Distribution Exact Savings Choices

This is the example vignette for function: **snw_ds_main** from the **PrjOptiSNW Package.** This function solves for vfi and gets distribution induced by policy functions and exogenous distributions. Looped to get distribution, but uses bisect vec for VFI.

Test SNW_DS_MAIN Defaults

Call the function with testing defaults.

```
mp_params = snw_mp_param('default_docdense');
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(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=496.3667
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari
V_VFI	1	1	6	4.37e+07	83	5.265e+05	-1.5339e+08	-3.5101	26.119	-7.441
ap_VFI	2	2	6	4.37e+07	83	5.265e+05	1.4159e+09	32.402	36.798	1.1357
cons_VFI	3	3	6	4.37e+07	83	5.265e+05	2.1402e+08	4.8975	8.3294	1.7007

```
xxx TABLE:V_VFI xxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c526499	c52650
r1	-346.51	-346.12	-343.63	-337.86	-328.51	21.702	21.852	22.003	22.154	22.30
r2	-334.38	-333.99	-331.51	-325.83	-316.83	21.724	21.869	22.015	22.163	22.31
r3	-322.45	-322.06	-319.6	-314.14	-305.6	21.745	21.885	22.027	22.171	22.31
r4	-310.63	-310.27	-307.99	-302.88	-294.87	21.767	21.903	22.041	22.182	22.32
r5	-299.94	-299.6	-297.46	-292.67	-285.12	21.775	21.907	22.042	22.18	22.32
r79	-9.9437	-9.9325	-9.8557	-9.6597	-9.3232	2.5394	2.5501	2.5602	2.5696	2.578
r80	-8.9023	-8.8911	-8.8143	-8.6183	-8.2818	2.3039	2.3121	2.3198	2.327	2.333
r81	-7.6363	-7.6251	-7.5484	-7.3524	-7.0159	2.0068	2.0124	2.0176	2.0226	2.027
r82	-5.9673	-5.9561	-5.8793	-5.6833	-5.3468	1.5958	1.5989	1.6018	1.6046	1.607
r83	-3.5892	-3.578	-3.5012	-3.3052	-2.9687	0.97904	0.98004	0.98097	0.98185	0.9826

```
xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c526499	c526500
r1	0	0	0.0005656	0.0075134	0.022901	114.75	120.41	126.27	132.38	138.8
r2	0	0	0.00051498	0.0065334	0.021549	114.86	120.53	126.41	132.54	138.95
r3	0	0	0.00051498	0.0049294	0.019875	114.97	120.65	126.56	132.7	139.12
r4	0	0	0.00051498	0.0047937	0.019672	115.73	121.42	127.34	133.51	139.92
r5	0	0	0.00048517	0.0046683	0.019484	116.5	122.21	128.15	134.32	140.74
r79	0	0	0	0	0	81.091	85.68	90.335	94.378	98.419
r80	0	0	0	0	0	76.669	80.563	84.304	88.04	91.693
r81	0	0	0	0	0	68.313	71.534	74.475	77.832	81.11
r82	0	0	0	0	0	50.126	53.467	56.953	58.745	60.587
r83	0	0	0	0	0	0	0	0	0	0

```
xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c526499	c
r1	0.036717	0.037251	0.040426	0.04363	0.048012	9.6491	9.817	9.9649	10.073	1
r2	0.036717	0.037251	0.040477	0.04461	0.049364	9.8118	9.9685	10.101	10.191	1
r3	0.036717	0.037251	0.040477	0.046214	0.051039	9.9779	10.12	10.234	10.302	1
r4	0.038144	0.038678	0.041903	0.047776	0.052666	10.131	10.258	10.354	10.405	1
r5	0.039534	0.040068	0.043323	0.04929	0.054241	10.272	10.384	10.463	10.5	
r79	0.2179	0.21844	0.22216	0.23228	0.25197	35.858	37.092	38.455	40.627	4
r80	0.2179	0.21844	0.22216	0.23228	0.25197	40.253	42.183	44.459	46.938	
r81	0.2179	0.21844	0.22216	0.23228	0.25197	48.587	51.19	54.266	57.123	6
r82	0.2179	0.21844	0.22216	0.23228	0.25197	66.755	69.238	71.77	76.192	8
r83	0.2179	0.21844	0.22216	0.23228	0.25197	116.87	122.69	128.71	134.92	1

```
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1498.3692
```

```
% [Phi_true,Phi_adj] = snw_ds_main(mp_params, mp_controls);
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	unweighted_sum	sd	coefofvar	gini	min	max	pYis
a_ss	4.2486	2228	6.7963	1.5996	0.68054	0	135	0.1
ap_ss	4.3473	5.3198e+08	6.834	1.572	0.68147	0	163.7	0.10
cons_ss	1.0676	5.0976e+07	0.69454	0.65055	0.3385	0.036717	141.66	
v_ss	-15.745	-2.1145e+07	21.68	-1.3769	-0.67203	-586.22	24.63	
n_ss	2.3554	21	1.4375	0.61029	0.3128	1	6	
y_all	1.415	8.3532e+07	1.4926	1.0548	0.47801	0	50.873	0.0072
y_head_inc	1.1087	1.9253e+06	1.0092	0.91029	0.41889	0.038108	24.357	
y_head_earn	0.88655	19732	0.92804	1.0468	0.53121	0	18.957	0.2
y_spouse_inc	0.35849	4.8273e+05	0.95494	2.6638	0.85255	0	26.627	0.52
yshr_interest	0.12214	3.8429e+06	0.16806	1.3759	0.66002	0	0.99299	0.1
yshr_wage	0.77513	8.8876e+06	0.33759	0.43553	0.2056	0	1	0.10
yshr_SS	0.10273	30336	0.23637	2.3009	0.91226	0	1	0.7
yshr_tax	0.17862	2.8339e+06	0.03519	0.19701	0.11226	0.036506	0.2552	
yshr_nttxss	0.075896	2.8036e+06	0.25563	3.3681	1.3974	-0.89184	0.2552	

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
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))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    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);
```

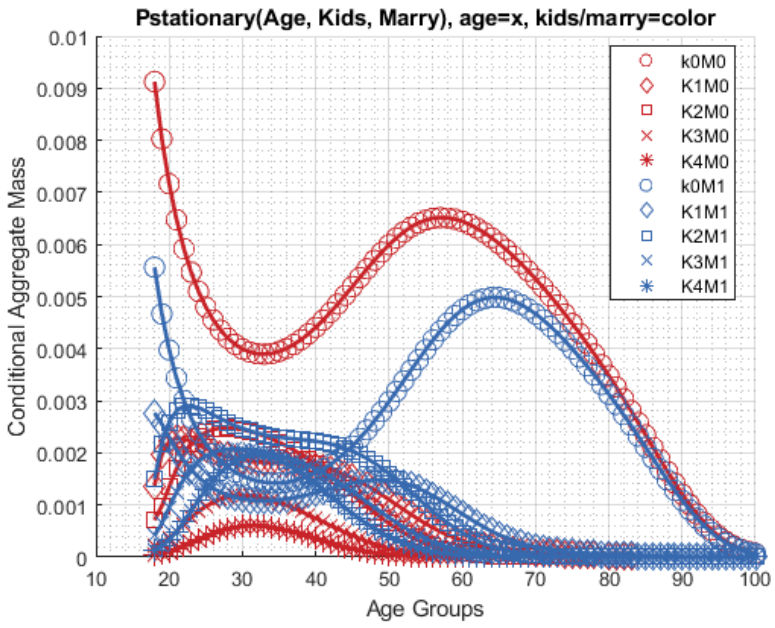
Kids and Marry By Age Mass

```
% 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))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
```

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=colo
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

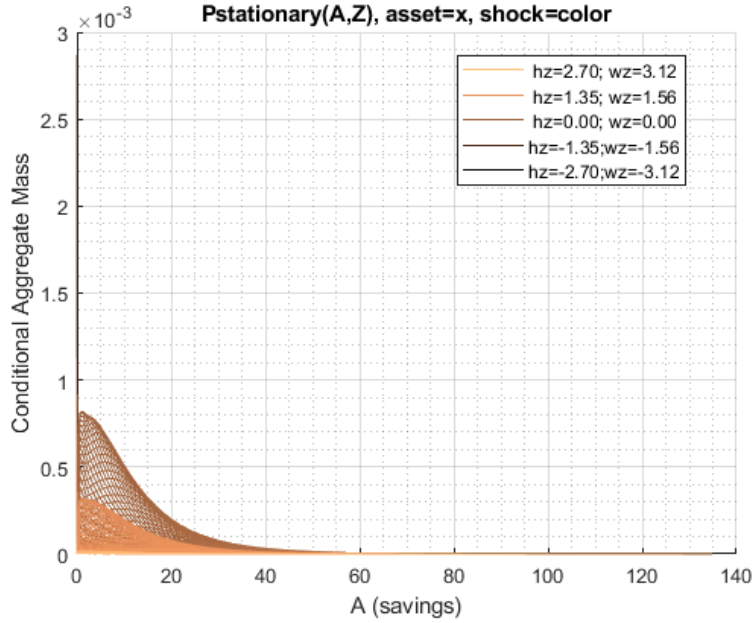
```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_prob_az = ff_summ_nd_array("P(A,Z))", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar_p
```

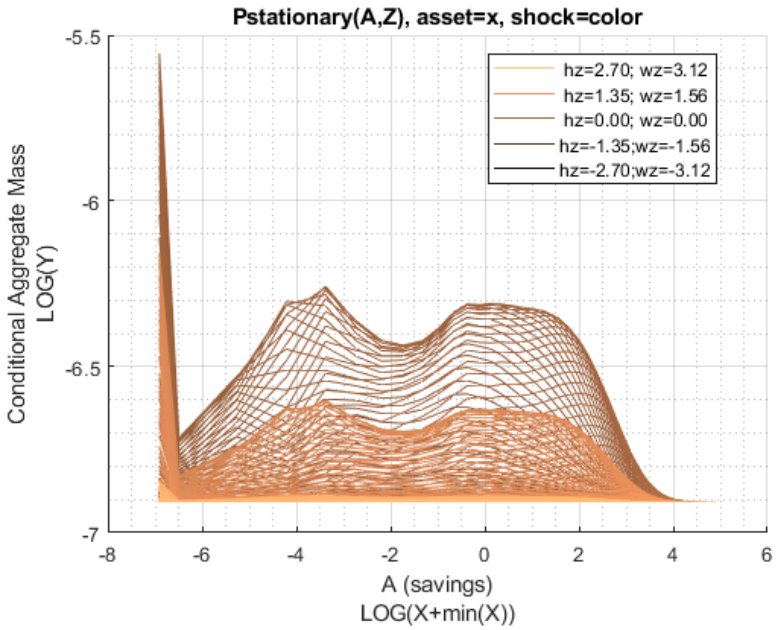
```
xxx  P(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
```

group	savings	sum_eta_1	sum_eta_2	sum_eta_3	sum_eta_4	sum_eta_5	sum_eta_6	sum_e
1	0	1.6824e-07	1.4406e-07	2.1911e-07	3.1913e-07	4.5491e-07	6.4002e-07	8.900
2	0.00051498	3.4279e-10	3.2632e-10	5.6501e-10	1.0203e-09	1.9975e-09	4.1764e-09	8.868
3	0.0041199	7.1369e-10	6.2373e-10	9.7246e-10	1.4702e-09	2.2039e-09	3.2988e-09	4.929
4	0.013905	1.573e-09	1.3633e-09	2.1044e-09	3.1331e-09	4.6025e-09	6.7334e-09	9.836
5	0.032959	5.494e-09	4.7235e-09	7.23e-09	1.0641e-08	1.5401e-08	2.211e-08	3.153
6	0.064373	6.5788e-09	5.6779e-09	8.702e-09	1.2804e-08	1.8492e-08	2.6448e-08	3.753

```
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
```





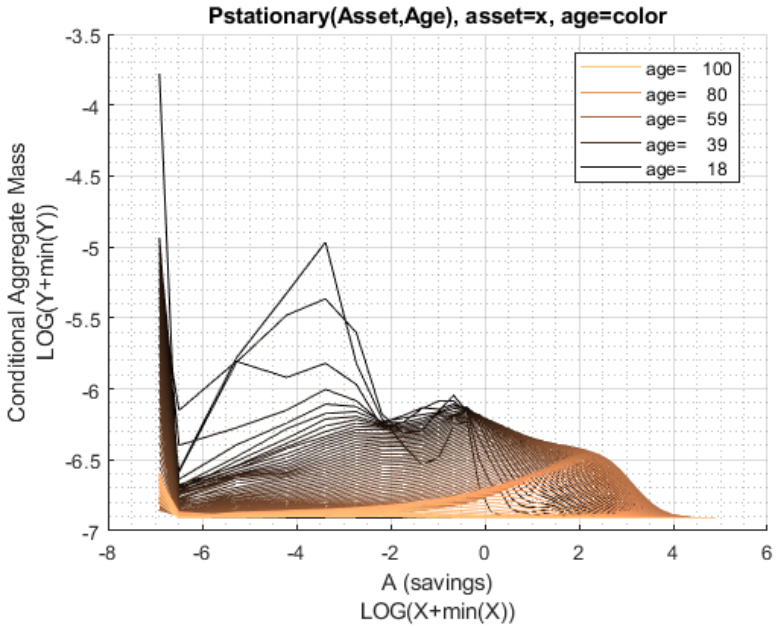
Asset Mass by Age

```
% 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_aage = ff_summ_nd_array("P(A,Z))", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar
```

```
xxx  P(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    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	0.0023507	0.0017993	0.0039371	0.0058435	0.0062157	0.00
2	0.00051498	0	0.00039608	0.00037932	0.0011301	0.00066626	0.00031095	0.000
3	0.0041199	0	0.0020816	0.0019888	0.002009	0.00088325	0.00066933	0.000
4	0.013905	0	0.0038656	0.0031682	0.001688	0.0011334	0.00094795	0.000
5	0.032959	0	0.0059678	0.0036757	0.0019686	0.0014691	0.0012275	0.00
6	0.064373	0	0.001968	0.0026857	0.0015598	0.0012805	0.0011877	0.00
7	0.11124	0	0.0010155	0.0010772	0.00089495	0.00094737	0.00096712	0.000
8	0.17664	0	0.00066497	0.00081578	0.0009608	0.0010548	0.0010166	0.000
9	0.26367	0	0.00045021	0.00085579	0.0011593	0.0011712	0.0010669	0.00
10	0.37542	0	0.00053095	0.0011218	0.0012745	0.0011467	0.0011044	0.00
11	0.51498	0	0.00090691	0.0013663	0.0012758	0.0012278	0.001195	0.00
12	0.68544	0	0.00097523	0.001111	0.0010957	0.0011325	0.0011755	0.00
13	0.88989	0	0.00023441	0.00050314	0.00074645	0.0009432	0.0010026	0.00
14	1.1314	0	4.5279e-05	0.00027467	0.00049029	0.00060869	0.00071831	0.00
15	1.4131	0	1.7339e-05	0.00019476	0.00030104	0.00040391	0.00050853	0.000
16	1.7381	0	8.1464e-06	6.6555e-05	0.00014925	0.00025602	0.00035404	0.000
17	2.1094	0	6.1188e-06	3.5994e-05	9.5417e-05	0.000162	0.00023392	0.000
18	2.5301	0	1.3448e-05	3.7101e-05	7.3464e-05	0.00012006	0.00017452	0.000
19	3.0034	0	2.2537e-05	4.8195e-05	7.7883e-05	0.00011025	0.00014444	0.000
20	3.5323	0	2.9909e-05	5.5599e-05	8.0928e-05	0.00010452	0.00012891	0.000
21	4.1199	0	3.0433e-05	5.458e-05	7.2693e-05	9.1664e-05	0.00011169	0.000
22	4.7693	0	2.0391e-05	3.7793e-05	5.5429e-05	7.2296e-05	8.9417e-05	0.000
23	5.4836	0	5.1199e-06	1.8361e-05	3.277e-05	4.8259e-05	6.4907e-05	8.218
24	6.2658	0	7.2528e-07	5.2955e-06	1.4093e-05	2.6887e-05	4.122e-05	5.674
25	7.1191	0	1.0524e-07	1.2817e-06	4.9228e-06	1.2149e-05	2.2923e-05	3.569
26	8.0466	0	1.7628e-08	5.0295e-07	2.0294e-06	5.2782e-06	1.1173e-05	2.001
27	9.0514	0	3.0056e-09	3.0395e-07	1.0911e-06	2.7755e-06	5.7099e-06	1.060
28	10.136	0	1.1825e-10	1.6421e-07	5.5086e-07	1.5801e-06	3.2779e-06	5.990
29	11.305	0	0	4.8037e-08	2.2122e-07	8.0726e-07	1.8918e-06	3.56
30	12.56	0	0	9.2865e-09	6.9448e-08	3.1086e-07	1.0023e-06	2.100
31	13.905	0	0	1.789e-09	2.077e-08	9.8086e-08	4.7382e-07	1.182
32	15.342	0	0	4.0984e-10	6.2012e-09	3.4485e-08	1.8776e-07	6.261
33	16.875	0	0	9.8855e-11	1.6718e-09	1.2956e-08	6.109e-08	3.03
34	18.507	0	0	2.1171e-11	4.7002e-10	4.2475e-09	2.1551e-08	1.251
35	20.241	0	0	8.4937e-13	1.3772e-10	1.2013e-09	8.274e-09	4.096
36	22.08	0	0	0	2.9206e-11	3.623e-10	2.7973e-09	1.389
37	24.027	0	0	0	3.6378e-12	1.1269e-10	8.3244e-10	5.364
38	26.085	0	0	0	7.7367e-13	2.3608e-11	2.7013e-10	1.817
39	28.258	0	0	0	1.7753e-13	3.9993e-12	8.0062e-11	5.768
40	30.548	0	0	0	8.3602e-15	1.0518e-12	1.7382e-11	1.866
41	32.959	0	0	0	0	1.9415e-13	3.6072e-12	5.371
42	35.493	0	0	0	0	1.4615e-14	9.1506e-13	1.193
43	38.154	0	0	0	0	2.3455e-15	1.4921e-13	2.800
44	40.945	0	0	0	0	2.9499e-16	1.7186e-14	6.544
45	43.868	0	0	0	0	6.0398e-18	3.2421e-15	1.046
46	46.928	0	0	0	0	0	3.3945e-16	1.583
47	50.126	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_aage{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_inc_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
    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'              }       0.13166      0.63405        -31.11        1.9854         0.71265       0.22832
    {'unweighted_sum'    }    1.0934e+07    8.5358e+05    -2.1835e+06            21           15541        5033.3
    {'sd'                }       0.34823      0.37905         29.813        1.0848         0.54567       0.56949
    {'coefofvar'         }         2.645      0.59783       -0.95831       0.54639         0.76569        2.4943
    {'gini'              }       0.77092      0.31105       -0.47974         0.268         0.36259       0.84016
```

{'min' }	0	0.036717	-586.22	1	0.038108	0	
{'max' }	145.07	10.212	24.63	6	13.784	10.368	
{'pYis0' }	0.10805	0	0	0	0	0.52499	
{'pYls0' }	0	0	0.93414	0	0	0	
{'pYgr0' }	0.89195	1	0.065859	1	1	0.47501	
{'pYisMINY' }	0.10805	1.3288e-05	5.8837e-08	0.41786	2.5312e-05	0.52499	
{'pYisMAXY' }	0	0	0	0.0060544	0	3.9814e-08	
{'p0_01' }	0	0.047727	-322.58	1	0.046651	0	
{'p10' }	0	0.24819	-67.491	1	0.23528	0	
{'p25' }	0.012186	0.36957	-41.871	1	0.35258	0	
{'p50' }	0.032959	0.55272	-24.354	2	0.56523	0	
{'p75' }	0.07477	0.80089	-11.18	3	0.90612	0.24502	
{'p90' }	0.47812	1.1198	-2.6906	4	1.3579	0.84753	
{'p99_99' }	5.4504	3.6593	17.393	6	6.8484	8.2655	
{'fl_cov_ap_ss' }	0.12126	0.055072	2.4507	0.026881	0.05	0.18249	
{'fl_cor_ap_ss' }	1	0.41721	0.23606	0.071158	0.26313	0.92021	
{'fl_cov_c_ss' }	0.055072	0.14368	8.0391	0.07643	0.18689	0.071644	
{'fl_cor_c_ss' }	0.41721	1	0.71138	0.18587	0.90355	0.33189	
{'fl_cov_v_ss' }	2.4507	8.0391	888.8	0.38384	10.004	3.4658	
{'fl_cor_v_ss' }	0.23606	0.71138	1	0.011868	0.61498	0.20413	
{'fl_cov_n_ss' }	0.026881	0.07643	0.38384	1.1768	1.1384e-17	0.13323	
{'fl_cor_n_ss' }	0.071158	0.18587	0.011868	1	1.9231e-17	0.21565	
{'fl_cov_y_head_inc' }	0.05	0.18689	10.004	1.1384e-17	0.29776	0.010455	
{'fl_cor_y_head_inc' }	0.26313	0.90355	0.61498	1.9231e-17	1	0.033645	
{'fl_cov_y_spouse' }	0.18249	0.071644	3.4658	0.13323	0.010455	0.32432	
{'fl_cor_y_spouse' }	0.92021	0.33189	0.20413	0.21565	0.033645	1	
{'fl_cov_yshr_wage' }	1.2236e-32	7.3426e-32	-4.4373e-30	-3.4513e-31	1.4096e-31	3.6232e-32	
{'fl_cor_yshr_wage' }	1.0549e-16	5.8159e-16	-4.4688e-16	-9.5519e-16	7.7559e-16	1.9101e-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'}	0.0057457	0.011176	0.85848	0.007516	0.01319	0.008688	
{'fl_cor_yshr_nttxss'}	0.48632	0.86907	0.84874	0.20421	0.71249	0.44966	
{'fracByP0_01' }	0	7.1684e-06	0.0013012	0.21046	7.788e-06	0	
{'fracByP10' }	0	0.030643	0.32088	0.21046	0.027495	0	
{'fracByP25' }	0.0067356	0.10365	0.58193	0.21046	0.092606	0	
{'fracByP50' }	0.04689	0.29058	0.83099	0.53024	0.26377	0	
{'fracByP75' }	0.13162	0.54875	0.97426	0.77109	0.5245	0.12959	
{'fracByP90' }	0.35822	0.76944	1.0077	0.92834	0.74403	0.33886	
{'fracByP99_99' }	0.99575	0.99938	1.0001	1	0.99912	0.99627	

age =59
xxx tb_outcomes: all stats xxx

OriginalVariableNames	ap_ss	c_ss	v_ss	n_ss	y_head_inc	y_spouse	y
{'mean' }	9.4506	1.2067	-9.9431	1.7239	1.6033	0.44918	
{'unweighted_sum' }	1.1247e+07	1.0819e+06	-3.4419e+05	21	45380	9578.2	1
{'sd' }	9.4598	0.76797	14.834	0.90777	1.2742	1.113	
{'coefofvar' }	1.001	0.63643	-1.4919	0.52659	0.79474	2.4779	
{'gini' }	0.48835	0.32979	-0.78368	0.23461	0.38321	0.83796	
{'min' }	0	0.05663	-208.18	1	0.059541	0	
{'max' }	158.43	12.311	14.965	6	23.47	20.112	
{'pYis0' }	0.0059691	0	0	0	0	0.52499	
{'pYls0' }	0	0	0.73383	0	0	0	
{'pYgr0' }	0.99403	1	0.26617	1	1	0.47501	
{'pYisMINY' }	0.0059691	9.8324e-06	2.9687e-09	0.48835	9.8989e-06	0.52499	1
{'pYisMAXY' }	9.0457e-09	3.8325e-11	5.2662e-07	0.0036816	1.4683e-06	3.6378e-08	
{'p0_01' }	0	0.07838	-101	1	0.08341	0	
{'p10' }	1.0833	0.41297	-30.14	1	0.49019	0	
{'p25' }	3.0034	0.65765	-16.23	1	0.7717	0	
{'p50' }	6.7818	1.0568	-6.363	2	1.2612	0	
{'p75' }	12.812	1.5534	0.45344	2	2.0256	0.48062	
{'p90' }	20.8	2.1542	4.9139	3	3.0996	1.7714	
{'p99_99' }	112.23	8.4857	13.926	6	15.937	16.033	
{'fl_cov_ap_ss' }	89.487	6.8831	97.649	0.8159	10.409	2.2143	

{'fl_cor_ap_ss' }	1	0.94746	0.69588	0.095013	0.86354	0.2103	
{'fl_cov_c_ss' }	6.8831	0.58977	8.5503	0.23192	0.85197	0.24542	
{'fl_cor_c_ss' }	0.94746	1	0.75055	0.33267	0.87063	0.28712	
{'fl_cov_v_ss' }	97.649	8.5503	220.04	2.4373	12.623	3.4887	
{'fl_cor_v_ss' }	0.69588	0.75055	1	0.181	0.66782	0.21131	
{'fl_cov_n_ss' }	0.8159	0.23192	2.4373	0.82404	0.055267	0.27625	
{'fl_cor_n_ss' }	0.095013	0.33267	0.181	1	0.04778	0.27342	
{'fl_cov_y_head_inc' }	10.409	0.85197	12.623	0.055267	1.6237	0.116	
{'fl_cor_y_head_inc' }	0.86354	0.87063	0.66782	0.04778	1	0.08179	
{'fl_cov_y_spouse' }	2.2143	0.24542	3.4887	0.27625	0.116	1.2388	
{'fl_cor_y_spouse' }	0.2103	0.28712	0.21131	0.27342	0.08179	1	
{'fl_cov_yshr_wage' }	-0.54196	-0.036396	-0.86915	0.0011758	-0.038212	0.020434	
{'fl_cor_yshr_wage' }	-0.56735	-0.46933	-0.58024	0.012827	-0.29697	0.18181	
{'fl_cov_yshr_SS' }	0	0	0	0	0	0	
{'fl_cor_yshr_SS' }	NaN	NaN	NaN	NaN	NaN	NaN	
{'fl_cov_yshr_nttxss'}	0.19452	0.017952	0.42036	0.0075501	0.027003	0.013338	-0
{'fl_cor_yshr_nttxss'}	0.67266	0.7647	0.92699	0.27208	0.69323	0.39202	
{'fracByP0_01' }	0	6.8812e-06	0.0011212	0.28329	5.8341e-06	0	3
{'fracByP10' }	0.004897	0.026408	0.43931	0.28329	0.022426	0	
{'fracByP25' }	0.037048	0.092569	0.77208	0.28329	0.081818	0	
{'fracByP50' }	0.16368	0.27051	1.0414	0.72028	0.23952	0	
{'fracByP75' }	0.41532	0.53706	1.1137	0.72028	0.48823	0.13542	
{'fracByP90' }	0.67288	0.76168	1.075	0.85389	0.72007	0.34015	
{'fracByP99_99' }	0.99866	0.99926	1.0001	1	0.99889	0.99665	

age =100
xxx tb_outcomes: all stats xxx

OriginalVariableNames	**ap_ss**	**c_ss**	**v_ss**	**n_ss**	**y_head_inc**	**y_spouse**	**yshr_w**
{'mean' }	0	0.34868	-3.0033	1.4797	0.2604	0.10125	0.1
{'unweighted_sum' }	0	1.2188e+05	458.94	21	213.14	33.546	81
{'sd' }	0	0.23392	1.043	0.50567	0.02289	0.24772	0
{'coefofvar' }	NaN	0.67088	-0.34728	0.34173	0.087904	2.4467	1.
{'gini' }	NaN	0.275	-0.17693	0.12034	0.041151	0.7872	0.7
{'min' }	0	0.2179	-10.065	1	0.24433	0	
{'max' }	0	141.66	0.99282	6	5.6926	3.115	0.9
{'pYis0' }	1	0	0	0	0	0.52499	0.5
{'pYls0' }	0	0	0.99285	0	0	0	
{'pYgr0' }	0	1	0.0071501	1	1	0.47501	0.4
{'pYisMINY' }	1	0.36483	1.5455e-10	0.5232	0.52813	0.52499	0.5
{'pYisMAXY' }	1	0	0	4.2206e-08	0	1.0335e-08	5.4646
{'p0_01' }	0	0.2179	-6.3349	1	0.24433	0	
{'p10' }	0	0.2179	-3.6603	1	0.24433	0	
{'p25' }	0	0.2179	-3.5892	1	0.24433	0	
{'p50' }	0	0.25824	-3.5892	1	0.24433	0	
{'p75' }	0	0.36458	-2.8095	2	0.29263	0.10311	0.2
{'p90' }	0	0.6134	-1.3055	2	0.29279	0.49115	0.6
{'p99_99' }	0	2.8989	0.51215	4	0.33789	2.9458	0.9
{'fl_cov_ap_ss' }	0	0	0	0	0	0	
{'fl_cor_ap_ss' }	NaN	NaN	NaN	NaN	NaN	NaN	
{'fl_cov_c_ss' }	0	0.054721	0.19746	0.059476	0.0015551	0.05178	0.03
{'fl_cor_c_ss' }	NaN	1	0.80934	0.50281	0.29042	0.89356	0.
{'fl_cov_v_ss' }	0	0.19746	1.0878	0.16711	0.01031	0.1649	0.1
{'fl_cor_v_ss' }	NaN	0.80934	1	0.31686	0.43183	0.63823	0.6
{'fl_cov_n_ss' }	0	0.059476	0.16711	0.2557	0.0019105	0.0533	0.08
{'fl_cor_n_ss' }	NaN	0.50281	0.31686	1	0.16506	0.4255	0.7
{'fl_cov_y_head_inc' }	0	0.0015551	0.01031	0.0019105	0.00052397	0.00067518	0.0006
{'fl_cor_y_head_inc' }	NaN	0.29042	0.43183	0.16506	1	0.11907	0.1
{'fl_cov_y_spouse' }	0	0.05178	0.1649	0.0533	0.00067518	0.061365	0.04
{'fl_cor_y_spouse' }	NaN	0.89356	0.63823	0.4255	0.11907	1	0.7
{'fl_cov_yshr_wage' }	0	0.039513	0.15927	0.083913	0.00067571	0.042915	0.04
{'fl_cor_yshr_wage' }	NaN	0.7643	0.69097	0.75087	0.13357	0.78388	
{'fl_cov_yshr_SS' }	0	-0.040547	-0.16461	-0.085285	-0.00072523	-0.042963	-0.04
{'fl_cor_yshr_SS' }	NaN	-0.77966	-0.70991	-0.75864	-0.14251	-0.78011	-0.9

{'fl_cov_yshr_nttxss'}	0	0.044511	0.18091	0.091879	0.00087698	0.047226	0.05
{'fl_cor_yshr_nttxss'}	NaN	0.78763	0.71798	0.75212	0.15859	0.78914	0.9
{'fracByP0_01' }	NaN	0.22799	0.00053042	0.35357	0.49553	0	
{'fracByP10' }	NaN	0.22799	0.22059	0.35357	0.49553	0	
{'fracByP25' }	NaN	0.22799	0.6552	0.35357	0.49553	0	
{'fracByP50' }	NaN	0.35394	0.6552	0.35357	0.49553	0	
{'fracByP75' }	NaN	0.55083	0.87677	0.99419	0.88359	0.19257	0.3
{'fracByP90' }	NaN	0.7612	0.97549	0.99419	0.89158	0.62793	0.5
{'fracByP99_99' }	NaN	0.99927	1	0.99999	0.99991	0.9996	0.9