```
// Feb 1, 2020 Gauss Tutorial
// Nmae: Jikhan Jeong
// Ref: https://www.aptech.com/resources/tutorials/econometrics/linear-regression/

// How to run code 1. command line 2. click right on mouse, it shows "run current line = F4"



//_____________________* OLS Basic_____________________________________________ //
// Clear the workspace
new;

// Set seed to replicate results
rndseed 23423;

// Number of observations

num_obs = 100;

// Generate x ~ N(0,1), with 'num_obs' rows and 1 column

x = rndn(num_obs,1);

// Compute 100 observations of an error term ~ N(0,1)

error_term = rndn(num_obs,1);


// Simulate our dependent variable

y = 1.3 + 5.7*x + error_term;


// The tilde operator, ~, horizontally concatenates two matrices or vectors into one larger matrix.

x_matrix = ones(num_obs, 1) ~ x;

//Compute OLS estimates, using matrix operations

beta_hat = inv(x_matrix'x_matrix)*(x_matrix'y);

print beta_hat;


call ols("", y, x);



//_____________________* Gauss Quick Reference_____________________________ //
// Ref: https://www.aptech.com/resources/gauss-quick-reference/



// 1. Matrix creation

x = {1 2, 3 4};
print x;
```

```
58
59  x = seqa(2,2,20); // creates a sequence of 'count' numbers starting at 'start' and
    increasing by 'step'.
60  print x;
61
62  x = zeros(2,3);   // creates an 'm' by 'n' matrix with all elements set to 0.
63  print x;
64
65  x = ones(2, 3);   // creates an 'm' by 'n' matrix with all elements set to 1.
66  print x;
67
68  x = rndn(2, 3);   // creates an 'm' by 'n' matrix of random normal numbers.
69  print x;
70
71  x = rndu(2, 3);   // creates an 'm' by 'n' matrix of uniformly distributed random
    numbers.
72  print x;
73
74
75  // 2. Matrix manipulation
76
77  x = rndn(2, 3);   // creates an 'm' by 'n' matrix of random normal numbers.
78  print x;
79
80  y = rndn(2, 3);   // creates an 'm' by 'n' matrix of random normal numbers.
81  print y;
82
83
84  a = x[1,3];       // extract the element of 'x' located at 'row:col'.
85  print a;
86
87  a = x[.,3];       // extract all rows of the specified column(s) of 'x'.
88  print a;
89
90  a = x[1:2,.];     // extract all columns from the row range 'row_start' to 'row_end'.
91  print a
92
93
94  a = x[2, 1 2];    // horizontally concatenate 'x' and 'y'.
95  print a;
96
97  a = x~y;          // vertically concatenate 'x' and 'y'.
98  print a;
99
100 a = reshape(a, 3,4);
101 print a
102
103
104 // 3. Operators
105
106 // 3.1 Element-by-element (ExE) operators
107
108 x;
109 y = ones(3,2);
110 print y;
111
112 z = x.*y'; // dimension must be the same due to ExE
113 print z;
114
115 z = x ./y'; // Element-by-element divide.
116 print z;
```

```
117
118   z = x.^y';  // Element-by-element exponentiation.
119   print z;
120
121   z = x+y ;   // Element-by-element addition.
122   print z;
123
124   z = x-y ;   // Element-by-element subtraction.
125   print z;
126
127
128   // 3.2 Matrix operators
129
130   z = x * y;  // Matrix multiply.
131   print z;
132
133   z = x .*.y; // Kronecker product.
134   print z;
135
136   Matrix transpose.z = x';     // Matrix transpose.
137   print z;
138
139
140
141
142
```