2020 V and C with Unemployment

This is the example vignette for function: **snw_a4chk_unemp_bisec_vec** from the **PrjOptiSNW Package.**
This function solves for the V(states, check) for individuals working. Dense solution. Bisection, most time for the test here taken to genereate the income matrixes. But these can be generated out of the check loops.

Test SNW_A4CHK_UNEMP_BISEC_VEC Defaults Dense

Call the function with defaults.

```
mp_params = snw_mp_param('default_docdense');
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_timer') = true;
[V_ss,~,cons_ss,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Elapsed time is 117.306855 seconds.
Completed SNW_VFI_MAIN;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test
```

```
welf_checks = 2;
xi=0.5;
b=0;
TR = 100/58056;
mp_params('TR') = TR;
mp_params('xi') = xi;
mp_params('b') = b;
[V_unemp,~,cons_unemp,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Elapsed time is 118.125995 seconds.
Completed SNW_VFI_MAIN 1 PERIOD UNEMP SHK;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test
```

```
[V_U, C_U] = snw_a4chk_unemp_bisec_vec(welf_checks, V_unemp, cons_unemp, mp_params, mp_controls
```

```
Elapsed time is 65.195251 seconds.
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=2;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_container_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std
C_U	1	1	6	1.9173e+06	83	23100	9.1143e+06	4.7537	8.3522
C_U_minus_C_unemp	2	2	6	1.9173e+06	83	23100	1264.8	0.00065966	0.00093317
V_U	3	3	6	1.9173e+06	83	23100	-4.8023e+06	-2.5047	18.304
V_U_minus_V_unemp	4	4	6	1.9173e+06	83	23100	20117	0.010492	0.041351
mn_MPC_unemp	5	5	6	1.9173e+06	83	23100	3.6714e+05	0.19149	0.27088

```
mn_V_U_gain_check = V_U - V_unemp;
mn_MPC_U_gain_share_check = (C_U - cons_unemp)./(welf_checks*mp_params('TR'));
```

Dense Param Results Define Frames

Define the matrix dimensions names and dimension vector values. 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 Difference in V and C with Check

The difference between V and V with Check, marginal utility gain given the check.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'eastoutside';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 21; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(MN_V_GAIN_CHECK(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
ar_permute = [1,4,5,6,3,2];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str(m
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc
```

```
xxx  MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mean_eta_6    mean_
    _____    _________    _________    _________    _________    _________    _________    _________    _____

      1              0       0.30204      0.14591      0.083016     0.058265     0.048632     0.044856     0.0
      2     0.00085734       0.29687      0.14443       0.08243     0.057893      0.04831     0.044545     0.0
      3      0.0068587       0.24517      0.12826      0.075796     0.053707     0.044719      0.04109     0.0
      4       0.023148       0.17496      0.10278      0.064774     0.046886        0.039     0.035634     0.0
      5        0.05487       0.11411     0.075197       0.05168     0.038779     0.032381     0.029426     0.0
      6        0.10717       0.07945     0.053698       0.03981     0.031363     0.026553     0.024076      0.
      7        0.18519      0.061381     0.041221      0.031035     0.025388     0.021951     0.019956     0.0
      8        0.29407      0.049672     0.033665      0.025164     0.020645     0.018059     0.016478     0.0
      9        0.43896       0.04097     0.028356      0.021184     0.017207     0.014985      0.01368     0.0
     10          0.625       0.03421     0.024175      0.018158     0.014629     0.012593     0.011418      0.
     11        0.85734      0.028804     0.020748      0.015685     0.012564     0.010694    0.0095968    0.00
     12         1.1411      0.024415     0.017898      0.013612     0.010849    0.0091391    0.0081091    0.00
```

13	1.4815	0.020781	0.015487	0.011845	0.009402	0.0078443	0.0068811	0.00
14	1.8836	0.017732	0.013429	0.01033	0.0081694	0.0067529	0.005857	0.0
15	2.3525	0.015155	0.011653	0.0090184	0.0071161	0.005829	0.0049994	0.00
16	2.8935	0.012965	0.010113	0.0078806	0.0062113	0.0050446	0.0042773	0.00
17	3.5117	0.011102	0.0087778	0.0068893	0.0054304	0.0043768	0.0036705	0.0
18	4.2121	0.0095159	0.0076201	0.0060242	0.0047539	0.0038073	0.0031598	0.00
19	5	0.0081617	0.0066157	0.0052682	0.0041656	0.0033198	0.0027281	0.00
20	5.8805	0.0070048	0.0057428	0.0046071	0.003653	0.002901	0.0023622	0.00
21	6.8587	0.0060148	0.004985	0.0040287	0.0032059	0.0025406	0.0020515	0.00
22	7.9398	0.0051679	0.0043275	0.0035243	0.0028158	0.0022296	0.0017872	0.00
23	9.1289	0.0044436	0.0037576	0.0030842	0.0024752	0.0019606	0.0015619	0.00
24	10.431	0.0038243	0.0032641	0.0027001	0.002178	0.0017273	0.0013695	0.00
25	11.852	0.0032946	0.0028369	0.0023652	0.0019184	0.0015245	0.0012044	0.000
26	13.396	0.0028417	0.0024674	0.0020734	0.0016914	0.0013477	0.0010623	0.000
27	15.069	0.0024542	0.0021479	0.001819	0.0014925	0.0011933	0.00093956	0.000
28	16.875	0.0021225	0.0018716	0.0015969	0.0013182	0.0010581	0.00083322	0.000
29	18.82	0.0018382	0.0016324	0.0014031	0.0011655	0.00093959	0.00074067	0.000
30	20.91	0.0015944	0.0014253	0.0012338	0.0010316	0.00083546	0.00065979	0.000
31	23.148	0.001385	0.0012459	0.001086	0.00091386	0.00074382	0.00058888	0.000
32	25.541	0.0012051	0.0010904	0.00095674	0.00081028	0.00066306	0.00052655	0.00
33	28.093	0.0010502	0.00095561	0.00084375	0.00071923	0.0005918	0.00047161	0.000
34	30.81	0.00091678	0.00083855	0.00074491	0.00063906	0.00052877	0.00042302	0.000
35	33.697	0.00080165	0.00073685	0.00065837	0.00056838	0.00047287	0.00037994	0.000
36	36.758	0.00070216	0.0006484	0.00058255	0.000506	0.00042333	0.0003417	0.000
37	40	0.0006161	0.00057139	0.00051608	0.00045093	0.00037946	0.00030772	0.000
38	43.427	0.00054153	0.00050427	0.00045774	0.00040227	0.0003405	0.00027746	0.000
39	47.044	0.00047681	0.0004457	0.0004065	0.00035923	0.00030581	0.00025047	0.000
40	50.856	0.00042057	0.00039453	0.00036145	0.00032113	0.0002749	0.00022636	0.000
41	54.87	0.00037159	0.00034976	0.00032182	0.00028739	0.0002474	0.0002048	0.000
42	59.089	0.00032891	0.00031056	0.00028689	0.00025747	0.00022288	0.00018547	0.000
43	63.519	0.00029162	0.00027616	0.00025609	0.00023092	0.00020099	0.00016812	0.000
44	68.164	0.000259	0.00024595	0.00022891	0.00020735	0.00018142	0.00015254	0.000
45	73.032	0.00023043	0.0002194	0.00020489	0.00018639	0.00016392	0.00013855	0.000
46	78.125	0.00020535	0.00019599	0.00018363	0.00016774	0.00014826	0.00012597	0.000
47	83.45	0.0001833	0.00017535	0.00016479	0.00015113	0.00013422	0.00011464	9.394
48	89.011	0.00016388	0.00015711	0.00014808	0.00013632	0.00012163	0.00010442	8.599
49	94.815	0.00014675	0.00014099	0.00013325	0.00012311	0.00011033	9.5184e-05	7.878
50	100.87	0.00013164	0.00012671	0.00012006	0.0001113	0.00010017	8.6848e-05	7.226
51	107.17	0.00011825	0.00011403	0.00010832	0.00010074	9.1041e-05	7.9314e-05	6.635
52	113.73	0.00010639	0.00010277	9.7844e-05	9.1281e-05	8.2823e-05	7.2501e-05	6.100
53	120.55	9.586e-05	9.2745e-05	8.8498e-05	8.2806e-05	7.5422e-05	6.6343e-05	5.620
54	127.64	8.65e-05	8.3818e-05	8.0149e-05	7.5211e-05	6.877e-05	6.0813e-05	5.203
55	135	8.65e-05	8.3818e-05	8.0149e-05	7.5211e-05	6.877e-05	6.0814e-05	5.203

```
% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 4, 1, cl_mp_dat
```

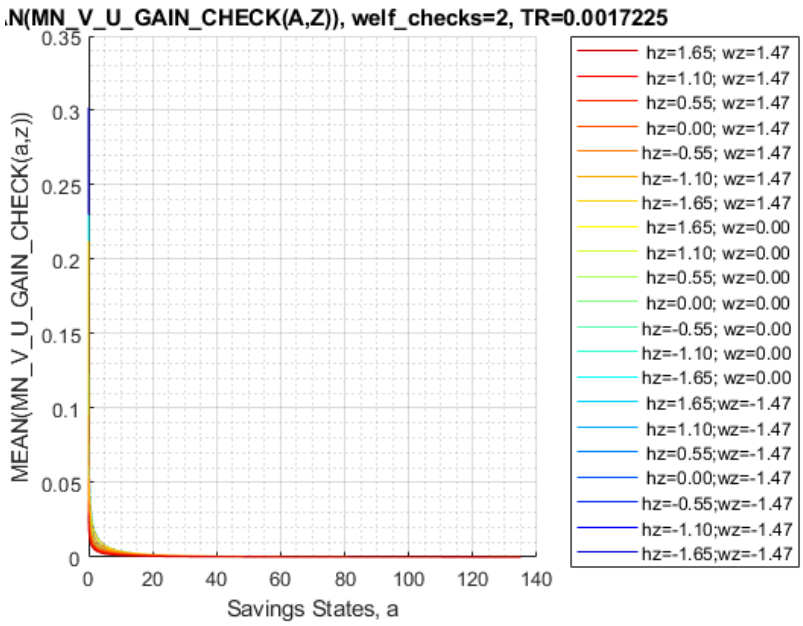
```
xxx  MEAN(MN_MPC_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
```

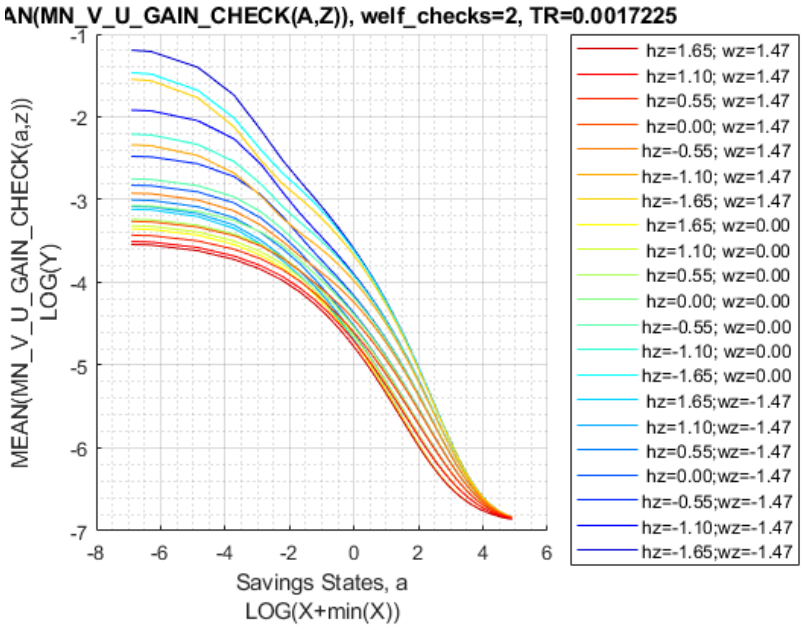
group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5	mean_eta_6	mean_
1	0	0.99676	0.99676	0.98968	0.96861	0.936	0.90173	0.85
2	0.00085734	0.99564	0.99564	0.9885	0.96775	0.93479	0.90066	0.85
3	0.0068587	0.96608	0.96608	0.96111	0.93769	0.90391	0.8643	0.82
4	0.023148	0.9144	0.91442	0.90273	0.87976	0.84669	0.80892	0.77
5	0.05487	0.83014	0.8675	0.85172	0.82953	0.79856	0.76121	0.72
6	0.10717	0.57928	0.6794	0.72166	0.70603	0.67574	0.64644	0.60
7	0.18519	0.40677	0.4866	0.57883	0.62471	0.61322	0.58844	0.5
8	0.29407	0.30075	0.32748	0.40471	0.48963	0.53386	0.52857	0.50
9	0.43896	0.23989	0.24678	0.26581	0.32826	0.40159	0.44032	0.42
10	0.625	0.20277	0.20016	0.20528	0.22484	0.27796	0.33388	0.34
11	0.85734	0.17483	0.17446	0.16482	0.16705	0.18499	0.22378	0.2
12	1.1411	0.15256	0.15408	0.15362	0.14656	0.14374	0.15146	0.16

13	1.4815	0.13901	0.14193	0.1396	0.13759	0.13213	0.12942	0.13
14	1.8836	0.13445	0.13173	0.13532	0.13183	0.12918	0.12469	0.12
15	2.3525	0.13073	0.12804	0.12598	0.12513	0.12402	0.12234	0.11
16	2.8935	0.12611	0.12259	0.11944	0.11919	0.11786	0.11735	0.11
17	3.5117	0.11866	0.11921	0.11564	0.11369	0.11418	0.11273	0.11
18	4.2121	0.11801	0.11572	0.11569	0.11296	0.11197	0.1131	0.11
19	5	0.11495	0.1165	0.11547	0.11464	0.11333	0.11355	0.11
20	5.8805	0.11283	0.11351	0.11397	0.11371	0.11268	0.11257	0.11
21	6.8587	0.11082	0.11067	0.11238	0.11122	0.11137	0.11149	0.11
22	7.9398	0.10947	0.10939	0.10939	0.11073	0.1094	0.1107	0.11
23	9.1289	0.11048	0.11044	0.11055	0.11147	0.1111	0.11147	0.11
24	10.431	0.10942	0.10943	0.10956	0.10966	0.1103	0.10999	0.11
25	11.852	0.10713	0.10714	0.10723	0.1074	0.10779	0.10791	0.10
26	13.396	0.10662	0.10662	0.10669	0.10689	0.10696	0.10733	0.10
27	15.069	0.10897	0.10898	0.10904	0.10931	0.10944	0.10955	0.11
28	16.875	0.11044	0.11044	0.1105	0.11079	0.11094	0.11077	0.11
29	18.82	0.1091	0.10911	0.10916	0.10934	0.10959	0.10939	0.11
30	20.91	0.10634	0.10635	0.10639	0.10655	0.10686	0.10668	0.10
31	23.148	0.10594	0.10594	0.10599	0.10622	0.10646	0.10635	0.10
32	25.541	0.10777	0.10778	0.10783	0.10802	0.10823	0.10822	0.10
33	28.093	0.10798	0.10799	0.10803	0.10814	0.10838	0.10849	0.10
34	30.81	0.10767	0.10768	0.10771	0.1078	0.10808	0.10825	0.10
35	33.697	0.10814	0.10815	0.10818	0.10827	0.10861	0.10878	0.10
36	36.758	0.10925	0.10925	0.10928	0.10936	0.10964	0.10987	0.10
37	40	0.10756	0.10757	0.10759	0.10766	0.10783	0.10814	0.10
38	43.427	0.1062	0.10621	0.10623	0.10629	0.10646	0.1068	0.10
39	47.044	0.10582	0.10583	0.10586	0.10592	0.10616	0.10642	0.10
40	50.856	0.10829	0.1083	0.10833	0.10839	0.10861	0.10882	0.10
41	54.87	0.10898	0.10899	0.10902	0.10908	0.10921	0.10945	0.10
42	59.089	0.10774	0.10775	0.10777	0.10782	0.10792	0.1082	0.10
43	63.519	0.10666	0.10668	0.1067	0.10674	0.10683	0.10717	0.10
44	68.164	0.1073	0.10731	0.10734	0.10738	0.10746	0.10775	0.10
45	73.032	0.1085	0.10851	0.10853	0.10857	0.10864	0.10885	0.10
46	78.125	0.10779	0.1078	0.10782	0.10785	0.10792	0.10807	0.10
47	83.45	0.10631	0.10632	0.10634	0.10637	0.10643	0.10659	0.10
48	89.011	0.10666	0.10667	0.10668	0.10671	0.10677	0.10698	0.10
49	94.815	0.10809	0.1081	0.10811	0.10814	0.1082	0.10838	0.10
50	100.87	0.10807	0.10808	0.10809	0.10811	0.10816	0.10829	0.10
51	107.17	0.10728	0.10729	0.1073	0.10732	0.10737	0.10745	0.10
52	113.73	0.10761	0.10762	0.10763	0.10764	0.10768	0.10772	0.10
53	120.55	0.10824	0.10825	0.10825	0.10827	0.10829	0.10822	0.10
54	127.64	0.10739	0.10738	0.10738	0.10736	0.10728	0.10683	0.10
55	135	0.10739	0.10738	0.10738	0.10736	0.10728	0.10683	0.10

Graph Mean Values:

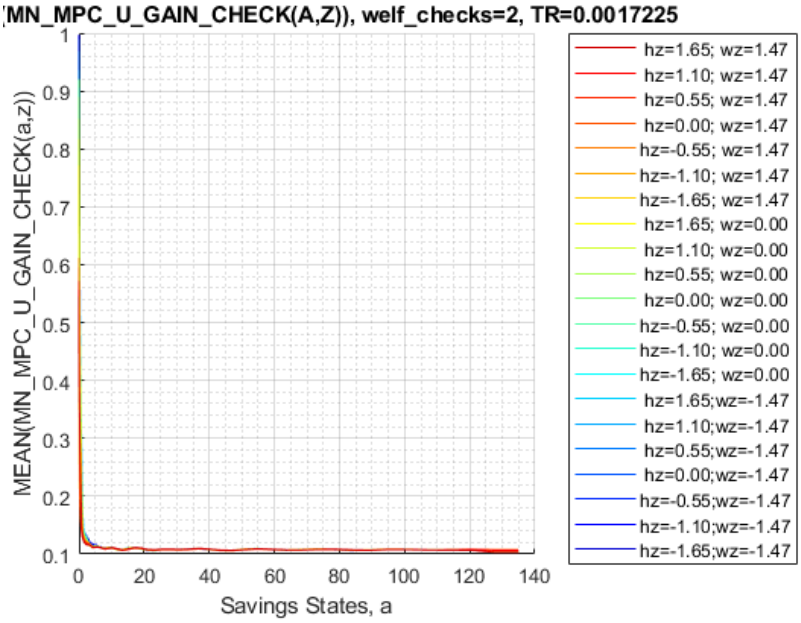
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

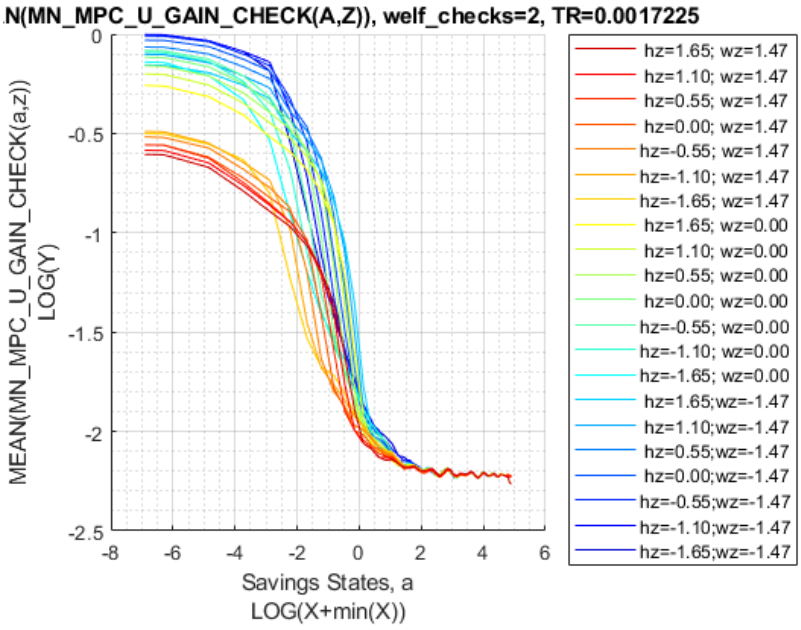




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' nu
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





Analyze Kids and Marriage and Age

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = [...
    "k0M0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "k0M1", "K1M1", "K2M1", "K3M1", "K4M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
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'};
```

MEAN(VAL(KM,J)), MEAN(AP(KM,J)), MEAN(C(KM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc
```

xxx MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225 xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	kids	marry	mean_age_18	mean_age_19	mean_age_20	mean_age_21	mean_age_22	mean_age_23
1	1	0	0.018051	0.017729	0.017395	0.015889	0.014636	0.013585
2	2	0	0.024841	0.024418	0.023963	0.021844	0.020076	0.01859
3	3	0	0.029625	0.029186	0.028697	0.026156	0.024037	0.022256
4	4	0	0.033827	0.033361	0.03283	0.029923	0.0275	0.025462
5	5	0	0.037408	0.03694	0.036392	0.033181	0.030504	0.028255
6	1	1	0.0061332	0.0057595	0.0054053	0.0049063	0.004487	0.0041337
7	2	1	0.0083066	0.0078178	0.007352	0.0066702	0.0060896	0.0056017
8	3	1	0.010082	0.0095157	0.0089749	0.0081404	0.007436	0.0068415
9	4	1	0.012214	0.011567	0.010941	0.0099258	0.0090678	0.0083464
10	5	1	0.015297	0.014584	0.013875	0.012596	0.011524	0.010618

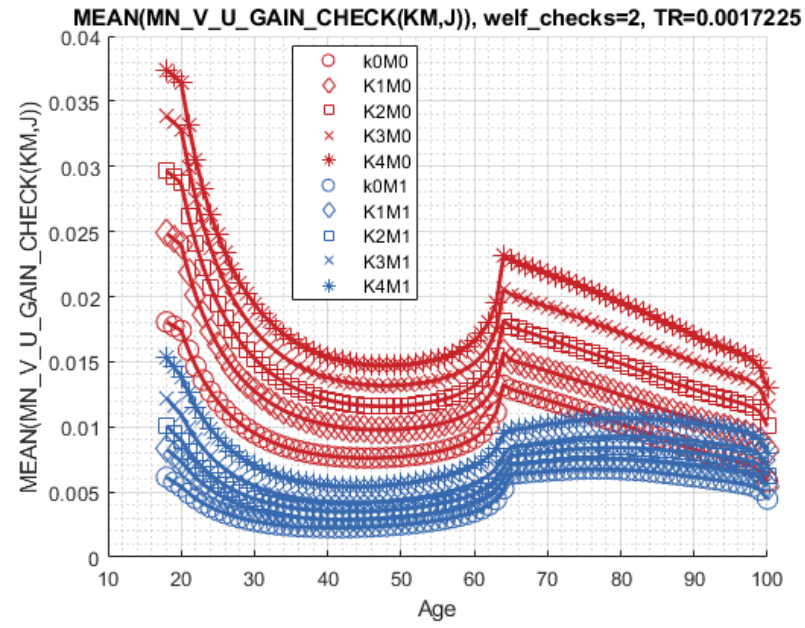
```
% Consumption Function
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2st
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_dat
```

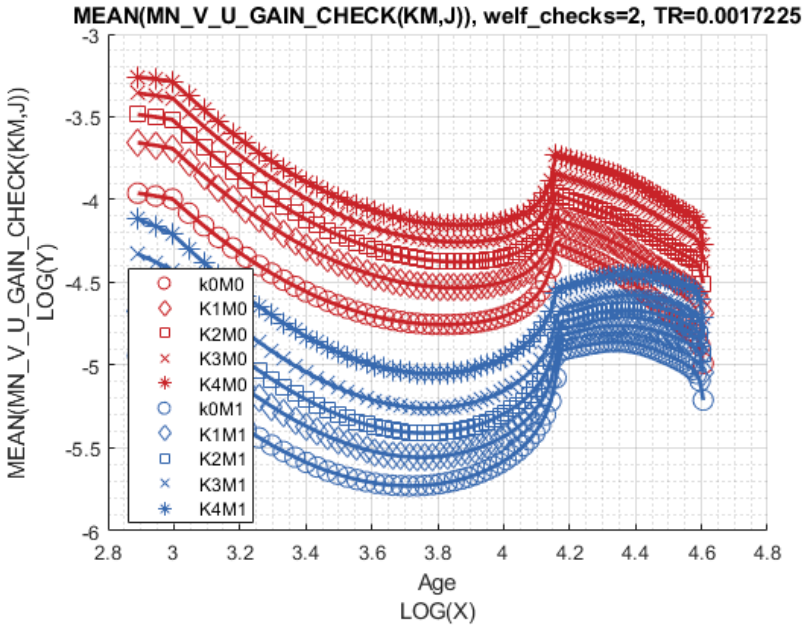
xxx MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225 xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	kids	marry	mean_age_18	mean_age_19	mean_age_20	mean_age_21	mean_age_22	mean_age_23
1	1	0	0.15281	0.15636	0.16001	0.16042	0.16057	0.16052
2	2	0	0.16175	0.1653	0.16916	0.17012	0.17091	0.17153
3	3	0	0.16911	0.17246	0.1763	0.17733	0.17815	0.17883
4	4	0	0.17289	0.17613	0.17991	0.18092	0.18173	0.18239
5	5	0	0.1764	0.17944	0.18311	0.18398	0.18469	0.18528
6	1	1	0.13726	0.14236	0.14341	0.14404	0.14148	0.14159
7	2	1	0.14642	0.14777	0.14963	0.15056	0.14937	0.14713
8	3	1	0.15601	0.15841	0.1601	0.16253	0.15944	0.1588
9	4	1	0.16238	0.1649	0.16712	0.16745	0.1673	0.16745
10	5	1	0.17026	0.1742	0.17738	0.17809	0.17684	0.1786

Graph Mean Values:

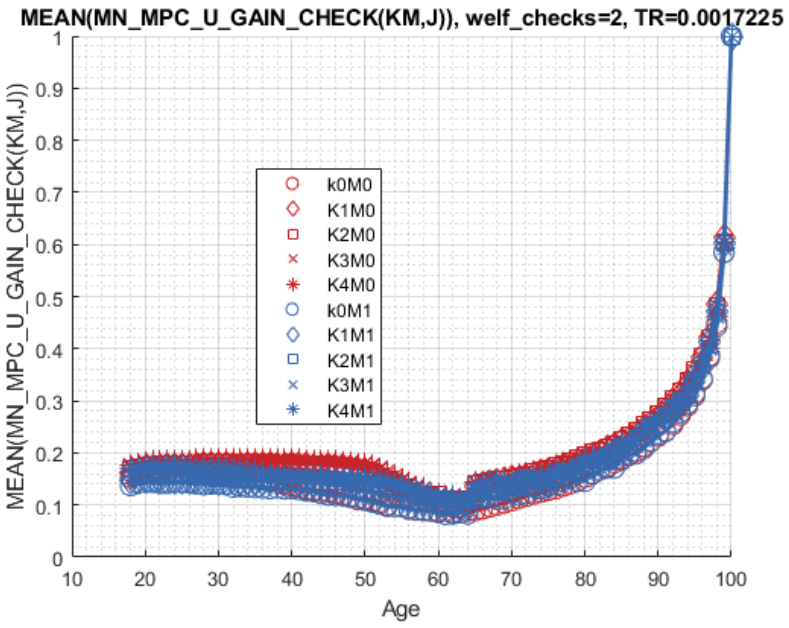
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

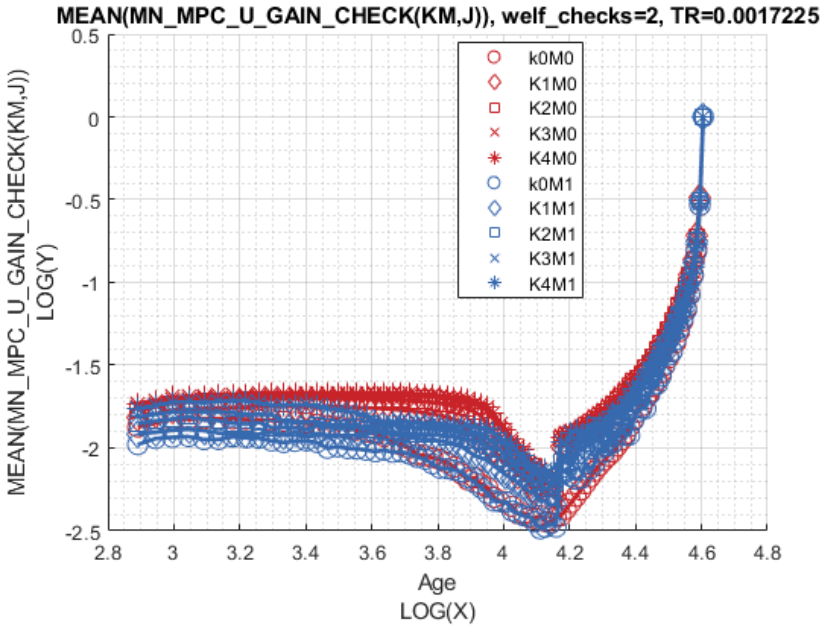




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' r
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





Analyze Education and Marriage

Aggregating over education, savings, and shocks, what are the differential effects of Marriage and Age.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
ar_row_grid = ["E0M0", "E1M0", "E0M1", "E1M1"];
mp_support_graph('cl_st_xtitle') = {'Age'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('st_rounding') = '6.2f'; % format shock legend
mp_support_graph('cl_scatter_shapes') = {'*', 'p', '*','p' };
mp_support_graph('cl_colors') = {'red', 'red', 'blue', 'blue'};
```

MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc
```

```
xxx  MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22    mean_age_23
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________    ___________

      1       0       0         0.029418       0.029071       0.028695       0.027067       0.025622       0.024336
      2       1       0         0.028083       0.027583       0.027015        0.02373       0.021079       0.018923
      3       0       1         0.011215       0.010661       0.010131       0.009383      0.0087272      0.0081586
      4       1       1        0.0095983      0.0090363      0.0084881      0.0075126      0.0067144      0.0060579
```

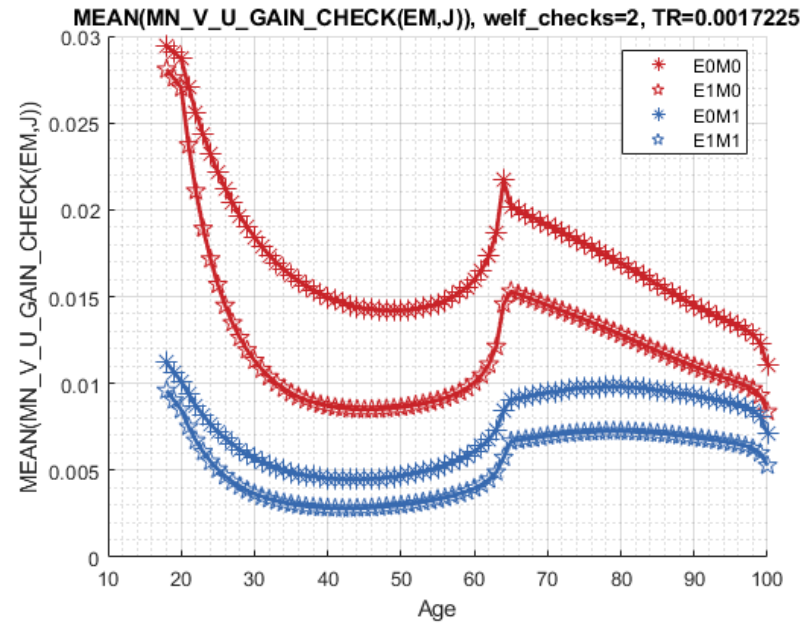
```
% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2st
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_dat
```

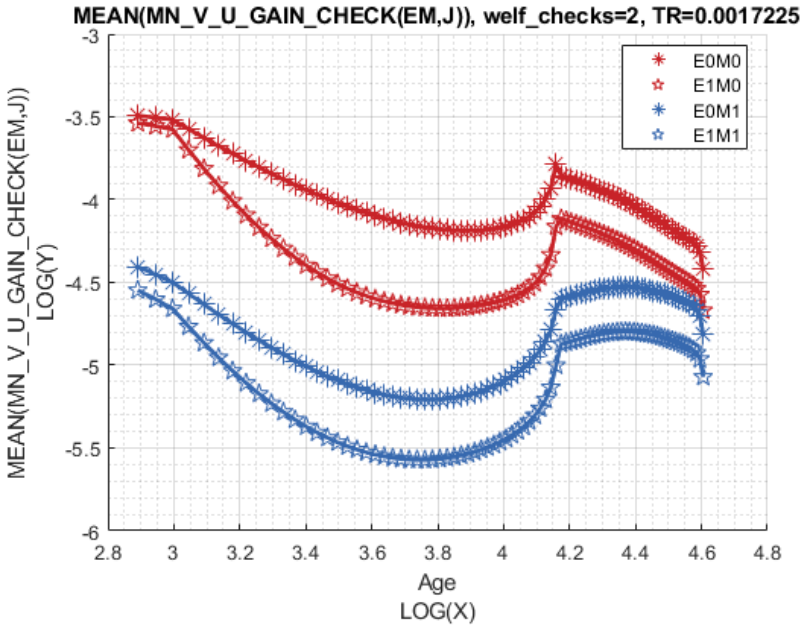
```
xxx  MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22    mean_age_23
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________    ___________

      1       0       0          0.15946        0.16202        0.16481        0.16552        0.16614        0.16668
      2       1       0          0.17373        0.17785        0.18259        0.18359        0.18428        0.18473
      3       0       1          0.14851        0.15124        0.15155        0.15153        0.15098         0.1508
      4       1       1          0.16042        0.16382        0.16751        0.16954        0.16679        0.16663
```

Graph Mean Values:

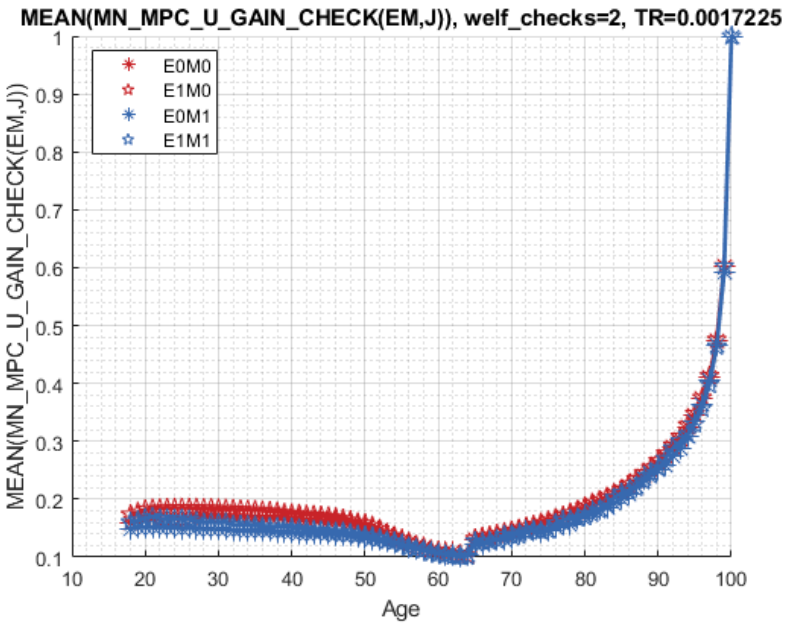
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

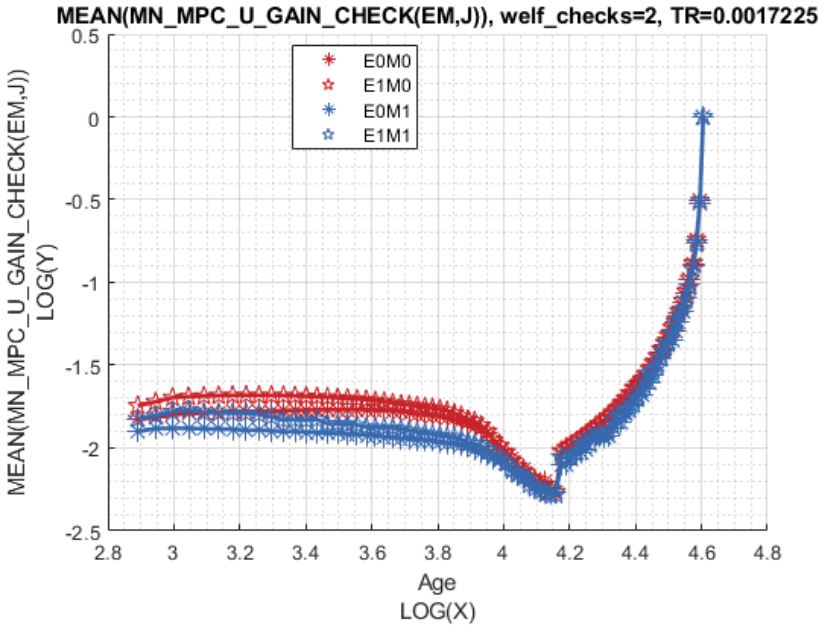




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' r
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```


MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225
E0M0
E1M0
E0M1
E1M1
MEAN(MN_MPC_U_GAIN_CHECK(EM,J))
Age


MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225
E0M0
E1M0
E0M1
E1M1
MEAN(MN_MPC_U_GAIN_CHECK(EM,J))
LOG(Y)
Age
LOG(X)