

**Basic OLS Estimates in Matlab**  
**ECON 60303**

**Bill Evans**  
**Spring 2011**

**Data**

The economic problem is the standard “human capital earnings function” that examines the relationship between measure of acquired skill (education and age) and the natural log of weekly earnings. The data is from the 1987 Current Population Survey and the information is for full time male workers aged 21-64. The data is stored in an Excel data file titled cps\_wage\_data.xlsx, a file format that can be read directly into Matlab. The excel file has 8 columns and data on over 19,000 male workers. The first column is the dependent variable, the natural log of usual weekly earnings. The next seven columns are the independent variables including a constant. The other covariates include age (in years) and its square, years of completed education, plus three dummy variables: union (equals 1 if the workers is a member of a union, equals 0 otherwise), black (equals 1 if the responded is African American, equals zero otherwise) and other\_race (equals 1 if the responded is a race other than white or black, equals zero otherwise). The reference group is white workers. You are to run a regression with these six variables as covariates plus a constant. The first 10 observations from the Excel data file are reproduced below.

ln_weekly_earn	constant	age	age2	years_educ	union	black	other_race
5.594711	1	38	1444	17	0	0	0
6.214608	1	49	2401	14	0	0	0
5.298317	1	26	676	14	0	0	0
6.086775	1	31	961	16	0	0	0
6.869014	1	26	676	16	0	0	0
6.139884	1	38	1444	12	1	0	0
5.192957	1	21	441	15	0	0	0
6.694562	1	55	3025	18	0	0	0
5.828946	1	44	1936	12	0	0	0

On the web page, there is a comma delimited data set (cps\_wage.data.csv) that you can load into stata if you want to know what answers you should get.

The attached Matlab program

- Loads the data into Matlab from an excel file
- Constructs the X matrix and the Y vector
- Produces the basic OLS estimates and other descriptive information
- Outputs the results to a text file.

Although Matlab has a built-in function that will produce all the OLS results, I have written code that produced

$$\hat{\beta} = (x'x)^{-1}x'y, \hat{\varepsilon} = y - x\hat{\beta}, \hat{\sigma}_\varepsilon^2 = \hat{\varepsilon}'\hat{\varepsilon} / (n - k), R^2 = 1 - SSE/SST, \text{ etc.}$$

The statistics produced include

- The name and the mean of the dependent variable
- The number of observations, covariates and degrees of freedom
- The SST, SSM, SSE,  $R^2$ , MSE ( $\hat{\sigma}_\varepsilon^2$ ), and root MSE ( $\hat{\sigma}_\varepsilon$ )

- The F-test on the null hypothesis that the non-constants in the model are all zero, the degrees of freedom for this test and the p-value for the null hypothesis.
- The name of each covariate, parameter estimates ( $\hat{\beta}$ ), standard errors, plus the t-statistic and p-value for the null hypothesis that the parameter estimates equal zero.

I spent some time producing a set of code that will print out the results in nice neat rows. Feel free to steal this code for future problem sets.

### Program: `ols_program_m`

```

%% load in the data from cps87.csv
%% column 1 is the dependent variable while
%% columns 2-k are independent variables
%% the constant is not in the data set
[w,varlist]=xlsread('cps_wage_data.xlsx');

dv=varlist{1,1};
%% get dimension of w
nk1=size(w);

% number of observations
n=nk1(1);

% number of independent variables
% k+1 is the no of columns of w. there are k
% covariates (including the constant)
k=nk1(2)-1;

%% extract y which is in the 1st column of w
y=w(:,1);

%% extract x which is in columns 2 through k
x=w(:,2:(k+1));

xpxi=inv(x'*x);
beta=xpxi*x'*y;
e=y-x*beta;
sse=e'*e;

%% perform analysis of covariance
dof=n-k;
s2=sse/dof;
rmse=sqrt(s2);
ym=sum(y)/n;
sst=(y-ym)'+(y-ym);
ssm=sst-sse;
r2=ssm/sst;

%% ftest all coefficients equal zero
ftest=((sst-sse)/(k-1))/s2;
pvaluef=1-fcdf(ftest,(k-1),dof);
kml=k-1;
%% get variance of beta, t-stat and p-values
stderr=sqrt(diag(s2*xpxi));

```

```

tstat=beta./stderr;
pvalue=2*(1-tcdf(abs(tstat),(n-k)));

file1=fopen('ols_program.txt','w');
fprintf(file1,'-----\n');
fprintf(file1,'Ordinary Least Square Estimates \n');
fprintf(file1,'-----\n');
fprintf(file1,'Dependent variable = %s \n' ,dv);
fprintf(file1,'Mean of dep variable = %9.4f \n' ,ym);
fprintf(file1,'-----\n');
fprintf(file1,'analysis of covariance \n');
fprintf(file1,'\n');
fprintf(file1,'observations = %9.f \n' ,n);
fprintf(file1,'parameters = %9.f \n' ,k);
fprintf(file1,'degrees of freedom = %9.f \n' ,dof);
fprintf(file1,'sse = %9.4f \n' ,sse);
fprintf(file1,'ssm = %9.4f \n' ,ssm);
fprintf(file1,'sst = %9.4f \n' ,sst);
fprintf(file1,'rsquared = %9.4f \n' ,r2);
fprintf(file1,'mse = %9.4f \n' ,s2);
fprintf(file1,'rmse = %9.4f \n' ,rmse);

fprintf(file1,'F-test Ho: all betas = 0, F(%6.f,%9.f)=%9.4f \n',kml,dof,ftest);
fprintf(file1,'P-value on the f-test =%10.8f \n', pvaluef);
c1='Covariate'; c2='Beta'; c3='Std Error'; c4='T-statistic'; c5='P-value: B=0';
fprintf(file1,'-----\n');
fprintf(file1,'%12s %12s %12s %12s %12s \n', c1,c2,c3,c4,c5);
fprintf(file1,'-----\n');
for i=1:k;
    rowname=varlist{1,i+1};
    fprintf(file1,'%12s %12.6f %12.6f %12.6f %12.6f \n',
rowname,beta(i,1),stderr(i,1),tstat(i,1),pvalue(i,1));
    end;
fprintf(file1,'-----\n');

fclose(file1);

```

**Results: ols\_program.txt**

-----  
Ordinary Least Square Estimates  
-----

Dependent variable = ln\_weekly\_earn  
Mean of dep variable = 6.0673  
-----

analysis of covariance

observations = 19906  
parameters = 7  
degrees of freedom = 19899  
sse = 3615.7878  
ssm = 1623.5512  
sst = 5239.3391  
rsquared = 0.3099  
mse = 0.1817  
rmse = 0.4263  
F-test Ho: all betas = 0, F( 6, 19899)=1489.1658  
P-value on the f-test =0.00000000  
-----

Covariate	Beta	Std Error	T-statistic	P-value: B=0
constant	3.619195	0.039418	91.815559	0.000000
age	0.067930	0.002001	33.941169	0.000000
age2	-0.000677	0.000024	-27.665524	0.000000
years_educ	0.070061	0.001132	61.864795	0.000000
union	0.131571	0.007289	18.051343	0.000000
black	-0.212975	0.011079	-19.223933	0.000000
other_race	-0.109640	0.013299	-8.244491	0.000000

-----