FF_DISC_RAND_VAR_MASS2COVCOR 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_mass2covcor** from the **MEconTools Package**.
This function calculates covariance and correlation based for two discrete random variables.

Test FF_DISC_RAND_VAR_MASS2COVCOR Defaults

Call the function with defaults.

```
ff_disc_rand_var_mass2covcor();
```

```
---------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: covvar_input_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari	min
mt_f_of_s	1	5	2	30	6	5	1	0.033333	0.035743	1.0723	3.7187e-
mt_x_of_s	2	6	2	30	6	5	25	0.83333	5.3051	6.3661	
mt_y_of_s	3	7	2	30	6	5	249.78	8.3259	7.1913	0.86373	-6.37

```
xxx TABLE:mt_f_of_s xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5
r1	0.028917	0.046484	0.022848	0.0036146	0.000119
r2	0.024097	0.092967	0.085679	0.024097	0.0014875
r3	0.0080324	0.074374	0.12852	0.064259	0.0074374
r4	0.0013387	0.02975	0.096388	0.085679	0.018593
r5	0.00011156	0.0059499	0.036146	0.057119	0.023242
r6	3.7187e-06	0.00047599	0.0054218	0.015232	0.011621

```
xxx TABLE:mt_x_of_s xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5
r1	-7	-6	-7	-6	-6
r2	-5	-3	-5	-3	-4
r3	-2	-1	-1	0	-1
r4	2	2	3	4	2
r5	6	5	5	6	5
r6	8	9	7	9	9

```
xxx TABLE:mt_y_of_s xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5
r1	13.231	21.786	18.136	19.35	13.901
r2	9.946	16.887	9.6914	15.71	8.6906
r3	16.255	6.2166	13.799	5.2138	11.641
r4	12.628	2.7525	6.5321	0.27238	13.357
r5	5.8844	4.0352	6.05	0.14102	0.50318
r6	3.5617	-0.72091	5.1855	-6.3772	-4.4805

```
---------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: covvar_input_map Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	value
fl_x_mean	1	1	-0.11081
fl_x_sd	2	2	4.1239
fl_y_mean	3	3	8.8423
fl_y_sd	4	4	6.5845

---
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: covvar_output_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefv
mt_cov_component_weighted	1	1	2	30	6	5	-22.084	-0.73612	1.0404	-1.41
mt_x_devi_from_mean	2	2	2	30	6	5	28.324	0.94415	5.3051	5.61
mt_x_y_multiply	3	3	2	30	6	5	-939.63	-31.321	36.564	-1.16
mt_y_devi_from_mean	4	4	2	30	6	5	-15.493	-0.51644	7.1913	-13.9

xxx TABLE:mt_cov_component_weighted xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	-0.87434	-3.5432	-1.4628	-0.22368	-0.0035451
r2	-0.13003	-2.1607	-0.35565	-0.47814	0.00087767
r3	-0.11248	0.17365	-0.56642	-0.025838	-0.018507
r4	0.010697	-0.38241	-0.69273	-3.0184	0.17717
r5	-0.0020165	-0.14618	-0.51584	-3.0371	-0.99056
r6	-0.00015927	-0.041473	-0.14098	-2.1121	-1.4106

xxx TABLE:mt_x_devi_from_mean xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	-6.8892	-5.8892	-6.8892	-5.8892	-5.8892
r2	-4.8892	-2.8892	-4.8892	-2.8892	-3.8892
r3	-1.8892	-0.88919	-0.88919	0.11081	-0.88919
r4	2.1108	2.1108	3.1108	4.1108	2.1108
r5	6.1108	5.1108	5.1108	6.1108	5.1108
r6	8.1108	9.1108	7.1108	9.1108	9.1108

xxx TABLE:mt_x_y_multiply xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	-30.237	-76.225	-64.023	-61.882	-29.792
r2	-5.396	-23.242	-4.151	-19.842	0.59004
r3	-14.003	2.3348	-4.4073	-0.40209	-2.4884
r4	7.9905	-12.854	-7.1868	-35.23	9.5287
r5	-18.075	-24.568	-14.271	-53.172	-42.62
r6	-42.83	-87.129	-26.003	-138.66	-121.38

xxx TABLE:mt_y_devi_from_mean xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	4.389	12.943	9.2933	10.508	5.0587
r2	1.1037	8.0444	0.84902	6.8677	-0.15171
r3	7.4123	-2.6258	4.9566	-3.6286	2.7985
r4	3.7855	-6.0898	-2.3103	-8.57	4.5142
r5	-2.9579	-4.8071	-2.7924	-8.7013	-8.3392
r6	-5.2806	-9.5633	-3.6568	-15.22	-13.323

fl_cov

```
   -22.0835

fl_cor
   -0.8133
```

Test FF_DISC_RAND_VAR_MASS2COVCOR Four States-Points

Over some (a,z) states that is 2 by 2, c matrix, and y matrix, find correlation. Positively related.

```
% Set Parameters
mt_c_of_s = [1,2;3,1];
mt_y_of_s = [2,10;5,1.1];
rng(123);
mt_f_of_s = rand(size(mt_c_of_s));
mt_f_of_s = mt_f_of_s/sum(mt_f_of_s, 'all');
bl_display_drvm2covcor = false;
% Call Function
[fl_cov_xy, fl_cor_xy] = ff_disc_rand_var_mass2covcor(...
    mt_c_of_s, mt_y_of_s, mt_f_of_s, bl_display_drvm2covcor);
display(['cov=' num2str(fl_cov_xy) ',cor=', num2str(fl_cor_xy)]);
```

```
cov=1.4446,cor=0.65723
```

Same as before, but now inputs are single column:

```
% Call Function
[fl_cov_xy, fl_cor_xy] = ff_disc_rand_var_mass2covcor(...
    mt_c_of_s(:), mt_y_of_s(:), mt_f_of_s(:), bl_display_drvm2covcor);
display(['cov=' num2str(fl_cov_xy) ',cor=', num2str(fl_cor_xy)]);
```

```
cov=1.4446,cor=0.65723
```

Test FF_DISC_RAND_VAR_MASS2COVCOR Two Random Vectors

Generate two random vectors, with random or even mass, correlation should be zero:

```
% Set Parameters
rng(4567);
mt_c_of_s = rand([20,1])*100;
mt_y_of_s = rand([20,1])*100;
mt_f_of_s = rand(size(mt_c_of_s));
mt_f_of_s = mt_f_of_s/sum(mt_f_of_s, 'all');
bl_display_drvm2covcor = false;
% Call Function
[fl_cov_xy, fl_cor_xy] = ff_disc_rand_var_mass2covcor(...
    mt_c_of_s, mt_y_of_s, mt_f_of_s, bl_display_drvm2covcor);
display(['cov=' num2str(fl_cov_xy) ',cor=', num2str(fl_cor_xy)]);
```

```
cov=-57.6533,cor=-0.062023
```

Test FF_DISC_RAND_VAR_MASS2COVCOR Provide Mean and SD

Same as above, but now provide means and sd for x andy directly. The results are the same as when mean and sd are calculated inside the function.

```
% Set Parameters
rng(4567);
mt_c_of_s = rand([20,1])*100;
mt_y_of_s = rand([20,1])*100;
mt_f_of_s = rand(size(mt_c_of_s));
mt_f_of_s = mt_f_of_s/sum(mt_f_of_s, 'all');
fl_c_mean = sum(mt_f_of_s.*mt_c_of_s);
fl_c_sd = sqrt(sum(mt_f_of_s.*(mt_c_of_s-fl_c_mean).^2));
fl_y_mean = sum(mt_f_of_s.*mt_y_of_s);
fl_y_sd = sqrt(sum(mt_f_of_s.*(mt_y_of_s-fl_y_mean).^2));
bl_display_drvm2covcor = false;
% Call Function
[fl_cov_xy, fl_cor_xy] = ff_disc_rand_var_mass2covcor(...
    mt_c_of_s, mt_y_of_s, mt_f_of_s, ...
    fl_c_mean, fl_c_sd, ...
    fl_y_mean, fl_y_sd, bl_display_drvm2covcor);
display(['cov=' num2str(fl_cov_xy) ',cor=', num2str(fl_cor_xy)]);
```

```
cov=-57.6533,cor=-0.062023
```