RATS Handbook

for Katarina Juselius'

The Cointegrated VAR Model

Part I

First Draft,

November, 2009

Estima

Introduction

This Handbook shows you how to use RATS and CATS to reproduce the results presented in Katarina Juselius' book *The Cointegrated VAR Model* (which we'll refer to as *TCVM*). The code presented here and in the accompanying RATS program files is based on the actual code used in writing the textbook. We are grateful to Katarina Juselius for providing that code and the associated data files.

Note that you will need Version 6.2 or later of RATS and Version 2.0 or later of CATS to run these examples.

Although we go over some of the basics of using both RATS and CATS, this is not intended as an introductory tutorial. If you are new to RATS, please begin by looking through the *Getting Started* booklet and working through the tutorial in Chapter 1 of the *RATS User's Guide*.

And, if you have not already done so, we strongly recommend that you read through at least the first three chapters of the CATS 2.0 user's manual. This will provide you with some theoretical background and the basic skills you will need to use CATS in RATS.

Contents

1. Getting Started

1.1 CATS Basics

CATS is not a stand-alone application. Rather, it is a set of “procedures” written in the RATS language. To use CATS, you start the RATS program, execute the instructions necessary to read your data into memory and do any transformations or other preliminary analysis. You then compile (or “source”) the CATS procedure file, and finally execute the procedure. Executing CATS adds several drop-down menus to the RATS menu bar, and you will use these menus to do your cointegration analysis. You can also save the resulting model and other information for further analysis using other RATS instructions.

If you are not familiar with using procedures in RATS, see Section 16.2.1 in the *RATS User’s Guide* for general instructions.

1.2 Getting Ready

The examples in *The Cointegrated VAR Model* book use a collection of Danish quarterly economic data, running from the first quarter of 1973 through the first quarter of 2003. The data are provided on an Excel spreadsheet called `book.xls`. Our first task is to read this data into RATS.

Begin by starting the RATS software. We recommend setting up the program with separate Input and Output windows, with the windows tiled so that both are visible on screen (see the *Getting Started* booklet if you don’t know how to do that). This setup makes it easy to save the instructions you write as a program file, and to save the output to another file for later reference.

You may also want to use *File–Directory...* to set the directory containing the `book.xls` data file as the default directory for this session (use the “Directories” tab on *File-Preferences...* if you want this to be the default directory each time you start RATS).

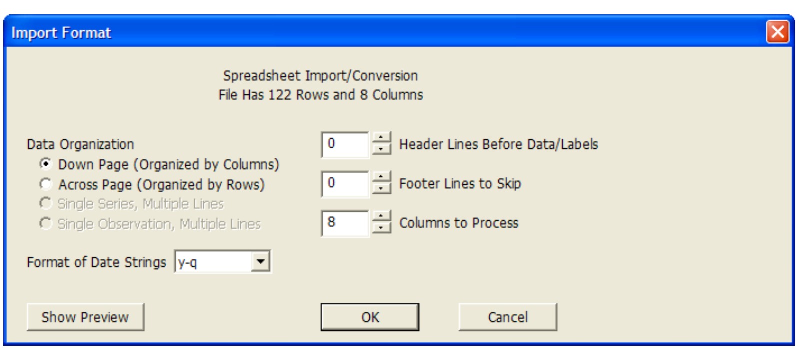
Note that we provide pre-written example programs containing all the code described in this handbook. The example files also include comment lines describing the necessary menu operations and input. You can follow along by executing the instructions provided in these files, or by typing in and executing the instructions yourself in the input window. If you choose to type in the instructions yourself (which is probably the best way to learn), be sure to save your work before quitting out of the program, so that you can easily pick up where you leave off.

1.3 Reading Data

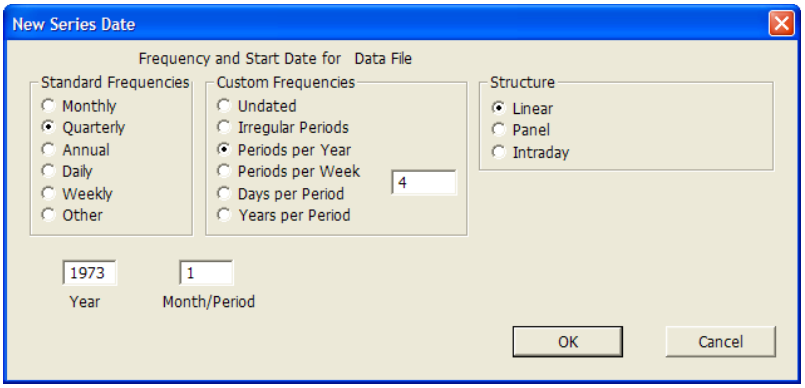
Once you have RATS open and set up the way you want, you are ready to read in the data. There are two ways to go about that.

Using the Data Wizard

One option is to use the *Data Wizard*. From the *Data* menu, select *Data (Other Formats)*. Use the “Files of type” field (which may have a slightly different name depending on the operating system you are using) to select “Excel Files (*.XLS)” as the desired format. Next, click on the `book.xls` file, and then click on “Open”. You will see a dialog box like this (older versions do not have the “Show Preview” button for previewing the contents of the file):



In this case, all of the information is correct, so just click OK. You will see a second dialog box:



RATS has correctly determined the frequency and starting date from the information on the file, so you can again just click on OK. This will generate and execute **OPEN DATA**, **CALENDAR**, **ALLOCATE**, and **DATA** instructions that set the frequency and starting date for this session and read in the data.

Typing in the Instructions

If you already know the starting date, frequency, and other details, you can just type in and execute the instructions directly. For this data set, you would enter the following instructions:

```
calendar(q) 1973:1
allocate  2003:1
open data book.xls
data(format=xls,org=obs) / lyr lpy lm3n Rm Rb lm3rC dpy
```

The **CALENDAR** instruction tells RATS you are working with quarterly data, starting in the first quarter of 1973. **ALLOCATE** sets the default ending period as the first quarter of 2003.

The **OPEN DATA** instruction tells RATS the name (and, optionally), the location of the file you want to read. The **DATA** instruction actually reads in the data. The FORMAT and ORGANIZATION options tell RATS the structure of the file. The “/” takes the place of the *start* and *end* parameters, and tells RATS to use the default data range. Finally, we list the names of the series we want to read in. Because we want all the series on the file, we could also omit the list of series names:

```
data(format=xls,org=obs)
```

This tells RATS to read all the data on the file.

1.4 The Data, Creating Series, and Data Transformations

Here’s how the series in the file relate to the notation used in the textbook. First, the five series that we will use in the main cointegration examples:

Series	Description
LM3RC	The (corrected) log of the real M3 money stock (m_t in the book)
DPY	The quarterly inflation rate (Δp_t)
LYR	The log of real income (the implicit price deflator of GNE) (y_t^r)
RM	An average deposit rate, or own interest on money stock, ($R_{m,t}$)
RB	The long-term government bond rate ($R_{b,t}$)

Two other series are provided on the file:

Series	Description
LPY	The log of the price levels
LM3N	The log of the nominal M3 money stock

As noted above, both the inflation rate (first differences of the price levels) and the real money stock variables are provided in the data. However, as your own analysis may require knowing how to do simple data transformations, it may be helpful to see how these two series could be computed from the LPY and LM3N series.

The real money stock series is defined as the difference between the nominal M3 series (LM3N) and the price levels (LPY). This could be computed in RATS using a **SET** instruction:

```
set lm3r = lm3n-lpy
```

However, the M3 series actually used for most of the analysis is deseasonalized, and includes a correction for a mistake in the 1991 data (see page 105 in *TCVM*). This corrected series is supplied as LM3RC on the data file (we'll look at the difference between the two later).

Similarly, the first difference of the price levels (Δp_t) is defined as

$$DPY_t = LPY_t - LPY_{t-1}$$

This is available on the data file, as noted above. If it weren't, it could be computed in RATS using a **SET** instruction similar to the one above:

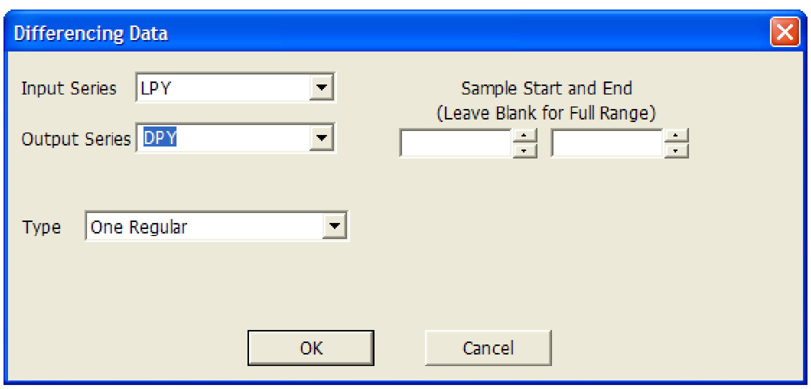
```
set dpy = lpy - lpy{1}
```

where `{1}` signifies the one period lag of the series (we call this *lag notation*).

RATS also offers a **DIFFERENCE** instruction specifically for doing various differencing operations. The equivalent to the **SET** instruction above would be:

```
diff lpy / dpy
```

Finally, you could also use either the *Differencing Wizard* or the *Transformations Wizard* (both on the *Data* menu) to do the differencing. Here's how it would look using the *Differencing Wizard*:



Again, though, the series DPY has already been provided on the data file, so you do not need to do this step yourself.

We *will* need to create some dummy variables for analysis later in the textbook (Section 6.6, pages 104-106), so we'll go ahead and explain how to do that here.

We'll need two dummies to deal with the possible effects of the temporary removal of the VAT in 1975. The first, which we'll call DT754, is a "transitory blip" dummy, defined with the value 1.0 in the 4th quarter of 1975 (hence the "754" name), the value –0.5 in each of the first two quarters of 1976, and 0 for all other periods.

We can create this dummy by using a **SET** instruction to define a series of zeros, and then **COMPUTE** instructions to set individual entries to the desired value.

```
set Dt754 = 0.0
compute Dt754(1975:4) = 1
```

```
compute Dt754(1976:1) = -0.5
compute Dt754(1976:2) = -0.5
```

This could also be done in other ways. For example:

```
set Dt754 = t==1975:4
set Dt754 1976:1 1976:2 = -0.5
```

The first line sets the series equal to zero, except for 1975:4, where the series is set to 1. The reserved variable *t* returns the entry being set, so this logical expression returns a one ("true") for 1975:4, and zero ("false") elsewhere. The second line sets the series to –0.5, but is run only over the first two periods of 1976. Note the use of the *start* and *end* parameters, which are normally omitted.

The second dummy variable is set to 1 in quarter 4 of 1976, and zero elsewhere. Again, this can be done as:

```
set Dp764 = 0.0
compute Dp764(1976:4) = 1
```

or simply

```
set Dp764 = t==1976:4
```

We need another dummy to deal with a permanent change—the removal of restrictions on capital movements in 1983. This command:

```
set Ds831 = t>=1983:1
```

sets DS831 to zero through 1982:4, and to 1 from 1983:1 onward.

1.5 Examining the data

Whenever you begin working with a new data set, we strongly recommend that you take some time to simply examine the data. This will help ensure there were no mistakes in the data itself or in the process of reading in the data. It also gives you a chance to observe the general behavior of the series you will be working with.

The **PRINT** and **TABLE** instructions are useful places to start: **PRINT** displays the values of one or more series, while **TABLE** provides a table of summary statistics for one or more series. To include all the series in memory, just issue the instruction with no parameters:

```
print
table
```

You may find it helpful to redirect output to a spreadsheet-style window, rather than as plain text in the output window. You can do that using the WINDOW option. For example:

```
print(window="Interest Rates") / Rm Rb
```

You may also want to graph your series. We'll discuss graphing more in the next section, but for now, try a command like:

```
graph(header="Interest Rates") 2
# Rm
# Rb
```

You can also use the *Graph Wizard* on the *Data* menu, or do *View–Series Window* and use the toolbar icons to graph selected series.

2. Regressions and Graphs

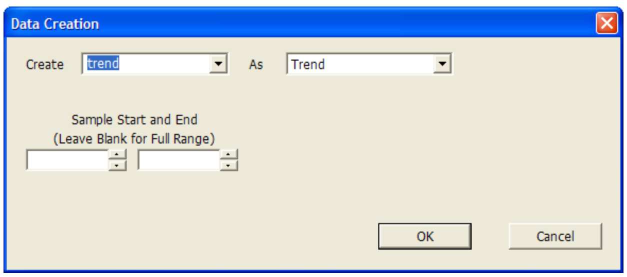
Next, we'll demonstrate how to create Figure 2.4, from page 25 of *TCVM*. To create the "trend-adjusted" price and income series, we need to run linear regressions of those variables on a constant and a deterministic trend series.

2.1 Creating a Trend Variable

The first step is to create the deterministic trend, which we can do easily using a **SET** instruction and the reserved variable *t*:

```
set trend = t
```

You could also do this using the *Trend/Seasonal/Dummies Wizard* on the *Data* menu. After selecting the wizard, just type in the series name (`trend`) in the dialog box and make sure "Trend" is selected in the "As" drop-down field, as shown below:



2.2 Linear Regressions

Now we can do the linear regressions and compute the trend-adjusted variables required for the graph. We use the **LINREG** instruction for doing linear regressions. The basic syntax is:

```
linreg depvar
# regressors
```

The trend adjusted income is defined as e_t from the regression

$$LYR_t = \beta_1 + \beta_2 \times TREND_t + e_t$$

We compute the regression as follows:

```
linreg lyr
# constant trend
```

Note that `CONSTANT` is a reserved variable name in RATS, used to include a constant term in an equation or regression.

The original code used for the textbook computed the trade adjusted income using the formula

$$LYR_t - \beta_1 + \beta_2 \times TREND_t$$

implemented as follows:

```
set tradjlyr = lyr - (0.0037216934*trend + 6.5482173338)
```

The numbers here are the coefficients from the regression, manually copied from the **LINREG** output table. This works, but the manual entry of the coefficient values is error-prone, and omits some of the precision of the computed values.

An easier and more accurate way to do this is to refer to the reserved variable `%BETA`, which is a vector RATS uses to store the coefficients from the most recent estimation. For example:

```
set tradjlyr = lyr - (%beta(2)*trend + %beta(1))
```

Given that the trend-adjusted series is really just the series of residuals from the regression, the *easiest* way to get this series is simply to use the *residuals* parameter on the **LINREG**:

```
linreg lyr / tradjlyr
# constant trend
```

In our example program, we use the first method for consistency with the published results, but we recommend using the second or third method in your own work.

Similarly, the trend-adjusted price level series was originally computed as follows:

```
linreg lpy
# constant trend
set tradjlpy = lpy - (0.012101365*trend - 1.095575207)
```

Again, in general, we would recommend just using the *residuals* parameter on **LINREG** instead:

```
linreg lpy / tradjlpy
# constant trend
```

2.3 Graphing the Results, Figure 2.4

Now that we've computed our trend-adjusted series, we can graph them. To graph the series separately, you can do something like:

```
graph(header="Trend adjusted real income")
# tradjlyr
```

and

```
graph(header="Trend adjusted price level")
# tradjlpy
```

To draw a single graph showing both series, we add "2" as a parameter indicating the number of series to be graphed, and add the KEY option to display the series names as a key:

```
graph(header="Trend adjusted income and prices",key=upleft) 2
# tradjlyr
# tradjlpy
```

To reproduce Figure 2.4 from page 25 of the *TCVM*, we use the **SPGRAPH** (SPecial GRAPH) instruction to display three graphs on a single "page". Here, we use the VFIELDS option to divide the page into three vertical segments. The **SPGRAPH(DONE)** instruction tells RATS that we have finished all the commands that comprise the special graph.

```
spgraph(vfields=3)
 graph(header="Trend adjusted log price: stochastic I(2) trend")
 # tradjlpy
 graph(header="Trend adjusted real agg. income: stochastic I(1) trend")
 # tradjlyr
 graph(header="Inflation rate: stochastic I(1) trend")
 # dpy
spgraph(done)
```

3. Using CATS: Preliminary Analysis

In this chapter, we'll use the CATS procedure to reproduce tables and figures from Chapters 3 and 4 of *TCVM*.

3.1 Compiling and Executing CATS

There are two ways to start the CATS procedure:

- Using the *CATS Wizard* on the *Statistics* menu, or
- By executing the procedure with a command of the form:

  ```
  @cats(options) parameters
  ```
  (followed by supplementary cards, depending on options chosen)

The *Wizard* only supports a limited set of the options available in CATS (although most other settings can be modified as needed using the *CATS–Model* operation). We'll briefly discuss using the *Wizard* below, but for the purposes of the textbook examples, we'll be typing in the **@CATS** commands directly.

3.1.1 The CATS Wizard

In order to use the CATS Wizard, you first need to make sure the "CATS Directory" field in the Preferences dialog box is pointing to the directory where the CATS files are installed on your system. You can do this by selecting *Preferences* from the *File* menu, and clicking on the "Directories" tab. If necessary, type in the name of the directory in the "CATS Directory" field, or use the "Browse" button to locate the directory. For Windows systems, the default location for CATS is

```
c:\cats2
```

but CATS may be installed in a different location on your system—if so, use that directory name.

Click on OK to save the changes, and answer "Yes" in the next dialog box if you want to save these changes for future sessions.

Once that is done, you can select *CATS* from the *Statistics* menu and use the dialog box to select the endogenous variables, the standard deterministic variable structure, and the estimation range. You can type variable names directly into the "Endogenous Variables" field, or click on the ... button to bring up a list of available series.

3.1.2 Using the @CATS Command

As with any RATS procedure, you can execute CATS using a syntax of the form

```
@procedurename(options) parameters
# supplementary card(s)
```

where the "**@**" symbol is a shortcut for the **EXECUTE** instruction.

As with other procedures, RATS needs to compile the procedure code by executing the commands on the file(s) that define the procedure. You can do that by using a **SOURCE** instruction. In the case of the CATS, you want to source in the file `CATS.SRC`. For example:

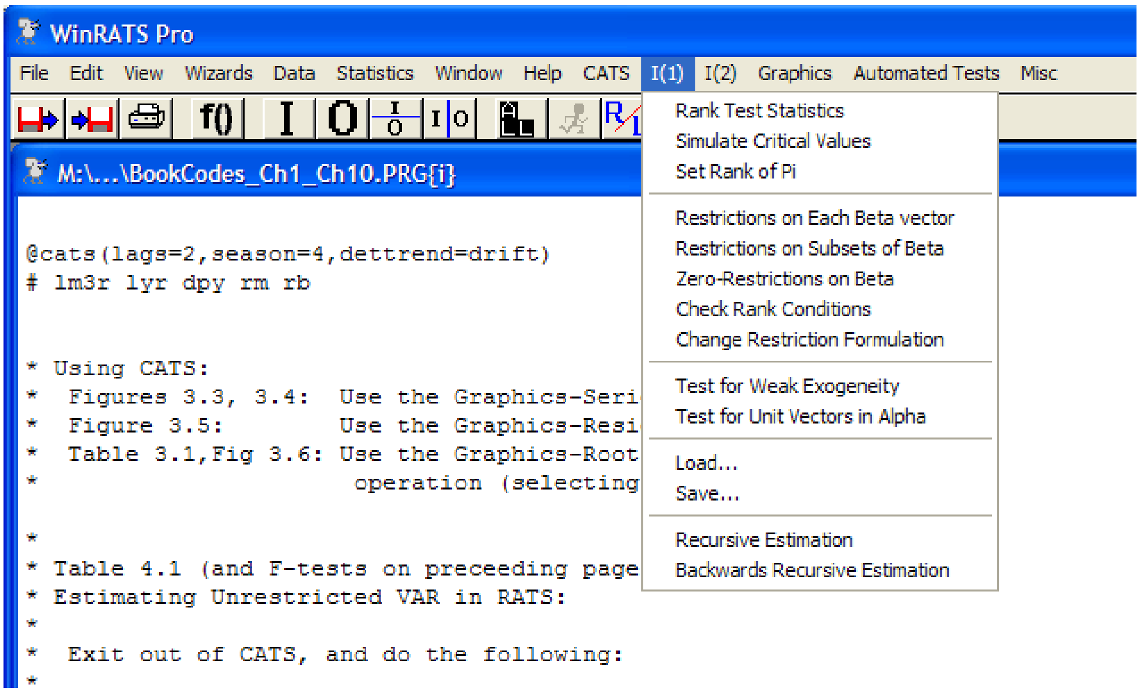
```
source c:\cats2\cats.src
```

Once you've done that, you can execute CATS. For the model used in Chapters 3 and 4 in the text, use the following instruction:

```
@cats(lags=2,season=4,dettrend=drift)
# lm3r lyr dpy rm rb
```

This specifies a model with two lags, four seasonal dummies, and with the “DRIFT” model for the deterministic variables. The supplementary card lists the names of the five endogenous variables. In this case, we are using the *uncorrected* version of the M3 series. We’ll switch to the corrected version later on.

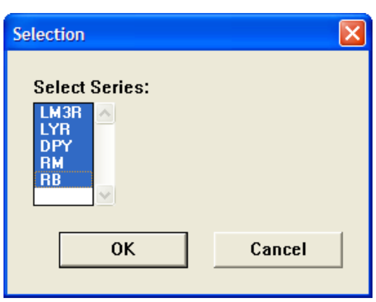
After you execute these two lines, CATS will do some initial computations and display some initial output. It then adds six new menus to the RATS menu bar. Depending on which release of RATS you are using, the screen will look something like this:



Note the addition of the *CATS*, *I(1)*, *I(2)*, *Graphics*, *Automated Tests*, and *Misc* menus. Here, we have the *I(1)* menu open, showing the operations available on that menu.

3.2 Graphing the Data in CATS, Figures 3.3 and 3.4

To generate Figures 3.3 and 3.4 in the textbook (page 41), select the *Series* operation from the *Graphics* menu. This opens a dialog you can use to select the series you want to graph. To graph all the series, highlight all the series names in the dialog box, like this:



and click on OK.

You should see five graph windows, with each graph window showing the series in both levels and differences. (If you only see one graph window, do *File–Preferences*, click on the “Graphics” tab, turn *off* the “One Graph Window Only” check, and click on OK).

3.3 Graphing the Residuals, Figure 3.5

To reproduce Figure 3.5 (page 47), which plots the residuals from the initial estimation, select *Residuals* from the *Graphics* menu. Again, you can select (highlight) all the series and click on OK to produce all five graphs. In addition to the (standardized) residuals, you'll get plots of actual and fitted values, an autocorrelations graph, and a histogram plot.

3.4 Roots of the Companion Matrix, Table 3.1, Figure 3.6

To produce Table 3.1 and Figure 3.6 (pages 51-52), just select *Roots of Companion Matrix* from the *Graphics* menu, and select "5" as the rank in the dialog box displayed by CATS. RATS will display the information on the roots in the output window, and the plot of the roots in another new graph window.

4. VAR and Error Correction Models

4.1 Unrestricted VAR(2) Estimates, *F*–tests, Table 4.1

Reproducing Table 4.1 (page 60 of *TCVM*) and the *F*-tests on the preceding page require estimating an unrestricted two-lag VAR model in levels. This is very easy to do in RATS.

If you still have CATS running, the first step is to exit out of the procedure by selecting *Exit* from the *CATS* menu (we'll be returning to CATS later in this chapter).

Next, we need to create a centered seasonal dummy variable, by executing the following instruction:

```
seasonal(centered) Dq
```

Now we can define our VAR model, using instructions specifically designed for this task:

```
system(model=var2mod)
variables lm3r lyr dpy rm rb
lags 1 2
deterministic dq{0 1 2} constant
end(system)
```

The **SYSTEM** instruction initiates the definition of the model, which will be saved with the name VAR2MOD. The **VARIABLES** instruction provides the list of endogenous variables, while **LAGS** provides the list of lags. The **DETERMINISTIC** instruction supplies the list of exogenous (deterministic) variables. Here, we use the contemporaneous value and first and second lags of DQ, as well as a constant term. **END(SYSTEM)** tells RATS that we are finished defining the model.

To estimate the VAR, we simply do:

```
estimate
```

This produces the standard RATS VAR estimation output, including virtually all of the information in Table 4.1 (plus a lot more). The residual covariance matrix, log likelihood and log determinant are automatically stored into reserved variables. You can view these values using **DISPLAY** instructions:

```
display "Residual covariance matrix = " ##.#### %sigma
display "Residual correlation matrix = " ##.#### %cvtocorr(%sigma)
display "Log likelihood = " %logl
display "Log determinant = " %logdet
```

The picture codes (`##.####`) on the first two **DISPLAY** instructions provide a template for the output, telling RATS to only display four digits after the decimal point for each element of the array.

The **ESTIMATE** instruction automatically does *F*–tests on omitting each variable from a given equation. However, the text reports *F*–tests on excluding each regressor (i.e. each variable/lag combination) from the entire model (page 59), and a global *F*–test on all regressors across all equations.

These can be computed fairly easily in RATS, as shown below. This code also uses **REPORT** instructions to format the output in a convenient table.

Note: The *F*–test results in the book appear to be off by a factor of 105/101, apparently due to an accidental double-counting the deterministic coefficients.

First, computing the individual regressor *F*–tests:

```
* Get the first equation and the coeffecients from the estimated model:
compute eqn = %modeleqn(var2mod,1)
compute betaols = %modelgetcoeffs(var2mod)
```

```
* Use REPORT feature to generate the output table:
report(action=define)
 * Add a row of column headers:
 report(atrow=1,atcol=1) "Label" "F-Stat"
 * Loop over the number of regressors in the equation:
 do i=1,%eqnsize(eqn)
   * Compute the F-statistic, using several reserved variables:
   compute fstat = %qform(inv(%sigma),%xrow(betaols,i))*$
     (%nobs-%nreg)/(5*%nobs*%xx(i,i))
   * Add the variable label in column 1, the F-stat in column 2:
   report(row=new,atcol=1) %eqnreglabels(eqn)(i) fstat
   * Flag the F-stat with a star if the condition below is true:
   if %ftest(fstat,%nvar,%nobs-%nreg)<0.05
      report(action=format,atrow=%reportrow,atcol=2,special=onestar)
 end do i
 * Apply some formatting to the report table:
 report(action=format,atcol=2,picture="*.##",align=decimal)
* Display the report:
report(action=show)
```

And now computing the global *F*–test (see the *RATS Reference Manual* for details on the TR, %XSUBMAT, and %DOT functions used in computing these):

```
compute xx10   = inv(%xsubmat(%xx,1,10,1,10))
compute beta10 = %xsubmat(betaols,1,10,1,5)
compute test   = tr(beta10)*xx10*beta10
display "F-test on all regressors" $
           %dot(test,inv(%sigma))*(%nobs-%nreg)/(50.0*%nobs)
```

4.2 Error Correction Formulation,Tables 4.2 through 4.4

Next, we'll look at Vector Error (or equilibrium) Correction Models, which are a different way of parameterizing a VAR model. CATS itself is designed around the VECM formulation, so you will generally want to use CATS to estimate these models. First, though, we'll show you how to do it using RATS instructions.

4.2.1 Using RATS Instructions

Rewriting the VAR model in VECM format is a matter of differencing the endogenous variables, reducing the lag length by one, and adding lags of the original endogenous variables as exogenous variables. First, we'll use **DIFFERENCE** instructions to create the necessary differenced series:

```
diff lm3r / dlm3r
diff lyr / dlyr
diff dpy / ddpy
diff rm / drm
diff rb / drb
```

Now we'll define the new VECM system:

```
system(model=vecm1mod)
variables dlm3r dlyr ddpy drm drb
lags 1
deter dq{0 1 2} lm3r{1} lyr{1} dpy{1} rm{1} rb{1} constant
end(system)
```

Note the differenced endogenous variables, the omission of the second lag from the list of lags, and the addition of the lagged levels (actually, lagged first differences in the case of DPY, which appears as a second difference in the endogenous variables list). Estimation is done as before:

```
estimate
```

This produces most of the results from Table 4.2 (page 62). Note that the likelihood terms are unchanged from the original VAR form:

```
display "Log likelihood = " %logl
display "Log determinant = " %logdet
```

Reproducing Table 4.3 (page 63) is just a matter of redefining the VAR, changing the **DETERMINSITIC** instruction to specify lag two rather than lag one on the lagged level variables:

```
deter dq{0 1 2} lm3r{2} lyr{2} dpy{2} rm{2} rb{2} constant
```

Table 4.4 is just another variation using second differences rather than first differences, and can be reproduced using instructions similar to those above if desired.

4.2.2 Using CATS

Now, restart CATS using the same procedure call as before:

```
@cats(lags=2,season=4,dettrend=drift)
# lm3r lyr dpy rm rb
```

CATS automatically transforms the model you supply into error correction form, so the two-lag reduced form model specified here is translated internally into the same VECM model described above. Executing the instruction above produces the same unrestricted VECM results, as presented in Table 4.2.

To see all of the relevant coefficients, select the *Short Run Parameters* option from the *Misc* menu.

4.3 Residuals, Correlations and Specification Tests

You can produce Figures 4.1 through 4.5 (pages 67-69) just as we did earlier, by selecting the *Graphics-Residuals* and graphing all five series. Note that CATS 2.0 produces a Histogram and some normality test statistics, rather than the QQ–plots shown in the manual, which were done using a preliminary version of CATS 2.0.

To produce Figure 4.6 (page 70), use the *Graphics–Correlations* operation, selecting “None” for the “Residual Transformation” when prompted.

For Table 4.5 (page 71), select *Lag Length Determination* from the *Misc* menu, and use the default setting of five lags.

For the various specification tests presented on pages 73-77, use the *Misc–Residual Analysis* operation.

5. Cointegrated VAR Model

Chapter 5 of the text provides an overview of the cointegrated VAR model. The **Π** matrix shown on page 80 of *TCVM* is displayed as part of the initial output when you load CATS, as in Section 4.2.2 above.

To generate the cross-plot on page 89, we need to exit out of CATS, and then do a simple regression of income on money stock:

```
linreg lyr
# constant lm3rc
```

Next, we compute fitted values using the **PRJ** (for PRoJect) instruction:

```
prj fitted
```

Finally, we use **SCATTER** to draw the scatter-, or cross-plot. Here, we plot the original data points using the DOTS style, and use the OVERLAY and OVSAME options to plot the fitted values using a line style.

```
scatter(style=dots,overlay=line,ovsame, $
   vlabel="Real Income",hlabel="Real Money Stock",frame=half) 2
# lyr lm3rc
# fitted lm3rc / 1
```

6. Deterministic Components

Chapter 6 of the text examine the various choices for including deterministic variables in the model.

6.1 Figure 6.3

From this point on, we'll be working with series LM3RC, which is the "corrected" and deseasonalized version of the M3 series. For Figure 6.3 (page 109), we first need to compute the nominal version of the corrected money stock series, which we can do with a simple **SET** instruction using the corrected real series and the price series:

```
set lm3nc = lm3rc + lpy
```

Now we can graph the corrected and uncorrected nominal M3 series:

```
graph(footer="The original and corrected M3 in logs.",key=upleft,$
      max=6.5,pattern) 2
# lm3nc
# lm3n
```

Note that we use a **$** symbol here to continue the **GRAPH** instruction onto a second line.

To get a better look at the effects of removing the seasonality, we can graph a susbsample of the data comparing the corrected and uncorrected nominal series:

```
graph(footer="The original and corrected M3 in logs.",key=upleft,$
      max=6.5,pattern) 2
# lm3nc 1985:1 1992:1
# lm3n 1985:1 1992:1
```

6.2 Tables 6.2 and 6.3

To produce Tables 6.2 and 6.3 (pages 111-112), we need to restart CATS with some changes to the model:

1. Because we are now using a seasonally adjusted version of the M3 series, we can remove the seasonal dummy terms by deleting the `SEASONAL=4` option.
2. We add DS831 (defined earlier) as a weakly exogenous variable. To do this, we add the `EXO` option and list the variable on a second supplementary card.
3. We include DT754 and DP764 as dummy variables (also defined earlier), using the `DUM` option and list these variables on a third supplementary card.

```
@cats(lags=2,exo,dum,dettrend=cidrift) 1973:1 2003:1
# lm3rc lyr dpy Rm Rb
# Ds831
# Dt754 Dp764
```

The Π matrix shown in the initial output provides the second array of coefficients in Table 6.3.

Now, select *Residual Analysis* from the *Misc* menu. The first portion of the resulting output, "Residual S.E. and Cross-Correlations", reproduces the standard deviations shown in Table 6.2 and the standardized residual covariance matrix from Table 6.3.

The rest of the output reproduces additional portions of the results in Table 6.2, with some variations. In addition to the information in the text, CATS now produces a Ljung-Box test result as well as Schwarz and Hannan-Quinn criterion values. Also, the second LM test is done for second order autocorrelation, rather than fourth order as shown in the textbook. Finally, you will also see LM tests for ARCH effects in the output, which are not included in the text.

Now select *Roots of Companion Matrix* from the *Graphics* menu, select rank "5", and click on OK. You will get the last section of Table 6.2 (with some variations in the "Imaginary" values due to changes in CATS since the textbook work was done).

To get the first and third coefficient arrays in Table 6.3, select *Short Run Parameters* from the *Misc* menu. The CATS output labeled "Lagged Differences" gives the first array of coefficients in 6.2 (on the lagged differences of the endogenous variables).

The coefficients from the third array in 6.2 are organized into the following separate tables in the CATS output: "Weakly Exogenous/Fixed Variables", "Time $t-1$", "Dummy Variables", and "Constant". *Note: the third coefficient array in Table 6.3 in the textbook appears to have been inadvertently transposed.*

As noted earlier, the middle coefficient array in Table 6.3 is the $\mathbf{\Pi}$ matrix produced in the initial CATS output.

7. Estimating the I(1) Model

Chapter 7 takes a closer look at estimating the I(1) (first-order cointegrated) model. In terms of using CATS, the primary element introduced here is the concept of normalizing the eigenvectors.

7.1 Restarting CATS Using SHIFT Rather Than EXO

If you still have CATS running, exit CATS and restart using the following instructions:

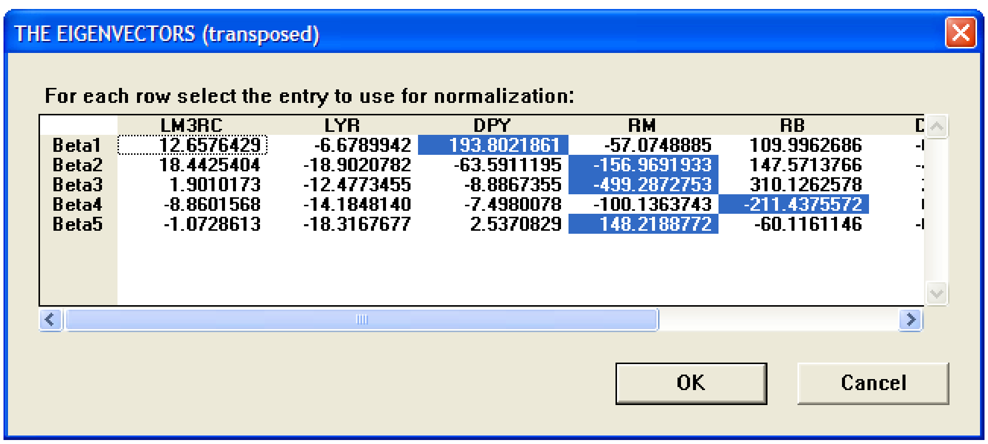
```
@cats(lags=2,shift,dum,dettrend=cidrift)
# lm3rc lyr dpy Rm Rb
# Ds831
# Dt754 Dp764
```

Note the one important change here: we are now using the `SHIFT` option, rather than the `EXO` option. This has the effect of eliminating $\Delta DS831_{t-1}$ (the lagged difference of DS831) from the set of dummy variables in the model (see pages 29-30 of the CATS manual for details on these options).

7.2 Table 7.1

The non-normalized eigenvectors and **Π** matrix of Table 7.1 (page 123) are shown automatically as part of the initial CATS output. To get the normalized eigenvectors and Alpha matrix, select *Set Rank of Pi* from the *I(1)* menu and enter 5 as the rank (5 is already the default rank, but this operation also allows us to select a normalization).

You will see the following dialog box.

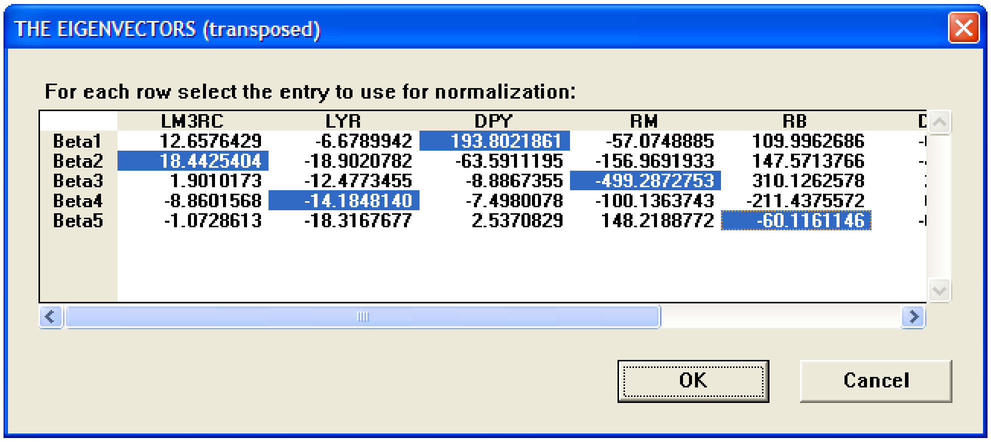


This dialog allows you to select the variable (column) on which you want to normalize each of the five beta vectors. Note that not all the columns are visible here—use the horizontal scroll bar in the dialog to see the other columns.

By default, CATS selects the largest (in magnitude) values in each row, as you can see above. To choose a different normalization, you just need to click on (highlight) the desired column in each row.

For the normalizations used in Table 7.1 (note the locations of the “1.00” values in the “Normalized eigenvectors” section of Table 7.1), normalize row 1 on DPY, row two on LM3RC, row three on RM, row four on LYR, and row five on RB.

The resulting dialog should look like the one on the following page:



Once you have highlighted the appropriate columns, click OK to impose the normalization.

The values shown in the "Normalized eigenvectors: $\hat{\beta}_i'$" table in the text should match those displayed in the "BETA(transposed)" section of the CATS output. The "weights to the eigenvectors: $\hat{\alpha}_i$" are shown in the "ALPHA" section of the output. The $\mathbf{\Pi}$ matrices are unchanged by the normalization.

Note: To see the eigenvalues shown in the first column of the "Normalized eigenvectors: $\hat{\beta}_i'$" table, you can use the *Rank Test Statistics* from the *I(1)* menu. We discuss that operation in more detail in the next chapter.

7.3 Figures 7.1 through 7.5

To generate Figures 7.1 through 7.5 from the text (pages 125-127), select *Cointegrating Relations* from the *Graphics* menu, highlight all 5 cointegration relations in the dialog box, and click OK. RATS will generate all five graphs.

8. Cointegration Rank

Chapter 8 of the text looks at tools available for choosing the appropriate cointegration rank. The choice of cointegration rank must be made carefully, as it has a very significant impact on any further analysis of the data.

8.1 Rank Test Statistics, Table 8.1

Reproducing the test statistics from Table 8.1 (page 144) is very easy. Continuing from where we left off in the previous Chapter, just select *Rank Test Statistics* from the *I(1)* menu to get the first half of the table. Here's the resulting output:

```
I(1)-ANALYSIS
 p-r  r  Eig.Value   Trace   Trace*  Frac95 P-Value P-Value*
   5  0      0.348 126.450 118.572 88.554    0.000    0.000
   4  1      0.230  75.631  70.948 63.659    0.003    0.010
   3  2      0.196  44.521  40.336 42.770    0.032    0.088
   2  3      0.085  18.612  16.840 25.731    0.311    0.435
   1  4      0.065   7.997   7.320 12.448    0.260    0.322
```

The *p–r* and *r* columns in the CATS output are identical to those in the text (although the text lists the *r* column first). The "Eig. Value" column is the λ_i in the text. The "Trace" column in CATS is the $\tau(p-r)$ trace test, while the "Trace*" column is the $\tau^*_{\text{Bart.}}(p-r)$ trace test computed using the Bartlett small sample correction.

The "Frac95" column is the 95% quantile value for the basic model, described as $C_{.95}$ in the text. The asymptotic critical values based on the inclusion a trend and a shift dummy in the cointegration relations (C^{Ds} in the text) are provided in Appendix C of the CATS User's Manual. You can refer to Section 6.7 of the CATS manual (pages 143-145) for more help in determining which table in Appendix C is appropriate for a given model. In this case, you would refer to Table C.4.

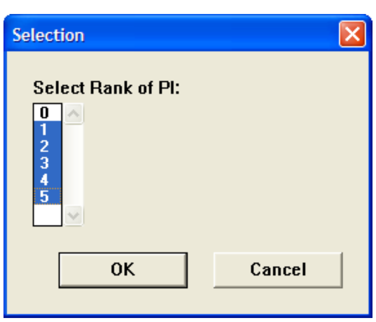
In addition, CATS produces uncorrected and Bartlett-corrected (approximate) *p*-values for the trace test statistics, in the "P-Value" and "P-Value*" columns, respectively.

The text book (page 143) takes you through the model selection process using comparisons against the two $C_{.95}$ values. Here, we'll go through the process using the *p*-values instead.

Using the Bartlett-corrected Trace tests and *p*-values, we would strongly reject the null of five unit roots ($p-r=5$). We would reject the hypothesis of four unit roots ($p-r=4$) at the 5% level, but not at the 1% level, with a *p*-value of .01. The null of three unit roots would be accepted at the 5% level (with a *p*-value just under .09). Using the uncorrected tests, the first two hypotheses are strongly rejected. The hypothesis of three unit roots is rejected at the 5% level, but not at the 1% level. Finally, the hypothesis of two unit roots is easily accepted. So, the results based on the *p*-values are very similar to those described in the text.

See page 45 of the CATS user's manual for more information details on this feature.

For the second part of the table, select *Roots of Companion Matrix* from the *Graphics* menu, and highlight ranks 1 through 5 in the dialog box, like this:



This will generate a graph showing plots of the roots for all five choices of rank, and tables showing all the roots for each case. Note that the tables are generated by CATS in the order

$r = 1, r = 2, \ldots, r = 5$

which is the opposite order of the columns in the textbook. Also, note that the textbook only shows the five largest roots for each case. Refer to the "Real" column in the CATS output. For example, here is the first set of output in CATS:

```
The Roots of the COMPANION MATRIX // Model: H(1)
        Real  Imaginary Modulus Argument
Root1   1.000     0.000   1.000    0.000
Root2   1.000     0.000   1.000    0.000
Root3   1.000     0.000   1.000    0.000
Root4   1.000     0.000   1.000    0.000
Root5   0.467    -0.000   0.467   -0.000
Root6   0.338    -0.291   0.446   -0.712
Root7   0.338     0.291   0.446    0.712
Root8  -0.266     0.117   0.290    2.728
Root9  -0.266    -0.117   0.290   -2.728
Root10 -0.125     0.000   0.125    3.142
```

The first five rows of the "Real" column match those of the $r = 1$ column in the text.

9. Recursive Tests of Constancy

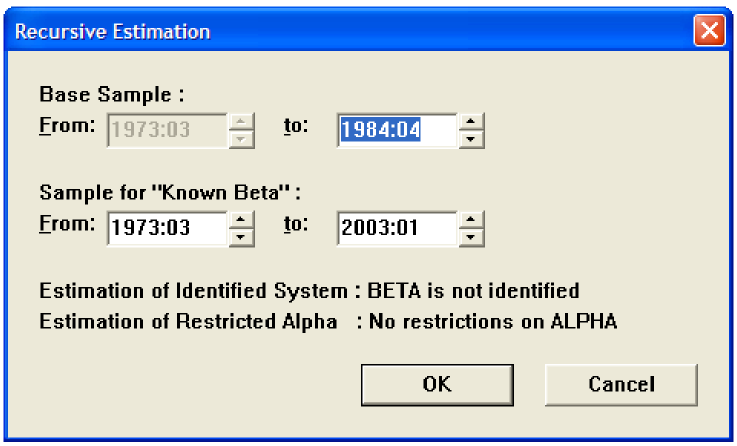
9.1 Setting the Rank of Pi

Some of the results in Chapter 9 of the text can be produced without setting a reduced rank, but as many of them do require a reduced rank, we will begin by setting the rank of the Π matrix to 3, as suggested at the conclusion of Chapter 8 in *TCVM*.

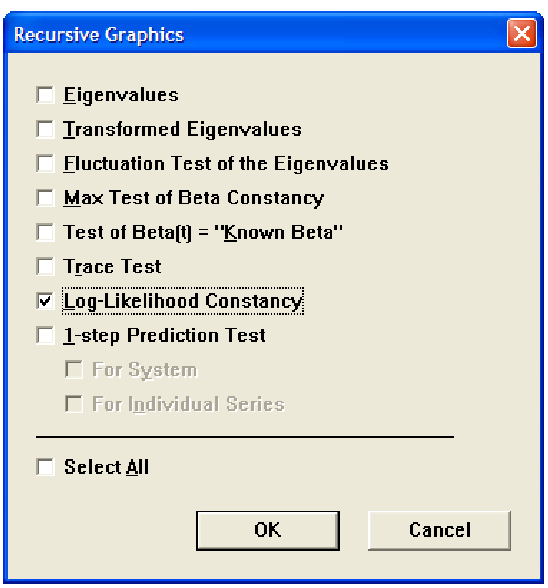
Select the *Set Rank of Pi* operation from the *I(1)* menu. CATS will display a dialog box prompting you to enter the rank. Enter the number 3 and click on OK. For now, we can just accept the default normalization suggested by the program, so just click on OK in the normalization dialog. CATS will display the results from estimating the reduced-rank model.

9.2 Recursive Log Likelihood, Figure 9.1

To begin the recursive estimation analysis, select *Recursive Estimation* from the *I(1)* menu. CATS will display a dialog box allowing you to set both the "Base Sample" and the "Known Beta" sample. As indicated on page 151 of *TCVM*, we want to use a base sample that extends through 1984:4. So, adjust the ending period for the base sample to 1984:4 as shown below, and click on OK:



CATS performs the recursive estimation and displays the dialog box below, which you use to choose the graphs you want to view. For now, turn on the checkbox for "Log Likelihood Constancy" as shown below and click on OK. CATS will ask if you want to save the recursive series. Answer "No". The program will then display the graph presented as Figure 9.1 in the text (page 152).



9.3 Recursive Trace Tests, Figure 9.2

Select the *Recursive Estimation* operation again. Because we have already done the recursive estimation, CATS will display a dialog box indicating that fact, and asking if you want to reestimate the model. If you wanted to change the sample ranges used for the estimation, you would say "Yes" and make your changes.

For now, though, just click "No" to the question about re-estimating. CATS will keep the previous estimates and display the "Recursive Graphics" dialog box again. This time, turn on the "Trace Test" check box and click on OK. CATS will generate the graph used for Figure 9.2 (page 153), displaying the recursive test results for the five rank hypotheses for both forms of the model.

9.4 Recursive Eigenvalues and Fluctuation Tests, Figures 9.3-9.5

If you don't need to change the recursive estimation sample, there is a faster way to display these recursive graphics. Rather than selecting *Recursive Estimation* again, choose the *Recursive Graphics* from the *Graphics* menu. You'll see the same "graphics" dialog box shown above, without having to answer the question about re-estimating the model.

This time, turn on the first three checkboxes in the "Graphics" dialog (for "Eigenvalues", "Transformed Eigenvalues", and "Fluctuation Test of the Eigenvalues"). This generates the graphs shown in Figures 9.3, 9.4, and 9.5 (pages 155-156).

9.5 Beta Constancy Tests, Figures 9.6 and 9.7

Choose the *Recursive Graphics* from the *Graphics* menu again, and select the "Max Test of Beta Constancy" and "Test of Beta(t) = 'Known Beta'" graphs. These operations generate the graphs shown in Figures 9.6 and 9.7 (page 158), respectively.

9.6 Changing the Estimation Sample, Figure 9.8

Figure 9.8 is produced using a different estimation sample. To produce this, do *Recursive Estimation* again, and this time answer "Yes" when asked if you want to re-estimate the model.

In the dialog box, set the starting date of the "Known Beta" range to 1986:1 (per page 162 of *TCVM*) and click OK. Turn on the checkbox for the "Test of Beta(t) = Known Beta" graph and click OK to generate the graph shown on page 159.

9.7 Another Estimation Sample, Figure 9.9

Figure 9.9 (page 160) is produced using yet another sample. To generate this, do *Recursive Estimation* again, and again answer "Yes" to re-estimate. In the dialog box, reset the starting date of the "Known Beta" range back to 1973:3. Then set the ending date for "Known Beta" to 1986:1. Click OK to estimate, and then generate the "Test of Beta(t) = Known Beta" .

9.8 Prediction Errors, Figures 9.10-9.12

For these prediction error graphs (page 161-163), we need to reset the end of the "Known Beta" range back to 2003:1. Do *Recursive Estimation* again, and reset that ending period to 2003:1, and click OK. In the graphics dialog box, turn on "1-step Prediction Test" , then turn on both the "For System" and "For Individual Series" boxes. Click OK to generate the graphs.

9.9 Backwards Recursive Tests, Figures 9.13-9.21

The procedure for the next set of graphs is basically the same as above, except you use the backwards version of the recursive estimation routine.

Start by selecting the *Backwards Recursive Estimation* operation from the *I(1)* menu. You'll be asked again if you want to re-estimate. Click on "Yes". In the estimation range dialog box, set the starting date for the "Base Sample" to 1986:1.

To produce all of the graphs, just click on "Select All" in the graphics dialog box and click on OK. This will generate all of the remaining graphs in Chapter 9, except for Figure 9.18, which is done using a different sample range.

To generate 9.18, select *Backwards Recursive Estimation* and change the "Known Beta" range to run from 1983:1 through 2003:1, leaving the start of the "Base Sample" set to 1986:1. After estimating, select "Test of Beta(t) = Known Beta" to generate the graph.

10. Testing Restrictions on Beta

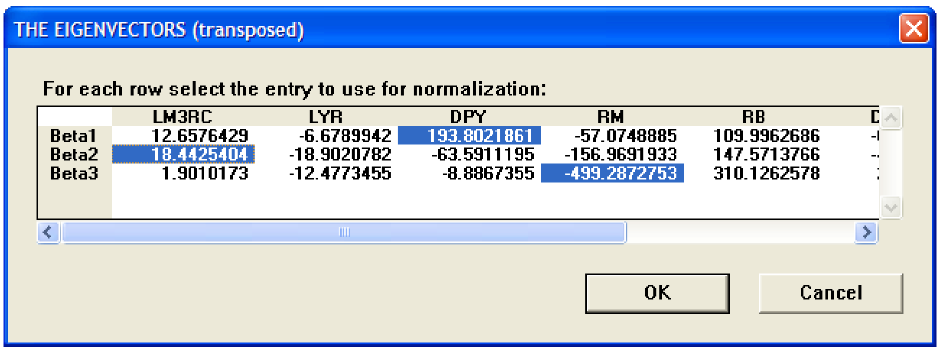
In Chapter 10, we'll look at various ways to impose and test restrictions on the Beta matrix. If you need to restart CATS, recall that we are using the following model and settings:

```
@cats(lags=2,shift,dum,dettrend=cidrift)
# lm3rc lyr dpy Rm Rb
# Ds831
# Dt754 Dp764
```

10.1 Normalizing

The first step is to select the normalization used in the text. To do that, do the *Set Rank of Pi* operation on the *I(1)* menu, and enter "3" as the rank.

In the normalization dialog box, normalize row 1 on DPY, row 2 on LM3RC, and row 3 on RM, like this:



Click OK to impose the normalization.

10.2 Two Restriction Formulations

CATS offers two ways to impose restrictions on the Beta matrix. You can switch between the two methods using the *Change Restriction Formulation* operation on the *I(1)* menu. You can also use the *Preferences* operation on the *CATS* menu to change the default setting for this.

The default model is "Beta = H*Phi", where Phi is a matrix of free parameters. You impose restrictions by providing an **H** matrix of 1s and 0s, such that the elements of the Beta matrix are defined by the linear combinations of free parameters resulting from H*Phi.

The other formulation is "R′*Beta = 0", where **R** is a matrix of 1s and 0s, such that certain linear combinations of elements of the Beta matrix are restricted to being equal to zero.

As noted in the CATS manual (page 56), choosing which formulation to use is often a matter of taste. However, some types of restrictions are much easier to conceptualize using one approach, while others lend themselves more easily to the other approach. We'll show examples of both approaches in this handbook.

10.3 Some Simple Restrictions, Table 10.1

Table 10.1 presents the results of imposing six different restrictions on Beta. These are all relatively simple restrictions, either zero restrictions on individual elements, simple equality restrictions, or a combination of both.

The first three rows in Table 10.1 are simply the unrestricted estimates, available in the initial CATS output.

10.3.1 Hypothesis H1 Using H*Phi

Hypothesis H1 is a test on the exclusion of the long range linear trend from the cointegrating relations. Using the H*Phi approach, this is accomplished by setting the **H** matrix as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Given the Phi matrix shown in the text, multiplying $\mathbf{H}\varphi$ gives:

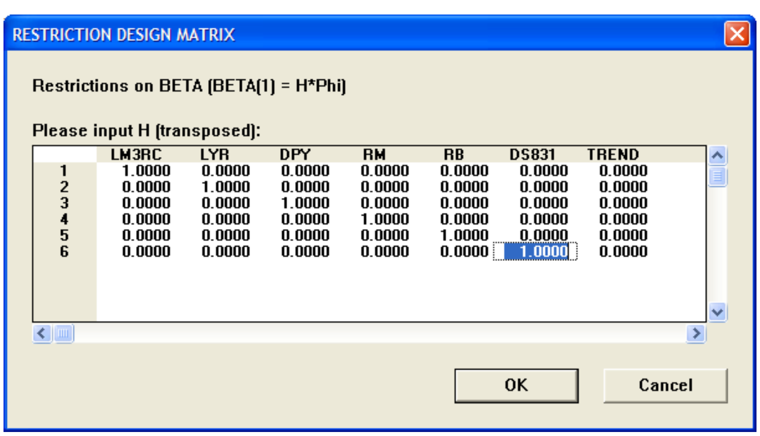
$$\begin{bmatrix} \varphi_{11} & \varphi_{12} & \varphi_{13} \\ \varphi_{21} & \varphi_{22} & \varphi_{23} \\ \varphi_{31} & \varphi_{32} & \varphi_{33} \\ \varphi_{41} & \varphi_{42} & \varphi_{43} \\ \varphi_{51} & \varphi_{52} & \varphi_{53} \\ \varphi_{61} & \varphi_{62} & \varphi_{63} \\ 0 & 0 & 0 \end{bmatrix}$$

Note that the last row contains only zeros, which means that there are no free parameters (φ terms) associated with the trend component, thereby excluding the trend from the model.

If you are ever confused about how the various **H** matrix setups presented in the text (and this handbook) produce the desired restrictions, you may find it helpful to multiply out the $\mathbf{H}\varphi$ product as shown above. This should make the formulation more clear.

To impose (and test) this restriction in CATS,

1. Select *Restrictions on Subsets of Beta* from the *I(1)* matrix,
2. Accept the default values on the first dialog (one subset, with three vectors in that subset),
3. Enter 1 as the number of restrictions
4. CATS prompts you to input the $\mathbf{H}'$ matrix. Note that this is the *transpose* of the matrix shown above. Enter 1's for the appropriate cells, as shown below, and then click "OK" to continue:



5. Accept the suggested degrees of freedom.
6. Use same normalization as described above.

The resulting “Beta(transposed)” table in the CATS output should match the H1 section of Table 10.1 (page 181).

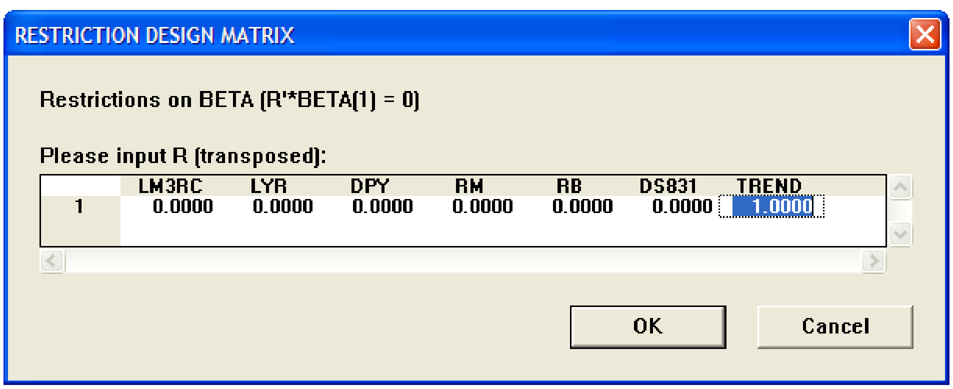
Note that by default, CATS displays output using 3 decimal places. If you want to changes this to two decimal places to match the textbook tables, select the Preferences operation from the CATS menu and change the value in the “Output Format: Number of Decimals” field from 3 to 2. Turn on the “Save Settings” switch if you want to save this as the new default setting, or leave the switch off if you only want to change the setting for this current session. Click on OK to close the dialog box. Any new output generated by cats should now use 2 decimal places.

10.3.2 Hypotheses H1 and H2, Using R*Beta=0

Now, we’ll do this using the “R′* Beta=0” form. Start by choosing *Change Restriction Formulation* from the *I(1)* menu and select the “R′* (Beta/Tau) = 0” formulation in the dialog box.

Then:

1. Select *Restrictions on Subsets of Beta* from the *I(1)* menu,
2. Accept the default values on the first dialog (one subset, with three vectors in that subset),
3. Enter 1 as the number of restrictions
4. Enter the value 1 in the “Trend” column of **R**:



5. Accept the suggest degrees of freedom
6. Use same normalization as described above.

Again, the resulting “Beta(transposed)” output should match the H1 section of Table 10.1.

The process for imposing hypothesis H2 is essentially identical, except that you put a 1 in the “DS831” column of **R**, rather than in the “Trend” column.

10.3.3 Hypotheses H3 and H4

For H3 and H4, the process is again the same as above, except that you will need to put 1s in two columns. For H3, you want 1s in the first two columns (for LM3RC and LYR), with zeros elsewhere, as shown on page 179 of the text. For H4, you want the 1s in columns four and five (for RM and RB).

10.3.4 Hypotheses H5, a Joint Test

Hypothesis H5 is a joint test, combining the H1 and H3 restrictions. Here are the steps—the major change is that we enter two as the number of restrictions in step 3:

1. Select *Restrictions on Subsets of Beta* from the *I(1)* menu,
2. Accept the default values on the first dialog (one subset, with three vectors in that subset),
3. Enter 2 as the number of restrictions,
4. Enter the **R** matrix as shown on page 180 of the text, with 1s in columns 1 and 2 (LM3RC and LYR) in row 1, and a 1 in column 7 (TREND) of row 3.
5. Accept the suggested degrees of freedom
6. Use same normalization as described above.

10.4 Hypothesis H6 and Table 10.2, Three Restrictions

Hypothesis H6 combines the restrictions from H1, H3, and H4. To implement this, the procedure is similar to the above, except that you enter 3 restrictions, rather than 2. The **R** matrix will combine the setups from H1, H3, and H4. That is, set row 1 of **R** according to the pattern for H1, row 2 according to the pattern used for H3, and row 3 to the pattern used for H4:

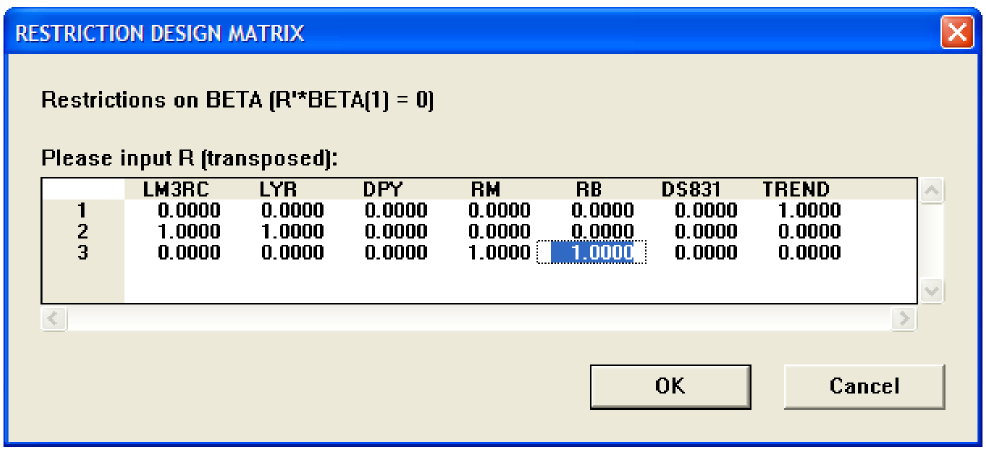


Table 10.2 (page 182) compares the unrestricted estimates of Π against the restricted H6 estimates. The restricted results in the Table are shown in the rows labelled "R", and these should match the results you just produced by imposing the H6 restrictions.

The unrestricted estimates are shown in the rows labeled "UR", and these should match the results you got from CATS immediately after setting the rank of 3 and normalizing, but before imposing any restrictions.

If you need to reproduce these unrestricted results again, you can repeat the *Set Rank of Pi* operation, enter 3 as the rank, and normalize as before.

10.5 Table 10.3, Unrestricted Constant Model

Now exit CATS, and restart with this model, using the `DETREND=DRIFT` option:

```
@cats(lags=2,shift,dum,dettrend=drift)
# lm3rc lyr dpy Rm Rb
# Ds831
# Dt754 Dp764
```

As before, set the rank to 3 and normalize as indicated in the text (row 1 on DPY, row 2 on LM3RC, and row 3 on RM).

10.5.1 Table 10.3, Hypotheses H7, H8, H9, Stationarity

As noted on page 185 of *TCVM*, hypotheses H7 through H12 test the stationarity of known vectors. As described on pages 183-4, this is done by decomposing the r cointegrating vectors (where r is three in this case) into nb known (i.e. restricted) vectors and $r - nb$ unknown vectors.

First, we'll look at hypotheses H7, H8, and H9, which each test a restriction on a single variable (DPY, RM, and RB, respectively). CATS actually offers a menu operation specifically for doing these stationarity tests. Just select the *Variable Stationarity* operation on the *Automated Tests* menu, which generates stationarity tests for all five variables, for four difference choices of rank. Note: Turn *off* the "Shift Dummies" switch in the dialog box to reproduce the results presented in the text.

The menu operation produces the following output:

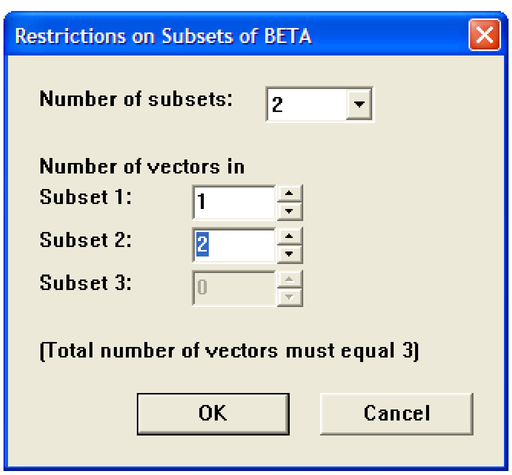
```
TEST OF STATIONARITY
LR-test, Chi-Square(6-r), P-values in brackets.

 r   DGF   5% C.V.   LM3RC    LYR      DPY      RM       RB
 1    5    11.070   48.229   47.854   36.307   46.175   49.248
                   [0.000]  [0.000]  [0.000]  [0.000]  [0.000]
 2    4     9.488   28.460   28.207   25.498   26.388   29.494
                   [0.000]  [0.000]  [0.000]  [0.000]  [0.000]
 3    3     7.815   23.207   23.006   20.851   21.133   24.190
                   [0.000]  [0.000]  [0.000]  [0.000]  [0.000]
 4    2     5.991    9.970    9.788    9.089   10.708   10.292
                   [0.007]  [0.007]  [0.011]  [0.005]  [0.006]
```

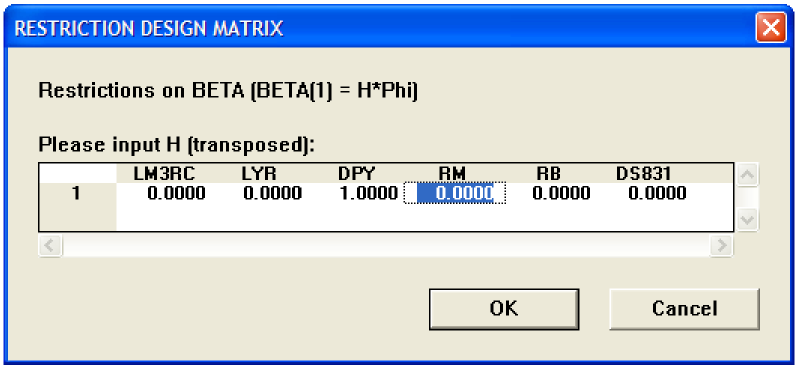
As we are working with a rank of three here, the relevant results are in the $r = 3$ row. The Chi-square test statistics and p-values corresponding to H7, H8, H9 in the text are highlighted in bold above. Compare these to the values in the last two columns of Table 10.3 (page 188) for H7, H8, and H9.

If you want to see the full estimation results as well as the test statistics, you can use the *Restrictions on Subsets of Beta* operation. For this and subsequent tests, we'll switch back to the "Beta=H*Phi" formulation for restrictions, so use the *Change Restriction Formulation* to select that formulation if necessary.

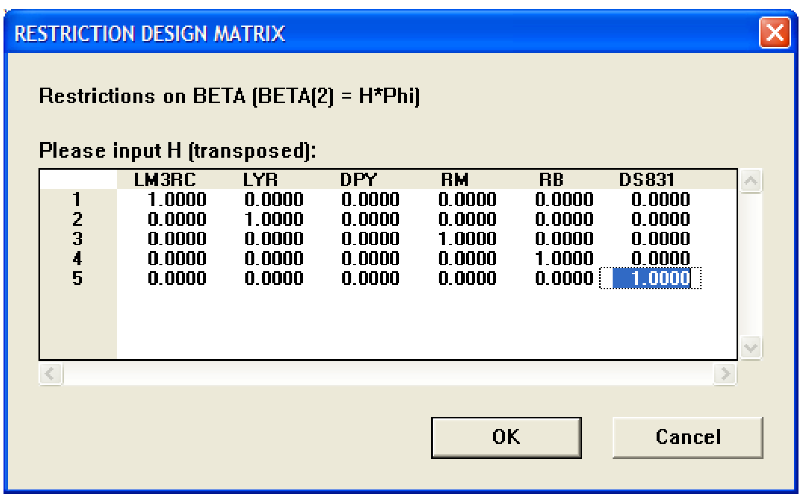
For H7, choose *Restrictions on Subsets of Beta* from the *I(1)* menu as usual, but this time use the dialog box to change the number of subsets to 2, with 1 vector in the first subset and 2 vectors in the second subset:



In the next dialog box, enter 5 as the number of restrictions on subset 1 and click OK. Then, in the following dialog, enter a 1 in the DPY column leaving zeros in all other columns. This imposes a stationarity constraint on DPY by excluding the other five variables from the first cointegration relation. The dialog should look like this:



In the next box, enter 1 as the number of restrictions on subset 2. You will then see the dialog box for the **H**′ matrix for the second subset. Here, enter a single 1 in each column of **H** (using a different row in each column) except for the DPY column, which should have zeros in all rows. This excludes DPY from the other two cointegrating vectors (the two vectors in this subset), leaving the other variables unrestricted (i.e. with one unrestricted ϕ parameter per variable) in both vectors. The dialog should look like this:



When prompted, normalize as indicated in the text (row 1 on DPY, row 2 on LM3RC, and row 3 on RM).

The Chi-Square test result (as shown in Table 10.3) is displayed at the top of the new set of output:

```
TEST OF RESTRICTED MODEL:    CHISQR(7) = 20.836 [0.004]
```

and you can see the results of the restrictions in the Beta vector table:

```
RE-NORMALIZATION OF THE EIGENVECTORS:

THE EIGENVECTOR(s)(transposed)
           LM3RC     LYR      DPY      RM        RB     DS831
Beta(1)    0.000    0.000 197.680    0.000     0.000    0.000
Beta(2) -17.632   10.617    0.000 170.732 -207.694    0.530
Beta(3)  14.333  -18.335    0.000  -61.588    58.380   -4.957
```

Hypotheses H8 and H9 are handled just as above, except that the restrictions involve RM in H8 and RB in H9. For example, for H8 the first $\mathbf{H}'$ matrix will have the "1" in the RM column (rather than in the DPY column) while in the second $\mathbf{H}'$ matrix the RM column will have zeros in all rows.

10.5.2 Table 10.3, Hypotheses H10, H11, and H12

These three hypotheses are similar to the previous three, except that the stationarity restrictions involve a pair of variables in each case.

H10 tests the stationarity of the real deposit rate, defined as $DPY_t - RM_t$. To test this, we again do *Restrictions on Subsets of Beta* and select the same number and allocation of subsets as above (2 subsets, with 1 vector in the first subset and 2 vectors in the second).

Enter 5 restrictions on subset 1, and this time use the following $\mathbf{H}'$ matrix definition:

Subset one: $\mathbf{H}' = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix}$

Enter one restriction on subset 2, and enter the following $\mathbf{H}'$ matrix:

Subset two: $\mathbf{H}' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Normalize vectors 1 and 3 on DPY, vector 2 on LM3RC.

The process for H11, which tests the real bond rate ($DPY_t - RB_t$) is virtually identical, but with the $\mathbf{H}'$ matrix on subset 1 (5 restrictions) defined as follows:

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

$\mathbf{H}'$ on subset 2 is defined similar to that used for H10 above, except with the column of zeros in the 5th (RB) column, rather than in the 4th (RM) column.

Also, this time, normalize vector 1 on DPY, vector 2 on LM3RC, vector 3 on RM.

For H12 (which tests the interest rate spread, $RM_t - RB_t$) everything is the same as above except for the $\mathbf{H}'$ matrix on the first subset, which should be:

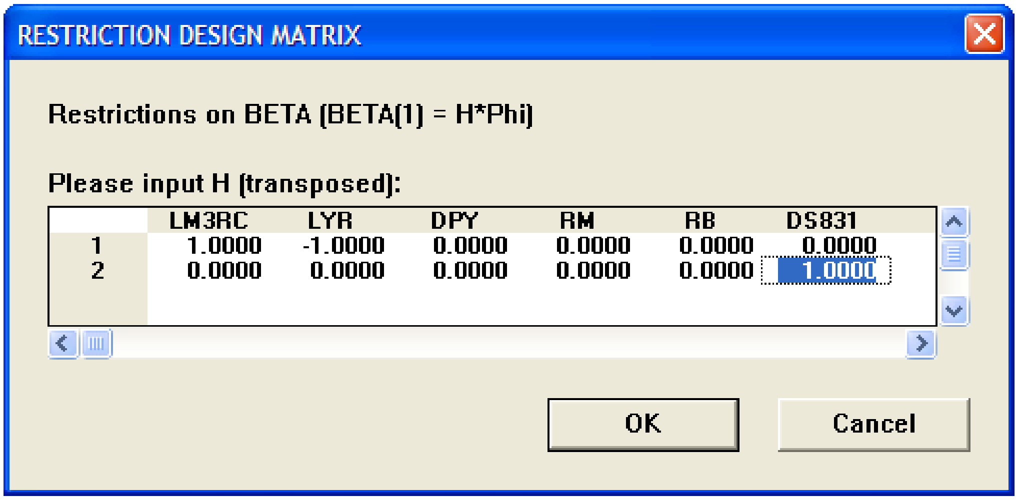
$$\mathbf{H}' = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}$$

and the normalization: here, you want to normalize vectors 1 and 3 on RM, vector 2 on LM3RC.

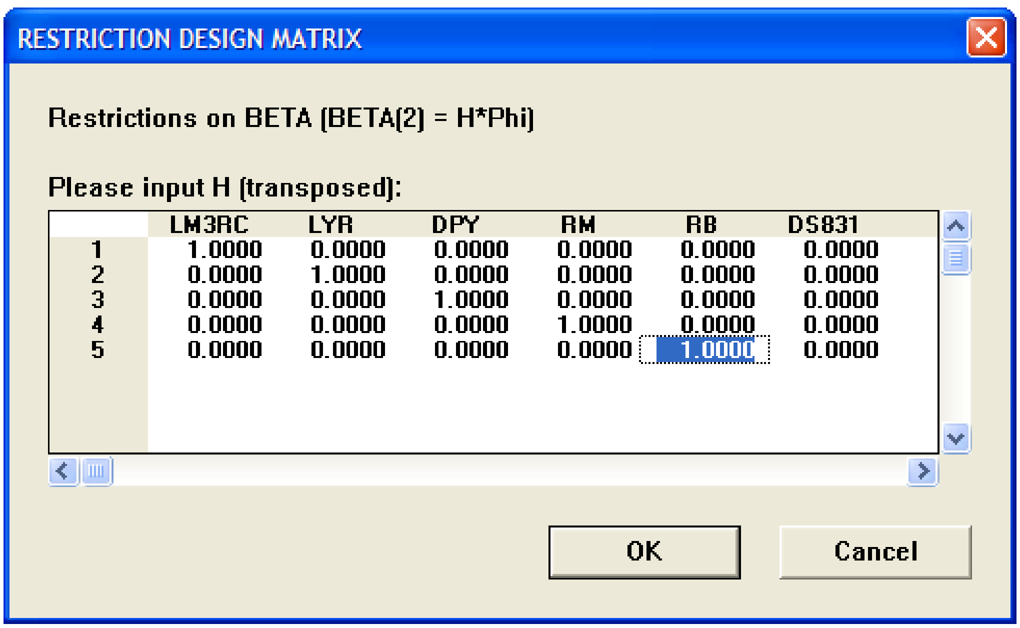
10.5.3 Table 10.3, Hypotheses H13 through H16

For H13, we want to test the stationarity restriction on the liquidity ratio ($LM3RC_t - LYR_t$), but also allow for a shift in the equilibrium mean by including the DS831 variable in the first vector. Thus we will only be imposing four restrictions on the first subset, rather than five.

As before, use *Restrictions on Subsets of Beta* and define 2 subsets, with 1 vector in the first subset and 2 vectors in the second. For the first subset, enter 4 as the number of restrictions. Set up the first $\mathbf{H}'$ matrix as follows:



For the second subset, enter 1 as the number of restrictions, and setup the second $\mathbf{H}'$ matrix as follows—note that the one restriction being imposed is the exclusion of DS831, accomplished by putting zeros in all the rows for that column.



Normalize the model on LM3RC, DPY, and RM, respectively. Here’s the first portion of the output:

```
TEST OF RESTRICTED MODEL:    CHISQR(6) = 1.491 [0.960]

RE-NORMALIZATION OF THE EIGENVECTORS:

THE EIGENVECTOR(s)(transposed)
           LM3RC     LYR      DPY        RM       RB     DS831
Beta(1)   16.711 -16.711     0.000     0.000     0.000 -5.586
Beta(2)  -12.037   3.960  -201.759    49.898  -105.218  0.000
Beta(3)   13.033 -20.127   -26.180  -524.419   360.666  0.000
```

```
THE MATRICES BASED ON 3 COINTEGRATING VECTORS:

BETA(transposed)
          LM3RC    LYR    DPY     RM      RB   DS831
Beta(1)   1.000 -1.000 0.000  0.000  0.000 -0.334
Beta(2)   0.060 -0.020 1.000 -0.247  0.522  0.000
Beta(3)  -0.025  0.038 0.050  1.000 -0.688  0.000
```

You can see that this imposes the liquidity ratio restriction on LM3RC and LYR while allowing for the DS831 term in the first vector, while excluding DS831 from the other two vectors.

H14 adds another variable (the inflation rate, DPY) to the first cointegrating vector. So, the procedure is the same as above, except that now we only have 3 restrictions on the first subset. For this case, the $\mathbf{H}'$ matrix would be:

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

The second $\mathbf{H}$ matrix is the same as for H13 (one restriction, with zeros in column for DS831), and we again normalize on LM3RC, DPY, and RM.

H15 uses the same number of subsets and restrictions, but incorporates the interest rate spread rather than the inflation variable. So, $\mathbf{H}'$ on subset one again has 3 restrictions, but is formulated as:

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

Everything else is done as in H14.

H16 includes both the inflation rate and interest rate spread, so we are now down to two restrictions on subset 1, with the $\mathbf{H}'$ as follows:

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

Again, everything else is unchanged from the previous case.

10.5.4 Table 10.3, Hypotheses H17 through H20

Hypotheses H17 through H20 involve tests on real aggregate income. All of these involve 3 restrictions on the first cointegrating vector (subset 1), so the setup is similar to H14 and H15. For H17, use the following $\mathbf{H}'$ matrix:

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

Use the same $\mathbf{H}'$ matrix as before on the second subset (excluding DS831), but this time, normalize on LYR, DPY, and RM.

For H18, everything is identical to H17, except for the $\mathbf{H}'$ matrix on subset one:

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

Again, H19 is identical to H18 except for the $\mathbf{H}'$ matrix on subset 1:

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

Finally, the $\mathbf{H}'$ matrix on subset 1 for H20 is:

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

10.5.5 Table 10.3, Hypotheses H21 through H24

These hypotheses test restrictions on the inflation rate, real interest rates, and the spread, with the inclusion of the shift variable. You will use four restrictions on the first subset for all four tests.

For H21, the $\mathbf{H}'$ matrix on the first cointegrating vector is:

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

The matrix on the second subset is unchanged from before (exclude DS831). For this model, normalize on DPY, LM3RC, and RM, respectively.

For H22, use the following $\mathbf{H}'$ matrix:

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

As in H21, normalize on DPY, LM3RC, and RM.

For H23, use this $\mathbf{H}'$ matrix:

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

Normalize on DPY, LM3RC, and RM.

Finally, for H24, use:

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

And, for this model, normalize on RM, LM3RC, and RM.

10.5.6 Table 10.3, Hypotheses H25 through H29

These hypotheses involve tests on various combinations of inflation rate and interest rates. Here, were have 3 restrictions on subset 1. For H25, the $\mathbf{H}'$ matrix is as follows:

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

Normalize on DPY, LM3RC, and RM.

For H26, the $\mathbf{H}'$ matrix is:

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

Again, normalize on DPY, LM3RC, and RM.

H27 requires the following $\mathbf{H}'$ matrix:

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

The normalization is different this time, normalize on RM, LM3RC, and RM, respectively.

For H28, use:

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

Normalize on RM, LM3RC, and RM.

Finally, for H29, use:

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

Normalize on RM, LM3RC, and RM.