

Distribution Grid Search

This is the example vignette for function: [snw_ds_main_grid_search](#) from the [PrjOptiSNW Package](#). This function solves for vfi and gets distribution induced by policy functions and exogenous distributions. **Grid Search** for VFI and **Grid Search** also for Distribution. The results are illustrative of the differences between using grid search and exact solution. The grid search solution here is not fully vectorized but loops over the state-space.

Test SNW_DS_MAIN_GRID_SEARCH Defaults More Dense

Due to the speed of running this, the example below only uses dense grid

```
mp_params = snw_mp_param('default_moredense');
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
[Phi_true,Phi_adj,A_agg,Y_inc_agg,it,mp_dsvfi_results] = snw_ds_main_grid_search(mp_params, mp_
```

Elapsed time is 6514.834013 seconds.

Completed SNW_VFI_MAIN_GRID_SEARCH;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test

Elapsed time is 8310.394598 seconds.

Completed SNW_DS_MAIN;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test

```
Phi_true = Phi_true/sum(Phi_true(:));
```

Show All Info in mp_dsvfi_results More Dense

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'))
```

	mean	sd	coefofvar	min	max	pYis0	pYls0	pYgr0	p
a_ss	3.6126	6.4914	1.7969	0	135	0.17087	0	0.82913	
ap_ss	12.366	8.591	0.69474	1	55	0	0	1	
cons_ss	1.1622	0.80935	0.69639	0.036857	140.65	0	0	1	1
v_ss	-15.043	17.999	-1.1965	-597.7	23.892	0	0.8378	0.1622	1
n_ss	2.3554	1.4375	0.61029	1	6	0	0	1	
y_all	1.5684	1.4453	0.92149	0.038325	47.427	0	0	1	1
y_head_inc	1.2411	1.1553	0.9309	0.038325	31.844	0	0	1	1
y_head_earn	1.0444	1.0725	1.0269	0	26.444	0.2016	0	0.7984	
y_spouse_inc	0.32734	0.73631	2.2494	0	15.702	0.52499	0	0.47501	
yshr_interest	0.096139	0.15385	1.6002	0	0.99295	0.17087	0	0.82913	
yshr_wage	0.79228	0.33742	0.42588	0	1	0.10584	0	0.89416	
yshr_SS	0.11158	0.25418	2.278	0	1	0.7984	0	0.2016	
yshr_tax	0.18447	0.034469	0.18686	0.038299	0.25519	0	0	1	1
yshr_nttxss	0.072887	0.27591	3.7854	-0.88844	0.25519	0	0.18437	0.81563	

More Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Probability mass matrixes, Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
```

```

age_grid = 18:100;
agrid = mp_params('agrid');
eta_H_grid = mp_params('eta_H_grid');
eta_S_grid = mp_params('eta_S_grid');
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2f;')]);
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});

```

Analyze Probability Mass Along Age Dimensions

Where are the mass at? Analyze mass given state space components.

```

% Get the Joint distribution over all states
% Define Graph Inputs
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = false; % do not log

```

Exogenous Permanent States Mass: Life Cycle, Edu and Marraige

Tabulate value and policies along savings and shocks:

```

% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,5,4];
% Value Function
tb_prob_aem = ff_summ_nd_array("P(Age, EDU, MARRY)", Phi_true, true, ["sum"], 3, 1, cl_mp_data

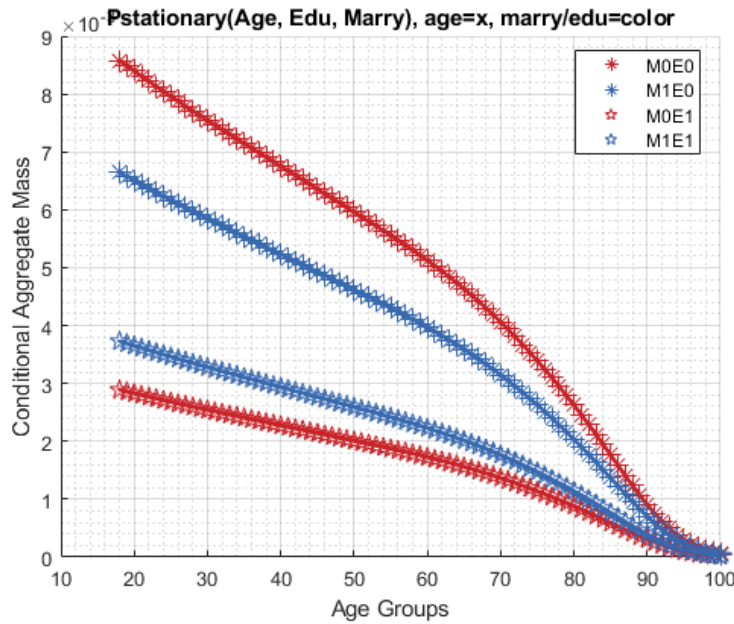
```

xxx	P(Age, EDU, MARRY))	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx								
group	marry	edu	sum_age_18	sum_age_19	sum_age_20	sum_age_21	sum_age_22	sum_age_23	sum	
1	0	0	0.0085768	0.0084866	0.0083969	0.0083078	0.0082194	0.0081317	0.	
2	1	0	0.0066438	0.0065739	0.0065044	0.0064354	0.0063669	0.006299	0.0	
3	0	1	0.0028875	0.0028571	0.002827	0.002797	0.0027672	0.0027377	0.0	
4	1	1	0.0037292	0.0036899	0.0036509	0.0036122	0.0035738	0.0035356	0.0	

```

mp_support_graph('cl_st_graph_title') = {'Pstationary(Age, Edu, Marry), age=x, marry/edu=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
ar_row_grid = ["M0E0", "M1E0", "M0E1", "M1E1"];
mp_support_graph('cl_st_xtitle') = {'Age Groups'};
mp_support_graph('cl_scatter_shapes') = {'*', '*', 'p', 'p' };
mp_support_graph('cl_colors') = {'red', 'blue', 'red', 'blue'};
ff_graph_grid((tb_prob_aem{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);

```

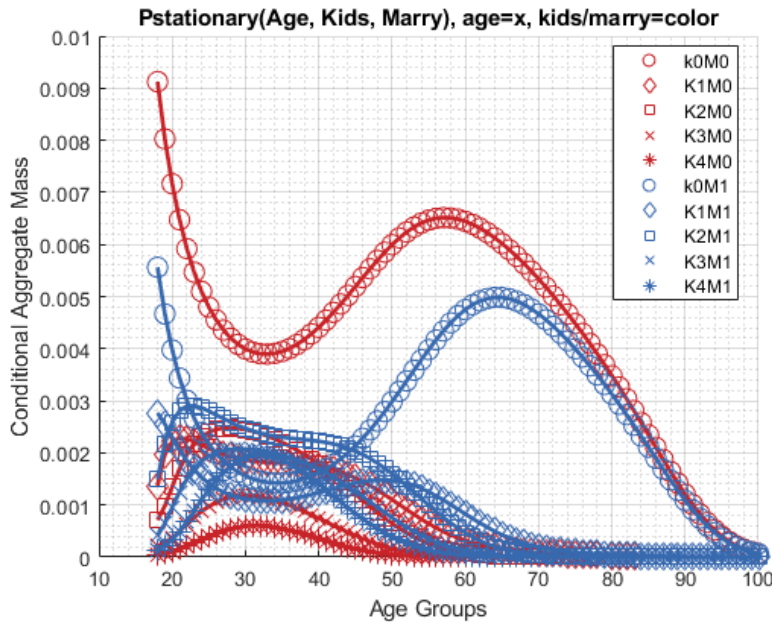


Kids and Marry By Age Mass

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_prob_amarrykids = ff_summ_nd_array("P(Age, Kids, Marry)", Phi_true, true, ["sum"], 3, 1, cl
```

xxx	P(Age, Kids, Marry))	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx								
group	kids	marry	sum_age_18	sum_age_19	sum_age_20	sum_age_21	sum_age_22	sum_age_23	su	
1	1	0	0.0091249	0.0080278	0.0071652	0.0064765	0.0059205	0.0054683	0	
2	2	0	0.0013699	0.0019743	0.0022187	0.0022858	0.0022687	0.0022149	0	
3	3	0	0.00071266	0.00098425	0.0013537	0.0016929	0.0019639	0.0021645	0	
4	4	0	0.00020622	0.00027865	0.00037326	0.00049476	0.00062818	0.00075864	0	
5	5	0	5.0761e-05	7.8715e-05	0.000113	0.00015485	0.00020534	0.00026306	0	
6	1	1	0.0055624	0.0046679	0.0039774	0.0034368	0.0030088	0.0026667	0	
7	2	1	0.0027682	0.0025539	0.0023005	0.0020611	0.0018525	0.0016773		
8	3	1	0.0014982	0.0021823	0.0025943	0.0028096	0.002896	0.0029031	0	
9	4	1	0.00041197	0.00064648	0.00095224	0.0012491	0.0015009	0.0016975	0	
10	5	1	0.00013221	0.0002132	0.00033097	0.00049097	0.00068255	0.0008901	0	

```
mp_support_graph('cl_st_graph_title') = {'Pstationary(Age, Kids, Marry), age=x, kids/marry=col
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
ar_row_grid = [...
    "k0M0", "k1M0", "k2M0", "k3M0", "k4M0", ...
    "k0M1", "k1M1", "k2M1", "k3M1", "k4M1"];
mp_support_graph('cl_scatter_shapes') = {...
    'o', 'd', 's', 'x', '*', ...
    'o', 'd', 's', 'x', '*'};
mp_support_graph('cl_colors') = {...
    'red', 'red', 'red', 'red', 'red'...
    'blue', 'blue', 'blue', 'blue', 'blue'};
mp_support_graph('cl_st_xtitle') = {'Age Groups'};
ff_graph_grid((tb_prob_amarrykids{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```



Analyze Probability Mass Asset and Shock Dimensions

Where are the mass at?

```
% Define Graph Inputs
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = false; % do not log
```

Asset and Shock Mass

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_prob_az = ff_summ_nd_array("P(A,Z)", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar_p
```

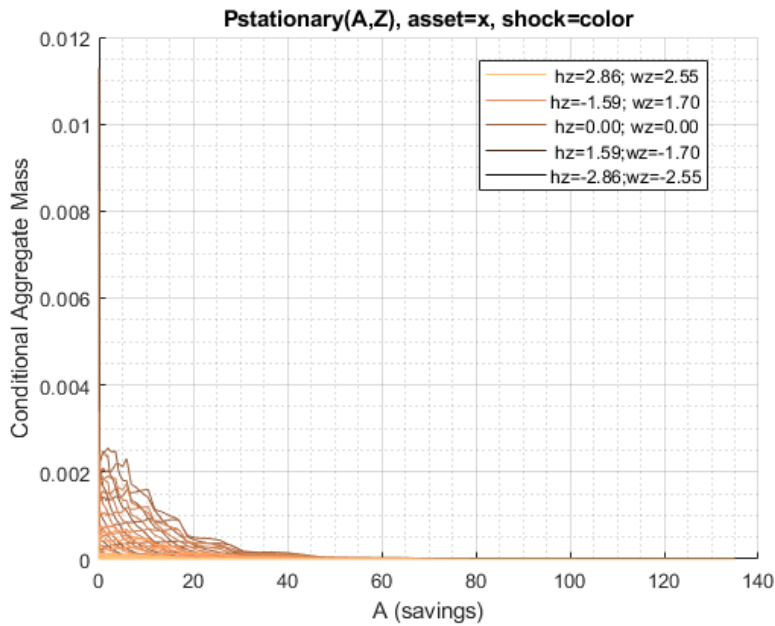
xxx	P(A,Z))	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx							
group	savings	sum_eta_1	sum_eta_2	sum_eta_3	sum_eta_4	sum_eta_5	sum_eta_6	sum_e	
1	0	3.3248e-08	5.983e-07	5.0468e-06	2.6071e-05	8.9773e-05	0.00021314	0.000	
2	0.00085734	1.0185e-10	1.5738e-09	3.3484e-08	6.0665e-07	5.2814e-06	1.7889e-05	2.91	
3	0.0068587	3.5085e-10	6.5198e-09	4.9979e-08	3.1065e-07	2.0919e-06	1.2505e-05	4.680	
4	0.023148	9.1216e-10	1.727e-08	1.5853e-07	9.1924e-07	4.228e-06	1.6829e-05	3.770	
5	0.05487	1.4512e-09	2.6722e-08	2.3417e-07	1.3076e-06	5.2407e-06	1.8046e-05	4.133	
6	0.10717	1.6379e-09	2.9149e-08	2.4125e-07	1.2884e-06	5.1297e-06	1.7682e-05	4.637	
7	0.18519	2.294e-09	4.0502e-08	3.4422e-07	1.7987e-06	6.7964e-06	2.0952e-05	5.119	
8	0.29407	2.7467e-09	4.5812e-08	3.5554e-07	1.8118e-06	6.8723e-06	2.0892e-05	5.255	
9	0.43896	2.8104e-09	4.8243e-08	3.9121e-07	1.9756e-06	7.1951e-06	2.0818e-05	4.920	
10	0.625	2.7355e-09	4.7422e-08	3.8625e-07	1.9709e-06	7.2487e-06	2.0707e-05	4.875	
11	0.85734	2.6045e-09	4.5648e-08	3.7801e-07	1.9446e-06	7.1135e-06	2.0218e-05	4.541	
12	1.1411	2.3091e-09	4.118e-08	3.427e-07	1.7857e-06	6.6881e-06	1.9205e-05	4.374	
13	1.4815	1.8391e-09	3.3807e-08	2.9225e-07	1.5886e-06	6.0926e-06	1.7531e-05	3.97	

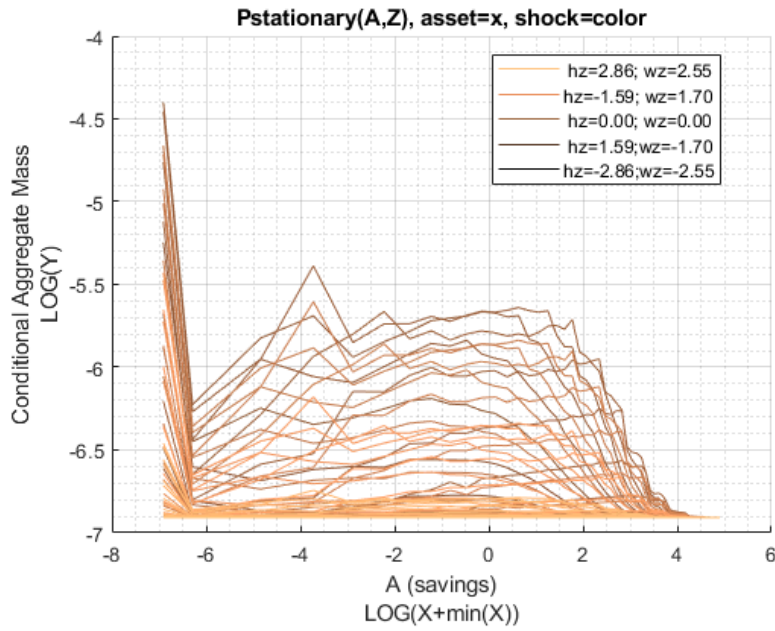
14	1.8836	1.4638e-09	2.7339e-08	2.4063e-07	1.3481e-06	5.3035e-06	1.5351e-05	3.61
15	2.3525	1.0398e-09	2.042e-08	1.9039e-07	1.1046e-06	4.5148e-06	1.3541e-05	3.112
16	2.8935	7.6831e-10	1.5168e-08	1.4306e-07	8.5678e-07	3.6129e-06	1.1466e-05	2.787
17	3.5117	5.0779e-10	1.0306e-08	1.0222e-07	6.4935e-07	2.7868e-06	9.0469e-06	2.307
18	4.2121	3.3725e-10	7.1325e-09	7.2301e-08	4.5701e-07	2.0605e-06	7.049e-06	1.869
19	5	2.0339e-10	4.5055e-09	4.7886e-08	3.1798e-07	1.4784e-06	5.2954e-06	1.510
20	5.8805	1.1697e-10	2.6614e-09	2.899e-08	2.04e-07	1.0085e-06	3.8316e-06	1.176
21	6.8587	6.4191e-11	1.5386e-09	1.7458e-08	1.2852e-07	6.6971e-07	2.7498e-06	9.309
22	7.9398	3.223e-11	8.2009e-10	1.023e-08	7.992e-08	4.4241e-07	1.8723e-06	6.916
23	9.1289	1.5935e-11	4.2547e-10	5.5799e-09	4.7148e-08	2.8701e-07	1.3269e-06	5.099
24	10.431	7.6602e-12	2.1449e-10	3.021e-09	2.7923e-08	1.8511e-07	9.7387e-07	3.8
25	11.852	3.4707e-12	1.1014e-10	1.6503e-09	1.6439e-08	1.1882e-07	6.7833e-07	2.801
26	13.396	1.5171e-12	5.0241e-11	8.4768e-10	9.2919e-09	7.4842e-08	4.48e-07	1.929
27	15.069	6.5407e-13	2.3567e-11	4.244e-10	4.9998e-09	4.3238e-08	2.8358e-07	1.235

```

mp_support_graph('cl_st_graph_title') = {'Pstationary(A,Z), asset=x, shock=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
mp_support_graph('cl_st_xtitle') = {'A (savings)'};
mp_support_graph('st_rowvar_name') = 'z=';
mp_support_graph('it_legend_select') = 5;
mp_support_graph('st_rounding') = '6.2f';
mp_support_graph('bl_graph_logy') = true;
mp_support_graph('cl_colors') = 'copper';
ff_graph_grid((tb_prob_az{1:end, 3:end}),'', ar_st_eta_HS_grid, agrid, mp_support_graph);% Consum

```





Asset Mass by Age

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [3,4,5,6,1,2];
% Value Function
tb_prob_age = ff_summ_nd_array("P(A,Z)", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar
```

xxx	P(A,Z))	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx							
group	savings	sum_age_18	sum_age_19	sum_age_20	sum_age_21	sum_age_22	sum_age_23	sum_a	
1	0	0.021837	0.0034489	0.0022777	0.0051619	0.0070701	0.0078351	0.00	
2	0.00085734	0	0.00068848	0.00063521	0.0016036	0.0010378	0.00037256	0.000	
3	0.0068587	0	0.0032017	0.0031082	0.0022491	0.0010432	0.00076245	0.000	
4	0.023148	0	0.0058625	0.0047177	0.0019351	0.0014397	0.0012162	0.00	
5	0.05487	0	0.0024915	0.0022549	0.0014027	0.001227	0.0010924	0.000	
6	0.10717	0	0.0014086	0.0019373	0.0017422	0.0016421	0.00161	0.00	
7	0.18519	0	0.0021091	0.0019779	0.0015628	0.0013724	0.0012726	0.00	
8	0.29407	0	0.00077614	0.0012753	0.0015936	0.0016414	0.0013435	0.00	
9	0.43896	0	0.00045252	0.001062	0.0009547	0.00098397	0.0013165	0.00	
10	0.625	0	0.00044667	0.0006628	0.0010416	0.00096275	0.00093205	0.00	
11	0.85734	0	0.00047477	0.00074294	0.0006549	0.00088876	0.00086767	0.000	
12	1.1411	0	5.5654e-05	0.00024727	0.00052409	0.00050156	0.00070796	0.000	
13	1.4815	0	6.2855e-05	0.00019563	0.00024402	0.00047382	0.00041139	0.000	
14	1.8836	0	5.9284e-05	0.00012121	0.00019477	0.00022476	0.00042115	0.000	
15	2.3525	0	5.4786e-05	9.1062e-05	0.00015255	0.0001713	0.0001937	0.000	
16	2.8935	0	1.2562e-05	5.2495e-05	7.1526e-05	0.00013824	0.00013812	0.000	
17	3.5117	0	1.3674e-06	1.1359e-05	3.0456e-05	4.9417e-05	0.00011344	0.000	
18	4.2121	0	1.7163e-07	5.944e-06	2.1633e-05	2.4249e-05	3.9771e-05	9.192	
19	5	0	7.8629e-08	1.35e-06	5.2915e-06	1.3252e-05	2.1725e-05	3.396	
20	5.8805	0	6.0581e-09	7.6218e-07	4.4269e-06	1.3802e-05	1.2238e-05	1.981	
21	6.8587	0	4.4396e-10	9.1165e-08	6.9508e-07	3.4569e-06	1.388e-05	1.258	
22	7.9398	0	0	3.4291e-08	4.1481e-07	2.9624e-06	4.2114e-06	1.324	
23	9.1289	0	0	2.5522e-08	3.1203e-07	5.0602e-07	2.4977e-06	3.850	
24	10.431	0	0	6.5868e-10	4.3387e-08	3.9922e-07	1.8125e-06	3.032	
25	11.852	0	0	2.4326e-10	1.8803e-08	2.6275e-07	4.5097e-07	1.296	
26	13.396	0	0	4.8838e-12	1.8589e-08	3.869e-08	2.8292e-07	1.489	
27	15.069	0	0	1.813e-12	4.7634e-10	2.1906e-08	2.1902e-07	2.796	
28	16.875	0	0	0	6.2066e-11	1.9079e-08	3.5513e-08	2.221	

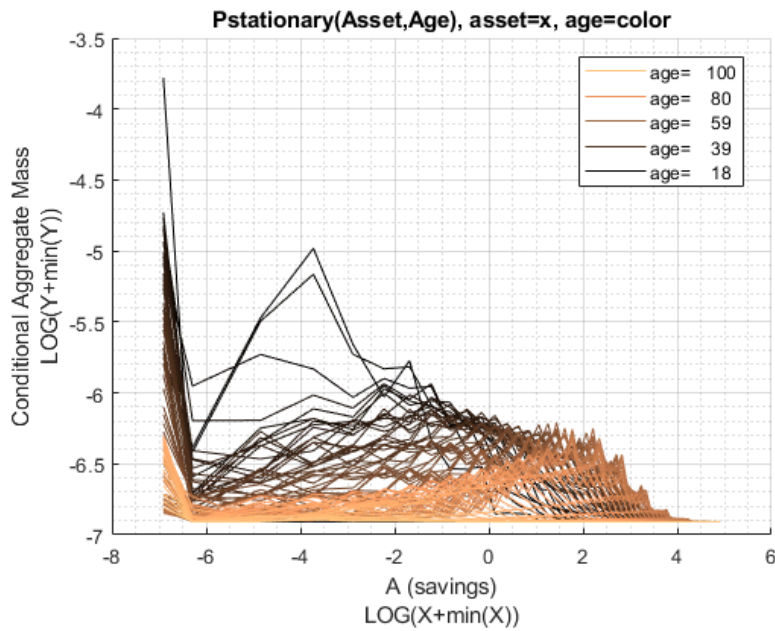
29	18.82	0	0	0	6.42e-12	6.8737e-10	2.0331e-08	1.702
30	20.91	0	0	0	1.6411e-13	9.995e-11	1.6987e-08	2.834
31	23.148	0	0	0	1.9878e-14	1.2652e-11	7.6143e-10	2.048
32	25.541	0	0	0	0	7.895e-13	1.3318e-10	1.173
33	28.093	0	0	0	0	1.2464e-14	1.45e-11	7.138
34	30.81	0	0	0	0	2.5551e-16	5.986e-13	1.196
35	33.697	0	0	0	0	0	1.2701e-14	9.134
36	36.758	0	0	0	0	0	2.0518e-16	3.324
37	40	0	0	0	0	0	2.4829e-18	9.849
38	43.427	0	0	0	0	0	0	1.426
39	47.044	0	0	0	0	0	0	2.188
40	50.856	0	0	0	0	0	0	
41	54.87	0	0	0	0	0	0	
42	59.089	0	0	0	0	0	0	
43	63.519	0	0	0	0	0	0	
44	68.164	0	0	0	0	0	0	
45	73.032	0	0	0	0	0	0	
46	78.125	0	0	0	0	0	0	
47	83.45	0	0	0	0	...		

```

mp_support_graph('cl_st_graph_title') = {'Pstationary(Asset,Age), asset=x, age=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
mp_support_graph('cl_st_xtitle') = {'A (savings)'};
mp_support_graph('st_rowvar_name') = 'age=';
mp_support_graph('it_legend_select') = 5;
mp_support_graph('st_rounding') = '6.0f';
mp_support_graph('bl_graph_logy') = true;
mp_support_graph('cl_colors') = 'copper';
ff_graph_grid((tb_prob_age{1:end, 3:end}),'', age_grid, agrid, mp_support_graph);% Consumption C

```





Probability Statistics A, C and V Conditional on Ages

Where are the mass at?

```

ap_ss = mp_dsvfi_results('ap_ss');
c_ss = mp_dsvfi_results('cons_ss');
v_ss = mp_dsvfi_results('v_ss');
n_ss = mp_dsvfi_results('n_ss');

y_head_inc = mp_dsvfi_results('y_head_inc_ss');
y_spouse_inc = mp_dsvfi_results('y_spouse_inc_ss');

yshr_wage = mp_dsvfi_results('yshr_wage_ss');
yshr_SS = mp_dsvfi_results('yshr_SS_ss');
yshr_nttxss = mp_dsvfi_results('yshr_nttxss_ss');

for it_ctr=1:size(ap_ss, 1)
    if (ismember(it_ctr, round(linspace(1, size(ap_ss, 1), 3))))
        display(['age = ' num2str(age_grid(it_ctr))]);

        % construct input data
        Phi_true_age = Phi_true(it_ctr, :, :, : ,: ,:);
        ap_ss_age = ap_ss(it_ctr, :, :, : ,: ,:);
        c_ss_age = c_ss(it_ctr, :, :, : ,: ,:);
        v_ss_age = v_ss(it_ctr, :, :, : ,: ,:);
        n_ss_age = n_ss(it_ctr, :, :, : ,: ,:);

        y_head_inc_age = y_head_inc(it_ctr, :, :, : ,: ,:);
        y_spouse_inc_age = y_spouse_inc(it_ctr, :, :, : ,: ,:);
        yshr_wage_age = yshr_wage(it_ctr, :, :, : ,: ,:);
        yshr_SS_age = yshr_SS(it_ctr, :, :, : ,: ,:);
        yshr_nttxss_age = yshr_nttxss(it_ctr, :, :, : ,: ,:);

        mp_cl_ar_xyz_of_s = containers.Map('KeyType','char', 'ValueType','any');

```



```

mp_cl_ar_xyz_of_s('ap_ss') = {ap_ss_age(:), zeros(1)};
mp_cl_ar_xyz_of_s('c_ss') = {c_ss_age(:), zeros(1)};
mp_cl_ar_xyz_of_s('v_ss') = {v_ss_age(:), zeros(1)};
mp_cl_ar_xyz_of_s('n_ss') = {n_ss_age(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_head_inc') = {y_head_inc_age(:), zeros(1)};
mp_cl_ar_xyz_of_s('y_spouse') = {y_spouse_age(:), zeros(1)};
mp_cl_ar_xyz_of_s('yshr_wage') = {yshr_wage_age(:), zeros(1)};
mp_cl_ar_xyz_of_s('yshr_SS') = {yshr_SS_age(:), zeros(1)};
mp_cl_ar_xyz_of_s('yshr_nttxss') = {yshr_nttxss_age(:), zeros(1)};
mp_cl_ar_xyz_of_s('ar_st_y_name') = ["ap_ss", "c_ss", "v_ss", "n_ss", ...
    "y_head_inc", "y_spouse", "yshr_wage", "yshr_SS", "yshr_nttxss"];

% controls
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('ar_fl_percentiles') = [0.01 10 25 50 75 90 99.99];
mp_support('bl_display_final') = true;
mp_support('bl_display_detail') = false;
mp_support('bl_display_drvm2outcomes') = false;
mp_support('bl_display_drvstats') = false;
mp_support('bl_display_drvm2covcor') = false;

% Call Function
mp_cl_mt_xyz_of_s = ff_simu_stats(Phi_true_age(:)/sum(Phi_true_age,'all'), mp_cl_ar_xyz_of_s);
end
end

```

age =18

xxx tb_outcomes: all stats xxx

OriginalVariableNames		ap_ss	c_ss	v_ss	n_ss	y_head_inc	y_spouse
{'mean'}	}	4.4841	0.73608	-24.737	1.9854	0.83951	0.20898
{'sd'}	}	2.5942	0.43457	24.426	1.0848	0.62548	0.43796
{'coefofvar'}	}	0.57854	0.59038	-0.98745	0.54639	0.74505	2.0956
{'min'}	}	1	0.036857	-597.7	1	0.038325	0
{'max'}	}	55	17.35	23.892	6	17.095	6.1141
{'pYis0'}	}	0	0	0	0	0	0.52499
{'pYls0'}	}	0	0	0.91009	0	0	0
{'pYgr0'}	}	1	1	0.089914	1	1	0.47501
{'pYisMINY'}	}	0.15961	2.0027e-06	8.8672e-09	0.41786	3.8147e-06	0.52499
{'pYisMAXY'}	}	0	0	0	0.0060544	0	7.6694e-09
{'p0_01'}	}	1	0.0682	-251.7	1	0.072367	0
{'p10'}	}	1	0.30119	-51.639	1	0.25802	0
{'p25'}	}	3	0.40819	-36.859	1	0.4872	0
{'p50'}	}	4	0.65163	-19.949	2	0.66948	0
{'p75'}	}	6	0.91178	-7.7939	3	0.91995	0.24381
{'p90'}	}	8	1.2709	-1.1471	4	1.7371	0.61906
{'p99_99'}	}	16	4.1437	17.666	6	6.1934	4.7312
{'fl_cov_ap_ss'}	}	6.73	0.61926	25.315	-0.086009	0.71874	0.77387
{'fl_cor_ap_ss'}	}	1	0.5493	0.3995	-0.030562	0.44295	0.68113
{'fl_cov_c_ss'}	}	0.61926	0.18885	7.7987	0.07295	0.25218	0.060176
{'fl_cor_c_ss'}	}	0.5493	1	0.7347	0.15474	0.92778	0.31618
{'fl_cov_v_ss'}	}	25.315	7.7987	596.64	-1.0333	10.003	2.2091
{'fl_cor_v_ss'}	}	0.3995	0.7347	1	-0.038995	0.65476	0.20651
{'fl_cov_n_ss'}	}	-0.086009	0.07295	-1.0333	1.1768	2.5745e-18	0.12195
{'fl_cor_n_ss'}	}	-0.030562	0.15474	-0.038995	1	3.7942e-18	0.25667
{'fl_cov_y_head_inc'}	}	0.71874	0.25218	10.003	2.5745e-18	0.39122	0.010942
{'fl_cor_y_head_inc'}	}	0.44295	0.92778	0.65476	3.7942e-18	1	0.039945
{'fl_cov_y_spouse'}	}	0.77387	0.060176	2.2091	0.12195	0.010942	0.19181
{'fl_cor_y_spouse'}	}	0.68113	0.31618	0.20651	0.25667	0.039945	1

{'fl_cov_yshr_wage' }	-1.9999e-30	-3.1429e-31	1.1155e-29	-5.0228e-31	-4.8634e-31	-1.3325e-31
{'fl_cor_yshr_wage' }	-1.7359e-15	-1.6286e-15	1.0284e-15	-1.0426e-15	-1.7509e-15	-6.8512e-16
{'fl_cov_yshr_SS' }	0	0	0	0	0	0
{'fl_cor_yshr_SS' }	NaN	NaN	NaN	NaN	NaN	NaN
{'fl_cov_yshr_nttxss' }	0.05377	0.011984	0.67106	0.0063843	0.014635	0.0061572
{'fl_cor_yshr_nttxss' }	0.65501	0.87148	0.8682	0.18598	0.73941	0.4443
{'fracByP0_01' }	0.035596	3.1006e-05	0.0014244	0.21046	5.4793e-05	0
{'fracByP10' }	0.035596	0.046234	0.31706	0.21046	0.031399	0
{'fracByP25' }	0.14894	0.11395	0.58865	0.21046	0.17954	0
{'fracByP50' }	0.39097	0.28419	0.84937	0.53024	0.32745	0
{'fracByP75' }	0.60677	0.54782	0.99563	0.77109	0.51036	0.13407
{'fracByP90' }	0.82323	0.77805	1.0163	0.92834	0.8397	0.41115
{'fracByP99_99' }	0.99971	0.99944	1	1	0.99925	0.99766

age =59

xxx tb_outcomes: all stats xxx

OriginalVariableNames	ap_ss	c_ss	v_ss	n_ss	y_head_inc	y_spouse	ys
{'mean' }	19.816	1.359	-12.596	1.7239	1.7902	0.41115	
{'sd' }	7.5997	0.94263	14.529	0.90777	1.4384	0.85562	
{'coefofvar' }	0.3835	0.69362	-1.1534	0.52659	0.80349	2.0811	
{'min' }	1	0.056816	-211.91	1	0.05988	0	
{'max' }	55	31.643	14.416	6	30.606	11.86	
{'pYis0' }	0	0	0	0	0	0.52499	
{'pYls0' }	0	0	0.80596	0	0	0	
{'pYgr0' }	1	1	0.19404	1	1	0.47501	
{'pYisMINY' }	0.0097508	1.4955e-06	4.4767e-10	0.48835	1.5128e-06	0.52499	3.
{'pYisMAXY' }	3.9575e-05	3.6923e-09	2.35e-07	0.0036816	5.7591e-07	7.0076e-09	
{'p0_01' }	1	0.1046	-83.156	1	0.11307	0	
{'p10' }	10	0.4435	-33.387	1	0.56354	0	
{'p25' }	15	0.68544	-20.857	1	0.82346	0	
{'p50' }	20	1.1501	-9.5768	2	1.4112	0	
{'p75' }	25	1.7983	-1.8874	2	2.3004	0.49169	
{'p90' }	29	2.5653	3.2283	3	3.3891	1.2002	
{'p99_99' }	52	10.514	13.528	6	17.142	9.1776	
{'fl_cov_ap_ss' }	57.755	5.9089	102.91	0.833	8.8962	1.5795	
{'fl_cor_ap_ss' }	1	0.82483	0.93205	0.12075	0.81382	0.24292	
{'fl_cov_c_ss' }	5.9089	0.88856	10.148	0.19066	1.1168	0.19114	-
{'fl_cor_c_ss' }	0.82483	1	0.74094	0.22282	0.82371	0.23699	
{'fl_cov_v_ss' }	102.91	10.148	211.1	2.8206	14.8	3.2552	
{'fl_cor_v_ss' }	0.93205	0.74094	1	0.21386	0.70816	0.26185	
{'fl_cov_n_ss' }	0.833	0.19066	2.8206	0.82404	0.051157	0.25285	-0
{'fl_cor_n_ss' }	0.12075	0.22282	0.21386	1	0.039179	0.32555	-
{'fl_cov_y_head_inc' }	8.8962	1.1168	14.8	0.051157	2.069	0.10701	-
{'fl_cor_y_head_inc' }	0.81382	0.82371	0.70816	0.039179	1	0.08695	
{'fl_cov_y_spouse' }	1.5795	0.19114	3.2552	0.25285	0.10701	0.73208	0
{'fl_cor_y_spouse' }	0.24292	0.23699	0.26185	0.32555	0.08695	1	
{'fl_cov_yshr_wage' }	-0.49211	-0.043084	-0.82571	-0.0010612	-0.051178	0.0085014	
{'fl_cor_yshr_wage' }	-0.75357	-0.53189	-0.66137	-0.013605	-0.41406	0.11563	
{'fl_cov_yshr_SS' }	0	0	0	0	0	0	
{'fl_cor_yshr_SS' }	NaN	NaN	NaN	NaN	NaN	NaN	
{'fl_cov_yshr_nttxss' }	0.18595	0.018907	0.38617	0.0064928	0.028554	0.0098282	-
{'fl_cor_yshr_nttxss' }	0.86513	0.70922	0.93979	0.2529	0.7019	0.40615	
{'fracByP0_01' }	0.00049206	1.8935e-05	0.0018764	0.28329	1.6223e-05	0	4.
{'fracByP10' }	0.038429	0.027522	0.35161	0.28329	0.022756	0	
{'fracByP25' }	0.15239	0.088557	0.65681	0.28329	0.081638	0	
{'fracByP50' }	0.38532	0.25604	0.94508	0.72028	0.23879	0	
{'fracByP75' }	0.65807	0.51752	1.0572	0.72028	0.4917	0.14854	
{'fracByP90' }	0.83502	0.75552	1.0505	0.85389	0.71833	0.42533	
{'fracByP99_99' }	0.99977	0.99911	1.0001	1	0.999	0.9979	

age =100

xxx tb_outcomes: all stats xxx

OriginalVariableNames	ap_ss	c_ss	v_ss	n_ss	y_head_inc	y_spouse
-----------------------	-------	------	------	------	------------	----------

{ 'mean' }	1	0.32267	-3.1409	1.4797	0.25976	0.090594
{ 'sd' }	3.5527e-15	0.17242	0.94302	0.50567	0.022536	0.18604
{ 'coefofvar' }	3.5527e-15	0.53435	-0.30024	0.34173	0.086757	2.0536
{ 'min' }	1	0.21707	-9.9745	1	0.24433	0
{ 'max' }	1	140.65	0.99282	6	5.6926	1.7565
{ 'pYis0' }	0	0	0	0	0	0.52499
{ 'pYls0' }	0	0	0.99701	0	0	0
{ 'pYgr0' }	1	1	0.0029896	1	1	0.47501
{ 'pYisMINY' }	1	0.38656	2.0113e-10	0.5232	0.55431	0.52499
{ 'pYisMAXY' }	1	0	0	4.2206e-08	0	1.321e-08
{ 'p0_01' }	1	0.21707	-6.4374	1	0.24433	0
{ 'p10' }	1	0.21707	-3.6791	1	0.24433	0
{ 'p25' }	1	0.21707	-3.6067	1	0.24433	0
{ 'p50' }	1	0.25723	-3.6067	1	0.24433	0
{ 'p75' }	1	0.36309	-2.8876	2	0.29263	0.10311
{ 'p90' }	1	0.50168	-1.819	2	0.29263	0.24116
{ 'p99_99' }	1	1.8447	0.23336	4	0.31763	1.6611
{ 'fl_cov_ap_ss' }	1.2622e-29	-6.2617e-30	6.0422e-29	-2.5736e-29	-5.0486e-30	-1.4744e-30
{ 'fl_cor_ap_ss' }	1	-1.0222e-14	1.8035e-14	-1.4326e-14	-6.3058e-14	-2.2307e-15
{ 'fl_cov_c_ss' }	-6.2617e-30	0.029727	0.1377	0.049528	0.0011258	0.029696
{ 'fl_cor_c_ss' }	-1.0222e-14	1	0.8469	0.56809	0.28973	0.92578
{ 'fl_cov_v_ss' }	6.0422e-29	0.1377	0.88928	0.13864	0.0090605	0.12887
{ 'fl_cor_v_ss' }	1.8035e-14	0.8469	1	0.29075	0.42634	0.73454
{ 'fl_cov_n_ss' }	-2.5736e-29	0.049528	0.13864	0.2557	0.0016977	0.047692
{ 'fl_cor_n_ss' }	-1.4326e-14	0.56809	0.29075	1	0.14898	0.50696
{ 'fl_cov_y_head_inc' }	-5.0486e-30	0.0011258	0.0090605	0.0016977	0.00050787	0.0005624
{ 'fl_cor_y_head_inc' }	-6.3058e-14	0.28973	0.42634	0.14898	1	0.13414
{ 'fl_cov_y_spouse' }	-1.4744e-30	0.029696	0.12887	0.047692	0.0005624	0.034611
{ 'fl_cor_y_spouse' }	-2.2307e-15	0.92578	0.73454	0.50696	0.13414	1
{ 'fl_cov_yshr_wage' }	-2.8197e-30	0.031233	0.14436	0.08326	0.00060599	0.034569
{ 'fl_cor_yshr_wage' }	-3.6954e-15	0.84344	0.71277	0.76664	0.1252	0.86517
{ 'fl_cov_yshr_SS' }	-1.9134e-29	-0.03166	-0.14677	-0.084206	-0.00062745	-0.03464
{ 'fl_cor_yshr_SS' }	-2.496e-14	-0.85097	-0.72129	-0.77173	-0.12903	-0.86288
{ 'fl_cov_yshr_nttxss' }	1.3054e-29	0.034596	0.16109	0.090587	0.00077099	0.037835
{ 'fl_cor_yshr_nttxss' }	1.5717e-14	0.8583	0.7307	0.76628	0.14634	0.8699
{ 'fracByP0_01' }	1	0.26006	0.00032592	0.35357	0.52138	0
{ 'fracByP10' }	1	0.26006	0.20378	0.35357	0.52138	0
{ 'fracByP25' }	1	0.26006	0.67085	0.35357	0.52138	0
{ 'fracByP50' }	1	0.41437	0.67085	0.35357	0.52138	0
{ 'fracByP75' }	1	0.59565	0.86725	0.99419	0.90276	0.17769
{ 'fracByP90' }	1	0.77472	0.96654	0.99419	0.90276	0.44354
{ 'fracByP99_99' }	1	0.99967	1	0.99999	0.99992	0.99968