Matlab Programming Help

Online Help:

1. http://www.mathworks.com/academia/student_center/tutorials/launchpad.html

2. http://www.math.ufl.edu/help/matlab-tutorial/

3. http://www.ag.unr.edu/moeltner/Matlab%20Tutorial/Matlab%20Tutorial.pdf

4. http://terpconnect.umd.edu/~nsw/ench250/for-mat.htm (Comparison between Matlab and Fortran f77)

Programming Basic

1. Input and output statement: output is written first.

Example 1:

X = Y + 1

Define X as Y+1.

X can be a scalar, vector, matrix. To do so, you have to define it first. For scalar case, you don't need to do so.

Code	Meaning
Y=1	Variable Y is defined as 1
Y=1+Y	Variable Y is defined as 1 + 1, so that Y becomes 2
X=Y+1	Variable X is defined as 2 + 1
Y = zeros(1,2)	Y becomes a 1x2 null vector. Y = (0, 0)
Y(1,1) =1 Y(1,2) =2	Y = (1,2)

2. Do Loop

Suppose that you want to add the sequence of numbers from 1 to 10. Let's program this.

Ex2. X = 1 + 2 + 3 + 4 + 5 + ... + 10

Sol 1: Write as

X =1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10

Sol 2: Write as

X = 1

X = X + 2

X = X + 3

...

X = X + 10

Sol 3: Use "For" statement

X = 0; For i = 1:10; X = X + i; End;	Assign 0 to X. Need ";" to continue program it. Start Do loop. First assign i to be 1, and increase it by 1 up to 10. X becomes X + i, Repeat this until i = 10

Let X = [1 3 4 2 2 2 4 1 23 5]; That is, X is a 10x1 vector. Calculate its mean by using For statement

Z = 0; For i = 1:10; Z = Z + X(i); End;	Assign 0 to Z. Start Do loop. First assign i to be 1, and increase it by 1 up to 10. Z becomes Z + X(i), Repeat this until i = 10

Exercise: For Statement

1. Add 1 through 100

2. Multiply 1 through 20

3. 2 x 4 x 6 x 8 x ... x 20

4. X(1)xY(1) + X(2)xY(2) + ... + X(n)xY(n)

3. IF Statement

Format:

IF condition statement End

Example: X = [1 -2 3 -4]

We want to change X to index such that Y = 0 if X >0, Y=1 o.w.

```
Y = X;
For i = 1:4;
 if X(i) > 0; Y(i) = 0; end;
 if X(i) < 0; Y(i) = 1; end;
end;
```

Exercise: IF Statement

1. X = [1 3 4 8]. Find the maximum of X.

2. Find the minimum of X

3. Sort X.

4. Data (Matrix & Vector) Modification

Ex: A = [1 2; 3 4] implies A = 1 2
                                3 4

Type the following commends.

1. A’
2. sum(A)
3. sum(A’)
4. sum(A’)’
5. diag(A)
6. sum(diag(A))
7. inv(A)
8. A(1,2)
9. A(1,1)
10. A(2,1)
11. A(:,1)
12. A(1,:)
13. A(:)
14. A(:,end)

Expression

.* element by element product
./ element by element division
\   inverse
.^ element by element power

Type A = [1 2; 3 4]; B = [1 1; 2 3];

1. a
2. A
3. b
4. B
5. A.*B
6. A./B
7. A\B
8. Inv(A)*B
9. A’*A
10. B’*B
11. Inv(A).*A
12. Inv(A)*A

Ex b = [2 3]. You want to calculate A – b = 1-2 2-3
                                            3-2 4-3

Important Functions

Type A = [-3 4; 1 3;2 2]

1. a

2. A
3. mean(A)
4. sum(A)
5. sort(A)
6. [B,id] = sort(A)
7. max(A)
8. min(A)
9. std(A)
10. var(A)
11. cov(A)
12. abs(A)

5. Function Statement

Format

Function output = functionname(inputs)

Example: Average

Function y = mymean(x)

t = length(x);

y = 0;

for i = 1:t;

y = y + x(i);

end;

Then in the main program, you can recall `mymean'.

z = mymean(x);

In Fortran, this function statement is called as `subroutine' program.

In Gauss, it is called as `proc' program.

Matlab library contains many function statements.

Assignment 1: Download X and Y variables from the class homepage.

A. Sort X from smallest to largest

B. Sort Y from largest to smallest

C. Calculate mean and variance of X and Y.

D. Calculate correlation between X and Y

E. Make functions (mymean, myvar, mycorr) and use them to calculate C,D and E.

F. Program OLS function.

Input = x and y. both them are Tx1 vectors.

Regression: y = bx + u.

output:

b = inv(x'x)*x'y

R^2,

ordinary t-value.

Function [b,r2,tb] = myols(y,x)

Lecture 2: OLS & GLS

Cross section or Time series data

Model y = a + X*b + u

Where X is a matrix (nxk, k is number of regressors), a is scalar, b is a vector.

Define a vector such that

```
n = length(y);
a = ones(n,1);
```

Next, define a matrix such that

```
Z = [a X];
[n,k] = size(Z);
```

OLS estimator:

```
bhat = inv(Z'*Z)*Z'*y;
   1. Regression residuals: uhat = y – Z*bhat;
         a. t-ratio needs variance of bhat:
               i.  IID case:
                   sigma = uhat.*uhat;
                   sigma = sum(sigma)/(n-k);
                   sigma = sigma*inv(Z'*Z);
                   sigma = diag(sigma);
              ii.  IDIN case:
                   uuhat = Z.*repmat(uhat,1,2);
                   sigma = uuhat'*uuhat;
                   sigma = sum(sigma)/(n-k);
                   sigma = inv(Z'*Z)*sigma*inv(Z'*Z)*n;
         b. R-squares Rbar-squares:
```

Example file: ex2.m

```
n = 100;
y = randn(n,1);
x = randn(n,1);

z = [ones(n,1) x];
b = inv(z'*z)*z'*y;
u = y - z*b;

sig1 = u'*u/(n-2);
sig1 = sig1*inv(z'*z);

h = z.*repmat(u,1,2);
sig2 = h'*h/(n-2);
sig2 = inv(z'*z)*sig2*inv(z'*z)*n;

tra1 = b./sqrt(diag(sig1));
tra2 = b./sqrt(diag(sig2));

[tra1 tra2]
```

GLS Estimator: AR(1) coefficient case

```
n = 100;
y = randn(n,1);
x = randn(n,1);

z = [ones(n,1) x];
b = inv(z'*z)*z'*y;
u = y - z*b;

% estimation of AR(1) coefficient
uy = u(2:n); ux = u(1:n-1);
rho = inv(ux'*ux)*ux'*uy; % or equivalently rho = sum(ux.*uy)/sum(ux.*ux)
e = uy-ux*rho;
ve = var(e);

% constructing co-variance and variance matrix

omega = eye(n).*ve./(1-rho^2);
for i = 1:n;
    for j = i+1:n;
        omega(i,j) = rho^(j-i);
        omega(j,i) = omega(i,j);
    end;
end;

% Cholesky Decomposition

P = chol(omega);

% Pre-multiplying P matrix

ys = P*y;
zs = P*z;

c = inv(zs'*zs)*zs'*ys;

% variance matrix for c

vc = inv(zs'*zs);

% t-values

tra = c./sqrt(diag(vc));

tra
```

Pooled OLS and LSDV

```
clear;
t=2;
n=10;

y = randn(t,n);
x = randn(t,n);
% LSDV
a = ones(t,1);
a = kron(eye(n),a);

vx = x(:);
z = [a vx];
b = inv(z'*z)*z'*y(:);
b

% POLS

a = ones(t,1);
a = repmat(a,1,n);
a = a(:);
z = [a vx];
b = inv(z'*z)*z'*y(:);
b
```

Assignment 3:

A. Suppose that you want to program the following regressions

$$y(it) = a_i + cX(it) + u(it)$$

1. Input must be y and x where X is a nxk matrix
2. Output must include point estimates, their standard errors (ordinary one, panel robust one), r-bar squares etc.

B. Suppose that you want to program the following regressions

$$Y(it) = a_i + bz(i) + cX(it) + u(it)$$

3. Make function for LSDV and POLS
4. Output must include point estimates, their standard errors (ordinary one, panel robust one), r-bar squares etc.