

Generate a Table and Fill with Data Row by Row or Random Data

back to [Fan's Intro Math for Econ](#), [Matlab Examples](#), or [MEconTools Repositories](#)

An Empty Table, Filled with Loop Row by Row

First, generate an empty table.

```
% Make N by 2 matrix of fieldname + value type
mt_st_variable_names_types = [{"category", "string"}; ...
    ["wage", "double"]; ...
    ["skilled", "logical"]; ...
    ["labdemand", "double"]; ...
    ["labsupply", "double"]; ...
    ["labsupplyprob", "string"]; ...
    ["rho_manual", "double"]; ...
    ["rho_routine", "double"]; ...
    ["rho_analytical", "double"]; ...
];
% Make table using fieldnames & value types from above
tb_equilibrium = table('Size',[0,size(mt_st_variable_names_types,1)],...
    'VariableNames', mt_st_variable_names_types(:,1),...
    'VariableTypes', mt_st_variable_names_types(:,2));
% display table
disp(size(tb_equilibrium));
```

0 9

Second, over a loop, fill the table with values row by row.

```
for it_rho_set=[1,2,3]
    if (it_rho_set == 1)
        cl_data_row = {'C001',1, true,1,1,0.5,0.5,0.5,0.5};
    elseif (it_rho_set == 2)
        cl_data_row = {'C002',1, 0 ,1.2,0.6,0.5,0.45,0.5,0.5};
    elseif (it_rho_set == 3)
        cl_data_row = {'C011',1.1,false,1.2,0.6,0.5,0.45,0.45,0.45};
    end
    tb_equilibrium = [tb_equilibrium; cl_data_row];
end
disp(tb_equilibrium);
```

category	wage	skilled	labdemand	labsupply	labsupplyprob	rho_manual	rho_routine	rho_analy
"C001"	1	1	1	1	"0.5"	0.5	0.5	0.5
"C002"	1	0	1.2	0.6	"0.5"	0.45	0.5	0.5
"C011"	1.1	0	1.2	0.6	"0.5"	0.45	0.45	0.45

Data Assignment to Table Cell Value

Given an existing table, we might want to replace values in specific tables cells. There are two main methods to accomplish as shown below. If the string name is known, the TAB.COL(ROW) replacement method is

dramatically faster. Test speed with timeit first loading the value in table cell via method a vs b, method a takes 20 to 30 times more time than method b. Then test assignment with tic toc, with about a 7 time speed difference.

```
% Two replacement functions
f_replace_method_a = @() tb_equilibrium{1, "labdemand"};
f_replace_method_b = @() tb_equilibrium.labdemand(1);

% Time replacing one value
fl_speed_method_a = timeit(f_replace_method_a);
fl_speed_method_b = timeit(f_replace_method_b);
fl_speed_a_b_ratio = fl_speed_method_a/fl_speed_method_b;
disp(['Load table cell time, fl_speed_a_b_ratio=' num2str(fl_speed_a_b_ratio)]);
```

Load table cell time, fl_speed_a_b_ratio=18.5554

```
% Timing assignment with Method A
ar_rand = rand([1,1e4]);
fl_time_start = tic;
for (fl_rand=ar_rand)
    tb_equilibrium{1, "labdemand"} = fl_rand;
end
fl_time_end = toc(fl_time_start);
disp(['Method A assignment fl_time_end = ' num2str(fl_time_end)]);
```

Method A assignment fl_time_end = 4.5898

```
% Timing assignment with Method B
fl_time_start = tic;
for (fl_rand=ar_rand)
    tb_equilibrium.labdemand(1) = fl_rand;
end
fl_time_end = toc(fl_time_start);
disp(['Method B assignment fl_time_end = ' num2str(fl_time_end)]);
```

Method B assignment fl_time_end = 0.70196

Generate a Table with M Variables of Random Data

Generate a numeric table with random varlues and a string column

```
% Numeric Matrix
it_num_cols = 4;
it_num_rows = 5;
mt_data = rand([it_num_rows, it_num_cols]);

% Generate Table
tb_test = array2table(mt_data);

% Generate Row and Column Names
cl_col_names = strcat('col_', string((1:it_num_cols)));
cl_row_names = strcat('row_', string((1:it_num_rows)));
```

```
tb_test.Properties.VariableNames = matlab.lang.makeValidName(cl_col_names);
tb_test.Properties.RowNames = matlab.lang.makeValidName(cl_row_names);
```

```
% Generate two string variable
```

```
rng(456);
cl_st_var1 = strcat('data=', string(rand([it_num_rows,1])));
cl_st_var2 = strcat('data=', string(rand([it_num_rows,1])));
tb_test = addvars(tb_test, cl_st_var1, cl_st_var2);
```

```
% Display Table
```

```
disp(tb_test);
```

	col_1	col_2	col_3	col_4	cl_st_var1	cl_st_var2
row_1	0.74528	0.078694	0.25903	0.90658	"data=0.24876"	"data=0.60411"
row_2	0.042619	0.46482	0.53907	0.45765	"data=0.16307"	"data=0.8857"
row_3	0.99381	0.15993	0.23508	0.47747	"data=0.78364"	"data=0.75912"
row_4	0.69712	0.56955	0.13719	0.69918	"data=0.80852"	"data=0.18111"
row_5	0.85195	0.47611	0.064861	0.49792	"data=0.62563"	"data=0.15017"