ECONOMETRIC METHODS II
TA session 2
Estimating VAR(p) processes

Fernando Pérez Forero*

May 3rd, 2012

1 Introduction

In this second session we will cover the estimation of Vector Autoregressive (VAR) processes of order p, Impulse Response Functions (IRF) and the Forecast Error Variance Decomposition (FEVD).

Most of the material covered in this session can be found in Lütkepohl (2005) (L) and Hamilton (1994) (H). In order to provide a better understanding of each step, I will indicate the number of equation of each reference as follows: (Book, equation).

2 Preliminaries

2.1 OLS estimation of a VAR(p) process

Consider a VAR(p) process (L,3.1.1),

$$y_t = v + A_1 y_{t-1} + \cdots + A_p y_{t-p} + u_t, \qquad t = 1, \ldots, T \tag{2.1}$$

where y_t, u_t and v are $(K \times 1)$ vectors and A_i are $(K \times K)$ matrices for each $i = 1, \ldots, p$. In addition, the error term u_t is a white noise random vector such that $E(u_t) = 0$, $E(u_t u_t') = \Sigma_u$ and $E(u_t u_s') = 0$ for $s \neq t$, where Σ_u is a $(K \times K)$ positive definite matrix.

*PhD student. Department of Economics, Universitat Pompeu Fabra, Ramon Trias Fargas 25–27, 08005 Barcelona, Spain (email: fernandojose.perez@upf.edu)

It is useful to define the following terms (L,3.2.1)

$$\begin{aligned} Y &= [y_1, y_2, \dots, y_T]_{(K\times T)}\,,\ B = [v, A_1, \dots, A_p]_{(K\times(Kp+1))} \\ Z_t &= [1, y_{t-1}, \dots, y_{t-p}]'_{((Kp+1)\times 1)} \\ Z &= [Z_0, Z_1, \dots, Z_{T-1}]_{((Kp+1)\times T)} \\ U &= [u_1, u_2, \dots, u_T]_{(K\times T)} \end{aligned} \tag{2.2}$$

As a result, equation (2.1) can be re-written as (L,3.2.2)

$$Y = BZ + U \tag{2.3}$$

The OLS estimator is therefore defined as (L,3.2.10)

$$\widehat{B} = YZ'\left(ZZ'\right)^{-1} \tag{2.4}$$

Moreover, the covariance matrix estimator is (L,3.2.18)

$$\begin{aligned} \widetilde{\Sigma}_u &= \frac{1}{T}\widehat{U}\widehat{U}' = \frac{1}{T}\left(Y - \widehat{B}Z\right)\left(Y - \widehat{B}Z\right)' \\ &= \frac{1}{T}Y\left(I_T - Z'\left(ZZ'\right)^{-1}Z\right)Y' \end{aligned} \tag{2.5}$$

2.2 Lag order selection

Following the slides from last class, we will also compute different lag order selection criteria. We briefly report them here:

1. Akaike's Final Prediction Error (L,4.3.1)

$$FPE\left(m\right) = \left[\frac{T + Km + 1}{T - Km - 1}\right]^K \det\widetilde{\Sigma}_u\left(m\right), \quad m \leq p \tag{2.6}$$

2. Akaike's Information Criterion (L,4.3.2)

$$AIC\left(m\right) = \ln\left|\widetilde{\Sigma}_u\left(m\right)\right| + \frac{2mK^2}{T}, \quad m \leq p \tag{2.7}$$

3. Hannan-Quinn criterion (L,4.3.8)

$$HQ\left(m\right) = \ln\left|\widetilde{\Sigma}_u\left(m\right)\right| + \frac{2\ln\ln T}{T}, \quad m \leq p \tag{2.8}$$

4. Schwarz criterion (L,4.3.9)

$$SIC(m) = \ln\left|\widetilde{\Sigma}_u(m)\right| + \frac{\ln T}{T} mK^2, \quad m \leq p \tag{2.9}$$

5. Log-Likelihood ratio test

$$\lambda_{LR} = T\left[\log\left|\widehat{\Omega}_{m-1}\right| - \log\left|\widehat{\Omega}_m\right|\right] \sim \chi^2_{(K^2)}, \quad m \leq p \tag{2.10}$$

3 Estimating the model

3.1 Data description

Data is taken from the Central Bank of Peru: http://www.bcrp.gob.pe. I've created the workspace Data_Peru_raw.mat that contains the data, dates and labels. The workspace must be saved in the same directory of the current m-file. In addition, the time series used (in monthly frequency) are:

1. CPI: Consumer Price index of Lima (2009=100).
2. GDP: GDP index (1994=100).
3. INT: Interbank Interest Rate in domestic currency (annual terms).
4. ER : Nominal Exchange Rate (S/. per US$).

We start the program loading the data:

```
load Data_Peru_raw
Y=Data_Peru_raw;
```

Then, we set the sample 2002:02-2011:02 creating the scalars `ini` and `fin`:

```
% Data availability:  1995.M10-2011.M2
% Initial date
% ------------
 yy=2002; %year
 mm=2; %month
 ini=yy+round(1e4*mm/12)/1e4;

% Final date
% ------------
 yy=2011; %year
```

```
mm=2; %month
fin=yy+round(1e4*mm/12)/1e4;
```

Up to this point, it is important to define the ordering in the VAR system, so that:

```
var_ord=[2 1 3 4];
```

and we define the vector Y_t, number of observations and variables:

```
Y=Y(var_ord,find(Dates==ini):find(Dates==fin));
Dates=Dates(:,find(Dates==ini):find(Dates==fin));
Names=Names(:,var_ord);
[K,T]=size(Y);
```

Now, let's take a look to the raw data in Figure 3.1:

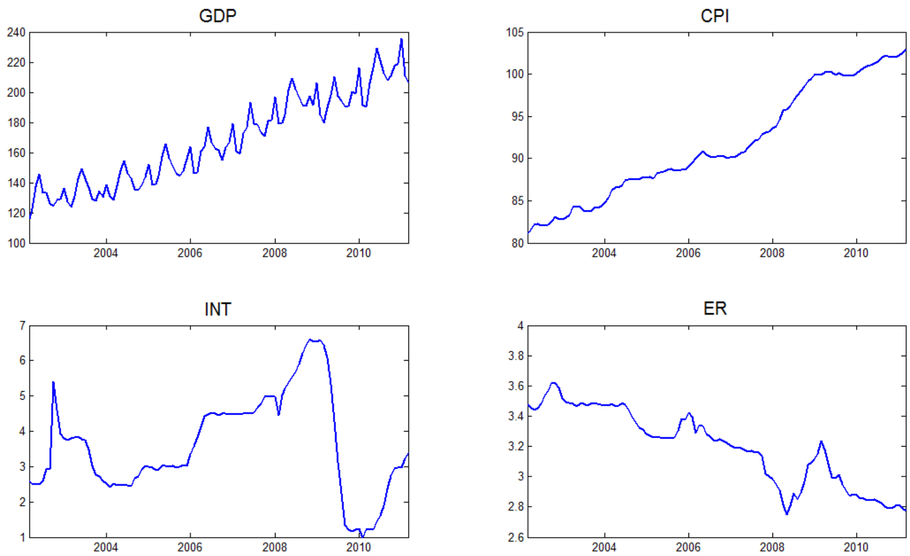

Figure 3.1: Raw Data

It can be inferred from last figure that GDP and CPI series exhibit a deterministic trend. Besides, GDP series exhibit a marked seasonal pattern. On the other hand, INT series is less volatile for the period after 2002 (after its adoption as core monetary policy instrument in the Inflation Targeting framework) and finally, ER series exhibit a marked persistence in different episodes along the sample. As a result, it is likely that an estimated VAR using this raw data will lead us a to a non-stationary companion form. We want the model to be stationary, therefore we make proceed to detrend the raw data as follows:

1. GDP: $\widetilde{y}_t = 12 * (1 - L) (1 - L^{12}) \log (y_t)$

2. CPI: $\widetilde{y}_t = 12 * (1 - L) \log (y_t)$

3. INT: $\widetilde{y}_t = 0.01 * y_t$

4. ER: $\widetilde{y}_t = 12 * (1 - L) \log (y_t)$

Some aspects are worth to remark. First, since the interbank interest rate is expressed in annual terms, we express the remaining variables in annual terms as well, i.e. $(1 - L) \log (y_t)$ is the monthly growth rate. We also remove the seasonal component of GDP applying the operator $(1 - L^{12})$. Finally, we leave INT unchanged as it is fairly standard in the VAR literature. In this regard, the function `data_transf.m` produces the latter transformation. However, when performing these transformations we lose some observations. As a result, in order to get a balanced dataset we have to drop the same number of observations for every variable. That is why the referred function produces also the number of observations that need to be eliminated (see details in appendix A). Detrended data is depicted in Figure 3.2

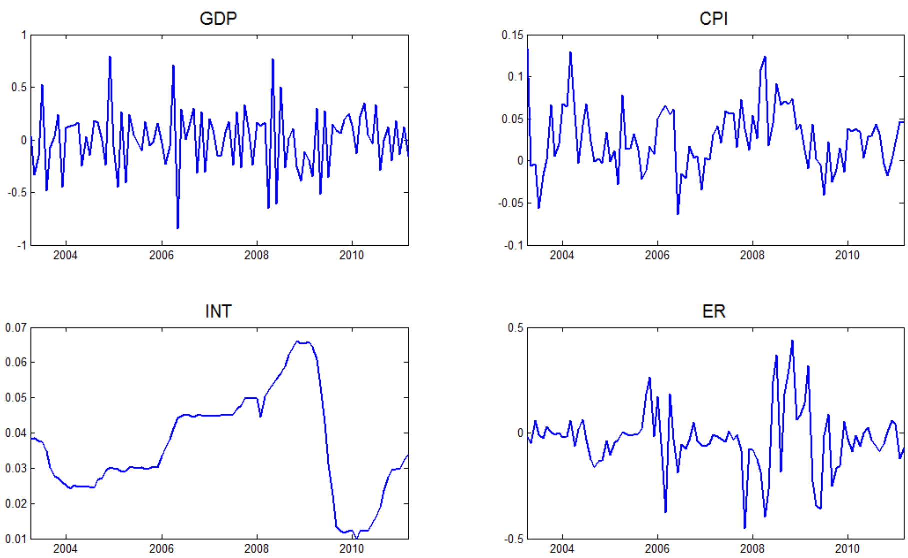

Figure 3.2: Detrended Data

3.2 Lag order selection

Given the data set depicted in Figure 3.2, we estimate equation (2.3) by OLS using a different lag length up to a limit $\overline{p} = 10$. Then we compute the five criteria described in equations $(2.6) - (3.4)$.

```
% Select a maximum lag
```

```
M=10;
Sigma_u_tilde=zeros(K,K,M);
IC=zeros(4,M);
DETSIGVEC=zeros(1,M);
for p=1:M
 Z=ones(1,T-p);
 for i=1:p
 Z=[Z;Y(:,p+1-i:T-i)];
 end
 Sigma_u_tilde(:,:,p)=(1/T)*Y(:,p+1:T)*(eye(T-p)-Z'*inv(Z*Z')*Z)*Y(:,p+1:T)';
% See eq (3.2.18)
 IC(1,p)=(((T+K*p+1)/(T-K*p-1))^K)*det(Sigma_u_tilde(:,:,p)); % See eq (4.3.1)
 IC(2,p)=log(det(Sigma_u_tilde(:,:,p)))+2*p*(K^2)/T; % See eq (4.3.2)
 IC(3,p)=log(det(Sigma_u_tilde(:,:,p)))+2*log(log(T))*p*(K^2)/T; % See eq
(4.3.9)
 IC(4,p)=log(det(Sigma_u_tilde(:,:,p)))+log(T)*p*(K^2)/T; % See eq (4.3.9)
 DETSIGVEC(p)=log(det(Sigma_u_tilde(:,:,p)));
 clear Z
end
```

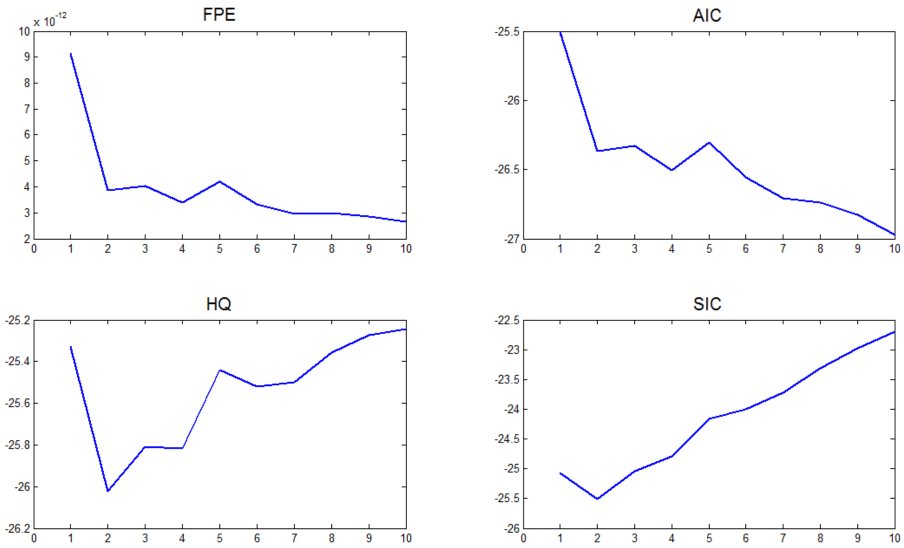

Figure 3.3: Lag order selection criteria

```
DETSIGDIFF=zeros(1,M);
```

```
DETSIGDIFF(1)=NaN;
for p=2:M
 DETSIGDIFF(p)=T*(DETSIGVEC(p-1)-DETSIGVEC(p));
end
crit=chi2inv(0.95,K^2)*ones(1,M-1);
crit=[NaN crit];
pt=min(find(DETSIGDIFF-crit<0))-1;
```

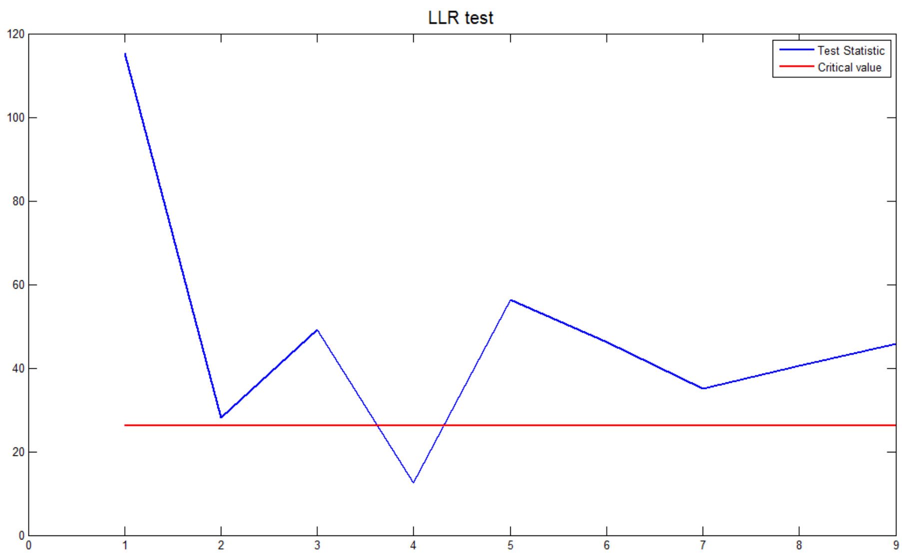

Figure 3.4: Log-Likelihood ratio test

After inspecting each of these criteria we select the optimal lag length for each of them:

```
% Select optimal lag-length
pt=min(find(DETSIGDIFF-crit<0))-1;
[IC_min p_opt]=min(IC,[],2);
p_opt=p_opt'-ones(1,4);
p_opt=[p_opt,pt];
```

where the vector `p_opt` contains this information.

4 Model Estimation

4.1 OLS-Estimation

Using the optimal lag length from `p_opt` and a particular criteria, we proceed to estimate equation (2.3). Without loss of generality, we select the Schwarz information criteria. OLS estimation is performed as follows:

```
   Z=[ones(1,T-p)];
   for i=1:p
    Z=[Z;Y(:,p+1-i:T-i)];
   end
   B_OLS=Y(:,p+1:T)*Z'*inv(Z*Z'); % See eq (3.2.10)
   Sigma_U=(1/(T-K*p-1))*(Y(:,p+1:T)*Y(:,p+1:T)'-Y(:,p+1:T)*Z'*inv(Z*Z')*Z*Y(:,p+1:T)
% See eq (3.2.23)
   U_resid=Y(:,p+1:T)-B_OLS*Z;
   v=B_OLS(:,1);
   A=zeros(K,K,p);
   for l=1:p
    A(:,:,l)=B_OLS(:,2+K*(l-1):1+K*l);
   end
   A_coeff=[];
   for l=1:p
    A_coeff=[A_coeff,A(:,:,l)];
   end
```

The next step is to compute the companion form (see first handout) and proceed to estimate the impulse responses.

4.2 Impulse Response Function (IRF)

Once we have generated the companion form components associated with (2.1), computing Impulse Responses is extremely simple. First, we set a finite horizon h and we generate matrices P and $\mathbf{J}$.

```
   h=24; % Horizon
   P=chol(Sigma_U); % Cholesky decomposition
   P=P'; % Following the notation of Lutkepohl - Section 3.7 :  U=P*E
   capJ=zeros(K,K*p);
   capJ(1:K,1:K)=eye(K);
```

Then we construct matrices $\Theta_{i(K\times K)}$

```
Iresp=zeros(K,K,h);
Iresp(:,:,1)=P;
for j=1:h-1
 temp=(F^j);
 Iresp(:,:,j+1)=capJ*temp*capJ'*P;
end
clear temp
```

Finally, we plot the Impulse Responses in Figure 4.1:

```
FS=15;
LW=2;
figure(5)
set(0,'DefaultAxesColorOrder',[0 0 1],...
 'DefaultAxesLineStyleOrder','-|-|-')
set(gcf,'Color',[1 1 1])
set(gcf,'defaultaxesfontsize',FS-5)
for i=1:K
 for j=1:K
 subplot(K,K,(j-1)*K+i)
 plot(squeeze(Iresp(i,j,:)),'Linewidth',LW)
 hold on
 plot(zeros(1,h),'k')
 title(sprintf('%s after %s',char(Names(i)),char(Names(j))),'FontSize',FS-5)
 xlim([1 h])
 set(gca,'XTick',0:ceil(h/4):h)
 end
end
```

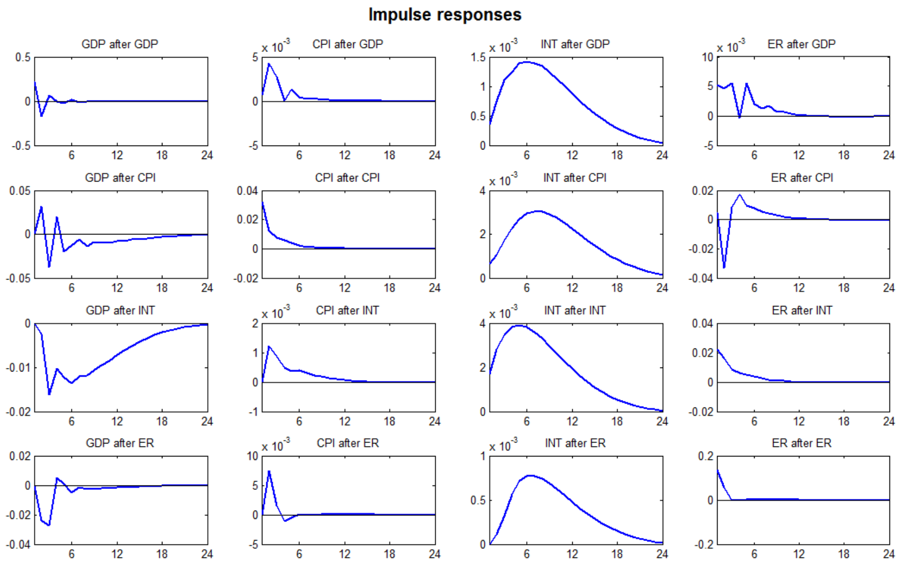

Figure 4.1: Impulse responses

Up to this point there is no economic interpretation for these results since we have not put emphasis on identification restrictions.

4.3 Forecast Error Variance Decomposition (FEVD)

Once we have computed the Impulse Responses, we can now proceed to compute the contribution of each variable to the Forecast Error variance $j = 1, \ldots, h$ periods ahead. We first compute the numerator and denominator separately, i.e. the absolute contribution and the Mean-Squared-Error (MSE).

```
MSE=zeros(K,h);
CONTR=zeros(K,K,h);
for j=1:h
 % Compute MSE
 temp2=eye(K);
 for i=1:K
 if j==1
MSE(i,j)=temp2(:,i)'*Iresp(:,:,j)*Iresp(:,:,j)'*temp2(:,i);
 for k=1:K
 CONTR(i,k,j)=(temp2(:,i)'*Iresp(:,:,j)*temp2(:,k))^2;
 end
 else
 MSE(i,j)=MSE(i,j-1)+temp2(:,i)'*Iresp(:,:,j)*Iresp(:,:,j)'*temp2(:,i);
```

```
 for k=1:K
 CONTR(i,k,j)=CONTR(i,k,j-1)+(temp2(:,i)'*Iresp(:,:,j)*temp2(:,k))^2; %
(L,2.3.36)
 end
 end
 end
end
```

Then we compute the contributions and plot the results in Figure 4.2:

```
VD=zeros(K,h,K);
for k=1:K
 VD(:,:,k)=squeeze(CONTR(:,k,:))./MSE; % (L,2.3.37)
end
clear temp2 CONTR MSE
figure(6)
set(0,'DefaultAxesColorOrder',[0 0 1],...
 'DefaultAxesLineStyleOrder','-|-|-')
set(gcf,'Color',[1 1 1])
set(gcf,'defaultaxesfontsize',FS-5)
for i=1:K
 for j=1:K
 subplot(K,K,(j-1)*K+i)
 plot(squeeze(VD(i,:,j)),'Linewidth',LW)
 hold on
 plot(zeros(1,h),'k')
 title(sprintf('%s to Var(e_{%s,t+h}).',char(Names(j)),char(Names(i))))
 xlim([1 h])
 set(gca,'XTick',0:ceil(h/4):h)
 ylim([0 1])
 set(gca,'YTick',0:0.25:1)
 end
end
```

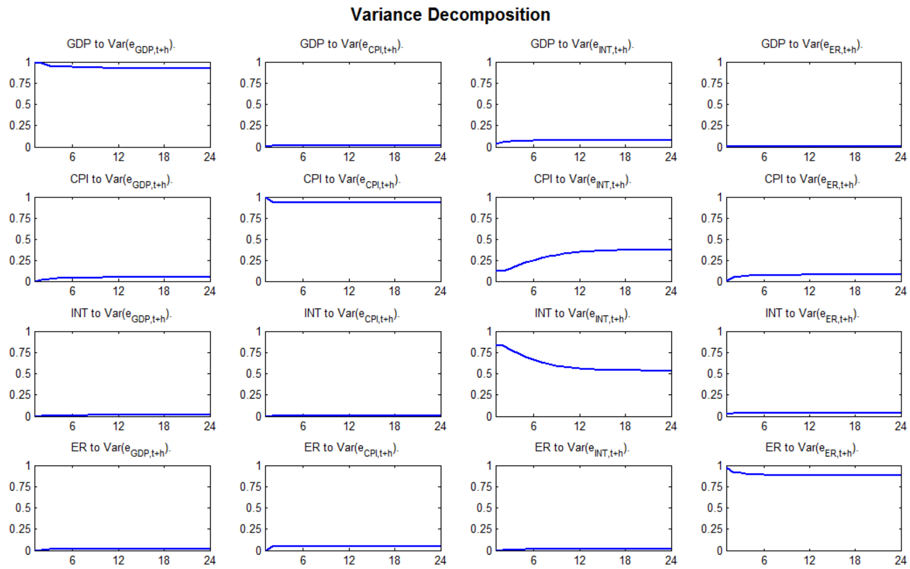

Figure 4.2: Forecast Error Variance decomposition

A Data transformation

```
function [y,n_el]=data_transf(x,a)
   % Data transformation
   % Fernando Pérez Forero
   % fernandojose.perez@upf.edu
   %
   % This file calls the functions hpfilter.m and penta2.m.
   % We acknowledge the author (Kurt Annen) for these files.
   % Categories:
   % ------------------------------------------------------------------------
   % 1.  Levels (Y_{t})
   % 2.  First Difference (Y_{t}-Y_{t-1})
   % 3.  Log-levels (Ln(Y_{t}))
   % 4.  Log-First Difference (Ln(Y_{t})-Ln(Y_{t-1}))
   % 5.  Seasonal difference in quarterly data (Ln(Y_{t})-Ln(Y_{t-4}))
   % 6.  Seasonal difference in monthly data (Ln(Y_{t})-Ln(Y_{t-12}))
   % 7.  Detrending Ln(Y_{t}) by HP filter using quarterly data
   % 8.  Detrending Ln(Y_{t}) by HP filter using monthly data
   % 9.  Seasonal difference of a detrended Ln(Y_{t}) by HP filter using quarterly
data
   % 10.  Seasonal difference of a detrended Ln(Y_{t}) by HP filter using monthly
data
   % 11.  Seasonal difference of a detrended Ln(Y_{t}) by removing a linear
trend
   % 12.  Detrended Ln(Y_{t}) by removing a linear trend (for already SA data)
   % =========================================================================
    T=size(x,1);
    y=zeros(T,1);
    switch(a)
    case 1
    y=x;
    n_el=0;
    case 2
    y(2:end,:)=x(2:end,:)-x(1:end-1,:);
    n_el=1;
    case 3
    y=log(x);
```

```
 n_el=0;
 case 4
 y(2:end,:)=log(x(2:end,:))-log(x(1:end-1,:));
 n_el=1;
 case 5
 y(5:end,:)=log(x(5:end,:))-log(x(1:end-4,:));
 n_el=4;
 case 6
 y(13:end,:)=log(x(13:end,:))-log(x(1:end-12,:));
 n_el=12;
 case 7
 yt=hpfilter(log(x),1600);
 y=log(x)-yt;
 n_el=0;
 case 8
 yt=hpfilter(log(x),14400);
 y=log(x)-yt;
 n_el=0;
 case 9
 yt=hpfilter(log(x),1600);
 yt=log(x)-yt;
 y(5:end,:)=yt(5:end,:)-yt(1:end-4,:);
 n_el=4;
 case 10
 yt=hpfilter(log(x),14400);
 yt=log(x)-yt;
 y(13:end,:)=yt(13:end,:)-yt(1:end-12,:);
 n_el=12;
 case 11
 yt=log(x);
 yt=yt(13:end,:)-yt(1:end-12,:);
 y(14:end,:)=yt(2:end,:)-yt(1:end-1,:);
 n_el=13;
 case 12
 y=detrend(log(x));
 n_el=0;
 end
end
```

References

Hamilton, J. D. (1994). *Time Series Analysis.* Princeton University Press.

Lütkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis.* Springer.