

# FF\_SIMU\_STATS Examples

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

This is the example vignette for function: `ff_simu_stats` from the [MEconTools Package](#). This is a gate-way function that computes mean, percentiles, covariance etc between several variables.

## Test FF\_SIMU\_STATS Defaults

Call the function with defaults.

```
ff_simu_stats();
```

```
xxx tb_outcomes: all stats xxx
```

OriginalVariableNames	cl_mt_pol_a	cl_mt_pol_c
{'mean' }	-0.11081	8.8423
{'sd' }	4.1239	6.5845
{'coefofvar' }	-37.215	0.74466
{'min' }	-7	-6.3772
{'max' }	9	21.786
{'pYls0' }	0.064259	0
{'pYls0' }	0.54867	0.027329
{'pYgr0' }	0.38707	0.97267
{'pYisMINY' }	0.051764	0.015232
{'pYisMAXY' }	0.027329	0.046484
{'p1' }	-7	-6.3772
{'p10' }	-6	0.27238
{'p25' }	-3	5.2138
{'p50' }	-1	6.5321
{'p75' }	3	13.799
{'p90' }	5	16.887
{'p99' }	9	21.786
{'fl_cov_cl_mt_pol_a' }	17.007	-22.084
{'fl_cor_cl_mt_pol_a' }	1	-0.81327
{'fl_cov_cl_mt_pol_c' }	-22.084	43.356
{'fl_cor_cl_mt_pol_c' }	-0.81327	1
{'fracByP1' }	3.2699	-0.010985
{'fracByP10' }	5.9889	-0.013362
{'fracByP25' }	14.165	0.041007
{'fracByP50' }	16.208	0.1893
{'fracByP75' }	12.702	0.59539
{'fracByP90' }	6.6611	0.8307
{'fracByP99' }	1	1

## Test FF\_SIMU\_STATS Four States-Points Matrix

Over some (a,z) states that is 3 by 3, c matrix, generate all stats

```
% Set Parameters
mt_x_of_s = [1, 2, 3.0;...
            3, 1, 1.5;...
            4, 3, 2.0];
mt_y_of_s = [2, -10, 9.0;...
            5, 1.1, 3.0;...
            1, 3, -1.5];
mt_z_of_s = [1.1, 2, 3.3;...
```

```

2.3, 1,1.5;...
4, 2.5,2.0];
mp_cl_mt_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
mp_cl_mt_xyz_of_s('cl_mt_x_of_s') = {mt_x_of_s, zeros(1)};
mp_cl_mt_xyz_of_s('cl_mt_y_of_s') = {mt_y_of_s, zeros(1)};
mp_cl_mt_xyz_of_s('cl_mt_z_of_s') = {mt_z_of_s, zeros(1)};
mp_cl_mt_xyz_of_s('ar_st_y_name') = ["cl_mt_x_of_s", "cl_mt_y_of_s", "cl_mt_z_of_s"];
% Mass
rng(123);
mt_f_of_s = rand(size(mt_x_of_s));
mt_f_of_s = mt_f_of_s/sum(mt_f_of_s, 'all');
% Call Function
mp_cl_mt_xyz_of_s_out = ff_simu_stats(mt_f_of_s, mp_cl_mt_xyz_of_s);

```

xxx tb\_outcomes: all stats xxx

OriginalVariableNames	cl_mt_x_of_s	cl_mt_y_of_s	cl_mt_z_of_s
{ 'mean' }	2.0763	1.9323	2.0668
{ 'sd' }	0.9071	5.2239	0.9042
{ 'coefofvar' }	0.43688	2.7034	0.43749
{ 'min' }	1	-10	1
{ 'max' }	4	9	4
{ 'pYls0' }	0	0	0
{ 'pYls0' }	0	0.20441	0
{ 'pYgr0' }	1	0.79559	1
{ 'pYisMINY' }	0.28039	0.10917	0.14247
{ 'pYisMAXY' }	0.044922	0.19422	0.044922
{ 'p1' }	1	-10	1
{ 'p10' }	1	-10	1
{ 'p25' }	1	1.1	1.1
{ 'p50' }	2	2	2
{ 'p75' }	3	5	2.5
{ 'p90' }	3	9	3.3
{ 'p99' }	4	9	4
{ 'fl_cov_cl_mt_x_of_s' }	0.82282	1.589	0.78646
{ 'fl_cor_cl_mt_x_of_s' }	1	0.33534	0.95887
{ 'fl_cov_cl_mt_y_of_s' }	1.589	27.289	1.8353
{ 'fl_cor_cl_mt_y_of_s' }	0.33534	1	0.38856
{ 'fl_cov_cl_mt_z_of_s' }	0.78646	1.8353	0.81758
{ 'fl_cor_cl_mt_z_of_s' }	0.95887	0.38856	1
{ 'fracByP1' }	0.13504	-0.56498	0.068934
{ 'fracByP10' }	0.13504	-0.56498	0.068934
{ 'fracByP25' }	0.13504	-0.53456	0.14234
{ 'fracByP50' }	0.42991	-0.39181	0.43856
{ 'fracByP75' }	0.91346	0.095425	0.60296
{ 'fracByP90' }	0.91346	1	0.91306
{ 'fracByP99' }	1	1	1

## Test FF\_SIMU\_STATS Four States-Points Matrix Single Column Inputs

Same as before, but now inputs are single column, should have identical results:

```

% Array Inputs
mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');
mp_cl_mt_xyz_of_s('cl_mt_x_of_s') = {mt_x_of_s(:), zeros(1)};
mp_cl_mt_xyz_of_s('cl_mt_y_of_s') = {mt_y_of_s(:), zeros(1)};
mp_cl_mt_xyz_of_s('cl_mt_z_of_s') = {mt_z_of_s(:), zeros(1)};
mp_cl_mt_xyz_of_s('ar_st_y_name') = ["cl_mt_x_of_s", "cl_mt_y_of_s", "cl_mt_z_of_s"];

```

### % Call Function

```
mp_cl_mt_xyz_of_s_out = ff_simu_stats(mt_f_of_s(:), mp_cl_mt_xyz_of_s);
```

xxx tb\_outcomes: all stats xxx

OriginalVariableNames	cl_mt_x_of_s	cl_mt_y_of_s	cl_mt_z_of_s
{'mean' }	2.0763	1.9323	2.0668
{'sd' }	0.9071	5.2239	0.9042
{'coefofvar' }	0.43688	2.7034	0.43749
{'min' }	1	-10	1
{'max' }	4	9	4
{'pYls0' }	0	0	0
{'pYls0' }	0	0.20441	0
{'pYgr0' }	1	0.79559	1
{'pYlsMINY' }	0.28039	0.10917	0.14247
{'pYlsMAXY' }	0.044922	0.19422	0.044922
{'p1' }	1	-10	1
{'p10' }	1	-10	1
{'p25' }	1	1.1	1.1
{'p50' }	2	2	2
{'p75' }	3	5	2.5
{'p90' }	3	9	3.3
{'p99' }	4	9	4
{'fl_cov_cl_mt_x_of_s' }	0.82282	1.589	0.78646
{'fl_cor_cl_mt_x_of_s' }	1	0.33534	0.95887
{'fl_cov_cl_mt_y_of_s' }	1.589	27.289	1.8353
{'fl_cor_cl_mt_y_of_s' }	0.33534	1	0.38856
{'fl_cov_cl_mt_z_of_s' }	0.78646	1.8353	0.81758
{'fl_cor_cl_mt_z_of_s' }	0.95887	0.38856	1
{'fracByP1' }	0.13504	-0.56498	0.068934
{'fracByP10' }	0.13504	-0.56498	0.068934
{'fracByP25' }	0.13504	-0.53456	0.14234
{'fracByP50' }	0.42991	-0.39181	0.43856
{'fracByP75' }	0.91346	0.095425	0.60296
{'fracByP90' }	0.91346	1	0.91306
{'fracByP99' }	1	1	1

## Test FF\_SIMU\_STATS Print Many Details

The Same As before, but now control which percentiles and other details to display.

### % Array Inputs

```
mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');  
mp_cl_ar_xyz_of_s('cl_ar_x_of_s') = {mt_x_of_s(:), zeros(1)};  
mp_cl_ar_xyz_of_s('cl_ar_z_of_s') = {mt_z_of_s(:), zeros(1)};  
mp_cl_ar_xyz_of_s('ar_st_y_name') = ["cl_ar_x_of_s", "cl_ar_z_of_s"];
```

### % controls

```
mp_support = containers.Map('KeyType','char', 'ValueType','any');  
mp_support('bl_display_detail') = false;  
mp_support('bl_display_final') = true;  
mp_support('bl_display_drvm2outcomes') = false;  
mp_support('ar_fl_percentiles') = [25 50 75];  
mp_support('bl_display_drvstats') = true;  
mp_support('bl_display_drvm2covcor') = false;
```

### % Call Function

```
mp_cl_mt_xyz_of_s_out = ff_simu_stats(mt_f_of_s(:), mp_cl_ar_xyz_of_s, mp_support);
```

```

-----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Summary Statistics for: cl_ar_x_of_s
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-----

```

```

fl_choice_mean
    2.0763

fl_choice_sd
    0.9071

fl_choice_coefofvar
    0.4369

fl_choice_prob_zero
    0

fl_choice_prob_below_zero
    0

fl_choice_prob_above_zero
    1

fl_choice_prob_max
    0.0449

```

tb_disc_cumu			
cl_ar_x_of_sDiscreteVal	cl_ar_x_of_sDiscreteValProbMass	CDF	cumsumFrac
1	0.28039	28.039	0.13504
1.5	0.13561	41.6	0.23301
2	0.20441	62.041	0.42991
3	0.33466	95.508	0.91346
4	0.044922	100	1
cl_ar_x_of_sDiscreteVal	cl_ar_x_of_sDiscreteValProbMass	CDF	cumsumFrac
1	0.28039	28.039	0.13504
1.5	0.13561	41.6	0.23301
2	0.20441	62.041	0.42991
3	0.33466	95.508	0.91346
4	0.044922	100	1

tb_prob_drv		
percentiles	cl_ar_x_of_sDiscreteValPercentileValues	fracOfSumHeldBelowThisPercentile
25	1	0.13504
50	2	0.42991
75	3	0.91346

```

-----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Summary Statistics for: cl_ar_z_of_s
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-----

```

```

fl_choice_mean
    2.0668

fl_choice_sd
    0.9042

```

fl\_choice\_coefofvar  
0.4375

fl\_choice\_prob\_zero  
0

fl\_choice\_prob\_below\_zero  
0

fl\_choice\_prob\_above\_zero  
1

fl\_choice\_prob\_max  
0.0449

tb\_disc\_cumu

cl_ar_z_of_sDiscreteVal	cl_ar_z_of_sDiscreteValProbMass	CDF	cumsumFrac
1	0.14247	14.247	0.068934
1.1	0.13792	28.039	0.14234
1.5	0.13561	41.6	0.24076
2	0.20441	62.041	0.43856
2.3	0.056663	67.708	0.50162
2.5	0.083786	76.086	0.60296
3.3	0.19422	95.508	0.91306
4	0.044922	100	1

cl_ar_z_of_sDiscreteVal	cl_ar_z_of_sDiscreteValProbMass	CDF	cumsumFrac
1	0.14247	14.247	0.068934
1.1	0.13792	28.039	0.14234
1.5	0.13561	41.6	0.24076
2	0.20441	62.041	0.43856
2.3	0.056663	67.708	0.50162
2.5	0.083786	76.086	0.60296
3.3	0.19422	95.508	0.91306
4	0.044922	100	1

tb\_prob\_drv

percentiles	cl_ar_z_of_sDiscreteValPercentileValues	fracOfSumHeldBelowThisPercentile
25	1.1	0.14234
50	2	0.43856
75	2.5	0.60296

xxx tb\_outcomes: all stats xxx

OriginalVariableNames	cl_ar_x_of_s	cl_ar_z_of_s
{'mean' }	2.0763	2.0668
{'sd' }	0.9071	0.9042
{'coefofvar' }	0.43688	0.43749
{'min' }	1	1
{'max' }	4	4
{'pYis0' }	0	0
{'pYls0' }	0	0
{'pYgr0' }	1	1
{'pYisMINY' }	0.28039	0.14247
{'pYisMAXY' }	0.044922	0.044922
{'p25' }	1	1.1
{'p50' }	2	2
{'p75' }	3	2.5

{'fl_cov_cl_ar_x_of_s'}	0.82282	0.78646
{'fl_cor_cl_ar_x_of_s'}	1	0.95887
{'fl_cov_cl_ar_z_of_s'}	0.78646	0.81758
{'fl_cor_cl_ar_z_of_s'}	0.95887	1
{'fracByP25'}	0.13504	0.14234
{'fracByP50'}	0.42991	0.43856
{'fracByP75'}	0.91346	0.60296