2008 Value and Optimal Savings and Consumption Given Stimulus

This is the example vignette for function: **snw_v08_jaeemk** from the **PrjOptiSNW Package.** This is similalr to **snw_evuvw20_jaeemk**, but for the 2008 Bush stimulus. **snw_v08p08_jaeemk** already solved for optimal policy and value functions in 2008, given expected unemployment shock in 2009. In this function, given some stimulus amount, we use **snw_a4chk_wrk_bisec_vec** to compute the updated optimal V and C in 2008 given the stimulus amount, based on the alues for V and C without stimulus computed by **snw_v08p08_jaeemk**.

Note that **snw_a4chk_wrk_bisec_vec** computes the adjustment in the savings state that would be equivalent to the increase in stimulus amount (which is not a state variable) to current resources, this is faster than resolving 2008 optimal V and C at specific stimulus check amount levels.

Note **snw_evuvw20_jaeemk** has EVUVW, but here, we only have V08, because in the 2020 problem, households receive checks ex-post of the COVID MIT shocks in 2020 and the EVUVW is the weighted average in V between the MIT unemployed and non-shock employed state. In 2008, however, there are no shocks yet, the state-space is the same as normal, the only difference is that households might receive stimulus checks from Bush. The Bush stimulus is provided ex-ante of the shock realization. The 2009 shocks due to the great recession is not a MIT shock, but expected shock. The effect of the 2009 shock on consumption, savings is solved by **snw_v08p08_jaeemk**. The expectation over shock, in another word, for the **snw_v08_jaeemk** is already included in EV' in 2008 for 2009.

Solve 2008 Value and Policy Function with SNW_V08p08_JAEEMK

Solve for the Value and Policy functions in 2008 given expected unemployment shock that is specific to age and education group in 2009, no stimulus amounts.

First, set various parameters

```
% 1. Paramters
% Parameters
mp_more_inputs = containers.Map('KeyType','char', 'ValueType','any');
mp_more_inputs('fl_ss_non_college') = 0.225;
mp_more_inputs('fl_ss_college') = 0.271;
fl_p50_hh_income_07 = 54831;
mp_more_inputs('fl_scaleconvertor') = fl_p50_hh_income_07;

% st_param_group = 'default_small';
% st_param_group = 'default_dense';
st_param_group = 'default_docdense';
mp_params = snw_mp_param(st_param_group, false, 'tauchen', false, 8, 8, mp_more_inputs);
mp_params('beta') = 0.95;

% Control parameters
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_v08p08_jaeemk') = false;
```

```
mp_controls('bl_print_v08p08_jaeemk_verbose') = false;
mp_controls('bl_print_v08_jaeemk') = true;
mp_controls('bl_print_v08_jaeemk_verbose') = true;
```

Second, solve the steady-state problem, same as employed results in 2009.

```
% 2. Solve value steady state (2009 employed)
[V_VFI_ss, ap_VFI_ss, cons_VFI_ss, mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_co
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=530.633
```

```
inc_VFI = mp_valpol_more_ss('inc_VFI');
spouse_inc_VFI = mp_valpol_more_ss('spouse_inc_VFI');
total_inc_VFI = inc_VFI + spouse_inc_VFI;
V_emp_2009 = V_VFI_ss;
% Solve for probability mass, needed for pre-compute
[Phi_true] = snw_ds_main_vec(mp_params, mp_controls, ap_VFI_ss, cons_VFI_ss, mp_valpol_more_ss)
```

```
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1271.649
```

```
% Solve for household head and spouse income and sum to total income.
inc_VFI = mp_valpol_more_ss('inc_VFI');
spouse_inc_VFI = mp_valpol_more_ss('spouse_inc_VFI');
total_inc_VFI = inc_VFI + spouse_inc_VFI;
```

Third, solve the unemployment problem in 2009. With 2009 specific unemployment parameters calibrated and found from data. Using b calibrated by **snw_calibrate_2009_b**.

```
% 3. Solve value unemployed 2009
% Set Unemployment Related Variables
mp_params('xi') = 0.532;
% Calibrated by snw_calibrate_2009_b
mp_params('b') = 0.37992;
mp_params('a2_covidyr') = mp_params('a2_greatrecession_2009');
mp_params('TR') = 100/fl_p50_hh_income_07; % Value of a stimulus check (can receive multiple ch
[V_unemp_2009] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_VFI_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time
```

Fourth, solve the 2008 problem, with 2008-specific value and policy functions.

```
% 4. Value and Optimal choice in 2009
[V_2008, ap_2008, cons_2008, ev_empshk_2009] = ...
    snw_v08p08_jaeemk(mp_params, mp_controls, V_emp_2009, V_unemp_2009);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time
Completed SNW_V08P08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=522.6257
```

```
% 5. pre-compute
cl_st_precompute_list = {'a', ...
    'inc', 'inc_unemp', 'spouse_inc', 'spouse_inc_unemp', 'ref_earn_wageind_grid'};
mp_controls('bl_print_precompute_verbose') = false;
[mp_precompute_res] = snw_hh_precompute(mp_params, mp_controls, cl_st_precompute_list, ap_VFI_s
```

```
Wage quintile cutoffs=0.4645     0.71528      1.0335      1.5632
SNW_HH_PRECOMPUTE: Finished Age Group:1 of 82, time-this-age:3.2746
```

```
SNW_HH_PRECOMPUTE: Finished Age Group:2 of 82, time-this-age:3.2852
SNW_HH_PRECOMPUTE: Finished Age Group:3 of 82, time-this-age:3.3869
SNW_HH_PRECOMPUTE: Finished Age Group:4 of 82, time-this-age:3.4589
SNW_HH_PRECOMPUTE: Finished Age Group:5 of 82, time-this-age:3.254
SNW_HH_PRECOMPUTE: Finished Age Group:6 of 82, time-this-age:3.3465
SNW_HH_PRECOMPUTE: Finished Age Group:7 of 82, time-this-age:3.4109
SNW_HH_PRECOMPUTE: Finished Age Group:8 of 82, time-this-age:3.4221
SNW_HH_PRECOMPUTE: Finished Age Group:9 of 82, time-this-age:3.4152
SNW_HH_PRECOMPUTE: Finished Age Group:10 of 82, time-this-age:3.4065
SNW_HH_PRECOMPUTE: Finished Age Group:11 of 82, time-this-age:3.1869
SNW_HH_PRECOMPUTE: Finished Age Group:12 of 82, time-this-age:3.3572
SNW_HH_PRECOMPUTE: Finished Age Group:13 of 82, time-this-age:3.4748
SNW_HH_PRECOMPUTE: Finished Age Group:14 of 82, time-this-age:3.3824
SNW_HH_PRECOMPUTE: Finished Age Group:15 of 82, time-this-age:3.4573
SNW_HH_PRECOMPUTE: Finished Age Group:16 of 82, time-this-age:3.3233
SNW_HH_PRECOMPUTE: Finished Age Group:17 of 82, time-this-age:3.3965
SNW_HH_PRECOMPUTE: Finished Age Group:18 of 82, time-this-age:3.3613
SNW_HH_PRECOMPUTE: Finished Age Group:19 of 82, time-this-age:3.3402
SNW_HH_PRECOMPUTE: Finished Age Group:20 of 82, time-this-age:3.479
SNW_HH_PRECOMPUTE: Finished Age Group:21 of 82, time-this-age:3.2899
SNW_HH_PRECOMPUTE: Finished Age Group:22 of 82, time-this-age:3.3222
SNW_HH_PRECOMPUTE: Finished Age Group:23 of 82, time-this-age:3.3025
SNW_HH_PRECOMPUTE: Finished Age Group:24 of 82, time-this-age:3.3682
SNW_HH_PRECOMPUTE: Finished Age Group:25 of 82, time-this-age:3.3555
SNW_HH_PRECOMPUTE: Finished Age Group:26 of 82, time-this-age:3.3701
SNW_HH_PRECOMPUTE: Finished Age Group:27 of 82, time-this-age:3.5453
SNW_HH_PRECOMPUTE: Finished Age Group:28 of 82, time-this-age:3.2218
SNW_HH_PRECOMPUTE: Finished Age Group:29 of 82, time-this-age:3.3947
SNW_HH_PRECOMPUTE: Finished Age Group:30 of 82, time-this-age:3.3024
SNW_HH_PRECOMPUTE: Finished Age Group:31 of 82, time-this-age:3.3699
SNW_HH_PRECOMPUTE: Finished Age Group:32 of 82, time-this-age:3.4931
SNW_HH_PRECOMPUTE: Finished Age Group:33 of 82, time-this-age:3.3584
SNW_HH_PRECOMPUTE: Finished Age Group:34 of 82, time-this-age:3.4127
SNW_HH_PRECOMPUTE: Finished Age Group:35 of 82, time-this-age:3.4113
SNW_HH_PRECOMPUTE: Finished Age Group:36 of 82, time-this-age:3.2095
SNW_HH_PRECOMPUTE: Finished Age Group:37 of 82, time-this-age:3.4244
SNW_HH_PRECOMPUTE: Finished Age Group:38 of 82, time-this-age:3.5012
SNW_HH_PRECOMPUTE: Finished Age Group:39 of 82, time-this-age:3.2675
SNW_HH_PRECOMPUTE: Finished Age Group:40 of 82, time-this-age:3.2625
SNW_HH_PRECOMPUTE: Finished Age Group:41 of 82, time-this-age:3.4011
SNW_HH_PRECOMPUTE: Finished Age Group:42 of 82, time-this-age:3.2533
SNW_HH_PRECOMPUTE: Finished Age Group:43 of 82, time-this-age:3.5132
SNW_HH_PRECOMPUTE: Finished Age Group:44 of 82, time-this-age:3.4771
SNW_HH_PRECOMPUTE: Finished Age Group:45 of 82, time-this-age:3.3133
SNW_HH_PRECOMPUTE: Finished Age Group:46 of 82, time-this-age:3.4673
SNW_HH_PRECOMPUTE: Finished Age Group:47 of 82, time-this-age:3.1794
SNW_HH_PRECOMPUTE: Finished Age Group:48 of 82, time-this-age:3.1958
SNW_HH_PRECOMPUTE: Finished Age Group:49 of 82, time-this-age:3.4545
SNW_HH_PRECOMPUTE: Finished Age Group:50 of 82, time-this-age:3.355
SNW_HH_PRECOMPUTE: Finished Age Group:51 of 82, time-this-age:3.2059
SNW_HH_PRECOMPUTE: Finished Age Group:52 of 82, time-this-age:3.2882
SNW_HH_PRECOMPUTE: Finished Age Group:53 of 82, time-this-age:3.3772
SNW_HH_PRECOMPUTE: Finished Age Group:54 of 82, time-this-age:3.3279
SNW_HH_PRECOMPUTE: Finished Age Group:55 of 82, time-this-age:3.5412
SNW_HH_PRECOMPUTE: Finished Age Group:56 of 82, time-this-age:3.504
SNW_HH_PRECOMPUTE: Finished Age Group:57 of 82, time-this-age:3.4961
SNW_HH_PRECOMPUTE: Finished Age Group:58 of 82, time-this-age:3.3629
SNW_HH_PRECOMPUTE: Finished Age Group:59 of 82, time-this-age:3.4105
SNW_HH_PRECOMPUTE: Finished Age Group:60 of 82, time-this-age:3.3755
SNW_HH_PRECOMPUTE: Finished Age Group:61 of 82, time-this-age:3.4102
SNW_HH_PRECOMPUTE: Finished Age Group:62 of 82, time-this-age:3.4844
SNW_HH_PRECOMPUTE: Finished Age Group:63 of 82, time-this-age:3.3864
SNW_HH_PRECOMPUTE: Finished Age Group:64 of 82, time-this-age:3.6674
SNW_HH_PRECOMPUTE: Finished Age Group:65 of 82, time-this-age:3.3549
SNW_HH_PRECOMPUTE: Finished Age Group:66 of 82, time-this-age:3.3543
```

```
SNW_HH_PRECOMPUTE: Finished Age Group:67 of 82, time-this-age:3.3775
SNW_HH_PRECOMPUTE: Finished Age Group:68 of 82, time-this-age:3.3574
SNW_HH_PRECOMPUTE: Finished Age Group:69 of 82, time-this-age:3.3706
SNW_HH_PRECOMPUTE: Finished Age Group:70 of 82, time-this-age:3.4841
SNW_HH_PRECOMPUTE: Finished Age Group:71 of 82, time-this-age:3.361
SNW_HH_PRECOMPUTE: Finished Age Group:72 of 82, time-this-age:3.3692
SNW_HH_PRECOMPUTE: Finished Age Group:73 of 82, time-this-age:3.4296
SNW_HH_PRECOMPUTE: Finished Age Group:74 of 82, time-this-age:3.4363
SNW_HH_PRECOMPUTE: Finished Age Group:75 of 82, time-this-age:3.391
SNW_HH_PRECOMPUTE: Finished Age Group:76 of 82, time-this-age:3.4927
SNW_HH_PRECOMPUTE: Finished Age Group:77 of 82, time-this-age:3.4073
SNW_HH_PRECOMPUTE: Finished Age Group:78 of 82, time-this-age:3.3416
SNW_HH_PRECOMPUTE: Finished Age Group:79 of 82, time-this-age:3.4042
SNW_HH_PRECOMPUTE: Finished Age Group:80 of 82, time-this-age:3.4844
SNW_HH_PRECOMPUTE: Finished Age Group:81 of 82, time-this-age:3.4572
SNW_HH_PRECOMPUTE: Finished Age Group:82 of 82, time-this-age:3.2411
SNW_HH_PRECOMPUTE: Finished Age Group:83 of 82, time-this-age:3.3428
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time cost=284.95
```

Solve for 2008 Value and Consumption with 0 and 2 Checks

Now we use the **snw_v08_jaeemk** function, which takes as inputs the 2008-specific value and policy we have already found, to compute the 2008 value and consumption based on different stimulus amounts via asset-equivalent transformation given stimulus amounts.

FIrst, obtain V and C with zero stimulus.

```
% Call Function
welf_checks = 0;
[V_2008_check0, C_2008_check0] = snw_v08_jaeemk(...
    welf_checks, mp_params, mp_controls, V_2008, cons_2008, mp_precompute_res);
```

```
Solve for V_2008_check for 0 stimulus checks
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=st_biden_or_trump_undefined;welf_checks=0;TR=0.0018238;SNW_MP_PARAM=d
Completed SNW_V08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=3e-05
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvar
C_2008_check	1	1	6	4.37e+07	83	5.265e+05	2.3277e+08	5.3267	8.4419	1.5848
V_2008_check	2	2	6	4.37e+07	83	5.265e+05	-6.6426e+08	-15.201	21.85	-1.4375

```
xxx TABLE:C_2008_check xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c526499	c
r1	0.036218	0.036736	0.038184	0.042735	0.048545	12.256	12.541	12.835	13.136	1
r2	0.036271	0.036736	0.038385	0.043404	0.049852	12.491	12.778	13.072	13.374	1
r3	0.036717	0.037251	0.039845	0.044907	0.051515	12.744	13.032	13.327	13.628	1
r4	0.038144	0.038678	0.041269	0.046371	0.053128	12.989	13.277	13.573	13.872	1
r5	0.039534	0.040068	0.042653	0.047793	0.054687	13.224	13.513	13.809	14.105	1
r79	0.2016	0.20214	0.20586	0.21598	0.23568	35.82	37.367	39.414	41.705	4
r80	0.2016	0.20214	0.20586	0.21598	0.23568	40.755	42.955	45.289	47.95	5
r81	0.2016	0.20214	0.20586	0.21598	0.23568	48.912	52.041	55.022	57.919	6
r82	0.2016	0.20214	0.20586	0.21598	0.23568	66.719	69.201	72.373	77.005	8
r83	0.2016	0.20214	0.20586	0.21598	0.23568	116.83	122.65	128.67	134.89	

```
xxx TABLE:V_2008_check xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c526499

r1	-295.66	-295.26	-292.66	-286.62	-277.22	-4.3615	-4.2673	-4.1741	-4.0822
r2	-286.11	-285.71	-283.12	-277.16	-268.03	-4.2548	-4.1641	-4.0744	-3.9858
r3	-276.49	-276.09	-273.59	-267.84	-259.11	-4.1461	-4.0589	-3.9727	-3.8874
r4	-266.77	-266.41	-264.08	-258.7	-250.49	-4.0342	-3.9507	-3.868	-3.7862
r5	-257.99	-257.65	-255.48	-250.43	-242.69	-3.9287	-3.8485	-3.769	-3.6903
r79	-13.356	-13.343	-13.253	-13.025	-12.638	-0.22088	-0.21055	-0.20083	-0.1917
r80	-12.025	-12.012	-11.923	-11.695	-11.308	-0.16977	-0.1618	-0.15428	-0.14721
r81	-10.382	-10.369	-10.28	-10.052	-9.6651	-0.11711	-0.11162	-0.10645	-0.10156
r82	-8.1742	-8.1611	-8.0716	-7.844	-7.457	-0.065329	-0.062239	-0.059357	-0.056632
r83	-4.9602	-4.9471	-4.8576	-4.6301	-4.2431	-0.020966	-0.019971	-0.019037	-0.01816

```
% Call Function
```

Second, obtain V and C with two stimulus checks.

```
welf_checks = 2;
[V_2008_check2, C_2008_check2] = snw_v08_jaeemk(...
    welf_checks, mp_params, mp_controls, V_2008, cons_2008, mp_precompute_res);
```

```
Solve for V_2008_check for 2 stimulus checks
Completed SNW_A4CHK_WRK_BISEC_VEC;SNW_MP_PARAM=st_biden_or_trump_undefined;welf_checks=2;TR=0.0018238;SNW_MP_PARAM=c
Completed SNW_V08_JAEEMK;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;timeEUEC=2.16e-05
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvar
C_2008_check	1	1	6	4.37e+07	83	5.265e+05	2.328e+08	5.3273	8.442	1.5847
V_2008_check	2	2	6	4.37e+07	83	5.265e+05	-6.6365e+08	-15.187	21.798	-1.4353

xxx TABLE:C_2008_check xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c526499	c
r1	0.037941	0.038148	0.039819	0.043807	0.049244	12.256	12.541	12.835	13.136	1
r2	0.038108	0.038344	0.040188	0.044594	0.050571	12.492	12.778	13.073	13.374	1
r3	0.03941	0.039781	0.041664	0.046126	0.052244	12.745	13.032	13.327	13.628	1
r4	0.040834	0.041205	0.043102	0.047618	0.053867	12.989	13.278	13.573	13.872	1
r5	0.04222	0.042589	0.0445	0.049065	0.055435	13.224	13.513	13.809	14.105	1
r79	0.20525	0.20579	0.20951	0.21963	0.23776	35.821	37.368	39.415	41.707	
r80	0.20525	0.20579	0.20951	0.21963	0.23776	40.756	42.957	45.29	47.952	5
r81	0.20525	0.20579	0.20951	0.21963	0.2378	48.914	52.043	55.024	57.92	6
r82	0.20525	0.20579	0.20951	0.21963	0.23814	66.72	69.203	72.375	77.007	8
r83	0.20525	0.20579	0.20951	0.21963	0.23933	116.84	122.66	128.68	134.89	1

xxx TABLE:V_2008_check xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c526496	c526497	c526498	c526499
r1	-293.09	-292.72	-290.49	-284.88	-275.86	-4.3615	-4.2672	-4.1741	-4.0821
r2	-283.55	-283.18	-280.98	-275.47	-266.73	-4.2548	-4.164	-4.0743	-3.9857
r3	-274.01	-273.65	-271.52	-266.23	-257.88	-4.146	-4.0589	-3.9726	-3.8874
r4	-264.47	-264.13	-262.14	-257.19	-249.33	-4.0341	-3.9506	-3.8679	-3.7861
r5	-255.84	-255.53	-253.66	-249	-241.59	-3.9286	-3.8484	-3.769	-3.6903
r79	-13.268	-13.255	-13.171	-12.954	-12.578	-0.22088	-0.21054	-0.20082	-0.1917
r80	-11.937	-11.924	-11.841	-11.623	-11.248	-0.16976	-0.16179	-0.15428	-0.1472
r81	-10.294	-10.281	-10.198	-9.9804	-9.6057	-0.11711	-0.11162	-0.10644	-0.10156
r82	-8.0862	-8.0735	-7.9895	-7.7724	-7.3981	-0.065327	-0.062237	-0.059355	-0.056631
r83	-4.8723	-4.8595	-4.7755	-4.5584	-4.1854	-0.020965	-0.01997	-0.019036	-0.018159

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check = V_2008_check2 - V_2008_check0;
mn_MPC_U_gain_share_check = (C_2008_check2 - C_2008_check0)./(welf_checks*mp_params('TR'));
```

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=0.0018238   xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5     mean_eta_6     mean_
```

1	0	1.0061	0.90365	0.8116	0.72825	0.65329	0.58628	0
2	0.00051498	0.99612	0.89566	0.80504	0.72289	0.64888	0.58264	0.
3	0.0041199	0.84723	0.77113	0.6992	0.63256	0.57176	0.51677	0.
4	0.013905	0.64775	0.59855	0.55009	0.50372	0.46037	0.42041	0.
5	0.032959	0.46258	0.43381	0.40406	0.37453	0.34616	0.31948	0.
6	0.064373	0.33044	0.31319	0.29451	0.27537	0.25656	0.23856	0.

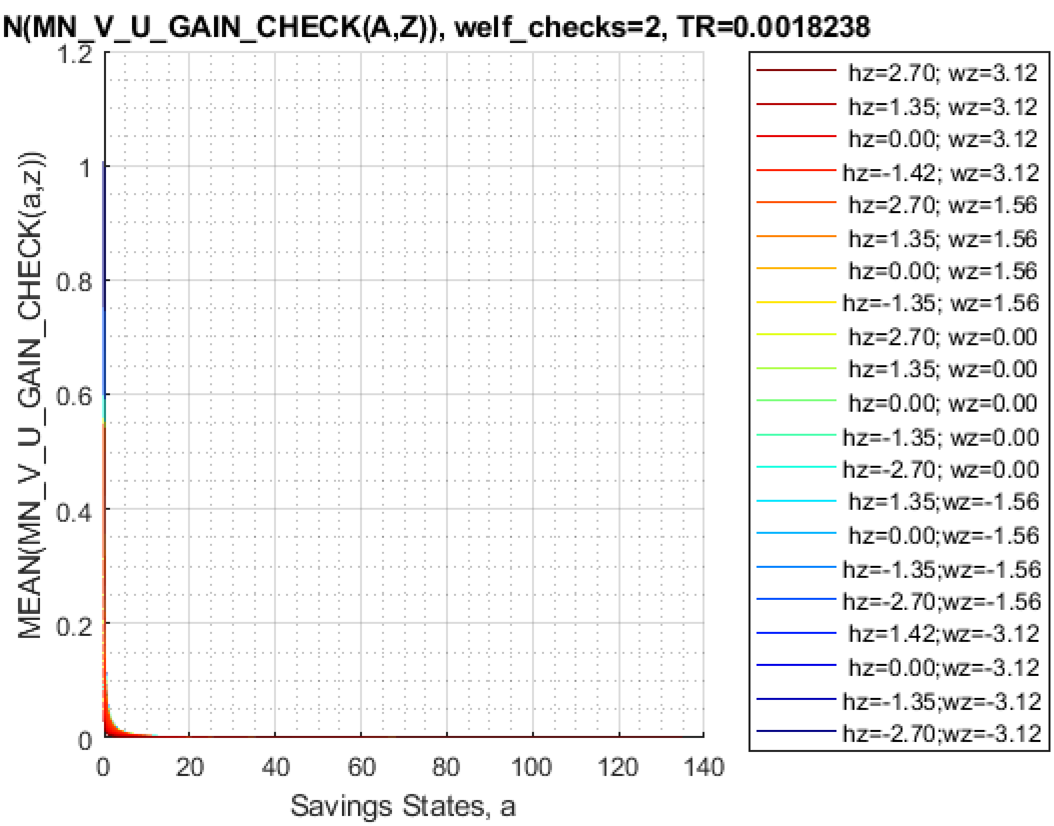
```
% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 4, 1, cl_mp_dat
```

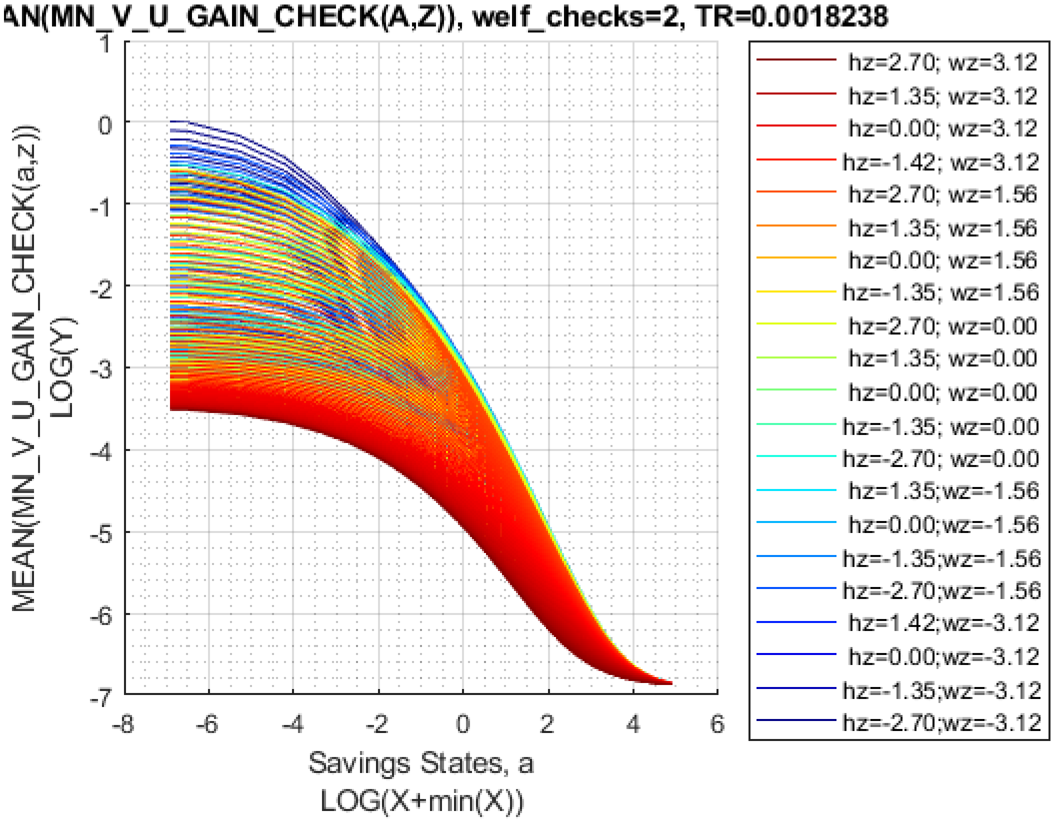
xxx MEAN(MN_MPC_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0018238 xxxxxxxxxxxxxxxxxxxxxxxxxx

group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5	mean_eta_6	mean_
1	0	0.96418	0.92776	0.90436	0.88966	0.87994	0.87411	0.86
2	0.00051498	0.95984	0.92012	0.8967	0.8773	0.86664	0.86053	0.8
3	0.0041199	0.87755	0.8775	0.87564	0.87396	0.87313	0.87278	0.87
4	0.013905	0.79847	0.79688	0.79546	0.79569	0.79653	0.79773	0.79
5	0.032959	0.71392	0.71064	0.71031	0.71109	0.71248	0.71436	0.71
6	0.064373	0.63466	0.6344	0.63533	0.63678	0.63877	0.64105	0.64

Graph Mean Values:

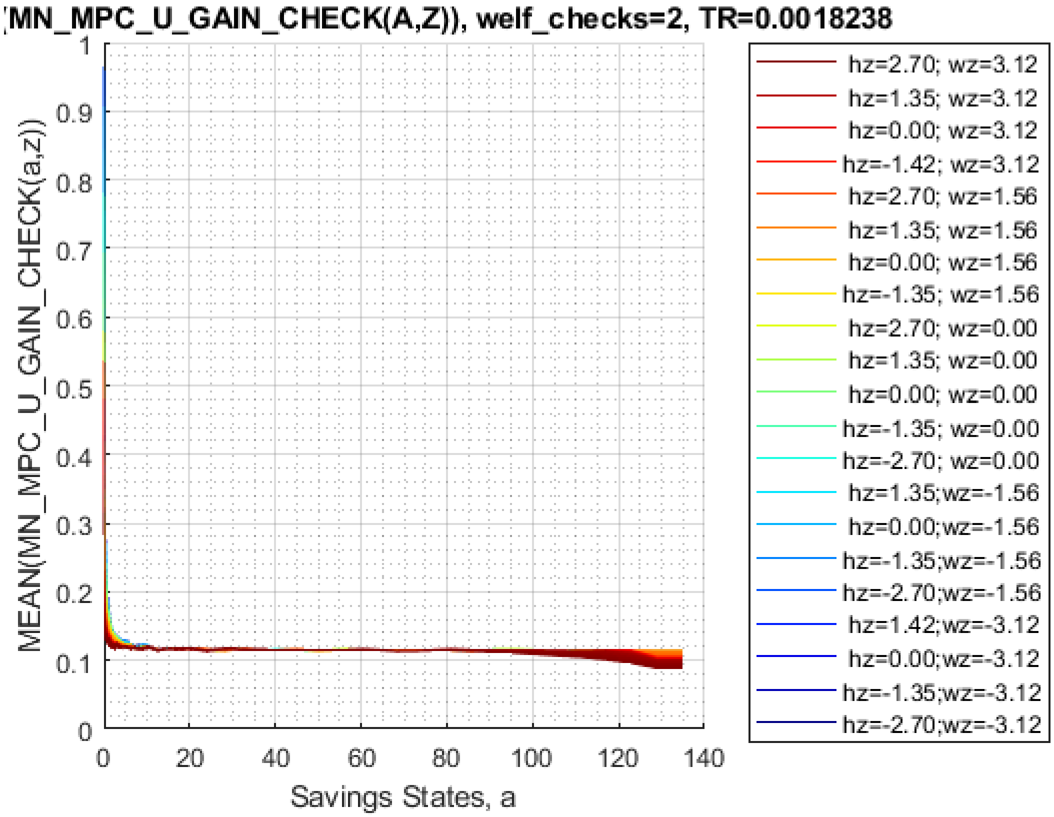
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);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

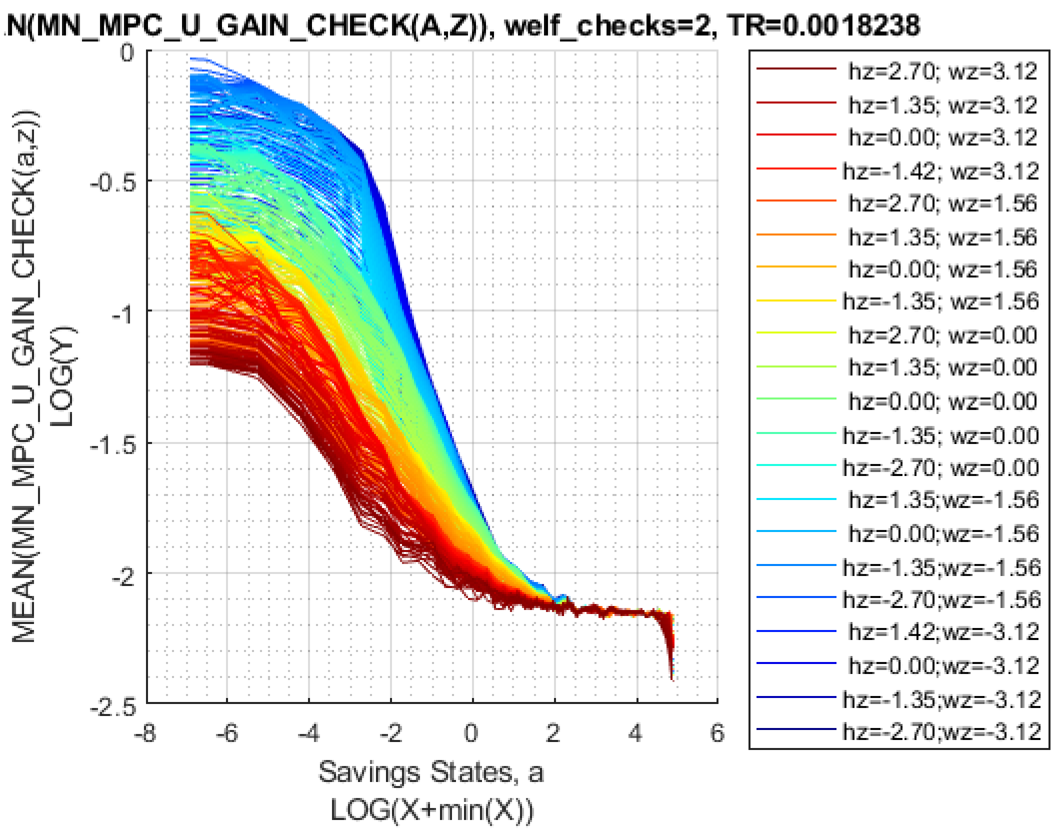




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
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 Marginal Value and MPC over Y(a,eta), Conditional On Kids, Marry, Age, Education

Income is generated by savings and shocks, what are the income levels generated by all the shock and savings points conditional on kids, marital status, age and educational levels. Plot on the Y axis MPC, and plot on the X axis income levels, use colors to first distinguish between different a levels, then use colors to distinguish between different eta levles.

Set Up date, Select Age, unmarried, no kids, lower education:

```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
% 38 year old, unmarried, no kids, lower educated
% Only Household Head Shock Matters so select up to 'n_eta_H_grid'
mn_total_inc_jemk = total_inc_VFI(19,:,1:mp_params('n_eta_H_grid'),1,1,1);
mn_V_W_gain_check_use = V_2008_check2 - V_2008_check0;
mn_C_W_gain_check_use = C_2008_check2 - C_2008_check0;
```

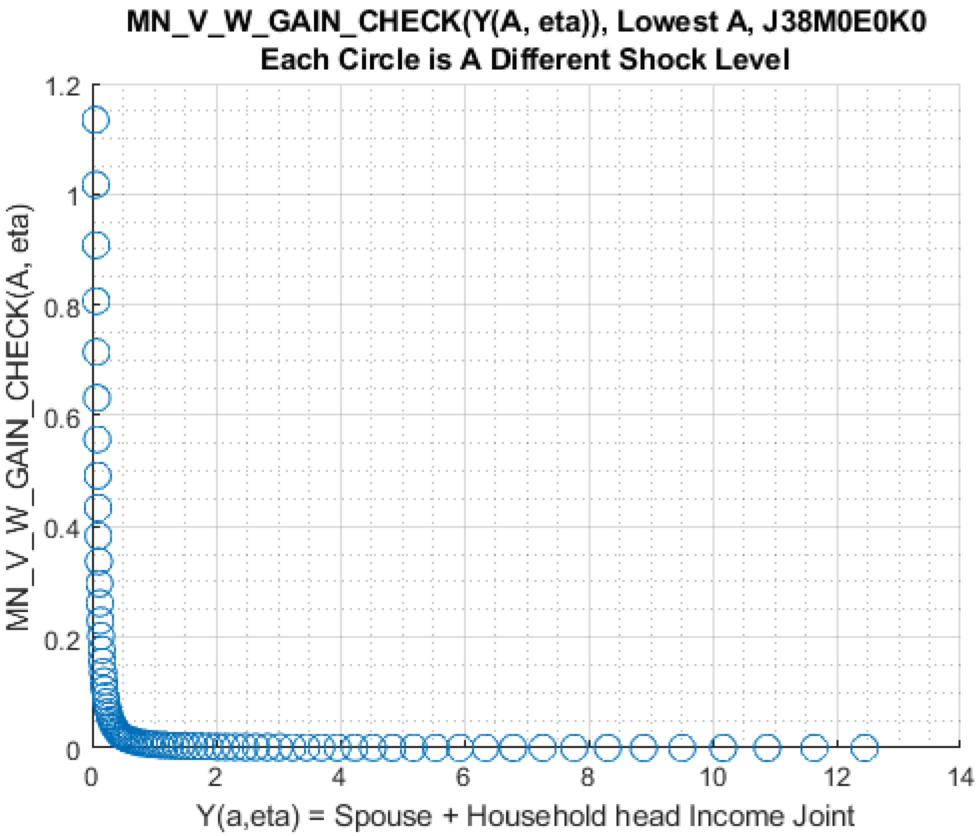
Select Age, Education, Marital, Kids Count:s

```
% Selections
it_age = 21; % +18
it_marital = 1; % 1 = unmarried
it_kids = 1; % 1 = kids is zero
it_educ = 1; % 1 = lower education
% Select: NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
mn_C_W_gain_check_jemk = mn_C_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ,
mn_V_W_gain_check_jemk = mn_V_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ,
% Reshape, so shock is the first dim, a is the second
mt_total_inc_jemk = permute(mn_total_inc_jemk,[3,2,1]);
mt_C_W_gain_check_jemk = permute(mn_C_W_gain_check_jemk,[3,2,1]);
mt_C_W_gain_check_jemk(mt_C_W_gain_check_jemk<=1e-10) = 1e-10;
mt_V_W_gain_check_jemk = permute(mn_V_W_gain_check_jemk,[3,2,1]);
mt_V_W_gain_check_jemk(mt_V_W_gain_check_jemk<=1e-10) = 1e-10;
% Generate meshed a and shock grid
[mt_eta_H, mt_a] = ndgrid(eta_H_grid(1:mp_params('n_eta_H_grid')), agrid);
```

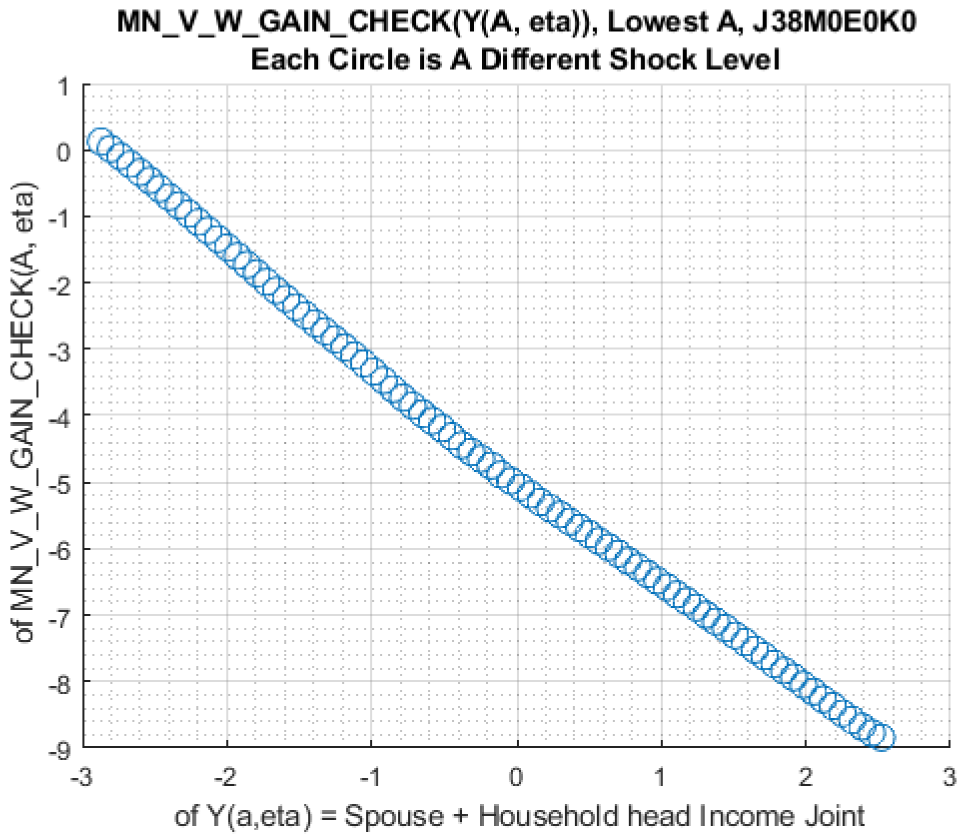
Marginal Value Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
it_a = 1;
scatter((mt_total_inc_jemk(:,it_a)), (mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

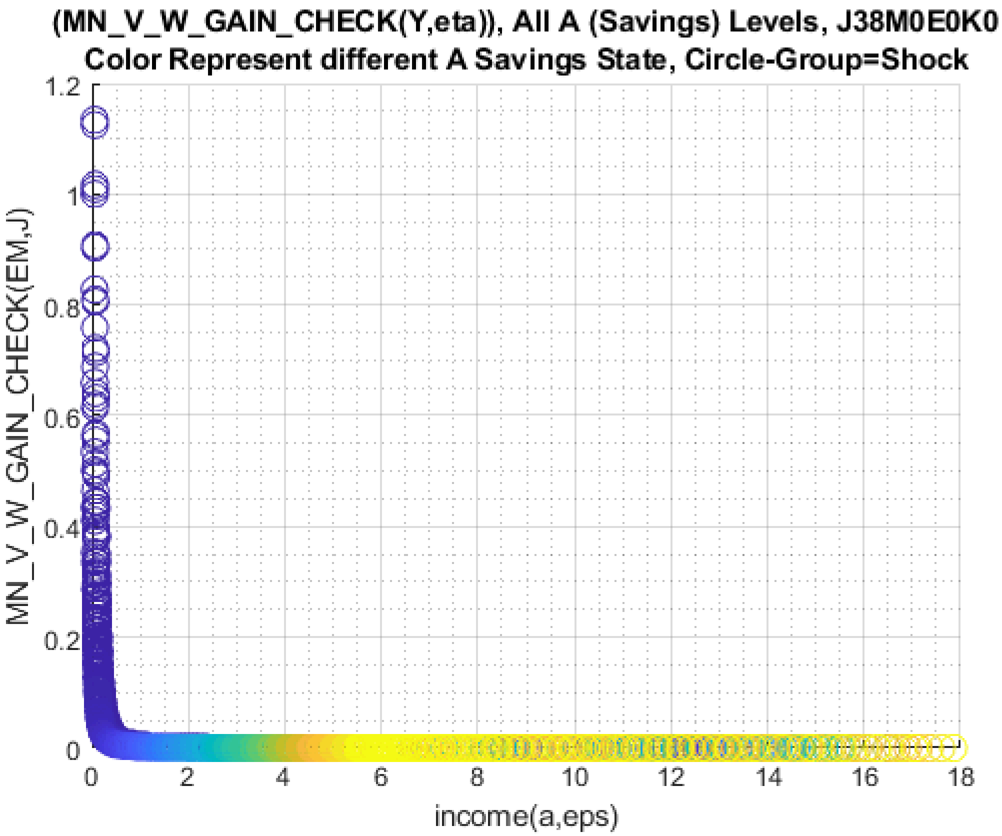



```
figure();
it_shock = 1;
scatter(log(mt_total_inc_jemk(:,it_a)), log(mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel(' of Y(a,eta) = Spouse + Household head Income Joint');
ylabel(' of MN\_V\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

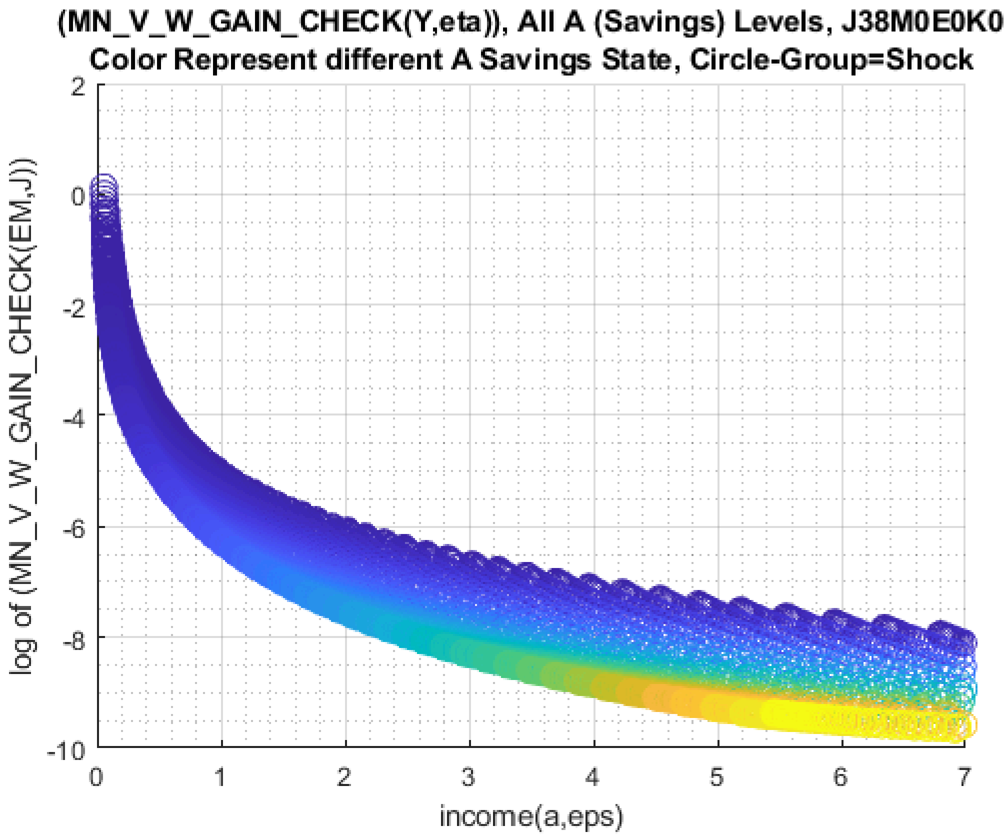


Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_V\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```



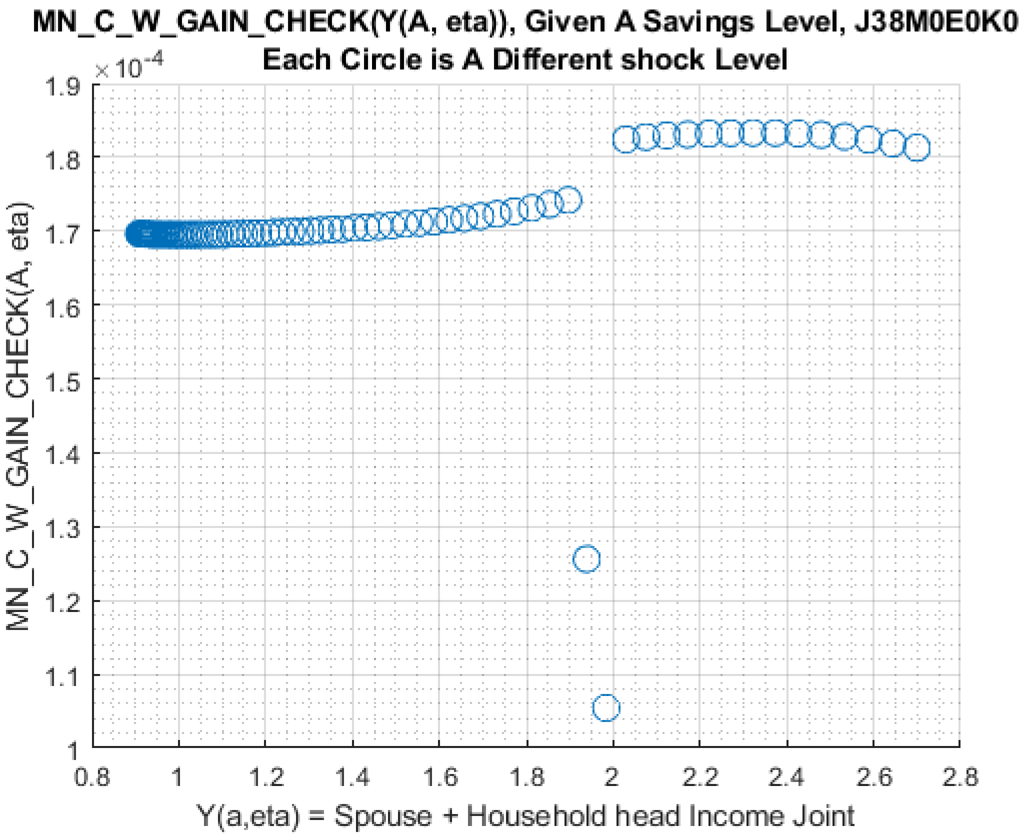

```
figure();
scatter((mt_total_inc_jemk(:)), log(mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
xlim([0,7]);
grid on;
grid minor;
```



Marginal Consumption Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

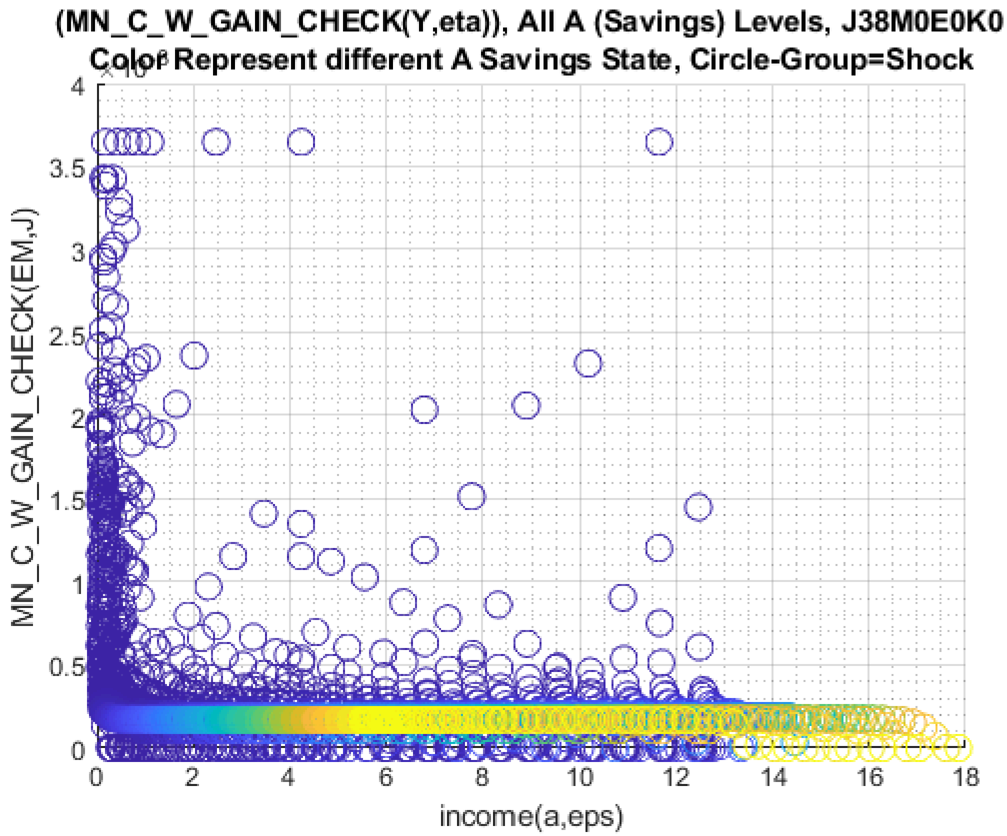
How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
figure();
it_a = 50;
scatter(log(mt_total_inc_jemk(:,it_a)), mt_C_W_gain_check_jemk(:,it_a), 100);
title({'MN\_C\_W\_GAIN\_CHECK(Y(A, eta)), Given A Savings Level, J38M0E0K0', ...
    'Each Circle is A Different shock Level'});
xlabel('Y(a,eta) = Spouse + Household head Income Joint');
ylabel('MN\_C\_W\_GAIN\_CHECK(A, eta)');
grid on;
grid minor;
```

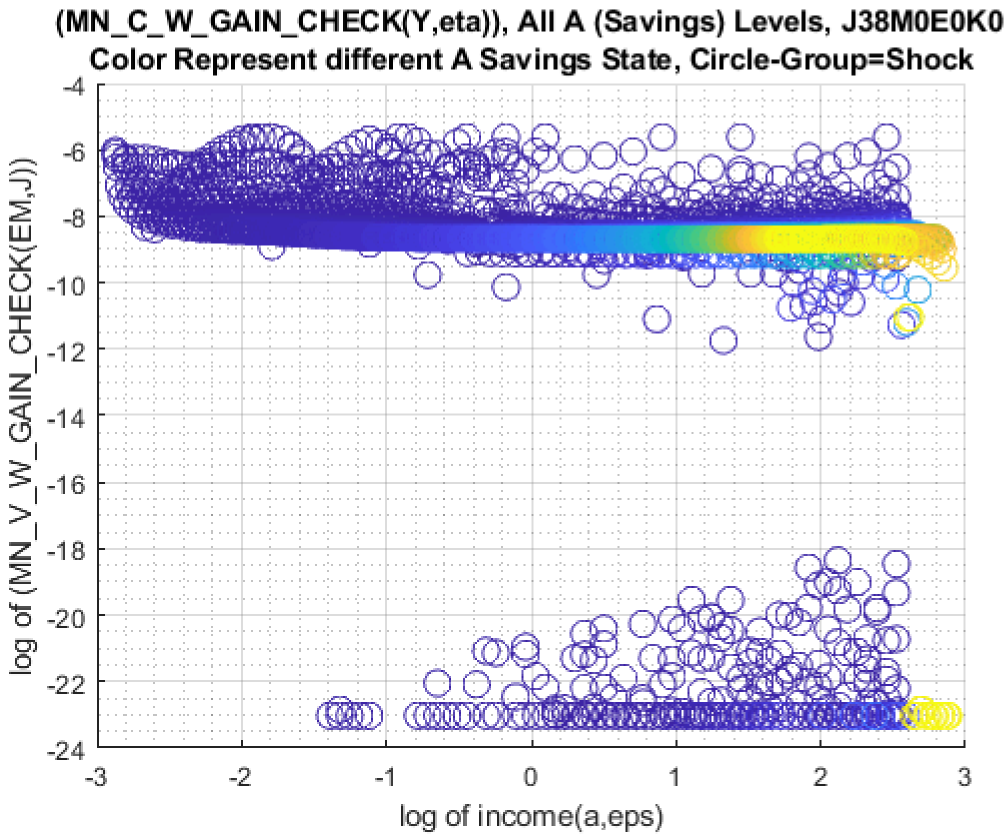


Plot all asset levels:

```
figure();
scatter((mt_total_inc_jemk(:)), (mt_C_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('MN\_C\_W\_GAIN\_CHECK(EM,J)');
grid on;
grid minor;
```




```
figure();
scatter(log(mt_total_inc_jemk(:)), log(mt_C_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_C\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('log of income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');
grid on;
grid minor;
```



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=0.0018238 xxxxxxxxxxxxxxxxxxxxxxxxxx

group	kids	marry	mean_age_18	mean_age_19	mean_age_20	mean_age_21	mean_age_22	mean_age_23
1	1	0	0.03116	0.030004	0.028268	0.025846	0.023825	0.022124
2	2	0	0.042925	0.041372	0.038951	0.035539	0.032682	0.03027
3	3	0	0.050042	0.048477	0.045918	0.041935	0.038603	0.035792
4	4	0	0.056814	0.055153	0.05234	0.047822	0.044043	0.040856
5	5	0	0.06224	0.060592	0.057683	0.052755	0.048636	0.045166
6	1	1	0.0089468	0.0085141	0.0080936	0.0073219	0.006674	0.0061254
7	2	1	0.012008	0.011426	0.01086	0.0098202	0.0089462	0.0082092
8	3	1	0.014485	0.01381	0.013144	0.011882	0.010826	0.0099375
9	4	1	0.017392	0.016611	0.015824	0.014325	0.013061	0.011988
10	5	1	0.021156	0.020287	0.019385	0.017573	0.016058	0.014774

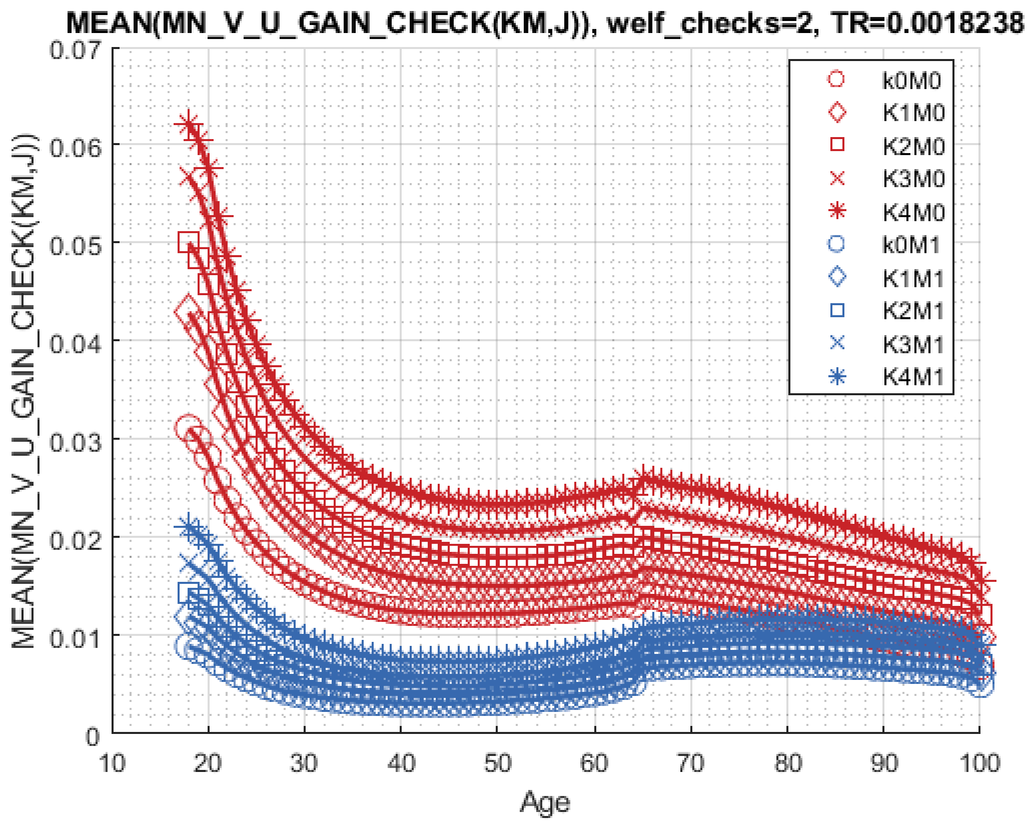
```
% Consumption Function
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2st
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_dat
```

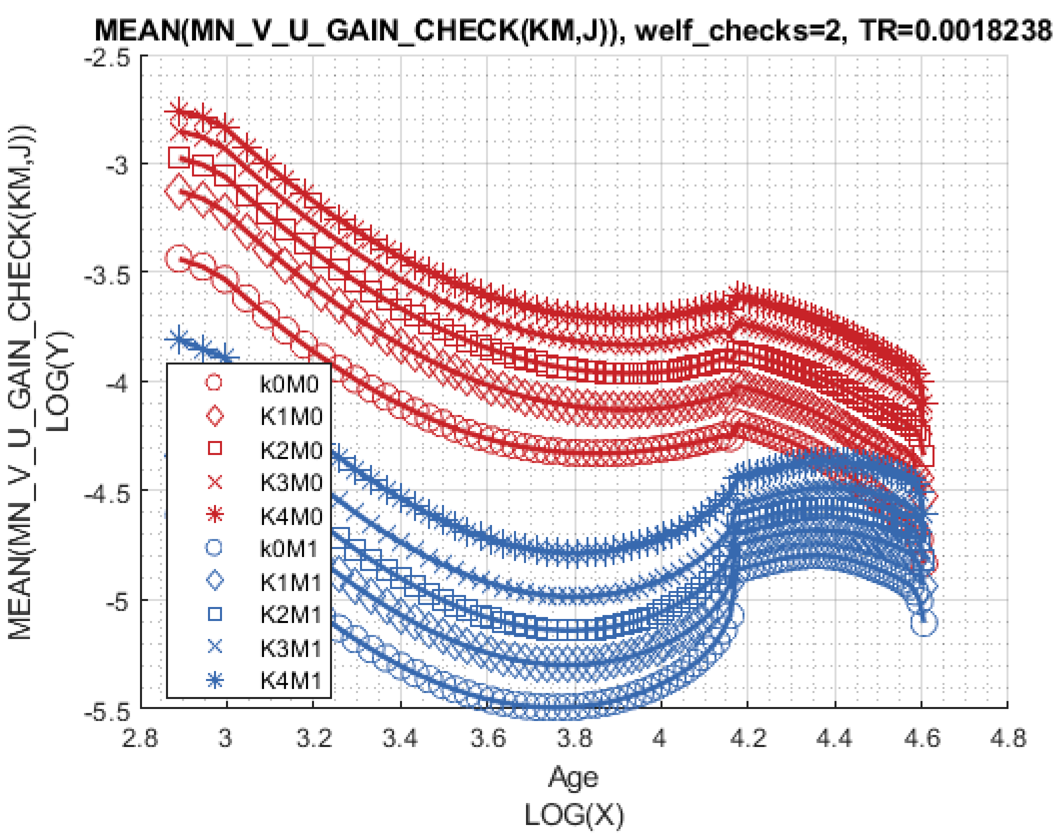
xxx MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0018238 xxxxxxxxxxxxxxxxxxxxxxxxxx

group	kids	marry	mean_age_18	mean_age_19	mean_age_20	mean_age_21	mean_age_22	mean_age_23
1	1	0	0.071209	0.076265	0.082919	0.080484	0.079666	0.078062
2	2	0	0.08025	0.085444	0.092048	0.090298	0.089234	0.088128
3	3	0	0.087972	0.095508	0.10343	0.10119	0.099548	0.097144
4	4	0	0.092255	0.099859	0.10923	0.10651	0.10388	0.10207
5	5	0	0.09665	0.1041	0.11456	0.11152	0.10814	0.10532
6	1	1	0.101	0.10439	0.10978	0.10855	0.10822	0.10834
7	2	1	0.10297	0.10717	0.11242	0.11166	0.10915	0.10816
8	3	1	0.10827	0.11355	0.11922	0.11713	0.11645	0.11572
9	4	1	0.10932	0.11394	0.12031	0.11884	0.11867	0.11712
10	5	1	0.11555	0.12093	0.1289	0.12462	0.12328	0.12196

Graph Mean Values:

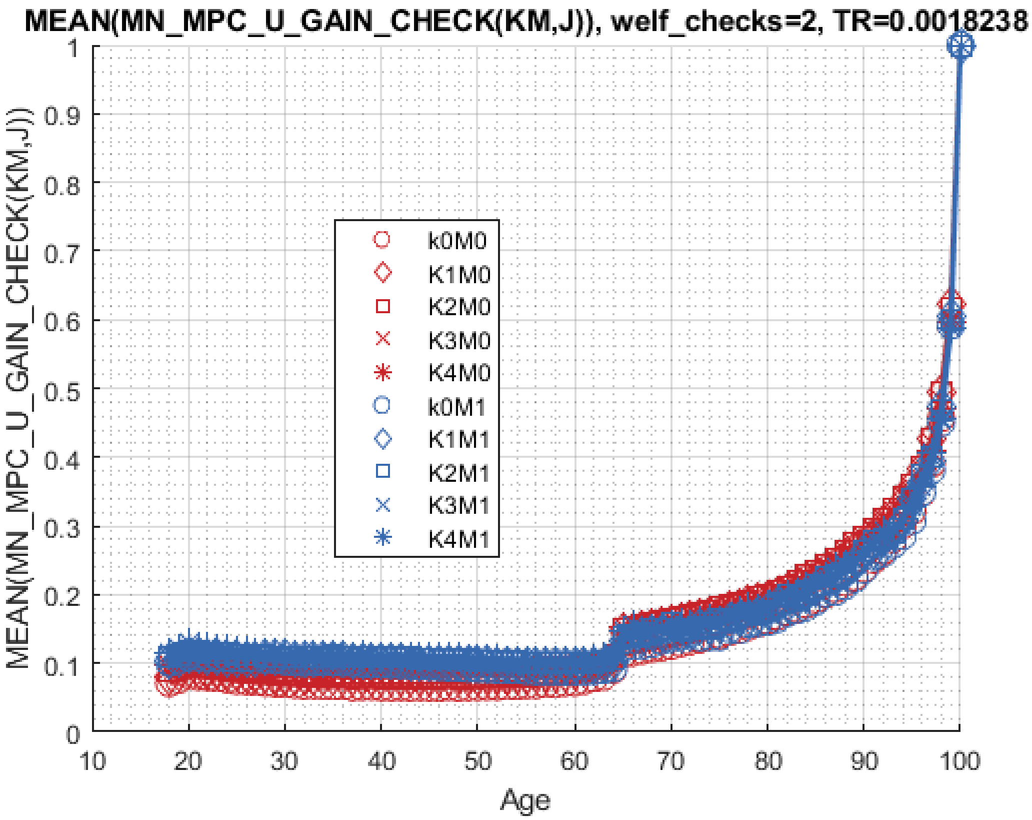
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

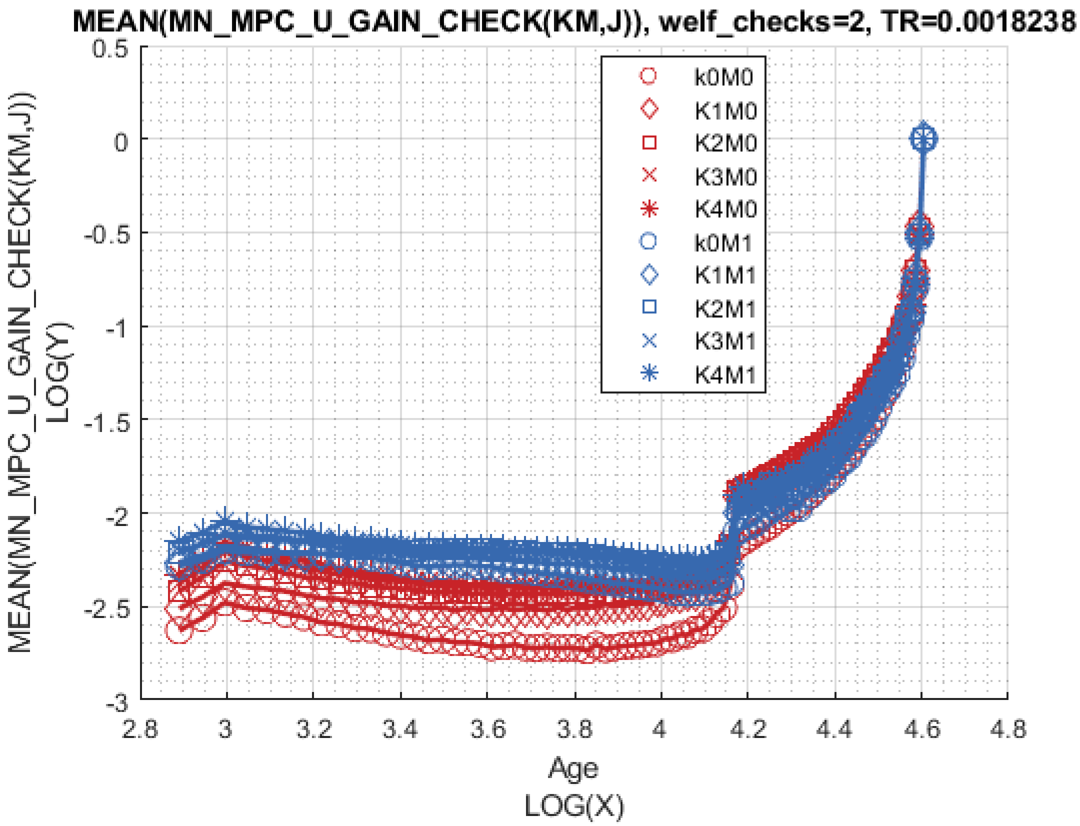




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' n
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=0.0018238  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	0.050249	0.049057	0.04709	0.044412	0.042032	0.039911
2	1	0	0.047023	0.045182	0.042174	0.037147	0.033084	0.029772
3	0	1	0.015742	0.015072	0.014412	0.01336	0.012437	0.011628
4	1	1	0.013853	0.013188	0.012511	0.011009	0.0097884	0.0087852

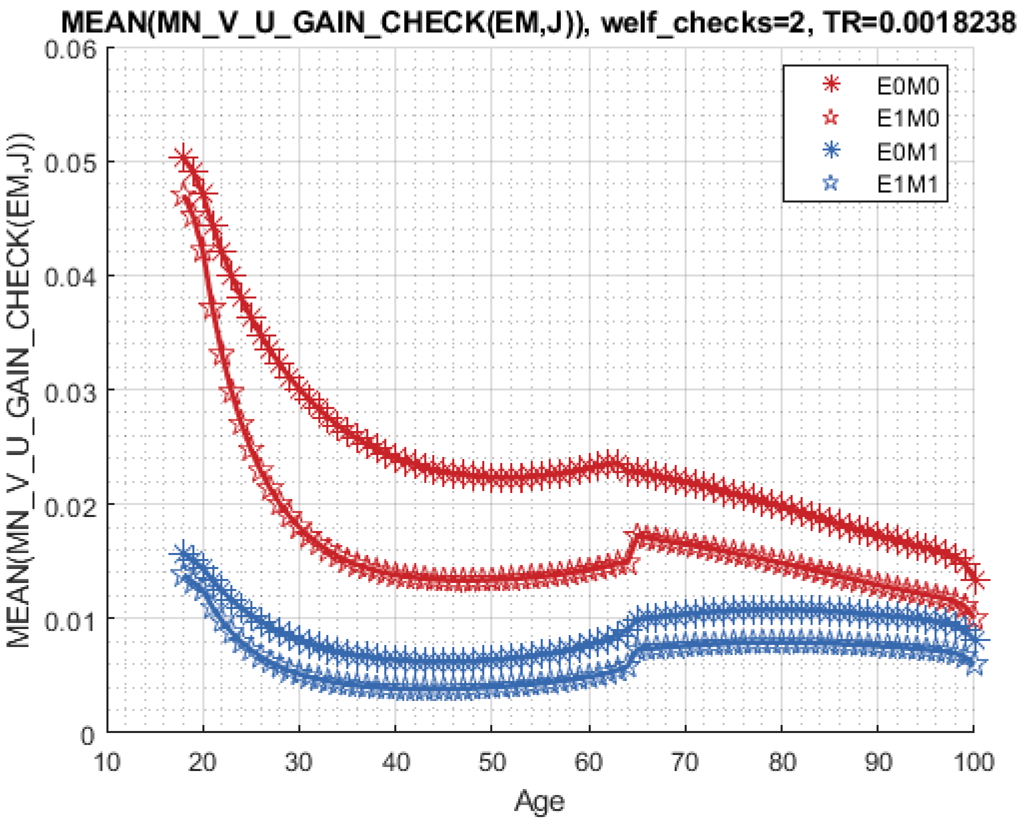
```
% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2st
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_dat
```

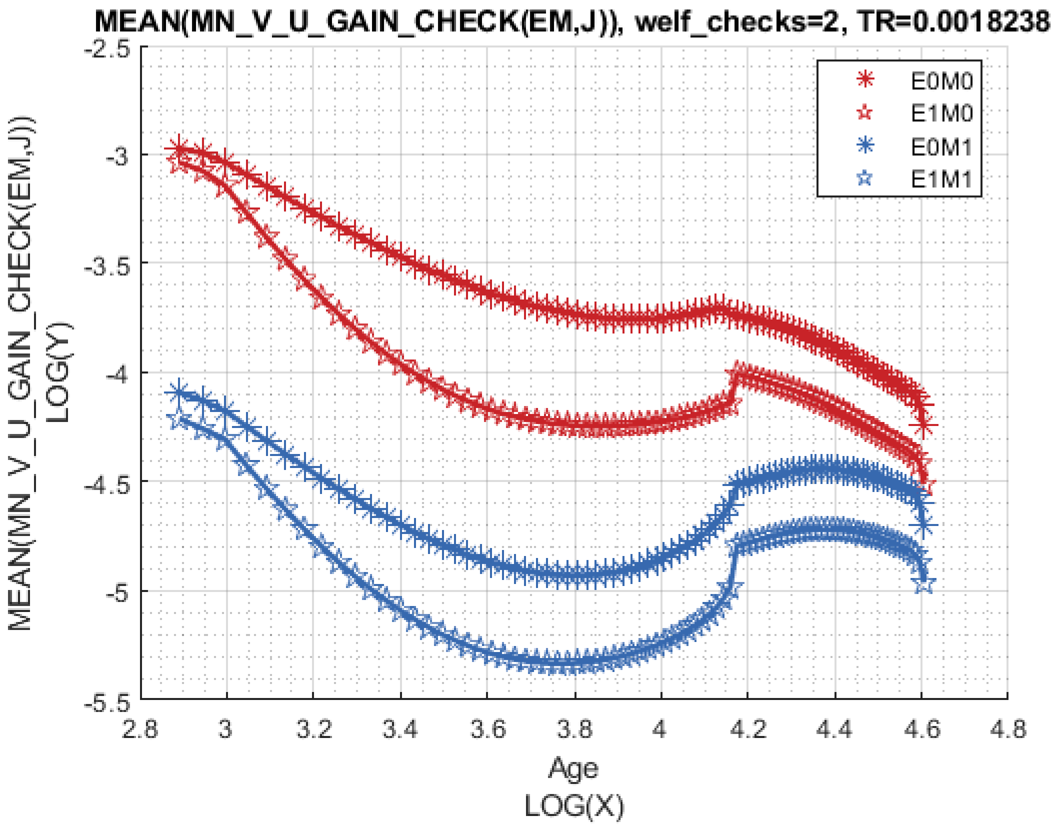
xxx MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0018238 xxxxxxxxxxxxxxxxxxxxxxxxxx

group	edu	marry	mean_age_18	mean_age_19	mean_age_20	mean_age_21	mean_age_22	mean_age_23
1	0	0	0.07766	0.081478	0.084248	0.083711	0.083393	0.082988
2	1	0	0.093675	0.10299	0.11663	0.11229	0.10879	0.1053
3	0	1	0.099571	0.10294	0.10634	0.10623	0.10602	0.10588
4	1	1	0.11528	0.12105	0.12991	0.12609	0.12429	0.12264

Graph Mean Values:

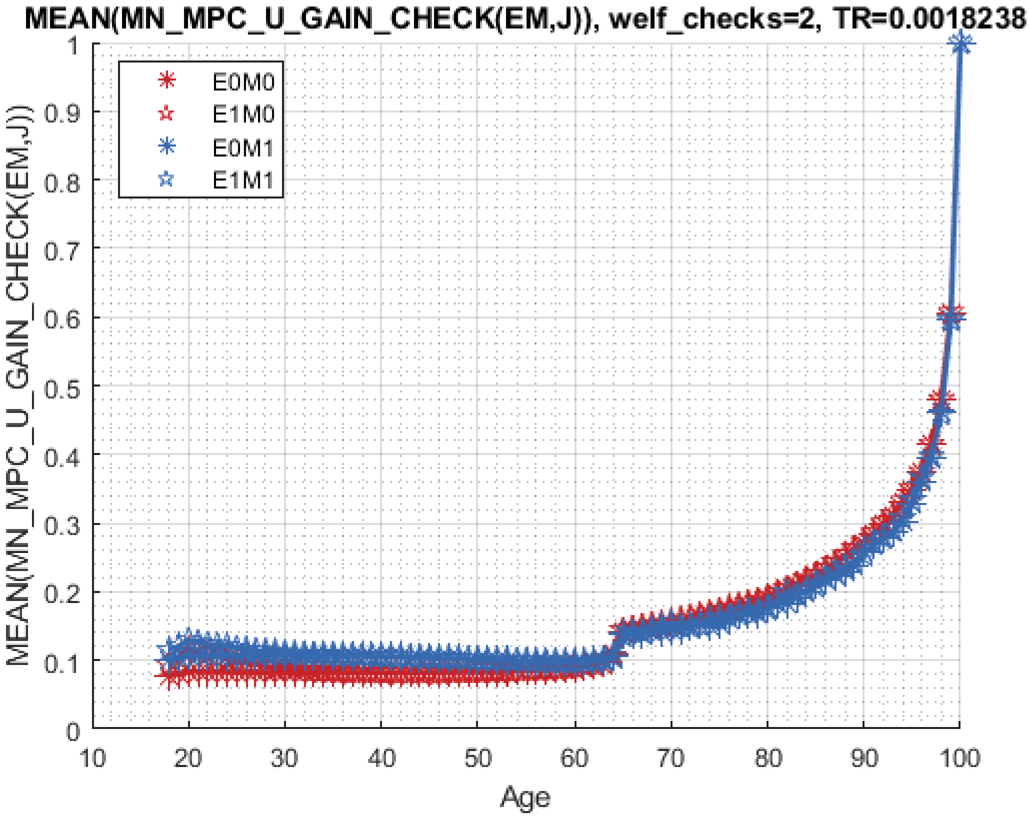
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

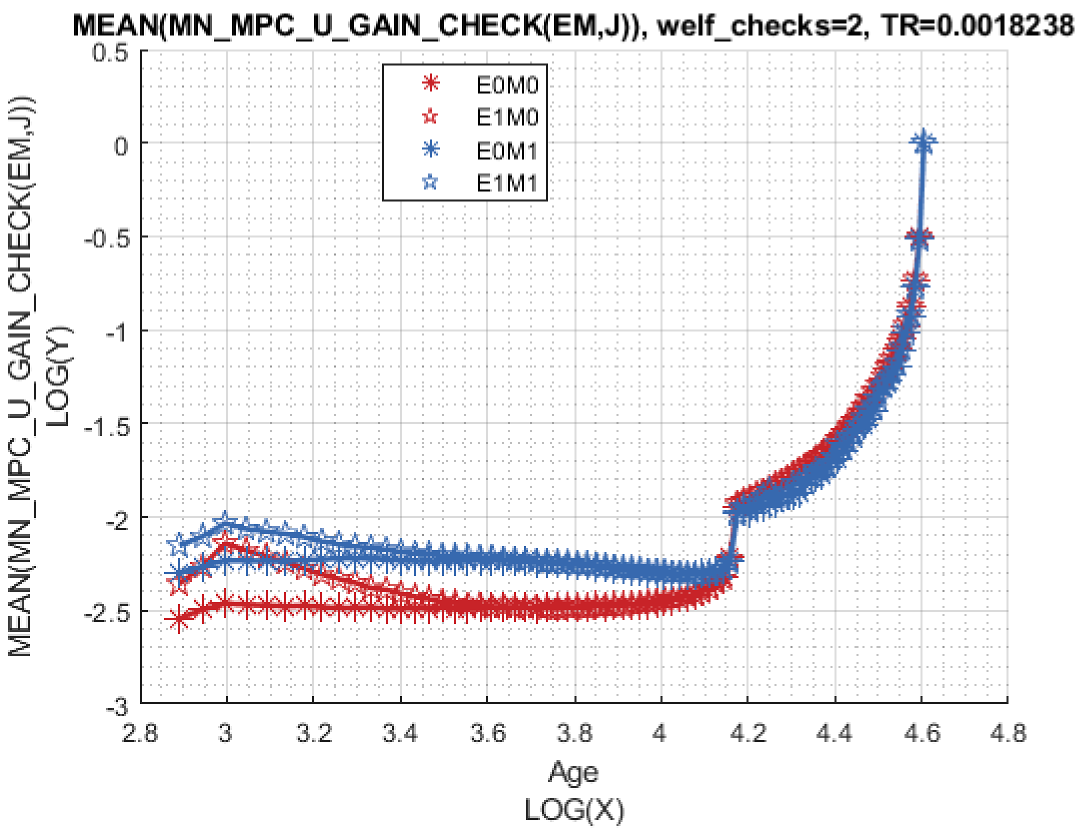




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' n
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=0.0018238
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=0.0018238
E0M0
E1M0
E0M1
E1M1
MEAN(MN_MPC_U_GAIN_CHECK(EM,J))
LOG(Y)
Age
LOG(X)