivariate Student (*t*) starts as a multivariate Normal draw, but is then divided through by a chi-square divided by its degrees of freedom. Change the final line in the code above to

```
COMPUTE UDRAW = X + $
                (.5*NU/ %RANGAMMA(0.5*NU))* $
                P*(UDRAW=%RAN(1.0))
```

The RATSletter

P.O. Box 1818
Evanston, IL 60204-1818
Phone: (847) 864-8772 Fax: (847) 864-6221
estima@estima.com www.estima.com