

RATS Handbook  
for Katarina Juselius'  
The Cointegrated VAR Model



## Introduction

This Handbook shows you how to use RATS and CATS to reproduce 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. The references to RATS menu operations and menu-driven "Wizards" are based on Version 8 of RATS, and some of these operations are located in different menus, or simply not available, in older versions. However, the instructions presented should work in any 6.2 or later release.

Although we go over some of the basics of using both RATS and CATS, this is not intended as an introductory tutorial for RATS. If you are new to RATS, please begin by working through the tutorial in the *Introduction to RATS* (found in the *User's Guide* for versions prior to 8.0)

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

<b>1. Getting Started .....</b>	<b>7</b>
1.1 Using CATS.....	7
1.2 The Pre-Written Example Programs.....	7
1.3 Getting Started .....	7
1.4 Reading Data .....	8
1.5 The Data, Creating Series, and Data Transformations .....	9
1.6 Examining the data.....	11
<b>2. Detrending and Graphing .....</b>	<b>12</b>
2.1 Detrending Data Using FILTER .....	12
2.1.1 Detrending Using LINREG.....	12
2.2 Graphing the Results, Figure 2.4 .....	13
<b>3. The Unrestricted Vector Autoregression Model .....</b>	<b>14</b>
3.1 Moving Averages and First Differences .....	14
3.1.1 Moving Averages .....	14
3.1.2 First Differences .....	14
3.2 Graphing Levels and Differences, Figures 3.3 and 3.4.....	14
3.3 Unrestricted VAR(2) Model.....	15
3.4 VAR Residuals, Figure 3.5 .....	16
3.5 Roots of the Companion Matrix, Table 3.1 .....	16
3.5.1 Graphing the Roots, Figure 3.6.....	17
3.6 VAR Estimation Results, Table 4.1.....	17
3.7 Computing F-Tests .....	18
3.8 The Vector Error Correction Model, Tables 4.2 through 4.4 .....	19
3.8.1 Using RATS Instructions.....	19
<b>4. Using CATS: Preliminary Analysis .....</b>	<b>20</b>
4.1 Compiling and Executing CATS.....	20
4.1.1 The CATS Wizard .....	20
4.1.2 Using the @CATS Command.....	20
4.2 The VECM Estimation Results, Table 4.2.....	21
4.3 Graphing the Data in CATS, Figures 3.3 and 3.4.....	21
4.4 Graphing the Residuals, Figure 3.5.....	22
4.5 Roots of the Companion Matrix, Table 3.1, Figure 3.6 .....	22
4.6 Residuals, Correlations and Specification Tests, Figures 4.1–4.6, Table 4.5.22	
<b>5. Cointegrated VAR Model.....</b>	<b>23</b>
<b>6. Deterministic Components .....</b>	<b>24</b>
6.1 Figure 6.3 .....	24
6.2 Tables 6.2 and 6.3 .....	24
6.2.1 Seasonal Autocorrelation/ARCH Tests—using the Preferences Dia- log Box.....	25
6.2.2 Interpreting the Test Results.....	25

<b>7. Estimating the I(1) Model .....</b>	<b>27</b>
7.1 New Example File .....	27
7.2 Using SHIFT Rather Than EXO in CATS.....	27
7.3 Table 7.1.....	27
7.4 Figures 7.1 through 7.5.....	28
<b>8. Cointegration Rank.....</b>	<b>29</b>
8.1 Rank Test Statistics, Table 8.1 .....	29
<b>9. Recursive Tests of Constancy .....</b>	<b>31</b>
9.1 Setting the Rank of Pi.....	31
9.2 Recursive Log Likelihood, Figure 9.1 .....	31
9.3 Recursive Trace Tests, Figure 9.2.....	32
9.4 Recursive Eigenvalues and Fluctuation Tests, Figures 9.3-9.5 .....	32
9.5 Beta Constancy Tests, Figures 9.6 and 9.7 .....	32
9.6 Changing the Estimation Sample, Figure 9.8.....	32
9.7 Another Estimation Sample, Figure 9.9.....	32
9.8 Prediction Errors, Figures 9.10-9.12.....	32
9.9 Backwards Recursive Tests, Figures 9.13-9.21.....	32
<b>10. Testing Restrictions on Beta.....</b>	<b>34</b>
10.1 Normalizing.....	34
10.2 Two Restriction Formulations.....	34
10.3 Some Simple Restrictions, Table 10.1 .....	34
10.3.1 Hypothesis H1 Using $H^*Phi$ .....	35
10.3.2 Hypotheses H1 and H2, Using $R^*Beta=0$ .....	36
10.3.3 Hypotheses H3 and H4.....	36
10.3.4 Hypotheses H5, a Joint Test.....	37
10.4 Hypothesis H6 and Table 10.2, Three Restrictions .....	37
10.5 Table 10.3, Unrestricted Constant Model .....	37
10.5.1 Table 10.3, Hypotheses H7, H8, H9, Stationarity.....	38
10.5.2 Table 10.3, Hypotheses H10, H11, and H12 .....	40
10.5.3 Table 10.3, Hypotheses H13 through H16 .....	40
10.5.4 Table 10.3, Hypotheses H17 through H20 .....	42
10.5.5 Table 10.3, Hypotheses H21 through H24 .....	43
10.5.6 Table 10.3, Hypotheses H25 through H29 .....	44
<b>11. Testing Restrictions on <math>\alpha</math>.....</b>	<b>45</b>
11.1 Testing for Weak Exogeneity.....	45
11.1.1 Tests of Long-run Weak Exogeneity .....	45
11.2 Models with Weakly Exogenous Variables .....	46
11.3 Testing a Known Vector in $\alpha$ .....	46
<b>12. Identification of the Long-run Structure .....</b>	<b>47</b>
12.1 Just-identifying Restrictions, Table 12.1 .....	47
12.1.1 Example One.....	47
12.1.2 Example Two.....	47

12.2	Over-identifying Restrictions, Table 12.3 .....	48
12.2.1	Example Three .....	48
12.2.2	Example Four .....	48
12.3	Lack of Identification, Table 12.4.....	49
12.3.1	Example Five.....	49
12.3.2	Example Six.....	49
12.4	Recursive tests of $\alpha$ and $\beta$ , Figures 12.1–12.7 .....	50
<b>13.</b>	<b>Identification of the Short-Run Structure .....</b>	<b>51</b>
13.1	The Data .....	51
13.2	Creating Error Correction Terms and First Differences.....	51
13.3	Estimating the VECM Model, and F-Tests.....	52
13.4	Restrictions on the Short-Run Model, Tables 13.1 and 13.2.....	52
13.5	The VAR in Triangular Form, Tables 13.3 and 13.4 .....	53
13.6	Tables 13.5-13.9 .....	54
<b>14.</b>	<b>Identification of Common Trends.....</b>	<b>55</b>
14.6.1	The Unrestricted MA Representation .....	55
14.6.2	The MA Representation Subject to Restrictions on $\alpha$ and $\beta$ .....	56
14.6.3	Remaining Sections of Chapter 14 .....	57
<b>15.</b>	<b>Identification of a Structural MA Model.....</b>	<b>58</b>
15.1	Structural Models and Variable Order.....	58
15.2	Structural MA Model Illustration .....	58
15.3	Are the Labels Credible?.....	61
15.3.1	Generating Table 15.1—SVAR 1 Model.....	61
15.3.2	Generating Table 15.1—SVAR 2 Model .....	61
15.3.3	Generating Table 15.1—VAR Model.....	62
<b>16.</b>	<b>Analyzing I(2) Data with the I(1) Model.....</b>	<b>63</b>
16.1	Stochastic and Deterministic Trends .....	63
16.1.1	Figure 16.1 .....	63
16.1.2	Figure 16.2 .....	63
16.2	I(2) Symptoms in I(1) Models.....	64
16.2.1	Dummy Variables .....	64
16.2.2	Characteristic Roots.....	64
16.2.3	Graphing Cointegration Relations.....	65
16.3	Evaluating the Nominal-to-Real Transformation .....	65
16.3.1	Testing Long-Run Price Homogeneity.....	65
<b>17.</b>	<b>Estimating the I(2) Model .....</b>	<b>69</b>
17.1	Estimating the I(2) Model and Determining Rank .....	69
17.2	Setting the Reduced Ranks .....	69
17.3	Graphing Cointegrating Relations.....	69
17.4	Identifying Restrictions .....	69

<b>18. Testing Hypotheses on the I(2) Model .....</b>	<b>71</b>
18.1 Testing Price Homogeneity .....	71
18.1.1 Long-run Price Homogeneity .....	71
18.2 Medium-run Price Homogeneity .....	71
18.3 Assessing I(1) Results Within the I(2) Model .....	72
18.3.1 Restrictions of the I(1) Model, Tables 18.2 and 18.3 .....	72
18.3.2 Restrictions on Beta, Table 18.4 .....	72
18.3.3 Data Consistent Long-Run Structure, Table 18.5 .....	73
<b>19. Specific-to-General and General-to-Specific.....</b>	<b>74</b>
19.1 Figures 19.2 and 19.3 .....	74
19.2 Figure 19.4.....	75
<b>20. Wage, Price and Unemployment Dynamics.....</b>	<b>76</b>
20.1 The Data .....	76
20.2 Nominal Versus Real Models .....	77
20.3 The EMS Period.....	77
20.3.1 Table 20.1.....	78
20.3.2 Tables 20.2 and 20.3.....	78
20.3.3 Tables 20.4.....	78
20.3.4 Identifying the Long-Run Structure, Table 20.5.....	78
20.4 The Post-Bretton–Woods Regime .....	79
20.4.1 Specification Tests, Tables 20.6 and 20.7 .....	80
20.4.2 Investigating the $\Pi$ Matrix, Table 20.8.....	80
20.4.3 Identification, Table 20.9.....	80
<b>21. Foreign Transmission Effects.....</b>	<b>81</b>
21.1 The Data .....	81
21.1.1 Figure 21.1, Price Differential and PPP .....	81
21.1.2 Figure 21.2, Interest Rate Differentials .....	81
21.1.3 Figure 21.3, Inflation Rate Differential.....	82
21.1.4 Figure 21.4, Long-Short Spreads.....	82
21.1.5 Figures 21.5 and 21.6, Real Long- and Short-Term Bond Rates ...	82
21.1.6 Figure 21.7, Real Interest Rate Differentials .....	83
21.2 The Models.....	83
21.2.1 Dummy Variables .....	83
21.2.2 The Two Cointegration Models .....	84
21.2.3 Tables 21.1 and 21.2.....	84
21.2.4 Tables 21.3 – 2.15, Long Run Relations .....	84
<b>22. Collecting the Threads.....</b>	<b>86</b>
22.1 The Data and Model.....	86
22.2 Tables 22.1 and 22.2.....	87
22.3 Table 22.3 .....	88

# 1. Getting Started

## 1.1 Using CATS

The first thing to note is that CATS is *not* a stand-alone application. Rather, it is a set of procedures written in the RATS language. So, to use CATS, you first launch the RATS program and use RATS instructions or menu operations to read in your data and do any preliminary analysis or data transformations.

You then execute the CATS procedure using the *CATS Cointegration* operation on the *Time Series* menu in RATS, or by directly executing a command like this:

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

The CATS procedure actually takes control of the program and adds several additional drop-down menus to the RATS menu bar. You do your cointegration analysis by selecting operations from these menus, and providing the necessary input using various pop-up dialog boxes. You have the option of saving your model, exiting CATS, and continuing your analysis or generating reports using additional RATS instructions.

If you are not familiar with using procedures in RATS, see the RATS manuals for general instructions. For example, if you have Version 8 of RATS, see pages 32-33 of the *Introduction* and Section 15.2 of the *User's Guide*.

## 1.2 The Pre-Written Example Programs

All of the steps you will need to follow to reproduce (most of) the results in the textbook are described here in this handbook. We have also provided pre-written program files containing the necessary commands as well as comments describing the necessary menu operations and other steps.

In general, you will probably find it convenient to use these example files for getting data loaded and launching CATS with the appropriate settings. You can then work through the handbook for details on the subsequent menu operations and other input.

However, if you are just getting started with RATS and CATS, we recommend that you work through at least the first three chapters of this handbook step-by-step, entering the instructions and using the relevant menu-driven Wizards yourself. This hands-on approach should get you up to speed more quickly and better prepare you for working with your own data sets and models. You can always refer to the provided program files for help if you run into any problems.

The instructions and other information for Chapters 1 through 6 of the textbook (and handbook) are provided in the file `CATSHandbook_Ch1_Ch6.rpf`, while code for Chapters 7 through 10 of the textbook is provided in the file `CATSHandbook_Ch7_Ch10.rpf`. For the rest of the text, we use individual example programs for each chapter. The file `CATSHandbook_DataRead.src` includes all of the code needed to read in the data set and do the data transformations for the models used in the majority of the text.

## 1.3 Getting Started

Most of the examples in *The Cointegrated VAR Model* book use a common data set—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. If you are going to use the pre-written program file, open the `CATSHandbook_Ch1_Ch6.rpf` file and set it as the Input window. Otherwise, just start with an empty Input window.

Either way, we recommend setting up the program with separate Input and Output windows, with the windows tiled so that both are visible on screen. 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. See the *Introduction to RATS* book (*Getting Started* for Version 7 or older) if you don't know how to do that.

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).

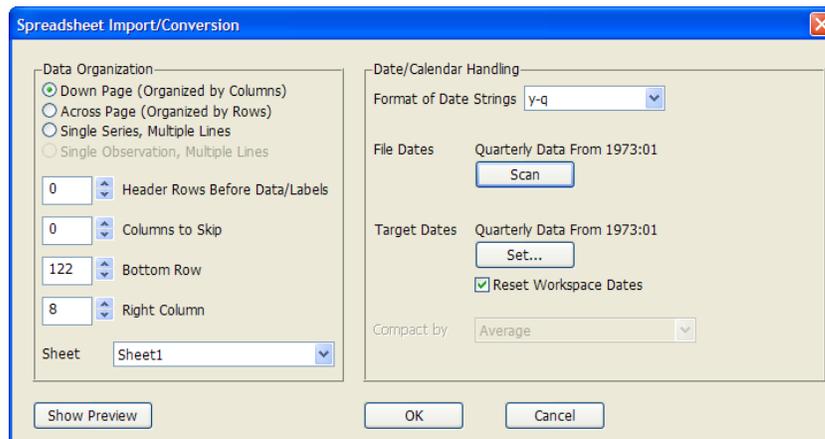
## 1.4 Reading Data

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 2.0–2003 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 the one shown below. Click on the “Scan” button to scan the file for dates, which it should find and report.<sup>1</sup> In the screen shot below, you can see the date information detected using the “Scan” button:



In this case, RATS has correctly determined the frequency and starting date from the information on the file, so just click on OK to read the data. This will generate and execute the 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
open data book.xls
data (format=xls,org=columns) 1973:1 2003:1 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. **OPEN DATA** specifies the data file to be read, while **DATA** actually reads in the data. The **FORMAT** and **ORG** options describe the structure of the file (XLS, organized in columns). This is followed by the range of observations and list of series to read. In most cases, you can use

<sup>1</sup> If you are using an older version of the Data Wizard that doesn't have a “Scan” button, RATS will scan the file automatically after you click OK in the first box. It will then display a second dialog that you can use to confirm or set the date information. Click OK on that second box to go ahead and read the data.

a “/” wildcard in place of the dates to let RATS determine range. Because we want all the series on the file, we could also omit the list of series names (and the date range), like this:

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

## 1.5 The Data, Creating Series, and Data Transformations

Here’s how the series in the file relate to the notation used in the textbook. First, here are the five series that we will use in the most of the examples:

Series	Description
LM3RC	The (corrected) log of the real M3 money stock ( $m_t^r$ in the book)
DPY	The quarterly inflation rate ( $\Delta p_t$ )
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 ( $\Delta p_t$ ) is defined as

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

DPY is already available on the data file as noted above, but 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*:

The screenshot shows a dialog box titled "Differencing Data". It has a blue title bar with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Input Series:** A dropdown menu with "LPY" selected.
- Output Series:** A dropdown menu with "DPY" selected.
- Sample Start and End:** Two empty text boxes with small arrows on the right side, with the text "(Leave Blank for Full Range)" centered above them.
- Type:** A dropdown menu with "One Regular" selected.
- Buttons:** "OK" and "Cancel" buttons at the bottom center.

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 (see pages 102-103 of the text for details on the dummy-variable terminology used by Juselius).

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 - .5*(t==1976:1.or.t==1976:2)
```

This uses logical expressions to set the series. The reserved variable  $t$  returns the entry being set, so the first logical expression returns a one ("true") for 1975:4, and zero ("false") elsewhere. The expression in parentheses returns a 1 for 1976:1 or 1976:2, which is then multiplied by  $-0.5$  to give the desired value for those two periods.

The second dummy variable is a "permanent intervention" dummy in Juselius' terminology, with a one for 1976:4 and zeros elsewhere. This can be done as:

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

We need another dummy (a "mean-shift 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.

## 1.6 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 you've read into memory. This will help ensure there were no mistakes in the data file 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 easiest way to do that is to select the *Series Window* from the *View* menu. This displays a list of all the series in memory, along with some basic information like the number of observations and date range of each series. You can double-click on an individual series to view the numbers in that series. If you select (highlight) one or more series, you can use various operations on the *View* menu (or the corresponding toolbar icons) to examine the data.

For example, selecting *View-Data Table* displays all of the selected series in a spreadsheet-style window. Scroll through the data to make sure it looks correct.

*View-Time Series Graph* will display a graph of all the selected series. This provides a quick way to check the overall behavior of the series.

*View-Statistics* displays some basic summary statistics (number of observations, mean, standard error, minimum value, and maximum value). Check for anything out of the ordinary, like unexpectedly small or large values, or differing observation counts.

You can also use instructions like **PRINT**, **TABLE**, and **GRAPH** to examine your data. For example:

```
print
table
print(window="Interest Rates") / Rm Rb
graph(header="Interest Rates") 2
# Rm
# Rb
```

## 2. Detrending and Graphing

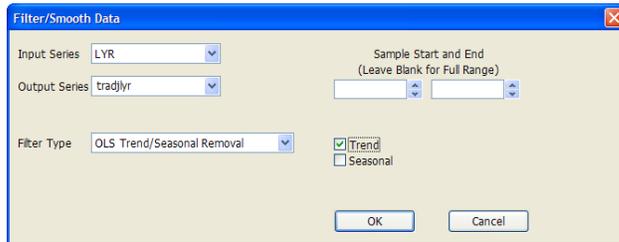
In this chapter, we demonstrate some simple filtering techniques and some of the graphing commands by reproducing Figure 2.4, from page 25 of *TCVM*.

### 2.1 Detrending Data Using FILTER

First, we need to create the “trend-adjusted” price and income series used in Figure 2.4. The easiest way to do that in RATS is to use the *Filter/Smooth Wizard* on the *Data/Graphics* menu, or the corresponding **FILTER** instruction.

To use the Wizard, start by selecting the *Filter/Smooth* operation.

In the dialog box, select LYR (income) as the input series. We’ll call the trend-adjusted series TRADJLYR, so type that in as the output series name. Select “OLS trend/Seasonal Removal” from the list of filter types, and turn on the “Trend” checkbox. The dialog should look like this:



Click “OK” to execute the filter. The instruction generated by the Wizard will be as follows:

```
filter(remove=trend) lyr / tradjlyr
```

To repeat the same process for the LPY (price) series, repeat similar steps using the Wizard, or just execute the following command:

```
filter(remove=trend) lpy / tradjlyr
```

#### 2.1.1 Detrending Using LINREG

If you wanted to do the trend-removal regression “manually” using a least squares regression, you would first need to create a trend series:

```
set trend = t
```

And then use **LINREG** to regress the variable to be filtered on a constant and the trend. The trend-adjusted series would be the residuals from this regression.

For example, you could create the TRADJLYR series either by setting it equal to the residuals that are automatically saved in %RESIDS, or by supplying the variable name as the *residuals* parameter on the **LINREG** instruction:

```
linreg lyr
# constant trend
set tradjlyr = %resids
```

or

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

## 2.2 Graphing the Results, Figure 2.4

Now that we've computed our trend-adjusted series, we can graph them.

As noted earlier, one way to graph series is to display the Series Window and use operations like *View-Time Series Graph* or the corresponding toolbar icon. Here, though, we want to create actual **GRAPH** instructions that can be saved as part of the program for reuse. We can either type those commands directly, or we can use the *Graph* operation on the *Data/Graphics* menu. The *Graph Wizard* is described in some detail in the *Introduction to RATS* book, so here we will focus on the **GRAPH** instructions themselves.

To graph the series separately, you can do something like:

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

and

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

To draw a single graph showing both series, we can add "2" as a parameter indicating the number of series to be graphed and then list the two series on separate supplementary cards. Here, we have also added the **KEY** option to include a graph key:

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

To combine multiple graphs into a single figure or page, as in Figure 2.4 from page 25 of the *TCVM*, we use **SPGRAPH** (SPecial GRAPH) instructions.

Here, we use the **VFIELDS** option on the first **SPGRAPH** to divide the graph page into three vertical fields (rows). Then we do the three **GRAPH** instructions that actually create the graphs we want to include. Here, the third command graphs the series **DPY**, which is the inflation rate (i.e. first differences of the price series). The **SPGRAPH(DONE)** at the end 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")
# tradjlp
graph(header="Trend adjusted real agg. income: stochastic I(1) trend")
# tradjlyr
graph(header="Inflation rate: stochastic I(1) trend")
# dpy
spgraph(done)
```

### 3. The Unrestricted Vector Autoregression Model

Chapters 3 and 4 of the text formally introduce the Vector Autoregression and Vector Error Correction models, and begin the process of analyzing the model that will be used for the majority of the empirical examples.

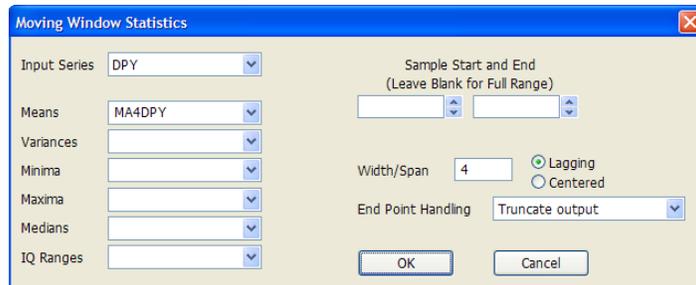
In this chapter of the Handbook, we will generate some of the graphs shown in Chapter 3 of the text, and show how to define and estimate VAR and VECM models, as described in Chapter 3 and the early portions of Chapter 4 of the textbook. We will cover the remainder of Chapter 4 of the textbook in Chapter 4 of the Handbook, where we introduce the use of CATS. Note that most of the results presented in this chapter can also be easily produced from within CATS, as we will demonstrate later.

#### 3.1 Moving Averages and First Differences

We've already shown how to read in the data that we will be using, but before generating Figures 3.3 and 3.4 from the text, we need to create some additional series.

##### 3.1.1 Moving Averages

First, we need to compute the four-quarter moving average of inflation, which is plotted in the lower-right corner of Figure 3.3. We can do this using the *Moving Window Statistics* Wizard on the *Data/Graphics* menu. After selecting the Wizard, select DPY as the input series and type in a name for the “means” output series—we'll use MA4DPY for this. Enter “4” as the “width/span”, as we want four-quarter averages. The dialog box should look like this:



Click “OK” to execute the Wizard, which generates the following **MVSTATS** command:

```
mvstats (width=4, means=ma4dpy) dpy
```

##### 3.1.2 First Differences

We also need to create first differences of five variables. These differenced series will serve as the dependent variables in the error-correction model. We will use the **DIFFERENCE** instruction (see page 9) to create these:

```
diff lm3r / dlm3r
diff lyr / dlyr
diff dpy / ddp
diff rm / dr
diff rb / drb
```

#### 3.2 Graphing Levels and Differences, Figures 3.3 and 3.4

We can now generate Figures 3.3 and 3.4. We'll use the **SPGRAPH** and **GRAPH** commands introduced earlier, with some additional options for labeling the graphs.

By default, RATS fills multi-column **SPGRAPHS** in column order, so we do the three **GRAPH** instructions for the first column first, followed by the **GRAPH** instructions for the second column.

First, for Figure 3.3:

```
spgraph(vfields=3,hfields=2,footer="Fig 3.3 The Danish data in levels.")
graph(header="The log of real M3")
# lm3r
graph(header="The long-term government bond rate")
# rb
graph(header="The quarterly inflation rate")
# dpy
graph(header="The log of real GNE")
# lyr
graph(header="An average deposit rate")
# rm
graph(header="A four quarter moving average of inflation")
# ma4dpy
spgraph(done)
```

For Figure 3.4:

```
spgraph(vfields=3,hfields=2,$
footer="Fig 3.3 The Danish data in first differences.")
graph(header="Changes in real M3")
# dlm3r
graph(header="Changes in the deposit rate")
# drm
graph(header="Changes in the inflation rate")
# ddp
graph(header="Changes in real GNE")
# dlyr
graph(header="Changes in the long-term government bond rate")
# drb
spgraph(done)
```

### 3.3 Unrestricted VAR(2) Model

As noted on page 59 of *TCVM*, the unrestricted VAR model considered here includes centered seasonal dummies as deterministic variables. We will handle these by creating a single centered seasonal dummy series, using the following instruction:

```
seasonal(centered) Dq
```

We can then use current and lagged values of this variable to give us the desired set of three dummies, along with a constant, as our deterministic variables.

We can now define our VAR model. You could use *VAR (Setup/Estimate)* on the *Time Series* menu to do this, but here we will just type in the necessary instructions. Execute the commands below to define the model:

```
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 we are calling VAR2MOD. The **VARIABLES** instruction provides the list of endogenous variables, while **LAGS** provides the list of lags. We are using the *uncorrected* version of the M3 series here. The **DETERMINISTIC** instruction supplies the list of exogenous (deterministic) variables. Here, we use the contemporaneous value (zero lag) 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 using equation-by-equation ordinary least squares, we use the **ESTIMATE** instruction. Here, we include a **RESIDS** option to save the residuals in a **VECTOR** of **SERIES**, which we can use to produce Figure 3.5.

```
estimate (resids=varresids)
```

### 3.4 VAR Residuals, Figure 3.5

We will take a closer look at the estimation results below, but first we will graph the residuals as shown in Figure 3.5. Having the residuals stored in a vector of series makes this easy, as we can just loop over the elements in the array. A very simple approach would be:

```
do i=1,5
  graph
  # varresids(i)
end do i
```

But to get the graphs in a single page, we need to add **SPGRAPH** instructions. We can also use the **%MODELLABEL** function to provide a header for each graph:

```
spgraph(vfields=3,hfields=2)
do i=1,5
  graph(header="Residuals for "+%modellabel(var2mod,i))
  # varresids(i)
end do i
spgraph(done)
```

The expression **%modellabel(var2mod,i)** returns the name of the dependent variable of equation *I* in the model **VAR2MOD**. We concatenate this with the “Residuals for ” prefix to create the desired labels.

### 3.5 Roots of the Companion Matrix, Table 3.1

Computing the roots of the companion matrix as discussed on pages 50-51 of the text is quite easy. First, we use the **%MODELCOMPANION()** function to derive the companion matrix from the estimated model, then we use the **EIGEN** instruction to compute the eigen decomposition of this array. The **CVALUES** option saves the complex eigenvalues as a **VECTOR** of **COMPLEX** numbers.

```
compute companionmat = %modelcompanion(var2mod)
eigen(cvalues=compval) companionmat
```

We can get the modulus values by defining a vector of the appropriate size, and then using an **EWISE** instruction to do an elementwise computation on the array:

```
declare vector absval(%rows(compval))
ewise absval(i) = %real(%cabs(compval(i)))
```

The **%CABS** function computes the complex absolute value, and then the **%REAL** function extracts the real portion of that result. The **EWISE** command is effectively the same as looping over a set of **COMPUTE** instructions. It automatically sets all entries of the array, where *I* is the number of the row being set.

To quickly view the values of these arrays, you can use **DISPLAY** or **WRITE**. For example:

```
write compval
```

However, to produce a nicely formatted table as shown in Table 3.1, we suggest that you use **REPORT** instructions. **REPORT** is an extremely powerful tool for creating formatted output, and (perhaps more importantly) for making it easy to *accurately* export sets of results for use in publications or other documents.

Here is one way of creating the report, which use a **DO** loop to loop over the rows in the arrays:

```
report(action=define,hlabels=|"Real", "Complex", "Modulus" | |)
do i=1,%rows(compval)
  report(atcol=1) %real(compval(i)) %imag(compval(i)) absval(i)
  report(action=format,picture="###.##")
end do
report(action=show)
```

The first line defines a new report, and adds the three column labels. We then loop over the number of rows in our arrays. The third line adds one value from each array to a new row in the report, always starting in column one of the report. The next **REPORT** command defines the formatting of the row just added using a “picture” code. The code “###.##” tells RATS to use three letter-spaces to the left of the decimal for each value, and to round to exactly two digits after the decimal. We then close the loop, and the last line displays the array. We could display the report to a spreadsheet-style window by doing:

```
report(action=show,window="Roots of the VAR(2) model")
```

If desired, you could then export the contents of this window to a file in any of several formats by doing *File-Export*. Or you could export the report to a file automatically by adding **UNIT** and **FORMAT** options to the **ACTION=SHOW** operation.

Here’s another way of creating the report. In this version, we add an entire array at a time with **REPORT** instructions, rather than using a loop:

```
report(action=define,hlabels=|"Real", "Complex", "Modulus" | |)
report(atcol=1,fillby=col) compval
report(atcol=3,atrow=1,fillby=col) absval
report(action=format,picture="###.##")
report(action=show)
```

The **FILLBY** options tell RATS to add the vectors as columns rather than as rows. The **ATCOL=3, ATROW=1** options on the third **REPORT** tell RATS to start the modulus values in column three, row one.

### 3.5.1 Graphing the Roots, Figure 3.6

Because RATS already contains sophisticated code for generating plots like the one shown in Figure 3.6, we will only look at a simplified example here.

We can draw this as an x–y scatter plot, with the real value of each root as the x-axis value, and the complex value of the root as the y-axis value. We just need to create series containing the x-axis and y-axis values respectively and then plot them using **SCATTER**. For example:

```
comp numroot = %rows(compval)
set xaxis 1 numroot = %real(compval(t))
set yaxis 1 numroot = %imag(compval(t))
scatter(hmax=1.5,hmin=-1.0,vmax=1.0,vmin=-1.0)
# xaxis yaxis
```

## 3.6 VAR Estimation Results, Table 4.1

The **ESTIMATE** command used earlier produces the standard VAR estimation output, which includes virtually all of the information in Table 4.1, and 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.

### 3.7 Computing F-Tests

The **ESTIMATE** instruction automatically does  $F$ -tests on the hypothesis that each variable can be omitted 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), as well as a global  $F$ -test on all regressors across all equations (bottom of Table 4.1).

These can be computed fairly easily in RATS, as shown below. We won't go into the details of this code—you can refer to the *Reference Manual* for information on the various functions used here. Note the use of **REPORT** instructions to format the output in a convenient table.

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

First, to compute the individual regressor  $F$ -tests:

```
* Get the first equation and the coefficients 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)
```

We use a \$ symbol on the eighth line above to continue the **COMPUTE** instruction onto a second line.

Finally, to compute 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)
```

### 3.8 The Vector Error Correction Model, 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.

#### 3.8.1 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. Recall that we created the necessary first difference series earlier, so we can now define the new VECM system:

```
system(model=vecm1mod)
variables dlm3r dlyr dpy 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 **DETERMINISTIC** 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. 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*.

### 4.1 Compiling and Executing CATS

There are two ways to start the CATS procedure:

- Using the *CATS Cointegration Wizard* on the *Time Series* 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 menu operations within CATS). 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.

#### 4.1.1 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 Cointegration* from the *Time Series* 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.

#### 4.1.2 Using the @CATS Command

As with any RATS procedure, you can also 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
```

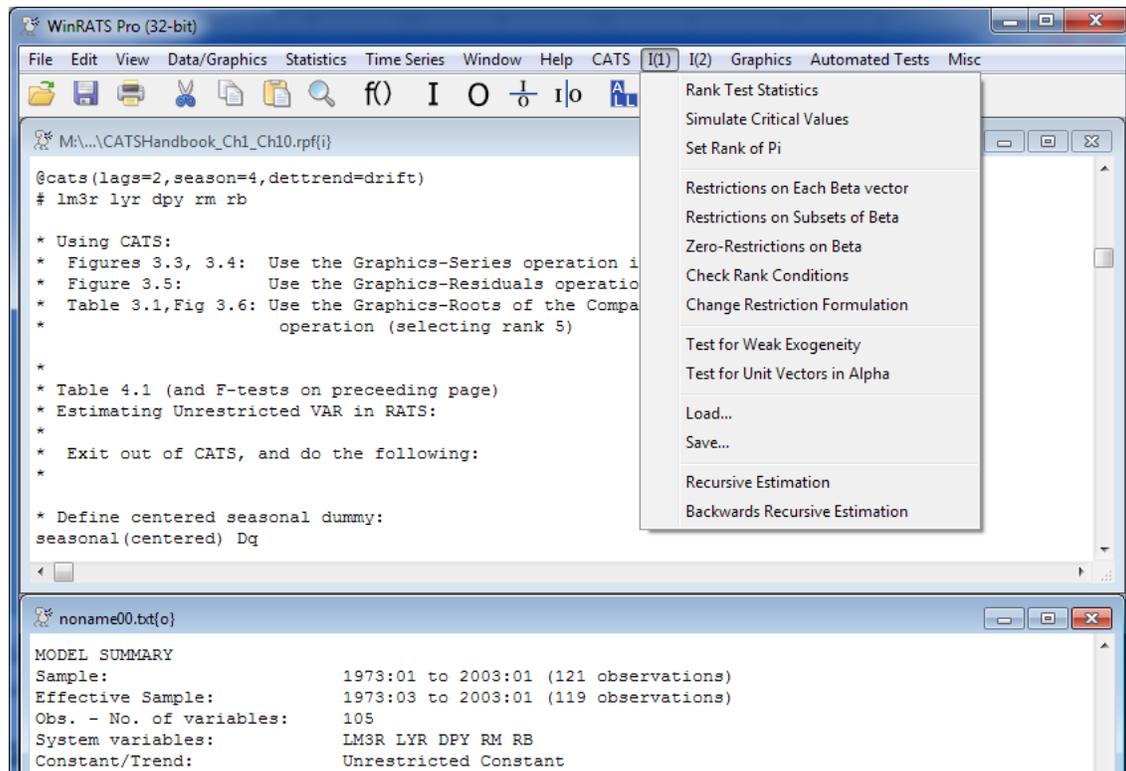
Once you've done that, you can execute CATS. For the model used in Chapters 3 and 4 in the text, use the following instruction:

```
@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. We are still 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.

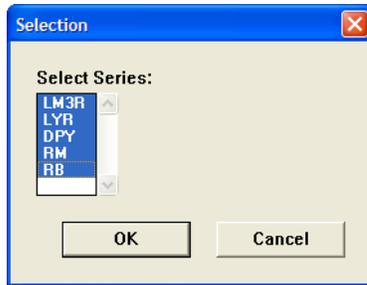
## 4.2 The VECM Estimation Results, Table 4.2

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 earlier, and the estimation results shown in the CATS output should match the unrestricted VECM results produced earlier, and as seen in Table 4.2. To see all of the relevant coefficients, select the *Short Run Parameters* option from the *Misc* menu.

Tables 4.3 and 4.4 can't be reproduced using CATS without making modifications to the code.

## 4.3 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).

#### 4.4 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.

#### 4.5 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.

#### 4.6 Residuals, Correlations and Specification Tests, Figures 4.1–4.6, Table 4.5

You can produce Figures 4.1 through 4.5 (pages 67-69) just as we did above, 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 text.

To produce Figure 4.6 (page 70), use the *Graphics-Correlations* operation, select “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  $\Pi$  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 second (“overlaid”) 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 examines some of the choices available for including deterministic variables in the cointegration 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
```

To get a better look at the effects of removing the seasonality, we can graph just a subsample of the data comparing the corrected and uncorrected nominal series. We do that by including specific date ranges on the supplementary cards:

```
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), by using the `DUM` option and listing these variables on a third supplementary card.

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

The  $\Pi$  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: First, CATS now produces a Ljung-Box test for serial correlation, along with Schwarz and Hannan-Quinn criterion values (which can be used to compare models). It also produces multivariate LM test for ARCH effects, in addition to the individual univariate tests.

Second, by default, the LM tests for autocorrelation and ARCH effects are done for first- and second-order effects, rather than first- and fourth-order (i.e. quarterly seasonal) effects.

### 6.2.1 Seasonal Autocorrelation/ARCH Tests—using the Preferences Dialog Box

You can actually control which lag lengths CATS uses for these tests (along with many other aspects of the program) via the *Preferences* operation on the *CATS* menu. Selecting *Preferences* brings up the dialog box shown below:

The lag-lengths for the autocorrelation and ARCH tests are controlled by the settings in the “AC/ARCH tests order” section, near the middle of the dialog box. To get fourth-order tests, you could replace the “2” with a “4” to get the 4th order *instead* of the 2nd order, or replace the “0” in the third cell with a “4” to get 1st, 2nd, *and* 4th-order autocorrelation and ARCH tests, like this:

Clicking on “OK” will save the selected settings for this session only. If you would like these new settings to be the defaults each time you launch cats, turn on the “Save Settings” check box first, then click on “OK”. See pages 32–40 in the CATS User’s Manual for details on using these settings.

After adding the 4th order test to the settings, select *Misc–Residual Analysis* again to reproduce the analysis including the 4th order tests.

### 6.2.2 Interpreting the Test Results

Looking at the results not covered in the textbook, the multivariate Ljung-Box rejects the null of no autocorrelation, albeit at the fairly long 29-lag horizon used here. And as discussed in the text, the multivariate LM autocorrelation tests indicate a lack of first-order effects, and that the fourth-order (seasonal) effects have been accounted for by switching to the corrected M3 series. There is some evidence of second-order correlation, however.

The multivariate ARCH tests reject the null of no ARCH effects, supporting the violation of Normality indicated by the Normality test. However, as noted in the text, this seems to results mostly from issues with the bond rate series.

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).

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 6.3 are organized into the following separate tables in the CATS output: “Weakly Exogenous/Fixed Variables”, “Time  $t - 1$ ”, “Dummy Variables”, and “Constant”.<sup>2</sup>

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

---

<sup>2</sup> The third coefficient array in Table 6.3 in the textbook appears to have been inadvertently transposed.

## 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 New Example File

The code for chapters seven through 10 is provided on the file `CATSHandbook_Ch7_Ch10.rpf`, so if you want to work through the existing code rather than typing in your own commands, please begin a fresh RATS session using that file.

Note that rather than repeating all of the commands for reading in and transforming data, we have stored a copy of the necessary commands in a separate file called `CATSHandbook_DataRead.src`. We can execute all of those commands using a single **SOURCE** instruction:

```
source CATSHandbook_DataRead.src
```

Then we just need to compile the CATS procedures using another **SOURCE**:

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

### 7.2 Using SHIFT Rather Than EXO in CATS

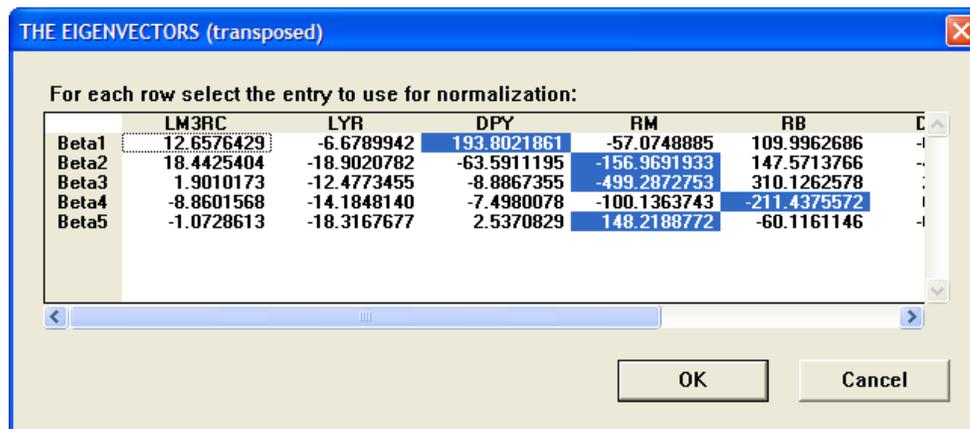
In this chapter, we examine a variation of the previous CATS model, which is executed using the following instructions (if you are continuing from the previous chapter, first exit out of CATS and then restart using these commands):

```
@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).

### 7.3 Table 7.1

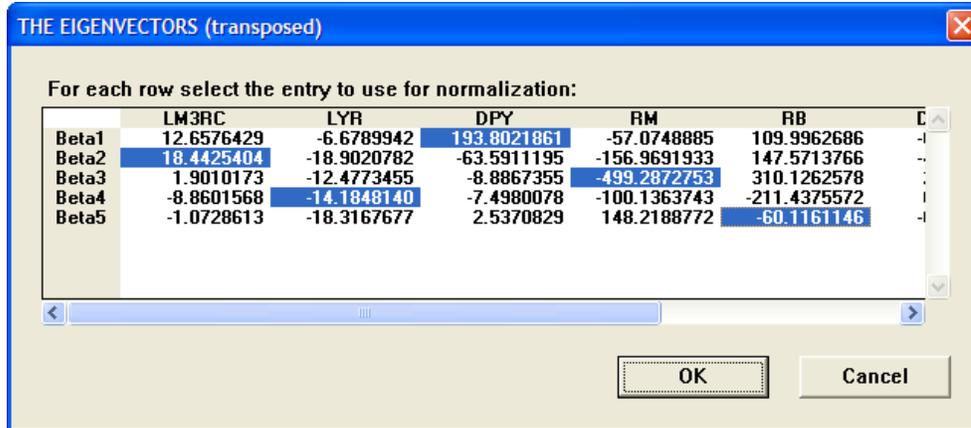
The non-normalized eigenvectors and  $\Pi$  matrix of Table 7.1 (page 123) are shown automatically as part of the initial CATS output. To get the normalized eigenvectors and Alpha matrix, select *Set Rank of Pi* from the *I(1)* menu and enter 5 as the rank (5 is already the default rank, but this operation 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 shown above. To choose a different normalization, you just need to click on (highlight) the desired column in each row.

For the normalizations in Table 7.1, as indicated by the “1.00” values in the “Normalized eigenvectors” section, we need to normalize row 1 on DPY, row two on LM3RC, row three on RM, row four on LYR, and row five on RB, as shown in the dialog box 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  $\Pi$  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.

### 7.4 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  $\lambda_i$  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.

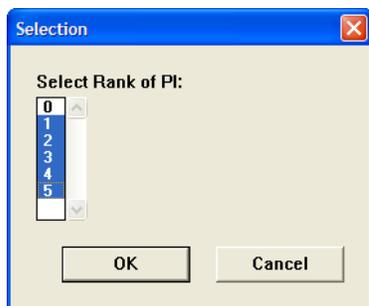
The textbook (page 143) takes you through the model selection process using comparisons against the two  $C_{.95}$  values. Here, we’ll go through the process using the  $p$ -values instead.

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 again 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 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:



and click on “OK”.

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

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

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

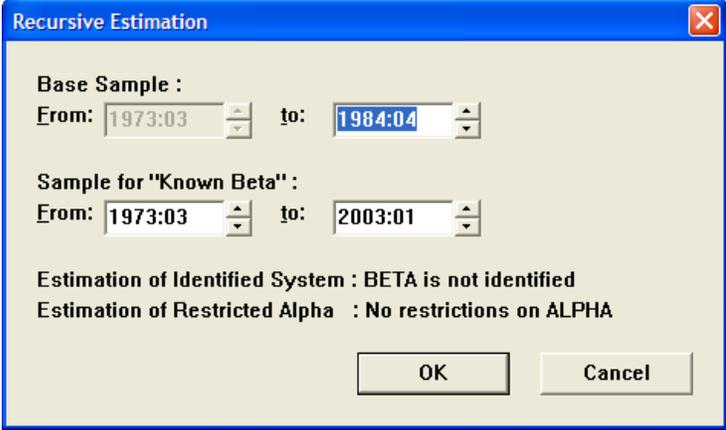
### 9.1 Setting the Rank of Pi

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

Continuing from where we left off, 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:

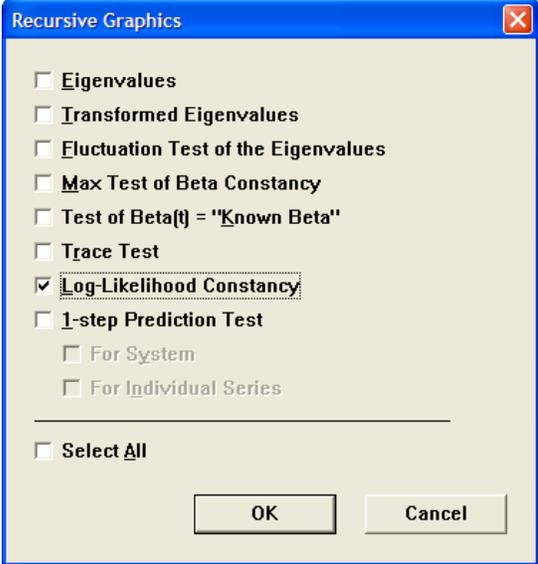


The dialog box titled "Recursive Estimation" has a blue title bar with a close button. It contains the following fields and text:

- Base Sample :**
  - From: 1973:03
  - to: 1984:04
- Sample for "Known Beta" :**
  - From: 1973:03
  - to: 2003:01
- Estimation of Identified System : BETA is not identified
- Estimation of Restricted Alpha : No restrictions on ALPHA

At the bottom are "OK" and "Cancel" buttons.

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).



The dialog box titled "Recursive Graphics" has a blue title bar with a close button. It contains the following options:

- Eigenvalues
- Transformed Eigenvalues
- Fluctuation Test of the Eigenvalues
- Max Test of Beta Constancy
- Test of Beta(t) = "Known Beta"
- Trace Test
- Log-Likelihood Constancy
- 1-step Prediction Test
  - For System
  - For Individual Series

At the bottom are "OK" and "Cancel" buttons.

### 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 generate 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, 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

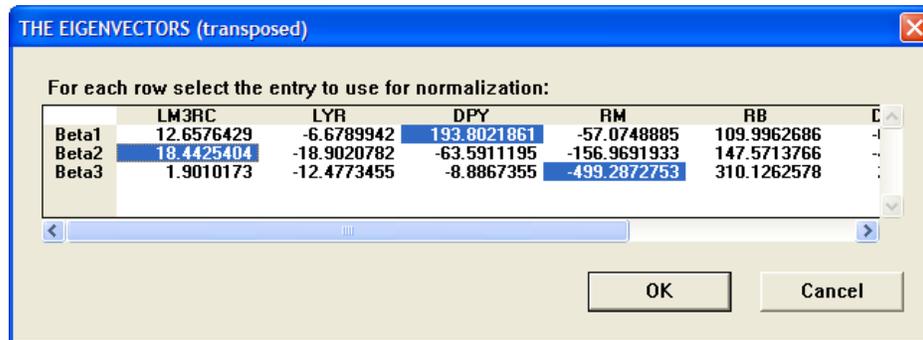
In Chapter 10, we'll look at various ways to impose and test restrictions on the Beta matrix. If you need to restart CATS, recall that we are using the following model and settings:

```
@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, select 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 ones and zeros, 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 ones and zeros, 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 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 **H** $\phi$  gives:

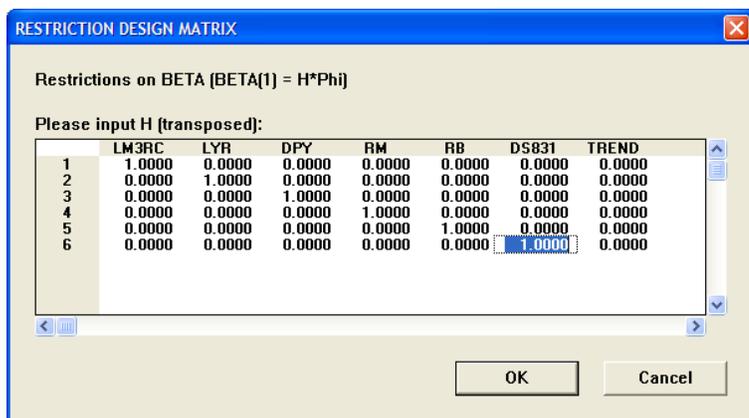
$$H\phi = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \\ \phi_{41} & \phi_{42} & \phi_{43} \\ \phi_{51} & \phi_{52} & \phi_{53} \\ \phi_{61} & \phi_{62} & \phi_{63} \\ 0 & 0 & 0 \end{bmatrix}$$

Note that the last row contains only zeros, which means that there are no free parameters ( $\phi$  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 **H** $\phi$  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 **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 three decimal places. If you want to change 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.

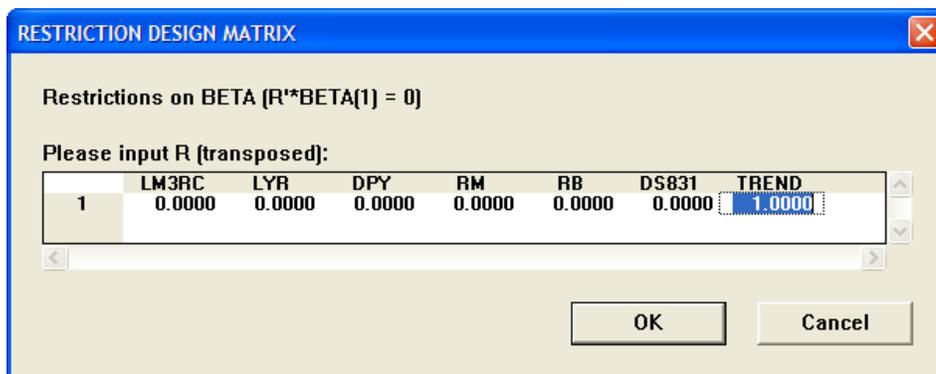
### 10.3.2 Hypotheses H1 and H2, Using $R \cdot \text{Beta} = 0$

Now, we’ll do this using the “ $R \cdot \text{Beta} = 0$ ” form.

Start by choosing the *Change Restriction Formulation* operation from the  $I(1)$  menu and selecting the formulation “ $R \cdot (\text{Beta}/\text{Tau}) = 0$ ” 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 the same as above, except that you will need to put ones in two columns. For H3, you want ones 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 ones 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 “2” 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 ones 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:

	LM3RC	LYR	DPY	RM	RB	DS831	TREND
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
2	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000

Table 10.2 (page 182) compares the unrestricted estimates of  $\Pi$  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 three 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 three 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 three 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  $n_b$  known (i.e. restricted) vectors and  $r - n_b$  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 and four different 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:

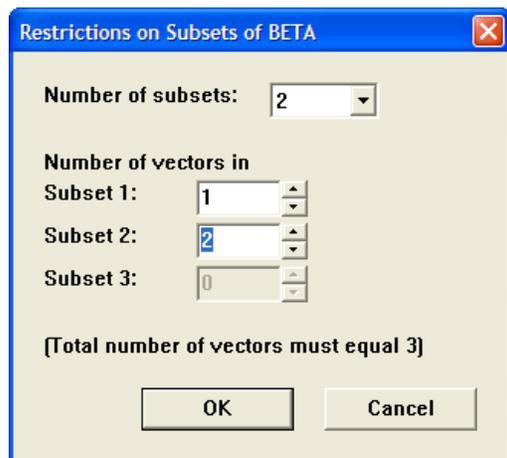
```
TEST OF STATIONARITY
LR-test, Chi-Square(6-r), P-values in brackets.
```

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	<b>20.851</b>	<b>21.133</b>	<b>24.190</b>
			[0.000]	[0.000]	<b>[0.000]</b>	<b>[0.000]</b>	<b>[0.000]</b>
4	2	5.991	9.970	9.788	9.089	10.708	10.292
			[0.007]	[0.007]	[0.011]	[0.005]	[0.006]

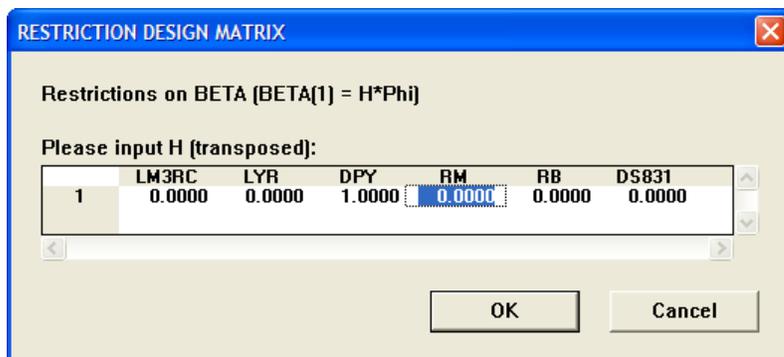
As we are working with a rank of three here, the relevant results are in the  $r = 3$  row. The Chi-square test statistics and  $p$ -values corresponding to H7, H8, H9 in the text are highlighted in bold above. Compare these to the values in the last two columns of Table 10.3 (page 188) for H7, H8, and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

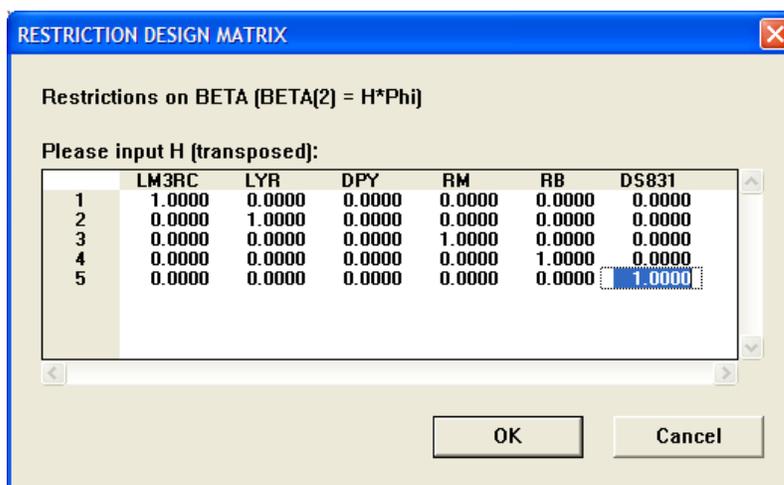
For H7, choose *Restrictions on Subsets of Beta* from the *I(1)* menu as usual, but this time use the dialog box to change the number of subsets to 2, with 1 vector in the first subset and 2 vectors in the second subset:



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  $\phi$  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, by testing the restriction that the first cointegrating vector is exactly  $DPY_t - RM_t$ , with RM being excluded from the other two cointegrating vectors. 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:

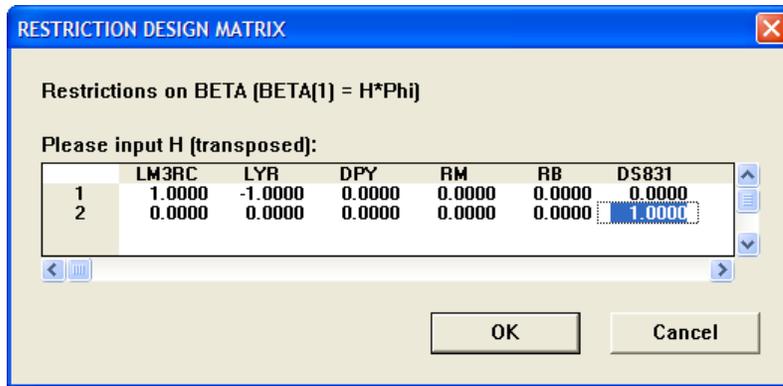
$\mathbf{H}' = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}$

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

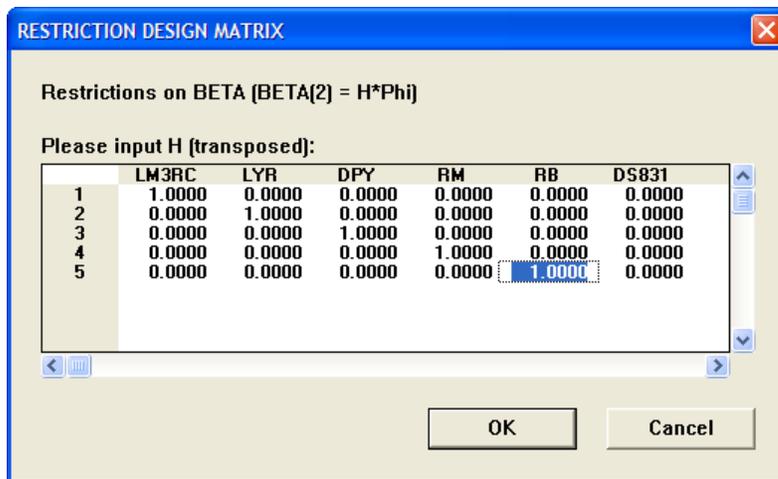
**10.5.3 Table 10.3, Hypotheses H13 through H16**

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 set up the second  $H'$  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, but 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 three 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 1:

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

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, we 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.

## 11. Testing Restrictions on $\alpha$

Here, we'll look at testing restrictions on the  $\alpha$  matrix. If you are starting a new RATS session, you just need to repeat the two **SOURCE** instructions shown on page 27 before continuing with the instructions shown below. See the file `CATSHandbook_Ch11.rpf` for the full example code.

### 11.1 Testing for Weak Exogeneity

We begin with Illustration 11.1.1, which estimates an unrestricted constant model. After loading the data and defining transformations as before, start CATS with the following instructions:

```
@cats (lags=2, shift, dum, detrend=drift) 1973:1 2003:1
# lm3rc lyr dpy Rm Rb
# Ds831
# Dt754 Dp764
```

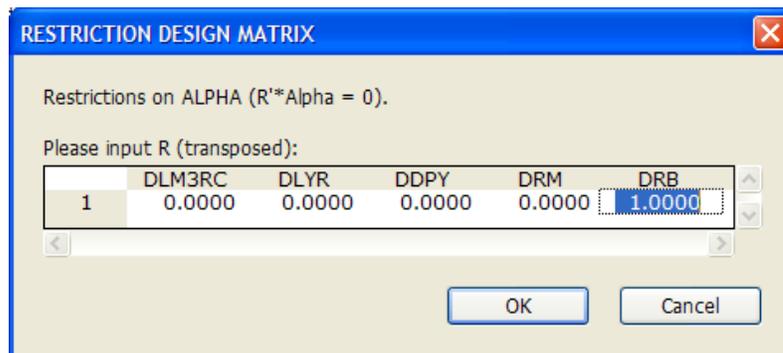
Then set the rank to three, and accept the default normalization.

#### 11.1.1 Tests of Long-run Weak Exogeneity

Next, we will test the hypothesis that RB is weakly exogenous. From the *I(1)* menu, select *Test for Weak Exogeneity*. Input 1 as the number of weakly exogenous variables. The weak exogeneity restrictions are imposed using the form

$$\mathbf{R}'\alpha = 0$$

We need to enter ones in the elements of  $\mathbf{R}$  corresponding to the (first differences of) variables we want to test for exclusion. To test RB, enter a 1 in the DRB column as shown below:



Choose None when asked about restrictions on Beta, and accept the default normalizations. CATS will report the test result as:

```
TEST OF RESTRICTED MODEL:      CHISQR(3) = 4.637 [0.200]
```

and display the recomputed model output. Here, the results indicate acceptance of the hypothesis that the bond rate is weakly exogenous.

CATS provides an easy way to test all of the variables individually, for all possible choices of rank. From the *Automated Tests* menu, select *Weak Exogeneity*. CATS will display Table 11.1 from the book. As noted in the text, the results show that the tests can be highly sensitive to the choice of rank, a further indication of the importance of choosing the rank properly.

The results for our chosen rank of three suggest that both LYR and RB are individually weakly exogenous. A logical step is to test whether they are jointly exogenous. Once again, choose *Test for Weak Exogeneity* from the *I(1)* menu. This time, input 2 as the number of exogenous variables. For the R matrix, enter a 1.0 in the DRB column on the first row, and in the DLYR column on the second row.

This time, the result is

```
TEST OF RESTRICTED MODEL:    CHISQR(6) = 15.048 [0.020]
```

so we reject the hypothesis that the variables are *jointly* exogenous.

## 11.2 Models with Weakly Exogenous Variables

Based on the results above, Juselius examines a model with RB treated as a weakly exogenous variable in Illustration 11.2.1.

Exit CATS, and restart using the following instructions. Note the addition of the EXO option and corresponding supplementary card:

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

To reproduce Table 11.2, select *Rank Test Statistics* from the *I(1)* menu.

Next, set the rank to three as usual, and normalize on DPY, LM3RC and RM in that order. To see the remainder of the information displayed at the bottom of page 199 in the text, do *Misc—Short Run Parameters*.

To produce Table 11.3, select *Residual Analysis* from the *Misc* menu.

## 11.3 Testing a Known Vector in $\alpha$

For Illustration 11.3.1, we return to the original model from 11.1.1. Exit CATS, and restart using the following instructions:

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

Set the rank to three as usual.

To do a test for a unit vector in  $\alpha$  for each of the variables, you just need to select *Unit Vector in Alpha* from the *Automated Tests* menu. The line for the rank 3 case is reproduced as Table 11.4 in the book.

## 12. Identification of the Long-run Structure

In this chapter, we examine techniques for identifying the structure of the long-run cointegrating relations. The example code for this section is provided on the file `CATSHandbook_Ch12.rpf`.

### 12.1 Just-identifying Restrictions, Table 12.1

For this section, we'll use the unrestricted constant model:

```
@cats (lags=2, noexo, shift, dum, detrend=drift) 1973:1 2003:1
# lm3rc lyr dpy Rm Rb
# Ds831
# Dt754 Dp764
```

Use  $I(1)$ —Set Rank of  $P_i$  and choose three cointegrating vectors.

#### 12.1.1 Example One

This example treats real money, inflation, and the short-term interest rate as endogenous variables, and the real income and bond rate as exogenous. We'll handle this using the *Restrictions on Each Beta vector* operation on the  $I(1)$  menu. For each vector, we will impose two restrictions—that the coefficients on the other two “endogenous” variables are zero.

Select *Restrictions on Each Beta vector* and enter two as the number of restrictions on the first vector. Input the first of the three “design” matrices shown on page 217 of *TCVM* as the  $\mathbf{H}'$  matrix. Note that CATS is asking for the *transpose* of  $\mathbf{H}$  as compared to the form shown in the book. So, the first dialog box should look like this:

	LM3RC	LYR	DPY	RM	RB	DS831
1	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Here, inflation (DPY) and the interest rate (RM) are being restricted to zero in this first vector, because there will be no  $\phi$  terms for those variables in the  $\mathbf{H}\phi$  product.

Repeat for the other two vectors for the other two  $\mathbf{H}$  matrices shown on page 217. When done, normalize on LM3RC in the first column, DPY in the second and RM in the third. This should give you the results in the left side of Table 12.1 on page 217.

#### 12.1.2 Example Two

The second example tests a more complex collection of restrictions. Once again, select *Restrictions on Each Beta vector*, and enter 2 as the number of restrictions for the first vector. Enter this design matrix (this is already transposed from the form shown in the text, so input it into the dialog box exactly as shown here):

```
1 -1 0 0 0 0
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1
```

This imposes an equality constraint on LM3RC and LYR, and a zero restriction on DPY.

Continue with two restrictions on each of the other two vectors, using the following design matrices:

```
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 -1 0
0 0 0 0 0 1
```

and

```
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 1 0 0
0 0 0 0 0 1
```

When done, normalize on LM3RC, DPY, and RM again. This produces the right side of Table 12.1.

## 12.2 Over-identifying Restrictions, Table 12.3

### 12.2.1 Example Three

Here, the first two relations are the first two (restricted) beta vectors from Example One with the additional restriction of zeroing out variables with insignificant coefficients, while the third relation comes from the homogenous inflation–interest rates relation from Table 10.3 in the text. To implement this, select *Restrictions on Each Beta vector* and enter 4 as the number of restrictions on the first vector. Enter the  $\mathbf{H}'$  matrix as:

```
1 -1 0 0 0 0
0 0 0 0 0 1
```

Enter 3 restrictions for the second vector, with  $\mathbf{H}'$  matrix

```
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 0 0 1
```

Finally, enter 3 restrictions for the third vector, with  $\mathbf{H}'$  matrix

```
0 0 1 -1 0 0
0 0 0 -1 1 0
0 0 0 0 0 1
```

When done, normalize on LM3RC, DPY, and RM again. This produces the left side of Table 12.3.

### 12.2.2 Example Four

This combines the  $H_{15}$ ,  $H_{17}$ , and  $H_{27}$  restrictions from Table 10.3 in the text. Use *Restrictions on Each Beta vector*, and enter 3 restrictions on first vector, with design matrix:

```
1 -1 0 0 0 0
0 0 0 -1 1 0
0 0 0 0 0 1
```

Enter 3 restrictions on second vector, with design matrix:

```
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 0 0 1
```

Enter 3 restrictions on third vector, with design matrix:

```
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1
```

When done, normalize on LM3RC, DPY, and RM again. This produces the right side of Table 12.3.

## 12.3 Lack of Identification, Table 12.4

### 12.3.1 Example Five

Once again, we'll use the *Restrictions on Each Beta vector* operation.

Enter 2 restrictions for the first vector, with  $\mathbf{H}'$  matrix:

```
1  0  0  0  0  0
0  1  0  0  0  0
0  0  0  1 -1  0
0  0  0  0  0  1
```

Enter 2 restrictions for the second vector:

```
0  0  1  0  0  0
0  0  0  1  0  0
0  0  0  0  1  0
0  0  0  0  0  1
```

Enter 3 restrictions for the third vector:

```
0  0  0  1  0  0
0  0  0  0  1  0
0  0  0  0  0  1
```

Let CATS estimate with non-identifying restrictions. Choose to override the recommended choice for degrees of freedom (by telling CATS that the suggested value is *not* correct), and enter 2 instead as indicated on page 223 of the text.

Normalize on LM3RC, DPY, and RM. This should produce the first (left) half of Table 12.4 from the text.

### 12.3.2 Example Six

Again use *Restrictions on Each Beta vector*. Enter 4 restrictions for the first vector, with  $\mathbf{H}'$  matrix:

```
1 -1  0  0  0  0
0  0  0  0  0  1
```

Enter 3 restrictions for the second vector:

```
0  0  0  1  0  0
0  0  0  0  1  0
0  0  0  0  0  1
```

Enter 2 restrictions for the third vector:

```
0  1  0  0  0  0
0  0  1  0  0  0
0  0  0  1  0  0
0  0  0  0  1  0
```

Normalize on LM3RC, RM, and DPY (note the reversal of the last two variables).

Note: the text suggests CATS will ask for confirmation on the recommended degrees of freedom (which is 3). That does not seem to be the case with the current version of CATS (it does not ask you to confirm the degrees of freedom), probably due to improvements made to the software after the textbook was written but before CATS 2.0 was released.

This model produces the results from the right side of Table 12.4.

## 12.4 Recursive tests of $\alpha$ and $\beta$ , Figures 12.1–12.7

Next, we apply the recursive estimation techniques to the restricted models. First, re-impose the restrictions from Example 3 (page 48), using the *Restrictions on Each Beta vector* operation. Recall that these are imposed as follows:

Four restrictions for the first vector, with  $\mathbf{H}'$  matrix:

```
1 -1  0  0  0  0
0  0  0  0  0  1
```

Three restrictions for the second:

```
0  1  0  0  0  0
0  0  1  0  0  0
0  0  0  0  0  1
```

Four restrictions for the third:

```
0  0  1 -1  0  0
0  0  0 -1  1  0
0  0  0  0  0  1
```

When done, normalize on LM3RC, DPY, and RM.

Next, select *Recursive Estimation* from the  $I(1)$  menu. Set the end of base sample to 1985:3, the beginning of the known sample to 1985:4, and the end of known sample to 2003:1.

After completing the estimation step, CATS will display the “Recursive Graphics” dialog box. You can turn on the checkboxes for the specific types of graphs you want to see, or just turn on the “Select All” box at the bottom if you want to see all the graphs. When you click on “OK”, CATS will display your selected graphs.

## 13. Identification of the Short-Run Structure

Chapter 13 examines the process of identifying the short-run structure of the model. The code for this chapter is provided on the file `CATSHandbook_Ch13.rpf`.

### 13.1 The Data

The initial setup instructions for reading in the data are unchanged. We will reproduce them here for reference—recall that you can just do “`source CATSHandbook_DataRead.src`” to execute all of these commands:

```
calendar(q) 1973
allocate 2003:1
open data book.xls
data(format=xls,org=obs) / lyr lpy lm3n Rm Rb lm3rc dpy
set lm3r = lm3n-lpy
* Nominal corrected M3 (in logs)
set lm3nc = lm3rc+lpy;

* Dummy variables
set Dt754 = 0.0
compute Dt754(1975:4) = 1
compute Dt754(1976:1) = -0.5
compute Dt754(1976:2) = -0.5

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

set Ds831 = T>=83:1
```

### 13.2 Creating Error Correction Terms and First Differences

We will be working with two different sets of pre-determined error correction terms derived from estimating the full model subject to the  $H_{S,3}$  and  $H_{S,4}$  hypothesis from Chapter 12. We need to create those error correction series using **SET** instructions.

First, for the  $H_{S,3}$  hypothesis:

```
set ecm1HS3 = lm3rc - lyr - 0.3355*ds831
set ecm2HS3 = dpy + 0.0274*lyr + 0.013*ds831
set ecm3HS3 = Rm - 0.2014*dpy - 0.7986*Rb - 0.0127*ds831
```

And for the  $H_{S,4}$  hypothesis:

```
set ecm1HS4 = lm3rc - lyr - 13.2670*(Rm-Rb) - 0.1533*Ds831
set ecm2HS4 = dpy + 0.0297*lyr + 0.0124*Ds831
set ecm3HS4 = Rm - 0.8087*Rb - 0.0090*Ds831
```

We also need to create differences of the dependent variables so that we can specify our model in error correction form:

```
diff lm3rc / dlm3rc
diff lyr / dlyr
diff dpy / ddp
diff rm / drm
diff rb / drb
diff ds831 / dp831
```

### 13.3 Estimating the VECM Model, and F-Tests

Now we can set up and estimate our VECM models. First, for the  $H_{S,3}$  hypothesis.

```
system(model=vecm1mod)
variables dlm3rc ddpv drb dlyr drb
lags 1
deter ecm1hs3{1} ecm2hs3{1} ecm3hs3{1} dt754 dp764 dp831 constant
end(system)
```

```
* Estimate the model:
estimate
```

To compute the F-statistics testing each variable for significance across all equations shown on page 237 (for the  $H_{S,3}$  model), we repeat the basic process at the beginning of Chapter 4 (page 15).

```
compute eqn=%modeleqn(vecm1mod,1)
compute betaols=%modelgetcoeffs(vecm1mod)

report(action=define)
report(atrow=1,atcol=1) "Label" "F-Stat"
do i=1,%eqnsize(eqn)
  compute fstat=%qform(inv(%sigma),$
    %xrow(betaols,i))*(%nobs-%nreg)/(5*%nobs*%xx(i,i))
  compute fpval=%ftest(fstat,%nvar,%nobs-%nreg)
  report(row=new,atcol=1) %eqnreglabels(eqn)(i) fstat fpval
  if fpval<0.05
    report(action=format,atrow=%reportrow,atcol=2,special=onestar)
end do i
report(action=format,atcol=2,picture="*.##",align=decimal)
report(action=show)
```

### 13.4 Restrictions on the Short-Run Model, Tables 13.1 and 13.2

Tables 13.1 and 13.2 present the results of estimated restricted models for each of the two hypothesis, where variables deemed insignificant have been omitted from the model.

For the  $H_{S,3}$  model, we define the following restricted equations (see the example program file for the full model definitions, from which these restricted equations are derived).

```
equation vecm1hs3 dlm3rc
# drb{1} ecm1hs3{1} ecm3hs3{1} constant
equation vecm2hs3 ddpv
# drb{1} ecm2hs3{1} constant
equation vecm3hs3 drb
# drb{1} ecm2hs3{1} dp764 constant
equation vecm4hs3 dlyr
# dlyr{1} drb{1} ecm1hs3{1} ecm2hs3{1} ecm3hs3{1} dt754 constant
equation vecm5hs3 drb
# dlyr{1} drb{1} ecm2hs3{1} ecm3hs3{1} dp831 constant
```

```
* Group into a model:
group vecmhs3mod_rest vecm1hs3 vecm2hs3 vecm3hs3 vecm4hs3 vecm5hs3
```

To produce the first part of Table 13.1, we estimate the model (note: the table in the book omits the constant terms):

```
estimate(model=vecmhs3mod_rest,resids=residvechs3)
```

For the second part of the table (the  $\Omega$  matrix), we use **CMOM(CORR)** to compute a residual correlation matrix. We then use **EWISE** to replace the diagonal elements of that correlation matrix with the standard errors (computed as the square root of the corresponding element of %SIGMA, the variance/covariance matrix set by **ESTIMATE**):

```
* The Omega matrix:

* Compute correlation matrix
cmom(corr,matrix=cmommat)
# residvechs3

* Replace diagonals with standard errors
ewise cmommat(i,j) = %if(i==j,sqrt(%sigma(i,j)),cmommat(i,j))

* Display the result
display #####.### cmommat
```

For Table 13.2 presenting the results for the  $H_{S,4}$  model, the steps are the same, but with a different set of restricted equations:

```
equation vecmlhs4 dlm3rc
# drb{1} ecm1hs4{1} constant
equation vecm2hs4 ddpv
# drb{1} ecm2hs4{1} constant
equation vecm3hs4 drm
# drb{1} ecm2hs4{1} ecm3hs4{1} dp764 constant
equation vecm4hs4 dlyr
# dlyr{1} drb{1} ecm1hs4{1} dt754 constant
equation vecm5hs4 drb
# dlyr{1} drb{1} ecm3hs4{1} dp831 constant

group vecmhs4mod_rest vecmlhs4 vecm2hs4 vecm3hs4 vecm4hs4 vecm5hs4

estimate(model=vecmhs4mod_rest,resids=residvechs4)

* The Omega matrix:
cmom(corr,matrix=cmommat)
# residvechs4
ewise cmommat(i,j) = %if(i==j,sqrt(%sigma(i,j)),cmommat(i,j))
display #####.### cmommat
```

### 13.5 The VAR in Triangular Form, Tables 13.3 and 13.4

These tables can be computed by doing OLS estimations as shown below. Note the inclusion of contemporaneous first differences of all of the endogenous variables other than the dependent variable:

For Table 13.3:

```
linreg dlm3rc
# ddpv drm dlyr drb dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
  dt754 dp764 dp831 constant

linreg ddpv
# dlm3rc drm dlyr drb dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
  dt754 dp764 dp831 constant

linreg drm
# dlm3rc ddpv dlyr drb dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
  dt754 dp764 dp831 constant
```

```
linreg dlyr
# dlm3rc ddpym drm drb dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
dt754 dp764 dp831 constant
```

```
linreg drb
# dlm3rc ddpym drm dlyr dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
dt754 dp764 dp831 constant
```

For Table 13.4, the regressions are as follows. Note that these are done from the last in the causal ordering towards the first.

```
linreg dlm3rc
# ddpym drm dlyr drb dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
dt754 dp764 dp831 constant
```

```
linreg ddpym
# drm dlyr drb dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
dt754 dp764 dp831 constant
```

```
linreg drm
# dlyr drb dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
dt754 dp764 dp831 constant
```

```
linreg dlyr
# drb dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
dt754 dp764 dp831 constant
```

```
linreg drb
# dlyr{1} drb{1} ecm1hs4{1} ecm2hs4{1} ecm3hs4{1} $
dt754 dp764 dp831 constant
```

### 13.6 Tables 13.5-13.9

We have not been able to reproduce the maximum likelihood estimation results reported in these tables, and thus will not go through the steps required here in the handbook.

We have included our maximum likelihood estimation code for these in the example program for this chapter, so please refer to that file if you want to see how we would recommend estimating these models in RATS.

## 14. Identification of Common Trends

In this chapter, we explore imposing identifying restrictions on the common trends. Sample code for this section is provided on the file `CATSHandbook_Ch14.rpf`. If necessary, load the data and compile CATS as shown on page 27.

We will again use the unrestricted constant model specified by the following instructions:

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

### 14.6.1 The Unrestricted MA Representation

Impose the rank of three and accept the default normalization.

To generate Table 14.1, do the following:

- 1) select *MA Representation* from the *Misc* menu
- 2) normalize on RB and LYR, respectively (these will *not* be the default choices, so be sure to normalize on these two variables to reproduce the results from the text).

This will display the restricted estimates of the (orthogonal)  $\alpha$  and  $\beta$  matrices and the  $\mathbf{C}$  matrix.

If you want to graph the cumulated residuals, you need to exit out of CATS. The estimated residuals series are automatically saved in variables named `RES_XXX` where `XXX` is the name of the original variable. You can use **ACCUMULATE** to accumulate the series, and then graph them. For example:

```
accumulate res_lm3rc / accres_lm3rc
accumulate res_lyr / accres_lyr
accumulate res_dpy / accres_dpy
accumulate res_rm / accres_rm
accumulate res_rb / accres_rb
```

You could also use a **DOFOR** loop and the `%S` and `%L` functions to create series by prefixing “ACC” to the beginning of each series name:

```
dofor ser = res_lm3rc res_lyr res_dpy res_rm res_rb
  accumulate ser / %s("acc"+%l(ser))
end do for
```

Then do:

```
spgraph(footer="Cumulated residuals from each equation", $
        vfields=3, hfields=2)
graph(header="Cumulated residuals from money equation")
# accres_lm3rc
graph(header="Cumulated residuals from inflation rate equation")
# accres_dpy
graph(header="Cumulated residuals from bond rate equation")
# accres_rb
graph(header="Cumulated residuals from income equation")
# accres_lyr
graph(header="Cumulated residuals from short rate equation")
# accres_rm
spgraph(done)
```

To display the common trend graphs of Figure 14.2, you would need to either do the calculation shown on page 262 by manually typing in the values for the unrestricted  $\alpha'_1$  displayed in the output, or by modifying the CATS procedures to save the values of  $\alpha'_1$  in a globally accessible variable.

If you wish to do the latter, the values that would need to be saved are computed in the line

```
COMP AlphaOrt = %f_Ortho%(%Alpha1)
```

in the `CATSToolPak2.src` file. Copying these values to a global variable would allow you to do the computation after exiting CATS. For example:

```
COMP AlphaOrt = %f_Ortho%(%Alpha1)
compute AlphaOrtGlobal = AlphaOrt
```

#### 14.6.2 The MA Representation Subject to Restrictions on $\alpha$ and $\beta$

If you exited out of CATS to display the graphs, restart using the model

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

Again, select three as the rank and accept the default normalization.

For Table 14.2, which treats RB as weakly exogenous, select *I(1)—Test for Weak Exogeneity*. Enter one as the number of weakly exogenous variables, and then enter 1.0 in the DRB column of the **R** matrix (leaving the other columns set to zero) and click on OK. When prompted, select no restrictions on Beta.

Once this is done, select *MA Representation* from *Misc* menu, and normalize on LYR and RB respectively, as above. This should reproduce Table 14.2.

For Table 14.3, we need to add the  $H_4$  restrictions from Chapter 12. Select *Restrictions on Each Beta* vector from the *I(1)* menu. When prompted, choose to “Keep restrictions on Alpha”. This preserves the weak exogeneity restriction imposed on RB.

Next, enter three as the number restrictions on the first vector, and enter the **H'** matrix as:

```
1 -1 0 0 0 0
0 0 0 -1 1 0
0 0 0 0 0 1
```

Enter three restrictions on second vector:

```
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 0 0 1
```

Enter three restrictions on third vector:

```
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1
```

Normalize on LM3RC, DPY, and RM.

To reproduce Table 14.3, select *MA Representation* from the *Misc* menu, and again normalize on LYR and RB, respectively. The moving average representation will now reflect both the weak exogeneity restriction and the restrictions on Beta.

For Table 14.4, we want to maintain the same restrictions on Beta, but impose two weak exogeneity restrictions instead of one. Fortunately, CATS makes this easy to implement by allowing us to retain the restrictions on Beta while changing the restrictions on Alpha.

Start by choosing *I(1)—Test for Weak Exogeneity*. CATS will indicate that there are existing restrictions on Alpha, and ask what you want to do. We want to replace the existing restriction with the two-variable restriction, so choose the “Set New Restrictions” option and click on OK.

Next, enter two as the number of weakly exogenous variables. Set the first row in the DRB column and the second row in the DLYR column to 1.0, leaving the other cells as zeros. Normalize on LM3RC, DPY and RM again.

Then, when prompted, choose “Current Restrictions on Beta” to retain the restrictions imposed above.

Finally, select *MA Representation* from the *Misc* menu, and again normalize on LYR and RB, respectively. The moving average representation will now reflect the two weak exogeneity restrictions in addition to the restrictions on Beta.

#### **14.6.3 Remaining Sections of Chapter 14**

The calculations in the remaining sections of Chapter 14 in the textbook were presumably done outside of CATS, using some standard RATS instructions. As our focus is on CATS, we won't try to reproduce those here.

## 15. Identification of a Structural MA Model

Chapter 15 looks at the use of structural restrictions, distinguishing between transitory and permanent shocks, and the empirical interpretation of results. The example code for this section is provided on the file `CATSHandbook_Ch15.rpf`.

### 15.1 Structural Models and Variable Order

As described in Chapter 15 of the text and in Section 3.11 of the CATS manual, the methodology for imposing structural restrictions depends in large part on the order in which the variables are listed in the model (we will examine this issue further below).

As a result, before you can test a set of identifying restrictions, you will need to make sure your variables are listed in an order that will allow the desired specification. Testing multiple structural hypothesis may in fact require starting CATS with different orderings for each hypothesis.

To implement the structural model described in the illustration in Section 15.4 of the textbook (see page 280), we start CATS using the variable order shown below:

```
@cats (lags=2, noexo, shift, dum, dettrend=drift) 1973:1 2003:1
# lm3rc dpy Rm lyr Rb      (rather than "# lm3rc lyr dpy Rm Rb")
# Ds831
# Dt754 Dp764
```

Set the rank to three as usual, and accept the default normalization.

### 15.2 Structural MA Model Illustration

Given  $p$  variables (five in our example), and rank  $r$  (three), there will be  $p-r$  (two) permanent, or long-term, shocks and  $r$  (three) transitory, or short-term, shocks. As described in the text and on page 89 of the CATS manual, you can impose identifying restrictions on these by specifying zero restrictions on elements of the long run impact matrix  $\tilde{C}$  and the contemporaneous impact matrix  $\tilde{C}_0^*$ .

Since the first  $r$  shocks are transitory, the first  $r$  columns of  $\tilde{C}$  are all zeros. The permanent shocks are associated with last  $p-r$  columns of  $\tilde{C}$ , and the restrictions on these are imposed by setting certain elements to zero, leaving the other elements unrestricted.

As noted on page 282 of the text, the long-run restrictions we will impose are derived from the assumption that a nominal shock cannot have a long-run impact on real income. This implies that the first shock  $u_{11}$  will be the real shock, which is allowed to affect all variables in the long run. The second shock,  $u_{12}$ , will be the nominal shock, which impacts all variables *except* real income (LYR).

We accomplish this by imposing a zero restriction on the fourth row of column five of  $\tilde{C}$ , so that LYR is only affected by the  $u_{11}$  (real) shock (the nominal shock is identified by the condition that it is orthogonal to the real shock). Here is a paraphrased version of the relevant portion of the representation from page 282 of *TCVM*, showing the zero restriction on the nominal shock to LYR:

$$\begin{bmatrix} lm3rc_t \\ dpy_t \\ rm_t \\ lyr_t \\ rb_t \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & * & * \end{bmatrix} \begin{bmatrix} \Sigma u_{s1,i} \\ \Sigma u_{s2,i} \\ \Sigma u_{s3,i} \\ \Sigma u_{11,i} \\ \Sigma u_{11,i} \end{bmatrix}$$

The three short-run exclusion restrictions described on page 282 are handled by putting two zeros on the third (RM) row of the contemporaneous impact matrix, one zero on the second (DPY) row, and no zeros on the first, resulting in the following:

$$\begin{bmatrix} * & * & * & * & * \\ * & * & 0 & * & * \\ * & 0 & 0 & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \begin{bmatrix} u_{s1,t} \\ u_{s2,t} \\ u_{s3,t} \\ u_{l1,t} \\ u_{l1,t} \end{bmatrix}$$

As you can see, the three short-term identifications from page 282 follow from multiplying out the terms above.

As noted on page 89 of the CATS manual, CATS accepts the long-run restrictions by allowing you set to zero the last  $j$  elements of row  $i_j$  of the  $\tilde{C}$  matrix, for  $j = 1, \dots, p - r - 1$  and  $1 \leq i_j \leq p$ .

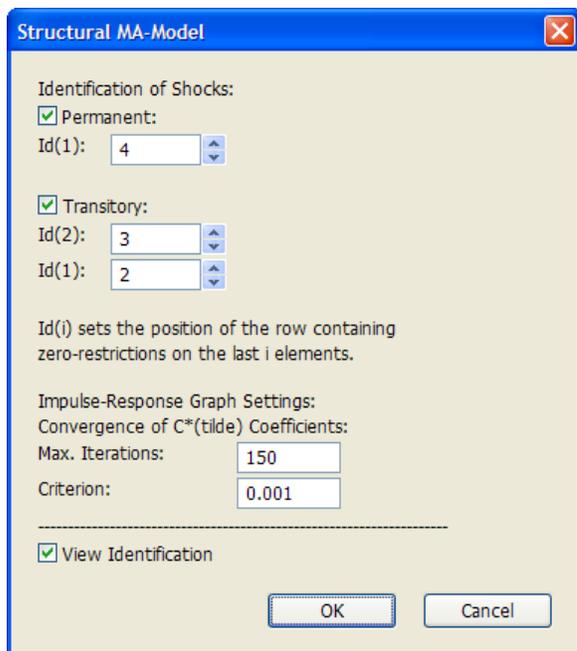
Similarly, the short-run restrictions are input by setting to zero the last  $j$  elements of row  $i_j$  in  $\tilde{C}_0^*$ , for  $j = 1, \dots, r - 1$  for  $1 \leq i_j \leq p$ .

Note that this is why you have to choose the variable ordering carefully—so that the pattern of zeros allowed by CATS will give the desired restrictions.

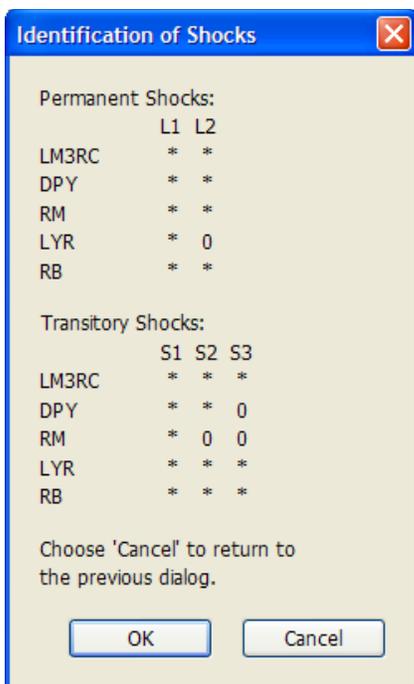
IMPORTANT NOTE: Unfortunately, the dialog box (see below) describes the identifications as  $\text{Id}(i)$  with  $i$  being the number of columns to be set, rather than using  $\text{Id}(j)$ , which would match the description used on page 89.

To impose these restrictions, select *Structural MA Model* from the *Misc* menu. In the sample dialog below, we have entered the appropriate row values for the restrictions, as described above. For the permanent shock  $\text{Id}(1)$  field, we set the row to four, thus setting one column (the last column) in the fourth row to zero.

For the short term shocks, we set  $\text{Id}(1)$  to two and  $\text{Id}(2)$  to three. This gives one zero in row two, and two zeros in row three. The dialog should look like this:



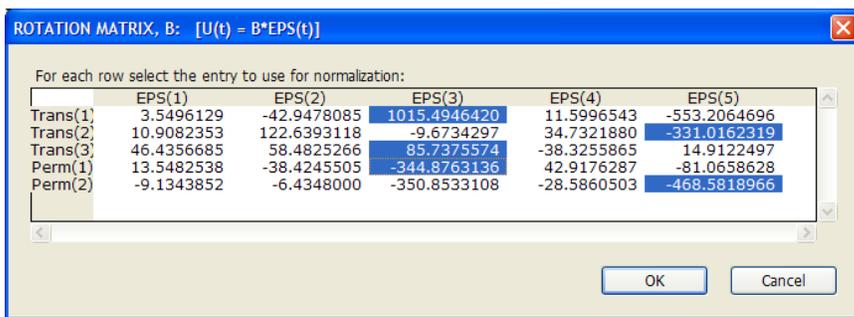
If the “View Identification” switch is turned on, clicking on “OK” will display the following dialog box, allowing you to verify that the restrictions have been entered correctly:



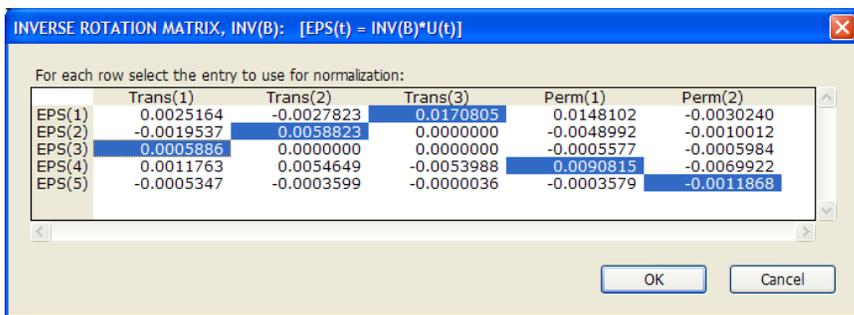
If the restrictions are not what you intended, you can click on “Cancel” to return to the previous dialog. If they are correct, click on “OK” to continue.

The next dialog asks you to select the variable on which each of the permanent shocks should be normalized. Normalize on LYR and RB, respectively.

Next, you are asked to choose the normalization of the rotation matrix as a function of the estimated VAR residuals. To match the textbook results, normalize as shown below:



Finally, you will be prompted to select the normalizations for the inverse rotation matrix, as a function of the structural shocks. Again, use the normalization settings shown below:



These settings should reproduce the impulse response plot (Figure 15.1) from page 285, and the various numerical results shown in the textbook, including (15.17), (15.18), (15.19), and (15.20).

### 15.3 Are the Labels Credible?

The illustration above involves, as the text suggests, putting “economic labels” on the empirical associations (p. 286). In Section 15.5 of the text, Juselius examines the question of whether the identifying assumptions are appropriate by comparing these results against an alternative structural model and the standard moving average model. We’ll look at the additional steps required for the comparison.

#### 15.3.1 Generating Table 15.1—SVAR 1 Model

The “SVAR 1” model in Table 15.1 is essentially the model just analyzed above, with the exception of a slightly different choice of normalization for the Rotation matrix, **B**. To reproduce the Table 15.1 results, repeat the analysis above but normalize on column 4, rather than column 3, of the Rotation matrix.

With that change, the SVAR 1 values in the “estimated common trends” section can be found in the last two rows of the normalized **B** matrix output in CATS:

Rotation Matrix, B (Normalized)					
	EPS (1)	EPS (2)	EPS (3)	EPS (4)	EPS (5)
Trans (1)	0.003	-0.042	1.000	0.011	-0.545
Trans (2)	-0.033	-0.370	0.029	-0.105	1.000
Trans (3)	0.542	0.682	1.000	-0.447	0.174
Perm(1)	0.316	-0.895	-8.036	1.000	-1.889
Perm(2)	0.019	0.014	0.749	0.061	1.000

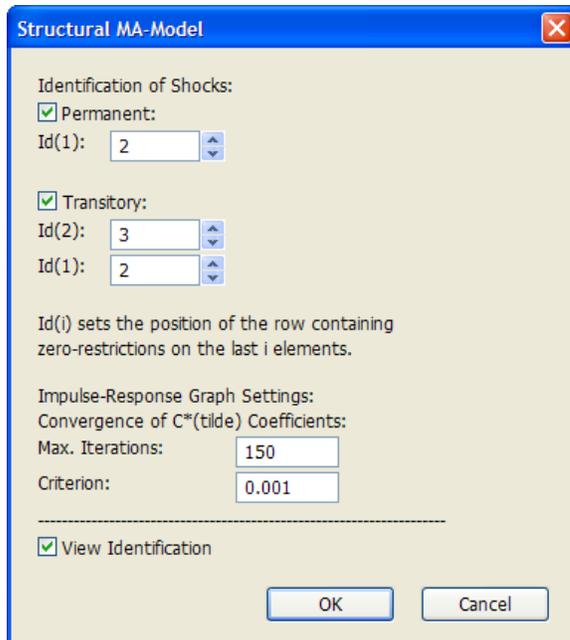
The “Perm(1)” row is the Real trend and the “Perm(2)” row is the Nominal trend (the Real trend values differ slightly from those in the text).

The values shown in the “loadings to” section of the table are from the last two columns of the  $\tilde{C}$  matrix output:

Structural Long-Run Impact Matrix, C(tilde) (Normalized)					
	Trans (1)	Trans (2)	Trans (3)	Perm(1)	Perm(2)
LM3RC	0.000	0.000	-0.000	1.041	-3.040
DPY	0.000	-0.000	-0.000	-0.033	-0.200
RM	-0.000	-0.000	0.000	-0.034	0.628
LYR	0.000	0.000	0.000	1.000	-0.000
RB	-0.000	-0.000	0.000	-0.034	1.000

#### 15.3.2 Generating Table 15.1—SVAR 2 Model

The process for generating the SVAR 2 model is identical to that described above, except that we enter two (rather than four) as the value for the Id(1) permanent shock:



This puts the zero on the second row, corresponding to the inflation variable, and is thus a restriction that the (real) shock has no long-run impact on inflation. Using the same normalizations as for SVAR 1 reproduces the SVAR 2 results in Table 15.1.

### 15.3.3 Generating Table 15.1—VAR Model

The VAR portions of Table 15.1 come from the just-identified MA model presented in Chapter 14. To reproduce the relevant results, just select *MA Representation* from the *Misc* menu. In the dialog asking you to specify the normalization for Alpha Orthogonal, normalize on RB and LYR respectively.

The “estimated common trends” can be found in the “ALPHA Orthogonal” output:

```
ALPHA Orthogonal (transposed)
      LM3RC    DPY      RM      LYR    RB
CT(1) 0.000    0.061    1.111  0.000  1.000
      (0.010)  (1.079)  (1.448) (.NA) (.NA)
CT(2) 0.316   -0.780   -5.937  1.000  0.000
      (1.630) (-1.488) (-0.839) (.NA) (.NA)
```

while the loadings can be found here:

```
The Loadings to the Common Trends, BETA_ORT(tilde):
      CT1    CT2
LM3RC -4.461    0.688
      (-2.064) (3.622)
DPY   -0.135   -0.039
      (-1.747) (-5.685)
RM     0.639    0.007
      (4.383)  (0.571)
LYR   -1.562    0.827
      (-0.786) (4.734)
RB     0.985    0.029
      (4.562)  (1.523)
```

## 16. Analyzing I(2) Data with the I(1) Model

Chapter 16 of the text looks at some of the differences between the I(2) and I(1) models, and at ways to check for I(2) behavior in an I(1) model. Sample code for this section is provided on the file `CATSHandbook_Ch16.rpf`.

### 16.1 Stochastic and Deterministic Trends

#### 16.1.1 Figure 16.1

As noted on page 294, one of the first steps in an I(2) analysis should be to examine graphs of the data in both levels and differences. Figure 16.1 in the text plots the levels and first differences of the logs of nominal money stock and prices. The first difference plots include moving averages of the data. These help make the long term behavior of the differences more visible by omitting much of the quarter-to-quarter noise.

After loading the data (see page 27), we already have the differences of prices available as the variable `DPY`, but we will need to compute the differences of money stock. You can do that using a **DIFFERENCE** instruction, a **SET** instruction, or either the *Transformations* or *Differences* menu operations. We'll use the **DIFFERENCE** instruction:

```
diff lm3nc / dlm3nc
```

The moving average filtering is accomplished using the **MVSTATS** instruction (which is also available via the *Moving Window Statistics* wizard), with options specifying a four-period centered moving window:

```
mvstats (width=4,means=madlm3nc,centered) dlm3nc
mvstats (width=4,means=madpy,centered) dpy
```

We can now generate the graphs. We'll use **SPGRAPH** instructions to arrange all four graphs on a single page:

```
spgraph (vfield=2,hfield=2,footer="Fig 16.1")
  graph (frame=half,key=above,nokbox)
  # lm3nc
  graph (frame=half,key=above,nokbox)
  # lpy
  graph (frame=half,key=above,nokbox) 2
  # dlm3nc
  # madlm3nc
  graph (frame=half,key=above,nokbox) 2
  # dpy
  # madpy
spgraph (done)
```

#### 16.1.2 Figure 16.2

To graph the co-movements shown in Figure 16.2, we need to compute the changes in value since the first observation for the logged prices and nominal money stock. We can do that using **SET** instructions:

```
set chlpy = lpy-lpy(1)
set chlnm = lm3nc-lm3nc(1)
```

Here, the "CH" (for "change") series will be set equal to difference between the variable at time  $t$  and its value at entry one. If you wanted to use the starting date rather than entry number one, you would write these as

```
set chlpy = lpy-lpy(1973:1)
set chlnm = lm3nc-lm3nc(1973:1)
```

The second graph in Figure 16.2 is a scatter plot graphing CHLPY against CHLNM. The solid line represents the fitted values from regressing CHLPY on a constant and CHLNM. Including this makes it easier to see where the LPY has increased at a faster or slower rate than CHLNM.

Here is how you can create the fitted-values series:

```
linreg chlpy
# constant chlnm
prj fitted
```

Finally, we use the following instructions to generate the graphs:

```
spgraph(vfield=2)
graph(key=upleft,nokbox,frame=half,patterns) 2
# chlpy
# chlnm

scatter(overlay=line,ovsame,pattern) 2
# chlpy chlnm / 9
# fitted chlnm / 1
spgraph(done)
```

## 16.2 I(2) Symptoms in I(1) Models

### 16.2.1 Dummy Variables

Before launching CATS again, we need to create a couple of additional dummy variables that will be used later in this and subsequent chapters. The first is a “permanent blip”, which will have zeros for all observations except for the first quarter of 1983, where it will be equal to 1. You can create this by doing:

```
set Dp831 = 0.0
compute Dp831(1983:1) = 1
```

or:

```
set Dp831 = t==1983:1
```

We also need a broken trend dummy starting in 1983. We can do that by multiplying the existing shift dummy DS831 by  $T$  (the current entry number):

```
set tDs831 / = t*Ds831
```

### 16.2.2 Characteristic Roots

Juselius considers two models here—one with a broken linear trend and one with a non-broken linear trend. To generate the characteristic roots for the first model, shown on page 298 of the text, we start CATS using the commands shown below. Note the following differences from our previous models:

- We are using LM3NC (nominal money stock) and LPY (prices in levels), rather than LM3RC (real money stock) and DLPY (prices in first differences), and
- We use the SHIFT option and list the broken trend dummy variable using the second supplementary card. The other dummy variables now appear on the third supplementary card.

```
@cats(lags=2,noexo,shift,dum,dettrend=cidrift) 1973:1 2003:1
# lm3nc lpy lyr Rm Rb
# tDs831
# Ds831 Dt754 Dp764
```

Next, choose *Roots of Companion Matrix* from the *Graphics* menu, and select ranks 2 through 5. CATS will display graphs and generate the tables of roots, which (generally) match the results

shown on page 298. There are some numerical discrepancies between these results and those shown in the book.

For the second model, the setup is as follows, which omits the broken trend but adds the permanent blip dummy:

```
@cats (lags=2, noexo, dum, dettrend=cidrift) 1973:1 2003:1
# lm3nc lpy lyr Rm Rb
# Ds831 Dp831 Dt754 Dp764
```

Again, select *Roots of Companion Matrix* from the *Graphics* menu and select ranks 2 through 5 to display the roots shown on page 299.

### 16.2.3 Graphing Cointegration Relations

To display graphs of the cointegration relations for this second model, select *Cointegrating Relations* from the *Graphics* menu. In the dialog box that appears, select (highlight) all five cointegrating relations, and click OK. This should display Figures 16.3 through 16.7 in the textbook.

Note: some of the graphs are sign-flipped or normalized differently from those shown in the book.

## 16.3 Evaluating the Nominal-to-Real Transformation

### 16.3.1 Testing Long-Run Price Homogeneity

For this section, we will demonstrate the use of the BREAK option (rather than the SHIFT option) to include the broken trend in the model. If CATS is running, exit out of the procedure and restart using these commands:

```
@cats (lags=2, noexo, break=trend, dum, dettrend=cidrift) 1973:1 2003:1
# lm3nc lpy lyr Rm Rb
# 1983:1
# Ds831 Dp831 Dt754 Dp764
```

Note the use of the BREAK=TREND option, with the date of the break (1983:1) specified on the second supplementary card.

Set the rank to three, and normalize on LM3NC, LPY, and LYR, respectively.

This reproduces the unrestricted estimates of  $\beta$  and  $\alpha$  shown in the first half of Table 16.1, and of  $\Pi$  as shown in Table 16.2.

As described on page 303, the first hypothesis we will test is that the broken trend can be excluded. We could use the *Restrictions on Each Beta Vector* operation for this, but as we are testing whether parameters are equal to zero, it is easier to use the *Zero-restrictions on Beta* operation (also on the *I(1)* menu).

In the book notation, the  $\beta_{11}$  coefficients are the terms on the full trend (that is, the trend over the full range), while the  $\beta_{12}$  coefficients are the terms for the broken trend (from 1983 on).

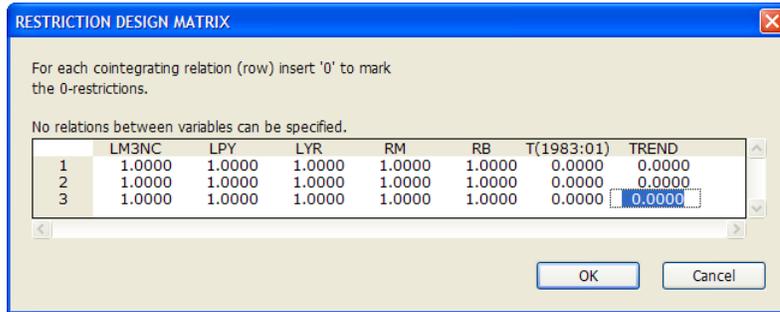
To test that all of the  $\beta_{12}$  terms are zero (omitting the broken trend), select *Zero-restrictions on Beta* and put zeros in all three rows for the “T(1983:1)” column. When prompted by the next dialog box, allow CATS to impose the identifying restrictions. The first line of the new output shows the test statistic and  $p$ -value for the hypothesis test:

```
TEST OF RESTRICTED MODEL:    CHISQR(3) = 34.472 [0.000]
```

strongly rejecting the hypothesis that the broken trend can be omitted.

To test  $\beta_{11}=0$  (no full trend, only broken trend), do the same but put zeros in all three rows of the “Trend” column.

For the joint test that both are zero, use the same operation, and put zeros in both columns. For example:



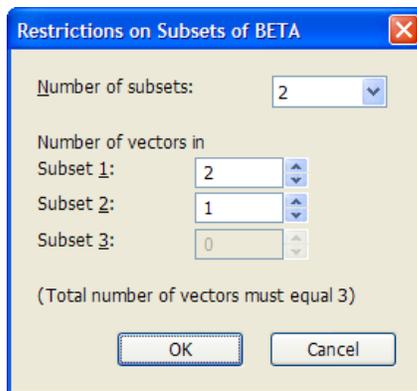
Next, Juselius tests the hypothesis  $\beta_{11} - \beta_{12} = 0$ , implying that a linear trend is only present in the first regime. We implement this using the *Restrictions on Each Beta* vector operation. With the using the  $\beta = \mathbf{H}\Phi$  formulation, impose one restriction on each vector, with a transposed  $\mathbf{H}$  matrix of the form:

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

for all three vectors.

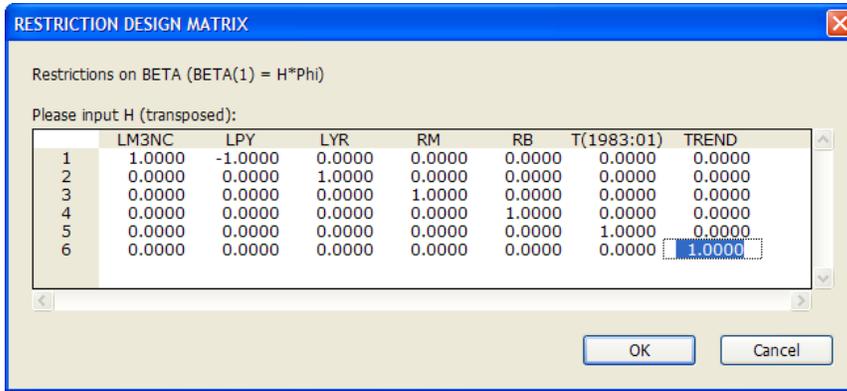
The right-hand side of Table 16.1 shows the estimates of a model where two of the vectors satisfy the long-run price homogeneity hypothesis, with the third left unrestricted.

To implement this (using the  $\beta = \mathbf{H}\Phi$  formulation), select *Restrictions on Subsets of Beta*. When prompted, set up two subsets, with two vectors in first subset, and one in the second:



So, the first subset will contain the first two vectors—the ones that we will be restricting. The second subset will contain the third vector, which we'll leave unrestricted.

The next dialog will ask for the number of restrictions for the first subset. Enter one, and then enter the following  $\mathbf{H}$  (transposed) matrix:



Enter zero as the number of restrictions on second subset, accept the suggested degrees of freedom (two), and normalize on LM3NC, LM3NC, and LYR, respectively.

This will generate the test statistic and the output for the restricted model.

The process for testing the long-run price homogeneity restriction on all three vectors is nearly identical. Just specify a single subset, with one restriction, and use the same  $H'$  matrix shown above. The estimates of this restricted model are shown in the left side of Table 16.3.

The right side of Table 16.3 presents estimates for an over-identified model imposing the interest rate and money demand relations of  $H_4$  from Chapter 12 of the text, plus the exclusion of the broken and regular trends, on the first two vectors, with the interest series excluded from the third vector.

This is most easily done using the  $R' \cdot \text{Beta} = 0$  formulation, which you can select using *Change Restriction Formulation* on the *I(1)* menu (or via the *Preferences*). Choose *I(1)—Restrictions on Each Beta vector*, and enter the restrictions as follows:

For the first vector, enter 5 as the number of restrictions and enter the  $R'$  matrix:

```
1  1  0  0  0  0  0
1  0  1  0  0  0  0
0  0  0  1  1  0  0
0  0  0  0  0  1  0
0  0  0  0  0  0  1
```

Enter 5 restrictions on the second vector, with  $R'$  matrix:

```
1  0  0  0  0  0  0
0  1  0  0  0  0  0
0  0  1  0  0  0  0
0  0  0  0  0  1  0
0  0  0  0  0  0  1
```

Enter 2 restrictions on third vector, with design matrix:

```
0  0  0  1  0  0  0
0  0  0  0  1  0  0
```

When prompted, normalize on LYR, RM, and LPY respectively.

Table 16.4 presents the same sets of restrictions as Tables 16.1 and 16.3, but using a model where the broken linear trend is restricted to zero in the cointegration relations, but still present in the data.

To implement this, quit out of CATS and restart using the following settings:

```
@cats(lags=2,noexo,dum,dettrend=cidrift) 1973:1 2003:1
# lm3nc lpy lyr Rm Rb
# Ds831 Dp831
```

Set the rank to three, and normalize on LM3NC, LM3NC, and LYR respectively. This produces the unrestricted model results in the upper left section of Table 16.4.

To implement the price homogeneity hypothesis on the first two vectors using the  $R^*Beta=0$  formulation, choose *Restrictions on subsets of Beta*, and specify two as the number of subsets, with two vectors in the first subset, and one in the second.

Enter one restriction on first subset, with an  $R'$  matrix of

```
1 1 0 0 0 0
```

Enter zero restrictions on the second subset, accept the default degrees of freedom, and normalize on LM3NC, LYR, LM3NC respectively. The results are shown in the upper right of Table 16.4.

To impose price homogeneity on all three vectors, choose *Restrictions on subsets of Beta*, but specify only one subset, containing three vectors.

Enter one restriction on the subset:

```
1 1 0 0 0 0
```

Normalize on LM3NC, LYR, and LYR respectively.

Finally, use the money demand and interest rate structure from H4 and the just-identified relation between money, prices, real income, and trend as follows.

Still using the using the  $R^*Beta=0$  formulation, choose  $I(1)$ —*Restrictions on Each Beta vector* and enter the following restrictions:

Four restrictions on the first vector, with  $R'$ :

```
1 1 0 0 0 0
1 0 1 0 0 0
0 0 0 1 1 0
0 0 0 0 0 1
```

Four restrictions on the second vector, with  $R'$ :

```
1 0 0 0 0 0
0 1 0 0 0 0
0 0 1 0 0 0
0 0 0 0 0 1
```

Two restrictions on the third vector, with  $R'$ :

```
0 0 0 1 0 0
0 0 0 0 1 0
```

When done, normalize on LM3NC, RM, and LM3NC.

This reproduces the results in the lower-right section of Table 16.4.

## 17. Estimating the I(2) Model

Chapter 17 of the text examines the I(2) model in more detail. Example code for this chapter is provided in the file `CATSHandbook_Ch17.rpf`. Note: some of the results and graphs produced using the instructions below do not match the results from the text exactly.

### 17.1 Estimating the I(2) Model and Determining Rank

The numerical examples begin in Section 17.4 of the text. Begin by restarting CATS with the following setup which includes the broken trend as a shift dummy and the permanent blip dummy `Dp831` and transitory blip dummy `Dt764` as exogenous dummy variables.

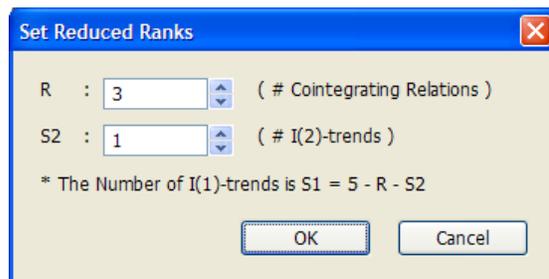
```
@cats (lags=2, noexo, shift, dum, dettrend=cidrift) 1973:1 2003:1
# lm3nc lpy lyr Rm Rb
# tDs831
# Dp831 Dt764
```

To generate Tables 17.1 and (the top portion of) 17.2, simply select *Rank Test Statistics* from the *I(2)* menu (not the *I(1)* menu!). The critical values for the broken trend, reported in the lower portion of 17.2, were simulated using separate code, and are not generated by CATS itself.

### 17.2 Setting the Reduced Ranks

Based on the results in the tables, Juselius suggests a model with the reduced rank ( $r$ ) of three, one first order stochastic trend ( $s_1$ ) and one second order stochastic trend ( $s_2$ ).

To impose these settings, select *Set Reduced Ranks* from the *I(2)* menu. In the reduced ranks dialog box, set  $r$  equal to three, and  $s_2$  equal to one, as shown below:



As noted in the dialog, the  $s_1$  value (one) is derived automatically as  $p - r - s_2$ .

Click on “OK” to accept these settings.

### 17.3 Graphing Cointegrating Relations

To reproduce the graphs shown on pages 324-325 of the text, select *Multi-Cointegration Relations* from the *Graphics* menu, select all three cointegrating relations, and click “OK”.

### 17.4 Identifying Restrictions

To impose the identifying restrictions described on page 327, select *Restrictions on Beta* from the *I(2)* menu.

Enter two restrictions on the first vector, and input the  $\mathbf{H}'$  matrix as:

```
1 -1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 -1
```

Enter two restrictions on the second vector:

---

```
0 1 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
```

Enter two restrictions on the third vector:

```
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
```

Normalize on LM3NC, LPY, LPY, and LM3NC in the first dialog box, LYR in the second dialog, and RM in the third dialog.

In comparing the results to the text, note that for the  $\delta'_n$  and  $\beta'_{\perp 1}$  terms, the “const.” and “Ds831” columns in the text correspond to the TREND and TDS831 columns in the CATS output (the text refers to the first differences while the CATS output uses the original undifferenced description of these two terms).

## 18. Testing Hypotheses on the I(2) Model

Chapter 18 of the text is devoted to hypothesis testing for I(2) models. Example code for this chapter is provided in the file `CATSHandbook_Ch18.rpf`.

We'll be using the same CATS model explored in Chapter 17:

```
@cats(lags=2,shift,dum,dettrend=cidrift) 1973:1 2003:1
# lm3nc lpy lyr Rm Rb
# tDs831
# Dp831 Dp764
```

After starting CATS, set the ranks to three, one, and one using the *I(2)–Set Reduced Ranks* operation just as in the previous chapter.

### 18.1 Testing Price Homogeneity

#### 18.1.1 Long-run Price Homogeneity

As noted on page 332, you can implement the long-run price homogeneity test in two ways. The default restriction formulation in CATS is  $\tau = \mathbf{H}\varphi$  (described in the *I(2)–Change Restriction Formulation* dialog box as “[Beta / Tau] = H\*Phi”), so we will demonstrate that approach first.

To test the restriction, select *Same Restrictions on All Tau* from the *I(2)* menu. As the name of the menu operation applies, this allows you to enter a single set of restrictions and have them automatically applied to all the elements of the  $\tau$  matrix.

In this case, we want a price homogeneity restriction on the first two variables, so enter 1 as the number of restrictions, then enter the  $\mathbf{H}$  matrix from page 332, in  $\mathbf{H}'$  form, as shown below:

RESTRICTION DESIGN MATRIX

Restrictions on TAU (Tau = H\*Phi).

Please input H (transposed):

	LM3NC	LPY	LYR	RM	RB	TDS831	TREND
1	1.0000	-1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

OK Cancel

Click “OK” to initiate the test. The test statistic will be displayed in a dialog box. Click “OK” to display the estimation results. When prompted, normalize on LM3NC, LPY, LPY, and LM3NC in the first dialog box, LM3NC in the second dialog, and RM in the third dialog. After choosing the normalizations, you will see the estimated model output.

The results should match the top portion of Table 18.1 from the text.

To use the  $\mathbf{R}'\tau = 0$  formulation for restrictions, select *Change Restriction Formulation* from the I(2) menu and select the “R\*[Beta/Tau] = 0” choice. Then select *I(2)–Same Restrictions on All Tau* and enter 1 as the number of restrictions. Enter  $\mathbf{R}'$  as:

```
1 1 0 0 0 0 0
```

The rest proceeds as described above.

### 18.2 Medium-run Price Homogeneity

We have not yet replicated this section.

### 18.3 Assessing I(1) Results Within the I(2) Model

#### 18.3.1 Restrictions of the I(1) Model, Tables 18.2 and 18.3

Table 18.2 presents the results of several tests of restrictions on the trend components of the model. These are most easily accomplished using the  $\mathbf{R}'\tau = 0$  formulation, so make sure that option is selected using the *I(2)–Change Restriction Formulation* operation.

For the first hypothesis that both the trend and the broken trend variables can be excluded, select *Same Restrictions on All Tau*, enter two as the number of restrictions, and enter zero restrictions on both trend variables by writing  $\mathbf{R}'$  as:

```
0 0 0 0 0 1 0
0 0 0 0 0 0 1
```

The test result will be displayed in a dialog box. Click “OK” if you want to see all the restricted estimation results. You can accept the default normalizations or use the ones described above if desired.

The next two hypotheses examine whether the trend or the broken trend (but not both) can be omitted. To test the omission of the broken trend, do *Same Restrictions on All Tau*, enter one as the number of restrictions, and enter a zero restrictions on the TDS831 variable by putting a 1 in the next-to-last column:

```
0 0 0 0 0 1 0
```

For the restriction that the full-sample trend can be omitted, repeat the same steps but put the 1 in the last column (labeled TREND):

```
0 0 0 0 0 0 1
```

To test the hypothesis that the two terms are equal (that is, that the restriction *trend–trend83* can be imposed on all the terms), do *Same Restrictions on All Tau*, enter one as the number of restrictions, and enter ones for both the TDS831 and TREND columns:

```
0 0 0 0 0 1 1
```

Finally, Juselius repeats the long-run homogeneity with the additional condition that the broken trend can be restricted to zero. To implement this, select *I(2)–Same Restrictions on All Tau* and enter 2 as the number of restrictions. Enter  $\mathbf{R}'$  as:

```
1 1 0 0 0 0 0
0 0 0 0 0 1 0
```

The first row imposes the homogeneity restriction, while the second row omits the TDS831 term.

The estimates from this last restriction are presented in Table 18.3. To produce these, click on “OK” in the test results dialog box to print the estimates, and normalize on RM, LPY, RB, and LM3NC in the first dialog, LM3NC in the second dialog, and LPY in the third dialog.

#### 18.3.2 Restrictions on Beta, Table 18.4

Table 18.4 presents the results obtained by imposing the restrictions from Example 4 of Chapter 12 (which were originally introduced in Chapter 10). We show how to implement these restrictions below, but first a few things to keep in mind when comparing results with the earlier chapters:

First, we are now using LPY (prices) rather than DPY (inflation), and where the model used in Chapters 10 and 12 included a zero restriction on inflation, we now have the long-run homogeneity restriction involving money supply and prices.

Also, the order in which the second and third variables are listed is reversed from the earlier chapters. The order of the last two cointegrating vectors (in terms of which restrictions are imposed on each) is also flipped in Chapter 18 versus that used in Chapter 12.

Here's how you implement the restrictions:

First, use *I(2)–Change Restriction Formulation* and switch back to the “[Beta / Tau] = H\*Phi” formulation, then select *I(2)–Restrictions on Beta* and enter restrictions as follows:

Enter five restrictions on first vector, with the following **H'** matrix:

```
1 -1 -1 0 0 0 0
0 0 0 -1 1 0 0
```

This imposes the desired equality constraint on money supply, prices, and income, a separate equality constraint on the two interest rate series, and zero restrictions on the trend and broken-trend dummies.

Enter five restrictions on the second vector:

```
0 0 0 1 0 0 0
0 0 0 0 1 0 0
```

Here, we impose zero restrictions on everything but the two interest rate variables.

Enter six restrictions on the third vector:

```
0 0 1 0 0 0 0
```

This imposes zero restrictions on everything other than income.

You will see the test statistic reported in a dialog box. Click “OK” to print the estimates, and normalize on LM3NC, RM, LYR, and LM3NC in the first dialog, LPY in the second dialog, and RM in the third dialog.

### 18.3.3 Data Consistent Long-Run Structure, Table 18.5

Table 18.5 presents the results of a test of an over-identified long-run structure. Using the setup

```
@cats(lags=2, shift, dum, dettrend=cidrift) 1973:1 2003:1
# lm3nc lpy lyr Rm Rb
# tDs831
# Dp831 Dp764
```

and the same reduced ranks used above (three for the  $r$  value and one for the  $s_2$  value), select *I(2)–Restrictions on Beta* and enter the following restrictions:

Four restrictions on the first relation, with **H'** as

```
1 -1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 -1 0 0
```

Three restrictions on the second relation, with **H'** as

```
0 1 1 0 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
```

Two restrictions on the third relation, with **H'** as

```
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
```

## 19. Specific-to-General and General-to-Specific

Chapter 19 discusses the relative merits of specific-to-general and general-to-specific modelling approaches, particularly in the context of VAR models and cointegration analysis. The empirical work here is limited to looking at the changes in behavior over time of some additional data. Example code for this chapter is provided in the file `CATSHandbook_Ch19.rpf`.

### 19.1 Figures 19.2 and 19.3

Figure 19.2 plots the annual inflation rate and unemployment rate for Denmark. Use the commands below to read in the data. Note that we need to set the **CALENDAR** to start in the first quarter of 1970:

```
calendar(q) 1970:1
allocate 2003:1
open data book.xls
data(format=xls,org=obs) / lyr lpy lm3n Rm Rb lm3rC dpy
```

Next, we read in some additional series from a second file.

```
open data chpt_19_data.xls
data(format=xls,org=obs) / Lwc LPC wedge Lc DLPC Urt bdk ppp
```

LPC is the log of the CPI. We will describe the other series in Chapter 20.

To compute the annual inflation rate, we seasonally difference the price series:

```
diff(sdiffs=1) lpc / inflc
```

To include a vertical line at 1983:1 in the graph, we use a **SET** instruction and a logical expression to create a “grid” series which will be zero for all periods except 1983:1:

```
set grid1983 = t==1983:1
```

To create the graph, we use **SPGRAPH** with a **VFIELDS** option to put two graphs on the same page, stacked vertically. We use the **GRID1983** series with the **GRID** option on the **GRAPH** instructions to include a vertical line at 1983:1, denoting a point of fiscal and political changes as described on page 354 of the text. We also specify an explicit starting date of 1970:1 to match the axis labelling used in the book:

```
spgraph(vfields=2)
  graph(header="Annual Inflation Rate",grid=grid1983)
  # inflc 1970:1 *
  graph(header='Unemployment Rate',grid=grid1983)
  # urt 1970:1 *
spgraph(done)
```

For Figure 19.3, we need to compute an annual growth rate. This computation comes close to reproducing what is depicted in the text:

```
diff(sdiffs=1) lyr / growth
```

We also need to compute the real bond rate from the nominal bond rate. This uses the annual inflation rate computed above, divided by four to get quarterly inflation values:

```
set rrb = rb-inflc/4.
```

The graphs are produced using code similar to that above:

```
spgraph(vfields=2)
graph(header="Annual Real Growth Rates",grid=grid1983)
# growth 1970:1 *
graph(header="Nominal and Real Bond Rates",grid=grid1983,key=upright) 2
# rb 1970:1 *
# rrb 1970:1 *
spgraph(done)
```

## 19.2 Figure 19.4

Some of the data required for Figure 19.4 was not available, but the Real Exchange Rate is provided as the series PPP (the Purchasing Power Parity is the same as the Real Exchange Rate). In this case, the signs are flipped from what is used in the graph. One way to flip them back is to use the absolute value function:

```
set pppa = abs(ppp)
```

You can then graph the absolute value series:

```
graph(grid=grid1983,header="Real Exchange Rate") 1
# pppa
```

## 20. Wage, Price and Unemployment Dynamics

Chapter 20 illustrates the specific-to-general modelling approach discussed in Chapter 19, working through the process of adding additional series to a model. It also looks at ways of dealing with structural changes in the data. Example code for this chapter is provided in the file `CATSHandbook_Ch20.rpf`

### 20.1 The Data

For this chapter, we'll just read in the "chapter 19" data file:

```
calendar(q) 1971:1
allocate 2003:1
open data chpt_19_data.xls
data(format=xls,org=obs) / lwc lpc dlpc wedge lc urt bdk ppp
```

The series on the file contain the following information (see page 368 of the text for additional information)

Series	Description
LWc	Wage costs (labelled as $w$ in the book)
LPC	Consumer prices ( $p_c$ )
DLPc	First difference of LPC (i.e., consumer price inflation, $\Delta p_c$ )
Wedge	Internal price wedge ( $pp$ , defined as $p_c - p_y$ , where $p_y$ is the log GDP deflator)
LC	Log of labor productivity ( $pr$ , where $pr = y^r - e$ , where $y^r$ is log of real GDP and $e$ is the log of total employment)
Urt	Unemployment rate ( $U = U^T/L$ , the total number unemployed/labor force)
BDk	Long-term bond rate ( $R_b$ )
PPP	Purchasing power parity ( $q$ , equivalent to the real exchange rate)

As in chapter 19, we will flip the sign on PPP to match results presented in the table. Here, we will just replace the values of the PPP series, rather than creating another series, and we will multiply by minus one rather than using the `ABS()` function:

```
set ppp = -1*ppp
```

We also need to compute the real consumption wage by subtracting consumer prices from the wage costs:

```
set lwcr = lwc-lpc
```

Later, we will also need to create several time dummies.

Juselius generates and graphs a number of different series in Section 20.1 of the text. However, the provided data file does not include all of the data series required to produce these graphs, and in any case, most of the computational and graphing commands are quite straightforward, so we will not demonstrate most of these.

As two examples, we'll look at Figures 20.3 and 20.6.

First, we will need to create some series, including the nominal wage growth rate, computed from the quarterly wage costs:

```
set wagegrowth = (lwc-lwc{4})/4
```

We also need the annual inflation rate:

```
diff(sdiffs=1) lpc / inflc
set inflc = inflc/4
```

For Figure 20.3, we will define some vectors of strings containing longer labels for the graph keys.

```
compute keylab1 = ||"Inflation Rate","Unemployment Rate"||
compute keylab2 = ||"Inflation Rate","The Price Wedge"||
```

For the graphs, we use the `OVERLAY` option to show series with different scales on the same graph. We also use `PICTURE` options to enforce a consistent number of decimal places on the vertical axis labels:

```
spgraph(vfields=2,footer="Figure 20.3")
  graph(overlay=line,key=above,klabel=keylab1,ovkey,picture="#.###") 2
  # inflc
  # urt
  graph(overlay=line,key=above,klabel=keylab2,picture="#.###") 2
  # inflc
  # wedge
spgraph(done)
```

For Figure 20.6, we use the `MAX` option, which will force both graphs to use the same vertical scale by specifying a common maximum scale value:

```
spgraph(vfields=2,footer="Figure 20.6")
  graph(header="Nominal wage growth",frame=half,picture="#.##",max=.06)
  # wagegrowth
  graph(header="The CPI Inflation Rate",frame=half,picture="#.##",max=.06)
  # inflc
spgraph(done)
```

## 20.2 Nominal Versus Real Models

For reference, in terms of the variables we are using, the nominal and real models discussed on page 370 would be set up as follows:

Create the  $p_y$  variable (the log GDP deflator):

```
set py = wedge - lpc
```

The nominal system:

```
@cats(lags=2,dettrend=cidrift)
# Lwc py lpc Lc urt bdk ppp
```

The real system:

```
@cats(lags=2,dettrend=cidrift)
# Lwcr wedge Lc DLpc urt Bdk ppp
```

## 20.3 The EMS Period

Section 20.3 of the text examines three different models estimated over the EMS sample period (1983:1–2003:1). The three models are estimated as shown below—note the inclusion of specific start and end periods to limit the estimation to the EMS sample:

### Model 1:

```
@cats(lags=2,dettrend=cidrift) 1983:1 2003:1
# Lwcr wedge Lc DLpc urt
```

### Model 2

```
@cats(lags=2,dettrend=cidrift) 1983:1 2003:1
# Lwcr wedge Lc DLpc urt bdk
```

**Model 3**

```
@cats(lags=2,dettrend=cidrift) 1983:1 2003:1
# Lwcr wedge Lc DLPC urt bdk ppp
```

All of the results for this section are produced using this EMS sample period.

**20.3.1 Table 20.1**

To generate Table 20.1, just estimate each model in succession using the commands shown above, and use the *I(1)–Rank Test Statistics* operation to generate the rank statistics for each.

As indicated in the text, Juselius selects a rank of 2 for Model 1, rank 3 for Model 2, and rank 4 for Model 3. As each model includes one additional variable, increasing the rank by one as well maintains the presence of three stochastic trends in each model.

**20.3.2 Tables 20.2 and 20.3**

Table 20.2 presents the results of long-run exclusion and stationarity tests in Model 3. The first two rows present results of a long-run exclusion test for two different choices of rank (3 and 4), while the last line presents results for a stationarity test using rank 4.

To implement the tests, estimate Model 3 as shown above. Then select *Automated Tests–Variable Exclusion* to produce the results for the various rank choices, including the two shown in the table. Next, select *Automated Tests–Variable Stationarity*. When the dialog box pops up, turn off the restricted trend switch. This produces stationarity tests for all variables for all choices of rank, including the rank 4 choice shown in the table.

Table 20.3 presents the results of some tests on  $\alpha$ . To reproduce these, just estimate the respective model and use the *Automated Tests–Weak Exogeneity* and *Automated Tests–Unit Vector in Alpha* operations.

You can compare the tables in the book against the CATS output by matching up the degrees of freedom shown in the book with the appropriate value in the DGF (degrees of freedom) column in the output.

**20.3.3 Tables 20.4**

Table 20.4 presents the estimates of the  $\Pi$  matrix for all three models, given the choice of rank described earlier (rank 2 for Model 1, rank 3 for Model 2, rank 4 for Model 3). To produce these results, just estimate each model in turn, and use *I(1)–Set Rank of Pi* to set the ranks accordingly.

Note: The values for the TREND term would need to be scaled up by 100 to match the results in the book. You can do this easily in CATS by selecting the *Preferences* operation from the CATS menu and setting the “Scale Restricted Trend By” field to 100.

**20.3.4 Identifying the Long-Run Structure, Table 20.5**

Table 20.5 reports the estimated  $\alpha$  and  $\beta$  values for three identified  $\beta$  structures—one for each model. Here’s how you can reproduce these.

**For Model 1**

Estimate the set up instructions and set the rank to 2. Use *Change Restriction Formulation* on the *I(1)* menu to select the  $\mathbf{R}^* \text{Beta} = 0$  restriction formulation. Then use *I(1)–Restrictions on each Beta Vector* to impose the identifying restrictions.

Select one restriction on vector 1, putting a 1 in the DLPC column of the  $\mathbf{R}'$  dialog box, leaving zeros in all other columns. For vector 2, impose 2 restrictions on vector 2, putting 1s in the WEDGE column in row one, and the LC column in row two. Normalize on LC and DLPC, respectively.

**For Model 2**

Estimate model 2, and set the rank to 3 and select the  $\mathbf{R}^* \text{Beta} = 0$  restriction formulation.

Select *I(1)–Restrictions on each Beta Vector* and impose three restrictions on vector 1, entering 1s in the LWCR, DLPC, and BDK columns. Impose four restrictions on vector 2, with 1s in the LWCR, WEDGE, LC, and TREND columns. Impose four restrictions on vector 3, with 1s in the WEDGE, LC, DLPC, and TREND columns. Normalize on LC, DLPC, and URT respectively.

**For Model 3**

Estimate model 3, and set the rank to 4 and select the  $\mathbf{R}^* \text{Beta} = 0$  restriction formulation. Select *I(1)–Restrictions on each Beta Vector* and use the following restrictions:

Impose four restrictions on vector 1 with the following  $\mathbf{R}'$  matrix:

```
1  0  0  0  0  0  0  0
0  0  0  1  0  0  0  0
0  0  0  0  0  1  0  0
0  0  0  0  0  0  1  0
```

Impose five restrictions on vector 2 with the following  $\mathbf{R}'$  matrix:

```
1  0  0  0  0  0  0  0
0  1  0  0  0  0  0  0
0  0  1  0  0  0  0  0
0  0  0  0  0  0  1  0
0  0  0  0  0  0  0  1
```

Impose 4 restrictions on vector 3, with the following  $\mathbf{R}'$  matrix:

```
0  1  0  0  0  0  0  0
0  0  1  0  0  0  0  0
0  0  0  0  0  0  1  0
0  0  0  0  0  0  0  1
```

Impose 5 restrictions on vector 4, with the following  $\mathbf{R}'$  matrix:

```
1 -1  0  0  0  0  0  0
1  0  1  0  0  0  0  0
0  0  0  1  0  0  0  0
0  0  0  0  1  0  0  0
0  0  0  0  0  1  0  0
```

Note: these restrictions are reported as not-identifying, so you have to tell CATS to estimate with the non-identifying restrictions. Normalize on LC, DLPC, URT, and LC. When prompted, set the degrees of freedom to 7.

**20.4 The Post-Bretton–Woods Regime**

The same basic approach is repeated in this section, but using the sample range 1971:1–1983:1. Also, the bond rate is excluded from model 3 for this regime. As noted in the text, there are potential issues with the use of real variables given the possible failure of the long-run price homogeneity condition.

The reported results are also based on the inclusion of a dummy variable for 1975, defined using the following instructions:

```
set D75 / = 0.0
compute D75(1975:4) = 1
compute D75(1976:1) = -0.5
compute D75(1976:2) = -0.5
```

The setup commands for the three models are as follows:

**Model 1**

```
@cats(lags=2,dum,dettrend=cidrift) 1971:1 1983:1
# Lwcr wedge Lc DLpc urt
# D75
```

**Model 2**

```
@cats(lags=2,dum,dettrend=cidrift) 1971:1 1983:1
# Lwcr wedge Lc DLpc urt bdk
# D75
```

**Model 3**

```
@cats(lags=2,dum,dettrend=cidrift) 1971:1 1983:1
# Lwcr wedge Lc DLpc urt ppp
# D75
```

As in the text, we will focus on the results for Model 3 here.

**20.4.1 Specification Tests, Tables 20.6 and 20.7**

To produce Table 20.6, estimate Model 3 as shown above and select *Rank Test Statistics* from the *I(1)* menu. For Table 20.7, use the *Variable Exclusion*, *Variable Stationarity* (with the restricted trend switch turned off), *Weak Exogeneity*, and *Unit Vector in Alpha* operations on the *Automated Tests* menu.

**20.4.2 Investigating the  $\Pi$  Matrix, Table 20.8**

As noted on page 381, Juselius selects a rank of 3 for further analysis, so to reproduce Table 20.8, select the *Set Rank of Pi* operation from the *I(1)* menu and enter 3 as the rank. Accept the default normalization when prompted.

As noted in the text, the TREND term has again been scaled up by a factor of 100, so use the *CATS-Preferences* operation to do the same if desired.

**20.4.3 Identification, Table 20.9**

To reproduce Table 20.9, use *Change Restriction Formulation* on the *I(1)* menu to select the  $\mathbf{R}^* \text{Beta} = 0$  restriction formulation. Then select *I(1)-Restrictions on each Beta Vector*.

Enter three as the number of restrictions on vector 1, with the following  $\mathbf{R}'$  matrix:

```
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
```

Enter two as the number of restrictions on vector 2, with the following  $\mathbf{R}'$  matrix:

```
1 0 0 0 0 0 0
0 1 0 0 0 0 0
```

Enter four as the number of restrictions on vector 3, with the following  $\mathbf{R}'$  matrix:

```
1 -1 0 0 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
```

Normalize on LWCR, URT, and WEDGE respectively.

## 21. Foreign Transmission Effects

In this chapter, Juselius demonstrates the specific-to-general approach and examines international transmission mechanisms by incorporating German interest rate and price data. Example code for this chapter is provided in the file `CATSHandbook_Ch21.rpf`.

### 21.1 The Data

The original data was not directly available, so our replication program uses some data sourced from the OECD's Main Economic Indicators database, which we provide in a separate RATS format data file. Unfortunately, the short-term interest rate series seems to differ significantly from the one used in producing the results in the textbook, so some of the results do not match exactly.

Here are the instructions we use to read in the data from our two files:

```
calendar(q) 1971:1
allocate 2003:1
open data chpt_19_data.xls
data(format=xls,org=obs) / Lwc Lpc DLpc wedge Lc Urt bdk ppp

open data catsoecd.rat
data(format=rats,ver)
```

For convenience, we use **SET** instructions to copy the data into a set of series using a consistent naming scheme. Also, we need to divide the interest rate series by 400 for comparability with the quarterly inflation rates in log differences (see page 395 of the text). The BDK series from the Chapter 19 file is already in this form, so is not divided here:

```
* Rename short-term interest rate series:
set shortdnk = dnkirstcb01st/400.
set shortger = deuir3tib01st/400.

* Rename long-term bond rate series:
set longger = deuir1tlt01st/400
set longdnk = bdk

* Rename price series
set lcpidnk = lpc
set lcpiger = log(deucpaltt01xob)
```

#### 21.1.1 Figure 21.1, Price Differential and PPP

Below are commands that reproduce Figure 21.1, excluding the nominal exchange rate series as we do not have the spot exchange rate available required to compute this.

```
* Compute the Denmark vs. Germany price differential:
set pricediff = lcpidnk - lcpiger

spgraph(vfields=2, footer="Figure 21.1")
graph(key=upleft, picture="#.###") 1
# pricediff
graph(key=upleft, picture="#.###") 1
# ppp
spgraph(done)
```

#### 21.1.2 Figure 21.2, Interest Rate Differentials

This graph plots the differential between the long-term bond rates for the two countries, and the differential between the short-term interest rates. As noted earlier, our short-term interest rate data for Germany differs significantly, so the bottom portion of the graph does not match what is shown in the text.

```

* Denmark vs. Germany rate differentials:
set shortdiff = shortdnk-shortger
set longdiff = longdnk-longger

spgraph(vfields=2,footer="Figure 21.2")
  graph(key=upleft,picture="#.###") 1
  # longdiff
  graph(key=upleft,picture="#.###") 1
  # shortdiff
spgraph(done)

```

### 21.1.3 Figure 21.3, Inflation Rate Differential

Again we do not have the spot rates, so here we only show the inflation rate differential. This graph uses annual inflation rates:

```

diff(sdiffs=1) lcpidnk / inflratednk
diff(sdiffs=1) lcpiger / inflrateger
set infldiff = inflratednk - inflrateger

spgraph(vfields=1,footer="Figure 21.3 (partial)")
  graph(key=upleft,picture="#.###") 1
  # infldiff
spgraph(done)

```

### 21.1.4 Figure 21.4, Long-Short Spreads

This graph shows the spreads between the long- and short-term rates for each country. Again, the German graph does not match that shown in the book due to the differences in the short-term series.

```

* Compute long-short spreads for each country:
set spreaddnk = longdnk-shortdnk
set spreadger = longger-shortger

spgraph(vfields=2,footer="Figure 21.4")
  graph(key=upleft,picture="#.###") 1
  # spreaddnk
  graph(key=upleft,picture="#.###") 1
  # spreadger
spgraph(done)

```

### 21.1.5 Figures 21.5 and 21.6, Real Long- and Short-Term Bond Rates

These two graphs show the real long-term (Figure 21.5) and short-term (Figure 21.6) interest rates, along with four-period moving averages. Again, the short-term graph for Germany will not match that shown in the book.

First, we calculate quarter-to-quarter inflation rates:

```

diff(diffs=1) lcpidnk / infldnk
diff(diffs=1) lcpiger / inflger

```

Then we use the inflation data to compute the real-valued long- and short-term rates:

```

set longdnkr = longdnk - infldnk
set longgerr = longger - inflger
set shortdnkr = shortdnk - infldnk
set shortgerr = shortger - inflger

```

Now, we use **MVSTATS** to compute the four-period moving averages:

```
mvstats (width=4, centered, means=longdnkrma) longdnkr
mvstats (width=4, centered, means=longgerma) longgerr
```

```
mvstats (width=4, centered, means=shortdnkrma) shortdnkr
mvstats (width=4, centered, means=shortgerma) shortgerr
```

Finally, we plot the two graphs. If you would like the MA series to be drawn as a solid line and the actual data drawn as a dashed line, you can add the `PATTERNS` option to the `GRAPH` instructions (or use the “Black&White/Color” button on the graph window in RATS to switch back and forth between representations). You can also use Graph Style Sheets to generate black-and-white graphs that use shades of gray to distinguish series.

```
* Figure 21.5:
spgraph (vfields=2, footer="Figure 21.5")
  graph (key=upleft, picture="#.###") 2
  # longdnkrma
  # longdnkr
  graph (key=upleft, picture="#.###") 2
  # longgerma
  # longgerr
spgraph (done)
```

```
* Figure 21.6:
spgraph (vfields=2, footer="Figure 21.6")
  graph (key=upleft, picture="#.###") 2
  # shortdnkrma
  # shortdnkr
  graph (key=upleft, picture="#.###") 2
  # shortgerma
  # shortgerr
spgraph (done)
```

### 21.1.6 Figure 21.7, Real Interest Rate Differentials

Finally, Figure 21.7 plots the differentials of the real long-term and short-term rates between the two countries.

```
* Compute the differentials:
set realdiffshort = shortdnkr - shortgerr
set realdifflong = longdnkr - longgerr
```

```
* Generate the graphs:
spgraph (vfields=2, footer="Figure 21.7")
  graph (key=upleft, picture="#.###") 1
  # realdifflong
  graph (key=upleft, picture="#.###") 1
  # realdiffshort
spgraph (done)
```

## 21.2 The Models

### 21.2.1 Dummy Variables

Before proceeding, we need to create several dummy variable series that will be used in our cointegration models. Several of these have been used earlier in the text, but a couple of the “permanent blip” series are new.

```
* Transitional dummy at 1975:4:
set Dt754 = 0.0
compute Dt754 (1975:4) = 1
compute Dt754 (1976:1) = -0.5
compute Dt754 (1976:2) = -0.5
```

```

* Permanent blip dummies for 1976:4, 1984:2, and 1993:1

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

set Dp842 = 0.0
compute Dp842 (1984:2) = 1

set Dp931 = 0.0
compute Dp931 (1993:1) = 1

* Shift dummy at 1983:1
set Ds831 = T>=83:1

```

### 21.2.2 The Two Cointegration Models

The analysis is done using two models. The first is the smaller, more specific model, which omits the short-term interest rate series. The second is the larger, more general model, which adds the short-term rates as well as a couple of corresponding dummy variables.

We will show the basic setup code for both models here. Be sure to use the appropriate setup for reproducing any of the subsequent results described below. Also, note the use of the CIMEAN choice for DETTREND here, as opposed to the CIDRIFT model used in most of our earlier examples:

```

* Model 1 (smaller model):
@cats(lags=2,shift,dum,dettrend=cimean) 1973:1 2003:1
# infldnk inflger longdnk longger ppp
# Ds831
# Dt754 Dp931

* Model 2 (larger model):
@cats(lags=2,shift,dum,dettrend=cimean) 1973:1 2003:1
# infldnk inflger longdnk longger ppp shortdnk shortger
# Ds831
# Dt754 Dp764 Dp842 Dp931

```

### 21.2.3 Tables 21.1 and 21.2

Table 21.1 presents statistics on the various choices of rank for both models. To reproduce this, estimate the desired model using the appropriate setup code as shown above. You can reproduce most of the information in the table using the *Rank Test Statistics* operation on the *I(1)* menu.

For the maximum root ( $\rho_{\max}$ ), use the *Roots of the Companion Matrix* operation on the *Graphics* menu, select all of the models from the dialog box, and then refer to the Modulus column in the results.

Note the apparent typo in the book, where the largest root for rank 3 is described as 0.80 in the text, but shown as 0.90 in the table—the latter value appears correct, and is the one produced by RATS. RATS also produces 0.51 rather than 0.52 for the largest root for the rank 1 case.

To produce Table 21.2, use the *Variable Stationarity*, *Weak Exogeneity*, and *Unit Vector in Alpha* operations on the *Automated Tests* menu and refer to the results for the indicated rank (and corresponding degrees of freedom) cases. For the *Unit Vector in Alpha* test, leave the “restricted constant” and “shift dummies” check boxes turned on.

### 21.2.4 Tables 21.3 – 2.15, Long Run Relations

Tables 21.3 and 21.4 present the values of  $\beta$  and  $\alpha$  for identified  $\alpha$  structures for the two models.

For the smaller model, load CATS using the Model 1 setup instructions from above, and set the rank to 2.

Use *I(1)–Change Restriction Formulation* to select the “ $R \cdot \beta = 0$ ” restriction formulation, then select *I(1)–Restrictions on Each Beta Vector* to impose restrictions.

For the first vector, enter 2 for the number of restrictions, and enter 1’s in the INFLGER column in row 1 and the LONGGER column in row 2.

For the second vector, again enter two as the number of restrictions, and enter  $R'$  as:

```
1 0 1 0 0 0 0
0 1 0 1 0 0 0
```

Normalize on INFLDNK and INFLGER, respectively.

For Table 21.4, the variables are listed in a different order than was used in the earlier tables, so use this setup code to estimate the model (if you wish to use the original setup, you will need to change the column-order of the restrictions):

```
@cats(lags=2,shift,dum,dettrend=cimean) 1973:1 2003:1
# infldnk inflger longdnk longger shortdnk shortger ppp
# Ds831
# Dt754 Dp764 Dp842 Dp931
```

Set the rank to 4 for the large model and, if necessary, use *I(1)–Change Restriction Formulation* to select the “ $R \cdot \beta = 0$ ” formulation. Use *I(1)–Restrictions on Each Beta Vector* and enter the restrictions as described below. Note that the results will not match the text exactly due to the differences in the German short-term interest rate data.

Enter five restrictions on vector 1, with the following  $R'$  matrix:

```
1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0
```

Four restrictions on vector 2:

```
1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 0
0 0 0 0 1 1 0 0 0
```

Four restrictions on vector 3:

```
1 0 1 0 0 0 0 0 0
0 1 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0
```

Four restrictions on vector 4:

```
0 1 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0
```

Normalize on LONGDNK, LONGDNK, INFLGER, and INFLDNK respectively.

For Table 21.5, use *Misc–MA Representation* to display the loadings to the common trends. The results in 21.5 are labelled as “The Loadings to the Common Trends, BETA\_ORT(tilde)” in the CATS output.

## 22. Collecting the Threads

Chapter 22 of the text ties together the work done in the previous chapters by examining a larger model and a comprehensive suite of (over)identifying restrictions, derived from the restrictions examined earlier. Some of the material in the textbook chapter is not reproduced here.

The code for this example is provided on the file `CATSHandbook_Ch22.rpf`.

### 22.1 The Data and Model

The following commands read in and create the necessary series (in some cases, just renaming existing series for convenience in constructing the error correction terms).

```
calendar(q) 1971:1
allocate 2003:1

open data book.xls
data(format=xls,org=obs) / lyr lpy lm3n Rm Rb lm3rC dpy

open data chpt_19_data.xls
data(format=xls,org=obs) / Lwc Lpc DLpc wedge Lc Urt bdk ppp

open data catsoecd.rat
data(format=rats,ver)

* Rename short-term interest rate series:
set shortdnk = dnkirstcb01st/400.
set shortger = DEUIR3TIB01ST/400.

* Rename long-term bond rates:
set longger = DEUIRLTLT01ST/400
set longdnk = bdk

* Rename price series
set lcpidnk = lpc
set lcpiger = log(DEUCPALTT01XOB)

set lwcr = Lwc-Lpc

* Annual and quarterly inflation rates:
diff(sdiffs=1) lcpidnk / inflratednk
diff(sdiffs=1) lcpiger / inflrateger
diff(diffs=1) lcpidnk / qinflratednk
diff(diffs=1) lcpiger / qinflrateger

* Rename some variables to match text
set pr = lc
set w = lwcr
set q = -1*ppp
set u = urt

* Dummy variables:
set Dt754 = 0.0
compute Dt754(1975:4) = 1
compute Dt754(1976:1) = -0.5
compute Dt754(1976:2) = -0.5

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

```

set ds831 = T>=83:1

* Define error correction terms per Table 22.2. These
* are derived from earlier analysis (e.g., Tables 20.5 and 20.9)

set mecml = lm3rc - lyr - 13.27*(Rm-Rb) - 0.15*ds831
set mec3 = Rm - 0.81*Rb - 0.10*ds831

set w73ecm1 = w - lpc + 0.41*(lpc - lpy) - 0.59*q - 0.01*t
set w73ecm2 = u + 5.06*dlpc - 0.33*q + 1.05*(pr - 0.01*t)
set w73ecm3 = w - lpy - 1.38*pr - 9.93*dlpc

set w83ecm1 = (pr - .01*t) - 0.39*(lpc - lpy) - 0.93*q
set w83ecm2 = dlpc - 0.53*Rb + 0.08*u
set w83ecm3 = u - 9.51*Rb - 0.55*(w-lpc)
set w83ecm4 = (w - lpy - pr) - 0.99*q

set fecm3 = (longger - lcpiger) - 0.33*(longdnc - lcpidnc) + 0.03*q
set fecm4 = dlpc - 0.25*Rb - 0.04*q

* Wedge regimes
set wedge73 = %if(t>=1973:1.and.t<=1984:4,wedge,0)
set wedge83 = %if(t>=1983:1.and.t<=2003:1,wedge,0)

* Take differences for VECM form
diff lm3rc / dlm3rc
diff dpy / ddp
diff rm / dr
diff rb / drb
diff lwcr / dlwcr
diff pr / dpr
diff u / du
diff q / dq
diff DLPC / ddLPC
diff ds831 / dp831
diff wedge73 / dwedge73
diff wedge83 / dwedge83
diff longger / dlongger
diff lcpiger / dlcpiger

```

## 22.2 Tables 22.1 and 22.2

Note: For reference, we define the full VECM model in the example program. The commands below create the restricted model used to produce Tables 22.1 and 22.2:

```

equation vecm1eqr dlm3rc
# drb{1} mecml{1} w73ecm1{1} w73ecm2{1} w73ecm3{1} w83ecm1{1} $
w83ecm3{1}

equation vecm2eqr ddlpc
# dwedge73 dwedge83 ddlpc{1} mecml{1} w73ecm1{1} w73ecm2{1} w83ecm2{1} $
w83ecm4{1} dt754

equation vecm3eqr dr
# drb{1} dlm3rc{1} dq{1} mec3{1} w73ecm1{1} w73ecm2{1} w73ecm3{1} $
w83ecm4{1} dp764

equation vecm4eqr drb
# dlcpiger dlongger drb{1} dlm3rc{1} du{1} dq{1} mec3{1} w83ecm1{1} $
w83ecm3{1} fecm3{1} dt754 dp831

```

```

equation vecm5eqr dlwcr
# dwedge73 dwedge83 w73ecm2{1} w73ecm3{1} w83ecm1{1} $
w83ecm4{1}

equation vecm6eqr dpr
# dwedge83 drb{1} ddlpc{1} mecm3{1} w73ecm1{1} w73ecm2{1} w83ecm1{1} $
w83ecm2{1} fecm4{1}

equation vecm7eqr du
# drb{1} du{1} w73ecm1{1} w73ecm3{1} $
w83ecm3{1} w83ecm4{1}

equation vecm8eqr dq
# dwedge73 dlcpiger dlm3rc{1} du{1} mecm3{1} w73ecm1{1} w83ecm2{1} $
w83ecm4{1} fecm3{1}

* Group equations into a MODEL:
group vecmmodr vecmleqr vecm2eqr vecm3eqr vecm4eqr vecm5eqr vecm6eqr $
vecm7eqr vecm8eqr

* Estimate the model, saving the residuals in an array of series:
estimate(model=vecmmodr,resids=residvec)

```

### 22.3 Table 22.3

Table 22.3 presents the standardized residual covariance matrix, with the residual standard errors shown on the diagonal. We can generate the main matrix using a **CMOM** instruction with the **CORR** option, and then replace the diagonal terms with the square roots of the diagonal terms of the residual covariance matrix (%SIGMA) for the standard errors.

```

cmom(corr,matrix=cmommat)
# residvec

* Replace diagonals with standard errors
ewise cmommat(i,j) = %if(i==j,sqrt(%sigma(i,j)),cmommat(i,j))

display "Table 22.3"
display #####.#### cmommat

```