

GAUSS Programming Tutorial for Basic Econometric Models

Spring 2009

Kyu Ho Kang

Washington University in Saint Louis

Instructor: Kyu Ho Kang, Seigle 374, khkang@wustl.edu

Time and Location: Sat 10:30-11:30pm – Seigle 103

Website: <http://kyukang.net/>

Office Hours: by appointment

No required text.

No prerequisites.

Prior experience with GAUSS (or Matlab or a related language) is not assumed.

Description: This workshop provides basic GAUSS programming techniques to statistical inference for data analysis in a variety of application. Students will be able to effectively construct any GAUSS code without difficulties once they fully understand the econometric models in their research. This workshop would be also helpful in doing econometrics homeworks or macroeconomics calibration. For this purpose, in each class we discuss one of the selected basic econometric models listed below, and then we turn to programming the corresponding model with simulated or real data. There will be weekly programming assignments. Each assignment will not take more than 30 minutes to complete, and should be submitted via email by 10pm Fridays. Please feel

free to seek advice from the instructor or other students, but students should write up the code in their own words. All of solution codes will be posted on the website (<http://kyukang.net>) after the due date.

- 1. Introduction to GAUSS Language**
- 2. Central Limit Theorem**
- 3. Ordinary Least Squares Estimation**
- 4. Dummy variables and Chow test**
- 5. Generalized Least Squares Estimation**
- 6. Instrumental Variable Estimation**
- 7. Maximum Likelihood Estimation**
- 8. ARCH and GARCH Models**
- 9. Probit Models**
- 10. Autoregressive Models**
- 11. Recursive Vector Autoregressive Models**
- 12. Panel Analysis**
- 13. State-Space Model and Kalman Filter**
- 14. Markov Switching Models**

<References>

GAUSS Language reference

Johnston, Jack and John Dinardo, 1996, *Econometric Methods*, McGraw-Hill/Irwin, 4th edition.

Kim, Changjin and Charles R. Nelson, 1999, *State-Space Models with Regime-Switching: Classical and Gibbs-Sampling Approaches with Applications*, MIT Press, Cambridge

Lin, Kuan-Pin, 2001, *Computational Econometrics: GAUSS Programming for Econometricians and Financial Analysts*, ETEXT Publishing.

1 Introduction to GAUSS Language

1.1 Matrix Operations

GAUSS is one of the languages in which each variable is matrix. Thus the fundamental operators are based on matrix algebra. Indeed, this respect is very useful since many of econometric and economic models are represented in matrix form. All Gauss codes are in *italic* and each command must end with a semi-colon(*;*).

Making Matrices

$$a = \mathit{eye}(2);$$

implies a identity matrix with 2 dimension

$$a = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

$$a = \{1\ 4, 2\ 5, 3\ 6\};$$

is a 3×2 matrix.

$$a = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}.$$

Also

$$a = \{1\ 2\ 3\};$$

gives a 1×3 vector.

In addition,

$$a = \text{zeros}(2, 3);$$

$$b = \text{ones}(2, 3);$$

gives 2×3 matrix of zeros and ones, respectively.

Matrix Operators

$$x = a + b;$$

The symbol '+' means a matrix addition, i.e. element-by-element addition. If one of them is scalar, say B, each element of X will be sum of each element of A and the scalar B.

If both are matrices but incompatible sizes, then GAUSS will complain.

$$x[3, 2];$$

gives (3,2) element of matrix x.

$$x = \{1 \ 4, \ 2 \ 5, \ 3 \ 6\};$$

$$x[2 : 3, 1 : 2];$$

equals

$$\begin{pmatrix} 2 & 5 \\ 3 & 6 \end{pmatrix}.$$

Then $x[:, 2]$ yields a matrix containing the entire row of the second column. Similarly $x[3, :]$ yields a matrix containing the entire column of the third row.

$$a = \{1\ 2, 3\ 4\};$$

$$b = \{3\ 4, 5\ 6\};$$

$$x = a|b;$$

$$y = a \sim b;$$

$$z = a .* b;$$

implies that x, y and z will be

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{pmatrix} \text{ and } \begin{pmatrix} 5 & 12 \\ 21 & 28 \end{pmatrix},$$

respectively where ".*" is element-by-element multiplication.

1.2 Basic GAUSS Built-in Functions

$$y = \text{mean}(x);$$

- Computes the mean of every column of a matrix.

Input x : NxK matrix.

Output y : Kx1 matrix containing the mean of every column of x.

$$y = \text{inv}(x);$$

- inv returns the inverse of an invertible matrix

Input x : NxN matrix

Output y : NxN matrix

$$y = \text{sqrt}(x);$$

Input x : NxK matrix.

Output y : NxK matrix, the square roots of each element of x .

$$y = \text{stdc}(x);$$

- Computes the standard deviation of the elements in each column of a matrix.

Input x : NxK matrix.

Output y : Kx1 vector, the standard deviation of each column of x .

$$y = \text{rndn}(r, c);$$

- Creates a matrix of standard normal random numbers.

Input r : scalar, row dimension.

c : scalar, column dimension.

Output y : $r \times c$ matrix of normal random numbers having a mean of 0 and standard deviation of 1.

$$y = \text{rndu}(r, c);$$

- Creates a matrix of uniform (pseudo) random variables.

Input r : scalar, row dimension.

c : scalar, column dimension.

Output y : $r \times c$ matrix of uniform random variables between 0 and 1.

1.3 Graphics

$$xy(x, y);$$

- Graphs X vs. Y using Cartesian coordinates.

Input x : Nx1 or NxM matrix. Each column contains the X values for a particular line.

Output y : Nx1 or NxM matrix. Each column contains the Y values for a particular line.

$$\{ b, m, freq \} = histp(x, v);$$

- Computes and graphs a percent frequency histogram of a vector.

Input x : Mx1 vector of data.

v : Nx1 vector, the breakpoints to be used to compute the frequencies, or scalar, the number of categories.

Output b : Px1 vector, the breakpoints used for each category.

m : Px1 vector, the midpoints of each category.

freq: Px1 vector of computed frequency counts. This is the vector of counts, not percentages.

2 Central Limit Theorem

2.1 Statement of the Theorem

The random variables X_1, X_2, \dots, X_n form a random sample of size n from any given distribution with mean μ and variance σ^2 . Then the distribution of the random variable $\sqrt{n}(\bar{X}_n - \mu)/\sigma$ converges to a standard normal distribution as n goes to infinity.

2.2 Flow Chart

Setp 0> Specify μ, σ, n and the distribution of X_i ($i = 1, 2, \dots, n$)

Step 1> Put $j = 1$

Setp 2> Given the fixed n , sample $(X_1^{(j)}, X_2^{(j)}, \dots, X_n^{(j)})$ and calculate $\bar{X}_n^{(j)} = \frac{\sum_{i=1}^n X_i^{(j)}}{n}$

Step 3> Compute $\sqrt{n}(\bar{X}_n^{(j)} - \mu)/\sigma$ and save it

Put $j = j + 1$ and go to Step 2 until $j > T$ (sufficiently large number)

Step 4> Plot the distribution of $\sqrt{n}(\bar{X}_n^{(j)} - \mu)/\sigma$ ($j = 1, 2, \dots, T$)

2.3 Code

1 *new;*

- All gauss codes start with "new" and end with "end".
- Every command should end with ";".

2 *cls;*

- Clear the screen(command window).

3 *library pgraph;*

4 *T = 1000; @ # of draws @*

5 *n = 1000; @ sample size @*

- Between two "@"s you can put in the footnote. Gauss does not recognize it.

```

6   Xmat = zeros(T,1);    @ make a storage to save the sample mean @
7   mu = 0.5;           @ mean of uniform dist over [0,1] @
8   sig2 = 1/12;        @ variance of uniform dist over [0,1] @

9   j = 1;
10  do until j>T;
11      tmp = rndu(n,1);    @ random sample of size n @
12      xmat[j,1] = sqrt(n)*(meanc(tmp) - mu)/sqrt(sig2); @ save the draws @
13      j = j + 1;
14  endo;

```

- "Do looping" always starts with "do" and ends with "endo".
- "do until j > T" implies "do while j is less than or equal to T".
- Once line 14 is reached, Gauss goes back to line 10 and check whether the statement "j > T" is true or not.
- If it is true, Gauss terminates the looping and goes to line 15. Otherwise, it continues the looping until the statement becomes true.
- Eventually, xmat gets filled with the T realizations.

```

15  {a,b,c} = histp(xmat,100);    @ Plot the realizations using histogram @
16  "mean of xmat " meanc(xmat);
17  "variance of xmat " vcx(xmat);
18  end;    @ end the program @

```

3 Ordinary Least Squares Estimation

3.1 Background

3.1.1 Three Representations of Classical Linear Regression Equations:

There are three conventional ways to describe classical linear regression equations as below.

(i) Scalar form

$$y_t = \beta_1 + x_{2t}\beta_2 + x_{3t}\beta_3 + e_t, \quad e_t \sim iid(0, \sigma^2)$$

(ii) Vector form

$$\begin{aligned} y_t &= \begin{pmatrix} 1 & x_{2t} & x_{3t} \end{pmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + e_t, \quad e_t \sim iid(0, \sigma^2) \\ &= x'_t \beta + e_t \text{ where } x_t = \begin{pmatrix} 1 & x_{2t} & x_{3t} \end{pmatrix}' \text{ and } \beta = \begin{pmatrix} \beta_1 & \beta_2 & \beta_3 \end{pmatrix}' \end{aligned}$$

(iii) Matrix form

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix}}_Y = \underbrace{\begin{bmatrix} 1 & x_{21} & x_{31} \\ 1 & x_{22} & x_{32} \\ \vdots & \vdots & \vdots \\ 1 & x_{2T} & x_{3T} \end{bmatrix}}_X \underbrace{\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}}_\beta + \underbrace{\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_T \end{bmatrix}}_e \quad \text{where } e \sim (0, \sigma^2 I_T)$$

3.1.2 OLS Estimator

$$\hat{\beta} = (X'X)^{-1}X'Y \text{ and } \hat{\sigma}^2 = \hat{e}'\hat{e}/(T-3) \text{ where } \hat{e} = Y - X\hat{\beta}$$

Under the classical assumptions,

(1) X is nonstochastic (or strictly exogenous) and has full column rank (no multicollinearity)

(2) The disturbances are mutually independent and the variance is constant at each sample point, which can be combined in the single statement, $e \sim (0, \sigma^2 I_T)$,

OLS estimator is a BLUE (best linear unbiased estimator)

<Note: OLS Estimator >

The OLS estimator minimizes the residual sum of squares $u'u$ where $u = Y - Xb$.

Namely

$$\hat{\beta} = \arg \min_{\mathbf{b}} RSS = u'u$$

Then,

$$\begin{aligned} RSS &= u'u \\ &= (Y - X\mathbf{b})'(Y - X\mathbf{b}) \\ &= Y'Y - \mathbf{b}'X'y - y'X\mathbf{b} + \mathbf{b}'X'X\mathbf{b} \\ &= Y'Y - 2\mathbf{b}'X'y + \mathbf{b}'X'X\mathbf{b} \end{aligned}$$

since the transpose of a scalar is the scalar and thus $\mathbf{b}'X'y = y'X\mathbf{b}$. The first order conditions are

$$\begin{aligned} \frac{\partial RSS}{\partial \mathbf{b}} &= -2X'y + 2X'X\mathbf{b} = \mathbf{0} \\ (X'X)\mathbf{b} &= X'y \end{aligned}$$

which gives the OLS estimator.

<Note: Estimation of σ^2 >

Let $M_X = I_T - X(X'X)^{-1}X'$. It can be easily seen that M_X is a symmetric ($M_X' = M_X$)

and idempotent ($M_X M_X = M_X$) matrix with the properties that $M_X X = 0_T$ and $M_X \hat{e} = M_X Y = \hat{e}$. It follows that $M_X Y$ is the vector of residuals when Y is regressed on X . Also note that $\hat{e} = M_X Y = M_X (X\beta + e) = M_X e$ since $M_X X = 0_T$.

Thus

$$\begin{aligned}
 E[\hat{e}'\hat{e}] &= E[e'M_X' M_X e] \\
 &= E[e'M_X e] \text{ since } M_X' = M_X \text{ and } M_X M_X = M_X \\
 &= E[\text{tr}(e'M_X e)] \text{ since the trace of a scalar is the scalar} \\
 &= E[\text{tr}(e e' M_X)] \text{ since } \text{tr}(ABC) = \text{tr}(BCA) = \text{tr}(CAB) \\
 &= \sigma^2 \text{tr}(M_X) \\
 &= \sigma^2 \text{tr}(I_T) - \sigma^2 \text{tr}[X(X'X)^{-1}X'] \\
 &= \sigma^2 T - \sigma^2 \text{tr}[(X'X)^{-1}X'X] \text{ since } \text{tr}(ABC) = \text{tr}(BCA) = \text{tr}(CAB) \\
 &= \sigma^2 (T - k) \text{ where } k \text{ is the number of the regressors}
 \end{aligned}$$

$$E\left[\frac{\hat{e}'\hat{e}}{(T - k)}\right] = \sigma^2$$

Thus $\hat{\sigma}^2 = \hat{e}'\hat{e}/(T - k)$ is an unbiased estimator of σ^2 .

By making the additional assumption that e is normally distributed, we have

$$\begin{aligned}
 \hat{\beta} &= (X'X)^{-1}X'Y \\
 &= (X'X)^{-1}X'(X\beta + e) \\
 &= \beta + (X'X)^{-1}X'e \\
 &\sim N(\beta, \sigma^2(X'X)^{-1})
 \end{aligned}$$

In general, σ^2 is unknown, and hence the estimated variance of $\hat{\beta}$ is

$$\widehat{\text{Var}}(\hat{\beta}) = \hat{\sigma}^2 (X'X)^{-1}$$

3.1.3 Simple hypothesis test

Now it remains to show how to use these estimators to test simple hypotheses test about β where joint hypotheses test will be covered by the following section.

<Note>

(i) If $z \sim N(0, I_T)$ and A is a symmetric and idempotent matrix of rank r , then

$$z'Az \sim \chi^2(r)$$

(ii) $\text{rank}(M_X) = T - k$

Notice that $\hat{e} = M_X Y = M_X e$. Then

$$\frac{\hat{e}'\hat{e}}{\sigma^2} = \frac{e'M_X M_X e}{\sigma^2} = z'M_X z \sim \chi^2(T - k) \text{ where } z = \frac{e}{\sigma}$$

Also since $\hat{e}'\hat{e} = \hat{\sigma}^2(T - k)$,

$$\frac{\hat{\sigma}^2(T - k)}{\sigma^2} \sim \chi^2(T - k)$$

<Note>

If $z \sim N(0, 1)$ and $B \sim \chi^2(T - k)$ independently, then

$$\frac{z}{\sqrt{B/(T - k)}} \sim t_{T-k}$$

Then t value for each coefficient is obtained as follows.

$$\begin{aligned}
 \frac{\frac{\hat{\beta}_i - b_i}{\sqrt{\sigma^2(X'X)_i^{-1}}}}{\sqrt{\frac{\hat{\sigma}^2(T-k)}{\sigma^2}/(T-k)}} &= \frac{\hat{\beta}_i - b_i}{\sqrt{\hat{\sigma}^2(X'X)_i^{-1}}} \\
 &= \frac{\hat{\beta}_i - b_i}{\sqrt{\widehat{Var}(\hat{\beta}_i)}} \\
 &= \frac{\hat{\beta}_i - b_i}{\sqrt{s.e.(\hat{\beta}_i)}} \sim t_{T-k}
 \end{aligned}$$

where $H : \beta_i = b_i$ and $\hat{\sigma}^2(X'X)_i^{-1}$ is (i, i) element of $\hat{\sigma}^2(X'X)^{-1}$.

3.1.4 Model comparison

The vector of the dependent variable Y can be decomposed into the part explained by the regressors and the unexplained part.

$$Y = \hat{Y} + \hat{e} \text{ where } \hat{Y} = X\hat{\beta}$$

Then it follows that

$$\underbrace{Y'Y - T\bar{Y}^2}_{TSS} = \underbrace{\hat{Y}'\hat{Y} - T\bar{Y}^2}_{ESS} + \underbrace{\hat{e}'\hat{e}}_{RSS}$$

The coefficient of multiple correlation R is defined as the positive square root of

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

R^2 is the proportion of the total variation of Y explained by the linear combination of the regressors. This value is increasing by including any additional regressors even if the added regressors are irrelevant to the dependent. However, the adjusted R^2 ,

denoted by \bar{R}^2 may decrease with the addition of variables of low explanatory power.

$$\bar{R}^2 = 1 - \frac{RSS}{TSS} \times \frac{T-1}{T-k}$$

Two of the frequently used criteria for comparing the fit of various specifications is the *Schwarz criterion*

$$SC = \ln \frac{\hat{e}'\hat{e}}{T} + \frac{k}{T} \ln T$$

and the *Akaike information criterion*

$$AIC = \ln \frac{\hat{e}'\hat{e}}{T} + \frac{2k}{T}$$

It is important to note that these criterions favor a model with smaller sum of the squared residuals, but each criterion adds on a penalty for model complexity measured by the number of the regressors. Most statistics computer programs routinely produce those criterions.

3.2 Flow Chart

Step 1> Data Generating Process(DGP)

- 1.1 Define the model parameters.
- 1.2 Specify the number of observations.
- 1.3 Generate the independent variables and random disturbances.
- 1.4 Derive the dependent variable.

Step 2> Estimation

- 2.1 Define the regressors and the regressand.
- 2.2 Estimate the coefficients and obtain the residuals.
- 2.3 Use the residuals to estimate the variance.

3.3 Code

3.3.1 Without procedure

```
1  new;
2  cls;
3  library pgraph;
4  pqgwin many;

5  /* Step 1: DGP */
6  b1 = 1;
7  b2 = 2;
8  b0 = 0|0; @ null hypothesis @
9  k = 2; @ # of regressors @
10 sig2 = 4;
11 tru_b = b1|b2; @ True parameters @
12 T = 200; @ # of observations @
13 x2 = rndu(T,1)*5;
14 emat = rndn(T,1)*sqrt(sig2); @ Error term @
15 ymat = b1 + x2*b2 + emat;

16 /* Step 2: Estimation(OLS) */
17 X = ones(T,1)~x2; @ T by 2, Independent variables @
18 Y = ymat; @ T by 1, dependent variable @
19 bhat = inv(X'*X)*X'*Y; @ k by 1, Estimates for b1 and b2 @

20 Yhat = X*bhat; @ T by 1, fitted value @
21 ehat = Y - Yhat; @ T by 1, residuals @
22 sig2hat = ehat'ehat/(T-k); @ Estimates of variance @
```

```

23  varbhat = sig2hat*invpd(X'X); @ k by k, variance of bhat @
24  stde = sqrt(diag(varbhat)); @ k by 1, standard error @
25  t_val = (bhat-b0)./stde; @ k by 1, t values @
26  mY = meanc(Y); @ mean of the dependent variable @

27  TSS = Y'Y - T*mY^2;
28  RSS = ehat'ehat;
29  R2 = 1 - RSS/TSS; @ R-squared @
30  R2_ = 1 - (n-1)*RSS/(TSS*(n-k)); @ adjusted R-squared @

31  SC = ln(RSS/T) - k/T*ln(T); @ Schwarz criterion @
32  AIC = ln(RSS/T) - 2*k/T; @ AIC @

33  /* Results */
34  "-----";
35  true values Estimates S.E. t value ;
36  tru_b ~bhat ~stde ~t_val;
37  "-----";
38  "S.E. of regression is " sqrt(sig2hat);
39  "R2 is " R2;
40  "adjusted R2 is " R2_;
41  "SC is " SC;
42  "AIC is " AIC;
43  "-----";

44  title("Y and fitted Y");
45  xy(T,Y~Yhat);
46  title("Error term and Residuals");
47  xy(T,emat ~ehat);
48  end;

```

3.3.2 Making procedure

```
1  new;
2  cls;
3  a = 2;
4  b = 3;
5  c = 4;
6  {d,e} = Kang(a,b,c);
7  "d is " d;
8  "e is " e;
9  end;

10 proc(2)=Kang(x,y,z); @ '2' = # of outputs @
11 local out1, out2;
    - List all local variables which are recognized within procedure only..

12 out1 = x + y;
13 out2 = x*z;
14 retp(out1,out2);
    - List the outputs

15 endp;
```

- A procedure starts with 'proc' and ends with 'endp'.

3.3.3 With procedure

```
1  new;
2  cls;
3  library pgraph;
```

```

4   ppgwin many;

5   /* Step 1: DGP */
6   b1 = 1;
7   b2 = 2;
8   b0 = 0|0; @ null hypothesis @
9   k = 2; @ # of regressors @
10  sig2 = 4;
11  Tru_b = b1|b2; @ True parameters @
12  T = 200; @ # of observations @
13  x2 = rndu(T,1)*5;
14  emat = rndn(T,1)*sqrt(sig2); @ Error term @
15  ymat = b1 + x2*b2 + emat;

16  /* Step 2: Estimation(OLS) */
17  X = ones(T,1)~x2; @ T by 2, Independent variables @
18  Y = ymat; @ T by 1, dependent variable @
19  printr = 1; @ print out the estimation results if 1 @
20  {bhat, Yhat, residual, sig2hat, stde, t_val, p_val, F_val, R2, R2_, SC, AIC} = ols1(Y, X, printr);

21  title("Y and fitted Y");
22  xy(T, Ymat~Yhat);
23  title("Error term and Residuals");
24  xy(T, emat~residual);
25  end;

26  proc(12) = ols1(Y, X, printr);
27  local k, T, bhat, Yhat, ehat, sig2hat, varbhat, stde,
28  t_val, R, L, gam, F_val, mY, TSS, RSS, R2, SC, AIC;

```

```

29     T = rows(Y);
30     k = cols(X);
31     bhat = inv(X'*X)*X'*Y; @ k by 1, Estimates for b1 and b2 @
32     Yhat = X*bhat; @ T by 1, fitted value @
33     ehat = Y - Yhat; @ T by 1, residuals @
34     sig2hat = ehat'ehat/(T-k); @ Estimates of variance @
35     varbhat = sig2hat*invpd(X'X); @ k by k, variance of bhat @
36     stde = sqrt(diag(varbhat)); @ k by 1, standard error @
37     t_val = (bhat-b0)./stde; @ k by 1, t values @
38     p_val = cdftc(t_val,T-k); @ k by 1, p value @

39     mY = meanc(Y); @ mean of the dependent variable @
40     TSS = Y'Y - T*mY^2;
41     RSS = ehat'ehat;
42     R2 = 1 - RSS/TSS;
43     R2_ = 1 - (T-1)*RSS/(TSS*(T-k));
44     SC = ln(RSS/T) - k/T*ln(T);
45     AIC = ln(RSS/T) - 2*k/T;

46     /* Test H0: All coefficients are zero. Refer to the section 4 for details */
47     R = eye(k);
48     L = k; @ # of Restrictions @
49     gam = zeros(k,1);
50     F_val = (R*bhat - gam)'invpd(R*invpd(X'X)*R')*(R*bhat-gam)/(L*sig2hat);

51     if printr == 1;
52         /* Results */
53         "-----";

```

```

54     "Parameter Estimates S.E. t value p value";
55     seqa(1,1,k)~bhat~stde~t_val~p_val;
56     "-----";
57     "S.E. of regression " sqrt(sig2hat);
58     "F value " F_val;
59     "p value(F) " cdffc(F_val,L,T-k);
60     "R2 " R2;
61     "adjusted R2 " R2_;
62     "SC " SC;
63     "AIC " AIC;
64     "-----";
65     endif;

66     retp(bhat,Yhat,ehat,sig2hat,stde,t_val,p_val,F_val,R2,R2_,SC,AIC);
67     endp;

```

4 Dummy variables and Test of Structural Change

4.1 Background

One of the most important diagnostic criteria for an estimated equation is parameter consistency. A structural break issue occurs if the parameters differs from one subset of the data to another. The test of structural change can be carried out in many different but equivalent ways given exogenous change points. In this material we discuss the approach with the so-called dummy variables that take either zero or one. Note that his dummy variable regression can be easily generalized to capturing seasonal patterns in the time series data or individual effect in panel data.

4.2 Testing for structural change in intercept and slopes

$$y_t = (\beta_1 + \delta_1 D_t) + (\beta_2 + \delta_2 D_t) x_{2t} + e_t, \quad e_t \sim i.i.d(0, \sigma^2)$$

where $D_t = \begin{cases} 0, & t = 1, 2, \dots, \tau \\ 1, & t = \tau + 1, \tau + 2, \dots, T \end{cases}$

(i) Before break, $y_t = \beta_1 + \beta_2 x_{2t} + e_t$

(ii) After break, $y_t = (\beta_1 + \delta_1 D_t) + (\beta_2 + \delta_2 D_t) x_{2t} + e_t$

In vector form,

$$\begin{array}{c} \left[\begin{array}{c} y_1 \\ \vdots \\ y_\tau \\ y_{\tau+1} \\ \vdots \\ y_T \end{array} \right] \\ Y \end{array} = \begin{array}{c} \left[\begin{array}{cccc} 1 & 0 & x_{21} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & x_{2\tau} & 0 \\ 1 & 1 & x_{2\tau+1} & x_{2\tau+1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & x_{2T} & x_{2T} \end{array} \right] \\ X \end{array} \begin{array}{c} \left(\begin{array}{c} \beta_1 \\ \delta_1 \\ \beta_2 \\ \delta_2 \end{array} \right) \\ \beta \end{array} + \begin{array}{c} \left[\begin{array}{c} e_1 \\ \vdots \\ e_\tau \\ e_{\tau+1} \\ \vdots \\ e_T \end{array} \right] \\ e \end{array}$$

Then the joint test hypothesis for parameter consistency becomes

$$H_0 : \delta_1 = \delta_2 = 0 \iff H_0 : R\beta = \gamma$$

where $R = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ and $\gamma = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

Recall that the F statistic is

$$\frac{(R\hat{\beta} - \gamma)' [R(X'X)^{-1}R']^{-1} (R\hat{\beta} - \gamma)}{L\hat{\sigma}^2} \sim F(L, T - k) \quad (1)$$

This Wald statistics can be alternatively expressed as

$$\frac{(T - k) [\hat{e}'_R \hat{e}_R - \hat{e}' \hat{e}]}{L\hat{e}' \hat{e}} \sim F(L, T - k) \quad (2)$$

4.2.1 Flow Chart

- Step 1> Run $y_t = \beta_1 + \delta_1 D_t + \beta_2 x_{2t} + \delta_2 D_t \times x_{2t} + e_t$ and get \hat{e} . (Unrestricted regression)
- Step 2> Run $y_t = \beta_1 + \beta_2 x_{2t} + e_t$ and get \hat{e}_R . (Restricted regression, i.e. $\delta_1 = \delta_2 = 0$)
- Step 3> (i) Specify R and γ to compute the F statistic (1).
(ii) Compute the F statistic (2) using \hat{e} and \hat{e}_R .

4.2.2 Code

```

1  new;
2  cls;

3  /**** Data loading ****/
4  t=144; @ 1960:1-2001:4 @
5  load data[t,3]=c:\data\watson.txt;
6  FFR = data[:,1]; @ Federal Funds rate @

```

```

7   inf = data[:,2]; @ inflation @
8   un = data[:,3]; @ Unemployment rate @

9   /**** Estimatin *****/
10  /* Step 1: Unrestricted OLS */
11  bp = 80;
12  D = zeros(80,1)|ones(t-80,1);
13  Y = FFR;
14  X1 = ones(t,1);
15  X2 = X1.*D;
16  X3 = inf;
17  X4 = inf.*D;
18  X5 = un;
19  X6 = un.*D;
20  X = X1~X2~X3~X4~X5~X6;
21  printr = 0;
22  {bhat,Yhat,resid,sig2hat,stde,t_val,p_val,F_val,R2,R2_,SC,AIC} = ols1(Y,X,printr);

23  /* method 1 : Wald Test */
24  @ 1st method @
25  R = 0~1~0~0~0~0|
26      0~0~0~1~0~0|
27      0~0~0~0~0~1;
28  L = rows(R); @ # of Restrictions @
29  gam=0|0|0;
30  F1_val=(R*bhat - gam)'inv(R*inv(X'X)*R')*(R*bhat-gam)/(L*sig2hat);

31  /* Step 2: Restricted OLS */
32  Y = FFR;

```

```

33   X0 = X1~X3~X5;
34   printr = 0;
35   {bhat0,Yhat0,resid0,sig2hat0,stde0,t_val0,p_val0,F_val0,R20,R20_,SC0,AIC0}
= ols1(Y,X0,printr);

36   /* method 2: Wald Test */
37   RSS_res=resid0'resid0; @ Restricted RSS @
38   RSS_unr=resid'resid; @ Unrestricted RSS @
39   k = cols(X); @ # of regressors in the unrestricted reg @
40   F2_val=(T-k)*(RSS_res - RSS_unr) / (L*RSS_unr);
41   "_____";
42   Result for Chow Test ;
43   "_____";
44   F value from method 1 F1_val;
45   F value from method 2 F2_val;
46   p value cdfc(F1_val,L,T-k);

47   end;

48   proc(12) = ols1(Y,X,printr);
49   local k,T,bhat,Yhat,ehat,sig2hat,varbhat,stde,p_val,R2_,
50   t_val,R,L,gam,F_val,mY,TSS,RSS,R2,SC,AIC;

51   T = rows(Y);
52   k = cols(X);
53   bhat = inv(X'*X)*X'*Y; @ k by 1, Estimates for b1 and b2 @
54   Yhat = X*bhat; @ T by 1, fitted value @
55   ehat = Y - Yhat; @ T by 1, residuals @
56   sig2hat = ehat'ehat/(T-k); @ Estimates of variance @
57   varbhat = sig2hat*invpd(X'X); @ k by k, variance of bhat @

```

```

58     stde = sqrt(diag(varbhat)); @ k by 1, standard error @
59     t_val = bhat./stde; @ k by 1, t values @
60     p_val = 2*cdfc(abs(t_val),T-k); @ k by 1, p value @

61     mY = meanc(Y); @ mean of the dependent variable @
62     TSS = Y'Y - T*mY^2;
63     RSS = ehat'ehat;
64     R2 = 1 - RSS/TSS;
65     R2_ = 1 - (T-1)*RSS/(TSS*(T-k));
66     SC = ln(RSS/T) - k/T*ln(T);
67     AIC = ln(RSS/T) - 2*k/T;

68     /* Test H0: All coefficients are zero */
69     R = eye(k);
70     L = k; @ # of Restrictions @
71     gam = zeros(k,1);
72     F_val = (R*bhat - gam)'invpd(R*invpd(X'X)*R')*(R*bhat-gam)/(L*sig2hat);

73     if printr == 1;
74         /* Results */
75         "-----";
76         "Parameter Estimates S.E. t value p value";
77         seqa(1,1,k)~bhat~stde~t_val~p_val;
78         "-----";
79         "S.E. of regression " sqrt(sig2hat);
80         "F value " F_val;
81         "p value " cdfc(F_val,L,T-k);
82         "R2 " R2;

```

```

83     "adjusted R2 " R2_;
84     "SC " SC;
85     "AIC " AIC;
86     "-----";
87     endif;
88     retp(bhat, Yhat, ehat, sig2hat, stde, t_val, p_val, F_val, R2, R2_, SC, AIC);
89     endp;

```

4.3 Testing for structural change in Variance

$$y_t = \beta_1 + \beta_2 x_{2t} + e_t, \quad e_t \sim i.i.d(0, \sigma_t^2)$$

where $\sigma_t^2 = \begin{cases} \sigma_1^2, & t = 1, 2, \dots, \tau \\ \sigma_2^2, & t = \tau + 1, \tau + 2, \dots, T \end{cases}$

Then the test hypothesis for parameter consistency becomes

$$H_0 : \sigma_1^2 = \sigma_2^2 \text{ and } H_0 : \sigma_1^2 \neq \sigma_2^2$$

This is one of the tests for heteroskedasticity. If the null is rejected, it implies that the OLS estimator is no longer a BLUE. Note that

$$\frac{\hat{\sigma}_1^2(\tau - k)}{\sigma_1^2} \sim \chi^2(\tau - k)$$

$$\frac{\hat{\sigma}_2^2(T - \tau - k)}{\sigma_2^2} \sim \chi^2(T - \tau - k)$$

Hence

$$\begin{aligned} \frac{\frac{\hat{\sigma}_1^2(\tau-k)}{\sigma_1^2} / (\tau - k)}{\frac{\hat{\sigma}_2^2(T-\tau-k)}{\sigma_2^2} / (T - \tau - k)} &= \frac{\hat{\sigma}_1^2 / \sigma_1^2}{\hat{\sigma}_2^2 / \sigma_2^2} \\ &= \frac{\hat{\sigma}_1^2}{\hat{\sigma}_2^2} \text{ under the null } \sigma_1^2 = \sigma_2^2 \\ &\sim F(\tau - k, T - \tau - k) \end{aligned}$$

Intuitively, if $\hat{\sigma}_1^2 \approx \hat{\sigma}_2^2$, then $\frac{\hat{\sigma}_1^2}{\hat{\sigma}_2^2} \approx 1$.

4.3.1 Flow Chart

- Step 1> Run $y_t = \beta_1 + \beta_2 x_{2t} + e_t$ for the first subsample and estimate $\hat{\sigma}_1^2$
- Step 2> Run $y_t = \beta_1 + \beta_2 x_{2t} + e_t$ for the second subsample and estimate $\hat{\sigma}_2^2$
- Step 3> Compute $\hat{\sigma}_1^2 / \hat{\sigma}_2^2$

4.3.2 Code

```

1  new;
2  cls;
3  library pgraph;
4  #include H:\KangProc.s;
   - KangProc.s has all the procedures that can be used in this program.
5  T = 246; @ sample period, 1947:II-2008:III @

6  /*===== loading data =====*/
7  load data[T,1] = c:\data\gdpgrowth.txt; @ quarterly GDP growth rate @
8  tau = 150; @ Great moderation since 1984:IV @

```

```

9   /*===== Step2: Testing =====*/
10  /*= Step 2-1 : The 1st period =*/
11   Y1 = data[1:tau];
12   X1 = ones(tau,1);
13   printr = 0;
14   {bhat1,Yhat1,resid1,sig2hat1,stde1,t_val1,p_val1,F_val1,R21,R21_,SC1,AIC1}
= ols1(Y1,X1,printr);

15   /*= Step 2-2 : The 2nd period =*/
16   Y2 = data[tau+1:T];
17   X2 = ones(T-tau,1);
18   {bhat2,Yhat2,resid2,sig2hat2,stde2,t_val2,p_val2,F_val2,R22,R22_,SC2,AIC2}
= ols1(Y2,X2,printr);

19   F_val = sig2hat1/sig2hat2; @ F statistic @
20   k = cols(X1); @ # of regressors @

21   /* Test Results */
22   "_____";
23   "F-value " F_val;
24   "P-value " cdfc(F_val,tau-k,T-tau-k);
25   "_____";
26   end;

```



```

5   T = 246; @ sample period, 1947:II-2008:III @

6   /*===== loading data =====*/
7   load data[T,1] = c:\data\gdpgrowth.txt; @ quarterly GDP growth rate @
8   tau = 150; @ Great moderation since 1984:IV @

9   /*===== Step2: Testing =====*/
10  /*= Step 2-1 : The 1st period =*/
11  Y1 = data[1:tau];
12  X1 = ones(tau,1);
13  printr = 0;
14  {bhat1,Yhat1,resid1,sig2hat1,stde1,t_val1,p_val1,F_val1,R21,R21_,SC1,AIC1}
= ols1(Y1,X1,printr);

15  /*= Step 2-2 : The 2nd period =*/
16  Y2 = data[tau+1:T];
17  X2 = ones(T-tau,1);
18  {bhat2,Yhat2,resid2,sig2hat2,stde2,t_val2,p_val2,F_val2,R22,R22_,SC2,AIC2}
= ols1(Y2,X2,printr);

19  F_val = sig2hat1/sig2hat2; @ F statistic @
20  k = cols(X1); @ # of regressors @

21  /* Test Results */
22  "_____";

```

```

23  "F-value " F_val;
24  "P-value " cdfc(F_val,tau-k,T-tau-k);
25  "_____";

26  /*===== Step 3: Constructing variance-covariance matrix =====*/
27  omega = zeros(T,T);
28  Dmat = zeros(tau,1)|ones(T-tau,1); @ Dummy variable @
29  v = sig2hat1*(1-Dmat) + sig2hat2*Dmat; @ daig(omega) @
30  omega = diagrv(omega,v);

31  /*===== Step 4: GLS Estimator =====*/
32  X = X1|X2; @ T by k @
33  Y = Y1|Y2; @ T by 1 @
34  b_hat = inv(X'inv(omega)*X)*X'inv(omega)*Y;
35  var_b = inv(X'inv(omega)*X);

36  /*===== Step 5: Results =====*/
37  "_____";
38  "GLS estimates " b_hat';
39  "S.E. " sqrt(diag(var_b))';
40  "_____";
41  end;

```

6 Instrumental Variable Estimation

6.1 Background

6.1.1 IV Estimator

Under the classical assumptions, OLS estimator is a BLUE. One of the central assumptions is no correlation of the disturbance term with one or more of the regressors. If this condition does not hold, OLS estimators biased and inconsistent. This condition can be violated by three major situations: measurement error, missing variables, simultaneous equations. A consistent estimator may be obtained by the use of instrumental variables, which is illustrated with a regression equation with missing variables.

(1) DGP

$$y_t = \beta_1 + x_{2t}\beta_2 + x_{3t}\beta_3 + w_t\lambda + e_t,$$

$$Cov[x_{2t}, x_{3t}] = 0, Cov[x_{2t}, w_t] \neq 0, \text{ and } E(x_{2t}e_t) = E(x_{3t}e_t) = E(w_t e_t) = 0$$

(2) Regression equation : w_t is a missing variable

$$y_t = \beta_1 + x_{2t}\beta_2 + x_{3t}\beta_3 + u_t, E(x_{2t}u_t) = 0, E(x_{3t}u_t) \neq 0$$

In a matrix form,

$$Y = X\beta + u = X\beta + W\lambda + e \text{ where } X = [X_1 \ X_2 \ X_3] \text{ and } \beta = [\beta_1 \ \beta_2 \ \beta_3]'$$

(i) Properties of OLS Estimator

Orthogonality between x_{3t} and disturbance term, u_t is not satisfied any longer because contains some information on w_t , which generates non-zero correlation between u_t and x_{3t} .

Then OLS estimator is

$$\begin{aligned}
 \hat{\beta}_{OLS} &= (X'X)^{-1}X'Y \\
 &= (X'X)^{-1}X'(X\beta + W\lambda + e) \\
 &= \beta + (X'X)^{-1}X'W\lambda + (X'X)^{-1}X'e \\
 &\rightarrow \frac{\partial Y}{\partial X} + \frac{\partial W}{\partial X} \frac{\partial Y}{\partial W} \\
 &= \beta + \frac{\partial W}{\partial X} \lambda
 \end{aligned}$$

with $\widehat{Var}(\hat{\beta}_{OLS}) = \hat{\sigma}^2 (X'X)^{-1}$ where $\hat{\sigma}^2 = \hat{u}'\hat{u}/(T - 3)$ and $\hat{u} = Y - X\hat{\beta}_{OLS}$.

Thus OLS Estimator is inconsistent because $\hat{\beta}_{OLS}$ captures the indirect effect $(\frac{\partial W}{\partial X} \lambda)$ as well as the direct effect (β) .

(ii) Properties of 2 SLS Estimator

Then suppose that it is possible to find a vector of instrumental variables, $\tilde{z}_t = [z_{1t} \ z_{2t}]'$ with two vital conditions:

(i) They are correlated with x_{3t} where $p \lim(\tilde{Z}'X_3)/T$ is a finite covariance matrix of full rank

(ii) They are asymptotically uncorrelated with the disturbance term u_t , namely $p \lim(\tilde{Z}'u)/T = 0$

Now define Z as

$$[X_1 \ X_2 \ Z_1 \ Z_2]$$

It should be noted that Z also satisfies those two properties.

<Note>—————

One of the OLS estimator properties is that the fitted values (\hat{Y}) and the residuals (\hat{u})

are orthogonal as follows.

$$\begin{aligned}
 \hat{Y}'\hat{u} &= \hat{\beta}'_{OLS}X'\hat{u} \\
 &= \hat{\beta}'_{OLS}X'(Y - X\hat{\beta}_{OLS}) \text{ since } \hat{Y} = X\hat{\beta}_{OLS} \text{ and } \hat{u} = Y - X\hat{\beta}_{OLS} \\
 &= \hat{\beta}'_{OLS}X'Y - \hat{\beta}'_{OLS}X'X\hat{\beta}_{OLS} \\
 &= \hat{\beta}'_{OLS}X'Y - \hat{\beta}'_{OLS}X'X(X'X)^{-1}X'Y \text{ since } \hat{\beta}_{OLS} = (X'X)^{-1}X'Y \\
 &= \hat{\beta}'_{OLS}X'Y - \hat{\beta}'_{OLS}X'Y = 0
 \end{aligned}$$

Hence $\hat{Y}'\hat{u} = 0$, which implies that OLS regression decomposes the dependent variable(Y) into two orthogonal variates, one of which is correlated with X whereas the other is not.

<Note>

2 Stage Least Squares Estimation

1st stage: Regress X on Z by OLS to obtain \hat{X} . That is,

$$X = Z\hat{\gamma} + \hat{v} = \hat{X} + \hat{v} \text{ and } \hat{X} = Z(Z'Z)^{-1}Z'X$$

2nd stage: Regress Y on \hat{X} by OLS to obtain $\hat{\beta}_{2SLS} = (\hat{X}'\hat{X})^{-1}\hat{X}'Y$.

Then, by 2SLS, we may have a IV Estimator, $\hat{\beta}_{IV}$ such that

$$\begin{aligned}
& \hat{\beta}_{IV} \\
&= (\hat{X}'\hat{X})^{-1}\hat{X}'Y \\
&= (X'Z(Z'Z)^{-1}Z'Z(Z'Z)^{-1}Z'X)^{-1}X'Z(Z'Z)^{-1}Z'Y \text{ since } \hat{X} = Z(Z'Z)^{-1}Z'X \\
&= (X'Z(Z'Z)^{-1}Z'X)^{-1}X'Z(Z'Z)^{-1}Z'Y \\
&= (X'Z(Z'Z)^{-1}Z'X)^{-1}X'Z(Z'Z)^{-1}Z'(X\beta + W\lambda + e) \text{ since } Y = X\beta + W\lambda + e \\
&= \beta + (X'Z(Z'Z)^{-1}Z'X)^{-1}X'Z(Z'Z)^{-1}Z'W\lambda \\
&\quad + (X'Z(Z'Z)^{-1}Z'X)^{-1}X'Z(Z'Z)^{-1}Z'e \\
&= \beta + \left(\frac{X'Z}{T} \left(\frac{Z'Z}{T} \right)^{-1} \frac{Z'X}{T} \right)^{-1} \frac{X'Z}{T} \left(\frac{Z'Z}{T} \right)^{-1} \frac{Z'W}{T} \lambda \\
&\quad + \left(\frac{X'Z}{T} \left(\frac{Z'Z}{T} \right)^{-1} \frac{Z'X}{T} \right)^{-1} \frac{X'Z}{T} \left(\frac{Z'Z}{T} \right)^{-1} \frac{Z'e}{T} \\
&\rightarrow {}^p\beta \left(= \frac{\partial Y}{\partial X} \right)
\end{aligned}$$

since $Cov(x_t, z_t) < \infty$, $Var(z_t) < \infty$, $Cov(w_t, z_t) = \infty$ and $Cov(z_t, e_t) = 0$

Notice that

$$\begin{aligned}
\widehat{Var}(\hat{\beta}_{IV}) &= \hat{\sigma}^2 (\hat{X}'\hat{X})^{-1} \\
&= \hat{\sigma}^2 (X'Z(Z'Z)^{-1}Z'Z(Z'Z)^{-1}Z'X)^{-1}
\end{aligned}$$

where $\hat{\sigma}^2 = \hat{u}'\hat{u}/(T-3)$ and $\hat{u} = Y - X\hat{\beta}_{IV}$. It is important to notice that the residuals are not $(Y - \hat{X}\hat{\beta}_{IV})$ but $(Y - X\hat{\beta}_{IV})$.

On the other hand, consider the case that Z contains the same number of columns as X , that is, the number of instrumental variables is equal to that of regressors correlated with the disturbance term. Now $X'Z$ is square, non-singular and thus invertible. Hence

the estimator simplifies to

$$\hat{\beta}_{IV} = (Z'X)^{-1}Z'Y$$

$$\text{and } \widehat{Var}(\hat{\beta}_{IV}) = \hat{\sigma}^2(Z'X)^{-1}(Z'Z)(X'Z)^{-1}.$$

6.1.2 Testing Endogeneity: Hausman Test

Given the assumption of $p\lim[Z'u/T] = 0$ and $p\lim[Z'X/T] \neq 0$, we test H_0 : $p\lim[X'u/T] = 0$.

Define $\hat{q} = \hat{\beta}_{IV} - \hat{\beta}_{OLS}$. Then Hausman has shown that

$$Var(\hat{q}) = Var(\hat{\beta}_{IV}) - Var(\hat{\beta}_{OLS})$$

Thus asymptotic test statistic could be a quadratic form such as

$$\hat{q}'Var(\hat{q})^{-1}\hat{q} \sim \chi^2(L)$$

where L is the number of the regressors that are correlated with the disturbance term u_t . In this particular case, $L = 1$.

<Note>

In the 1st stage, X and $\hat{X} + \hat{v}$ and $X'X = \hat{X}'\hat{X} + \hat{v}'\hat{v} > \hat{X}'\hat{X}$. Therefore,

$$Var(\hat{\beta}_{IV}) = \sigma^2 \left(\hat{X}'\hat{X} \right)^{-1} > Var(\hat{\beta}_{OLS}) = \sigma^2 (X'X)^{-1} \text{ under the null.}$$

The intuition behind this test statistic is as follows.

Under the null hypothesis, both estimators are consistent so that $|\hat{q}|$ becomes very small. Moreover OLS estimator is more efficient under the null in sense that $Var(\hat{\beta}_{IV}) - Var(\hat{\beta}_{OLS}) \gg 0$. Hence the statistic should be small and the hypothesis is not rejected.

Meanwhile under the alternative, the difference should be very different from zero due to the inconsistency of $\hat{\beta}_{OLS}$, which makes the statistic large and the null hypothesis would be rejected.

6.1.3 Flow Chart

Step 1> Data Generating Process (DGP)

- 1.1 Define the model parameters.
- 1.2 Specify the number of observations.
- 1.3 Generate the independent variables correlated with random disturbances.
- 1.4 Generate instrumental variables satisfying the conditions.
- 1.5 Construct the dependent variable.

Step 2> IV Estimation

- 2.1 Define the regressors, the regressand, and the collection of instrumental variables.
- 2.2 Compute the IV Estimates and its variance.

Step 3> Endogeneity Testing

- 3.1 Compute OLS estimates and its variance.
- 3.2 Obtain the Hausman test statistic and make an inference.

6.2 Code

```
1  new;
2  cls;
3  #include F:\KangProc.s;

4  /*===== Step 1: DGP =====*/
5  T = 500;
6  b1 = 1;
7  b2 = 2;
```

```

8   b3 = 3;
9   lam = 1;
10  tru_b = b1|b2|b3;
11  C_X = rndn(T,1)*1; @ common factor between X3 and W @
12  C_Z = rndn(T,1)*1; @ common factor between X3 and Z @
    - C_X creates the non-zero correlation between X3 and W.
    - C_Z creates the non-zero correlation between X3 and IV's.

```

```

13  X2 = rndn(T,1); @ independent of W @
14  X3 = C_X + C_Z + rndn(T,1);
15  W = C_X + rndn(T,1); @ missing variable @
16  emat = rndn(T,1);
17  ymat = b1 + X2*b2 + X3*b3 + W*lam + emat;

```

```

18  @ Generating IV @
19  Z1 = C_Z + rndn(T,1); @ IV1 @
19  Z2 = C_Z + rndn(T,1); @ IV2 @
20  Z = ones(T,1)~Z1~Z2~X2;

```

- However, Z and W are completely orthogonal, and thus Z and the disturbance term in the regression equation are uncorrelated.

```

21  /*===== Step 2: IV Estimation =====*/
22  X = ones(T,1)~X2~X3; @ X3 is missing @
23  Y = ymat;

24  /* OLS */
25  printr = 0;
26  {b_OLS,Yhat,resid,sig2hat,stde,t_val,p_val,F_val,R2,R2_,SC,AIC} = ols1(Y,X,printr);
27  Varb_OLS = sig2hat*invpd(X'X);

```

```

28  /* IV */
29  b_IV = inv(X'Z*inv(Z'Z)*Z'X)*X'Z*inv(Z'Z)*Z'Y; @ IV Estimator @
30  ehat = Y - X*b_IV;
31  sig2hat = ehat'ehat/(T-3);
32  Varb_IV = sig2hat*inv(X'Z*inv(Z'Z)*Z'X);

33  /*===== Step 3: Hausman-Test =====*/
34  qhat = b_IV - b_OLS;
35  Varq = Varb_IV - Varb_OLS;
36  HT = qhat'inv(Varq)*qhat;

37  "_____";
38  "True b2 b2_OLS b2_IV";
39  tru_b ~ b_ols ~ b_IV;
40  "_____";
41  "Hausman test statistics " HT;
42  "p value " cdfchic(HT,1);

43  end;

```

7 Maximum Likelihood Estimation

7.1 Maximum Likelihood Estimator

Consider the following simple linear model,

$$y_t = \beta_1 + x_t\beta_2 + e_t, \quad e_t \sim iidN(0, \sigma^2)$$

The likelihood function specifies the plausibility or likelihood of the data given the parameter vector, $\theta = \left(\beta_1 \quad \beta_2 \quad \sigma^2 \right)'$. In the maximum likelihood method, the ML estimates, $\hat{\theta}_{ML}$ can be obtained by maximizing the log of the likelihood function as follows.

$$\hat{\theta}_{ML} = \arg \max \ln L(\theta|Y)$$

For constructing the likelihood function, it is extremely useful to consider *the prediction error decomposition*.

Note that $y_t|I_{t-1} \sim N(E[y_t|I_{t-1}], Var[y_t|I_{t-1}])$ where the prediction of y_t conditional on $I_{t-1} = (x_1, x_2, \dots, x_t, y_1, y_2, \dots, y_{t-1})$, $E[y_t|I_{t-1}] = \beta_1 + x_t\beta_2$ and $Var[y_t|I_{t-1}] = \sigma^2$ by the model specification. Then, the prediction error is,

$$\eta_t = y_t - E[y_t|I_{t-1}] = y_t - \beta_1 - x_t\beta_2 = e_t$$

and the variance of η_t ,

$$\sigma_t^2 = Var[y_t|I_{t-1}] = Var[y_t - E[y_t|I_{t-1}]] = Var[y_t - \beta_1 - x_t\beta_2] = \sigma^2$$

Once the prediction error and its variance are specified, constructing the log likelihood

becomes so easy.

$$\begin{aligned}
 f(y_t|\theta, I_{t-1}) &= f(\eta_t|\theta) \\
 &= \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left(-\frac{1}{2} \frac{\eta_t^2}{\sigma_t^2}\right) \\
 \ln f(y_t|\theta, I_{t-1}) &= -\frac{1}{2} \ln(2\pi\sigma_t^2) - \frac{1}{2} \frac{\eta_t^2}{\sigma_t^2} \\
 \ln L(\theta|Y) &= \sum_t^T \ln f(y_t|\theta, I_{t-1}) \text{ since } L(\theta|Y) = \prod_{t=1, \dots, T} f(y_t|\theta, I_{t-1}).
 \end{aligned}$$

By using the inverse of the negative of the second derivative of the log likelihood function (Hessian) evaluated at $\theta = \hat{\theta}_{ML}$, the variance of $\hat{\theta}_{ML}$, $Var(\hat{\theta}_{ML})$ can be estimated.

Recall the properties of MLE,

- *asymptotical normality*
- *asymptotical consistency*
- *asymptotical efficiency*

which can be described as

$$\hat{\theta}_{ML} \sim^a N(\theta, I^{-1}(\theta))$$

and those properties hold even if the specified distribution of the prediction error is not normally distributed.

7.2 Wald and Likelihood Ratio Test

Any joint linear hypothesis $H_0 : h(\theta) = 0$ can be expressed as follows.

$$H_0 : h(\theta) = R\theta - r = 0$$

For example, $\beta_1 = \beta_2 = 0$ is implied by $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ and $r = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Notice that the number of rows (columns) of R is equal to the number of the restrictions (parameters).

(i) In the Wald test, the unrestricted regression equation only is required to be estimated to obtain the estimates $\hat{\theta}_{UNR}$. The Wald test statistics is expressed by

$$Wald = h(\hat{\theta}_{UNR})' [Var(h(\hat{\theta}_{UNR}))]^{-1} h(\hat{\theta}_{UNR}) \sim^a \chi^2(q)$$

where q is the number of restrictions and $Var(h(\hat{\theta}_{UNR})) = RVar(\hat{\theta}_{UNR})R'$.

(ii) The likelihood ratio test statistic is defined as

$$LR = -2 [\ln L_{RES} - \ln L_{UNR}] \sim^a \chi^2(q)$$

where $\ln L_{RES}$ and $\ln L_{UNR}$ are the log likelihood value for the restricted and the unrestricted model, respectively. Thus the calculation of the LR statistic thus requires the fitting of both the restricted and the unrestricted model.

<Note>

It is very useful to notice the famous inequality for the test statistics in linear models, namely,

$$Wald \geq LR \geq LM$$

Hence even though they are asymptotically equivalent but in general may give different numerical results in finite samples.

7.3 Flow Chart for MLE

(i) The procedure in Gauss

Step 1> DGP

Step 2> Specify the initial value for each parameter to be estimated.

Step 3> Define and construct the log likelihood function. (See (ii) below)

Step 4> ML estimates are obtained.

Step 5> Evaluate the Hessian at the ML estimates to get their variance-covariance.

(ii) Constructing the log likelihood function

Step 0> Put $t = 1$ and $\ln L=0$.

Step 1> Define the parameters to be estimated.

Step 2> Specify η_t and σ_t^2 at time t .

Step 3> Compute $\ln f(\eta_t|\theta) = -0.5 \ln(2\pi\sigma_t^2) - 0.5\eta_t^2/\sigma_t^2$

Step 4> $\ln L = \ln L + \ln f(\eta_t|\theta)$, $t = t + 1$ and go to Step 2> while $t \leq T$.

7.4 Constrained MLE and Delta Method

Since the variance, σ^2 should not be negative, we need to impose a restriction that can not have a negative value in the process of optimization. For the constrained MLE, one more step which is a parameter transformation is added between Step 1 and 2 in the flow chart (ii). Then one of the most convenient ways is as follows.

$$\pi : R^2 \rightarrow R \times R^{++} \text{ when } \pi(\alpha_1, \alpha_2) = (\alpha_1, \exp(\alpha_2)) = (\mu, \sigma^2)$$

That is, we specify a new function as above by constructing one more procedure for this transformation.

More importantly, we should notice that the initial values in Step 1 in (2) are NOT for μ and σ^2 , but for α_1 and α_2 . Thus basically ‘xout’ refers to $(\hat{\alpha}_1, \hat{\alpha}_2)$, not $(\hat{\mu}, \hat{\sigma}^2)$ even though $\hat{\alpha}_1 = \hat{\mu}$ in this model. Finally $(\hat{\alpha}_1, \exp(\hat{\alpha}_2)) = (\hat{\mu}, \hat{\sigma}^2)$.

For the variance-covariance of $(\hat{\mu}, \hat{\sigma}^2)$, we can use Delta method as follows:

$$\begin{aligned} \text{Var}(\hat{\mu}, \hat{\sigma}^2) &= \text{Var}(\pi(\hat{\alpha}_1, \hat{\alpha}_2)) \approx \left(\frac{\partial \pi(\hat{\alpha})}{\partial \alpha} \right) \text{Var}(\hat{\alpha}) \left(\frac{\partial \pi(\hat{\alpha})}{\partial \alpha} \right)' \\ \text{where } \hat{\alpha} &= (\hat{\alpha}_1, \hat{\alpha}_2) \end{aligned}$$

Example) $X \sim (0, 3)$ and $\pi(X) = 2X$. Then

$$\begin{aligned} \text{Var}(\pi(X)) &= \text{Var}(2X) = \left(\frac{\partial \pi(X)}{\partial X} \right) \text{Var}(X) \left(\frac{\partial \pi(X)}{\partial X} \right)' \\ &= 2\text{Var}(X)2' = 2 \times 3 \times 2 = 12 \end{aligned}$$

7.5 Code

7.5.1 ML Estimation

```

1  new;
2  cls;

3  /*===== Step 1: DGP =====*/
4  T = 200; @ # of observation @
5  b1 = 2;
6  b2 = 2;
7  sig2 = 4; @ True conditional variance @
8  tru_para = b1|b2|sig2;
9  emat = rndn(T,1)*sqrt(sig2); @ error term @
10 X1 = ones(T,1);
11 X2 = rndu(T,1)*4;
12 X = X1~X2;
13 y = X1*b1+X2*b2+emat;

14 /*=====Step 2: Unrestricted MLE =====*/

```

```

15   _qn_PrintIters = 0; @if 1, print iteration information. Default = 0 @
16   prmtr_in = 0|0|3; @ Initial values for b1, b2, and sig2 @
17   {xout1, fout1, gout1, cout1} = qnewton(ℰln_L1,prmtr_in);

```

- For MLE, we should pass two elements into Gauss.

(i) Objective function

(ii) Initial values

- (i) ln_L1 = objective function. This procedure must have one input argument, a vector of parameter values, and one output argument, the value of the function evaluated at the input vector of parameter values.

(ii) prmtr_in = a vector of initial values

- xout1 = $\hat{\theta}$

- fout1 = the negative value of the unrestricted log likelihood function evaluated at $\hat{\theta}_{UNR}$

$$= -\ln L(\hat{\theta}|Y)$$

```

18   cov1 = inv(hessp(ℰln_L1,xout1));
19   lnL_unr = -fout1; @ Log likelihood under H1 @

20   /*=====Step 3: Restricted MLE =====*/
21   /* Suppose that H0: b2 = 0 */
22   _qn_PrintIters = 0; @if 1, print iteration information. Default = 0 @
23   prmtr_in = 0|3; @ Initial values for b1 and sig2 @
24   {xout0, fout0, gout0, cout0} = qnewton(ℰln_L0,prmtr_in);
25   lnL_res = -fout0; @ Log likelihood under H0 @

26   /*=====Step 4: Testing: b2=0 =====*/
27   @ LR Test @
28   LR_stat = -2*(lnL_res-lnL_unr);

```

```

29   @ Wald Test @
30   R = 0~1~0;
31   gam = 0;
32   Wald_stat = (R*xout1-gam)'inv(R*cov1*R')*(R*xout1-gam);

33   /*===== Resutls =====*/
34   "===== ";
35   "True values MLstimates" ;
36   -----;
37   tru_para ~xout1;
38   "===== ";
39   "variance of ML Estimates" diag(cov1[1:2,1:2])';
40   "===== ";
41   "LR statistics " LR_stat;
42   "Wald statistics " Wald_stat;
43   "===== ";

44   /*===== Log Likelihood Function for Unrestricted MLE =====*/
45   proc ln_L1(prmtr);
46   local b1hat, b2hat, sig2_hat, lnL, itr,eta_t,sig2_t,lnf;
47   b1hat = prmtr[1];
48   b2hat = prmtr[2];
49   sig2_hat = prmtr[3];

50   lnL = 0;
51   itr = 1;
52   do until itr>t;

53       /* Prediction error */
54       eta_t = y[itr] - b1hat*X1[itr] - b2hat*X2[itr];

```

```
55      /* Variance of prediction error */
```

```
56      sig2_t = sig2_hat;
```

- Prediction decomposition

```
57      /* log likelihood density */
```

```
58      lnf = -0.5*ln(2*pi*sig2_t) - 0.5*(eta_t^2)/sig2_t;
```

- log of the normal density for each observation.

```
59      lnL = lnL + lnf;
```

```
60      itr = itr + 1;
```

```
61      endo;
```

```
62      retp(-lnL);
```

- Indeed, Gauss minimizes any given objective function. To maximize the likelihood function, we thus define the output as $-\ln L$, instead of $\ln L$.

```
63      endp;
```

```
64      /*===== Log Likelihood Function for Restricted MLE =====*/
```

```
65      proc ln_L0(prmtr);
```

```
66      local b1hat, b2hat, sig2_hat, lnL, itr, eta_t, sig2_t, lnf;
```

```
67      b1hat = prmtr[1];
```

```
68      b2hat = 0;
```

```
69      sig2_hat = prmtr[2];
```

```
70      lnL = 0;
```

```
71      itr = 1;
```

```
72      do until itr>t;
```

```
73          /* Prediction error */
```

```
74          eta_t = y[itr] - b1hat*X1[itr] - b2hat*X2[itr];
```

```
75          /* Variance of prediction error */
```

```

76     sig2_t = sig2_hat;
77     /* log likelihood density */
78     lnf = -0.5*ln(2*pi*sig2_t) - 0.5*(eta_t^2)/sig2_t;
79     lnL = lnL + lnf;
80     itr = itr + 1;
81     endo;

82     retp(-lnL);
83     endp;

```

7.5.2 Constrained MLE

```

1     new;

2     /*===== loading data =====*/
3     T = 246; @ sample period, 1947:II-2008:III @
4     load data[T,1] = c:\data\gdpgrowth.txt; @ quarterly GDP growth rate @
5     Y = data[2:T];
6     X = data[1:T-1];
7     T = rows(Y); @ sample size @

8     /*=====Step 2: MLE =====*/
9     prmtr_in = 1|1|0; @ Initial values for a1, a2 and a2 @
10    _qn_PrintIters = 1;
11    {xout, fout, gout, cout} = qnewton(ℰlik_f,prmtr_in);
12    xout_fnl = trans(xout); @ beta @
13    cov = inv(hessp(ℰlik_f,xout)); @ variance-covariance of alpha @
14    grad = gradp(ℰtrans,xout);

```

```

15  cov_fnl = grad*cov*grad'; @ variance-covariance of beta by Delta method @
16  stde = sqrt(diag(cov_fnl));

17  /*===== Step 3: Results =====*/
18  "===== ";
19  "MLstimates   S.D. " ;
20  "_____";
21  xout_fnl~stde;
22  "===== ";
23  " Log likelihood function value  " -fout;
24  "===== ";

25  /*===== Procedure for constructing the likelihood function =====*/
26  proc lik_f(prmtr);
27  local prmtr1, mu,phi,sig2, lnL, itr,eta_t,sig2t,lnf;
28  prmtr1 = trans(prmtr); @ parameter transformation @
29  mu = prmtr1[1];
30  phi = prmtr1[2];
31  sig2 = prmtr1[3];
32  lnL = 0;

33  itr = 1;
34  do until itr>t;
36      eta_t = Y[itr]- mu - phi*X[itr];
37      sig2t = sig2;
38      lnf = -0.5*ln(2*pi*sig2t)-0.5*(eta_t^2)/sig2t;
39      lnL = lnL + lnf;
40      itr = itr + 1;
41  endo;

```

```
42  retp(-lnL);
43  endp;

44  /* Procedure for trans */
45  proc trans(prmtr);
46  local prmtr1;
47      prmtr1 = prmtr;
48      prmtr1[2] = prmtr[2]/(1+abs(prmtr[2])); @ phi @
49      prmtr1[3] = exp(prmtr[3]); @ sig2 @
50      retp(prmtr1);
51  endp;
```

8 ARCH and GARCH Models

8.1 Background

Conditional heteroscedasticity is very common in many of financial time series data. One of the most popular approaches to deal with it is GARCH specification where ARCH belongs to this category. The basic idea is that the mean corrected dependent is serially uncorrelated, but dependent.

8.1.1 ARCH

The first model for conditional volatility modeling is the ARCH model of Engle (1982). Consider the simplest case, ARCH(1):

$$y_t = \mu + e_t,$$

where $e_t = \sigma_t \varepsilon_t$, $\varepsilon_t \sim iidN(0, 1)$ and $\sigma_t^2 = \alpha_0 + \alpha_1 e_{t-1}^2$

Note that e_t has zero mean and its conditional variance is given by $\sigma_t^2 = \alpha_0 + \alpha_1 e_{t-1}^2$. On the other hand, the unconditional variance σ^2 is $\alpha_0/(1 - \alpha_1)$, which only exists if $\alpha_0 > 0$ and $|\alpha_1| < 1$.

Thus the prediction error decomposition is the following:

$$\begin{aligned} \text{the prediction error, } \quad \eta_t &= y_t - E[y_t | I_{t-1}] = y_t - \mu \\ \text{the variance of } \eta_t, \quad \sigma_t^2 &= Var[\eta_t | I_{t-1}] = \alpha_0 + \alpha_1 e_{t-1}^2 \end{aligned}$$

8.1.2 GARCH

Consider the most frequent application in practice, GARCH(1,1):

$$y_t = \mu + e_t, \quad \varepsilon_t \sim iidN(0, 1)$$

where $e_t = \sigma_t \varepsilon_t$ and $\sigma_t^2 = \alpha_0 + \alpha_1 e_{t-1}^2 + \gamma_1 \sigma_{t-1}^2$

The only difference from ARCH(1) is that the conditional variance is expressed as a linear function of one lagged conditional variance as well as one lagged squared disturbance. In GARCH, the unconditional variance is $\alpha_0/(1 - \alpha_1 - \gamma_1)$, which only exists if $\alpha_0 > 0$ and $|\alpha_1 + \gamma_1| < 1$.

For the prediction error decomposition,

$$\text{the prediction error, } \eta_t = y_t - E[y_t|I_{t-1}] = y_t - \mu$$

$$\text{the variance of } \eta_t, \quad \sigma_t^2 = Var[\eta_t|I_{t-1}] = \alpha_0 + \alpha_1 e_{t-1}^2 + \gamma_1 \sigma_{t-1}^2$$

8.1.3 Flow Chart for MLE

(i) Main

Step 1> DGP

Step 2> Specify the initial value for each parameter to be estimated.

Step 3> Define and construct the log likelihood function. (See below)

Step 4> ML estimates are obtained

Step 5> Evaluate the Hessian at the ML estimates to get their variance-covariance.

(ii) Constructing the log likelihood function

Step 0> Put $t = 1$ and $\ln L = 0$

Step 1> Define the parameters to be estimated.

Step 2> Parameter transformation. ($|\alpha_1| < 1$ for ARCH, $|\alpha_1 + \gamma_1| < 1$ for GARCH)

Step 3> Specify the η_t and σ_t^2 at time t such that.

$$\eta_t = y_t - \mu = e_t$$

$$\sigma_t^2 = \begin{cases} \alpha_0 + \alpha_1 e_{t-1}^2 \text{ where } e_0^2 = \alpha_0/(1 - \alpha_1) & : ARCH \\ \alpha_0 + \alpha_1 e_{t-1}^2 + \gamma_1 \sigma_{t-1}^2 \text{ where } e_0^2 = \alpha_0/(1 - \alpha_1 - \gamma_1) & : GARCH \end{cases}$$

Step 4> Compute $\ln f(\eta_t|\theta) = -0.5 \ln(2\pi\sigma_t^2) - 0.5\eta_t^2/\sigma_t^2$

Step 5> $\ln L = \ln L + \ln f(\eta_t|\theta)$

Step 6> Put $\eta_{t-1} = \eta_t$, $\sigma_{t-1}^2 = \sigma_t^2$ (for GARCH case only), and

$t = t + 1$ and go to Step 3 while $t \leq T$.

8.2 Code

8.2.1 ARCH

```
1  new;
2  cls;

3  /*===== DGP =====*/
4  T = 400;
5  mu = 4;
6  a0 = 1;
7  a1 = 0.5;
8  tru_para = mu|a0|a1; @True parameter @
9  emat = zeros(T,1);
10  tru_var = zeros(T,1); @ To save the true conditional variance @
11  tru_var[1] = a0/(1-a1); @ Initial variance @
12  emat[1,1] = rndn(1,1)*sqrt(tru_var[1]);

13  itr = 2;
14  do until itr>t;
15      tru_var[itr] = a0 + a1*emat[itr-1]^2;
16      emat[itr] = rndn(1,1)*sqrt(tru_var[itr]);
17      itr = itr + 1;
18  endo;

19  ymat = mu + emat;

20  /*===== Estimation =====*/
21  prmtr_in = 3|1|0;
22  {xout,fout,gout,cout} = qnewton(ℓlik_f,prmtr_in);
```

```

23  xout_fnl = trans(xout);
24  cov = inv(hessp(ℰlik_f,xout));
25  grad = gradp(ℰtrans,xout);
26  cov_fnl = grad*cov*grad';
27  stde = sqrt(diag(cov_fnl));

28  /*===== Results =====*/
29  "_____";
30  "true parameters estimates S.D." ;
31  tru_para ~ xout_fnl ~ stde;
32  "_____";

33  /*===== Estimates for conditonal variance =====*/
34  {var_hat,e_hat} = archout(xout);
35  title(" Conditional variance");
36  xy(t,tru_var ~ var_hat);
37  end;

38  /*===== Procedure 1 for lnL ===== */
39  proc lik_f(prmtr0);
40  local prmtr1, mmu,bb0, bb1, aa0, e_L, aa1, itr, lnL, lnf, eta_t, f_t, sig2t;
41  prmtr1 = trans(prmtr0);
42  mmu = prmtr1[1];
43  aa0 = prmtr1[2];
44  aa1 = prmtr1[3];
45  lnL = 0;
46  e_L = sqrt(aa0/(1-aa1));

47  itr = 1;
48  do until itr>t;

```

```

49     eta_t = ymat[itr]-mmu;
50     sig2t = aa0+aa1*(e_L^2);
51     lnf = -0.5*ln(2*pi*sig2t) - 0.5*(eta_t^2)/sig2t;
52     lnL = lnL + lnf;
53     e_L = eta_t;
54     itr = itr + 1;
55     endo;

56

57 retp(-lnL);
58 endp;

59 /*===== Procedure 2 for ARCH ===== */
60 proc(2) = archout(prmtr0);
61 local prmtr1, mmu,bb0, bb1, aa0, aa1, itr, lnL, lnf, eta_t, e_L, sig2t,varmat,ehat;
62 prmtr1 = trans(prmtr0);
63 mmu = prmtr1[1];
64 aa0 = prmtr1[2];
65 aa1 = prmtr1[3];
66 lnL = 0;
67 varmat = zeros(T,1); @ to save the conditional variance @
68 ehat = zeros(T,1); @ to save the residuals @
69 e_L = sqrt(aa0/(1-aa1)); @ Initial residual @

70 itr = 1;
71 do until itr>T;
72     eta_t = ymat[itr]-mmu;
73     sig2t = aa0+aa1*(e_L^2);
74     varmat[itr] = sig2t;
75     ehat[itr] = eta_t;

```

```

76      lnf = -0.5*ln(2*pi*sig2t)-0.5*(eta_ t^2)/sig2t;
77      lnL = lnL + lnf;
78      e_ L = eta_ t;
79      itr = itr + 1;
80      endo;
81      retp(varmat,ehat);
82      endp;

```

```

83      /* ===== procedure3 for trans ===== */
84      proc trans(prmtr); @ Trans @
85      local prmtr1;
86      prmtr1 = prmtr;
87      prmtr1[2] = abs(prmtr[2]);
88      prmtr1[3] = exp(prmtr[3])/(1+exp(prmtr[3]));
89      retp(prmtr1);

```

```

90      endp;

```

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} \rightarrow \begin{bmatrix} \beta_0 \\ |\beta_1| \\ \frac{\exp(\beta_2)}{1+\exp(\beta_2)} \end{bmatrix} = \begin{bmatrix} \mu \\ \alpha_0 \\ \alpha_1 \end{bmatrix} : ARCH$$

8.2.2 GARCH

```

1      new;
2      cls;

3      /*===== DGP ===== */
4      T = 400;
5      mu = 4;
6      a0 = 1;

```

```

7   a1 = 0.5;
8   gam1 = 0.3;
9   tru_para = mu|a0|a1|gam1; @True parameter @
10  emat = zeros(T,1);
11  tru_var = zeros(T,1); @ To save the true conditional variance @
12  tru_var[1] = a0/(1-a1-gam1); @ Initial variance @
13  emat[1,1] = rndn(1,1)*sqrt(tru_var[1]);

14  itr=2;
15  do until itr>t;
16      tru_var[itr] = a0 + a1*emat[itr-1]^2 + gam1*tru_var[itr-1];
17      emat[itr] = rndn(1,1)*sqrt(tru_var[itr]);
18      itr = itr + 1;
19  endo;
20  ymat = mu + emat;

21  /*=====Estimation =====*/
22  prmtr_in = 3|1|0|0;
23  {xout,fout,gout,cout} = qnewton(ℰlik_f,prmtr_in);
24  xout_fnl = trans(xout);
25  cov = inv(hessp(ℰlik_f,xout));
26  grad = gradp(ℰtrans,xout);
27  cov_fnl = grad*cov*grad';
28  stde = sqrt(diag(cov_fnl));

29  /*===== Results =====*/
30  "_____";
31  "true parameters estimates S.D." ;
32  tru_para ~ xout_fnl ~ stde;

```

```

33  "_____";

34  /*===== Estimates for conditonal variance =====*/
35  {var_hat,e_hat} = archout(xout);
36  title(" conditional variance");
37  xy(T,tru_var~var_hat);
38  end;

39  /*===== Procedure 1 for lnL ===== */
40  proc lik_f(prmtr0);
41  local prmtr1, mmu,bb0, bb1, aa0, e_L, aa1, itr, lnL, lnf, eta_t, f_t, sig2t,ggam1,sig2_L;
42  prmtr1 = trans(prmtr0);
43  mmu = prmtr1[1];
44  aa0 = prmtr1[2];
45  aa1 = prmtr1[3];
46  ggam1 = prmtr1[4];
47  lnL = 0;
48  e_L = sqrt(aa0/(1-aa1-ggam1));
49  sig2_L = aa0/(1-aa1-ggam1);

50  itr = 1;
51  do until itr>t;
52      eta_t = ymat[itr] - mmu; @ prediction error @
53      sig2t = aa0+aa1*(e_L^2)+ggam1*sig2_L; @ variance of the prediction
error @
54      lnf = -0.5*ln(2*pi*sig2t) - 0.5*(eta_t^2)/sig2t;
55      lnL = lnL + lnf;
56      e_L = eta_t; @ a lagged prediction error @
57      sig2_L = sig2t; @ variance of the lagged prediction error @

```

```

58   itr = itr + 1;
59   endo;

60   retp(-lnL);
61   endp;

62   /*===== Procedure 2 for GARCH ===== */
63   proc(2)=archout(prmtr0);
64   local prmtr1, mmu,bb0, bb1, aa0, aa1, itr, lnL, lnf, eta_t, e_L, sig2t,
65   varmat, ehat, sig2_L, ggam1;
66   prmtr1 = trans(prmtr0);
67   mmu = prmtr1[1];
68   aa0 = prmtr1[2];
69   aa1 = prmtr1[3];
70   ggam1 = prmtr1[4];
71   lnL = 0;
72   varmat = zeros(T,1); @ to save the conditional variance @
73   ehat = zeros(T,1); @ to save the residuals @
74   e_L = sqrt(aa0/(1-aa1-ggam1)); @ Initial residual @
75   sig2_L = aa0/(1-aa1-ggam1);

76   itr = 1;
77   do until itr>T;
78       eta_t = ymat[itr] - mmu;
79       sig2t = aa0 + aa1*(e_L^2) + ggam1*sig2_L;
80       varmat[itr] = sig2t;
81       ehat[itr] = eta_t;
82       lnf = -0.5*ln(2*pi*sig2t) - 0.5*(eta_t^2)/sig2t;
83       lnL = lnL + lnf;

```

```

84     e_L = eta_t;
85     sig2_L = sig2t;
86     itr = itr + 1;
87     endo;

88     retp(varmat,ehat);
89     endp;

90     /*===== procedure3 for trans =====*/
91     proc trans(prmtr); @ Trans @
92     local prmtr1;
93     prmtr1 = prmtr;
94     prmtr1[2] = abs(prmtr[2]);
95     prmtr1[3] = exp(prmtr[3])/(1+exp(prmtr[3])+exp(prmtr[4]));
96     prmtr1[4] = exp(prmtr[4])/(1+exp(prmtr[3])+exp(prmtr[4]));
97     retp(prmtr1);
98     endp

```

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \rightarrow \begin{bmatrix} \beta_0 \\ |\beta_1| \\ \frac{\exp(\beta_2)}{1+\exp(\beta_2)+\exp(\beta_3)} \\ \frac{\exp(\beta_3)}{1+\exp(\beta_2)+\exp(\beta_3)} \end{bmatrix} = \begin{bmatrix} \mu \\ \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} : GARCH$$

9 Probit

9.1 Background

Probit models are very convenient to analyze binary endogenous variables. We observe some variable y_t that takes on either 0 or 1. Define a latent variable y_t^* such that

$$y_t^* = x_t' \beta + \varepsilon_t, \quad \varepsilon_t \sim iidN(0, 1)$$

Then we do not observe y_t^* , but observable data y_t is assumed to take on 0 or 1 according to the following rule:

$$y_t = 1 \text{ if } y_t^* > 0 \text{ and } y_t = 0, \text{ otherwise.}$$

The likelihood density is given by

$$\begin{aligned} \Pr [y_t = 1] &= \Pr [y_t^* > 0] \\ &= \Pr [x_t' \beta + \varepsilon_t > 0] \\ &= \Pr [\varepsilon_t > -x_t' \beta] \\ &= \Pr [\varepsilon_t < x_t' \beta] \\ &= \Phi [x_t' \beta] \\ \Pr [y_t = 0] &= 1 - \Phi [x_t' \beta] \end{aligned}$$

9.2 Likelihood Function

If we have iid sampling, the likelihood should be the product of the probability of each observation. Hence

$$L = \prod_{t=1}^T \Phi [x_t' \beta]^{y_t} (1 - \Phi [x_t' \beta])^{1-y_t}$$

or

$$\ln L = \sum_{t=1}^T y_t \ln \Phi [x'_t \beta] + (1 - y_t) \ln (1 - \Phi [x'_t \beta])$$

9.3 Code

```
1  new;
2  cls;

3  /*=====DGP=====*/
4  T = 200;
5  Ymat = zeros(t,1);
6  X1mat = rndn(t,1);
7  X2mat = rndn(t,1);
8  xmat = X1mat~X2mat;
9  b1 = 2;
10 b2 = -1;
11 b = b1|b2;
12 tru_para = b;

13 itr = 1;
14 do until itr > t;
15     tmp = rndu(1,1);
16     xb = xmat[itr,]*b;
17     if tmp < cdfn(xb); ymat[itr] = 1; endif;
18     itr = itr + 1;
19 endo;

20 /*===== Estimation =====*/
21 b1_in = b1 + 0.5;
```

```

22  b2_in = b2 - 0.5;
23  prmtr_in = b1_in|b2_in;
24  {xout,fout,cout,gout} = qnewton(ℓlik_f,prmtr_in);
25  cov = inv(hessp(ℓlik_f,xout));
26  T_val = xout./sqrt(diag(cov));
27  "===== ";
28  "True value Estimates t value" ;
29  "-----";
30  tru_para ~ xout ~ t_val;
31  "===== ";

32  /*===== Likelihood Function =====*/
33  proc lik_f(prmtr);
34  local bb,lnL,cuml_den,lnf,itr;
35  bb = prmtr[1:2];
36  lnL = 0;

37  itr = 1;
38  do until itr > t;
39  cuml_den = cdfn(xmat[itr,]*bb);
40  lnf = ymat[itr]*ln(cuml_den) + (1 - ymat[itr])*ln(1-cuml_den);
41  lnL = lnL + lnf;
42  itr = itr + 1;
43  endo;

44  retp(-lnL);
45  endp;
46  end;

```

10 AR models

10.1 Background

Autoregressive models are widely used in forecasting stationary financial data. The analysis of a simple case, AR(2) model can readily be generalized to the general AR(p) model.

10.1.1 AR(2) model

An AR(2) model assumes the form

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \quad e_t \sim iid(0, \sigma^2)$$

We can represent this into an AR(1) process using a vector form.

$$\begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_{t-2} \end{bmatrix} + \begin{bmatrix} e_t \\ 0 \end{bmatrix}, \quad \begin{bmatrix} e_t \\ 0 \end{bmatrix} \sim iid \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2 & 0 \\ 0 & 0 \end{bmatrix} \right).$$

Then by recursive substitutions,

$$\begin{aligned} \begin{bmatrix} y_t \\ y_{t-1} \end{bmatrix} &= \begin{bmatrix} e_t \\ 0 \end{bmatrix} + \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} e_{t-1} \\ 0 \end{bmatrix} + \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix}^2 \begin{bmatrix} e_{t-2} \\ 0 \end{bmatrix} + \dots \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e_t \\ 0 \end{bmatrix} + F \begin{bmatrix} e_{t-1} \\ 0 \end{bmatrix} + F^2 \begin{bmatrix} e_{t-2} \\ 0 \end{bmatrix} + \dots \text{ if } F = \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \end{aligned}$$

Hence the impulse response of y_{t+j} to a shock at time t , $\frac{\partial y_{t+j}}{\partial e_t} = \frac{\partial y_t}{\partial e_{t-j}}$ can be obtained by computing (1,1) element of F^j .

10.1.2 Flow Chart

- Step 1> Load data
- Step 2> Regress $(y_t - \bar{y})$ on $(y_{t-1} - \bar{y})$ and $(y_{t-2} - \bar{y})$ to obtain ϕ_i 's
- Step 3> Construct F matrix
- Step 4> Save the (1,1) element of F^j ($j=1, 2, \dots, J$)

10.2 Code

```
-----  
1  new;  
2  cls;  
3  library pgraph;  
4  T = 167; @ Population period, 1960:1 ~2001:3 @  
5  j = 40; @ Length of horizon for impulse responses @  
  
6  /*===== loading data =====*/  
7  load data[T,3] =c:\data\watson.txt;  
- 'load' is a command for loading data.  
8  data = data[1:T,]; @ sample data @  
9  inf = data[:,1]; @ inflation rate @  
10  Un = data[:,2]; @ Un @  
11  ffr = data[:,3]; @ ffr @  
12  Ymat = Un;  
13  title("Unemployment rate");  
14  xy(T,ymat);  
  
15  /*===== OLS Estimation =====*/  
16  y0 = ymat[3:T]; @ Y(t) @  
17  Y = y0 - meanc(y0);  
18  x0 = ymat[2:T-1]~ymat[1:T-2]; @ Y(t-1) ~Y(t-2) @
```

```
19    X = x0 - mean(x0)';
```

- For convenience, we use the demeaned data so that we do not have to consider the intercept term.

```
20    phi_hat = inv(X'X)*X'Y;
```

```
21    ehat = Y - X*phi_hat;
```

```
22    sig2_hat = ehat'ehat/(T-4);
```

```
23    /*===== Impluse Response =====*/
```

```
24    imp = zeros(j,1); @ To save the estimated impulse response @
```

```
25    F_hat = phi_hat\1~0; @ F matrix @
```

```
26    F_j = eye(2);
```

- Constructing F matrix.

```
27    itr = 1;
```

```
28    do until itr > j;
```

```
29        Imp[itr] = F_j[1,1]; @ impulse response at horizon itr @
```

```
30        F_j = F_j*F_hat;
```

```
31        itr = itr + 1;
```

```
32    endo;
```

```
33    title("Impluse Response Function");
```

```
34    xy(seqa(0,1,j),imp~zeros(j,1));
```

```
35    end;
```

- y=seqa(start, inc, N) creates an additive sequence.

Input: start = scalar specifying the first element.

inc = scalar specifying the increment.

N = scalar specifying the number of elements in the sequence.

Output: y = Nx1 vector containing the specified sequence.

11 Recursive VAR

11.1 Background

To illustrate recursive VAR models, we focus on a simple case, VAR(2). The generalization of VAR(2) to VAR(p) models is straightforward.

11.1.1 Structural VAR(2) representation

A multivariate time series Y_t is a structural VAR process of order 2 if it follows the model

$$\underbrace{\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}}_B \underbrace{\begin{bmatrix} y_{1t} \\ y_{2t} \end{bmatrix}}_{Y_t} \quad (1)$$

$$= \underbrace{\begin{bmatrix} \gamma_{11}^1 & \gamma_{12}^1 \\ \gamma_{21}^1 & \gamma_{22}^1 \end{bmatrix}}_{\Gamma_1} \underbrace{\begin{bmatrix} y_{1t-1} \\ y_{2t-1} \end{bmatrix}}_{Y_{t-1}} + \underbrace{\begin{bmatrix} \gamma_{11}^2 & \gamma_{12}^2 \\ \gamma_{21}^2 & \gamma_{22}^2 \end{bmatrix}}_{\Gamma_2} \underbrace{\begin{bmatrix} y_{1t-2} \\ y_{2t-2} \end{bmatrix}}_{Y_{t-2}} + \underbrace{\begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix}}_{e_t}$$

Structural shocks, $e_t = \begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix} \sim iid \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right)$

Due to the endogeneity, these structural equations can not be simply estimated. Hence we need to consider the following reduced form VAR.

11.1.2 Reduced VAR(2) representation

$$\begin{aligned} Y_t &= B^{-1}\Gamma_1 Y_{t-1} + B^{-1}\Gamma_2 Y_{t-2} + B^{-1}e_t \\ &= \Phi_1 Y_{t-1} + \Phi_2 Y_{t-2} + u_t \end{aligned} \quad (2)$$

where $\Phi_1 = B^{-1}\Gamma_1$, $\Phi_2 = B^{-1}\Gamma_2$ and $u_t = B^{-1}e_t \sim (0, \Omega = B^{-1}B^{-1'})$

Note that

Structural Parameters : Γ_1, Γ_2, B

Reduced Parameters : Φ_1, Φ_2, Ω

In this reduced form, the endogeneity is resolved and so we can estimate these equations instead of the structural form with OLS. However this creates another problem as follows. The total number of the model parameters in the structural form is 12, but 11 in the reduced form. Thus all of parameters in (1) are not recovered by those of (2), which implies that at least one restriction on (1) is required.

For a recursive VAR, we set the (1,2) element of B^{-1} zero, that is

$$B^{-1} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix} : \text{lower triangular}$$

Our aim is to estimate the impulse responses,

$$\frac{\partial Y_{t+j}}{\partial e_t} = \frac{\partial Y_t}{\partial e_{t-j}} = \begin{bmatrix} \frac{\partial y_{1t}}{\partial e_{1t-j}} & \frac{\partial y_{1t}}{\partial e_{2t-j}} \\ \frac{\partial y_{2t}}{\partial e_{1t-j}} & \frac{\partial y_{2t}}{\partial e_{2t-j}} \end{bmatrix} : (2 \times 2)$$

From (2), Y_t can be expressed as MA(∞):

$$\begin{aligned} Y_t &= u_t + \psi_1 u_{t-1} + \psi_2 u_{t-2} + \psi_3 u_{t-3} + \dots \\ &= B^{-1} e_t + \psi_1 B^{-1} e_{t-1} + \psi_2 B^{-1} e_{t-2} + \psi_3 B^{-1} e_{t-3} + \dots \\ &= \theta_0 e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \theta_3 e_{t-3} + \dots \end{aligned}$$

Thus the impulse response can be obtained by the following 2×2 matrix:

$$\frac{\partial Y_{t+j}}{\partial e_t} = \frac{\partial Y_t}{\partial e_{t-j}} = \theta_j = \psi_j B^{-1} \text{ for each } j = 1, 2, \dots$$

Therefore, we shall estimate ψ_j and B^{-1} , respectively.

11.2 Flow Chart

Step 1> Regress Y_t on Y_{t-1} and Y_{t-2} .

$$\begin{aligned}
 & \begin{bmatrix} y_{13} & y_{23} \\ y_{14} & y_{24} \\ \vdots & \vdots \\ y_{1T} & y_{2T} \end{bmatrix} \\
 = & \begin{bmatrix} y_{12} & y_{22} \\ y_{13} & y_{23} \\ \vdots & \vdots \\ y_{1T-1} & y_{2T-1} \end{bmatrix} \begin{bmatrix} \Phi_{11}^1 & \Phi_{12}^1 \\ \Phi_{21}^1 & \Phi_{22}^1 \end{bmatrix} + \begin{bmatrix} y_{11} & y_{21} \\ y_{12} & y_{22} \\ \vdots & \vdots \\ y_{1T-2} & y_{2T-2} \end{bmatrix} \begin{bmatrix} \Phi_{11}^2 & \Phi_{12}^2 \\ \Phi_{21}^2 & \Phi_{22}^2 \end{bmatrix} + \begin{bmatrix} u_{13} & u_{23} \\ u_{14} & u_{24} \\ \vdots & \vdots \\ u_{1T} & u_{2T} \end{bmatrix} \\
 = & \begin{bmatrix} y_{12} & y_{22} & y_{11} & y_{21} \\ y_{13} & y_{23} & y_{12} & y_{22} \\ \vdots & \vdots & \vdots & \vdots \\ y_{1T-1} & y_{2T-1} & y_{1T-2} & y_{2T-2} \end{bmatrix} \begin{bmatrix} \Phi_{11}^1 & \Phi_{12}^1 \\ \Phi_{21}^1 & \Phi_{22}^1 \\ \Phi_{11}^2 & \Phi_{12}^2 \\ \Phi_{21}^2 & \Phi_{22}^2 \end{bmatrix} + \begin{bmatrix} u_{13} & u_{23} \\ u_{14} & u_{24} \\ \vdots & \vdots \\ u_{1T} & u_{2T} \end{bmatrix} : (T-2) \times 2.
 \end{aligned}$$

Step 2> Obtain $\hat{\Phi}_1(2 \times 2)$, $\hat{\Phi}_2(2 \times 2)$ and $\hat{u}((T-2) \times 2)$.

Step 3> For ψ_j , construct a \hat{F} matrix such as $\hat{F} = \begin{bmatrix} \hat{\Phi}_1 & \hat{\Phi}_2 \\ I_2 & 0_{2 \times 2} \end{bmatrix}$: 4×4 like in

univariate AR models.

Step 4> For B^{-1} , use \hat{u} to get $\hat{\Omega} = \hat{u}'\hat{u}/(T-2)$ and $\hat{B}^{-1} = Chol(\hat{\Omega})$.

Step 5> Compute $\hat{\theta} = \hat{\psi}_j \hat{B}^{-1}$ where $\hat{\psi}_j = [1:2, 1:2]$ block of \hat{F}^j

11.3 Code

```

1   New;
2   library pgraph;
3   cls;
4   t0 = 167; @ Population period, 1960:1 ~2001:3 @
5   k = 3; @ Dimension of VAR @
6   p = 4; @ # of Lag @
7   j = 24; @ Length of horizon for impulse response @

8   /*===== loading data =====*/
9   load data[t0,3] =c:\data\watson.txt;
10  data = data[1:t0,.]; @ sample @
11  inf = data[.,1]; @ inflation rate @
12  un = data[.,2]; @ Un @
13  ffr = data[.,3]; @ ffr @
14  y = inf~un~ffr;

- The order of the variables is determined here based on the degree of endogeniety.

15  y0 = y[p+1:t0,.]; @ t0-p by k, Y(t) @
16  y1 = y[4:t0-1,.]; @ Y(t-1) @
17  y2 = y[3:t0-2,.]; @ Y(t-2) @
18  y3 = y[2:t0-3,.]; @ Y(t-3) @
19  y4 = y[1:t0-4,.]; @ Y(t-4) @
20  y_lag = y1~y2~y3~y4;
21  @ Demean @
22  y0 = y0 - meanc(y0)';
23  y_lag = y_lag - meanc(y_lag)';

24  /*===== Estimation of Phi and Omega_hat =====*/
25  phi_hat = inv(y_lag'y_lag)*y_lag'y0; @ p*k by k @

```

```

26  F = phi_hat\'(eye((p-1)*k)~zeros(k*(p-1),k)); @ p*k by p*k @
27  T = rows(y0); @ rows of Y @
28  @ Omega_hat @
29  u_hat = y0 - y_lag*phi_hat; @ t-p by k @
30  Omeg = u_hat'*u_hat/T; @ variance- covariance matrix, k by k @
31  /* === Estimation of inv(B) using Cholesky decomposition === */
32  @ calculating the Inverse matrix of B by Cholesky Decomposition @
33  inv_B = chol(Omeg)'; @ Lower triangular matrix @

34  /*===== Impulse Responses =====*/
35  theta_i = zeros(j+1,k); @ Impluse response of inflation to each shock @
36  theta_u = zeros(j+1,k); @ Impluse response of Un to each shock @
37  theta_f = zeros(j+1,k); @ Impluse response of FFR to each shock @
- For example, theta_f[:,1] = response to e1t, theta_f[:,2] = response to e2t,
theta_f[:,3] = response to e3t.

```

```

38  FF = eye(p*k);
39  itr = 1;
40  Do until itr > (j+1);
41  psi_j = FF[1:k,1:k]; @ k by k @
42  theta = psi_j*inv_B; @ k by k @
43  theta_i[itr,:] = theta[1,]; @ 1 by k, Inf @
44  theta_u[itr,:] = theta[2,]; @ 1 by k, Un @
45  theta_f[itr,:] = theta[3,]; @ 1 by k, FFR @
46  FF = FF*F;
47  itr = itr + 1;
48  endo;

49  /*===== PLOT IMPULSE RESPONSES ===== */

```

```

50  graphset;
51  Begwind;
52  window(3,3,0);
53  Setwind(1);
54  title("Inflation shock to inflation");
55  xy(seqa(0,1,j+1), theta_i[.,1]~zeros(j+1,1));
56  Setwind(2);
57  title("Inflation shock to Un");
58  xy(seqa(0,1,j+1), theta_u[.,1]~zeros(j+1,1));
59  Setwind(3);
60  title("Inflation shock to FFR");
61  xy(seqa(0,1,j+1), theta_f[.,1]~zeros(j+1,1));
62  Setwind(4);
63  title("Un shock to Inflation");
64  xy(seqa(0,1,j+1), theta_i[.,2]~zeros(j+1,1));
65  Setwind(5);
66  title("Un shock to Un");
67  xy(seqa(0,1,j+1), theta_u[.,2]~zeros(j+1,1));
68  Setwind(6);
69  title("Un shock to FFR");
70  xy(seqa(0,1,j+1), theta_f[.,2]~zeros(j+1,1));
71  Setwind(7);
72  title("FFR shock to Inflation");
73  xy(seqa(0,1,j+1), theta_i[.,3]~zeros(j+1,1));
74  Setwind(8);
75  title("FFR shock to Un");
76  xy(seqa(0,1,j+1), theta_u[.,3]~zeros(j+1,1));
77  Setwind(9);

```

```
78  title("FFR shock to FFR");
79  xy(seqa(0,1,j+1),theta_f[:,3]~zeros(j+1,1));
80  Endwind;
81  end;
```

12 Panel Analysis (Fixed Effect Model)

12.1 Background

The starting point is the following:

$$y_{it} = \mu_i + x'_{it}\beta + e_{it} \text{ where } e_{it} \sim (0, \sigma^2) \text{ for all } i = 1, 2, \dots, n \text{ and } t = 1, 2, \dots, T$$

$y_{it} = 1 \times 1$ dependent variable for cross-section unit i at time t

$x_{it} = k \times 1$ explanatory variable for cross-section unit i at time t

For simplicity, we restrict our discussion to estimation with balanced panels. That is, we have the same number of observations on each cross-section, so that the total number of observation is $n \times T$. Also we consider the simple case of $n = 3$.

For the fixed effects, in which the individual effect μ_i is correlated with x_{it} , the time-invariant μ_1, μ_2 and μ_3 are treated as unknown parameters to be estimated.

Thus this model can be rewritten as

$$\begin{aligned} Y_1 &= \mu_1 + X_1\beta + e_1 : T \times 1, \quad e_1 \sim (0, \sigma^2 I_T) \\ Y_2 &= \mu_2 + X_2\beta + e_2 : T \times 1, \quad e_2 \sim (0, \sigma^2 I_T) \\ Y_3 &= \mu_3 + X_3\beta + e_3 : T \times 1, \quad e_3 \sim (0, \sigma^2 I_T) \end{aligned} \tag{1}$$

or

$$\begin{aligned} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} &= \begin{bmatrix} 1_T \\ 0_T \\ 0_T \end{bmatrix} \mu_1 + \begin{bmatrix} 0_T \\ 1_T \\ 0_T \end{bmatrix} \mu_2 + \begin{bmatrix} 0_T \\ 0_T \\ 1_T \end{bmatrix} \mu_3 + \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \beta + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} : 3T \times 1 \\ \text{where } \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} &\sim (0, \sigma^2 I_{3T}) \end{aligned}$$

or

$$Y = D_1\mu_1 + D_2\mu_2 + D_3\mu_3 + X\beta + e : 3T \times 1 \text{ where } e \sim (0, \sigma^2 I_{3T})$$

There are two simple ways to get a fixed effect estimator (or within group estimator).

(i) One way is using dummy variables as follows.

$$\begin{bmatrix} y_{11} \\ \vdots \\ y_{1T} \\ y_{21} \\ \vdots \\ y_{2T} \\ y_{31} \\ \vdots \\ y_{3T} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mu_1 + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \mu_2 + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \mu_3 + \begin{bmatrix} x_{11} \\ \vdots \\ x_{1T} \\ x_{21} \\ \vdots \\ x_{2T} \\ x_{31} \\ \vdots \\ x_{3T} \end{bmatrix} \beta + \begin{bmatrix} e_{11} \\ \vdots \\ e_{1T} \\ e_{21} \\ \vdots \\ e_{2T} \\ e_{31} \\ \vdots \\ e_{3T} \end{bmatrix} : 3T \times 1$$

Let $X^* = \begin{pmatrix} D_1 & D_2 & D_3 & X \end{pmatrix}$. Then

$$\begin{bmatrix} \hat{\mu}_1 \\ \hat{\mu}_2 \\ \hat{\mu}_3 \\ \hat{\beta} \end{bmatrix} = (X^{*'}X^*)^{-1} X^{*'}Y \text{ and } \hat{\sigma}^2 = \hat{e}'\hat{e}/(3T - 3 - k) \text{ where } \hat{e} = Y - X^* \begin{bmatrix} \hat{\mu}_1 \\ \hat{\mu}_2 \\ \hat{\mu}_3 \\ \hat{\beta} \end{bmatrix}$$

(ii) The other useful way to implement a fixed effects estimator is transforming all the variables by subtracting individual-specific means as the following:

$$\bar{Y}_i = \bar{\mu}_i + \bar{X}_i\beta + \bar{e}_i \quad (2)$$

Because the mean of for individual i is merely μ_i , we can difference (1) and (2) to yield

$$\begin{aligned} Y_1 - \bar{Y}_1 &= (X_1 - \bar{X}_1) \beta + (e_1 - \bar{e}_1) : T \times 1 \\ Y_2 - \bar{Y}_2 &= (X_2 - \bar{X}_2) \beta + (e_2 - \bar{e}_2) : T \times 1 \\ Y_3 - \bar{Y}_3 &= (X_3 - \bar{X}_3) \beta + (e_3 - \bar{e}_3) : T \times 1 \end{aligned}$$

or

$$\begin{aligned} Y^{**} &= \begin{bmatrix} y_{11} - \bar{Y}_1 \\ \vdots \\ y_{1T} - \bar{Y}_1 \\ y_{21} - \bar{Y}_2 \\ \vdots \\ y_{2T} - \bar{Y}_2 \\ y_{31} - \bar{Y}_3 \\ \vdots \\ y_{3T} - \bar{Y}_3 \end{bmatrix} = \begin{bmatrix} x_{11} - \bar{X}_1 \\ \vdots \\ x_{1T} - \bar{X}_1 \\ x_{21} - \bar{X}_2 \\ \vdots \\ x_{2T} - \bar{X}_2 \\ x_{31} - \bar{X}_3 \\ \vdots \\ x_{3T} - \bar{X}_3 \end{bmatrix} \beta + \begin{bmatrix} e_{11} - \bar{e}_1 \\ \vdots \\ e_{1T} - \bar{e}_1 \\ e_{21} - \bar{e}_2 \\ \vdots \\ e_{2T} - \bar{e}_2 \\ e_{31} - \bar{e}_3 \\ \vdots \\ e_{3T} - \bar{e}_3 \end{bmatrix} \\ &= X^{**} \beta + e^{**} \end{aligned}$$

Then

$$\hat{\beta} = (X^{**'} X^{**})^{-1} X^{**'} Y^{**} \text{ and } \hat{\sigma}^2 = \hat{e}' \hat{e} / (3T - 3 - k) \text{ where } \hat{e} = Y^{**} - X^{**} \hat{\beta}$$

12.1.1 Flow Chart

Step 1> DGP

Step 2> Transform all the variables by subtracting individual-specific means.

Step 3> Run OLS on the demeaned variables.

12.2 Code

```
1  new;
2  cls;

3  /*===== DGP =====*/
4  t = 20; @ size of times series @
5  n = 3; @ # of Cross section @
6  k = 1; @ # of regressors @

7  /*=== True parameters===*/
8  mu1 = -2;
9  mu2 = 0;
10 mu3 = 2;
11 Tru_b = 1;
12 sig2 = 2;
13 tru_para = tru_b|sig2;
14 x1 = rndu(T,1)*5 + mu1;
15 x2 = rndu(T,1)*5 + mu2;
16 x3 = rndu(T,1)*5 + mu3;
17 x = x1|x2|x3;
18 y1 = mu1 + x1*tru_b + rndn(T,1)*sqrt(sig2);
19 y2 = mu2 + x2*tru_b + rndn(T,1)*sqrt(sig2);
20 y3 = mu3 + x3*tru_b + rndn(T,1)*sqrt(sig2);
21 y = y1|y2|y3;

22 /*===== Estimation 1: Fixed Effect Estimator =====*/
23 @ Demean for individual 1 @
24 y1_dm = y1 - meanc(y1);
```

```

25  x1_dm = x1 - meanc(x1);
26  @ Demean for individual 2 @
27  y2_dm = y2 - meanc(y2);
28  x2_dm = x2 - meanc(x2);
29  @ Demean for individual 3 @
30  y3_dm = y3 - meanc(y3);
31  x3_dm = x3 - meanc(x3);
32  y_dm = y1_dm|y2_dm|y3_dm; @ Demeaned regressors @
33  x_dm = x1_dm|x2_dm|x3_dm; @ Demeaned independent variables @
34  b_hat = inv(x_dm'x_dm)*x_dm'y_dm; @ Fixed Effect Estimator @
35  ehat = y_dm-x_dm*b_hat; @ residuals @
36  sig2hat = ehat'ehat/(n*T-n-k); @ n=3, k=1 @
37  cov = sig2hat*inv(x_dm'x_dm);

38  "=====<Results>=====";
39  True value ;;tru_para';
40  Fixed Effect Estimates ;;b_hat'~sig2hat;
41  S.D of Estimates ;;sqrt(diag(cov))';
42  "=====";

43  /*==== Estimation 2: The Pooled Estimator =====*/
44  Ymat = y;
45  Xmat = ones(n*T,1)~X;
46  B_pool = inv(Xmat'Xmat)*Xmat'Ymat;
47  "=====";
48  Pooled Estimates ;;b_pool[2];
49  "=====";
50  end;

```

13 State Space model: Dynamic Factor Model

13.1 Background

A state-space representation consists of two equations: measurement and transition equation. The measurement equation specifies the relationship between observed and unobserved factors. The transition equation defines the evolution of unobserved factors as AR(1) form.

All techniques are illustrated by an example, a bivariate dynamic factor model. In this model, two observable variables share one dynamic common factor. Hence each variable has its own idiosyncratic component and common factor.

Measurement equation

$$\begin{bmatrix} y_{1t} \\ y_{2t} \end{bmatrix} = \begin{bmatrix} 1 \\ \gamma \end{bmatrix} \beta_t + \begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix} = H\beta_t + e_t$$

where $H = \begin{bmatrix} 1 \\ \gamma \end{bmatrix}$, $e_t = \begin{bmatrix} e_{1t} \\ e_{2t} \end{bmatrix} \sim iidN\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, R\right)$ and $R = \begin{bmatrix} \sigma_1^2 & \\ & \sigma_2^2 \end{bmatrix}$ for all time t

Transition equation

$$\beta_t = F\beta_{t-1} + v_t$$

where $v_t \sim iidN(0, Q)$, $F = \phi$ and $Q = \sigma_v^2$

Note that $\gamma = 0$ implies no correlation between two observable variables.

In order to estimate the parameters of this model using MLE, we should be able to construct log likelihood function. Kalman filter recursions provide a suitable method for computing prediction error decomposition. Kalman filter consists of six key equations. More importantly, once the underlying matrices (H, R, Q, F) are specified, implementation of the Kalman filter is straightforward and mechanical.

13.2 Code

```
1  new;
2  cls;
3  library pgraph;
4  /*===== DGP =====*/
5  t = 400; @ Sample size @
6  n = 2; @ # of variables @
7  k = 1; @ Dimension of beta @
8  gam = 2;
9  phi = 0.5;
10 sig21 = 1;
11 sig22 = 2;
12 sig2v = 0.8;
13 tru_para = gam| phi|sig2v|sig21| sig22;
14 beta = zeros(t,1);
15 b_L = 0;
16 b_var = sig2v/(1-phi^2);
17 itr = 1;
18 do until itr > t;
19     beta[itr] = phi*b_L+sqrt(sig2v)*rndn(1,1);
20     b_L = beta[itr];
21     itr = itr + 1;
22 endo;
23 y1 = beta + sqrt(sig21)*rndn(t,1);
24 y2 = gam*beta + sqrt(sig22)*rndn(t,1);
```

```

25  ymat = y1 ~ y2;

26  /*===== MLE =====*/
27  _qn_PrintIters = 1; @ print iteration information @
28  prmtr_in = 1|0|ln(1)|ln(1)|ln(1); @ initial value @
29  {xout,fout,gout,cout} = qnewton(ℓlik_f, prmtr_in);
30  xout_fnl = trans(xout);
31  "-----";
32  " true values estimates";
33  "-----";
34  tru_para ~ xout_fnl;
35  "-----";

36  /*===== filtering =====*/
37  xmat = filter(xout);
38  @ Filtered common factor @
39  title("Common Factor");
40  xy(t , xmat ~ beta ~ zeros(t,1));

41  /*===== Lik_f =====*/
42  proc lik_f(prmtr1);
43  local lnf, Ht,prmtr,Q,R,F,beta_ll,p_ll,lnL,ittr, pphi,
44  P_tl, eta_tl, f_tl, Kt, beta_tt, P_tt,beta_tl,
45  ggam,ssig21,ssig22,ssig2v;
46  prmtr = trans(prmtr1);
47  ggam = prmtr[1];
48  pphi = prmtr[2];
49  ssig2v = prmtr[3];
50  ssig21 = prmtr[4];

```

```

51  ssig22 = prmr[5];
52  Q = ssig2v; @ k by k @
53  R = ssig21~0|0~ssig22; @ 1 by 1 @
54  F = pphi; @ k by k @
55  Ht = 1\ggam; @ n by k @
56  beta_ll = zeros(k,1); @ starting value for beta_ll @
57  P_ll = ssig2v/(1-pphi^2); @ starting value for P_ll @
58  lnL = 0;

59  itr=1;
60  do until itr > t;
61      Beta_tl = F*beta_ll; @ k by 1 @
62      P_tl = F*P_ll*F' + Q; @ k by k @

63      eta_tl = ymat[itr,.]'-Ht*beta_tl; @ n by 1 @
64      f_tl = Ht*P_tl*Ht' + R; @ n by n @

65      lnL = lnL - 0.5*ln(2*pi*det(f_tl)) - 0.5*eta_tl'*inv(f_tl)*eta_tl;

66      Kt = P_tl*Ht'*inv(f_tl); @Kalman gain @
67      beta_tt= beta_tl + Kt*eta_tl;
68      P_tt = P_tl - Kt*Ht*P_tl;

69      beta_ll = beta_tt;
70      P_ll = P_tt;

71      itr = itr + 1;
72  endo;

73  retp(-lnL);
74  endp;

```

```

75  /*===== Filter =====*/
76  proc Filter(prmtr1);
77  local xmat,lnf, Ht,prmtr,Q,R,F,beta_ll,p_ll,lnL,ittr,pphi,
78  P_tl, eta_tl, f_tl, Kt, beta_tt, P_tt,beta_tl,
79  ggam,ssig21,ssig22,ssig2v;
80  prmtr = trans(prmtr1);
81  ggam = prmtr[1];
82  pphi = prmtr[2];
83  ssig2v = prmtr[3];
84  ssig21 = prmtr[4];
85  ssig22 = prmtr[5];
86  Q = ssig2v;
87  R = ssig21~0|0~ssig22; @ 1 by 1 @
88  F = pphi; @ k by k @
89  beta_ll = zeros(k,1); @ k by 1 @
90  P_ll = ssig2v/(1-pphi^2);
91  Ht = 1\ggam; @ n by k @
92  xmat = zeros(t,1);
93  lnL = 0;

94  ittr = 1;
95  do until ittr > t;
96      Beta_tl = F*beta_ll; @ k by 1 @
97      P_tl = F*P_ll*F' + Q; @ k by k @

98      eta_tl = ymat[ittr,.]' - Ht*beta_tl; @ n by 1 @
99      f_tl = Ht*P_tl*Ht' + R; @ n by n @

100     lnL = lnL - 0.5*ln(2*pi*det(f_tl)) - 0.5*eta_tl'*inv(f_tl)*eta_tl;

```

```

101     Kt = P_tl*Ht'*inv(f_tl);
102     beta_tt = beta_tl + Kt*eta_tl;
103     P_tt = P_tl - Kt*Ht*P_tl;

104     xmat[itr] = beta_tt;

105     beta_ll = beta_tt;
106     P_ll = P_tt;

107     itr = itr + 1;
108 endo;

109 retp(xmat);
110 endp;

111 /*===== Trans =====*/
112 proc trans(prmtr1);
113 local prmtr;
114 prmtr = prmtr1;
115 prmtr[2] = prmtr1[2]/(abs(prmtr1[2])+1); @ AR coefficient @
116 prmtr[3:5] = exp(prmtr1[3:5]); @ variances @
117 retp(prmtr);
118 endp;

```

14 Markov Switching Model

14.1 Background

Hamilton(1989)'s Markov switching models have drawn much attention in modeling regime shifts in dependent data. We consider the simplest case that the evolution of the discrete variable S_t follows a first-order two-state Markov-switching process governed by the following transition probabilities

$$\Pr [S_t = 1|S_{t-1} = 1] = p_{11}$$

$$\Pr [S_t = 0|S_{t-1} = 0] = p_{00}$$

For illustration, we focus on an AR(1) model, in which all parameters are subject to regime shift.

$$y_t - \mu_{S_t} = \phi(y_{t-1} - \mu_{S_{t-1}}) + e_t, \quad e_t \sim iidN(0, \sigma_{S_t}^2)$$

In this specification, AR coefficient, mean and conditional variance switch simultaneously depending on S_t . If S_t ($t = 1, 2, \dots, T$) is known a priori, then this model would be nothing but a dummy variable model. However, S_t and S_{t-1} are assumed to be unobserved. Therefore, in calculating the density, we consider the joint density of y_t , S_t and S_{t-1} .

The joint density of y_t , S_t and S_{t-1} conditioned on past information I_{t-1} is

$$f(y_t, S_t, S_{t-1}|I_{t-1}) = f(y_t|S_t, S_{t-1}, I_{t-1}) \Pr[S_t, S_{t-1}|I_{t-1}]$$

where

$$f(y_t|S_t, S_{t-1}, I_{t-1}) = \frac{1}{\sqrt{2\pi\sigma_{S_t}^2}} \exp\left(-\frac{1}{2} \frac{(y_t - \mu_{S_t} - \phi(y_{t-1} - \mu_{S_{t-1}}))^2}{\sigma_{S_t}^2}\right)$$

and $\Pr[S_t, S_{t-1}|I_{t-1}] = \Pr[S_t|S_{t-1}] \Pr[S_{t-1}|I_{t-1}]$.

To get the likelihood density $f(y_t|I_{t-1})$ we integrate out S_t and S_{t-1} out of the joint density by summing the joint density over all possible values of S_t and S_{t-1} .

$$f(y_t|I_{t-1}) = \sum_{S_t=0}^1 \sum_{S_{t-1}=0}^1 f(y_t, S_t, S_{t-1}|I_{t-1})$$

Then the log likelihood function is given by

$$\ln L = \sum_{t=1}^T \ln f(y_t|I_{t-1})$$

It is important to notice that calculation of the likelihood density requires $\Pr[S_{t-1}|I_{t-1}]$. Once y_t is observed at the end of time t , we can update the probability of S_t using the Bayes' rule in the following way.

$$\begin{aligned} \Pr[S_t = j, S_{t-1} = i|I_t] &= \Pr[S_t = j, S_{t-1} = i|I_{t-1}, y_t] \\ &= \frac{\Pr[S_t = j, S_{t-1} = i, y_t|I_{t-1}]}{f(y_t|I_{t-1})} \\ &= \frac{f(y_t|S_t = j, S_{t-1} = i, I_{t-1}) \Pr[S_t = j, S_{t-1} = i|I_{t-1}]}{f(y_t|I_{t-1})} \\ \Pr[S_t = j|I_t] &= \sum_{i=0}^1 \Pr[S_t = j, S_{t-1} = i|I_t] \end{aligned}$$

Repeating the above steps for $t=1, 2, \dots, T$ enables us to construct the likelihood sequentially. The starting probability of $S_t(= S_0)$ is given by the steady-state or

unconditional probability

$$\Pr[S_0 = 1] = \frac{1 - p_{00}}{2 - p_{11} - p_{00}}.$$

For the details, refer to Kim and Nelson (1999).

14.2 Flow Chart

Step 0> $\ln L = 0, t = 1, \Pr[S_0 = 1] = \frac{1-p_{00}}{2-p_{11}-p_{00}}$

Step 1> $\Pr[S_t, S_{t-1}|I_{t-1}] = \Pr[S_t|S_{t-1}] \Pr[S_{t-1}|I_{t-1}]$

Step 2> $f(y_t, S_t, S_{t-1}|I_{t-1}) = f(y_t|S_t, S_{t-1}, I_{t-1}) \Pr[S_t, S_{t-1}|I_{t-1}]$

Step 3> $f(y_t|I_{t-1}) = \sum_{S_t=0}^1 \sum_{S_{t-1}=0}^1 f(y_t, S_t, S_{t-1}|I_{t-1})$

Step 4> $\ln L = \ln L + \ln f(y_t|I_{t-1})$

Step 5> $\Pr[S_t = j, S_{t-1} = i|I_t] = \frac{\Pr[S_t=j, S_{t-1}=i, y_t|I_{t-1}]}{f(y_t|I_{t-1})}$

Step 6> $\Pr[S_t = j|I_t] = \sum_{i=0}^1 \Pr[S_t = j, S_{t-1} = i|I_t]$

Step 7> If $t < T$, then $t = t + 1$ and goto Step 1>

14.3 Code

```
1 new;  
2 library pgraph;  
3 cls;
```

```

4   T = 246;
5   load data[T,1] =c:\data\gdpgrowth.txt;
6   ymat = data*4; @ annual real GDP change in percent @

7   /* Initial value */
8   imu0 = 3;
9   imu1 = 3;
10  iphi0 = 0.5;
11  iphi1 = 0.5;
12  isig20 = 3;
13  isig21 = 0;
14  ip00 = 4;
15  ip11 = 10;
16  _qn_PrintIters = 1;
17  prmtr_in = imu0|imu1|iphi0|iphi1|isig20|isig21|ip00|ip11;
18  {xout,fout,gout,cout} = qnewton(ℰlik_f,prmtr_in);
19  xout_fnl = trans(xout);
20  cov = inv(hessp(ℰlik_f,xout));
21  grad = gradp(ℰtrans,xout);
22  cov_fnl = grad*cov*grad';
23  stde = sqrt(diag(cov_fnl));
24  t_val = xout_fnl./stde;
25  "_____";
26  ML estimates Standard errors t_values;
27  "_____";
28  xout_fnl~stde~t_val;

29  Prb1mat = filter(xout); @ Filtered probability of St=1 @
30  title(" filtered prb of St=1 ");

```

```

31  xy(t,prb1mat);
32  end;

33  /*===== Likelihood function =====*/
34  proc lik_f(prmtr0);
35  local prmtr,mu0,mu1,phi0,phi1,sig20,sig21,p00,p11,
36  pll0,pll1,lnL,itr,
37  ptl00,ptl10,ptl01,ptl11,
38  et00,et01,et10,et11,
39  pdf00,pdf10,pdf01,pdf11,
40  pdf,lnf,ptt00,ptt10,ptt01,ptt11,
41  ptt1,ptt0;

42  prmtr = trans(prmtr0);
43  mu0 = prmtr[1];
44  mu1 = prmtr[2];
45  phi0 = prmtr[3];
46  phi1 = prmtr[4];
47  sig20 = prmtr[5];
48  sig21 = prmtr[6];
49  p00 = prmtr[7];
50  p11 = prmtr[8];

51  pll0 = (1-p00)/(2-p11-p00); @ Unconditional Probability for St=0 @
52  pll1 = 1 - pll0; @ Unconditional Probability for St=1 @
53  lnL = 0;
54  itr = 2;
55  do until itr>t;

56  ptl00 = pll0*p00 ; @ Pr(S(t)=0,S(t-1)=1 | I(t-1) ) @

```

```

57     ptl01 = pll0*(1-p00) ; @ Pr(S(t)=0,S(t-1)=0 l I(t-1) ) @
58     ptl10 = pll1*(1-p11) ; @ Pr(S(t)=1,S(t-1)=0 l I(t-1) ) @
59     ptl11 = pll1*p11 ; @ Pr(S(t)=1,S(t-1)=1 l I(t-1) ) @

60     et00 = ymat[itr] - mu0 - phi0*(ymat[itr-1] - mu0);
61     et01 = ymat[itr] - mu1 - phi1*(ymat[itr-1] - mu0);
62     et10 = ymat[itr] - mu0 - phi0*(ymat[itr-1] - mu1);
63     et11 = ymat[itr] - mu1 - phi1*(ymat[itr-1] - mu1);

64     pdf00 = (1/sqrt(2*pi*sig20))*exp(-0.5*et00^2/sig20);
65     pdf01 = (1/sqrt(2*pi*sig21))*exp(-0.5*et01^2/sig21);
66     pdf10 = (1/sqrt(2*pi*sig20))*exp(-0.5*et10^2/sig20);
67     pdf11 = (1/sqrt(2*pi*sig21))*exp(-0.5*et11^2/sig21);

68     @ Probability Density Function for ymat[itr] @
69     pdf = pdf00*ptl00 + pdf01*ptl01 + pdf10*ptl10 + pdf11*ptl11;
70     lnf = ln(pdf);
71     lnL = lnL + lnf;

72     @ Probability update @
73     ptt00 = pdf00*ptl00/pdf;
74     ptt01 = pdf01*ptl01/pdf;
75     ptt10 = pdf10*ptl10/pdf;
76     ptt11 = pdf11*ptl11/pdf;

77     ptt1 = ptt01 + ptt11;
78     ptt0 = 1- ptt1;

79     pll0 = ptt0;
80     pll1 = ptt1;

```

```

81     itr = itr + 1;
82     endo;

83     retp(-lnL);
84     endp;

85     /*===== Procedure 2: Filtering =====*/
86     proc filter(prmtr0);
87     local prmtr,mu0,mu1,phi0,phi1,sig20,sig21,p00,p11,
88     pll0,pll1,prb1m,itr,
89     ptl00,ptl10,ptl01,ptl11,
90     et00,et01,et10,et11,
91     pdf00,pdf10,pdf01,pdf11,
92     pdf,lnf,ptt00,ptt10,ptt01,ptt11,
93     ptt1,ptt0;

94     prmtr = trans(prmtr0);
95     mu0 = prmtr[1];
96     mu1 = prmtr[2];
97     phi0 = prmtr[3];
98     phi1 = prmtr[4];
99     sig20 = prmtr[5];
100    sig21 = prmtr[6];
101    p00 = prmtr[7];
102    p11 = prmtr[8];

103    pll0 = (1-p00)/(2-p11-p00); @ Unconditional Probability for St=0 @
104    pll1 = 1 - pll0; @ Unconditional Probability for St=1 @
105    prb1m = zeros(T,1);
106    prb1m[1] = pll1;

```

```

107   itr = 2;
108   do until itr>t;

109       ptl00 = pll0*p00 ; @ Pr(S(t)=0,S(t-1)=1 l I(t-1) ) @
110       ptl01 = pll0*(1-p00) ; @ Pr(S(t)=0,S(t-1)=0 l I(t-1) ) @
111       ptl10 = pll1*(1-p11) ; @ Pr(S(t)=1,S(t-1)=0 l I(t-1) ) @
112       ptl11 = pll1*p11 ; @ Pr(S(t)=1,S(t-1)=1 l I(t-1) ) @

113       et00 = ymat[itr] - mu0 - phi0*(ymat[itr-1] - mu0);
114       et01 = ymat[itr] - mu1 - phi1*(ymat[itr-1] - mu0);
115       et10 = ymat[itr] - mu0 - phi0*(ymat[itr-1] - mu1);
116       et11 = ymat[itr] - mu1 - phi1*(ymat[itr-1] - mu1);

117       pdf00 = (1/sqrt(2*pi*sig20))*exp(-0.5*et00^2/sig20);
118       pdf01 = (1/sqrt(2*pi*sig21))*exp(-0.5*et01^2/sig21);
119       pdf10 = (1/sqrt(2*pi*sig20))*exp(-0.5*et10^2/sig20);
120       pdf11 = (1/sqrt(2*pi*sig21))*exp(-0.5*et11^2/sig21);

121       @ Final Probability Density Function for ymat[itr] @
122       pdf = pdf00*ptl00 + pdf01*ptl01 + pdf10*ptl10 + pdf11*ptl11;

123       @ At t = itr>1, to calculate conditional probability St=1 and St=0
124       using all information up to t-1 @
125       ptt00 = pdf00*ptl00/pdf;
126       ptt01 = pdf01*ptl01/pdf;
127       ptt10 = pdf10*ptl10/pdf;
128       ptt11 = pdf11*ptl11/pdf;

129       ptt1 = ptt01 + ptt11;
130       ptt0 = 1- ptt1;

```

```

131     prb1m[itr] = ptt1;

132     pll0 = ptt0;
133     pll1 = ptt1;

134     itr = itr + 1;
135     endo;

136     retp(prb1m);
137     endp;

139     /*===== Procedure 3:Trans =====*/
140     proc trans(prmtr1);
141     local prmtr2;

143     prmtr2 = prmtr1;
144     prmtr2[3:4] = prmtr1[3:4]./(1+abs(prmtr1[3:4])); @ phi's @
145     prmtr2[5:6] = exp(prmtr1[5:6]); @ sig2's @
146     prmtr2[7:8] = exp(prmtr1[7:8])./(1+exp(prmtr1[7:8])); @ transition prb @

148     retp(prmtr2);
149     endp;

```