Chapter 17

Sample Study C and New R Packages

THE CORE SAMPLE STUDY for **Part V** Programming as a Contributor is Sun (2011). In this chapter, the underlying statistical model and manuscript and program versions for this study are presented first. The research issue is asymmetric price transmission (APT) between China and Vietnam in the import wooden bed market of the United States. This is closely related to the issue examined in Wan et al. (2010a), i.e., the main sample study for **Part IV** Programming as a Wrapper. At the stage of proposal and project design, some aspects of designing several projects in one area are discussed. The relevant discussion can be found at **Section 5.3.3** Design with challenging models (Sun 2011) on page 73.

The model employed is at the frontier of time series statistics, i.e., nonlinear threshold cointegration analysis. It involves hundreds of linear regressions even for a very small data set. Thus, writing new functions and even preparing a new package are needed to have an efficient data analysis. For this specific model, a new package called `apt` is created. The program version for Sun (2011) is organized with the help of this package, so the whole program has become more concise and readable.

17.1 Manuscript version for Sun (2011)

Recall that an empirical study has three versions: proposal, program, and manuscript. A proposal provides a guide like a road map for setting up the first draft of a manuscript. Like an engine, an R program can generate detailed tables and figures for the final manuscript. For this study, the brief proposal is presented at **Section 5.3.3** Design with challenging models (Sun 2011). The R program for this study is presented later in this chapter. The final manuscript version is published as Sun (2011). Below is the very first manuscript version that is developed from the proposal.

In constructing the first manuscript version for an empirical study, the key components are the tables and figures. The contents should be predicted as much as possible before a researcher works on an R program. The prediction is based on the understanding of the issue, data, model, and literature. The more a researcher can predict at this stage, the more efficient the programming will become. At the end, both the content and format of tables and figures need to be written down in the manuscript draft. For example, the results of Engle-Granger and threshold cointegration tests are reported in combination as Table 3 in Sun (2011). The first draft of these results is presented here as **Table 17.1**. Some hypothetical values are put in the columns to provide formatting guides for R programming later.

The First Manuscript Version for Sun (2011)

1. *Abstract* (200 words). Have one or two sentences for research issue, study need, objective, methodology, data, results, and contributions.

2. *Introduction* (3 pages in double line spacing). Have a paragraph for each of the following items: an overview of wooden bed imports in the United States, market price analyses and asymmetric price transmission (APT), sources of APT, models of APT, objective, and manuscript organization.

3. *Import wooden bed market in the United States* (4 pages). A review of the US wooden bed market is presented, with the emphasis on expansion of China and Vietnam in the import wooden bed market.

   — Factors behind China's export growth

   — Antidumping investigation against China

   — Vietnam's growth

4. *Methodology* (6 pages): A brief introduction of the methods and then three subsections.

   — Linear cointegration analysis

   — Threshold cointegration analysis

   — Asymmetric error correction model with threshold cointegration

5. *Data and software* (0.5 page). Monthly cost-insurance-freight values in dollar and quantities in piece are reported by country. The period covered in this study is from January 2002 to January 2010. Threshold cointegration and asymmetric error correction model are combined and used in this study. A new R package named as `apt` is created in the process.

6. *Empirical results* (4 pages of text, 4 pages of tables, and 3 pages of figure).

   — Descriptive statistics and unit root test

   — Results of the linear cointegration analysis

   — Results of the threshold cointegration analysis

   — Results of the asymmetric error correction model

   Table 1. Results of descriptive statistics and unit root tests

   Table 2. Results of Johansen cointegration tests on the import prices

   Table 3. Results of Engle-Granger and threshold cointegration tests

   Table 4. Results of asymmetric error correction model

   Figure 1. Monthly import values of wooden beds from China and Vietnam

   Figure 2. Monthly import prices of wooden beds from China and Vietnam

   Figure 3. Sum of squared errors by threshold value for threshold selection

7. *Conclusion and discussions* (3 pages). A brief summary of the study is presented first. Then about three key results from the empirical findings will be highlighted and discussed.

8. *References* (3 pages). No more than 30 studies will be cited.

end

Table 17.1 A draft table for the cointegration analyses in Sun (2011)

Item	Engle	TAR	CTAR	MTAR	CMTAR
Estimate					
Threshold	—	0		0	
ρ_1	−0.666***	−0.666***			
	(−3.333)	(−3.333)			
ρ_2	—	−0.666***			
	—	(−3.333)			
Diagnostics					
AIC	888.888	888.888			
BIC					
QLB(4)					
QLB(8)					
QLB(12)					
Hypotheses					
$\Phi(H_0 : \rho_1 = \rho_2 = 0)$	—	10.123***			
$F(H_0 : \rho_1 = \rho_2)$	—	4.444***			

17.2 Statistics: threshold cointegration and APT

In this section, the relevant statistics for threshold cointegration and asymmetric price transmission is presented. Emphases are put on the key information relevant for R implementation in the `apt` package. For a comprehensive coverage of this methodology, read these references cited in Sun (2011). Nonstationarity and unit root tests, Johansen-Juselius cointegration analysis, and most model diagnostics are not covered here for brevity. In contrast, Engle-Granger linear cointegration, threshold cointegration, and asymmetric error correction model are described here with some detail. Linear cointegration analysis is the foundation of threshold cointegration.

17.2.1 Linear cointegration analysis

For linear cointegration analysis, there exist two major methods: Johansen-Juselius and Engle-Granger two-step approaches (Enders, 2010). Both of them assume symmetric relations between variables. The Johansen approach is a multivariate generalization of the Dickey-Fuller test. The Engle-Granger approach is the foundation of threshold cointegration so it is explained in detail first.

The focal variables here are monthly import prices of wooden beds from two supplying countries, i.e., Vietnam (V_t) and China (H_t). Their properties of nonstationarity and order of integration can be assessed using the Augmented Dickey-Fuller test. If both series have a unit root, then it is appropriate to conduct cointegration analysis to evaluate their interaction. With the Engle-Granger two-stage approach, the property of residuals from the long-term equilibrium relation is analyzed (Engle and Granger, 1987). For the two focal price variables, the two-stage approach can be expressed as:

$$V_t = \alpha_0 + \alpha_1 H_t + \xi_t \tag{17.1}$$

$$\Delta\widehat{\xi}_t = \rho\,\widehat{\xi}_{t-1} + \sum_{i=1}^{P} \phi_i\,\Delta\widehat{\xi}_{t-i} + \mu_t \tag{17.2}$$

where α_0, α_1, ρ, and ϕ_i are coefficients, ξ_t is the error term, $\widehat{\xi}_t$ is the estimated residuals, Δ indicates the first difference, μ_t is a white noise disturbance term, and P is the lag number.

In the first stage of estimating the long-term relation among the price variables, the price of China is chosen to be placed on the right side and assumed to be the driving force. This considers the fact that China has been the leading supplier in the import wooden bed market of the United States over the study period from 2002 to 2010. In the second stage, the estimated residuals $\widehat{\xi}_t$ are used to conduct a unit root test. Special critical values are needed for this test because the series is not raw data but a residual series. The number of lags is chosen so there is no serial correlation in the regression residuals μ_t. It can be selected with several statistics, e.g., the Akaike Information Criterion (AIC) or Ljung-Box Q test. If the null hypothesis of $\rho = 0$ is rejected, then the residual series from the long-term equilibrium is stationary and the focal variables of V_t and H_t are cointegrated.

17.2.2 Threshold cointegration analysis

In recent years, nonlinear cointegration has been increasingly used in price transmission studies. Among various developments of nonlinear cointegration, one branch is called threshold cointegration. The nonlinearity comes from two linear regressions combined, and the linear regressions are based on the above Engle-Granger linear cointegration approach. Thus, the threshold cointegration regression considered here is *piecewise only* and not smooth. Specifically, Enders and Siklos (2001) propose a two-regime threshold cointegration approach to entail asymmetric adjustment in cointegration analysis. This modifies **Equation (17.2)** such that:

$$\Delta\widehat{\xi}_t = \rho_1 \, I_t \, \widehat{\xi}_{t-1} + \rho_2 \, (1 - I_t) \, \widehat{\xi}_{t-1} + \sum_{i=1}^{P} \varphi_i \, \Delta\widehat{\xi}_{t-i} + \mu_t \tag{17.3}$$

$$I_t = 1 \text{ if } \widehat{\xi}_{t-1} \geq \tau, 0 \text{ otherwise; or} \tag{17.4}$$

$$I_t = 1 \text{ if } \Delta\widehat{\xi}_{t-1} \geq \tau, 0 \text{ otherwise} \tag{17.5}$$

where I_t is the Heaviside indicator, P the number of lags, ρ_1, ρ_2, and φ_i the coefficients, and τ the threshold value. The lag (P) is specified to account for serially correlated residuals and it can be similarly selected as in linear cointegration analysis.

The Heaviside indicator I_t can be specified with two alternative definitions of the threshold variable, either the lagged residual ($\widehat{\xi}_{t-1}$) or the change of the lagged residual ($\Delta\widehat{\xi}_{t-1}$). **Equations (17.3)** and **(17.4)** together have been referred to as the Threshold Autoregression (TAR) model, while **Equations (17.3)** and **(17.5)** are named as the Momentum Threshold Autoregression (MTAR) model. The threshold value τ can be specified as zero, or it can be estimated. Thus, a total of four models can be estimated. They are TAR with $\tau = 0$, consistent TAR with τ estimated, MTAR with $\tau = 0$, and consistent MTAR with τ estimated. In general, a model with the lowest AIC is deemed to be the most appropriate.

Insights into the asymmetric adjustment in the context of a long-term cointegration relation can be obtained with two tests. First, an F-test is employed to examine the null hypothesis of no cointegration ($H_0 : \rho_1 = \rho_2 = 0$) against the alternative of cointegration with either TAR or MTAR threshold adjustment. The test statistic is represented by Φ. This test does not follow a standard distribution and the critical values in Enders and Siklos (2001) should be used. The second one is a standard F-test to evaluate the null hypothesis of symmetric adjustment in the long-term equilibrium ($H_0 : \rho_1 = \rho_2$). Rejection of the null hypothesis indicates the existence of an asymmetric adjustment process. Results from the two tests are the key outputs from threshold cointegration analysis.

The challenge of threshold cointegration analysis comes from estimating the threshold value of τ. With a given value for τ, **Equation (17.3)** is just a linear regression and it can be easily estimated by any software application, e.g., the `lm()` function in R. At present, the method by Chan (1993) has been widely followed to obtain a consistent estimate of the threshold value. A super consistent estimate of the threshold value can be attained with several steps. First, the process involves sorting in ascending order the threshold variable, i.e., $\widehat{\xi}_{t-1}$ for the TAR model or the $\Delta\widehat{\xi}_{t-1}$ for the MTAR model. Second, the possible threshold values are determined. If the threshold value is to be meaningful, the threshold variable must actually cross the threshold value (Enders, 2010). Thus, the threshold value τ should lie between the maximum and minimum value of the threshold variable. In practice, the highest and lowest 15% of the values are excluded from the search to ensure an adequate number of observations on each side. The middle 70% values of the sorted threshold variable are generally used as potential threshold values. The percentage can be higher if the total number of observations in a study is larger, e.g., 90% for 1,000 observations. Third, the TAR or MTAR model is estimated with each potential threshold value. The sum of squared errors for each trial can be calculated and the relation between the sum of squared errors and the threshold value can be examined. Finally, the threshold value that minimizes the sum of squared errors is deemed to be the consistent estimate of the threshold.

17.2.3 Asymmetric error correction model

The Granger representation theorem (Engle and Granger, 1987) states that an error correction model can be estimated where all the variables in consideration are cointegrated. The specification assumes that the adjustment process due to disequilibrium among the variables is symmetric. Two extensions on the standard specification in the error correction model have been made for analyzing asymmetric price transmission. Granger and Lee (1989) first extend the specification to the case of asymmetric adjustments. Error correction terms and first differences on the variables are decomposed into positive and negative components. This allows detailed examinations on whether positive and negative price differences have asymmetric effects on the dynamic behavior of prices. The second extension follows the development of threshold cointegration (Enders and Granger, 1998). When the presence of threshold cointegration is validated, the error correction terms are modified further.

The asymmetric error correction model with threshold cointegration in Sun (2011) is developed as follows:

$$\begin{aligned}\Delta H_t = \theta_H + \delta_H^+ E_{t-1}^+ + \delta_H^- E_{t-1}^- + \sum_{j=1}^{J} \alpha_{Hj}^+ \Delta H_{t-j}^+ + \sum_{j=1}^{J} \alpha_{Hj}^- \Delta H_{t-j}^- + \\ \sum_{j=1}^{J} \beta_{Hj}^+ \Delta V_{t-j}^+ + \sum_{j=1}^{J} \beta_{Hj}^- \Delta V_{t-j}^- + \vartheta_{Ht}\end{aligned} \tag{17.6}$$

$$\begin{aligned}\Delta V_t = \theta_V + \delta_V^+ E_{t-1}^+ + \delta_V^- E_{t-1}^- + \sum_{j=1}^{J} \alpha_{Vj}^+ \Delta H_{t-j}^+ + \sum_{j=1}^{J} \alpha_{Vj}^- \Delta H_{t-j}^- + \\ \sum_{j=1}^{J} \beta_{Vj}^+ \Delta V_{t-j}^+ + \sum_{j=1}^{J} \beta_{Vj}^- \Delta V_{t-j}^- + \vartheta_{Vt}\end{aligned} \tag{17.7}$$

where ΔH and ΔV are the import prices of China and Vietnam in first difference; θ, δ, α, and β are coefficients; and ϑ is error terms. The subscripts of H and V differentiate the coefficients by country, t denotes time, and j represents lags. All the lagged price variables in first difference (i.e., ΔH_{t-j} and ΔV_{t-j}) are split into positive and negative components, as indicated by the superscripts + and $-$. For instance, ΔV_{t-1}^{+} is equal to $(V_{t-1} - V_{t-2})$ if $V_{t-1} > V_{t-2}$ and equal to 0 otherwise; ΔV_{t-1}^{-} is equal to $(V_{t-1} - V_{t-2})$ if $V_{t-1} < V_{t-2}$ and equal to 0 otherwise. The maximum lag of J is chosen with the AIC statistic and Ljung-Box Q test so the residuals have no serial correlation.

The error correction terms are the key component of the asymmetric error correction model. They are defined as $E_{t-1}^{+} = I_t \, \widehat{\xi}_{t-1}$ and $E_{t-1}^{-} = (1 - I_t) \, \widehat{\xi}_{t-1}$, and are a direct result from the above threshold cointegration regression. This definition of the error correction terms not only considers the possible asymmetric price responses to positive and negative shocks on the long-term equilibrium, but also incorporates the impact of threshold cointegration through the construction of Heaviside indicator in **Equations (17.4)** and **(17.5)**.

The signs of estimated coefficients can offer a first insight on the presence of asymmetric price behavior and can reveal the response of individual variables to the disequilibrium in the previous periods. Note the price of China is assumed to be the driving force and the long-term disequilibrium is measured as the price spread between Vietnam and China. Thus, the expected signs for the error correction terms should be positive for China (i.e., $\delta_H^{+} > 0, \ \delta_H^{-} > 0$) and negative for Vietnam (i.e., $\delta_V^{+} < 0, \ \delta_V^{-} < 0$).

Single or joint hypotheses can be formally assessed. In this study, four types of hypotheses and F-tests are examined, as detailed in Frey and Manera (2007). The first one is Granger causality test. Whether the Chinese price *Granger causes* its own price or the Vietnamese price can be tested by restricting all the Chinese prices to be zero ($H_{01} : \alpha_i^{+} = \alpha_i^{-} = 0$ for all lags i simultaneously). Similarly, the test can be applied to the Vietnamese price ($H_{02} : \beta_i^{+} = \beta_i^{-} = 0$ for all lags). The second type of hypothesis is concerned with the distributed lag asymmetric effect. At the first lag, for instance, the null hypothesis is that the Chinese price has symmetric effect on its own price or the Vietnamese price ($H_{03} : \alpha_1^{+} = \alpha_1^{-}$). This can be repeated for each lag and both countries (i.e., $H_{04} : \beta_4^{+} = \beta_4^{-}$). The third type of hypothesis is cumulative asymmetric effect. The null hypothesis of cumulative symmetric effect can be expressed as $H_{05} : \sum_{i=1}^{J} \alpha_i^{+} = \sum_{i=1}^{J} \alpha_i^{-}$ for China and $H_{06} : \sum_{i=1}^{J} \beta_i^{+} = \sum_{i=1}^{J} \beta_i^{-}$ for Vietnam. Finally, the equilibrium adjustment path asymmetry can be examined with the null hypothesis of $H_{07} : \delta^{+} = \delta^{-}$ for each equation estimated.

17.3 Needs for a new package

Estimating the statistical models as described in the previous section is almost impossible by clicking pull-down menus in a statistical software application. Computer programming must be employed, and within R's language structure, new functions must be created. As the number of functions is relatively large and some of them need to be repeatedly called, it is also more efficient to wrap up these new functions together in an R package. To reveal the need for new functions and packages, three particular aspects of the models employed in Sun (2011) are analyzed here.

The first challenge is to estimate the threshold cointegration model, as expressed in **Equations (17.3)** to **(17.5)** as a group. When the threshold value τ and a lag value P is given, variables in **Equation (17.3)** can be easily defined. Thus, the regression per se is a linear model and can be estimated by the `lm()` function. The problem is that the number of regressions is too large. Image that the total number of observations is 120 (e.g., monthly data for 10 years). If 70% of the residual values are used as the potential values for τ, then the number is about 84. Furthermore, assume the potential value of P can vary from 1 to 12. In combination, the number of regressions is $84 \times 12 \times 2 = 2,016$ for TAR and MTAR specification. At the end of each regression, the sum of squared errors and the threshold values should be documented. Note a data set with 120 observations is pretty small. If the data set is a little bit larger (e.g., 500 observations or more), then the task quickly becomes unmanageable or extremely inefficient.

The solution is using flow control statements, such as `if` and `for`. Multiple looping statements can be nested with each other, and outputs from each loop can be selected and collected. This has been presented in **Chapter 13** Flow Control Structure on page 269. Furthermore, as functions in R can divide a large programming job with interlinked components into small units, several new functions will be created in estimating the threshold cointegration model.

The second challenge is to estimate the asymmetric error correction model. Variables used in the regression needs to be created with a given value of lag J. The number of variables on the right side rises fast with a larger value for lag J. Furthermore, the value of J is unknown in advance so there is a need to estimate the model repeatedly with different values. Thus, the whole process includes selecting a lag value, composing variables, estimating the linear model, collecting regression outputs, and repeating it by J times. Therefore, while the asymmetric error correction model is linear, the process can be very tedious and inefficient without programming.

The third challenge is hypothesis tests on the coefficients from the asymmetric error correction model. Many hypotheses can be formed and F-tests can be employed. Individually, they are easier to implement; collectively, the work is inefficient without programming. This is because whenever the value of lag J changes, the number and positions of coefficients from the regression change too. Again, using new functions and flow control structure in R can easily solve these problems.

The linkage between new functions and a package is worthy of a note here. When the number of functions created in a project is large, the need and marginal benefit of building a new package can become significant. In fact, the threshold cointegration analysis serves as a good example. Walter Enders has made great contributions in this area through his book and journal articles (e.g., Enders, 2010; Enders and Siklos, 2001). He also programmed the main components of threshold cointegration analysis through the commercial software RATS and distributed it on the Internet. I have benefited from these sources in learning the method. However, RATS does not have the concept of package or library as R has clearly defined. As a result, the functions created in RATS have no good documentation and are pretty fragmented. In the following chapters, we will learn how to wrap up a group of new functions into the `apt` package. The step from many new functions to a new package will make programming efficient and pleasant to everyone, including the package author.

In summary, conducting an empirical study with a sophisticated statistical model like threshold cointegration has become almost impossible without programming. New functions can be created and called to address recurring regressions. When the number of new functions is large, a new package can be used to document the linkage about them clearly, organize the R program for a project logically, and eventually, improve research productivity.

17.4 Program version for Sun (2011)

When the program for an empirical study is very long (e.g., 30 pages), it may be better to organize it through several documents. The R program for Sun (2011) is five pages long only and it may not be necessary in this particular case. Nevertheless, to demonstrate the benefits of splitting a long program, two R programs are presented below. One is for the main statistical analyses and tables. The other is used to generate three figures.

17.4.1 Program for tables

The main program is listed in **Program 17.1** Main program version for generating tables in Sun (2011). This contains all the statistical analyses and can generate the four tables. Specifically, the data used in this study is pretty simple. It has four time series: import values and prices for China and Vietnam each from January 2002 to January 2010. They are saved as the data object of `daVich()` in the `apt` library. The main steps in the program correspond to the study design in the proposal and desired outputs in the manuscript. These include summary statistics (Table 1), Johansen cointegration tests (Table 2), threshold cointegration tests (Table 3), and asymmetric error correction model (Table 4).

As you read along the program, you will notice that a number of new functions have been created and wrapped together in the `apt` package. This is the focus of **Part V** Programming as a Contributor and will be elaborated gradually later on. At this point, it should be evident that the program version is well organized with the help of a new package. Except some minor format differences, the tables generated from this R program are highly similar to the final versions reported in Sun (2011).

Some results in Table 3 as published in Sun (2011) were inaccurate because of a mistake made when the data was processed in 2009. The mistake was identified after the paper was published. For example, for the consistent MTAR, the coefficient for the positive term was reported as −0.251 (−2.130) in Sun (2011), but it should be −0.106 (−0.764), as calculated from below codes. This is also explained on the help page of `daVich()`. The main conclusions from all the analyses are still qualitatively the same.

A large portion of **Program 17.1** has been distributed with the `apt` library as sample codes. A number of users worldwide have raised a similar question to me in recent several years. The question is simple from my perspective. However, as it occurs repeatedly from time to time, it is worthy of a note here. Briefly, the data used in Sun (2011) are just two single time series. It is tempting for another user to have two new data series imported into R, and then copy and run the sample program. Unfortunately, this will generate errors at various stages in the middle. This is because several key choices have to be made in **Program 17.1**, e.g., the lag and threshold values. The choices are dependent on individual data. Thus, one cannot just simply copy the whole R program for another data.

Program 17.1 Main program version for generating tables in Sun (2011)

```r
# Title: R Program for Sun (2011 FPE)
library(apt); library(vars); setwd('C:/aErer')
options(width = 100, stringsAsFactors = FALSE)

# -------------------------------------------------------------------------
# 1. Data and summary statistics
# Price data for China and Vietnam are saved as 'daVich'
data(daVich); head(daVich); tail(daVich); str(daVich)
prVi <- daVich[, 1]; prCh <- daVich[, 2]
(dog <- t(bsStat(y = daVich, digits = c(3, 3))))
dog2 <- data.frame(item = rownames(dog), CH.level = dog[, 2],
  CH.diff = '__', VI.level = dog[, 1], VI.diff = '__')[2:6, ]
rownames(dog2) <- 1:nrow(dog2); str(dog2); dog2

# -------------------------------------------------------------------------
# 2. Unit root test (Table 1)
ch.t1 <- ur.df(type = 'trend', lags = 3,  y = prCh); slotNames(ch.t1)
ch.d1 <- ur.df(type = 'drift', lags = 3,  y = prCh)
ch.t2 <- ur.df(type = 'trend', lags = 3,  y = diff(prCh))
ch.d2 <- ur.df(type = 'drift', lags = 3,  y = diff(prCh))
vi.t1 <- ur.df(type = 'trend', lags = 12, y = prVi)
vi.d1 <- ur.df(type = 'drift', lags = 11, y = prVi)
vi.t2 <- ur.df(type = 'trend', lags = 10, y = diff(prVi))
vi.d2 <- ur.df(type = 'drift', lags = 10, y = diff(prVi))
dog2[6, ] <- c('ADF with trend',
  paste(round(ch.t1@teststat[1], digits = 3), '[', 3,  ']', sep = ''),
  paste(round(ch.t2@teststat[1], digits = 3), '[', 3,  ']', sep = ''),
  paste(round(vi.t1@teststat[1], digits = 3), '[', 12, ']', sep = ''),
  paste(round(vi.t2@teststat[1], digits = 3), '[', 10, ']', sep = ''))
dog2[7, ] <- c('ADF with drift',
  paste(round(ch.d1@teststat[1], digits = 3), '[', 3,  ']', sep = ''),
  paste(round(ch.d2@teststat[1], digits = 3), '[', 3,  ']', sep = ''),
  paste(round(vi.d1@teststat[1], digits = 3), '[', 11, ']', sep = ''),
  paste(round(vi.d2@teststat[1], digits = 3), '[', 10, ']', sep = ''))
(table.1 <- dog2)

# -------------------------------------------------------------------------------
# 3. Johansen-Juselius and Engle-Granger cointegration analyses
# JJ cointegration
VARselect(daVich, lag.max = 12, type = 'const')
summary(VAR(daVich, type = 'const', p = 1))
K <- 5; two <- cbind(prVi, prCh)
summary(j1 <- ca.jo(x = two, type = 'eigen', ecdet = 'trend', K = K))
summary(j2 <- ca.jo(x = two, type = 'eigen', ecdet = 'const', K = K))
summary(j3 <- ca.jo(x = two, type = 'eigen', ecdet = 'none' , K = K))
summary(j4 <- ca.jo(x = two, type = 'trace', ecdet = 'trend', K = K))
summary(j5 <- ca.jo(x = two, type = 'trace', ecdet = 'const', K = K))
summary(j6 <- ca.jo(x = two, type = 'trace', ecdet = 'none' , K = K))
slotNames(j1)
out1 <- cbind('eigen', 'trend', K, round(j1@teststat, digits = 3), j1@cval)
out2 <- cbind('eigen', 'const', K, round(j2@teststat, digits = 3), j2@cval)
out3 <- cbind('eigen', 'none',  K, round(j3@teststat, digits = 3), j3@cval)
out4 <- cbind('trace', 'trend', K, round(j4@teststat, digits = 3), j4@cval)
out5 <- cbind('trace', 'const', K, round(j5@teststat, digits = 3), j5@cval)
out6 <- cbind('trace', 'none',  K, round(j6@teststat, digits = 3), j6@cval)
jjci <- rbind(out1, out2, out3, out4, out5, out6)
colnames(jjci) <- c('test 1', 'test 2', 'lag', 'statistic',
    'c.v 10%', 'c.v 5%', 'c.v 1%')
rownames(jjci) <- 1:nrow(jjci)
(table.2 <- data.frame(jjci))

# EG cointegration
LR <- lm(formula = prVi ~ prCh); summary(LR)
(LR.coef <- round(summary(LR)$coefficients, digits = 3))
(ry <- ts(data = residuals(LR), start = start(prCh), end = end(prCh),
  frequency = 12))
```

```
eg <- ur.df(y = ry, type = c('none'), lags = 1)
eg2 <- ur.df2(y = ry, type = c('none'), lags = 1)
(eg4 <- Box.test(eg@res, lag = 4, type = 'Ljung') )
(eg8 <- Box.test(eg@res, lag = 8, type = 'Ljung') )
(eg12 <- Box.test(eg@res, lag = 12, type = 'Ljung'))
EG.coef <- coefficients(eg@testreg)[1, 1]
EG.tval <- coefficients(eg@testreg)[1, 3]
(res.EG <- round(t(data.frame(EG.coef, EG.tval, eg2$aic, eg2$bic,
  eg4$p.value, eg8$p.value, eg12$p.value)), digits = 3))

# ---------------------------------------------------------------------------
# 4. Threshold cointegration
# best threshold
test <- ciTarFit(y = prVi, x = prCh); test; names(test)
t3 <- ciTarThd(y = prVi, x = prCh, model = 'tar', lag = 0); plot(t3)
time.org <- proc.time()
(th.tar <- t3$basic)
for (i in 1:12) {  # about 20 seconds
  t3a <- ciTarThd(y = prVi, x = prCh, model = 'tar', lag = i)
  th.tar[i+2] <- t3a$basic[, 2]
}
th.tar
time.org - proc.time()

t4 <- ciTarThd(y = prVi, x = prCh, model = 'mtar', lag = 0)
(th.mtar <- t4$basic); plot(t4)
for (i in 1:12) {  # about 36 seconds
  t4a <- ciTarThd(y = prVi, x = prCh, model = 'mtar', lag = i)
  th.mtar[i+2] <- t4a$basic[,2]
}
th.mtar

t.tar <- -8.041; t.mtar <- -0.451     # lag = 0 to 4; final choices
# t.tar <- -8.701 ; t.mtar <- -0.451  # lag = 5 to 12

mx <- 12  # lag selection
(g1 <-ciTarLag(y=prVi, x=prCh, model='tar',  maxlag = mx, thresh = 0))
(g2 <-ciTarLag(y=prVi, x=prCh, model='mtar', maxlag = mx, thresh = 0))
(g3 <-ciTarLag(y=prVi, x=prCh, model='tar',  maxlag = mx, thresh = t.tar))
(g4 <-ciTarLag(y=prVi, x=prCh, model='mtar', maxlag = mx, thresh = t.mtar))
plot(g1)

# Figure of threshold selection: mtar at lag = 3 (Figure 3 data)
(t5 <- ciTarThd(y=prVi, x=prCh, model = 'mtar', lag = 3, th.range = 0.15))
plot(t5)

# Table 3 Results of EG and threshold cointegration combined
vv <- 3
(f1 <- ciTarFit(y=prVi, x=prCh, model = 'tar',  lag = vv, thresh = 0))
```

```
(f2 <- ciTarFit(y=prVi, x=prCh, model = 'tar',  lag = vv, thresh = t.tar ))
(f3 <- ciTarFit(y=prVi, x=prCh, model = 'mtar', lag = vv, thresh = 0))
(f4 <- ciTarFit(y=prVi, x=prCh, model = 'mtar', lag = vv, thresh = t.mtar))

r0 <- cbind(summary(f1)$dia, summary(f2)$dia,
            summary(f3)$dia, summary(f4)$dia)
diag <- r0[c(1:4, 6:7, 12:14, 8, 9, 11), c(1, 2, 4, 6, 8)]
rownames(diag) <- 1:nrow(diag); diag

e1 <- summary(f1)$out; e2 <- summary(f2)$out
e3 <- summary(f3)$out; e4 <- summary(f4)$out; rbind(e1, e2, e3, e4)
ee <- list(e1, e2, e3, e4); vect <- NULL
for (i in 1:4) {
  ef <- data.frame(ee[i])
  vect2 <- c(paste(ef[3, 'estimate'], ef[3, 'sign'], sep = ''),
    paste('(', ef[3, 't.value'], ')', sep = ''),
    paste(ef[4, 'estimate'], ef[4, 'sign'], sep = ''),
    paste('(', ef[4, 't.value'], ')', sep = ''))
  vect <- cbind(vect, vect2)
}
item <- c('pos.coeff','pos.t.value', 'neg.coeff','neg.t.value')
ve <- data.frame(cbind(item, vect)); colnames(ve) <- colnames(diag)
(res.CI <- rbind(diag, ve)[c(1:2, 13:16, 3:12), ])
rownames(res.CI) <- 1:nrow(res.CI)
res.CI$Engle <- '__'
res.CI[c(3, 4, 9:13), 'Engle'] <- res.EG[, 1]
res.CI[4, 6] <- paste('(', res.CI[4, 6], ')', sep = '')
(table.3 <- res.CI[, c(1, 6, 2:5)])

# -------------------------------------------------------------------------
# 5. Asymmstric error correction model
(sem <- ecmSymFit(y = prVi, x = prCh, lag = 4)); names(sem)
(aem <- ecmAsyFit(y = prVi, x = prCh, lag = 4, model = 'mtar',
   split = TRUE, thresh = t.mtar))
(ccc <- summary(aem))
coe <- cbind(as.character(ccc[1:19, 2]),
  paste(ccc[1:19, 'estimate'], ccc$signif[1:19], sep = ''),
  ccc[1:19, 't.value'],
  paste(ccc[20:38, 'estimate'], ccc$signif[20:38],sep = ''),
  ccc[20:38, 't.value'])
colnames(coe) <- c('item', 'CH.est', 'CH.t', 'VI.est','VI.t')

(edia <- ecmDiag(aem, 3)); (ed <- edia[c(1, 6:9), ])
ed2 <- cbind(ed[, 1:2], '_', ed[, 3], '_'); colnames(ed2) <- colnames(coe)
(tes <- ecmAsyTest(aem)$out); (tes2 <- tes[c(2, 3, 5, 11:13, 1), -1])
tes3 <- cbind(as.character(tes2[, 1]),
  paste(tes2[, 2], tes2[, 6], sep = ''),
  paste('[', round(tes2[, 4], digits = 2), ']', sep = ''),
  paste(tes2[, 3], tes2[, 7], sep = ''),
  paste('[', round(tes2[, 5], digits = 2), ']', sep = ''))
colnames(tes3) <- colnames(coe)
(table.4 <- data.frame(rbind(coe, ed2, tes3)))

# ------------------------------------------------------------------------
# 6. Output
(output <- listn(table.1, table.2, table.3, table.4))
write.list(z = output, file = 'OutBedTable.csv')
```

Note: Major functions used in **Program 17.1** are: `ur.df()`, `ca.jo()`, `VAR()`, `ciTarThd()`, `ciTarLag()`, `ciTarFit()`, `ecmSymFit()`, `ecmAsyFit()`, `ecmDiag()`, `bsStat()`, `Box.test()`, and `lm()`.

```
# Selected results from Program 17.1
> table.1
            item  CH.level   CH.diff   VI.level    VI.diff
1           mean   148.791        __    115.526         __
2           stde    11.461        __      9.882         __
3           mini   119.618        __     99.335         __
4           maxi   177.675        __    150.721         __
5           obno        97        __         97         __
6 ADF with trend -2.956[3] -7.394[3] -2.936[12] -5.777[10]
7 ADF with drift -2.422[3] -7.195[3] -1.161[11]  -5.74[10]

> table.2
   test.1 test.2 lag statistic c.v.10. c.v.5. c.v.1.
1   eigen  trend   5    10.001   10.49  12.25  16.26
2   eigen  trend   5    20.253   16.85  18.96  23.65
3   eigen  const   5     4.461    7.52   9.24  12.97
4   eigen  const   5    14.304   13.75  15.67   20.2
5   eigen   none   5     4.438     6.5   8.18  11.65
6   eigen   none   5      14.3   12.91   14.9  19.19
7   trace  trend   5    10.001   10.49  12.25  16.26
8   trace  trend   5    30.254   22.76  25.32  30.45
9   trace  const   5     4.461    7.52   9.24  12.97
10  trace  const   5    18.765   17.85  19.96   24.6
11  trace   none   5     4.438     6.5   8.18  11.65
12  trace   none   5    18.738   15.66  17.95  23.52

> table.3
          item    Engle       tar     c.tar      mtar    c.mtar
1          lag       __         3         3         3         3
2       thresh       __         0    -8.041         0    -0.451
3    pos.coeff   -0.407  -0.328**   -0.28**    -0.116    -0.106
4  pos.t.value (-4.173)  (-2.523)  (-2.306)  (-0.824)  (-0.764)
5    neg.coeff       __ -0.515*** -0.721*** -0.658*** -0.677***
6  neg.t.value       __  (-3.119)  (-3.942)  (-4.754)  (-4.888)
7    total obs       __        97        97        97        97
8    coint obs       __        93        93        93        93
9          aic  669.627   658.998   654.863   650.612   649.495
```

```
10         bic  677.351   674.193   670.059   665.808    664.69
11  LB test(4)    0.773     0.961     0.879     0.988     0.987
12  LB test(8)    0.919     0.992     0.964     0.999     0.998
13 LB test(12)    0.239     0.122     0.084     0.289     0.333
14   H1: no CI       __     6.539     8.836    11.307    11.976
15  H2: no APT       __     1.033     5.081     9.435    10.612
16 H2: p.value       __     0.312     0.027     0.003     0.002

> table.4[1:7, ]
                  item    CH.est   CH.t  VI.est   VI.t
1          (Intercept)   -0.146  -0.052 -3.853* -1.777
2 X.diff.prCh.t_1.pos -0.622*** -2.755 -0.155  -0.897
3 X.diff.prCh.t_2.pos    0.082   0.344 -0.144  -0.795
4 X.diff.prCh.t_3.pos   -0.282  -1.264  0.146   0.854
5 X.diff.prCh.t_4.pos   -0.324  -1.403 -0.193  -1.091
6 X.diff.prCh.t_1.neg   -0.314. -1.464 -0.105  -0.641
7 X.diff.prCh.t_2.neg -0.584*** -2.651  0.085   0.508
```

17.4.2 Program for figures

The three figures reported in Sun (2011) can be created by base R graphics or the `ggplot2` package. These codes for graphs are organized separately as a document to increase readability, and they are all presented the following R program. When the codes for figure generation are long, this can make the main program more concise.

There are several ways to connect individual programs for a specific empirical study. First, the main program can be called by the `source()` function and all data will become available for another program. Alternatively, if it takes a long time to run the main program each time or the data used in another program is small, then the relevant data can be copied or generated directly. This is exactly true for the relation between the two programs here. In general, figures use fewer data than statistical analyses. A threshold cointegration analysis often takes quite some time to finish. Thus, at the beginning of **Program 17.2**, the value data for Figure 1, price data for Figure 2, and sum of squared errors for Figure 3 are generated directly, without calling the main program.

Figure 17.1 is generated from traditional graphics system, and **Figure 17.2** is from `ggplot2`. The main difference is that the ggplot version has a gray background and grid lines. Which version is more attractive is largely a personal choice. The codes used for the ggplot version is generally longer than these for the base R version. One can also customize the appearance of the ggplot version and make it very similar to the version from base R. This is left as **Exercise 17.6.1** on page 411.

In Sun (2011), Figure 1 is monthly import values for China and Vietnam, and Figure 2 is their monthly import prices. Both the figures can be created with the `ggplot2` package. Recall that `%+%` is defined in `ggplot2` to replace one data frame with another one. It is tempting to use this operator to generate Figure 2 with a substitution of the underlying data frame. However, the value and price data are quite different in scale. As a result, it is faster in this case to copy all the codes for Figure 1 and then revise them for Figure 2.

Program 17.2 Graph program version for generating figures in Sun (2011)

```
# Title: Graph codes for Sun (2011 FPE)
library(apt); library(ggplot2); setwd('C:/aErer'); data(daVich)
```

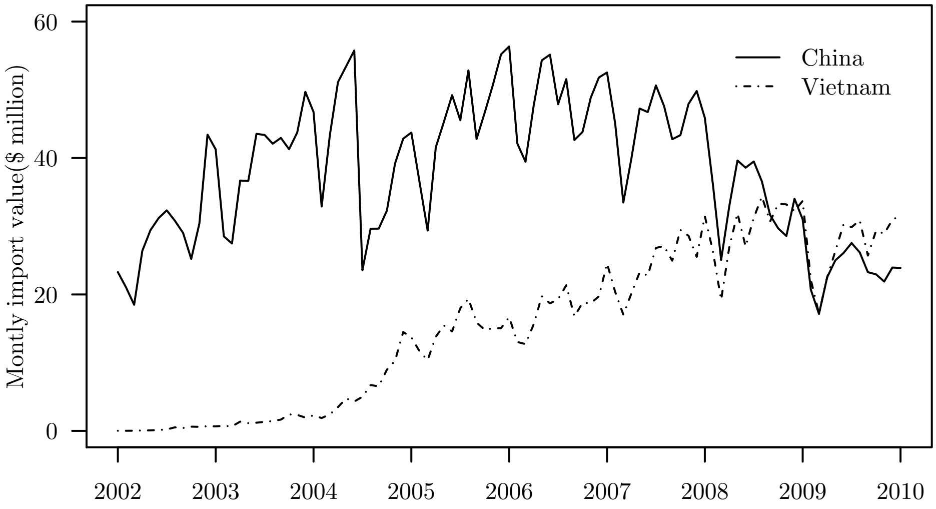

Figure 17.1 Monthly import value of beds from China and Vietnam (base R)

```

# -----------------------------------------------------------------------
# A. Data for graphs: value, price, and t5$path
prVi <- daVich[, 1]; prCh <- daVich[, 2]
vaVi <- daVich[, 3]; vaCh <- daVich[, 4]
(date <- as.Date(time(daVich), format = '%Y/%m/%d'))
(value <- data.frame(date, vaCh, vaVi))
(price <- data.frame(date, prVi, prCh))
(t5 <- ciTarThd(y=prVi, x=prCh, model = 'mtar', lag = 3, th.range = 0.15))

# -----------------------------------------------------------------------
# B. Traditonal graphics
# Figure 1 Import values from China and Vietnam
win.graph(width = 5, height = 2.8, pointsize = 9); bringToTop(stay = TRUE)
par(mai = c(0.4, 0.5, 0.1, 0.1), mgp = c(2, 1, 0), family = "serif")
plot(x = vaCh, lty = 1, lwd = 1, ylim = c(0, 60), xlab = '',
  ylab = 'Montly import value($ million)', axes = FALSE)
box(); axis(side = 1, at = 2002:2010)
axis(side = 2, at = c(0, 20, 40, 60), las = 1)
lines(x = vaVi, lty = 4, lwd = 1)
legend(x = 2008.1, y = 59, legend = c('China', 'Vietnam'),
  lty = c(1, 4), box.lty = 0)
fig1.base <- recordPlot()

# Figure 2 Import prices from China and Vietnam
win.graph(width = 5, height = 2.8, pointsize = 9)
par(mai = c(0.4, 0.5, 0.1, 0.1), mgp = c(2, 1, 0), family = "serif")
plot(x = prCh, lty = 1, type = 'l', lwd = 1, ylim = range(prCh, prVi),
  xlab = '', ylab = 'Monthly import price ($/piece)' )
```

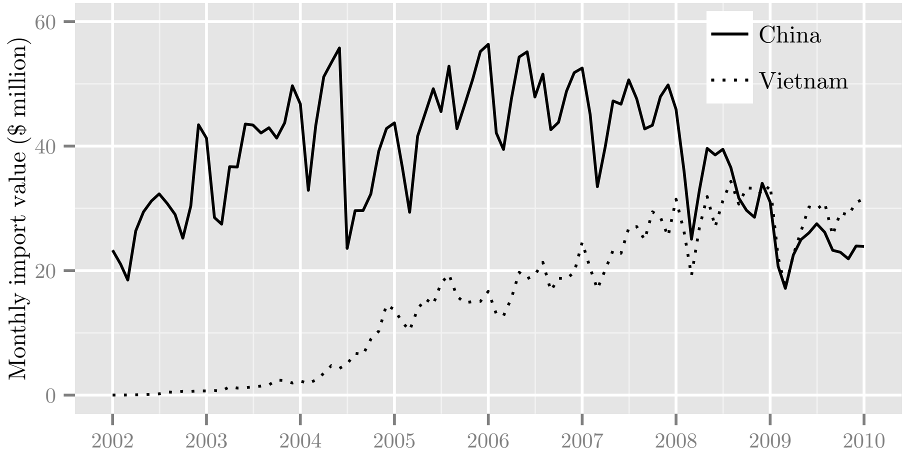

Figure 17.2 Monthly import value of beds from China and Vietnam (ggplot2)

```
lines(x = prVi, lty = 3, type = 'l', lwd = 1)
legend(x = 2008.5, y = 175, legend = c('China', 'Vietnam'),
  lty = c(1, 3), box.lty = 0)

# Figure 3 Sum of dquared errors by threshold value from MTAR
win.graph(width = 5.1, height = 3.3, pointsize = 9)
par(mai = c(0.5, 0.5, 0.1, 0.1),  mgp = c(2.2, 1, 0), family = "serif")
plot(formula = path.sse ~ path.thr, data = t5$path, type = 'l',
     ylab = 'Sum of Squared Errors', xlab = 'Threshold value')

# -------------------------------------------------------------------------
# C. ggplot for three figures
pp <- theme(axis.text   = element_text(size = 8, family = "serif")) +
      theme(axis.title  = element_text(size = 9, family = "serif")) +
      theme(legend.text = element_text(size = 9, family = "serif")) +
      theme(legend.position = c(0.85, 0.9) ) +
      theme(legend.key = element_rect(fill = 'white', color = NA)) +
      theme(legend.background = element_rect(fill = NA, color = NA))

fig1 <- ggplot(data = value, aes(x = date)) +
  geom_line(aes(y = vaCh, linetype = 'China')) +
  geom_line(aes(y = vaVi, linetype = 'Vietnam')) +
  scale_linetype_manual(name = '', values = c(1, 3)) +
  scale_x_date(name = '', labels = as.character(2002:2010), breaks =
    as.Date(paste(2002:2010, '-1-1', sep = ''), format = '%Y-%m-%d')) +
  scale_y_continuous(limits = c(0, 60),
    name = 'Monthly import value ($ million)') + pp

fig2 <- ggplot(data = price, aes(x = date)) +
  geom_line(aes(y = prCh, linetype = 'China')) +
  geom_line(aes(y = prVi, linetype = 'Vietnam')) +
  scale_linetype_manual(name = '', values = c(1, 3))+
  scale_x_date(name = '', labels = as.character(2002:2010), breaks =
    as.Date(paste(2002:2010, '-1-1', sep = ''), format = '%Y-%m-%d')) +
  scale_y_continuous(limits = c(98, 180),
    name = 'Monthly import price ($/piece)') + pp

fig3 <- ggplot(data = t5$path) +
  geom_line(aes(x = path.thr, y = path.sse)) +
  labs(x = 'Threshold value', y = 'Sum of squared errors') +
  scale_y_continuous(limits = c(5000, 5700)) +
  scale_x_continuous(breaks = c(-10:7)) +
  theme(axis.text  = element_text(size = 8, family = "serif")) +
  theme(axis.title = element_text(size = 9, family = "serif"))

# ----------------------------------------------------------------------
# D. Show on screen devices or save on file devices
pdf(file = 'OutBedFig1base.pdf', width = 5, height = 2.8, pointsize = 9)
replayPlot(fig1.base); dev.off()

windows(width = 5, height = 2.8); fig1
windows(width = 5, height = 2.8); fig2
windows(width = 5, height = 2.8); fig3

ggsave(fig1, filename = 'OutBedFig1ggplot.pdf', width = 5, height = 2.8)
ggsave(fig2, filename = 'OutBedFig2ggplot.pdf', width = 5, height = 2.8)
ggsave(fig3, filename = 'OutBedFig3ggplot.pdf', width = 5, height = 2.8)
```

17.5 Road map: developing a package and GUI (Part V)

Two large parts for R programming have been presented so far in this book. In **Part III** Programming as a Beginner, basic R concepts and data manipulations are elaborated. Using predefined functions for specific analyses is emphasized. In **Part IV** Programming as a Wrapper, the structure of an R function is examined and how to write new functions is demonstrated through various applications. Assuming you have learned these techniques well, we now reach the final stage of the growing-up process: creating a new package for a statistical model or research issue.

In general, the materials in the part for beginner are more difficult than these in the part for wrapper. The current materials in **Part V** Programming as a Contributor are probably the easiest. The main challenge for creating a new package is to design the structure and put appropriate contents inside the folders. This is covered in **Chapter 18** Contents of a New Package. Once the contents for a new package are finalized, the procedure of building up the package is straightforward, and it takes no more than a few days to learn it. This is covered in **Chapter 19** Procedures for a New Package.

It is possible to transform an R package into a graphical user interface (GUI). The decision of building a graphical user interface is related to the associated benefit and cost. The benefit of GUIs includes a more intuitive appearance and low requirement on user's programming skills. The cost of this extra step is that package authors will need to learn new commands to develop an application with a clear interface. If R is selected as the language in developing a GUI, then a programmer should have a solid understanding of R.

The basics of developing graphical user interfaces in R are presented in **Chapter 20** Graphical User Interfaces. With a good knowledge base, one just needs to learn a few new concepts related to GUIs and a few more packages in R. In the `apt` package, its core functions are programmed into a GUI. This demonstrates well the growing process with R from preparing individual functions, to a new package, and finally to a graphical user interface.

17.6 Exercises

17.6.1 *Customize Figure 1 in Sun (2011) by `ggplot2`.* In **Program 17.2** Graph program version for generating figures in Sun (2011) on page 407, Figure 1 is generated by base R graphics and `ggplot2` separately. Customize the version by `ggplot2` so its appearance looks like the version from base R graphics. This may like a trivial exercise, but it will let you learn more about `ggplot2`.

17.6.2 *Analyze two empirical studies for a similar issue.* The purpose of this exercise is to learn and compare design techniques for several studies in the same area, similar to the relation between Wan et al. (2010a) and Sun (2011). Recall that in **Exercise 3.6.2** on page 41, one empirical study has been selected. This selected study can be one of the sample studies (i.e., Sun, 2006a,b; Sun and Liao, 2011), or one from the literature. For this exercise, find another empirical study in the literature that is closely related to the research issue covered in the selected study. Read and compare the objectives, methods, and other aspects of the two related studies, with an emphasis on the linkage.