

Row and Column Combine Stack Tables and Matrices

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

Generate Some Tables and Matrixes for Combination

```
close all;

% Generate Table 1
ar_fl_abc1 = [0.4 0.1 0.25 0.3 0.4];
ar_fl_abc2 = [0.4 0.1 0.2 0.3 0.4];
number1 = '123';
number2 = '456';
mt_data_a = [ar_fl_abc1' ar_fl_abc2'];

tb_test_a = array2table(mt_data_a);
cl_col_names_a = {'col' num2str(number1)}, ['col' num2str(number2)]};
cl_row_names_a = strcat('rowA=', string((1:size(mt_data_a,1))));

tb_test_a.Properties.VariableNames = cl_col_names_a;
tb_test_a.Properties.RowNames = cl_row_names_a;
disp(tb_test_a);
```

	col123	col456
rowA=1	0.4	0.4
rowA=2	0.1	0.1
rowA=3	0.25	0.2
rowA=4	0.3	0.3
rowA=5	0.4	0.4

```
% Generate Table 2
rng(123);
ar_fl_abc3 = rand(size(ar_fl_abc1));
ar_fl_abc4 = rand(size(ar_fl_abc1));
ar_fl_abc5 = rand(size(ar_fl_abc1));

mt_data_b = [ar_fl_abc3' ar_fl_abc4' ar_fl_abc5'];

tb_test_b = array2table(mt_data_b);
cl_col_names_b = {'col' num2str(33)}, ['col' num2str(44)], ['col' num2str(55)]};
cl_row_names_b = strcat('rowB=', string((1:size(mt_data_a,1))));

tb_test_b.Properties.VariableNames = cl_col_names_b;
tb_test_b.Properties.RowNames = cl_row_names_b;
disp(tb_test_b);
```

	col33	col44	col55
rowB=1	0.69647	0.42311	0.34318
rowB=2	0.28614	0.98076	0.72905
rowB=3	0.22685	0.68483	0.43857
rowB=4	0.55131	0.48093	0.059678

rowB=5 0.71947 0.39212 0.39804

Combine Tables Together Stack Columns

Tables with the same number of rows, add more columns with named variables

```
% a and b must have the same row names
tb_test_b_withArownames = tb_test_b;
tb_test_b_withArownames.Properties.RowNames = tb_test_a.Properties.RowNames;
tb_ab_col_stacked = [tb_test_a tb_test_b_withArownames];
disp(tb_ab_col_stacked);
```

	col123	col456	col133	col144	col155
rowA=1	0.4	0.4	0.69647	0.42311	0.34318
rowA=2	0.1	0.1	0.28614	0.98076	0.72905
rowA=3	0.25	0.2	0.22685	0.68483	0.43857
rowA=4	0.3	0.3	0.55131	0.48093	0.059678
rowA=5	0.4	0.4	0.71947	0.39212	0.39804

Combine Tables Together Stack Rows

Tables with the same number of columns, add more rows variables

```
% Select only 2 columns to match table a column count
tb_test_b_subset = tb_test_b(:,1:2);

% Make Column Names consistent
tb_test_b_subset.Properties.VariableNames = cl_col_names_a;

% Reset Row Names, can not have identical row names
tb_test_a.Properties.RowNames = strcat('row=', string((1:size(mt_data_a,1))));
tb_test_b_subset.Properties.RowNames = ...
    strcat('row=', string(((size(mt_data_a,1)+1):(size(mt_data_a,1)+size(tb_test_b_subset,1)))));
% tb_test_b_subset.Properties.RowNames =

% Stack Rows
tb_ab_row_stacked = [tb_test_a; tb_test_b_subset];
disp(tb_ab_row_stacked);
```

	col123	col456
row=1	0.4	0.4
row=2	0.1	0.1
row=3	0.25	0.2
row=4	0.3	0.3
row=5	0.4	0.4
row=6	0.69647	0.42311
row=7	0.28614	0.98076
row=8	0.22685	0.68483
row=9	0.55131	0.48093
row=10	0.71947	0.39212

ND Dimensional Parameter Arrays, Simulate Model and Stack Output Tables

Now we will first column combine matrixes, model parameters and model outcomes, and then row combine matrixes from different simulations.

A model takes a N parameters, solve the model over M sets of parameters. Each time when the model is solved, a P by Q table of results is generated. Each column is a different statistics (mean, std, etc.), and each row is a different outcome variable (consumption, asset choices, etc.). Stack these P by Q Tables together, and add in information about the N parameters, each of the tables been stacked initially had the same column and row names.

The resulting table should have P times M rows, for M sets of model simulations each with P rows of results. And there should be N + Q columns, storing the N parameters as well as the Q columns of different outcomes.

```
rng(123);
% Generate A P by Q matrix of random parameter Values
it_param_groups_m = 5;
it_params_n = 2;
it_outcomes_p = 3;
it_stats_q = 3;

% Parameter Matrix and Names
ar_param_names = strcat('param_', string(1:it_params_n));
mt_param_m_by_n = round(rand([it_param_groups_m, it_params_n])*5, 2);

% Loop over the parameters
for it_cur_param_group=1:1:it_param_groups_m

    % Current Parameters
    ar_param = mt_param_m_by_n(it_cur_param_group,:);

    % Some Model is simulated
    mt_model_simu = normrnd(mean(ar_param), std(ar_param), [it_outcomes_p, it_stats_q]);

    % Model Results are Saved As Table With Column and Row Information
    tb_model_simu = array2table(mt_model_simu);
    cl_col_names = strcat('stats_', string((1:size(mt_model_simu,2))));
    cl_row_names = strcat('outvar_', string((1:size(mt_model_simu,1))));
    tb_model_simu.Properties.VariableNames = cl_col_names;
    tb_model_simu.Properties.RowNames = cl_row_names;

    % Convert Row Variable Names to a Column String
    outvar = string(tb_model_simu.Properties.RowNames);
    tb_model_simu = addvars(tb_model_simu, outvar, 'Before', 1);

    % Parameter Information Table that Shares Row Names as Simu Results
    mt_param_info = zeros([it_outcomes_p,it_params_n]) + ar_param;
    tb_param_info = array2table(mt_param_info);
    tb_param_info.Properties.VariableNames = ar_param_names;
    tb_param_info.Properties.RowNames = cl_row_names;

    % Combine Parameter Information and Simulation Contents
    tb_model_simu_w_info = [tb_param_info tb_model_simu];
    % Update Row Names based on total row available
    ar_rows_allsimu = (1:it_stats_q)' + (it_cur_param_group-1)*it_stats_q;
```

```

tb_model_simu_w_info.Properties.RowNames = strcat('row=', string(ar_rows_allsimu));

% Show One Example Table before Stacking
if (it_cur_param_group == round(it_param_groups_m/2))
    disp(tb_model_simu);
    disp(tb_param_info);
    disp(tb_model_simu_w_info);
end

% Stack all results
if(it_cur_param_group == 1)
    tb_model_allsimu_w_info = tb_model_simu_w_info;
else
    tb_model_allsimu_w_info = [tb_model_allsimu_w_info; tb_model_simu_w_info];
end

end

```

	outvar	stats_1	stats_2	stats_3		
outvar_1	"outvar_1"	0.056853	2.1703	2.1098		
outvar_2	"outvar_2"	3.1545	2.0634	0.7798		
outvar_3	"outvar_3"	-0.49033	2.2566	1.7896		
	param_1	param_2				
outvar_1	1.13	3.42				
outvar_2	1.13	3.42				
outvar_3	1.13	3.42				
	param_1	param_2	outvar	stats_1	stats_2	stats_3
row=7	1.13	3.42	"outvar_1"	0.056853	2.1703	2.1098
row=8	1.13	3.42	"outvar_2"	3.1545	2.0634	0.7798
row=9	1.13	3.42	"outvar_3"	-0.49033	2.2566	1.7896

Show all Simulation Joint Table Outputs:

```
disp(tb_model_allsimu_w_info);
```

	param_1	param_2	outvar	stats_1	stats_2	stats_3
row=1	3.48	2.12	"outvar_1"	2.2665	1.1885	1.924
row=2	3.48	2.12	"outvar_2"	3.3427	2.4647	2.3548
row=3	3.48	2.12	"outvar_3"	2.6714	3.6132	2.918
row=4	1.43	4.9	"outvar_1"	3.3859	5.3759	1.5816
row=5	1.43	4.9	"outvar_2"	3.9499	3.8698	2.2693
row=6	1.43	4.9	"outvar_3"	5.7745	4.6871	1.7334
row=7	1.13	3.42	"outvar_1"	0.056853	2.1703	2.1098
row=8	1.13	3.42	"outvar_2"	3.1545	2.0634	0.7798
row=9	1.13	3.42	"outvar_3"	-0.49033	2.2566	1.7896
row=10	2.76	2.4	"outvar_1"	2.9611	2.6847	2.4986
row=11	2.76	2.4	"outvar_2"	2.9333	2.3457	3.0629
row=12	2.76	2.4	"outvar_3"	2.5814	2.4372	2.4806
row=13	3.6	1.96	"outvar_1"	2.7199	3.3129	3.0577
row=14	3.6	1.96	"outvar_2"	3.9804	1.4529	2.9285
row=15	3.6	1.96	"outvar_3"	2.8445	4.4117	2.6576