

# FF\_DISC\_RAND\_VAR\_MASS2OUTCOMES Examples

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

This is the example vignette for function: `ff_disc_rand_var_mass2outcomes` from the [MEconTools Package](#).  
This function generates sorted discrete random variable from state-space joint distribution.

## Test FF\_DISC\_RAND\_VAR\_MASS2OUTCOMES Defaults

Call the function with defaults.

```
ff_disc_rand_var_mass2outcomes();
```

```
INPUT f(a,z): mt_dist_bystates
0.0289  0.0465  0.0228  0.0036  0.0001
0.0241  0.0930  0.0857  0.0241  0.0015
0.0080  0.0744  0.1285  0.0643  0.0074
0.0013  0.0297  0.0964  0.0857  0.0186
0.0001  0.0059  0.0361  0.0571  0.0232
0.0000  0.0005  0.0054  0.0152  0.0116
```

```
INPUT y(a,z): mt_choice_bystates
-5  -4  -5  -4  -4
-3  -2  -3  -2  -3
-1  -1  -1   0   0
 1   1   2   3   1
 4   3   3   4   3
 5   6   5   6   6
```

```
OUTPUT f(y): ar_choice_prob_byY
0.0518
0.0502
0.1113
0.1171
0.2109
0.0717
0.0497
0.0964
0.1510
0.0572
0.0054
0.0273
```

```
OUTPUT f(y,z): mt_choice_prob_byYZ
0.0289  0  0.0228  0  0
0  0.0465  0  0.0036  0.0001
0.0241  0  0.0857  0  0.0015
0  0.0930  0  0.0241  0
0.0080  0.0744  0.1285  0  0
0  0  0  0.0643  0.0074
0.0013  0.0297  0  0  0.0186
0  0  0.0964  0  0
0  0.0059  0.0361  0.0857  0.0232
0.0001  0  0  0.0571  0
0.0000  0  0.0054  0  0
0  0.0005  0  0.0152  0.0116
```

```
OUTPUT f(y,a): mt_choice_prob_byYA
0.0518  0  0  0  0  0
0.0502  0  0  0  0  0
0  0.1113  0  0  0  0
```

0	0.1171	0	0	0	0
0	0	0.2109	0	0	0
0	0	0.0717	0	0	0
0	0	0	0.0497	0	0
0	0	0	0.0964	0	0
0	0	0	0.0857	0.0653	0
0	0	0	0	0.0572	0
0	0	0	0	0	0.0054
0	0	0	0	0	0.0273

OUTPUT f(y) and y in table: tb\_choice\_drv\_cur\_byY

binomtestOutcomes	probMassFunction
-5	0.051764
-4	0.050217
-3	0.11126
-2	0.11706
-1	0.21092
0	0.071696
1	0.049682
2	0.096388
3	0.15102
4	0.057231
5	0.0054256
6	0.027329

## Test FF\_DISC\_RAND\_VAR\_MASS2OUTCOMES Four States-Points

Over some (a,z) states that is 2 by 2, matrix or vectorized inputs identical results.

```
% Set Parameters
st_y_name = 'consumption';
% consumption matrix: c(a,z)
mt_c_of_s = [1,2;3,1];
% stationary mass over assets adn shocks: f(a,z)
mt_f_of_s = rand(size(mt_c_of_s));
mt_f_of_s = mt_f_of_s/sum(mt_f_of_s, 'all');
% Call Function
[ar_f_of_y, ar_y_unique_sorted] = ...
    ff_disc_rand_var_mass2outcomes(st_y_name, mt_c_of_s, mt_f_of_s);
% print
disp([ar_f_of_y ar_y_unique_sorted]);
```

0.4039	1.0000
0.2971	2.0000
0.2990	3.0000

Same as before, but now inputs are single column:

```
% Call Function
[ar_f_of_y, ar_y_unique_sorted] = ...
    ff_disc_rand_var_mass2outcomes(st_y_name, mt_c_of_s(:), mt_f_of_s);
disp([ar_f_of_y ar_y_unique_sorted]);
```

0.4039	1.0000
0.2971	2.0000
0.2990	3.0000

## Test FF\_DISC\_RAND\_VAR\_MASS2OUTCOMES Conditional Mass Outputs

Same inputs as before, but now, also output additional conditional statistics,  $f(y, a)$ , where  $a$  is the row state variable for  $f(a, z)$ . For conditional statistics, must provide matrix based inputs.

```
% Set Parameters
st_y_name = 'consumption';
% consumption matrix: c(a,z)
mt_c_of_s = [1,2,0.5;
             3,1,2.0];
% stationary mass over assets adn shocks: f(a,z)
mt_f_of_s = rand(size(mt_c_of_s));
mt_f_of_s = mt_f_of_s/sum(mt_f_of_s, 'all');
% Call Function
[ar_f_of_y, ar_y_unique_sorted, mt_f_of_y_srow, mt_f_of_y_scol] = ...
    ff_disc_rand_var_mass2outcomes(st_y_name, mt_c_of_s, mt_f_of_s);
% print
disp([ar_f_of_y ar_y_unique_sorted]);
```

```
0.2695    0.5000
0.3765    1.0000
0.2649    2.0000
0.0891    3.0000
```

```
disp(mt_f_of_y_srow);
```

```
0.2695         0
0.1215    0.2550
0.1217    0.1432
         0    0.0891
```

```
disp(mt_f_of_y_scol);
```

```
         0         0    0.2695
0.1215    0.2550         0
         0    0.1217    0.1432
0.0891         0         0
```