SNW Dynamic Life Cycle and Welfare Checks Code Companion

Fan Wang

2020-08-10

Contents

Preface

This is a work-in-progress Matlab package consisting of functions that solve the dynamic life cycle model in Nygaard, Sorensen and Wang (2020) (Vegard M. Nygaard, 2020). The code companion presents solutions to the dynamic life-cycle problem, and methods for evaluating the marginal gains from allocating additional welfare checks. Tested with Matlab 2019a (The MathWorks Inc, 2019).

All functions are parts of a matlab toolbox that can be installed:

> Download and install the Matlab toolbox: PrjOptiSNW.mltbx

This bookdown file is a collection of mlx based vignettes for functions that are available from PrjOptiSNW. Each Vignette file contains various examples for invoking each function. The goal of this repository is to make it easier to find/re-use codes produced for various projects.

The package relies on MEconTools, which needs to be installed first. The package does not include allocation functions, only simulation code to generate the value of each welfare check increments for households. Allocation functions rely the R optimal allocation package PrjOptiAlloc.

The site is built using Bookdown (Xie, 2020).

Please contact FanWangEcon for issues or problems.

Chapter 1

Parameters

1.1 Model Parameters

This is the example vignette for function: **snw_mp_param** from the **PrjOptiSNW Package.** This function sets and gets different parameters

1.1.1 Documentation Run Parameters Docdense

Parameters used for documentation vig.

```
mp_params = snw_mp_param('default_docdense', true, 100, 6);
```

```
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_preftechpricegov Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                            i     idx     value
                            --    ---    -------

    Bequests                 1     1           0
    a2                       2     2      1.5286
    bequests_option          3     3           1
    beta                     4     4     0.97116
    cons_allocation_rule     5     5           2
    g_cons                   6     6     0.17576
    g_n                      7     7        0.01
    gamma                    8     8           2
    jret                     9     9          48
    r                       10    10        0.04
    theta                   11    11     0.56523
    throw_in_ocean          12    12           1

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_intlen Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                     i    idx    value
                     -    ---    -----

    n_agrid          1     1      65
    n_educgrid       2     2       2
    n_eta_H_grid     3     3      61
    n_eta_S_grid     4     4       5
```

```
    n_etagrid        5     5      305
    n_jgrid          6     6       83
    n_kidsgrid       7     7        5
    n_marriedgrid    8     8        2

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim    numel    rowN    colN     sum        mean        std       coefv
                -    ---    ----    -----    ----    ----    ------    --------    --------    -----

    inc_grid    1     3      2       201      201      1      578.5      2.8781      1.8836    0.654
    pi_unemp    2     6      2       415       83      5     9.5319    0.022968    0.024679     1.07

xxx TABLE:inc_grid xxxxxxxxxxxxxxxxxx
               c1
            --------

    r1             0
    r2      0.026667
    r3      0.053333
    r4          0.08
    r5       0.10667
    r6       0.13333
    r7          0.16
    r8       0.18667
    r9       0.21333
    r10         0.24
    r11      0.26667
    r12      0.29333
    r13         0.32
    r14      0.34667
    r15      0.37333
    r16          0.4
    r17      0.42667
    r18      0.45333
    r19         0.48
    r20      0.50667
    r21      0.53333
    r22         0.56
    r23      0.58667
    r24      0.61333
    r25         0.64
    r26      0.66667
    r27      0.69333
    r28         0.72
    r29      0.74667
    r30      0.77333
    r31          0.8
    r32      0.82667
    r33      0.85333
    r34         0.88
    r35      0.90667
    r36      0.93333
    r37         0.96
    r38      0.98667
    r39       1.0133
```

r40	1.04
r41	1.0667
r42	1.0933
r43	1.12
r44	1.1467
r45	1.1733
r46	1.2
r47	1.2267
r48	1.2533
r49	1.28
r50	1.3067
r152	4.06
r153	4.12
r154	4.18
r155	4.24
r156	4.3
r157	4.36
r158	4.42
r159	4.48
r160	4.54
r161	4.6
r162	4.66
r163	4.72
r164	4.78
r165	4.84
r166	4.9
r167	4.96
r168	5.02
r169	5.08
r170	5.14
r171	5.2
r172	5.26
r173	5.32
r174	5.38
r175	5.44
r176	5.5
r177	5.56
r178	5.62
r179	5.68
r180	5.74
r181	5.8
r182	5.86
r183	5.92
r184	5.98
r185	6.04
r186	6.1
r187	6.16
r188	6.22
r189	6.28
r190	6.34
r191	6.4
r192	6.46
r193	6.52
r194	6.58
r195	6.64
r196	6.7
r197	6.76
r198	6.82

r199	6.88
r200	6.94
r201	7

xxx TABLE:pi_unemp xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0.080278	0.051706	0.041502	0.03538	0.025176
r2	0.080278	0.051706	0.041502	0.03538	0.025176
r3	0.080278	0.051706	0.041502	0.03538	0.025176
r4	0.080278	0.051706	0.041502	0.03538	0.025176
r5	0.080278	0.051706	0.041502	0.03538	0.025176
r6	0.080278	0.051706	0.041502	0.03538	0.025176
r7	0.080278	0.051706	0.041502	0.03538	0.025176
r8	0.080278	0.051706	0.041502	0.03538	0.025176
r9	0.080278	0.051706	0.041502	0.03538	0.025176
r10	0.080278	0.051706	0.041502	0.03538	0.025176
r11	0.080278	0.051706	0.041502	0.03538	0.025176
r12	0.080278	0.051706	0.041502	0.03538	0.025176
r13	0.080278	0.051706	0.041502	0.03538	0.025176
r14	0.070703	0.042132	0.031928	0.025805	0.015601
r15	0.070703	0.042132	0.031928	0.025805	0.015601
r16	0.070703	0.042132	0.031928	0.025805	0.015601
r17	0.070703	0.042132	0.031928	0.025805	0.015601
r18	0.070703	0.042132	0.031928	0.025805	0.015601
r19	0.070703	0.042132	0.031928	0.025805	0.015601
r20	0.070703	0.042132	0.031928	0.025805	0.015601
r21	0.070703	0.042132	0.031928	0.025805	0.015601
r22	0.070703	0.042132	0.031928	0.025805	0.015601
r23	0.070703	0.042132	0.031928	0.025805	0.015601
r24	0.067512	0.038941	0.028736	0.022614	0.01241
r25	0.067512	0.038941	0.028736	0.022614	0.01241
r26	0.067512	0.038941	0.028736	0.022614	0.01241
r27	0.067512	0.038941	0.028736	0.022614	0.01241
r28	0.067512	0.038941	0.028736	0.022614	0.01241
r29	0.067512	0.038941	0.028736	0.022614	0.01241
r30	0.067512	0.038941	0.028736	0.022614	0.01241
r31	0.067512	0.038941	0.028736	0.022614	0.01241
r32	0.067512	0.038941	0.028736	0.022614	0.01241
r33	0.067512	0.038941	0.028736	0.022614	0.01241
r34	0.068576	0.040004	0.0298	0.023678	0.013474
r35	0.068576	0.040004	0.0298	0.023678	0.013474
r36	0.068576	0.040004	0.0298	0.023678	0.013474
r37	0.068576	0.040004	0.0298	0.023678	0.013474
r38	0.068576	0.040004	0.0298	0.023678	0.013474
r39	0.068576	0.040004	0.0298	0.023678	0.013474
r40	0.068576	0.040004	0.0298	0.023678	0.013474
r41	0.068576	0.040004	0.0298	0.023678	0.013474
r42	0.068576	0.040004	0.0298	0.023678	0.013474
r43	0.068576	0.040004	0.0298	0.023678	0.013474
r44	0.080278	0.051706	0.041502	0.03538	0.025176
r45	0.080278	0.051706	0.041502	0.03538	0.025176
r46	0.080278	0.051706	0.041502	0.03538	0.025176
r47	0.080278	0.051706	0.041502	0.03538	0.025176
r48	0.080278	0.051706	0.041502	0.03538	0.025176
r49	0	0	0	0	0
r50	0	0	0	0	0

r51	0	0	0	0	0
r52	0	0	0	0	0
r53	0	0	0	0	0
r54	0	0	0	0	0
r55	0	0	0	0	0
r56	0	0	0	0	0
r57	0	0	0	0	0
r58	0	0	0	0	0
r59	0	0	0	0	0
r60	0	0	0	0	0
r61	0	0	0	0	0
r62	0	0	0	0	0
r63	0	0	0	0	0
r64	0	0	0	0	0
r65	0	0	0	0	0
r66	0	0	0	0	0
r67	0	0	0	0	0
r68	0	0	0	0	0
r69	0	0	0	0	0
r70	0	0	0	0	0
r71	0	0	0	0	0
r72	0	0	0	0	0
r73	0	0	0	0	0
r74	0	0	0	0	0
r75	0	0	0	0	0
r76	0	0	0	0	0
r77	0	0	0	0	0
r78	0	0	0	0	0
r79	0	0	0	0	0
r80	0	0	0	0	0
r81	0	0	0	0	0
r82	0	0	0	0	0
r83	0	0	0	0	0

```
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	value
TR	1	1	0.0017225
b	2	2	1
n_incgrid	3	4	201
n_welfchecksgrid	4	5	45
scaleconvertor	5	7	58056
xi	6	8	0.75

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

	i	idx	ndim	numel	rowN	colN	sum	mean	std
agrid	1	1	2	65	65	1	2228	34.277	39.432
eta_H_grid	2	2	2	305	305	1	2.0872e-13	6.8433e-16	1.5853
eta_S_grid	3	3	2	305	305	1	-1.7764e-14	-5.8241e-17	2.2112

xxx TABLE:agrid xxxxxxxxxxxxxxxxxx

	c1
r1	0
r2	0.00051498
r3	0.0041199
r4	0.013905
r5	0.032959
r6	0.064373
r7	0.11124
r8	0.17664
r9	0.26367
r10	0.37542
r11	0.51498
r12	0.68544
r13	0.88989
r14	1.1314
r15	1.4131
r16	1.7381
r17	2.1094
r18	2.5301
r19	3.0034
r20	3.5323
r21	4.1199
r22	4.7693
r23	5.4836
r24	6.2658
r25	7.1191
r26	8.0466
r27	9.0514
r28	10.136
r29	11.305
r30	12.56
r31	13.905
r32	15.342
r33	16.875
r34	18.507
r35	20.241
r36	22.08
r37	24.027
r38	26.085
r39	28.258
r40	30.548
r41	32.959
r42	35.493
r43	38.154
r44	40.945
r45	43.868
r46	46.928
r47	50.126
r48	53.467
r49	56.953
r50	60.587
r51	64.373
r52	68.313
r53	72.411

```
    r54        76.669
    r55        81.091
    r56         85.68
    r57        90.439
    r58        95.371
    r59        100.48
    r60        105.77
    r61        111.24
    r62        116.89
    r63        122.74
    r64        128.77
    r65           135

xxx TABLE:eta_H_grid xxxxxxxxxxxxxxxxxx
                c1
            ----------

    r1         -2.6968
    r2         -2.6069
    r3          -2.517
    r4         -2.4271
    r5         -2.3372
    r6         -2.2473
    r7         -2.1574
    r8         -2.0675
    r9         -1.9777
    r10        -1.8878
    r11        -1.7979
    r12         -1.708
    r13        -1.6181
    r14        -1.5282
    r15        -1.4383
    r16        -1.3484
    r17        -1.2585
    r18        -1.1686
    r19        -1.0787
    r20       -0.98883
    r21       -0.89893
    r22       -0.80904
    r23       -0.71915
    r24       -0.62925
    r25       -0.53936
    r26       -0.44947
    r27       -0.35957
    r28       -0.26968
    r29       -0.17979
    r30      -0.089893
    r31     8.0491e-16
    r32       0.089893
    r33        0.17979
    r34        0.26968
    r35        0.35957
    r36        0.44947
    r37        0.53936
    r38        0.62925
    r39        0.71915
    r40        0.80904
    r41        0.89893
```

r42	0.98883
r43	1.0787
r44	1.1686
r45	1.2585
r46	1.3484
r47	1.4383
r48	1.5282
r49	1.6181
r50	1.708
r256	-1.708
r257	-1.6181
r258	-1.5282
r259	-1.4383
r260	-1.3484
r261	-1.2585
r262	-1.1686
r263	-1.0787
r264	-0.98883
r265	-0.89893
r266	-0.80904
r267	-0.71915
r268	-0.62925
r269	-0.53936
r270	-0.44947
r271	-0.35957
r272	-0.26968
r273	-0.17979
r274	-0.089893
r275	8.0491e-16
r276	0.089893
r277	0.17979
r278	0.26968
r279	0.35957
r280	0.44947
r281	0.53936
r282	0.62925
r283	0.71915
r284	0.80904
r285	0.89893
r286	0.98883
r287	1.0787
r288	1.1686
r289	1.2585
r290	1.3484
r291	1.4383
r292	1.5282
r293	1.6181
r294	1.708
r295	1.7979
r296	1.8878
r297	1.9777
r298	2.0675
r299	2.1574
r300	2.2473
r301	2.3372
r302	2.4271
r303	2.517
r304	2.6069

```
    r305        2.6968

xxx TABLE:eta_S_grid xxxxxxxxxxxxxxxxxx
              c1
            ------

    r1      -3.122
    r2      -3.122
    r3      -3.122
    r4      -3.122
    r5      -3.122
    r6      -3.122
    r7      -3.122
    r8      -3.122
    r9      -3.122
    r10     -3.122
    r11     -3.122
    r12     -3.122
    r13     -3.122
    r14     -3.122
    r15     -3.122
    r16     -3.122
    r17     -3.122
    r18     -3.122
    r19     -3.122
    r20     -3.122
    r21     -3.122
    r22     -3.122
    r23     -3.122
    r24     -3.122
    r25     -3.122
    r26     -3.122
    r27     -3.122
    r28     -3.122
    r29     -3.122
    r30     -3.122
    r31     -3.122
    r32     -3.122
    r33     -3.122
    r34     -3.122
    r35     -3.122
    r36     -3.122
    r37     -3.122
    r38     -3.122
    r39     -3.122
    r40     -3.122
    r41     -3.122
    r42     -3.122
    r43     -3.122
    r44     -3.122
    r45     -3.122
    r46     -3.122
    r47     -3.122
    r48     -3.122
    r49     -3.122
    r50     -3.122
    r256     3.122
    r257     3.122
```

```
    r258     3.122
    r259     3.122
    r260     3.122
    r261     3.122
    r262     3.122
    r263     3.122
    r264     3.122
    r265     3.122
    r266     3.122
    r267     3.122
    r268     3.122
    r269     3.122
    r270     3.122
    r271     3.122
    r272     3.122
    r273     3.122
    r274     3.122
    r275     3.122
    r276     3.122
    r277     3.122
    r278     3.122
    r279     3.122
    r280     3.122
    r281     3.122
    r282     3.122
    r283     3.122
    r284     3.122
    r285     3.122
    r286     3.122
    r287     3.122
    r288     3.122
    r289     3.122
    r290     3.122
    r291     3.122
    r292     3.122
    r293     3.122
    r294     3.122
    r295     3.122
    r296     3.122
    r297     3.122
    r298     3.122
    r299     3.122
    r300     3.122
    r301     3.122
    r302     3.122
    r303     3.122
    r304     3.122
    r305     3.122

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_exotrans ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                            i    idx    ndim      numel       rowN    colN     sum        mean
                            -    ---    ----    ----------    ----    ----    -----    ----------

    cl_mt_pi_jem_kidseta    1     1      2      2.3256e+06    1525    1525    1525     0.00065574
    pi_H_eta                2     2      2            3721      61      61      61       0.016393
```

pi_S_eta	3	3	2	25	5	5	5	0.2
pi_eta	4	4	2	93025	305	305	305	0.0032787
pi_kids	5	5	5	8300	5	1660	1660	0.2
psi	6	6	2	83	83	1	78.16	0.94169

xxx TABLE:cl_mt_pi_jem_kidseta xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c1523	c1524	c1525
r1	0.005125	0.0027446	0.0018598	0	0	0
r2	0.0025409	0.0026419	0.0027349	0	0	0
r3	0.00090694	0.0016787	0.002655	0	0	0
r4	0.00022557	0.00070391	0.0017014	0	0	0
r5	3.8282e-05	0.00019466	0.00071949	0	0	0
r6	4.3734e-06	3.5476e-05	0.00020067	0	0	0
r7	3.333e-07	4.2565e-06	3.6884e-05	0	0	0
r8	1.6842e-08	3.3588e-07	4.4633e-06	0	0	0
r9	5.6182e-10	1.7412e-08	3.5523e-07	0	0	0
r10	1.2334e-11	5.9238e-10	1.8574e-08	0	0	0
r11	1.7779e-13	1.3212e-11	6.3738e-10	0	0	0
r12	1.6798e-15	1.93e-13	1.4339e-11	0	0	0
r13	1.0388e-17	1.8449e-15	2.1128e-13	0	0	0
r14	4.2001e-20	1.1531e-17	2.0372e-15	0	0	0
r15	1.1095e-22	4.709e-20	1.2844e-17	0	0	0
r16	1.9132e-25	1.2558e-22	5.291e-20	0	0	0
r17	2.1526e-28	2.1857e-25	1.4233e-22	0	0	0
r18	1.5795e-31	2.4815e-28	2.4989e-25	0	0	0
r19	7.5549e-35	1.8372e-31	2.862e-28	0	0	0
r20	2.3549e-38	8.8661e-35	2.1375e-31	0	0	0
r21	4.7821e-42	2.7881e-38	1.0406e-34	0	0	0
r22	6.3249e-46	5.7119e-42	3.301e-38	0	0	0
r23	5.4473e-50	7.6214e-46	6.8221e-42	0	0	0
r24	3.0545e-54	6.6219e-50	9.1828e-46	0	0	0
r25	1.1149e-58	3.7458e-54	8.0487e-50	0	0	0
r26	2.6486e-63	1.3793e-58	4.593e-54	0	0	0
r27	4.0946e-68	3.3055e-63	1.7061e-58	0	0	0
r28	4.119e-73	5.1553e-68	4.1248e-63	0	0	0
r29	2.6959e-78	5.2318e-73	6.4897e-68	0	0	0
r30	1.1479e-83	3.4544e-78	6.6439e-73	0	0	0
r31	3.1796e-89	1.4838e-83	4.4254e-78	0	0	0
r32	5.7288e-95	4.1463e-89	1.9177e-83	0	0	0
r33	6.7133e-101	7.5363e-95	5.4059e-89	0	0	0
r34	5.1166e-107	8.9094e-101	9.9124e-95	0	0	0
r35	2.5362e-113	6.8503e-107	1.1822e-100	0	0	0
r36	8.1749e-120	3.4254e-113	9.1697e-107	0	0	0
r37	1.7135e-126	1.1139e-119	4.6256e-113	0	0	0
r38	2.3355e-133	2.3554e-126	1.5174e-119	0	0	0
r39	2.0699e-140	3.2387e-133	3.237e-126	0	0	0
r40	1.1927e-147	2.8956e-140	4.4902e-133	0	0	0
r41	4.4687e-155	1.6833e-147	4.05e-140	0	0	0
r42	1.0885e-162	6.3622e-155	2.3751e-147	0	0	0
r43	1.7238e-170	1.5634e-162	9.0564e-155	0	0	0
r44	1.7748e-178	2.4978e-170	2.2451e-162	0	0	0
r45	1.1879e-186	2.5943e-178	3.6186e-170	0	0	0
r46	5.1689e-195	1.7518e-186	3.7916e-178	0	0	0
r47	1.4621e-203	7.6897e-195	2.5828e-186	5.8763e-24	0	0
r48	2.6885e-212	2.1943e-203	1.1438e-194	9.0496e-22	5.3421e-24	0
r49	3.2135e-221	4.0705e-212	3.2927e-203	9.3883e-20	8.1948e-22	4.8079e-24

r50	2.4969e-230	4.9085e-221	6.162e-212	6.3716e-18	8.5761e-20	7.4629e-22
r1476	3.4485e-22	3.9622e-20	2.9437e-18	0	0	0
r1477	2.1325e-24	3.7874e-22	4.3374e-20	0	0	0
r1478	8.6227e-27	2.3672e-24	4.1822e-22	0	0	0
r1479	2.2777e-29	9.6675e-27	2.6367e-24	0	0	0
r1480	3.9278e-32	2.5781e-29	1.0862e-26	0	0	0
r1481	4.4192e-35	4.4871e-32	2.922e-29	0	0	0
r1482	3.2426e-38	5.0945e-35	5.1302e-32	0	0	0
r1483	1.551e-41	3.7717e-38	5.8757e-35	0	0	0
r1484	4.8345e-45	1.8202e-41	4.3882e-38	0	0	0
r1485	9.8174e-49	5.7239e-45	2.1363e-41	0	0	0
r1486	1.2985e-52	1.1726e-48	6.7769e-45	0	0	0
r1487	1.1183e-56	1.5647e-52	1.4006e-48	0	0	0
r1488	6.2707e-61	1.3595e-56	1.8852e-52	0	0	0
r1489	2.2888e-65	7.69e-61	1.6524e-56	0	0	0
r1490	5.4374e-70	2.8316e-65	9.4292e-61	0	0	0
r1491	8.4061e-75	6.7861e-70	3.5026e-65	0	0	0
r1492	8.4562e-80	1.0584e-74	8.468e-70	0	0	0
r1493	5.5346e-85	1.0741e-79	1.3323e-74	0	0	0
r1494	2.3566e-90	7.0917e-85	1.364e-79	0	0	0
r1495	6.5276e-96	3.0463e-90	9.0853e-85	0	0	0
r1496	1.1761e-101	8.5122e-96	3.937e-90	0	0	0
r1497	1.3782e-107	1.5472e-101	1.1098e-95	0	0	0
r1498	1.0504e-113	1.8291e-107	2.035e-101	0	0	0
r1499	5.2066e-120	1.4063e-113	2.427e-107	0	0	0
r1500	1.6783e-126	7.0322e-120	1.8825e-113	0	0	0
r1501	3.5178e-133	2.2867e-126	9.4963e-120	0	0	0
r1502	4.7948e-140	4.8355e-133	3.1152e-126	0	0	0
r1503	4.2493e-147	6.6489e-140	6.6455e-133	0	0	0
r1504	2.4486e-154	5.9445e-147	9.2183e-140	0	0	0
r1505	9.174e-162	3.4557e-154	8.3145e-147	0	0	0
r1506	2.2347e-169	1.3061e-161	4.8761e-154	0	0	0
r1507	3.539e-177	3.2097e-169	1.8592e-161	0	0	0
r1508	3.6436e-185	5.1279e-177	4.6092e-169	0	0	0
r1509	2.4388e-193	5.3261e-185	7.4288e-177	0	0	0
r1510	1.0612e-201	3.5964e-193	7.7841e-185	0	0	0
r1511	3.0016e-210	1.5787e-201	5.3025e-193	1.409e-17	0	0
r1512	5.5194e-219	4.5049e-210	2.3481e-201	2.1698e-15	1.2809e-17	0
r1513	6.5973e-228	8.3567e-219	6.7598e-210	2.2511e-13	1.9649e-15	1.1528e-17
r1514	5.126e-237	1.0077e-227	1.265e-218	1.5277e-11	2.0563e-13	1.7894e-15
r1515	2.5889e-246	7.8987e-237	1.5389e-227	6.791e-10	1.4077e-11	1.8943e-13
r1516	8.4994e-256	4.0246e-246	1.2169e-236	1.979e-08	6.3116e-10	1.3141e-11
r1517	1.8137e-265	1.3329e-255	6.2552e-246	3.7848e-07	1.8552e-08	5.986e-10
r1518	2.5158e-275	2.8695e-265	2.09e-255	4.7555e-06	3.5786e-07	1.7944e-08
r1519	2.2681e-285	4.0153e-275	4.539e-265	3.9298e-05	4.5351e-06	3.5512e-07
r1520	1.3291e-295	3.6521e-285	6.4076e-275	0.00021381	3.7798e-05	4.6596e-06
r1521	5.0624e-306	2.159e-295	5.8794e-285	0.00076658	0.0002074	4.0788e-05
r1522	1.2533e-316	8.296e-306	3.5064e-295	0.0018127	0.00074998	0.00024034
r1523	0	2.0719e-316	1.3592e-305	0.0028288	0.0017886	0.0009663
r1524	0	0	3.4246e-316	0.0029139	0.0028149	0.0027072
r1525	0	0	0	0.0019815	0.0029243	0.0054605

xxx TABLE:pi_H_eta xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c59	c60	c61
r1	0.47329	0.25346	0.17175	0	0	0
r2	0.23465	0.24398	0.25257	0	0	0

r3	0.083755	0.15503	0.24518	0	0	0
r4	0.020831	0.065005	0.15712	0	0	0
r5	0.0035353	0.017977	0.066444	0	0	0
r6	0.00040387	0.0032761	0.018532	0	0	0
r7	3.078e-05	0.00039308	0.0034061	0	0	0
r8	1.5553e-06	3.1018e-05	0.00041218	0	0	0
r9	5.1884e-08	1.608e-06	3.2805e-05	0	0	0
r10	1.139e-09	5.4706e-08	1.7153e-06	0	0	0
r11	1.6419e-11	1.2201e-09	5.8861e-08	0	0	0
r12	1.5512e-13	1.7823e-11	1.3242e-09	0	0	0
r13	9.5928e-16	1.7037e-13	1.9511e-11	0	0	0
r14	3.8788e-18	1.0648e-15	1.8813e-13	0	0	0
r15	1.0246e-20	4.3487e-18	1.1861e-15	0	0	0
r16	1.7668e-23	1.1597e-20	4.8861e-18	0	0	0
r17	1.9879e-26	2.0184e-23	1.3144e-20	0	0	0
r18	1.4586e-29	2.2917e-26	2.3077e-23	0	0	0
r19	6.9768e-33	1.6966e-29	2.6431e-26	0	0	0
r20	2.1747e-36	8.1877e-33	1.9739e-29	0	0	0
r21	4.4162e-40	2.5748e-36	9.6095e-33	0	0	0
r22	5.8409e-44	5.2749e-40	3.0485e-36	0	0	0
r23	5.0305e-48	7.0383e-44	6.3001e-40	0	0	0
r24	2.8208e-52	6.1152e-48	8.4802e-44	0	0	0
r25	1.0296e-56	3.4592e-52	7.4328e-48	0	0	0
r26	2.4459e-61	1.2737e-56	4.2415e-52	0	0	0
r27	3.7813e-66	3.0526e-61	1.5756e-56	0	0	0
r28	3.8039e-71	4.7609e-66	3.8092e-61	0	0	0
r29	2.4896e-76	4.8314e-71	5.9931e-66	0	0	0
r30	1.0601e-81	3.1901e-76	6.1355e-71	0	0	0
r31	2.9363e-87	1.3703e-81	4.0868e-76	0	0	0
r32	5.2904e-93	3.829e-87	1.771e-81	0	0	0
r33	6.1997e-99	6.9596e-93	4.9923e-87	0	0	0
r34	4.7251e-105	8.2277e-99	9.1539e-93	0	0	0
r35	2.3421e-111	6.3261e-105	1.0917e-98	0	0	0
r36	7.5494e-118	3.1633e-111	8.4681e-105	0	0	0
r37	1.5824e-124	1.0286e-117	4.2717e-111	0	0	0
r38	2.1568e-131	2.1752e-124	1.4013e-117	0	0	0
r39	1.9115e-138	2.9909e-131	2.9894e-124	0	0	0
r40	1.1015e-145	2.674e-138	4.1467e-131	0	0	0
r41	4.1268e-153	1.5545e-145	3.7401e-138	0	0	0
r42	1.0052e-160	5.8754e-153	2.1934e-145	0	0	0
r43	1.5919e-168	1.4438e-160	8.3634e-153	0	0	0
r44	1.639e-176	2.3067e-168	2.0733e-160	0	0	0
r45	1.097e-184	2.3958e-176	3.3417e-168	0	0	0
r46	4.7734e-193	1.6178e-184	3.5015e-176	0	0	0
r47	1.3502e-201	7.1013e-193	2.3852e-184	1.2212e-15	0	0
r48	2.4828e-210	2.0264e-201	1.0563e-192	1.8807e-13	1.1102e-15	0
r49	2.9676e-219	3.7591e-210	3.0408e-201	1.9511e-11	1.7031e-13	9.992e-16
r50	2.3058e-228	4.5329e-219	5.6905e-210	1.3242e-09	1.7823e-11	1.551e-13
r51	1.1646e-237	3.5531e-228	6.9224e-219	5.8861e-08	1.2201e-09	1.6419e-11
r52	3.8233e-247	1.8104e-237	5.474e-228	1.7153e-06	5.4706e-08	1.139e-09
r53	8.1587e-257	5.9959e-247	2.8138e-237	3.2805e-05	1.608e-06	5.1884e-08
r54	1.1317e-266	1.2908e-256	9.4014e-247	0.00041218	3.1018e-05	1.5553e-06
r55	1.0203e-276	1.8062e-266	2.0418e-256	0.0034061	0.00039308	3.078e-05
r56	5.9788e-287	1.6428e-276	2.8823e-266	0.018532	0.0032761	0.00040387
r57	2.2772e-297	9.7119e-287	2.6447e-276	0.066444	0.017977	0.0035353
r58	5.6375e-308	3.7318e-297	1.5773e-286	0.15712	0.065005	0.020831
r59	9.0709e-319	9.32e-308	6.1143e-297	0.24518	0.15503	0.083755
r60	0	1.5129e-318	1.5405e-307	0.25257	0.24398	0.23465

r61	0	0	2.5227e-318	0.17175	0.25346	0.47329

xxx TABLE:pi_S_eta xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0.012224	0.2144	0.54675	0.2144	0.012224
r2	0.012224	0.2144	0.54675	0.2144	0.012224
r3	0.012224	0.2144	0.54675	0.2144	0.012224
r4	0.012224	0.2144	0.54675	0.2144	0.012224
r5	0.012224	0.2144	0.54675	0.2144	0.012224

xxx TABLE:pi_eta xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c303	c304	c305
r1	0.0057857	0.0030985	0.0020995	0	0	0
r2	0.0028684	0.0029825	0.0030875	0	0	0
r3	0.0010239	0.0018951	0.0029973	0	0	0
r4	0.00025465	0.00079465	0.0019207	0	0	0
r5	4.3217e-05	0.00021976	0.00081224	0	0	0
r6	4.9371e-06	4.0049e-05	0.00022654	0	0	0
r7	3.7627e-07	4.8052e-06	4.1638e-05	0	0	0
r8	1.9013e-08	3.7918e-07	5.0387e-06	0	0	0
r9	6.3425e-10	1.9657e-08	4.0102e-07	0	0	0
r10	1.3924e-11	6.6875e-10	2.0969e-08	0	0	0
r11	2.0071e-13	1.4915e-11	7.1954e-10	0	0	0
r12	1.8963e-15	2.1788e-13	1.6187e-11	0	0	0
r13	1.1727e-17	2.0827e-15	2.3851e-13	0	0	0
r14	4.7416e-20	1.3017e-17	2.2998e-15	0	0	0
r15	1.2525e-22	5.3161e-20	1.4499e-17	0	0	0
r16	2.1598e-25	1.4177e-22	5.973e-20	0	0	0
r17	2.4301e-28	2.4674e-25	1.6068e-22	0	0	0
r18	1.7831e-31	2.8014e-28	2.821e-25	0	0	0
r19	8.5288e-35	2.074e-31	3.231e-28	0	0	0
r20	2.6585e-38	1.0009e-34	2.413e-31	0	0	0
r21	5.3985e-42	3.1476e-38	1.1747e-34	0	0	0
r22	7.1402e-46	6.4483e-42	3.7266e-38	0	0	0
r23	6.1496e-50	8.6039e-46	7.7016e-42	0	0	0
r24	3.4482e-54	7.4756e-50	1.0367e-45	0	0	0
r25	1.2586e-58	4.2287e-54	9.0863e-50	0	0	0
r26	2.99e-63	1.5571e-58	5.185e-54	0	0	0
r27	4.6225e-68	3.7316e-63	1.926e-58	0	0	0
r28	4.65e-73	5.8199e-68	4.6565e-63	0	0	0
r29	3.0435e-78	5.9062e-73	7.3263e-68	0	0	0
r30	1.2959e-83	3.8997e-78	7.5004e-73	0	0	0
r31	3.5895e-89	1.6751e-83	4.9959e-78	0	0	0
r32	6.4672e-95	4.6808e-89	2.1649e-83	0	0	0
r33	7.5788e-101	8.5078e-95	6.1028e-89	0	0	0
r34	5.7762e-107	1.0058e-100	1.119e-94	0	0	0
r35	2.8631e-113	7.7334e-107	1.3346e-100	0	0	0
r36	9.2288e-120	3.867e-113	1.0352e-106	0	0	0
r37	1.9344e-126	1.2575e-119	5.2219e-113	0	0	0
r38	2.6366e-133	2.659e-126	1.713e-119	0	0	0
r39	2.3367e-140	3.6562e-133	3.6543e-126	0	0	0
r40	1.3465e-147	3.2689e-140	5.0691e-133	0	0	0
r41	5.0447e-155	1.9003e-147	4.5721e-140	0	0	0
r42	1.2288e-162	7.1823e-155	2.6813e-147	0	0	0

r43	1.9461e-170	1.765e-162	1.0224e-154	0	0	0
r44	2.0036e-178	2.8198e-170	2.5346e-162	0	0	0
r45	1.3411e-186	2.9288e-178	4.085e-170	0	0	0
r46	5.8353e-195	1.9776e-186	4.2804e-178	0	0	0
r47	1.6506e-203	8.681e-195	2.9158e-186	1.4929e-17	0	0
r48	3.0351e-212	2.4772e-203	1.2912e-194	2.2991e-15	1.3572e-17	0
r49	3.6278e-221	4.5953e-212	3.7172e-203	2.3851e-13	2.0819e-15	1.2215e-17
r50	2.8187e-230	5.5412e-221	6.9563e-212	1.6187e-11	2.1788e-13	1.896e-15
r256	1.8963e-15	2.1788e-13	1.6187e-11	0	0	0
r257	1.1727e-17	2.0827e-15	2.3851e-13	0	0	0
r258	4.7416e-20	1.3017e-17	2.2998e-15	0	0	0
r259	1.2525e-22	5.3161e-20	1.4499e-17	0	0	0
r260	2.1598e-25	1.4177e-22	5.973e-20	0	0	0
r261	2.4301e-28	2.4674e-25	1.6068e-22	0	0	0
r262	1.7831e-31	2.8014e-28	2.821e-25	0	0	0
r263	8.5288e-35	2.074e-31	3.231e-28	0	0	0
r264	2.6585e-38	1.0009e-34	2.413e-31	0	0	0
r265	5.3985e-42	3.1476e-38	1.1747e-34	0	0	0
r266	7.1402e-46	6.4483e-42	3.7266e-38	0	0	0
r267	6.1496e-50	8.6039e-46	7.7016e-42	0	0	0
r268	3.4482e-54	7.4756e-50	1.0367e-45	0	0	0
r269	1.2586e-58	4.2287e-54	9.0863e-50	0	0	0
r270	2.99e-63	1.5571e-58	5.185e-54	0	0	0
r271	4.6225e-68	3.7316e-63	1.926e-58	0	0	0
r272	4.65e-73	5.8199e-68	4.6565e-63	0	0	0
r273	3.0435e-78	5.9062e-73	7.3263e-68	0	0	0
r274	1.2959e-83	3.8997e-78	7.5004e-73	0	0	0
r275	3.5895e-89	1.6751e-83	4.9959e-78	0	0	0
r276	6.4672e-95	4.6808e-89	2.1649e-83	0	0	0
r277	7.5788e-101	8.5078e-95	6.1028e-89	0	0	0
r278	5.7762e-107	1.0058e-100	1.119e-94	0	0	0
r279	2.8631e-113	7.7334e-107	1.3346e-100	0	0	0
r280	9.2288e-120	3.867e-113	1.0352e-106	0	0	0
r281	1.9344e-126	1.2575e-119	5.2219e-113	0	0	0
r282	2.6366e-133	2.659e-126	1.713e-119	0	0	0
r283	2.3367e-140	3.6562e-133	3.6543e-126	0	0	0
r284	1.3465e-147	3.2689e-140	5.0691e-133	0	0	0
r285	5.0447e-155	1.9003e-147	4.5721e-140	0	0	0
r286	1.2288e-162	7.1823e-155	2.6813e-147	0	0	0
r287	1.9461e-170	1.765e-162	1.0224e-154	0	0	0
r288	2.0036e-178	2.8198e-170	2.5346e-162	0	0	0
r289	1.3411e-186	2.9288e-178	4.085e-170	0	0	0
r290	5.8353e-195	1.9776e-186	4.2804e-178	0	0	0
r291	1.6506e-203	8.681e-195	2.9158e-186	1.4929e-17	0	0
r292	3.0351e-212	2.4772e-203	1.2912e-194	2.2991e-15	1.3572e-17	0
r293	3.6278e-221	4.5953e-212	3.7172e-203	2.3851e-13	2.0819e-15	1.2215e-17
r294	2.8187e-230	5.5412e-221	6.9563e-212	1.6187e-11	2.1788e-13	1.896e-15
r295	1.4236e-239	4.3435e-230	8.4623e-221	7.1954e-10	1.4915e-11	2.0071e-13
r296	4.6738e-249	2.2131e-239	6.6917e-230	2.0969e-08	6.6875e-10	1.3924e-11
r297	9.9736e-259	7.3296e-249	3.4397e-239	4.0102e-07	1.9657e-08	6.3425e-10
r298	1.3834e-268	1.5779e-258	1.1493e-248	5.0387e-06	3.7918e-07	1.9013e-08
r299	1.2472e-278	2.208e-268	2.496e-258	4.1638e-05	4.8052e-06	3.7627e-07
r300	7.3088e-289	2.0082e-278	3.5235e-268	0.00022654	4.0049e-05	4.9371e-06
r301	2.7838e-299	1.1872e-288	3.233e-278	0.00081224	0.00021976	4.3217e-05
r302	6.8916e-310	4.5619e-299	...			

1.1.2 Parameters Used for Test Simulation

Rather than solving for all ages between 18 to 100, this solves for age groups, and has limited shocks and asset levels.

```
mp_params = snw_mp_param('default_small', true, 100, 6);

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_preftechpricegov Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                            i     idx     value
                            --    ---    -------

    Bequests                 1     1           0
    a2                       2     2      1.5286
    bequests_option          3     3           1
    beta                     4     4     0.86389
    cons_allocation_rule     5     5           2
    g_cons                   6     6     0.17576
    g_n                      7     7     0.05101
    gamma                    8     8           2
    jret                     9     9          13
    r                       10    10     0.21665
    theta                   11    11     0.56523
    throw_in_ocean          12    12           1

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_intlen Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                     i    idx    value
                     -    ---    -----

    n_agrid          1     1      25
    n_educgrid       2     2       2
    n_eta_H_grid     3     3       5
    n_eta_S_grid     4     4       1
    n_etagrid        5     5       5
    n_jgrid          6     6      18
    n_kidsgrid       7     7       3
    n_marriedgrid    8     8       2

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim    numel    rowN    colN     sum        mean         std        coefv
                -    ---    ----    -----    ----    ----    ------    --------    --------    -----

    inc_grid    1     3      2       201     201      1      578.5      2.8781      1.8836    0.654
    pi_unemp    2     6      2       240      48      5     9.5319    0.039716    0.019674    0.495

xxx TABLE:inc_grid xxxxxxxxxxxxxxxxx
              c1
           --------

    r1            0
    r2     0.026667
```

r3	0.053333
r4	0.08
r5	0.10667
r6	0.13333
r7	0.16
r8	0.18667
r9	0.21333
r10	0.24
r11	0.26667
r12	0.29333
r13	0.32
r14	0.34667
r15	0.37333
r16	0.4
r17	0.42667
r18	0.45333
r19	0.48
r20	0.50667
r21	0.53333
r22	0.56
r23	0.58667
r24	0.61333
r25	0.64
r26	0.66667
r27	0.69333
r28	0.72
r29	0.74667
r30	0.77333
r31	0.8
r32	0.82667
r33	0.85333
r34	0.88
r35	0.90667
r36	0.93333
r37	0.96
r38	0.98667
r39	1.0133
r40	1.04
r41	1.0667
r42	1.0933
r43	1.12
r44	1.1467
r45	1.1733
r46	1.2
r47	1.2267
r48	1.2533
r49	1.28
r50	1.3067
r152	4.06
r153	4.12
r154	4.18
r155	4.24
r156	4.3
r157	4.36
r158	4.42
r159	4.48
r160	4.54
r161	4.6

r162	4.66
r163	4.72
r164	4.78
r165	4.84
r166	4.9
r167	4.96
r168	5.02
r169	5.08
r170	5.14
r171	5.2
r172	5.26
r173	5.32
r174	5.38
r175	5.44
r176	5.5
r177	5.56
r178	5.62
r179	5.68
r180	5.74
r181	5.8
r182	5.86
r183	5.92
r184	5.98
r185	6.04
r186	6.1
r187	6.16
r188	6.22
r189	6.28
r190	6.34
r191	6.4
r192	6.46
r193	6.52
r194	6.58
r195	6.64
r196	6.7
r197	6.76
r198	6.82
r199	6.88
r200	6.94
r201	7

xxx TABLE:pi_unemp xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
	--------	--------	--------	--------	--------
r1	0.080278	0.051706	0.041502	0.03538	0.025176
r2	0.080278	0.051706	0.041502	0.03538	0.025176
r3	0.080278	0.051706	0.041502	0.03538	0.025176
r4	0.080278	0.051706	0.041502	0.03538	0.025176
r5	0.080278	0.051706	0.041502	0.03538	0.025176
r6	0.080278	0.051706	0.041502	0.03538	0.025176
r7	0.080278	0.051706	0.041502	0.03538	0.025176
r8	0.080278	0.051706	0.041502	0.03538	0.025176
r9	0.080278	0.051706	0.041502	0.03538	0.025176
r10	0.080278	0.051706	0.041502	0.03538	0.025176
r11	0.080278	0.051706	0.041502	0.03538	0.025176
r12	0.080278	0.051706	0.041502	0.03538	0.025176
r13	0.080278	0.051706	0.041502	0.03538	0.025176

r14	0.070703	0.042132	0.031928	0.025805	0.015601
r15	0.070703	0.042132	0.031928	0.025805	0.015601
r16	0.070703	0.042132	0.031928	0.025805	0.015601
r17	0.070703	0.042132	0.031928	0.025805	0.015601
r18	0.070703	0.042132	0.031928	0.025805	0.015601
r19	0.070703	0.042132	0.031928	0.025805	0.015601
r20	0.070703	0.042132	0.031928	0.025805	0.015601
r21	0.070703	0.042132	0.031928	0.025805	0.015601
r22	0.070703	0.042132	0.031928	0.025805	0.015601
r23	0.070703	0.042132	0.031928	0.025805	0.015601
r24	0.067512	0.038941	0.028736	0.022614	0.01241
r25	0.067512	0.038941	0.028736	0.022614	0.01241
r26	0.067512	0.038941	0.028736	0.022614	0.01241
r27	0.067512	0.038941	0.028736	0.022614	0.01241
r28	0.067512	0.038941	0.028736	0.022614	0.01241
r29	0.067512	0.038941	0.028736	0.022614	0.01241
r30	0.067512	0.038941	0.028736	0.022614	0.01241
r31	0.067512	0.038941	0.028736	0.022614	0.01241
r32	0.067512	0.038941	0.028736	0.022614	0.01241
r33	0.067512	0.038941	0.028736	0.022614	0.01241
r34	0.068576	0.040004	0.0298	0.023678	0.013474
r35	0.068576	0.040004	0.0298	0.023678	0.013474
r36	0.068576	0.040004	0.0298	0.023678	0.013474
r37	0.068576	0.040004	0.0298	0.023678	0.013474
r38	0.068576	0.040004	0.0298	0.023678	0.013474
r39	0.068576	0.040004	0.0298	0.023678	0.013474
r40	0.068576	0.040004	0.0298	0.023678	0.013474
r41	0.068576	0.040004	0.0298	0.023678	0.013474
r42	0.068576	0.040004	0.0298	0.023678	0.013474
r43	0.068576	0.040004	0.0298	0.023678	0.013474
r44	0.080278	0.051706	0.041502	0.03538	0.025176
r45	0.080278	0.051706	0.041502	0.03538	0.025176
r46	0.080278	0.051706	0.041502	0.03538	0.025176
r47	0.080278	0.051706	0.041502	0.03538	0.025176
r48	0.080278	0.051706	0.041502	0.03538	0.025176

----

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_covid_unemploy Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

	i	idx	value
	-	---	---------
TR	1	1	0.0017225
b	2	2	1
n_incgrid	3	4	201
n_welfchecksgrid	4	5	45
scaleconvertor	5	7	58056
xi	6	8	0.75

----

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_statesgrid ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

	i	idx	ndim	numel	rowN	colN	sum	mean	std
	-	---	----	-----	----	----	-----------	-----------	------
agrid	1	1	2	25	25	1	878.91	35.156	41.372

eta_H_grid	2	2	2	5	5	1	-2.2204e-16	-4.4409e-17	1.4543
eta_S_grid	3	3	2	5	5	1	0	0	0

xxx TABLE:agrid xxxxxxxxxxxxxxxxxx

	c1
r1	0
r2	0.0097656
r3	0.078125
r4	0.26367
r5	0.625
r6	1.2207
r7	2.1094
r8	3.3496
r9	5
r10	7.1191
r11	9.7656
r12	12.998
r13	16.875
r14	21.455
r15	26.797
r16	32.959
r17	40
r18	47.979
r19	56.953
r20	66.982
r21	78.125
r22	90.439
r23	103.98
r24	118.82
r25	135

xxx TABLE:eta_H_grid xxxxxxxxxxxxxxxxxx

	c1
r1	-1.8395
r2	-0.91976
r3	0
r4	0.91976
r5	1.8395

xxx TABLE:eta_S_grid xxxxxxxxxxxxxxxxxx

	c1
r1	0
r2	0
r3	0
r4	0
r5	0

----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_exotrans ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

i	idx	ndim	numel	rowN	colN	sum	mean	std

	-	---	----	-----	----	----	------	--------	-----
cl_mt_pi_jem_kidseta	1	1	2	225	15	15	15	0.066667	0.200
pi_H_eta	2	2	2	25	5	5	5	0.2	0.385
pi_eta	3	4	2	25	5	5	5	0.2	0.385
pi_kids	4	5	5	648	3	216	216	0.33333	0.356
psi	5	6	2	18	18	1	14.251	0.79171	0.312

xxx TABLE:cl_mt_pi_jem_kidseta xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c13	c14	c15
r1	0.8194	0.066439	4.258e-10	1.3413e-12	0	0
r2	0.0023536	0.85739	0.026096	8.2205e-05	7.2608e-14	0
r3	1.0239e-12	0.008585	0.86867	0.0027364	2.7043e-05	3.2254e-15
r4	1.1339e-29	2.3049e-11	0.026096	8.2205e-05	0.0027009	7.414e-06
r5	2.525e-54	1.6656e-27	4.258e-10	1.3413e-12	0.00020929	0.0025812
r6	0.047493	0.0038508	2.468e-11	1.3763e-10	0	0
r7	0.00013641	0.049695	0.0015125	0.0084347	7.4499e-12	0
r8	5.9343e-14	0.00049759	0.050348	0.28077	0.0027748	3.3094e-13
r9	6.5721e-31	1.3359e-12	0.0015125	0.0084347	0.27712	0.00076071
r10	1.4635e-55	9.6537e-29	2.468e-11	1.3763e-10	0.021474	0.26484
r11	0.0013898	0.00011269	7.2224e-13	4.4952e-10	0	0
r12	3.9921e-06	0.0014543	4.4263e-05	0.02755	2.4333e-11	0
r13	1.7366e-15	1.4562e-05	0.0014734	0.91706	0.0090632	1.0809e-12
r14	1.9233e-32	3.9096e-14	4.4263e-05	0.02755	0.90515	0.0024847
r15	4.2828e-57	2.8251e-30	7.2224e-13	4.4952e-10	0.07014	0.86505

xxx TABLE:pi_H_eta xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0.925	0.075001	4.8068e-10	0	0
r2	0.0026569	0.96788	0.029459	2.602e-11	0
r3	1.1558e-12	0.0096913	0.98062	0.0096913	1.1559e-12
r4	1.28e-29	2.602e-11	0.029459	0.96788	0.0026569
r5	2.8504e-54	1.8802e-27	4.8068e-10	0.075001	0.925

xxx TABLE:pi_eta xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0.925	0.075001	4.8068e-10	0	0
r2	0.0026569	0.96788	0.029459	2.602e-11	0
r3	1.1558e-12	0.0096913	0.98062	0.0096913	1.1559e-12
r4	1.28e-29	2.602e-11	0.029459	0.96788	0.0026569
r5	2.8504e-54	1.8802e-27	4.8068e-10	0.075001	0.925

xxx TABLE:pi_kids xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c214	c215	c216
r1	0.88584	0.11137	0.0027905	1	0	0
r2	0.051343	0.66234	0.28632	1	0	0
r3	0.0015025	0.063309	0.93519	1	0	0

xxx TABLE:psi xxxxxxxxxxxxxxxxxx

c1

```
        -------

   r1   0.99935
   r2   0.99623
   r3   0.99635
   r4   0.99537
   r5   0.99299
   r6   0.98956
   r7   0.98547
   r8   0.98022
   r9   0.96914
   r10  0.95071
   r11  0.92082
   r12  0.87772
   r13  0.81394
   r14  0.70638
   r15  0.54032
   r16  0.34767
   r17  0.18848
   r18        0

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_exotrans Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    value
                -    ---    -----

   pi_S_eta     1     3       1

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_typelife ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
               i    idx    ndim    numel    rowN    colN     sum        mean        std      coefvar
               -    ---    ----    -----    ----    ----    ------    --------    -------    -------

   SS          1     1      2       36       18      2      3.2218    0.089493    0.12913      1.443
   epsilon     2     2      2       36       18      2      39.526      1.0979    0.85451    0.77828

xxx TABLE:SS xxxxxxxxxxxxxxxxxx
             c1         c2
          -------    -------

   r1          0          0
   r2          0          0
   r3          0          0
   r4          0          0
   r5          0          0
   r6          0          0
   r7          0          0
   r8          0          0
   r9          0          0
   r10         0          0
   r11         0          0
   r12         0          0
   r13   0.24433    0.29263
   r14   0.24433    0.29263
```

```
    r15    0.24433    0.29263
    r16    0.24433    0.29263
    r17    0.24433    0.29263
    r18    0.24433    0.29263

xxx TABLE:epsilon xxxxxxxxxxxxxxxxxx
             c1        c2
           ------    ------

    r1          1         1
    r2     1.0778    1.1836
    r3     1.2546    1.6124
    r4      1.397    1.9418
    r5     1.5022    2.1452
    r6     1.5712    2.2394
    r7     1.6064    2.2588
    r8     1.6097    2.2341
    r9     1.5815     2.182
    r10    1.5204    2.1034
    r11    1.4243    1.9846
    r12    1.2917    1.8041
    r13         0         0
    r14         0         0
    r15         0         0
    r16         0         0
    r17         0         0
    r18         0         0

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_stat ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                          i    idx    ndim    numel    rowN    colN     sum       mean        std
                          -    ---    ----    -----    ----    ----    ------    -------    --------

    Pop                   1     1      2       18       18      1      9.8945    0.54969     0.31889
    stat_distr_educ       2     3      2        2        1      2           1        0.5      0.2786
    stat_distr_eta        3     4      2        5        1      5           1        0.2     0.24003
    stat_distr_kids       4     5      3       12        2      6           4    0.33333     0.33166
    stat_distr_married    5     6      2        4        2      2           2        0.5    0.073381

xxx TABLE:Pop xxxxxxxxxxxxxxxxxx
              c1
           --------

    r1            1
    r2      0.95085
    r3      0.90129
    r4      0.85442
    r5      0.80919
    r6      0.76452
    r7      0.71982
    r8      0.67493
    r9      0.62947
    r10     0.58044
    r11     0.52505
    r12     0.46001
    r13     0.38416
```

```
    r14     0.29751
    r15     0.19995
    r16      0.1028
    r17    0.034004
    r18    0.006098

xxx TABLE:stat_distr_educ xxxxxxxxxxxxxxxxxx
           c1       c2
          -----    -----

    r1    0.697    0.303

xxx TABLE:stat_distr_eta xxxxxxxxxxxxxxxxxx
             c1           c2         c3         c4           c5
          ---------    -------    -------    -------    ---------

    r1    0.0069316    0.19567    0.59479    0.19567    0.0069316

xxx TABLE:stat_distr_kids xxxxxxxxxxxxxxxxxx
            c1         c2          c3          c4           c5           c6
          -------    -------    --------    -------    ----------    --------

    r1    0.75801    0.44877      0.1564    0.32041       0.08559     0.23083
    r2    0.97627     0.7604    0.023626     0.2173    0.00010011    0.022305

xxx TABLE:stat_distr_married xxxxxxxxxxxxxxxxxx
            c1        c2
          ------    ------

    r1    0.5635    0.4365
    r2    0.4364    0.5636

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_params_stat String
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                           i      idx                          string
                          ---     ---    ---------------------------------------------------------

    st_old_age_depend     "1"     "2"    "Old-age dependency ratio (ratio of 65+/(18-64))=0.1155"
```

1.1.3 Parameters Used for Paper Simulations

Using 266 household head income shocks. Requires 150GB of memory.

```
% mp_params = snw_mp_param('default_moredense_a65zh266zs5_e2m2', true, 100, 6);
```

1.2 Model Controls

This is the example vignette for function: **snw_mp_controls** from the **PrjOptiSNW Package.** This function sets and gets different parameters

1.2.1 Test SNW_MP_CONTROLS Defaults

Call the function with defaults.

```
mp_controls = snw_mp_control('default_base', true);
```

```
pos = 21 ; key = options
  fmincon options:

   Options used by current Algorithm ('interior-point'):
   (Other available algorithms: 'active-set', 'sqp', 'sqp-legacy', 'trust-region-reflective')

   Set properties:
                     Display: 'off'

   Default properties:
                   Algorithm: 'interior-point'
              CheckGradients: 0
         ConstraintTolerance: 1.0000e-06
     FiniteDifferenceStepSize: 'sqrt(eps)'
        FiniteDifferenceType: 'forward'
        HessianApproximation: 'bfgs'
                  HessianFcn: []
          HessianMultiplyFcn: []
                 HonorBounds: 1
      MaxFunctionEvaluations: 3000
               MaxIterations: 1000
              ObjectiveLimit: -1.0000e+20
         OptimalityTolerance: 1.0000e-06
                   OutputFcn: []
                     PlotFcn: []
                ScaleProblem: 0
    SpecifyConstraintGradient: 0
     SpecifyObjectiveGradient: 0
               StepTolerance: 1.0000e-10
         SubproblemAlgorithm: 'factorization'
                    TypicalX: 'ones(numberOfVariables,1)'
                 UseParallel: 0

   Show options not used by current Algorithm ('interior-point')

pos = 22 ; key = options2
  fsolve options:

   Options used by current Algorithm ('trust-region-dogleg'):
   (Other available algorithms: 'levenberg-marquardt', 'trust-region')

   Set properties:
                    Display: 'off'

   Default properties:
                  Algorithm: 'trust-region-dogleg'
             CheckGradients: 0
    FiniteDifferenceStepSize: 'sqrt(eps)'
       FiniteDifferenceType: 'forward'
          FunctionTolerance: 1.0000e-06
     MaxFunctionEvaluations: '100*numberOfVariables'
              MaxIterations: 400
        OptimalityTolerance: 1.0000e-06
                  OutputFcn: []
                    PlotFcn: []
    SpecifyObjectiveGradient: 0
              StepTolerance: 1.0000e-06
                   TypicalX: 'ones(numberOfVariables,1)'
```

```
                 UseParallel: 0

   Show options not used by current Algorithm ('trust-region-dogleg')

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_controls Scalars
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                  i     idx    value
                                  --    ---    -----

    A_aux                          1     1       NaN
    Aeq                            2     2       NaN
    B_aux                          3     3       NaN
    Beq                            4     4       NaN
    bl_compute_drv_stats           5     5         1
    bl_ds_store_all                6     6         0
    bl_print_a4chk                 7     7         1
    bl_print_a4chk_verbose         8     8         0
    bl_print_ds                    9     9         1
    bl_print_ds_verbose           10    10         0
    bl_print_vfi                  11    11         1
    bl_print_vfi_verbose          12    12         0
    bl_print_vu_vw                13    13         1
    bl_print_vu_vw_verbose        14    14         0
    bl_timer                      15    15         1
    bl_vfi_store_all              16    16         0
    err                           17    17         1
    fl_max_trchk_perc_increase    18    18       1.5
    nonlcon                       19    20       NaN
    tol                           20    23     0.005

----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_controls String
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                       i      idx        string
                      ---    ----    --------------

    mp_params_name    "1"    "19"    "default_base"
```

Chapter 2

Solving the Dynamic Life Cycle Problem

2.1 Life Cycle Dynamic Programming with Marital Status, Children and Savings

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for policy function with vectorized bisection. More Dense Solution Analysis.

2.1.1 Test SNW_VFI_MAIN_BISECT_VEC Defaults More Dense

Call the function with defaults.

```
mp_param = snw_mp_param('default_docdense');
[V_VFI,ap_VFI,cons_VFI] = snw_vfi_main_bisec_vec(mp_param);
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:83 of 82, time-this-age:1.3477
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:82 of 82, time-this-age:2.772
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:81 of 82, time-this-age:2.6886
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:80 of 82, time-this-age:2.6034
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:79 of 82, time-this-age:2.4025
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:78 of 82, time-this-age:2.5892
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:77 of 82, time-this-age:2.4135
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:76 of 82, time-this-age:2.5798
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:75 of 82, time-this-age:2.4211
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:74 of 82, time-this-age:2.5768
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:73 of 82, time-this-age:2.5754
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:72 of 82, time-this-age:2.6072
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:71 of 82, time-this-age:2.5104
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:70 of 82, time-this-age:2.5658
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:69 of 82, time-this-age:2.4148
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:68 of 82, time-this-age:2.5182
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:67 of 82, time-this-age:2.3762
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:66 of 82, time-this-age:2.5053
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:65 of 82, time-this-age:2.4224
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:64 of 82, time-this-age:2.5013
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:63 of 82, time-this-age:2.4094
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:62 of 82, time-this-age:2.4546
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:61 of 82, time-this-age:2.3964
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:60 of 82, time-this-age:2.5196
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:59 of 82, time-this-age:2.3948
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:58 of 82, time-this-age:2.5256
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:57 of 82, time-this-age:2.3984
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:56 of 82, time-this-age:2.5195
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:55 of 82, time-this-age:2.4202
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:54 of 82, time-this-age:2.5087
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:53 of 82, time-this-age:2.3967
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:52 of 82, time-this-age:2.5908
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:51 of 82, time-this-age:2.4273
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:50 of 82, time-this-age:2.4934
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:49 of 82, time-this-age:2.4106
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:48 of 82, time-this-age:2.4987
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:47 of 82, time-this-age:2.5508
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:46 of 82, time-this-age:2.6959
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:45 of 82, time-this-age:2.5636
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:44 of 82, time-this-age:2.6548
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:43 of 82, time-this-age:2.5629
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:42 of 82, time-this-age:2.6513
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:41 of 82, time-this-age:2.542
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:40 of 82, time-this-age:2.6668
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:39 of 82, time-this-age:2.543
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:38 of 82, time-this-age:2.6489
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:37 of 82, time-this-age:2.5832
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:36 of 82, time-this-age:2.6481
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:35 of 82, time-this-age:2.6059
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:34 of 82, time-this-age:2.7
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:33 of 82, time-this-age:2.5998
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:32 of 82, time-this-age:2.6749
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:31 of 82, time-this-age:2.5805
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:30 of 82, time-this-age:2.649
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:29 of 82, time-this-age:2.5965
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:28 of 82, time-this-age:2.6474
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:27 of 82, time-this-age:2.593
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:26 of 82, time-this-age:2.6669
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:25 of 82, time-this-age:2.6025
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:24 of 82, time-this-age:2.6214
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:23 of 82, time-this-age:2.5912
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:22 of 82, time-this-age:2.7036
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:21 of 82, time-this-age:2.5499
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:20 of 82, time-this-age:2.6382
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:19 of 82, time-this-age:2.6067
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:18 of 82, time-this-age:2.6643
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:17 of 82, time-this-age:2.5918
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:16 of 82, time-this-age:2.6505
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:15 of 82, time-this-age:2.6052
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:14 of 82, time-this-age:2.6762
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:13 of 82, time-this-age:2.6089
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:12 of 82, time-this-age:2.6791
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:11 of 82, time-this-age:2.6401
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:10 of 82, time-this-age:2.8111
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:9 of 82, time-this-age:2.6367
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:8 of 82, time-this-age:2.6476
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:7 of 82, time-this-age:2.6136
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:6 of 82, time-this-age:2.69
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:5 of 82, time-this-age:2.6314
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:4 of 82, time-this-age:2.6429
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:3 of 82, time-this-age:2.6004
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:2 of 82, time-this-age:2.7677
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:1 of 82, time-this-age:2.625
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_base;time=213.
```

2.1.2 More 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_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('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});
```

2.1.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% 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(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

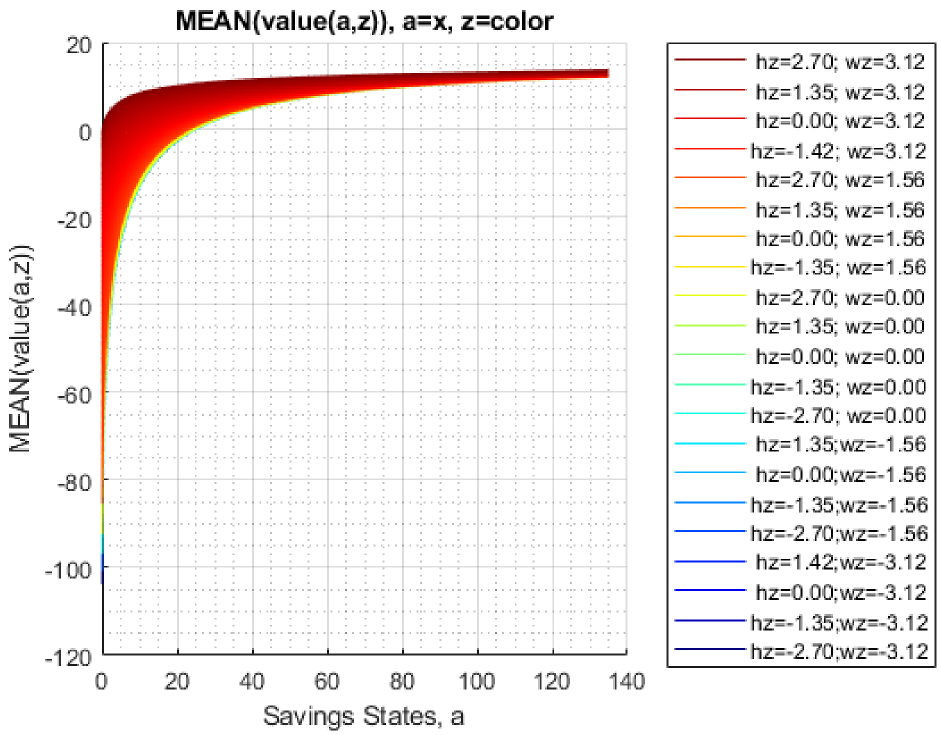
Tabulate value and policies along savings and shocks:

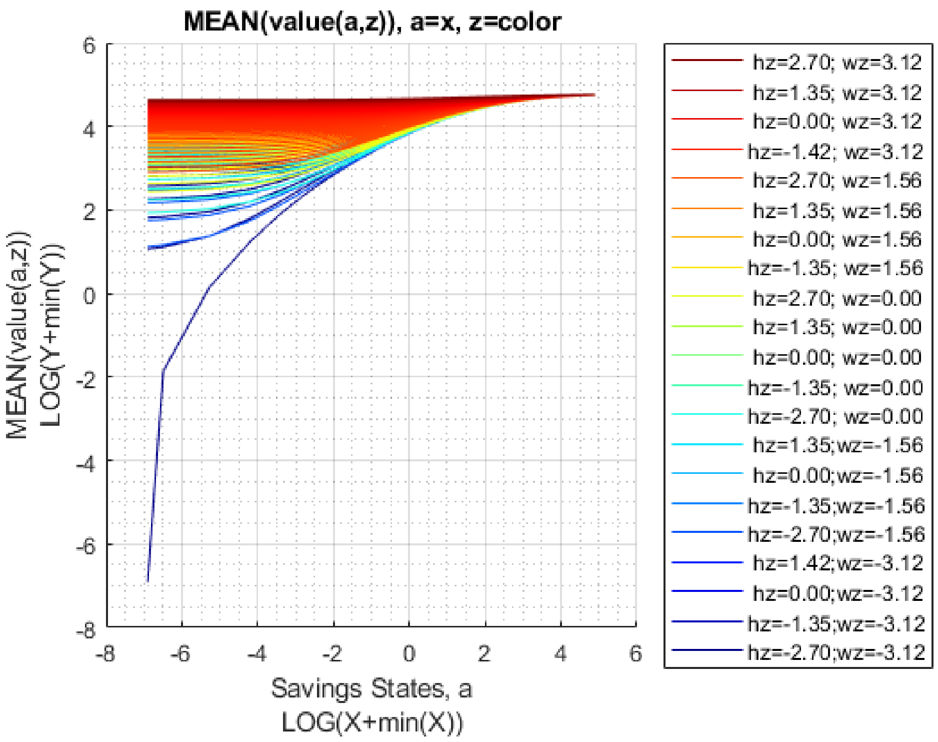
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm

xxx  MEAN(VAL(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1               0     -103.74        -100.83       -97.586         -94.14       -90.628       -
```

```
    -----     ----------     ----------     ----------     ----------     ----------     ----------     ---

      1                0              0              0              0              0              0
```

```
xxx  MEAN(C(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group      savings       mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5     mea
    -----     ----------     ----------     ----------     ----------     ----------     ----------     ---

      1                0       0.14271        0.14506        0.14755        0.15021        0.15304       0.
```

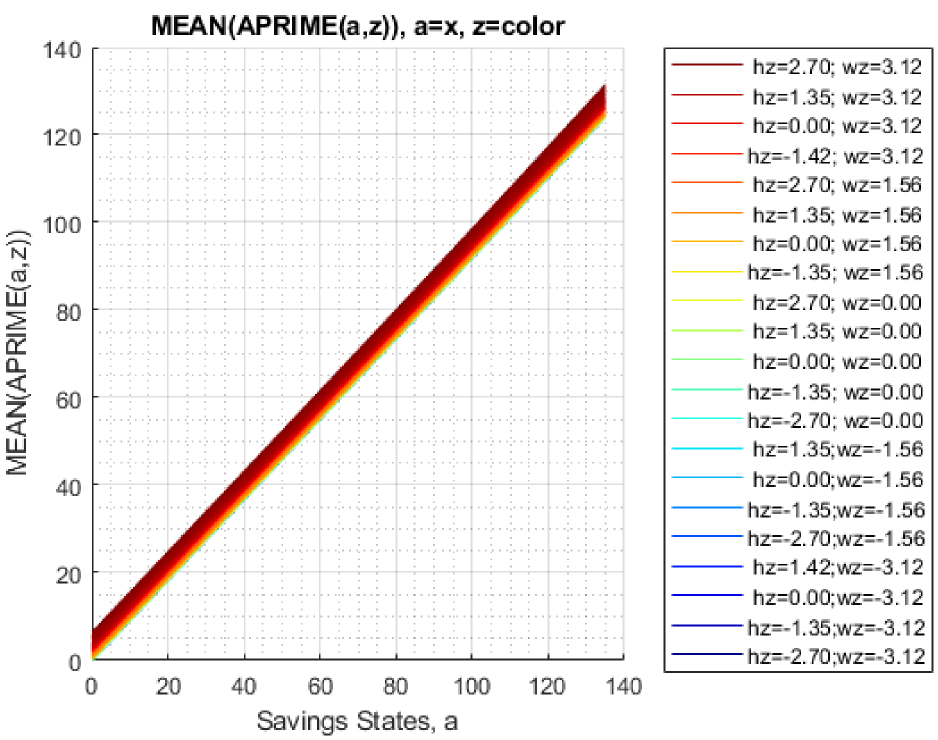
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

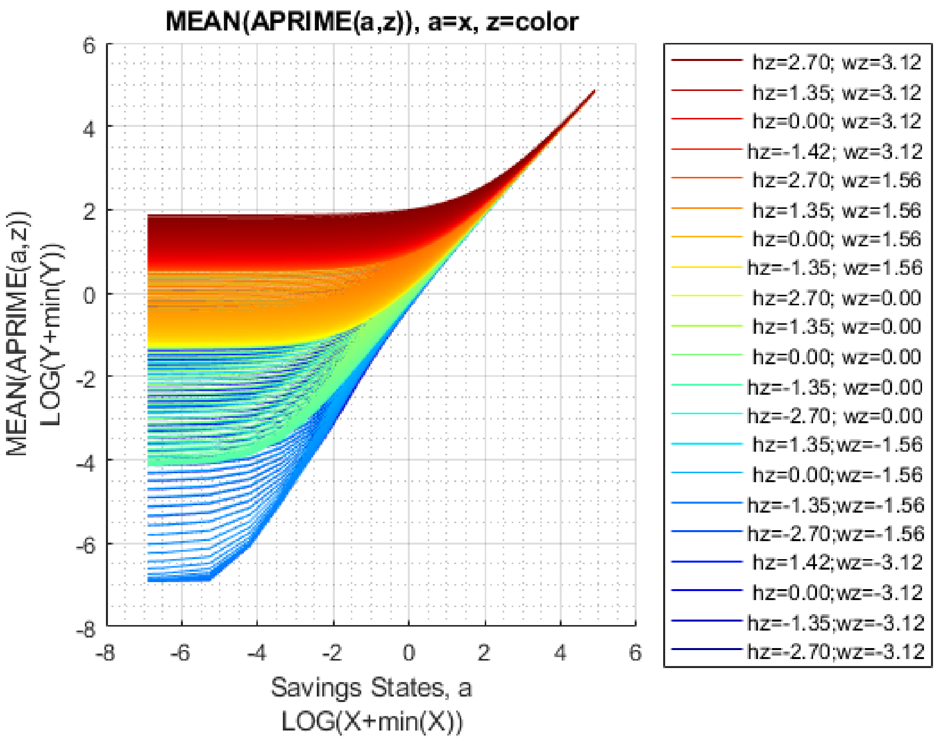




Graph Mean Savings Choices:

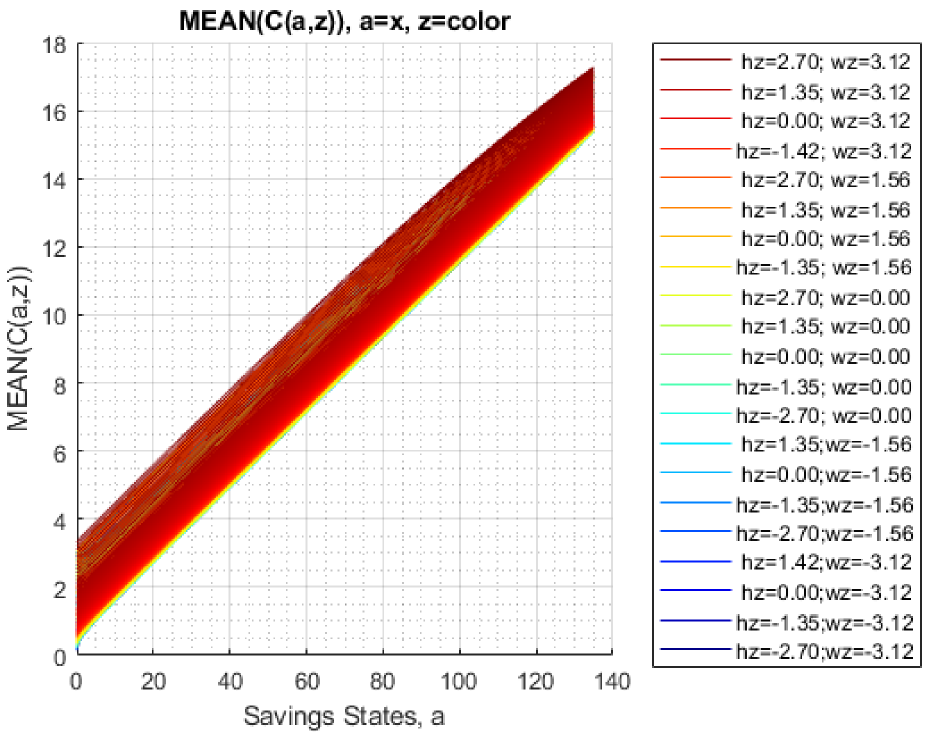
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(a,z))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

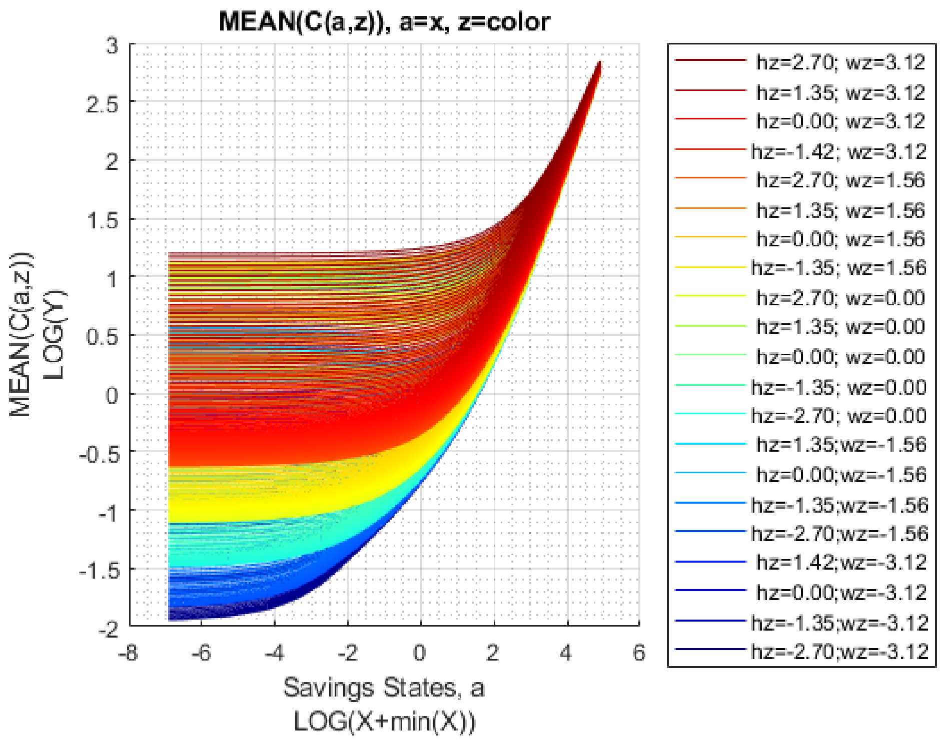




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





2.1.4 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
tb_az_v = ff_summ_nd_array("MEAN(VAL(KM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(VAL(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        -9.6123         -8.574        -7.5952        -6.6749        -5.8609
      2        2       0        -17.183        -15.851        -14.558        -13.309        -12.171
```

```
       3        3         0        -20.909         -19.563         -18.242         -16.949         -15.768
       4        4         0        -24.758         -23.406          -22.06         -20.727           -19.5
       5        5         0        -27.561         -26.288         -25.009          -23.73         -22.552
       6        1         1         2.1559          3.0013          3.7773          4.4944          5.1268
       7        2         1        -2.4375         -1.4691        -0.55596         0.31118          1.0968
       8        3         1        -4.6483          -3.672         -2.7454         -1.8583         -1.0517
       9        4         1        -7.2434         -6.2806         -5.3574         -4.4633         -3.6454
      10        5         1        -9.2948         -8.3935         -7.5263         -6.6822         -5.9134
```

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(KM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

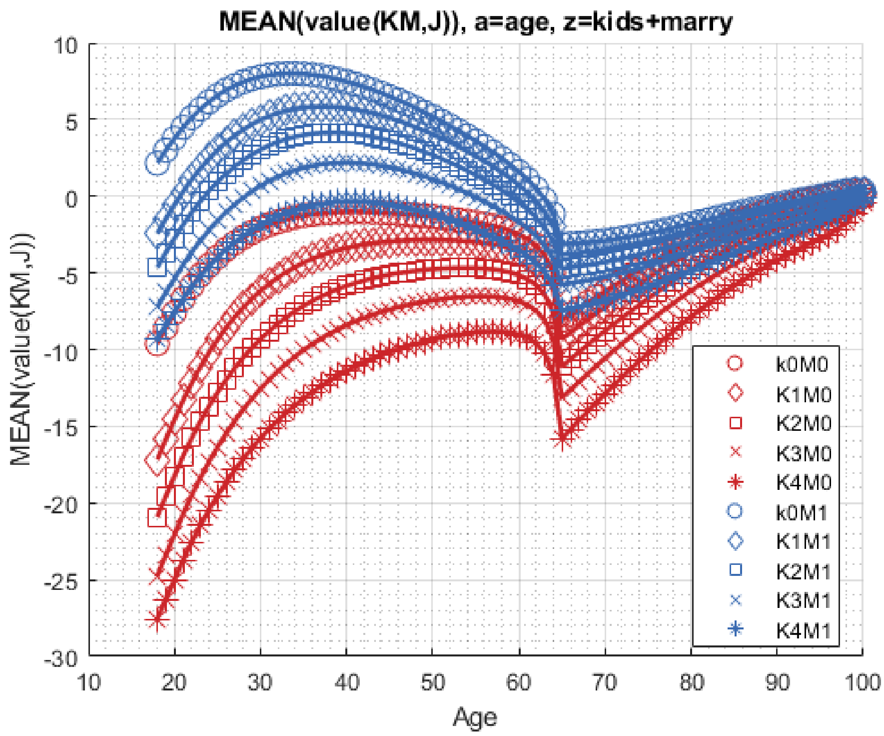
```
xxx  MEAN(AP(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1       1        0        34.494         34.456         34.416         34.452         34.489
      2       2        0          34.3         34.256          34.21         34.238         34.268
      3       3        0        34.146         34.101         34.055         34.082          34.11
      4       4        0        34.053          34.01         33.964         33.991          34.02
      5       5        0         33.97         33.929         33.885         33.915         33.946
      6       1        1        35.208         35.246         35.285         35.413         35.545
      7       2        1        34.951         34.976             35          35.11         35.222
      8       3        1        34.708         34.724         34.739         34.838         34.939
      9       4        1        34.506         34.516         34.523         34.613         34.704
     10       5        1        34.221         34.218         34.212         34.286         34.363
```

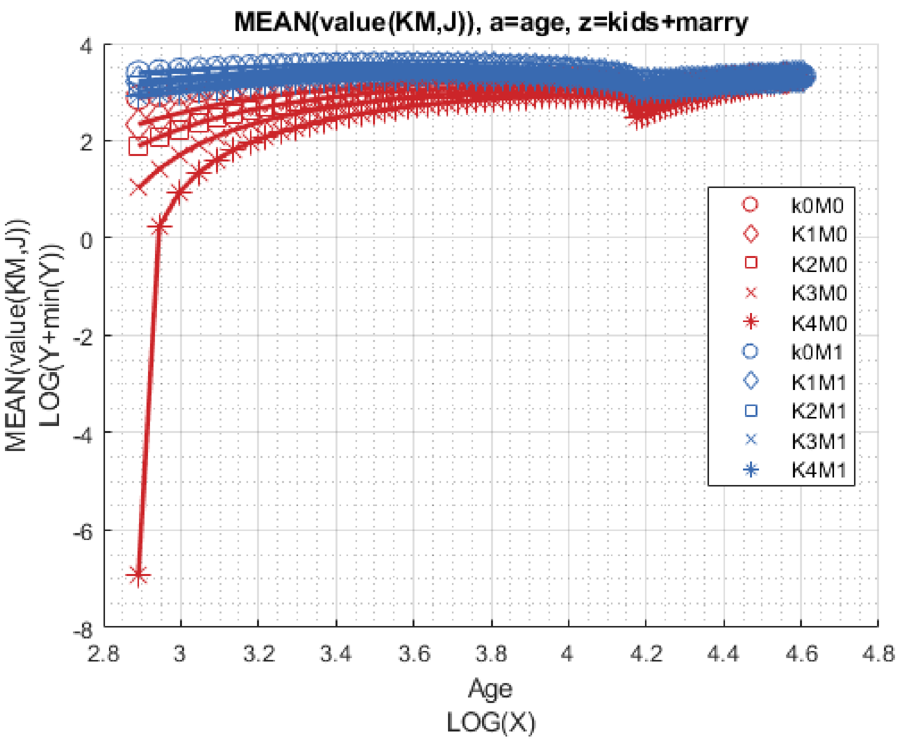
% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(C(KM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permute);

```
xxx  MEAN(C(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1       1        0        2.0632          2.102         2.1418          2.184         2.2244
      2       2        0        2.2579         2.3019          2.348         2.3975         2.4457
      3       3        0        2.4119         2.4563          2.503         2.5537         2.6031
      4       4        0        2.5046         2.5481          2.594         2.6445         2.6938
      5       5        0        2.5877         2.6287         2.6724         2.7207         2.7678
      6       1        1        2.6183         2.6787         2.7402         2.8051         2.8674
      7       2        1         2.681         2.7395         2.8002         2.8656         2.9293
      8       3        1        2.7896         2.8462         2.9054         2.9698         3.0325
      9       4        1        2.8528         2.9056         2.9612         3.0222         3.0816
     10       5        1        2.9174          2.966         3.0172         3.0737         3.1281
```

Graph Mean Values:

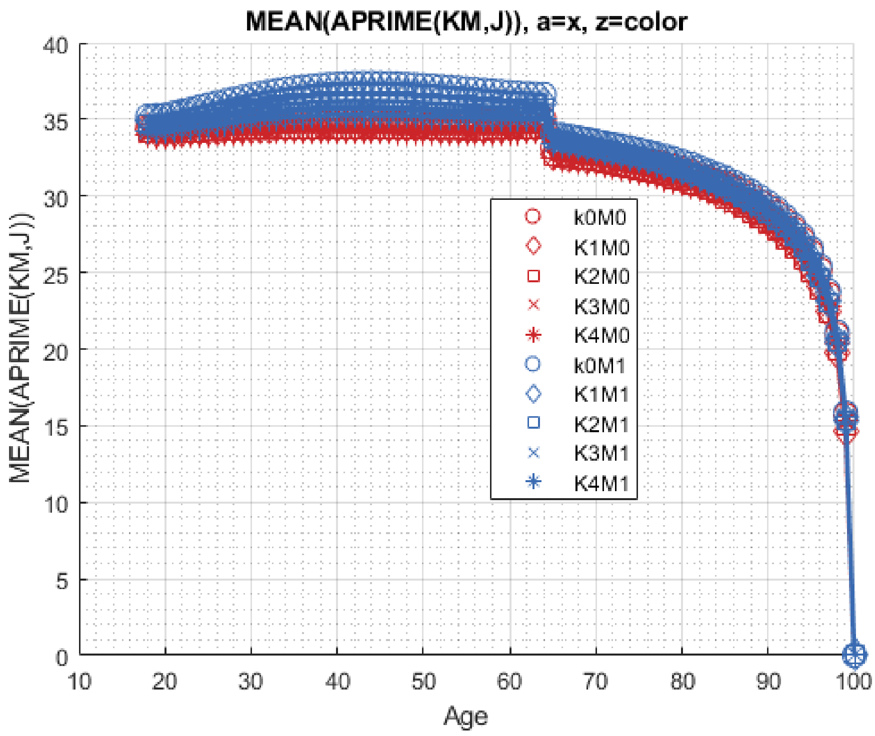
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(KM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

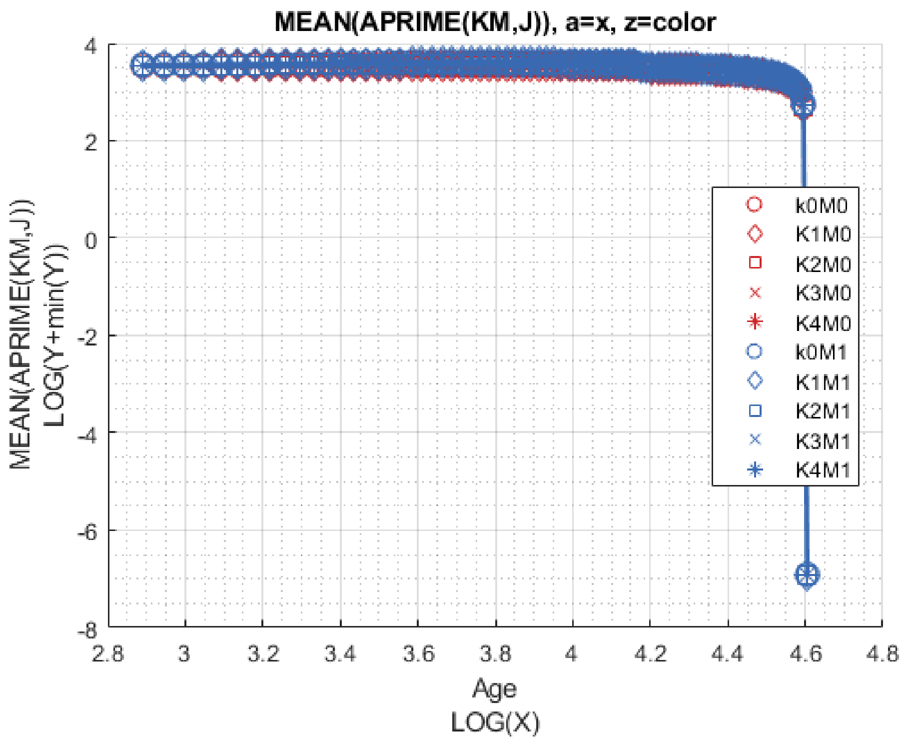




Graph Mean Savings Choices:

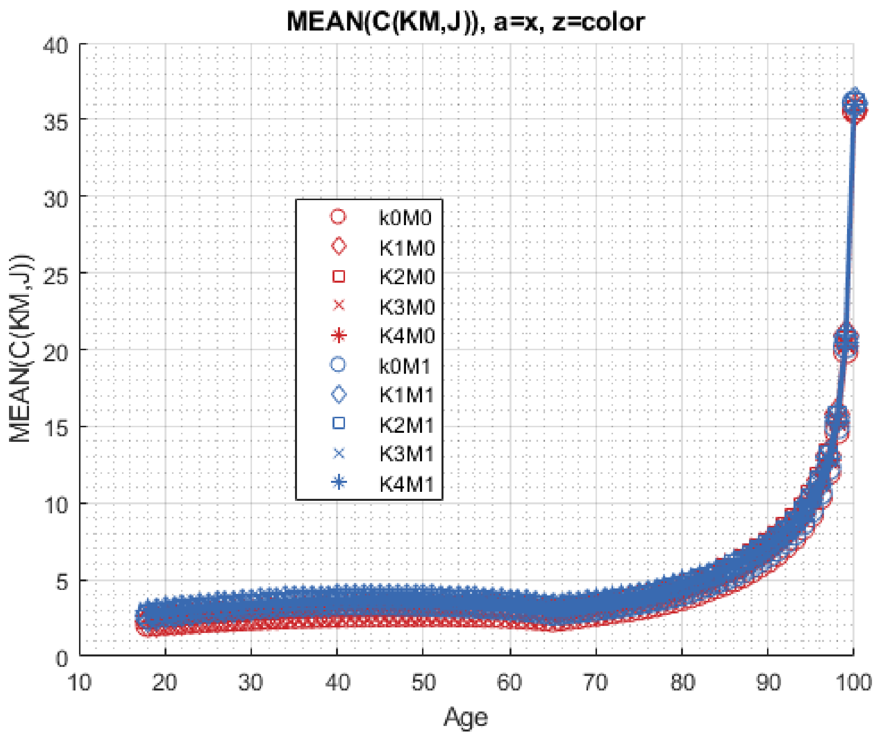
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(KM,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

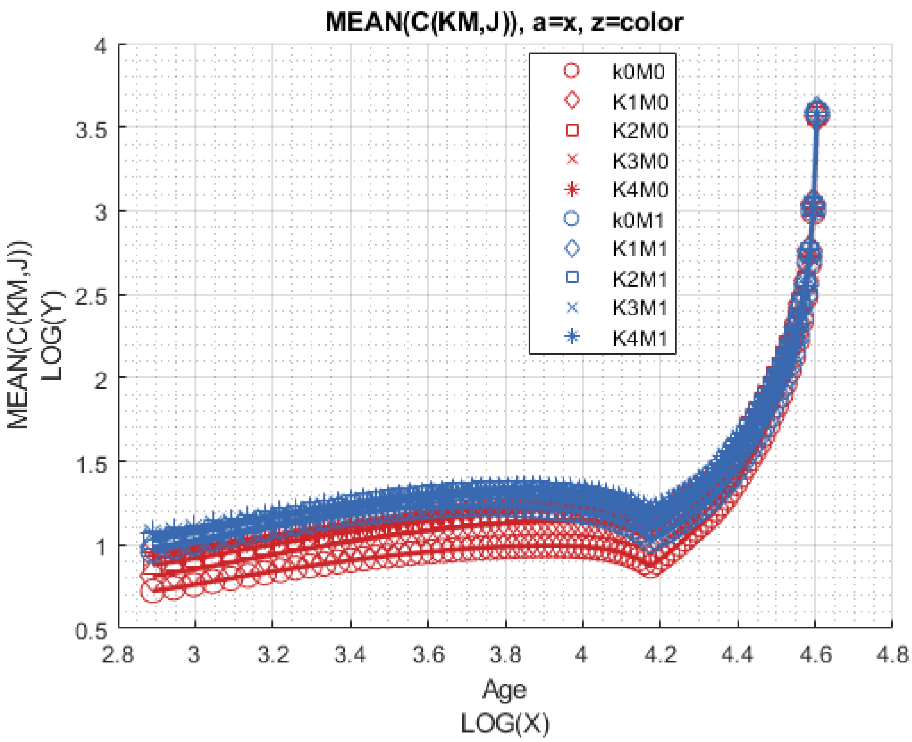




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





2.1.5 Analyze Education 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 = ["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'};
```

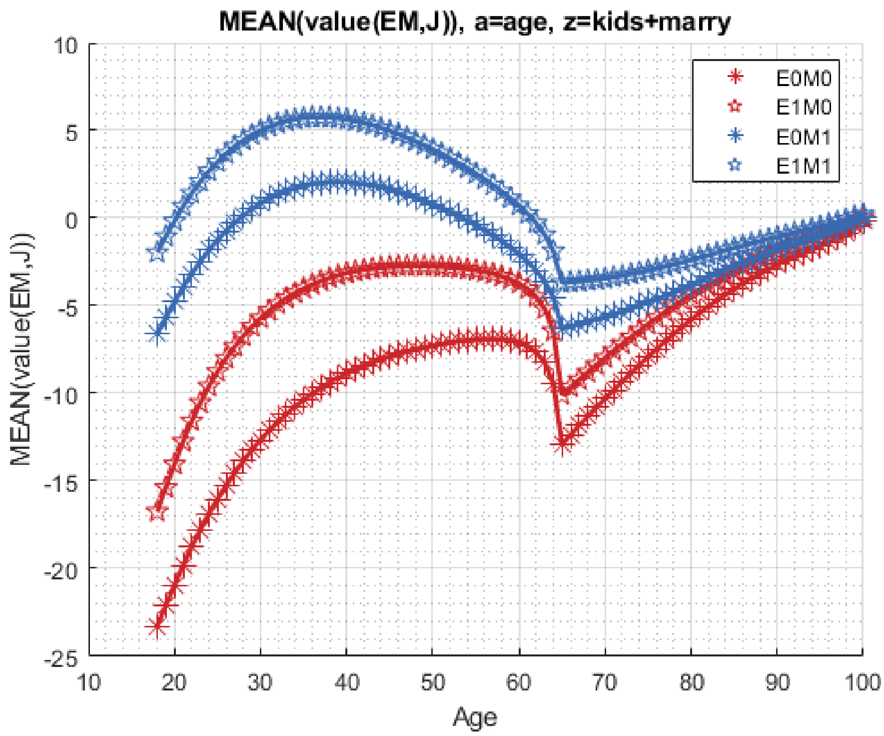
MEAN(VAL(EKM,J)), MEAN(AP(EKM,J)), MEAN(C(EKM,J))

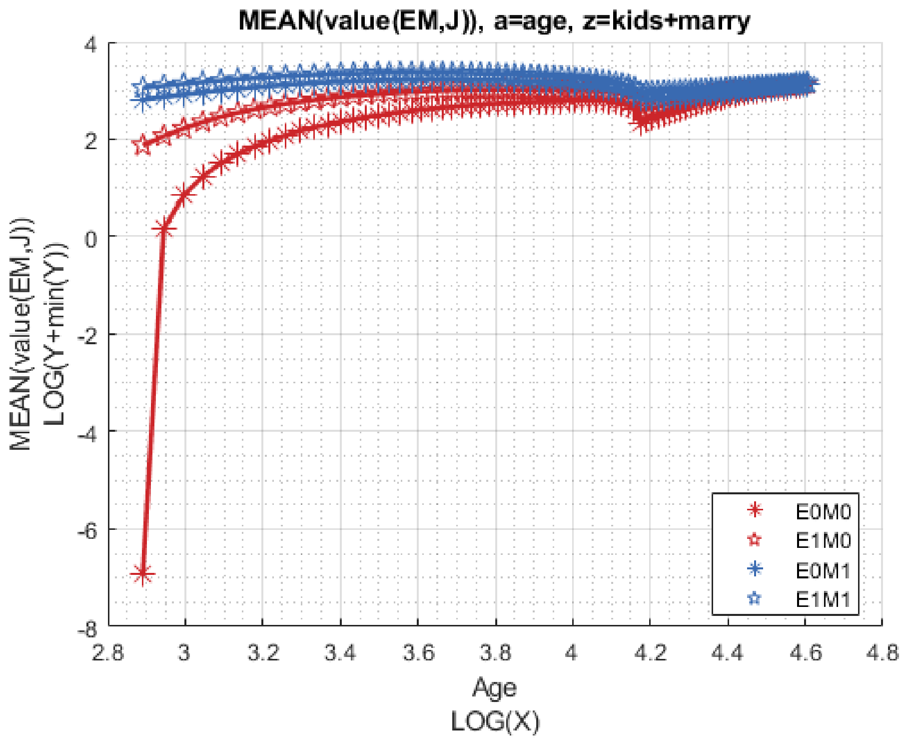
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(EKM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(VAL(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0         -23.27        -22.094        -20.941        -19.811        -18.761
      2       1       0        -16.739        -15.379        -14.045        -12.745         -11.58
      3       0       1        -6.6189        -5.6779        -4.7885        -3.9435        -3.1707
      4       1       1        -1.9684        -1.0477       -0.17465        0.66417         1.4159

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(EKM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(AP(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        34.294         34.261         34.226         34.237         34.247
      2       1       0        34.091          34.04         33.986         34.035         34.087
      3       0       1        34.769         34.789         34.809          34.88         34.951
      4       1       1        34.669         34.683         34.695         34.824         34.958

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(EKM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(C(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        2.2635         2.2969         2.3317         2.3683         2.4043
      2       1       0        2.4666         2.5178          2.572         2.6319         2.6896
      3       0       1        2.6261         2.6712         2.7175         2.7661         2.8135
      4       1       1        2.9175         2.9832         3.0522         3.1285          3.202
```

Graph Mean Values:

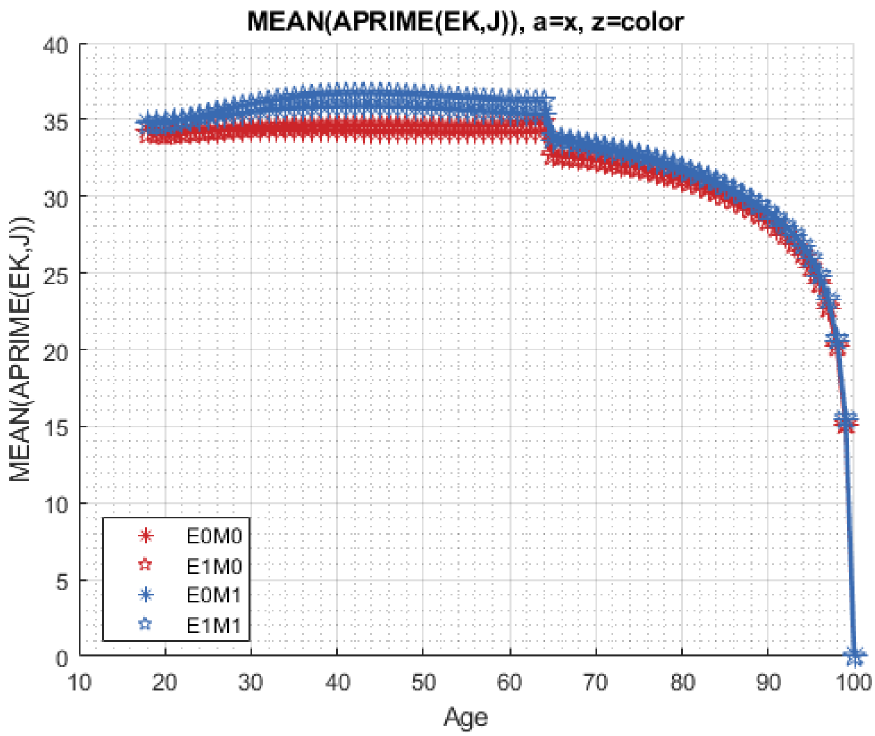
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(EM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

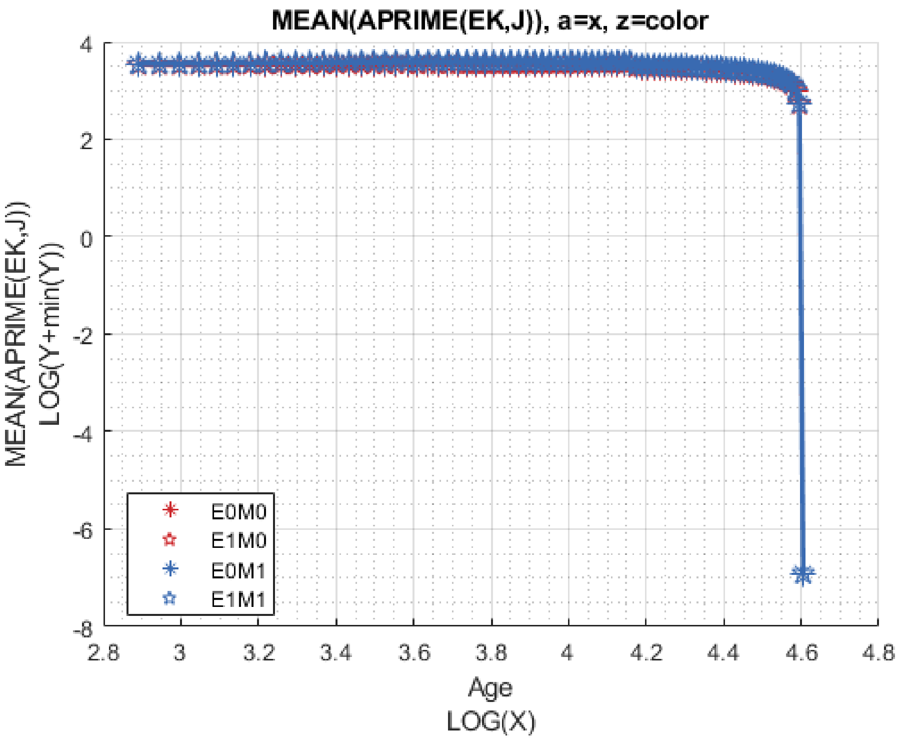




Graph Mean Savings Choices:

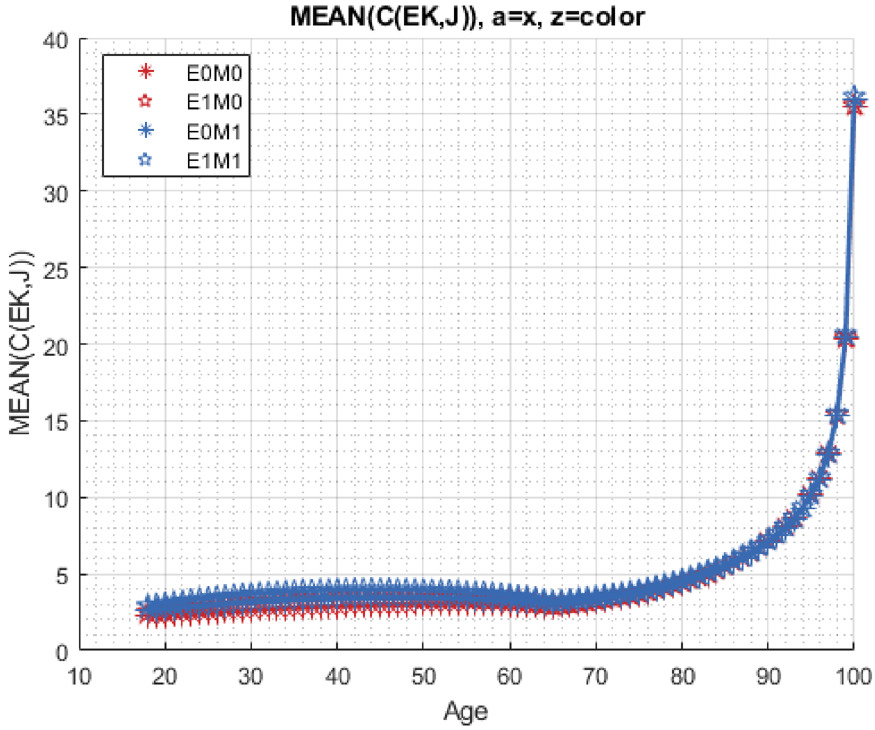
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(EK,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

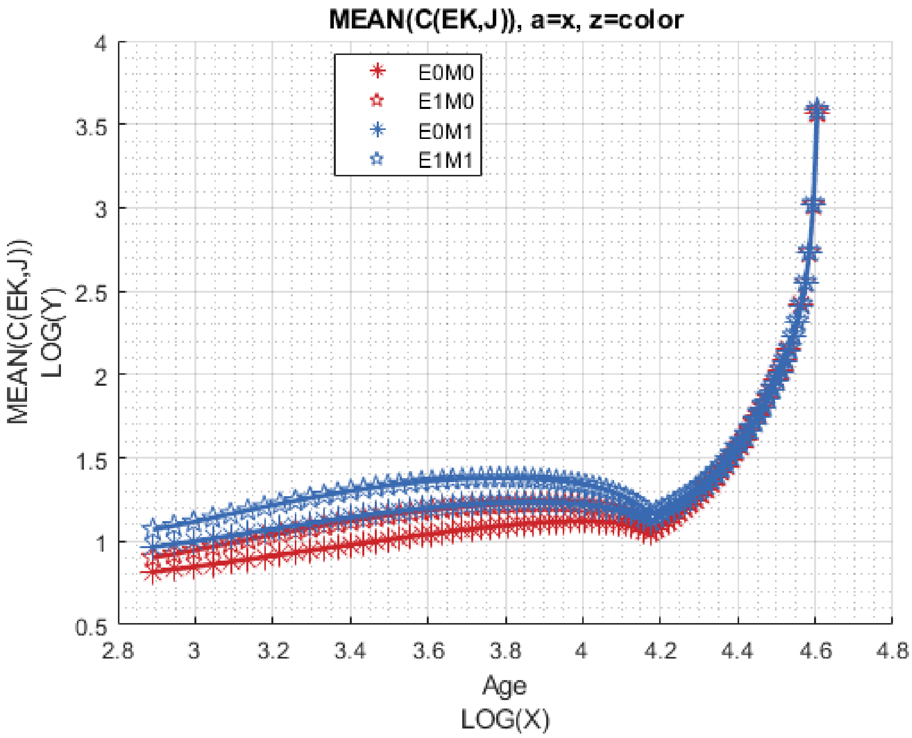




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(EK,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```


MEAN(C(EK,J)), a=x, z=color
E0M0
E1M0
E0M1
E1M1
MEAN(C(EK,J))
Age


MEAN(C(EK,J)), a=x, z=color
E0M0
E1M0
E0M1
E1M1
MEAN(C(EK,J))
LOG(Y)
Age
LOG(X)

Chapter 3

Alternative Value Function Solution Testing

3.1 SNW_VFI_PARAM Small Solution Analysis

This is the example vignette for function: **snw_vfi_main** from the **PrjOptiSNW Package.** This function solves for policy function fully iteratively using matlab minimizer. Small Solution Analysis.

3.1.1 Test SNW_VFI_MAIN Defaults Small

Call the function with defaults.

```
mp_param = snw_mp_param('default_small');
[V_VFI,ap_VFI,cons_VFI,mp_valpol_more] = snw_vfi_main(mp_param);
```

```
SNW_VFI_MAIN: Finished Age Group:18 of 18
```

3.1.2 Small 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 = [19, 22:5:97, 100];
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('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});
```

3.1.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% 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') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(A,Z))", ap_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per
```

% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(C(A,Z))", cons_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_permute);

Graph Mean Values:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

Graph Mean Savings Choices:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(a,z))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

3.1.4 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", "k0M1", "K1M1", "K2M1"];
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', 'o', 'd', 's'};
mp_support_graph('cl_colors') = {'red', 'red', 'red', '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
tb_az_v = ff_summ_nd_array("MEAN(VAL(KM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(KM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(C(KM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permute);

Graph Mean Values:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(KM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

Graph Mean Savings Choices:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(KM,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

3.1.5 Analyze Education 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 = ["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(EKM,J)), MEAN(AP(EKM,J)), MEAN(C(EKM,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
tb_az_v = ff_summ_nd_array("MEAN(VAL(EKM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(EKM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(EKM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p
```

Graph Mean Values:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(EM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

Graph Mean Savings Choices:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(EK,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(EK,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

3.2 SNW_VFI_MAIN_GRID_SEARCH Small Solution Analysis

This is the example vignette for function: **snw_vfi_main_grid_search** from the **PrjOptiSNW Package.** This function solves for policy function using grid search. Small Solution Analysis. Small Solution Analysis, husband 5 shocks, wife 1 shocks.

3.2.1 Test SNW_VFI_MAIN_GRID_SEARCH Defaults Small

Call the function with defaults.

```
mp_param = snw_mp_param('default_small');
[V_VFI,ap_VFI,cons_VFI,mp_valpol_more] = snw_vfi_main_grid_search(mp_param);
```

```
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:18 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:17 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:16 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:15 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:14 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:13 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:12 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:11 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:10 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:9 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:8 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:7 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:6 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:5 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:4 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:3 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:2 of 18
SNW_VFI_MAIN_GRID_SEARCH: Finished Age Group:1 of 18
Elapsed time is 5.839670 seconds.
Completed SNW_VFI_MAIN_GRID_SEARCH;SNW_MP_PARAM=default_small;SNW_MP_CONTROL=default_base
```

3.2.2 Small 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 = [19, 22:5:97, 100];
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('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'}, {'Hshock', 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});
```

3.2.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% 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') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm
```

```
xxx  MEAN(VAL(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings     mean_Hshock__0_91976    mean_Hshock__0_45988    mean_Hshock_0    mean_Hsho
    _____    _________    ____________________    ____________________    _____________    _________

      1              0          -7.2621                 -5.3086               -3.7492            -2.
      2      0.0097656          -7.1445                 -5.2223               -3.6785            -2.
      3       0.078125           -6.489                 -4.7412               -3.2884            -2.
      4        0.26367          -5.3573                 -3.8789               -2.6221            -1.
      5          0.625          -4.0454                 -2.8494               -1.8168            -0.8
      6         1.2207          -2.7343                 -1.8181              -0.98298            -0.2
      7         2.1094          -1.5234                -0.86783              -0.22453             0.3
      8         3.3496         -0.46769               -0.030108                0.4355             0.8
      9              5          0.39914                 0.68023               0.99893              1.
     10         7.1191           1.0817                  1.2609                1.4733              1.
     11         9.7656           1.6112                  1.7245                1.8649
     12         12.998           2.0172                  2.0904                 2.183              2.
     13         16.875           2.3301                  2.3771                 2.439              2.
     14         21.455           2.5712                  2.6024                2.6436              2.
     15         26.797            2.758                   2.779                2.8073              2.
     16         32.959           2.9047                  2.9189                2.9383              2.
     17             40           3.0205                  3.0304                 3.044              3.
     18         47.979           3.1125                  3.1195                3.1293              3.
     19         56.953           3.1866                  3.1917                3.1987              3.
     20         66.982           3.2468                  3.2505                3.2556              3.
     21         78.125            3.296                  3.2988                3.3026              3.
     22         90.439           3.3366                  3.3386                3.3415              3.
     23         103.98           3.3704                  3.3719                3.3741              3.
     24         118.82           3.3987                  3.3999                3.4015              3.
     25            135           3.4225                  3.4234                3.4247              3.
```

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(A,Z))", ap_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per
```

```
xxx  MEAN(AP(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings     mean_Hshock__0_91976    mean_Hshock__0_45988    mean_Hshock_0    mean_Hsho
    _____    _________    ____________________    ____________________    _____________    _________

      1              0         1.1204                  1.3194                1.7407              2.4
      2      0.0097656         1.1389                  1.3611                 1.787              2.4
      3       0.078125         1.8241                  1.9861                2.3009              2.7
      4        0.26367          2.912                  3.0509                3.2315              3.5
```

```
     5          0.625          3.9861                 4.1435            4.2269           4.4
     6         1.2207          5.0231                 5.1806            5.2407           5.3
     7         2.1094          6.0741                 6.1806            6.2037           6.2
     8         3.3496          7.0463                 7.1157            7.1528           7.1
     9              5          7.9537                 7.9954            8.0509           8.0
    10         7.1191          8.8657                 8.9028            8.9398           8.9
    11         9.7656           9.787                  9.787            9.8426            9.
    12         12.998          10.606                  10.63            10.639           10.
    13         16.875          11.481                 11.495            11.532           11.
    14         21.455          12.407                 12.407            12.421           12.
    15         26.797          13.259                 13.287            13.296           13.
    16         32.959          14.093                 14.102            14.125           14.
    17             40          14.972                 14.977            14.986           15.
    18         47.979          15.843                 15.866             15.87            15
    19         56.953           16.75                  16.75            16.773           16.
    20         66.982          17.653                 17.657            17.667           17.
    21         78.125          18.477                 18.486            18.495           18.
    22         90.439          19.315                 19.319            19.329           19.
    23         103.98          20.218                 20.222            20.227           20.
    24         118.82          21.083                 21.083            21.083           21.
    25            135          21.944                 21.949            21.954           21.
```

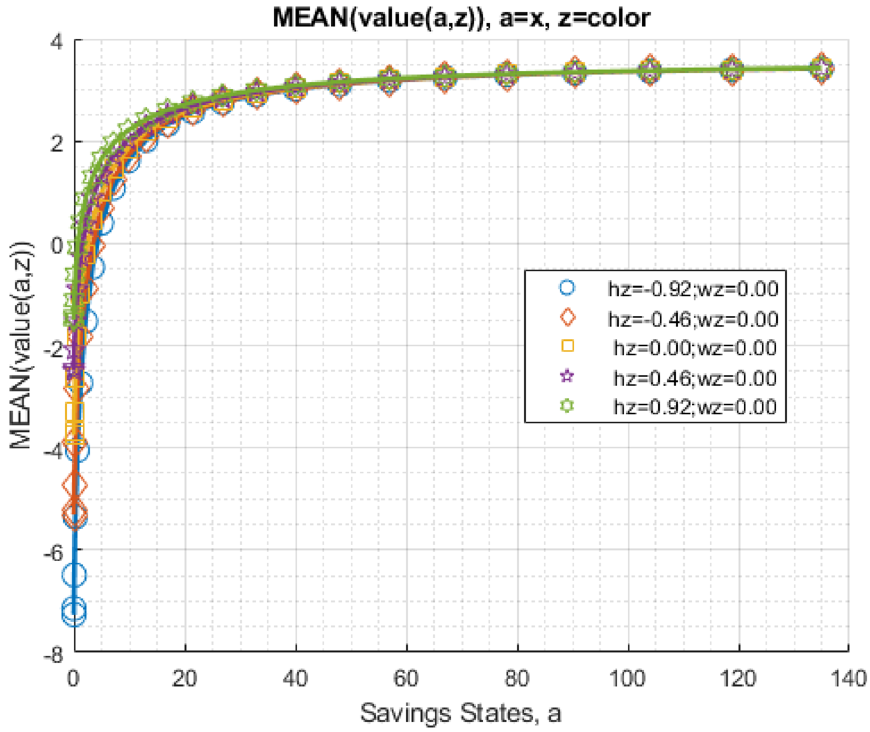
% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(C(A,Z))", cons_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_permute);

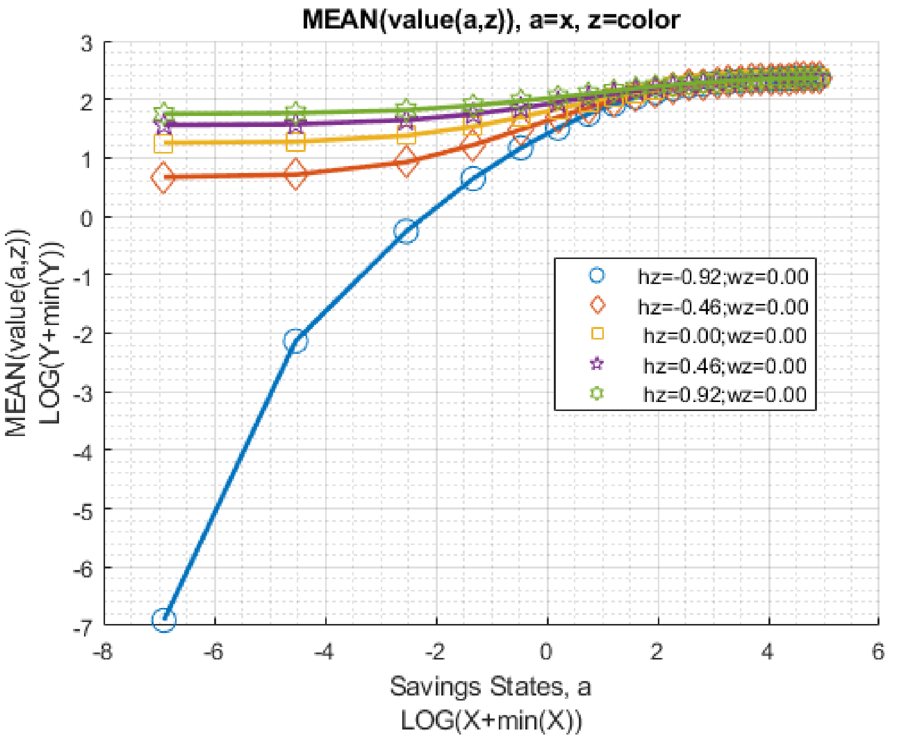
```
xxx  MEAN(C(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings     mean_Hshock__0_91976    mean_Hshock__0_45988    mean_Hshock_0    mean_Hsho
    -----    ---------    --------------------    --------------------    -------------    ---------

      1              0          0.47561                0.60477            0.78755           1.0
      2      0.0097656          0.48688                0.61444            0.79661           1.0
      3       0.078125          0.55276                0.67316            0.84544           1.0
      4        0.26367          0.68068                0.79387             0.9582           1.1
      5          0.625          0.87413                0.95556             1.1377           1.3
      6         1.2207           1.1386                 1.1823              1.359           1.6
      7         2.1094           1.4348                 1.4767             1.6678           1.9
      8         3.3496           1.8593                 1.9088             2.0737           2.3
      9              5           2.4822                 2.5483             2.6665           2.9
     10         7.1191            3.276                 3.3318             3.4625           3.6
     11         9.7656           4.2223                  4.356             4.4176           4.6
     12         12.998           5.5639                 5.6224             5.8007           5.9
     13         16.875           7.1191                 7.1983             7.2626           7.4
     14         21.455           8.7496                 8.8825             9.0265           9.1
     15         26.797            10.88                 10.865             11.023           11.
     16         32.959           13.483                 13.559             13.624           13.
     17             40           16.255                 16.355             16.497           16.
     18         47.979           19.414                 19.362             19.532           19.
     19         56.953           22.728                 22.861              22.86           23.
     20         66.982           26.496                 26.587             26.701           26.
     21         78.125           31.189                 31.223             31.332           31.
     22         90.439           36.369                 36.444             36.537           36.
     23         103.98           41.558                 41.627             41.772           41.
     24         118.82           47.433                 47.565             47.772           47.
     25            135           53.934                 53.998              54.13           54.
```

Graph Mean Values:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(a,z))'};
```

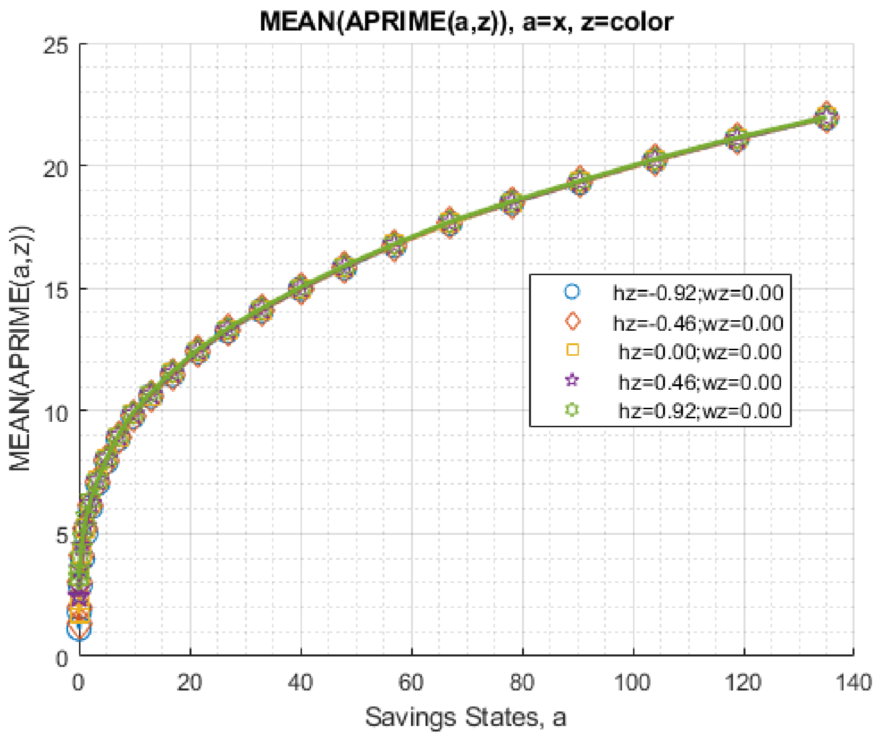
```
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

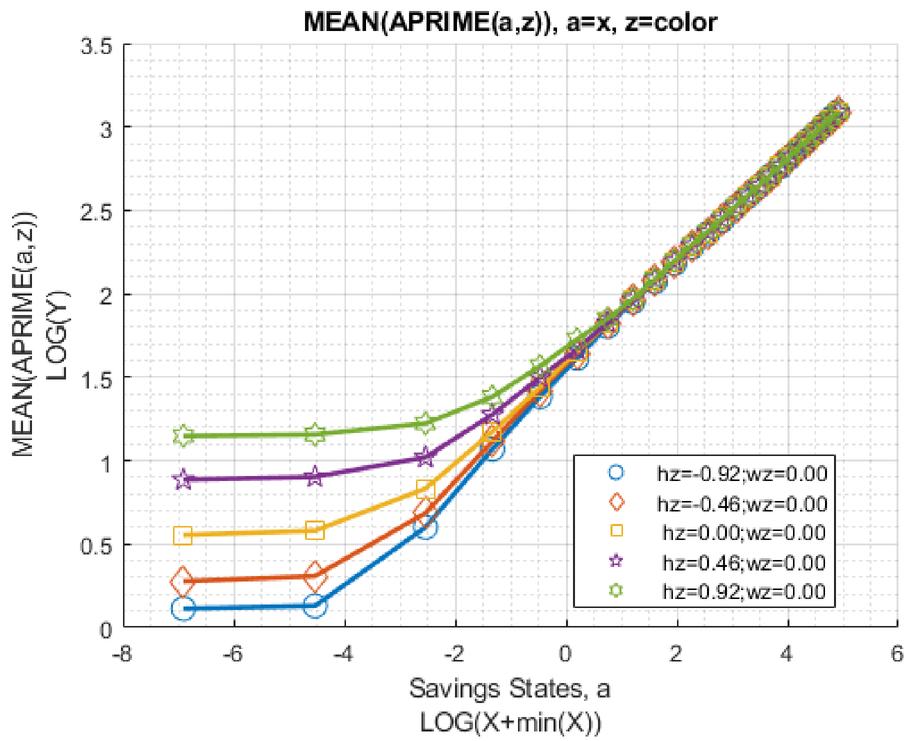




Graph Mean Savings Choices:

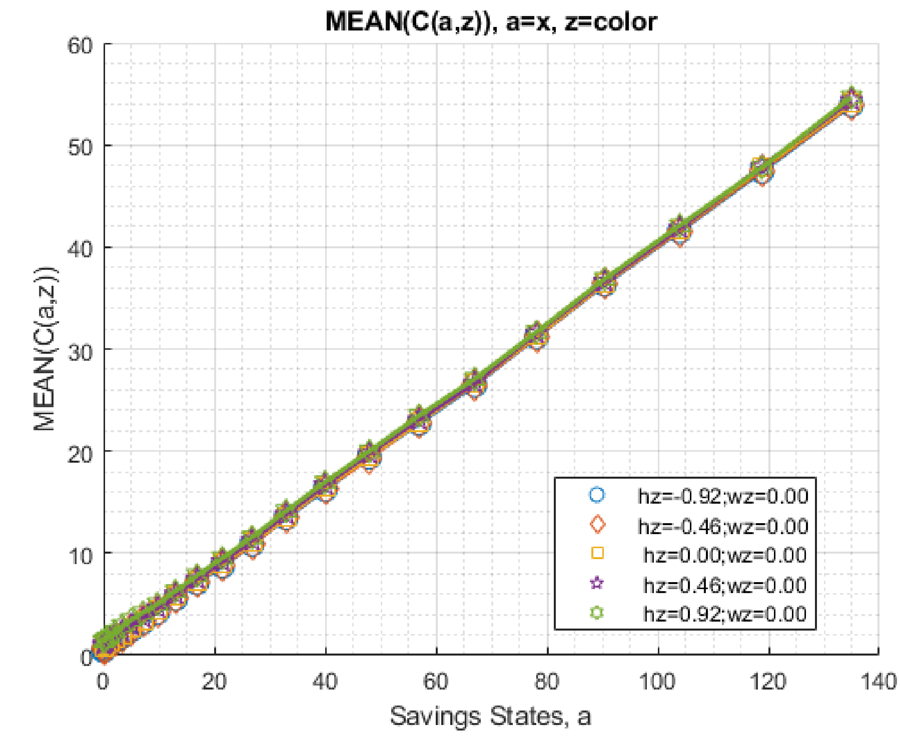
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(a,z))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

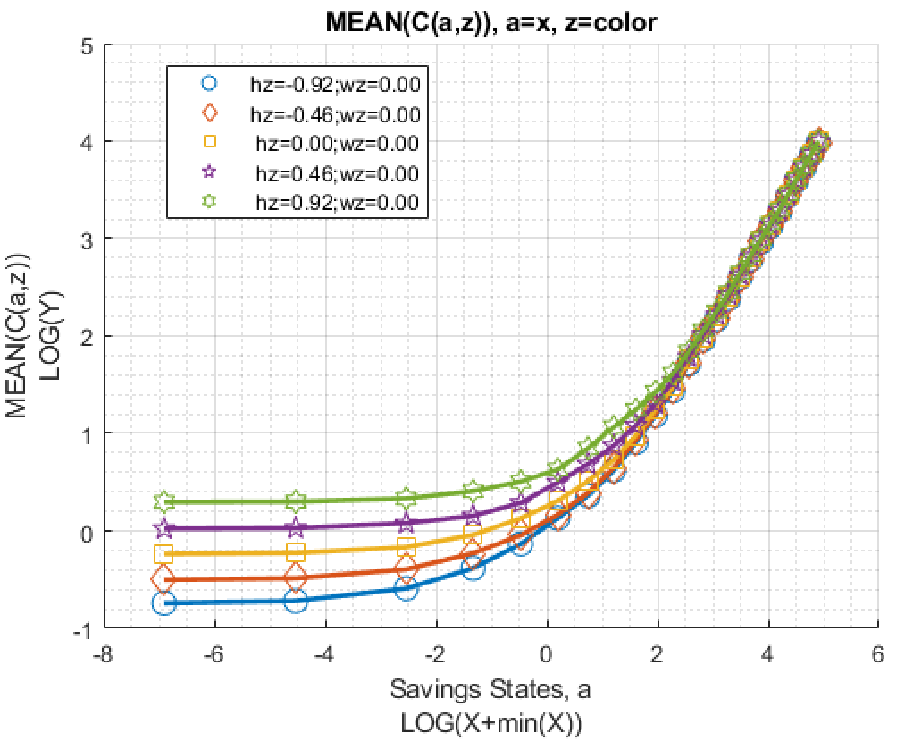




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





3.2.4 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", "k0M1", "K1M1", "K2M1"];
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', 'o', 'd', 's'};
mp_support_graph('cl_colors') = {'red', 'red', 'red', 'blue', 'blue', 'blue'};
```

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

Tabulate value and policies:

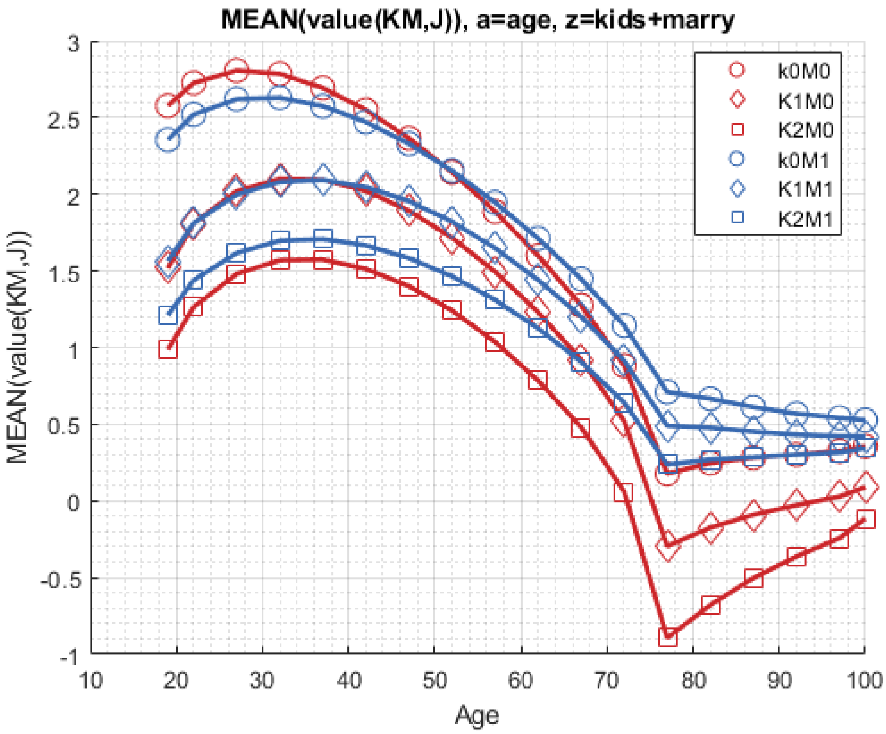
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(KM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(VAL(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        2.5769          2.726         2.8073         2.7855         2.6939
      2        2       0        1.5197         1.8098         2.0206         2.1004         2.0952
      3        3       0        0.9869         1.2649         1.4811         1.5698         1.5738
      4        1       1        2.3544         2.5201         2.6205         2.6297         2.5748
      5        2       1         1.564         1.8114         1.9978         2.0809         2.0936
      6        3       1        1.2123         1.4401         1.6171         1.6965         1.7071

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(KM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(AP(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        12.876          12.86         12.976         13.068          13.14
      2        2       0         12.86          12.84         12.916         13.016          13.06
      3        3       0        12.824         12.792         12.884         12.988          12.94
      4        1       1        12.832         12.796         12.892          12.98         13.052
      5        2       1        12.824         12.788         12.856          12.94         13.004
      6        3       1        12.768         12.724         12.828         12.904         12.972
```

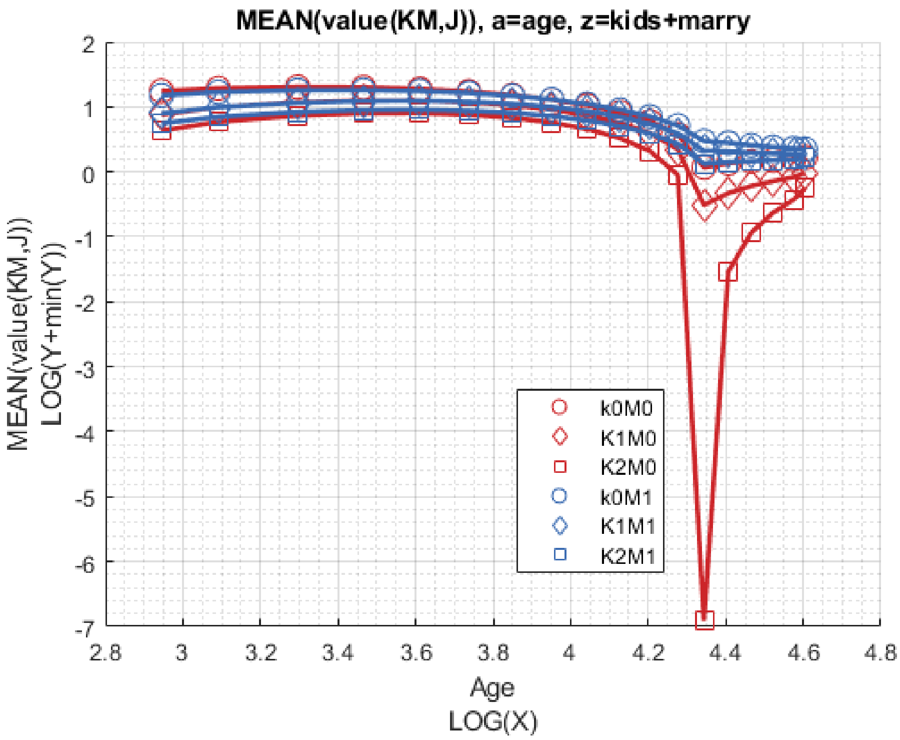
% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(C(KM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permute);

```
xxx  MEAN(C(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        6.3895         6.4629         6.6288         6.7554         6.8327
      2        2       0        6.4025         6.4709         6.6411         6.7667         7.2326
      3        3       0        6.4139         6.4906         6.6473         6.7745         8.3105
      4        1       1        6.6365         6.7334         6.9186         7.0691         7.1681
      5        2       1        6.6219         6.7043         6.8923         7.0386         7.1354
      6        3       1        6.6135         6.7111         6.8733         7.0145         7.1123
```

Graph Mean Values:

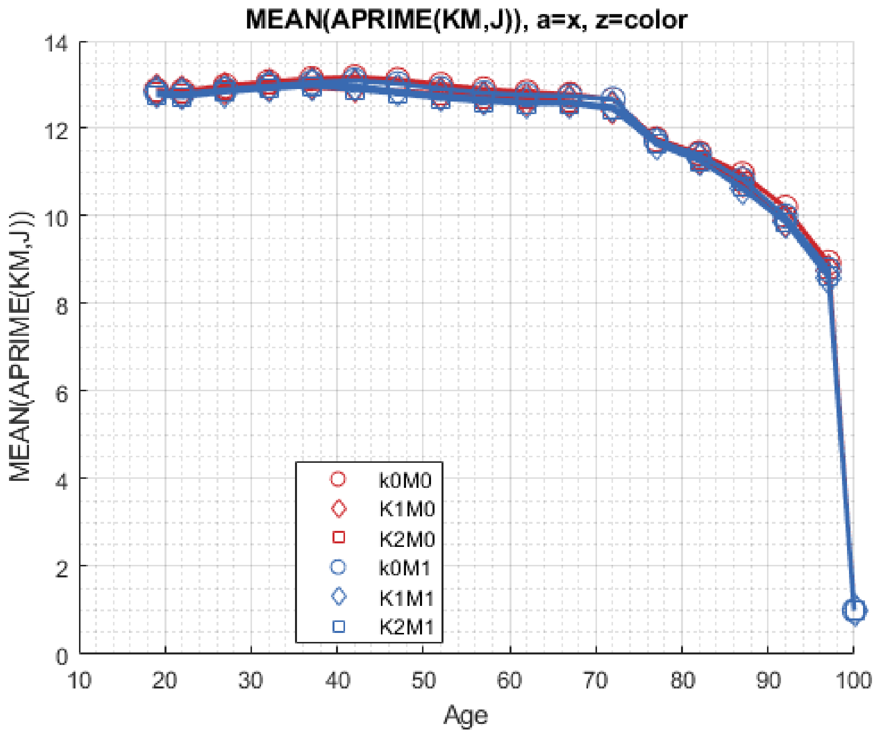
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(KM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

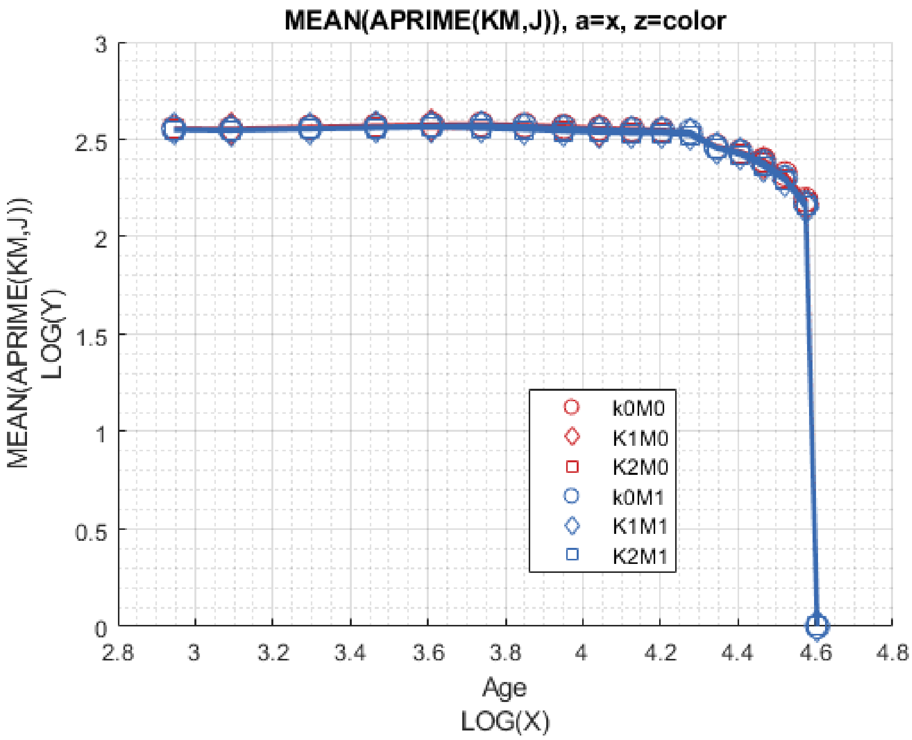




Graph Mean Savings Choices:

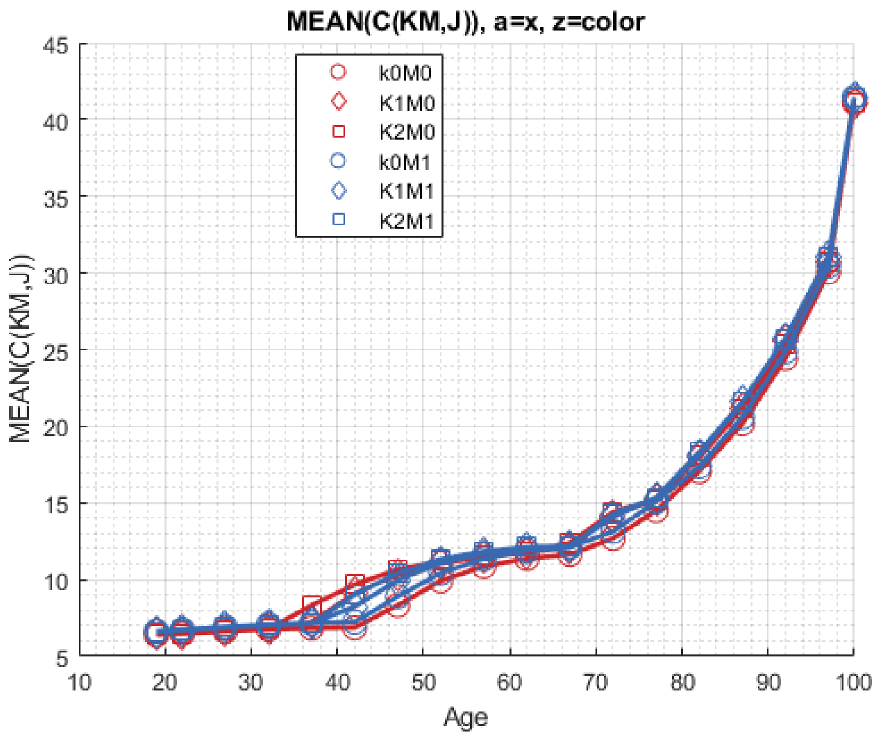
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(KM,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

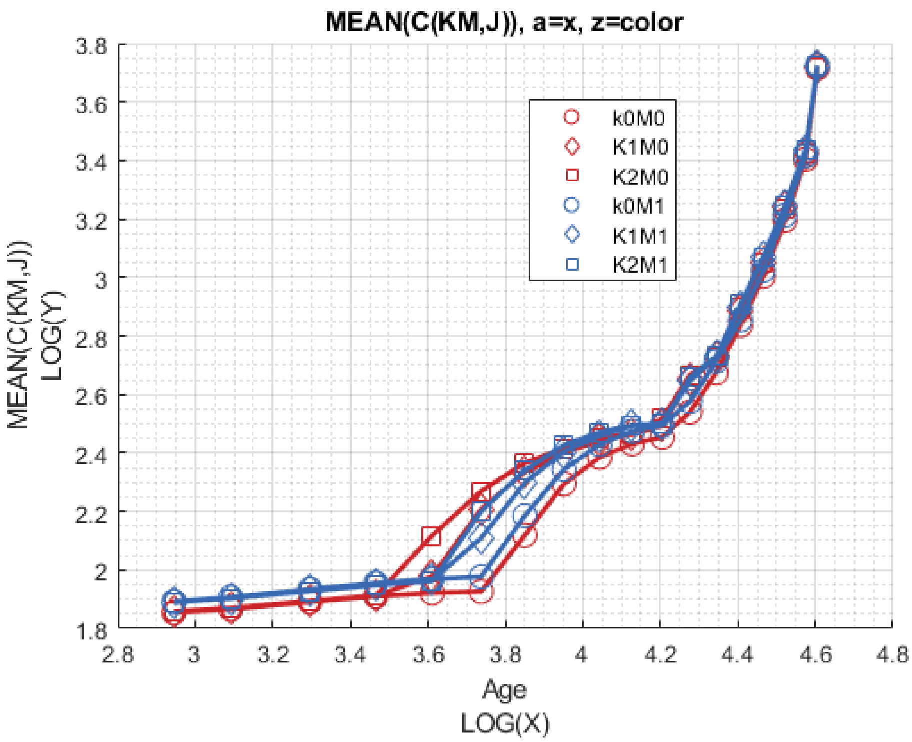




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





3.2.5 Analyze Education 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 = ["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'};
```

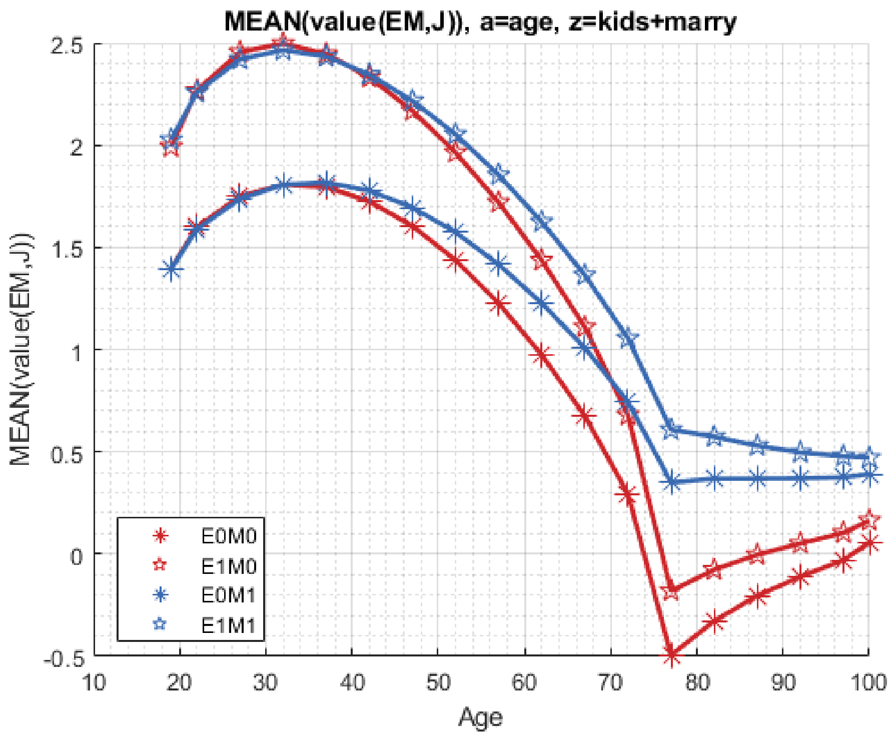
MEAN(VAL(EKM,J)), MEAN(AP(EKM,J)), MEAN(C(EKM,J))

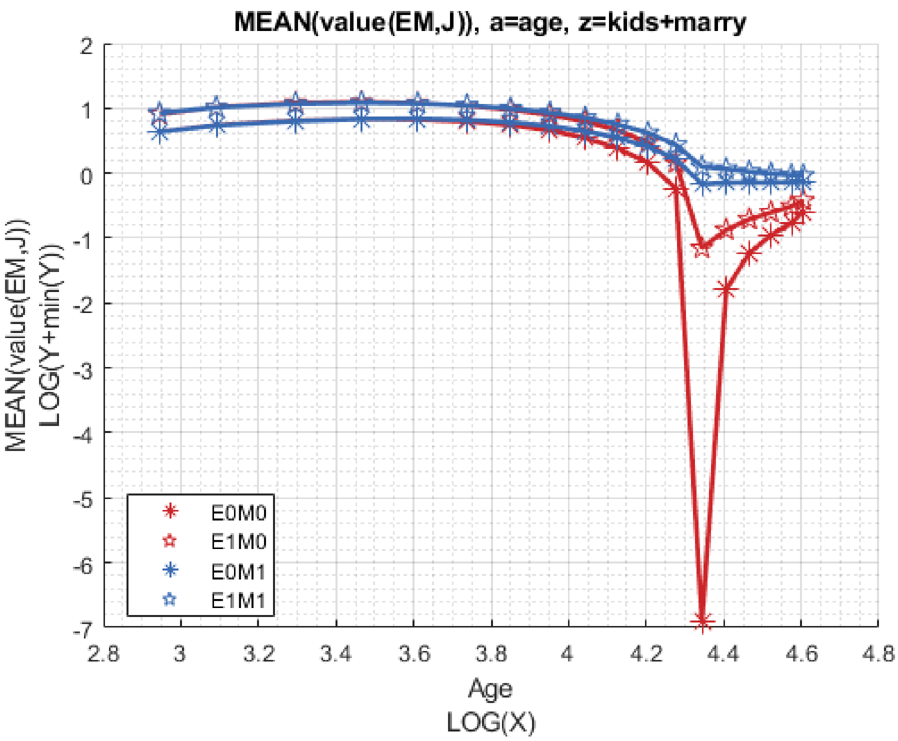
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(EKM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(VAL(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0         1.397         1.5992         1.7503          1.807         1.7952
      2       1       0        1.9919          2.268         2.4556         2.4968         2.4467
      3       0       1        1.3943          1.589         1.7371         1.8066         1.8173
      4       1       1        2.0262         2.2588         2.4199         2.4648         2.4331

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(EKM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(AP(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0        12.923         12.909         12.957         13.011         13.016
      2       1       0        12.784         12.752         12.893         13.037         13.077
      3       0       1        12.883         12.837         12.899         12.949         12.987
      4       1       1        12.733         12.701         12.819         12.933         13.032

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(EKM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(C(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0        6.3781         6.4224         6.5232         6.6018         7.3509
      2       1       0        6.4259         6.5271         6.7549         6.9292         7.5663
      3       0       1        6.5686         6.6336         6.7481         6.8408         6.9145
      4       1       1        6.6793          6.799         7.0414         7.2407         7.3627
```

Graph Mean Values:

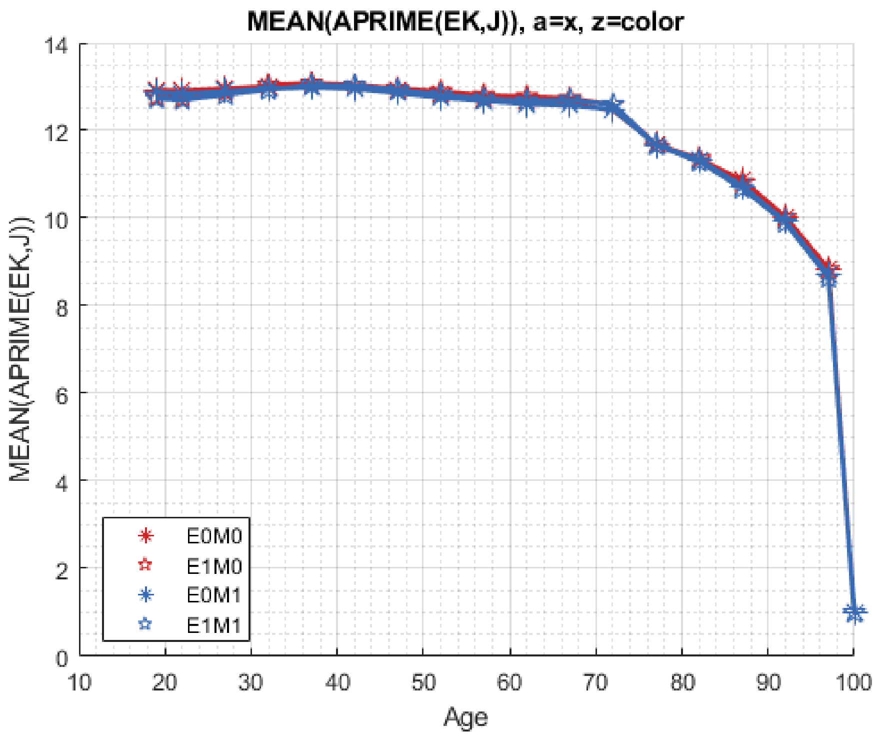
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(EM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

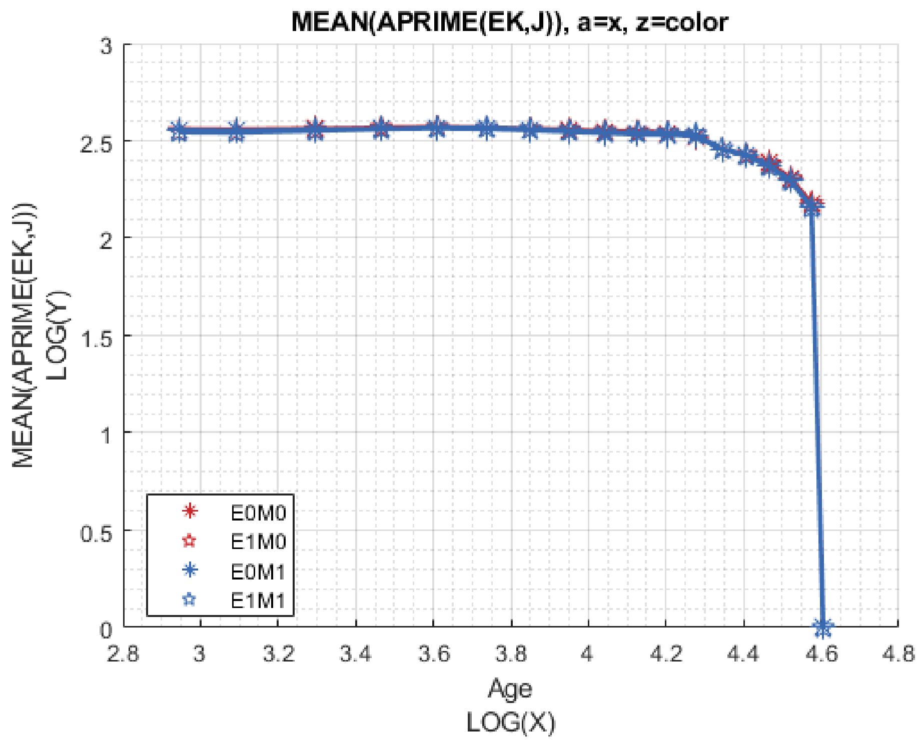




Graph Mean Savings Choices:

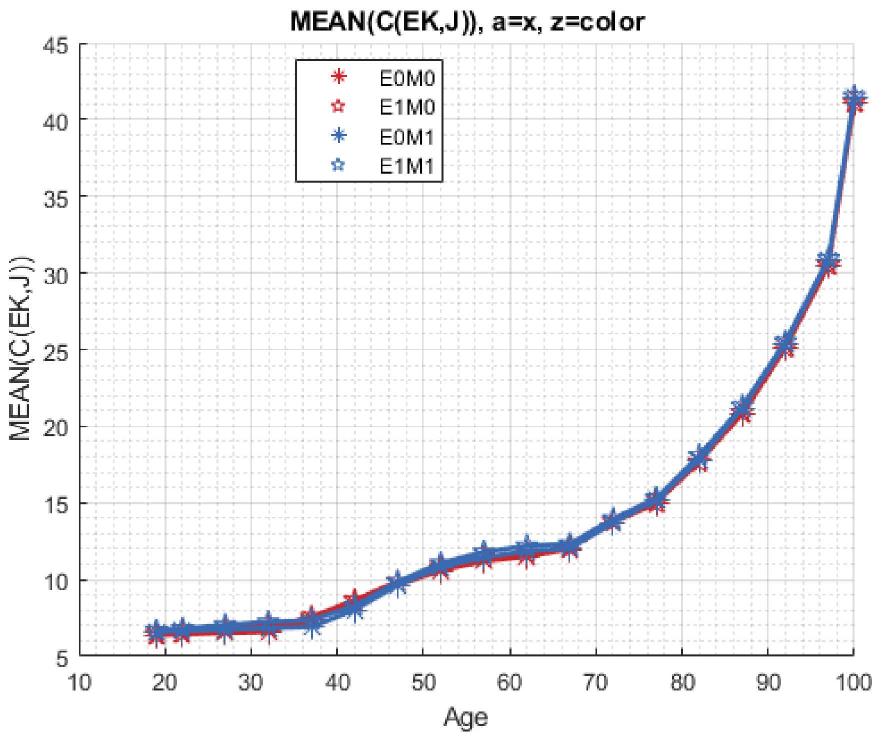
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(EK,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

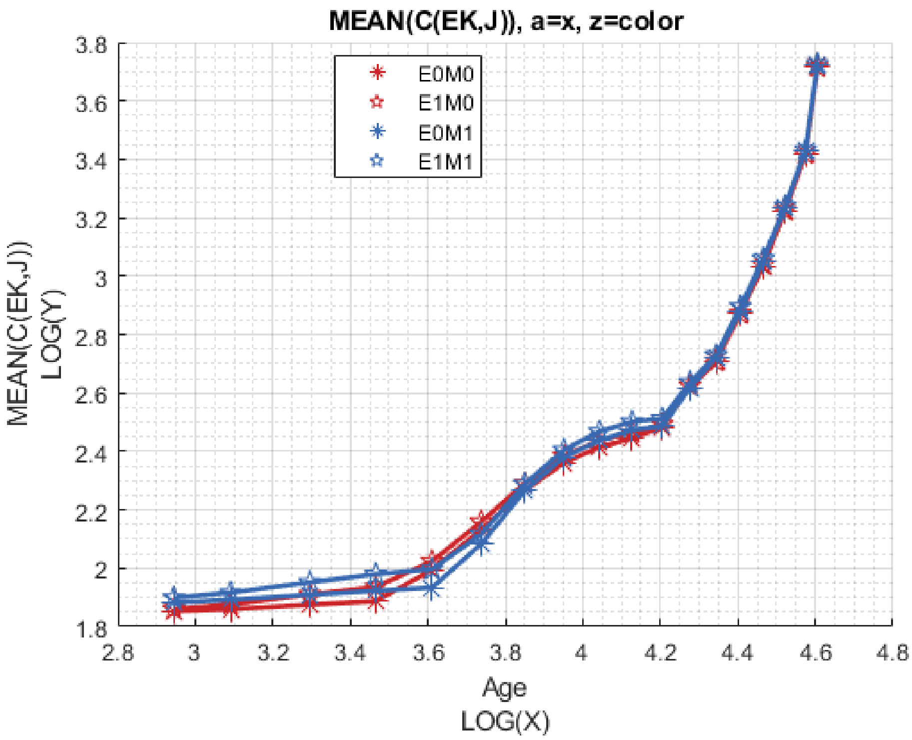




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(EK,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





3.3 SNW_VFI_PARAM Small Solution Analysis

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for policy function with vectorized bisection. Small Solution Analysis. Small Solution Analysis, husband 5 shocks, wife 1 shocks.

3.3.1 Test SNW_VFI_MAIN Defaults Small

Call the function with defaults.

```
mp_param = snw_mp_param('default_small');
[V_VFI,ap_VFI,cons_VFI,mp_valpol_more] = snw_vfi_main_bisec_vec(mp_param);
```

```
SNW_VFI_MAIN: Finished Age Group:18 of 18
SNW_VFI_MAIN: Finished Age Group:17 of 18
SNW_VFI_MAIN: Finished Age Group:16 of 18
SNW_VFI_MAIN: Finished Age Group:15 of 18
SNW_VFI_MAIN: Finished Age Group:14 of 18
SNW_VFI_MAIN: Finished Age Group:13 of 18
SNW_VFI_MAIN: Finished Age Group:12 of 18
SNW_VFI_MAIN: Finished Age Group:11 of 18
SNW_VFI_MAIN: Finished Age Group:10 of 18
SNW_VFI_MAIN: Finished Age Group:9 of 18
SNW_VFI_MAIN: Finished Age Group:8 of 18
SNW_VFI_MAIN: Finished Age Group:7 of 18
SNW_VFI_MAIN: Finished Age Group:6 of 18
SNW_VFI_MAIN: Finished Age Group:5 of 18
SNW_VFI_MAIN: Finished Age Group:4 of 18
SNW_VFI_MAIN: Finished Age Group:3 of 18
SNW_VFI_MAIN: Finished Age Group:2 of 18
SNW_VFI_MAIN: Finished Age Group:1 of 18
Elapsed time is 0.300958 seconds.
Completed SNW_VFI_MAIN;SNW_MP_PARAM=default_small;SNW_MP_CONTROL=default_base
```

3.3.2 Small 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 = [19, 22:5:97, 100];
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('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'}, {'Hshock', 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});
```

3.3.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% 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') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm

xxx  MEAN(VAL(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_Hshock__0_91976    mean_Hshock__0_45988    mean_Hshock_0    mean_Hsho
    _____    _________    ____________________    ____________________    _____________    _________

      1              0          -7.2503                 -5.2835             -3.7134             -2.
      2      0.0097656          -7.1324                 -5.1969             -3.6417             -2.
      3       0.078125          -6.4761                 -4.7107             -3.2467             -2.
      4        0.26367          -5.3284                 -3.8496             -2.5707             -1.
      5          0.625           -3.966                 -2.8108             -1.7735             -0.8
      6         1.2207           -2.597                 -1.7598            -0.95324             -0.2
      7         2.1094          -1.3566                -0.78465            -0.19312              0.3
      8         3.3496         -0.30963                0.066182             0.47754              0.9
      9              5          0.53291                  0.7744              1.0517               1.
     10         7.1191           1.1909                  1.3448              1.5289               1.
     11         9.7656           1.6974                  1.7957              1.9172               2.
     12         12.998           2.0855                  2.1488              2.2293               2.
     13         16.875           2.3829                  2.4244              2.4781               2.
     14         21.455           2.6121                  2.6396              2.6759               2.
     15         26.797             2.79                  2.8086              2.8335               2.
     16         32.959           2.9295                  2.9423              2.9596               2.
     17             40           3.0397                  3.0487               3.061               3.
     18         47.979           3.1277                  3.1342              3.1429               3.
     19         56.953           3.1987                  3.2033              3.2097               3.
     20         66.982           3.2563                  3.2597              3.2645                3
     21         78.125           3.3036                  3.3062              3.3097               3.
     22         90.439           3.3427                  3.3446              3.3473               3.
     23         103.98           3.3753                  3.3768              3.3788               3.
     24         118.82           3.4026                  3.4038              3.4054               3.
     25            135           3.4257                  3.4266              3.4279               3.

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(A,Z))", ap_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per

xxx  MEAN(AP(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_Hshock__0_91976    mean_Hshock__0_45988    mean_Hshock_0    mean_Hsho
    _____    _________    ____________________    ____________________    _____________    _________

      1              0        0.0031715                0.018272             0.06218              0.1
      2      0.0097656        0.0041856                 0.02014            0.065449               0.
      3       0.078125           0.0195                0.043246            0.094774              0.2
      4        0.26367          0.11997                 0.14608             0.20791              0.3
      5          0.625          0.37091                 0.40356              0.4536              0.5
      6         1.2207          0.81324                 0.85499             0.90105               1.
      7         2.1094            1.495                  1.5366              1.6037               1.
      8         3.3496           2.4456                  2.4842                2.56               2.
      9              5           3.7173                  3.7541              3.8276               3.
     10         7.1191           5.3706                  5.4067              5.4784               5.
     11         9.7656            7.409                   7.445              7.5165               7.
     12         12.998            9.829                  9.8609              9.9322               10
     13         16.875           12.836                  12.867               12.93               13
     14         21.455           16.418                  16.447               16.51               16
     15         26.797           20.467                  20.493              20.551                2
```

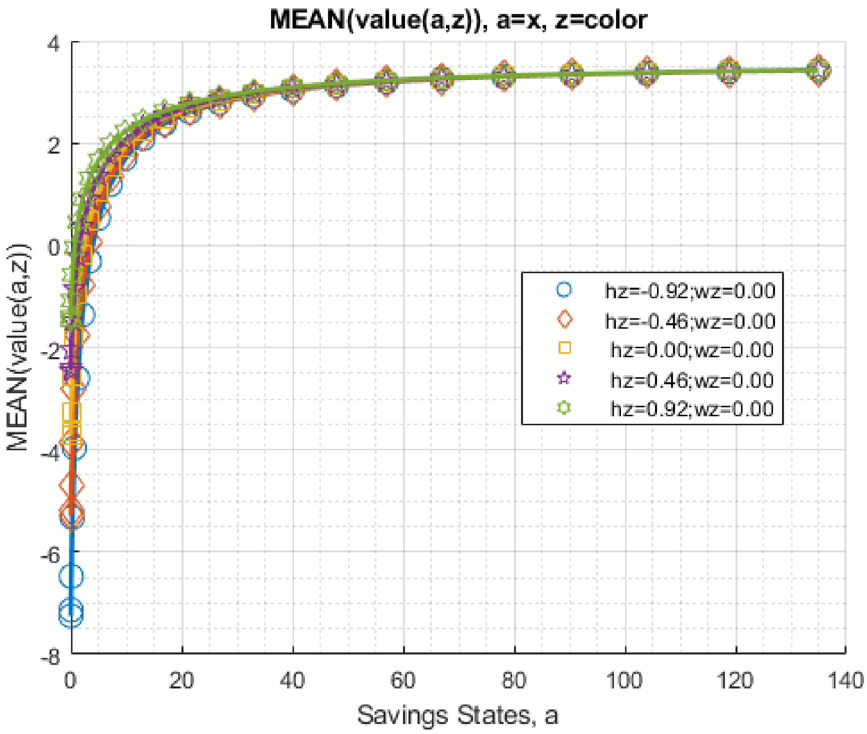
```
    16        32.959            25.147                  25.172              25.226             25
    17            40            30.496                  30.527              30.586             30
    18        47.979             36.57                  36.598               36.66             36
    19        56.953            43.556                  43.586              43.646             43
    20        66.982             51.34                   51.37              51.434             51
    21        78.125            59.622                  59.655              59.719             59
    22        90.439            68.907                  68.939              69.003             69
    23        103.98             79.38                   79.41               79.47             79
    24        118.82            90.679                  90.711              90.776             90
    25           135            102.99                  103.02              103.08             10
```

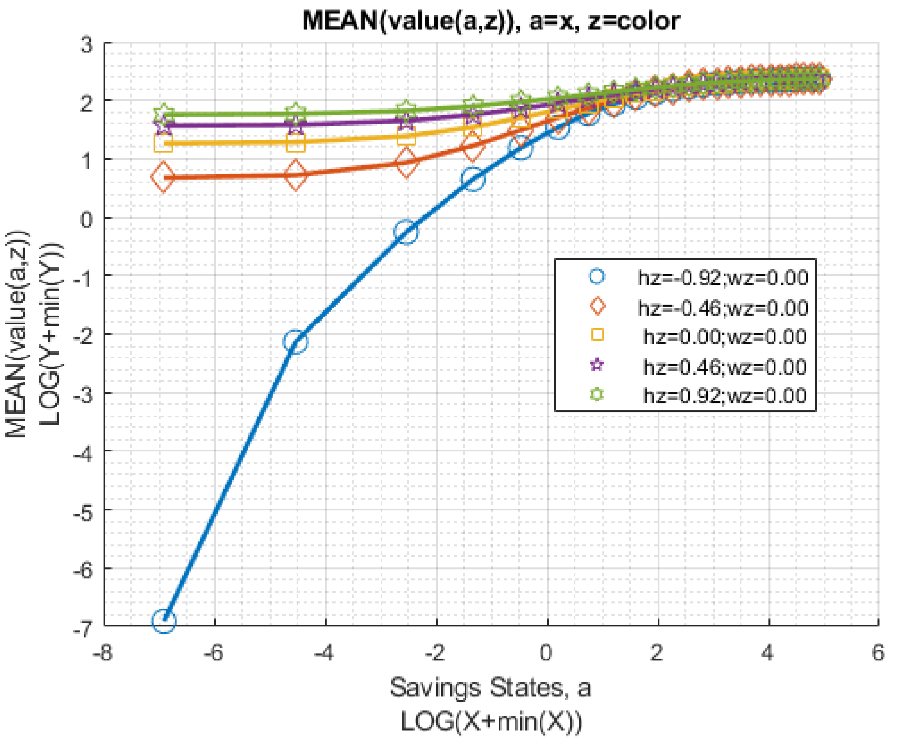
% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(C(A,Z))", cons_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_permute);

```
xxx  MEAN(C(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings     mean_Hshock__0_91976    mean_Hshock__0_45988    mean_Hshock_0    mean_Hsho
    -----    ---------    --------------------    --------------------    -------------    ---------

      1             0           0.47551                 0.59993              0.77031            1.0
      2     0.0097656           0.48595                 0.60949              0.77845            1.0
      3      0.078125            0.5508                  0.6664              0.82902            1.0
      4       0.26367           0.66781                 0.78066              0.93266            1.1
      5         0.625              0.84                 0.94566               1.1089            1.3
      6        1.2207            1.0944                  1.1901               1.3565             1.
      7        2.1094            1.4504                  1.5453               1.6898            1.9
      8        3.3496            1.9462                  2.0432               2.1782            2.4
      9             5             2.597                   2.695               2.8314            3.0
     10        7.1191            3.4097                  3.5079               3.6454            3.8
     11        9.7656             4.449                  4.5467               4.6839            4.8
     12        12.998            5.7861                  5.8876               6.0245            6.2
     13        16.875            7.2831                  7.3861               7.5301            7.7
     14        21.455            9.0219                  9.1255               9.2698            9.4
     15        26.797            11.176                  11.283               11.433            11.
     16        32.959            13.652                   13.76               13.913            14.
     17            40            16.478                   16.58               16.728            16.
     18        47.979            19.667                  19.772               19.917            20.
     19        56.953            23.101                  23.202                23.35            23.
     20        66.982             26.96                  27.062               27.205             27
     21        78.125            31.613                  31.712               31.855             32
     22        90.439            36.623                  36.724               36.866            37.
     23        103.98            41.873                  41.975               42.122            42.
     24        118.82            47.794                  47.894               48.036            48.
     25           135            54.264                  54.368               54.515            54.
```

Graph Mean Values:

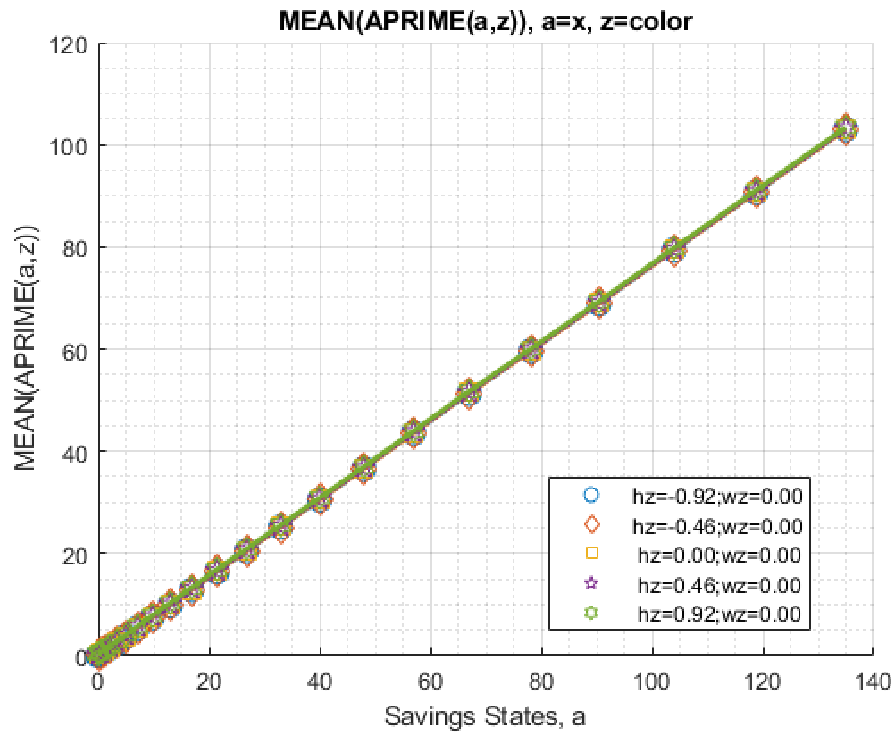
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

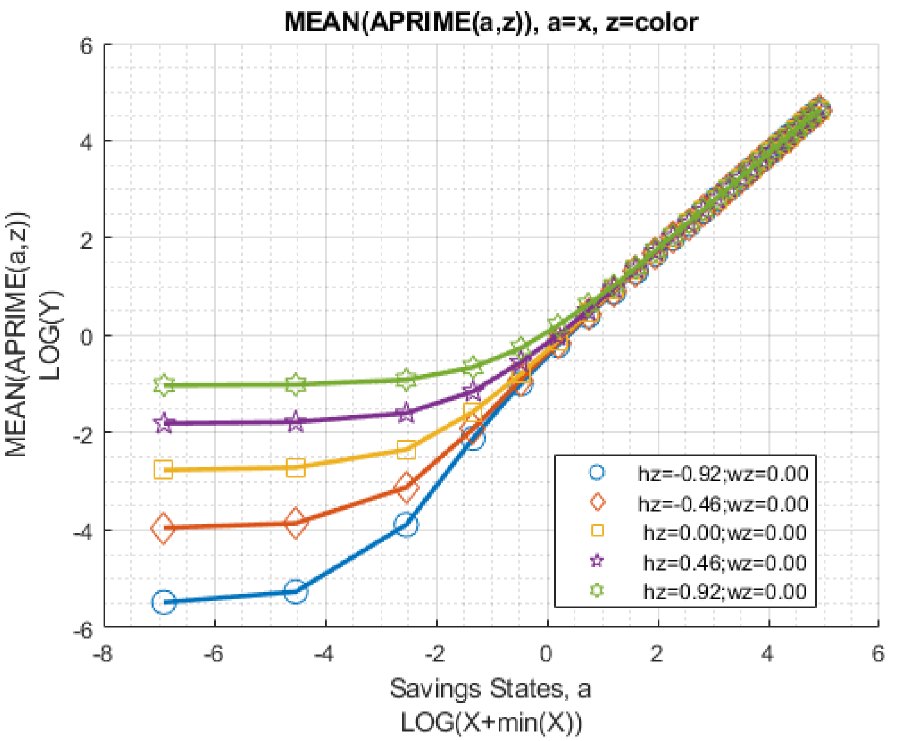




Graph Mean Savings Choices:

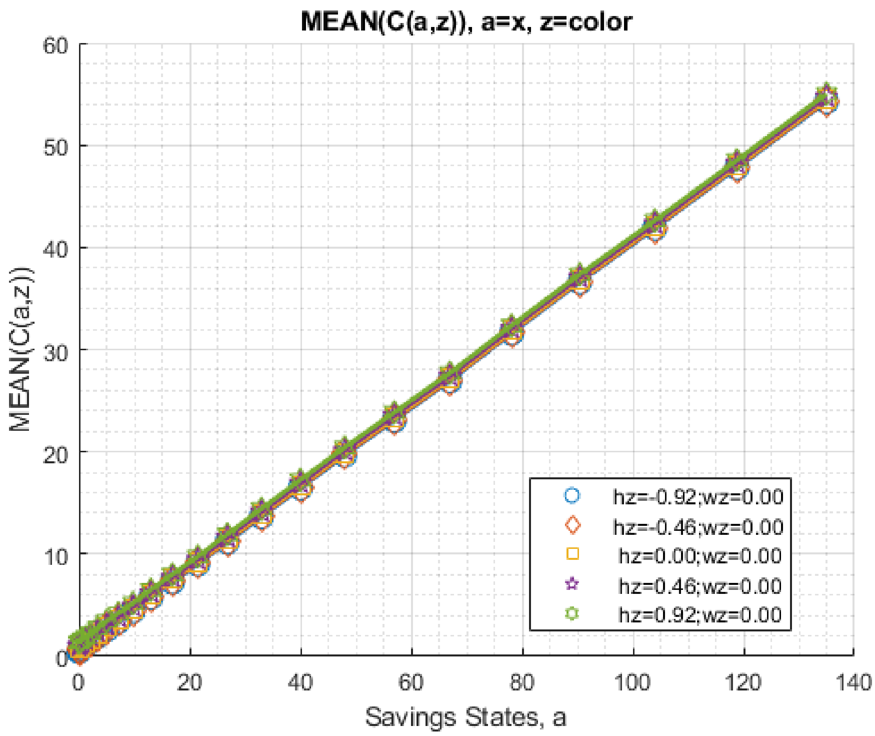
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(a,z))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

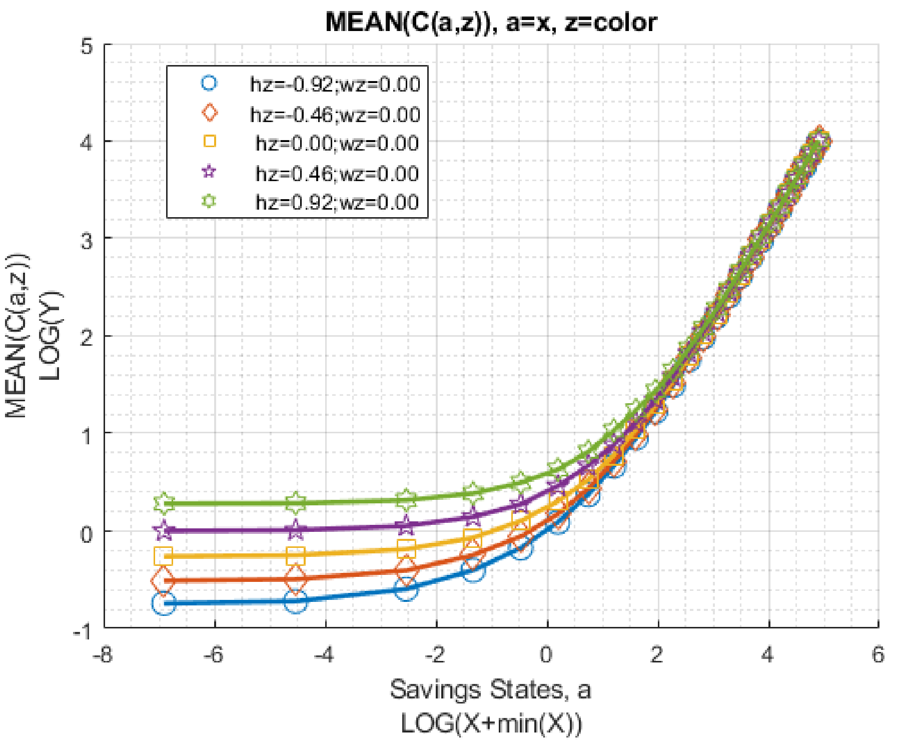




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





3.3.4 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", "k0M1", "K1M1", "K2M1"];
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', 'o', 'd', 's'};
mp_support_graph('cl_colors') = {'red', 'red', 'red', '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
tb_az_v = ff_summ_nd_array("MEAN(VAL(KM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per
```

```
xxx  MEAN(VAL(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        2.6201         2.7665         2.8454         2.8242         2.7343
      2        2       0        1.5887         1.8727         2.0791         2.1577         2.1527
      3        3       0        1.0708         1.3439         1.5546         1.6415         1.6452
      4        1       1         2.395         2.5572         2.6553         2.6638          2.609
      5        2       1        1.6234         1.8656         2.0481         2.1297         2.1416
      6        3       1        1.2806         1.5046         1.6784         1.7562          1.766
```

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(KM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

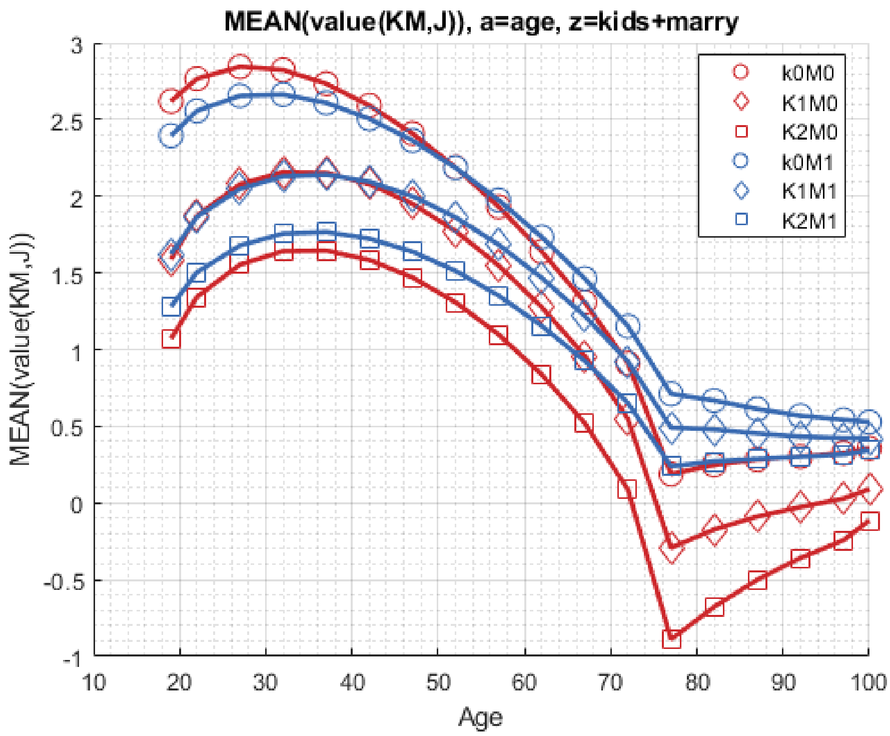
```
xxx  MEAN(AP(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0         34.74         34.523         34.415         34.265         34.042
      2        2       0        34.413         34.138         33.952         33.709         33.376
      3        3       0        34.001         33.777         33.635         33.423         33.115
      4        1       1        34.621         34.408         34.308          34.16         33.941
      5        2       1        34.473         34.246         34.109         33.911         33.625
      6        3       1        34.106         33.916         33.813         33.643         33.381
```

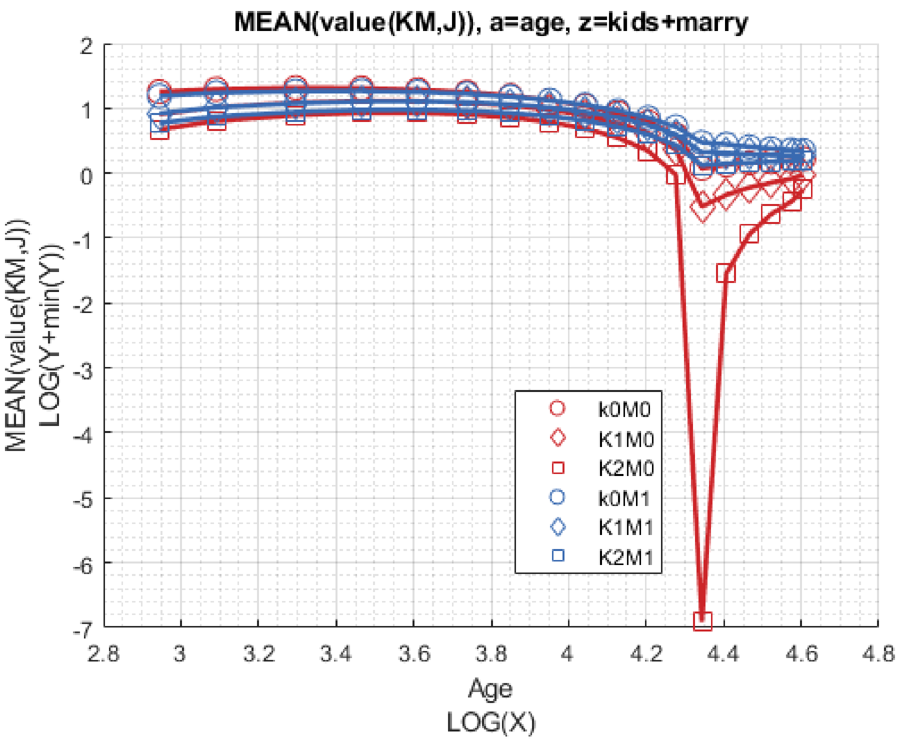
% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(C(KM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permute);

```
xxx  MEAN(C(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0         6.778         7.0757         7.3697         7.6652         7.9826
      2        2       0        7.1055         7.4611         7.8334         8.2212         8.6487
      3        3       0        7.5174         7.8216         8.1497         8.5069           8.91
      4        1       1        7.1206         7.4352         7.7476         8.0644         8.4009
      5        2       1        7.2345         7.5595         7.9048         8.2677         8.6677
      6        3       1        7.5771         7.8633         8.1717         8.5043         8.8779
```

Graph Mean Values:

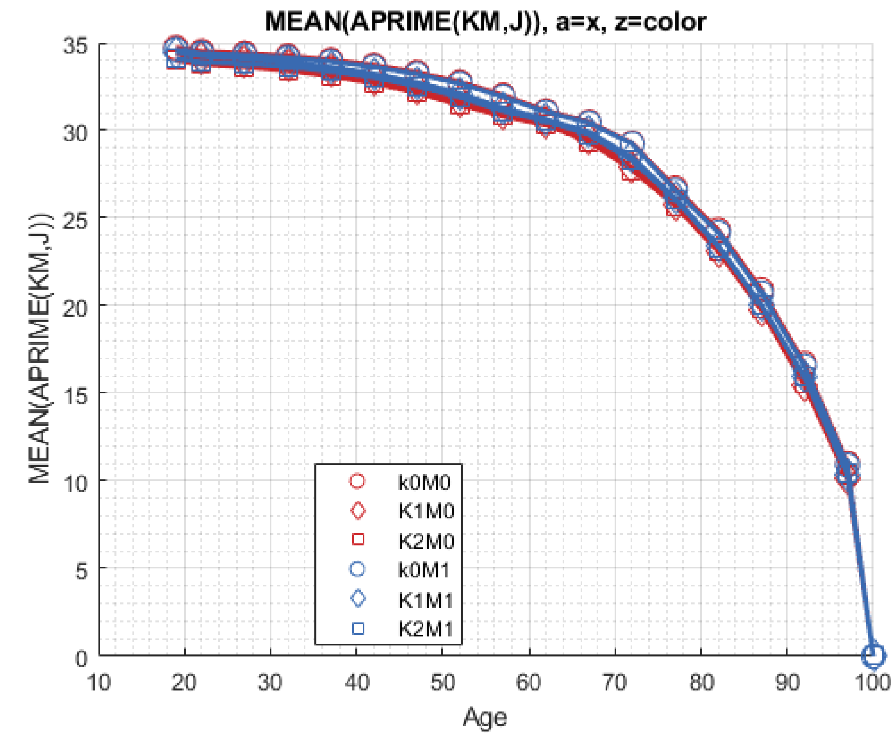
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(KM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

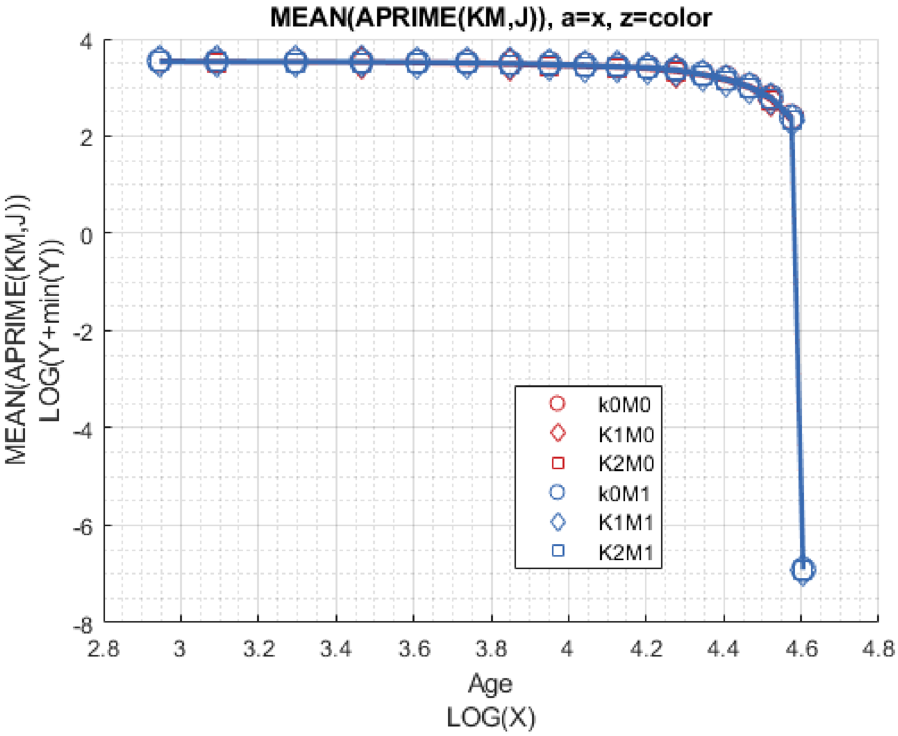




Graph Mean Savings Choices:

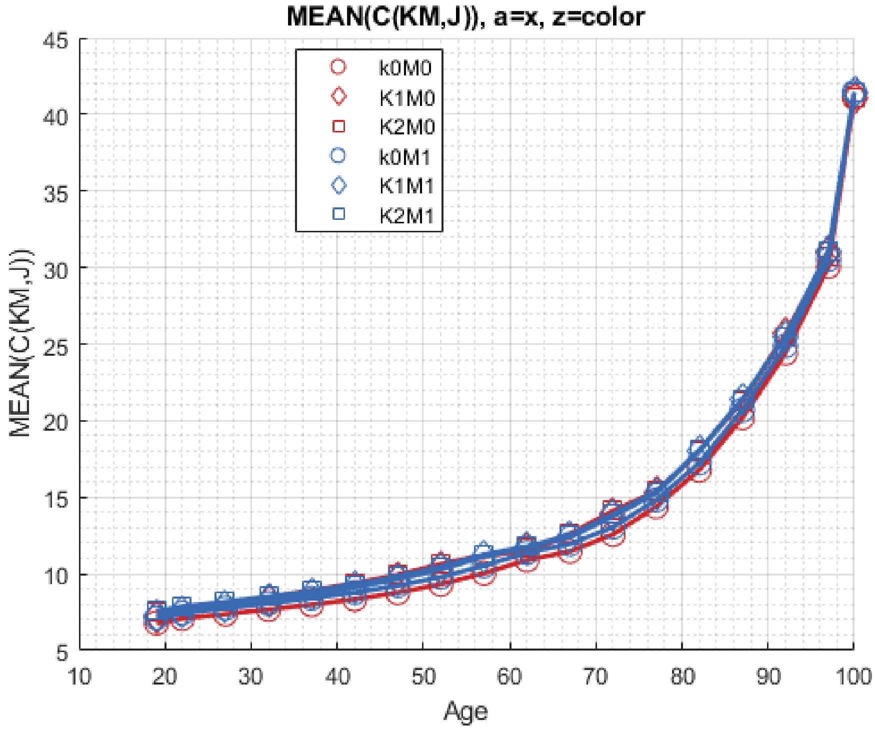
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(KM,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

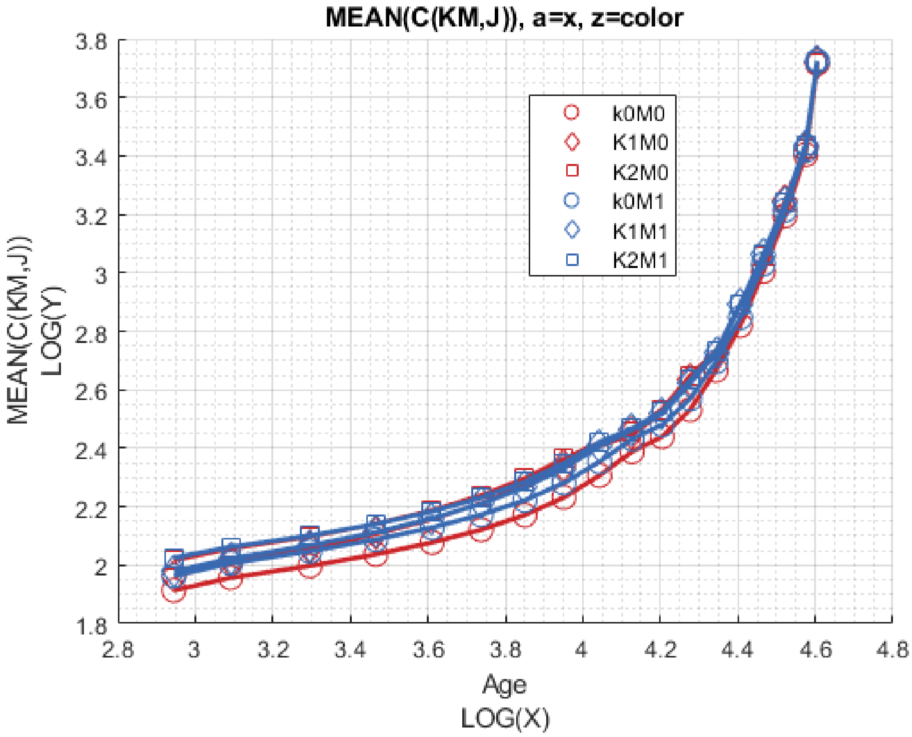




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





3.3.5 Analyze Education 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 = ["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'};
```

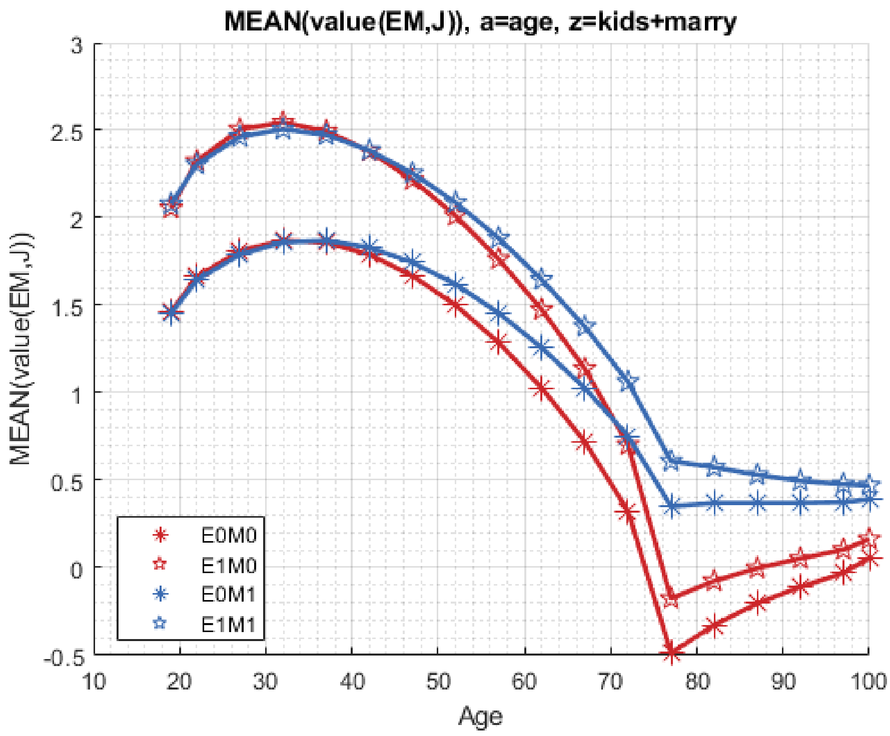
MEAN(VAL(EKM,J)), MEAN(AP(EKM,J)), MEAN(C(EKM,J))

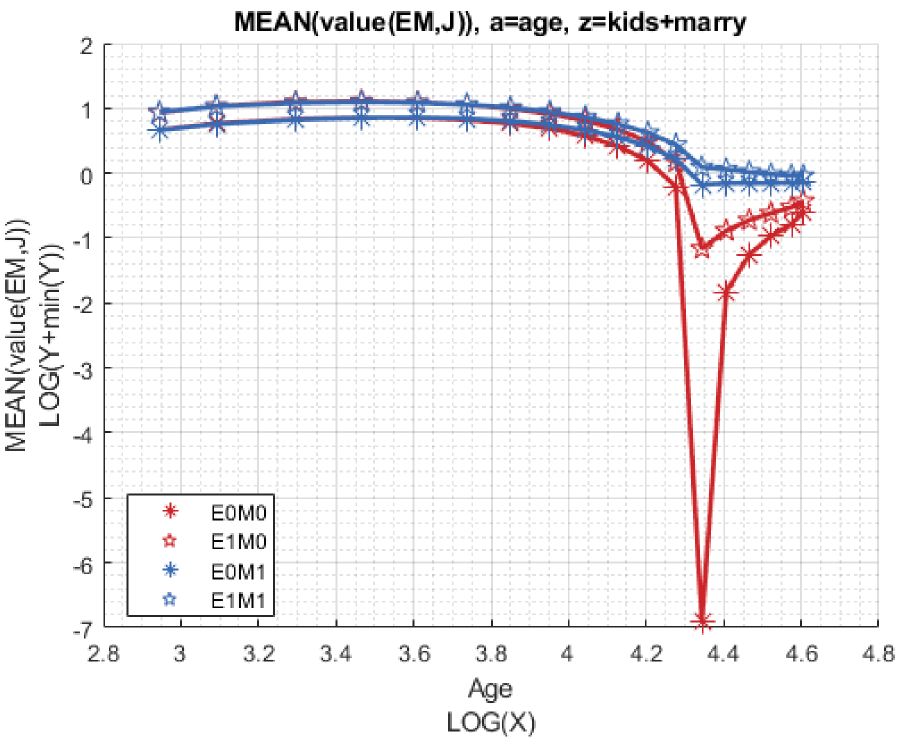
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(EKM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(VAL(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0         1.4646         1.6636         1.8129         1.8698         1.8591
      2       1       0         2.0551         2.3251         2.5065         2.5458         2.4958
      3       0       1         1.4539         1.6452         1.7914         1.8602         1.8706
      4       1       1         2.0788         2.3064         2.4632         2.5062         2.4738

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(EKM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(AP(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0         34.471         34.227         34.028         33.781         33.465
      2       1       0         34.298         34.065         33.974         33.817         33.557
      3       0       1         34.505         34.289          34.12         33.903         33.618
      4       1       1         34.294         34.091         34.033         33.907         33.681

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(EKM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(C(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0          7.047         7.3391          7.647          7.982         8.3628
      2       1       0         7.2203         7.5665         7.9215         8.2802         8.6647
      3       0       1         7.1855          7.465         7.7602         8.0826         8.4489
      4       1       1          7.436         7.7737         8.1225          8.475         8.8488
```

Graph Mean Values:

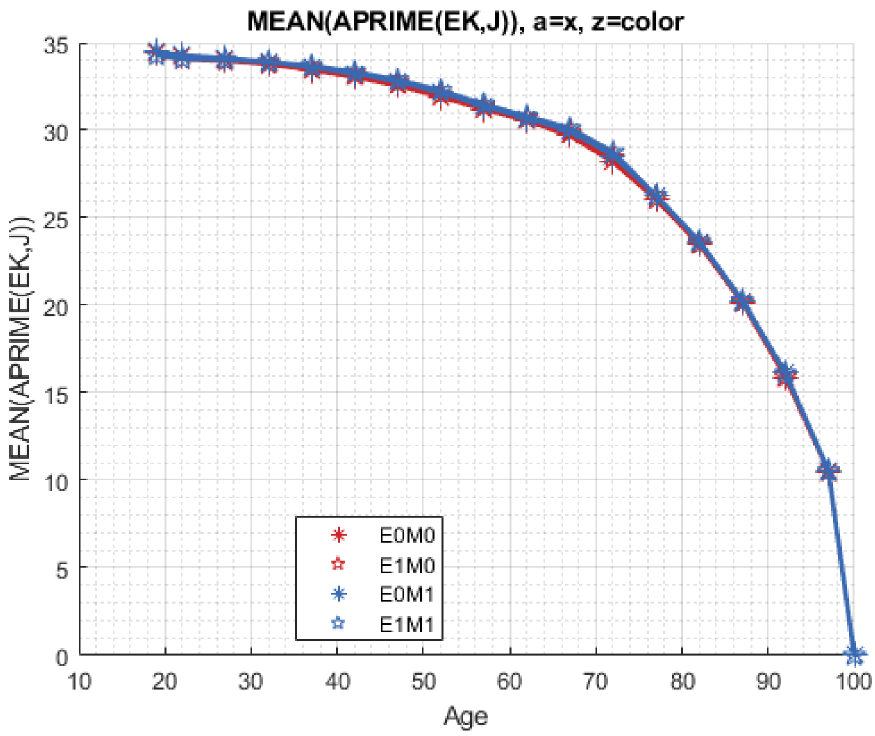
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(EM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

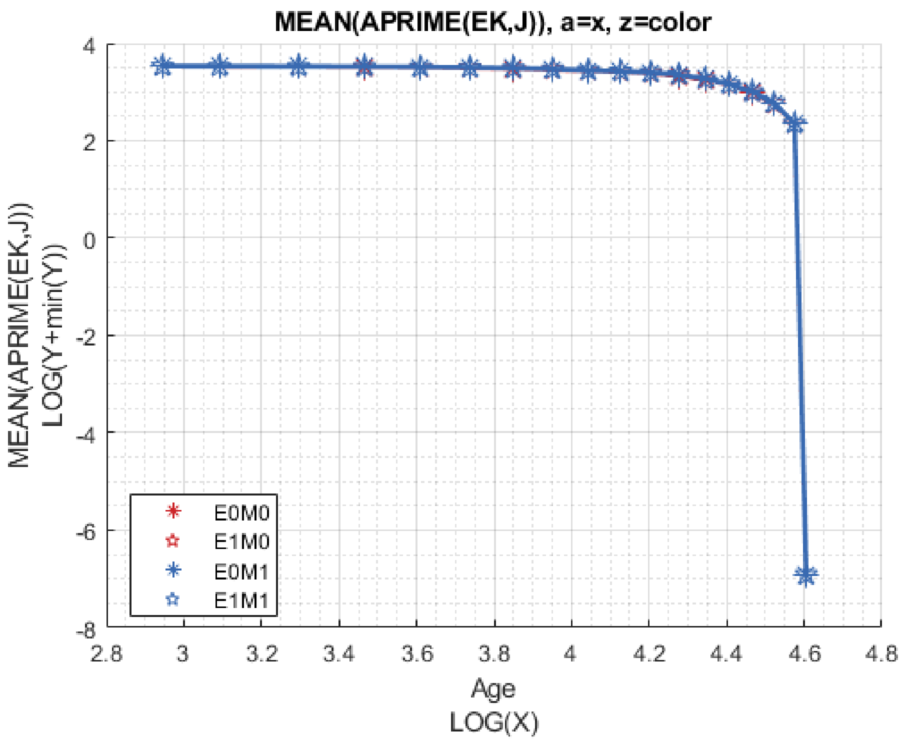




Graph Mean Savings Choices:

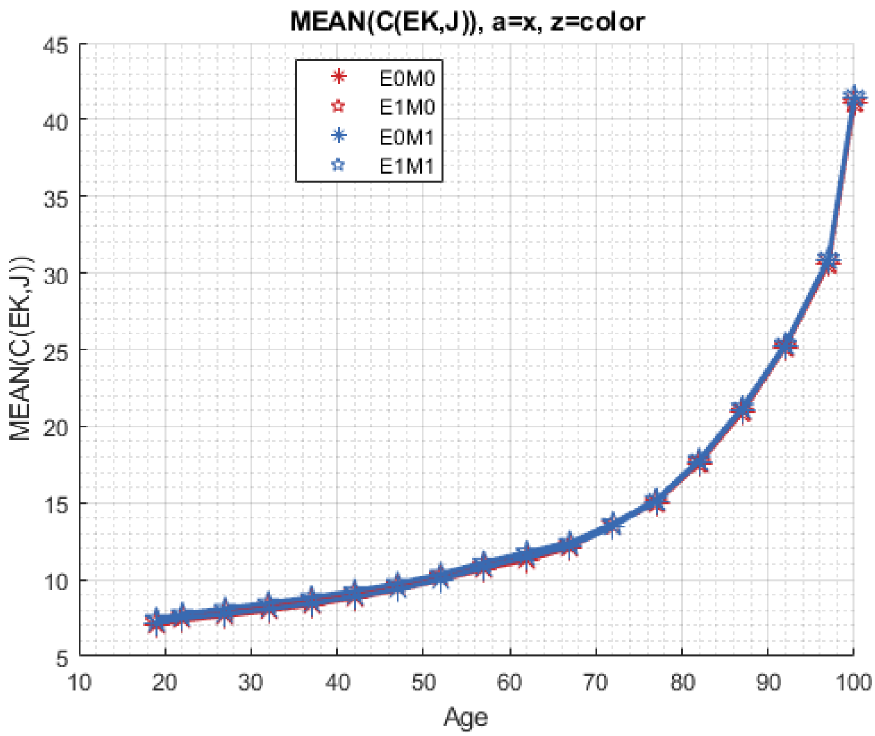
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(EK,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

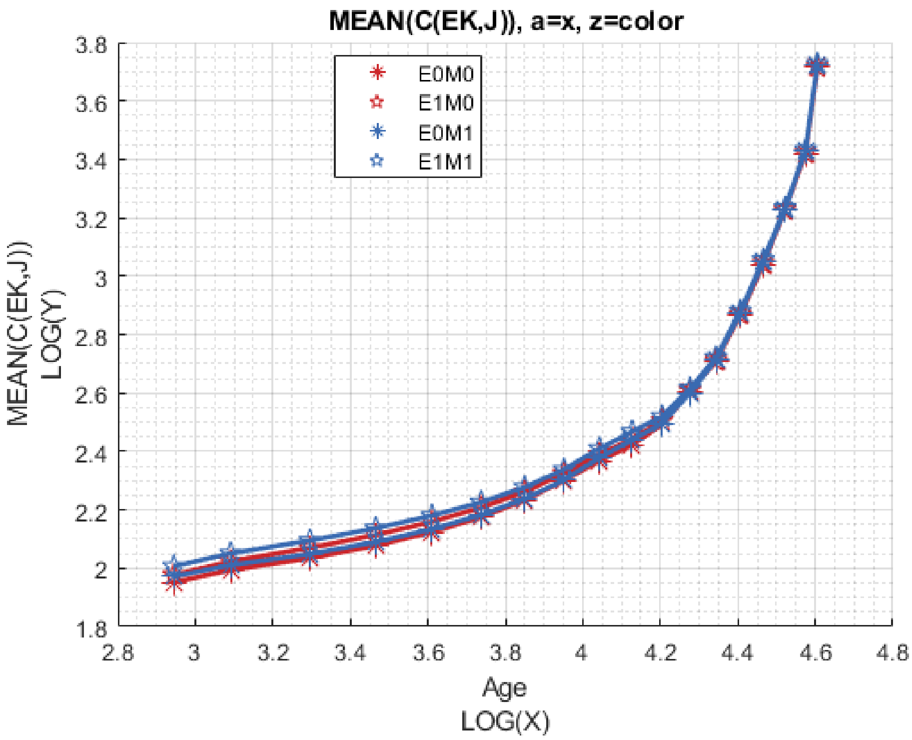




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(EK,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





3.4 SNW_VFI_PARAM Small 5/3 Solution Analysis

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for policy function with vectorized bisection. Small Solution Analysis, husband 5 shocks, wife 3 shocks.

3.4.1 Test SNW_VFI_MAIN Defaults Small

Call the function with defaults.

```
mp_param = snw_mp_param('default_small53');
[V_VFI,ap_VFI,cons_VFI,mp_valpol_more] = snw_vfi_main_bisec_vec(mp_param);

SNW_VFI_MAIN: Finished Age Group:18 of 18
SNW_VFI_MAIN: Finished Age Group:17 of 18
SNW_VFI_MAIN: Finished Age Group:16 of 18
SNW_VFI_MAIN: Finished Age Group:15 of 18
SNW_VFI_MAIN: Finished Age Group:14 of 18
SNW_VFI_MAIN: Finished Age Group:13 of 18
SNW_VFI_MAIN: Finished Age Group:12 of 18
SNW_VFI_MAIN: Finished Age Group:11 of 18
SNW_VFI_MAIN: Finished Age Group:10 of 18
SNW_VFI_MAIN: Finished Age Group:9 of 18
SNW_VFI_MAIN: Finished Age Group:8 of 18
SNW_VFI_MAIN: Finished Age Group:7 of 18
SNW_VFI_MAIN: Finished Age Group:6 of 18
SNW_VFI_MAIN: Finished Age Group:5 of 18
SNW_VFI_MAIN: Finished Age Group:4 of 18
SNW_VFI_MAIN: Finished Age Group:3 of 18
SNW_VFI_MAIN: Finished Age Group:2 of 18
SNW_VFI_MAIN: Finished Age Group:1 of 18
Elapsed time is 4.405878 seconds.
Completed SNW_VFI_MAIN;SNW_MP_PARAM=default_small53;SNW_MP_CONTROL=default_base
```

3.4.2 Small 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 = [19, 22:5:97, 100];
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz=%3.2
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('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});
```

3.4.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% 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') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
mp_support_graph('it_legend_select') = 9; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_perm
```

```
xxx  MEAN(VAL(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group      savings      mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5    mean
    -----     ---------     ----------     ----------     ----------     ----------     ----------    ----

      1              0        -7.882        -5.7348         -4.085        -2.8085        -1.8188        -7
      2      0.0097656       -7.7212        -5.6181        -3.9885        -2.7215        -1.7365        -6
      3       0.078125       -6.8428        -4.9741        -3.4642        -2.2559        -1.3014        -6
      4        0.26367       -5.4385        -3.9243        -2.6322        -1.5472       -0.66003        -5
      5          0.625       -3.9212        -2.7733        -1.7454       -0.83276      -0.049731        -3
      6         1.2207       -2.4938        -1.6768       -0.89179       -0.17035        0.48277        -2
      7         2.1094       -1.2492       -0.69701       -0.12719        0.42663        0.94331        -1
      8         3.3496      -0.21896        0.14154          0.536        0.94666         1.3467       -0.
      9              5       0.60263        0.83385         1.0995         1.3921         1.6958        0.
     10         7.1191         1.242         1.3897         1.5663         1.7699         1.9935         1
     11         9.7656        1.7339         1.8287         1.9458          2.086          2.247         1
     12         12.998        2.1109         2.1725         2.2505         2.3468         2.4616         2
     13         16.875        2.4005         2.4411         2.4935         2.5599         2.6414         2
     14         21.455        2.6243         2.6513          2.687          2.733         2.7911         2
     15         26.797        2.7986          2.817         2.8415         2.8737         2.9153         2
     16         32.959        2.9355         2.9482         2.9654         2.9882         3.0181         2
     17             40        3.0441          3.053         3.0651         3.0815         3.1033         3
     18         47.979        3.1309         3.1373          3.146         3.1579         3.1739         3
     19         56.953         3.201         3.2056          3.212         3.2207         3.2326         3
     20         66.982         3.258         3.2614         3.2662         3.2727         3.2816         3
     21         78.125        3.3049         3.3074          3.311         3.3159         3.3226         3
     22         90.439        3.3437         3.3456         3.3483          3.352         3.3572         3
     23         103.98        3.3761         3.3775         3.3796         3.3824         3.3864         3
     24         118.82        3.4032         3.4044         3.4059         3.4082         3.4113         3
     25            135        3.4262         3.4271         3.4283         3.4301         3.4325         3
```

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(A,Z))", ap_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_per
```

```
xxx  MEAN(AP(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group      savings      mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5    mean
    -----     ---------     ----------     ----------     ----------     ----------     ----------    ----

      1              0     0.0019699       0.012569       0.042745        0.11279        0.27641      0.00
      2      0.0097656     0.0026822       0.013934       0.045039        0.11613        0.28116      0.00
      3       0.078125      0.012725       0.028983       0.063429        0.14286        0.31616       0.0
      4        0.26367      0.086315        0.10163        0.14743        0.23696        0.42136         0
      5          0.625       0.30688        0.32748        0.37158        0.48022        0.66179        0.
      6         1.2207       0.71364        0.74681        0.79866        0.90672         1.1008         0
      7         2.1094          1.38         1.4122          1.473         1.5674         1.7516         1
      8         3.3496        2.3237         2.3547         2.4215         2.5241         2.6817         2
      9              5        3.5866         3.6174         3.6852         3.8049         3.9657
     10         7.1191        5.2284          5.258         5.3256         5.4521          5.645         5
     11         9.7656        7.2439         7.2717         7.3376         7.4667         7.6795         7
     12         12.998         9.683         9.7074         9.7687         9.8969         10.122
     13         16.875         12.69         12.713         12.768         12.888         13.118         1
     14         21.455        16.245         16.267         16.322         16.434         16.656         1
```

15	26.797	20.297	20.314	20.362	20.471	20.681	2
16	32.959	24.972	24.994	25.04	25.136	25.339	
17	40	30.336	30.36	30.416	30.523	30.708	3
18	47.979	36.418	36.442	36.497	36.607	36.811	3
19	56.953	43.405	43.428	43.48	43.586	43.789	4
20	66.982	51.176	51.202	51.26	51.37	51.568	5
21	78.125	59.419	59.447	59.507	59.624	59.834	5
22	90.439	68.737	68.762	68.819	68.932	69.146	6
23	103.98	79.211	79.236	79.291	79.402	79.606	7
24	118.82	90.517	90.543	90.601	90.715	90.921	9
25	135	102.82	102.85	102.9	103.01	103.22	

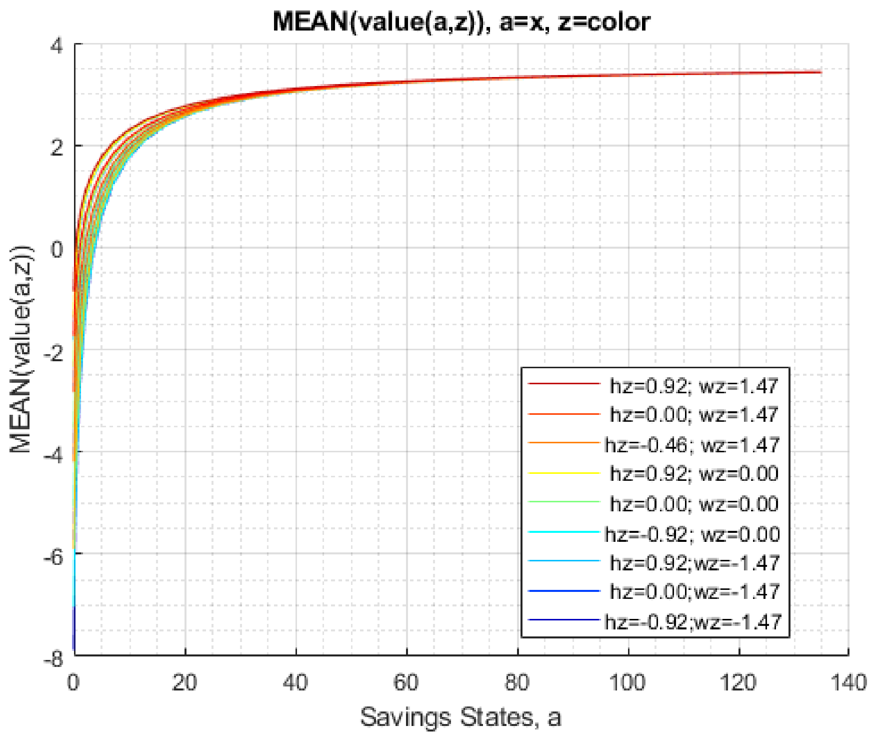
% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(C(A,Z))", cons_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_permute);

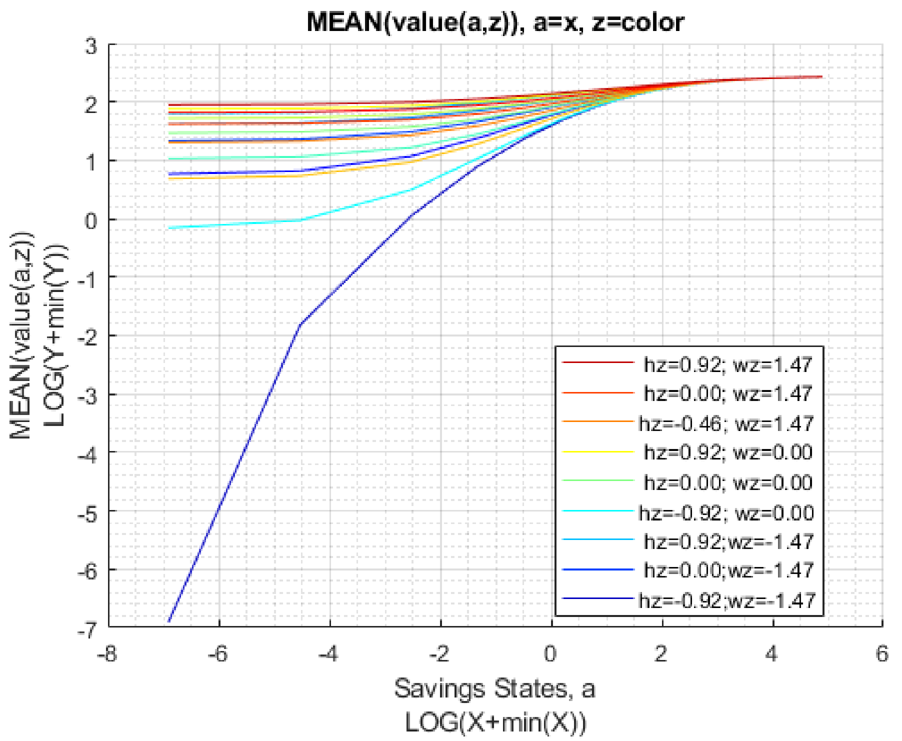
xxx MEAN(C(A,Z)) xxxxxxxxxxxxxxxxxxxxxxxxxxxx

group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5	mean
1	0	0.36263	0.48831	0.66884	0.92666	1.2761	0.4
2	0.0097656	0.37339	0.4984	0.67798	0.93473	1.2828	0.
3	0.078125	0.44367	0.56349	0.73958	0.98789	1.3276	0.5
4	0.26367	0.58796	0.70825	0.87262	1.1106	1.439	0.6
5	0.625	0.7912	0.90541	1.0708	1.2892	1.6202	0.8
6	1.2207	1.082	1.1826	1.3394	1.5577	1.8756	1.
7	2.1094	1.4544	1.5549	1.7017	1.9328	2.26	1.
8	3.3496	1.958	2.0586	2.1984	2.4204	2.7735	
9	5	2.6182	2.7182	2.8561	3.0601	3.4093	2.
10	7.1191	3.4432	3.5437	3.6809	3.8773	4.1937	3.
11	9.7656	4.5057	4.6075	4.7459	4.939	5.2349	4.
12	12.998	5.824	5.9289	6.0713	6.2649	6.5482	5.
13	16.875	7.322	7.4281	7.5761	7.7778	8.0549	7
14	21.455	9.0868	9.1934	9.342	9.5513	9.8361	9.
15	26.797	11.239	11.351	11.506	11.717	12.014	1
16	32.959	13.72	13.826	13.983	14.207	14.511	13
17	40	16.53	16.635	16.782	16.996	17.316	16
18	47.979	19.712	19.817	19.964	20.174	20.477	19
19	56.953	23.144	23.25	23.4	23.614	23.917	23
20	66.982	27.016	27.118	27.263	27.473	27.781	27
21	78.125	31.708	31.809	31.951	32.154	32.451	31
22	90.439	36.685	36.789	36.934	37.141	37.433	36
23	103.98	41.935	42.038	42.186	42.395	42.696	41
24	118.82	47.848	47.951	48.095	48.301	48.601	47
25	135	54.325	54.429	54.577	54.788	55.087	5

Graph Mean Values:

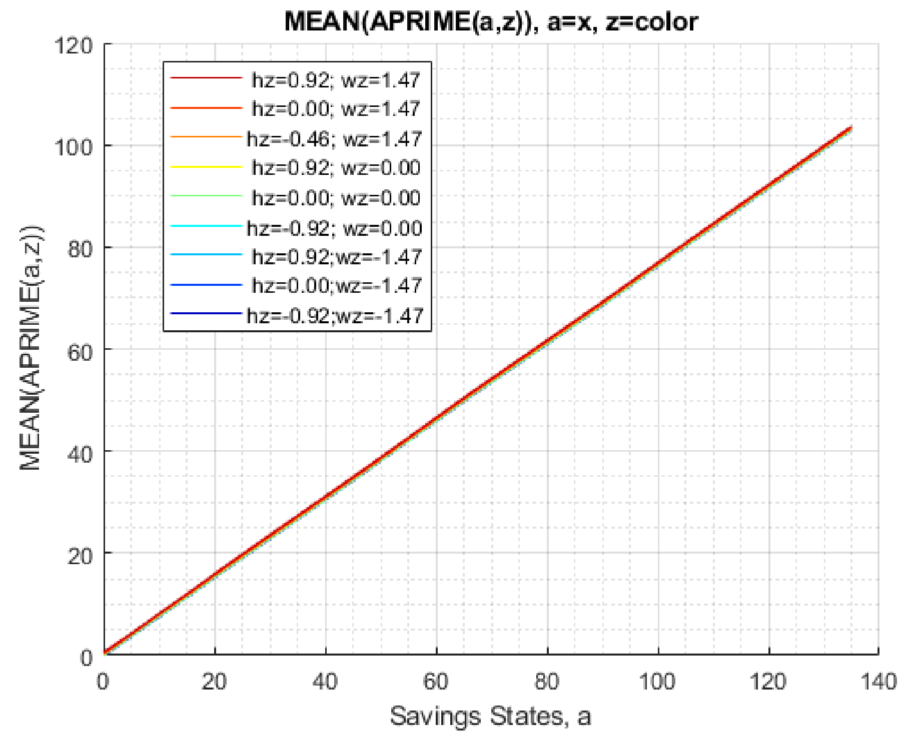
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

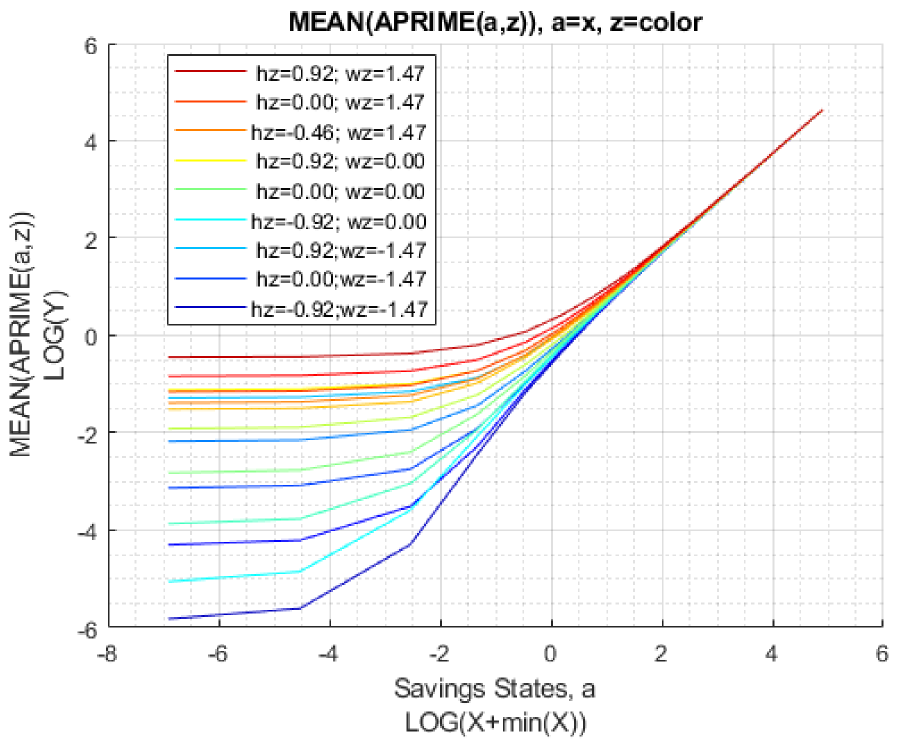




Graph Mean Savings Choices:

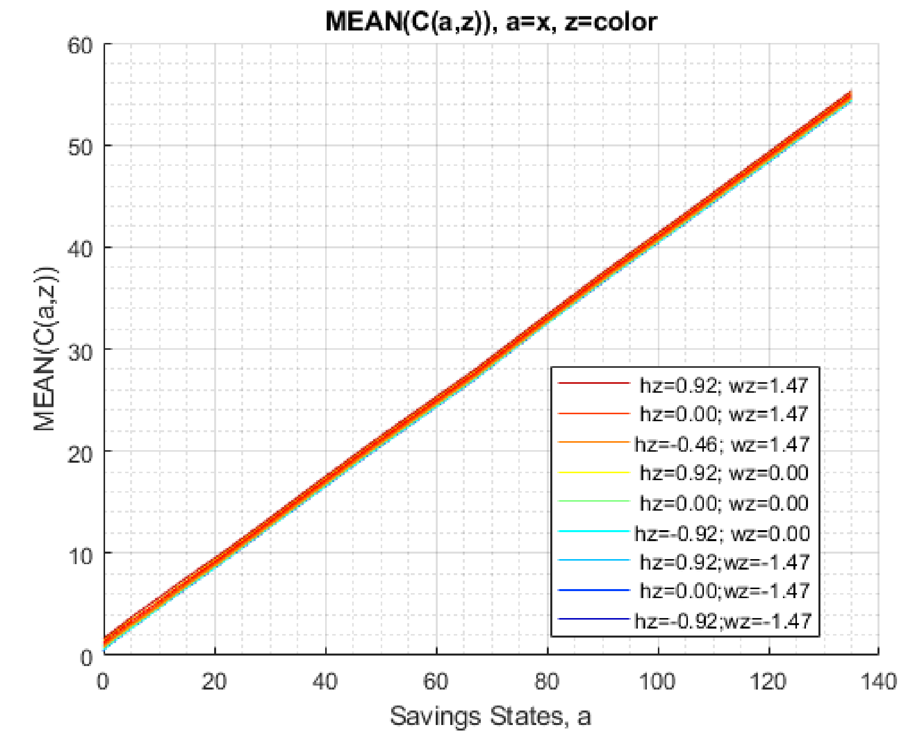
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(a,z))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

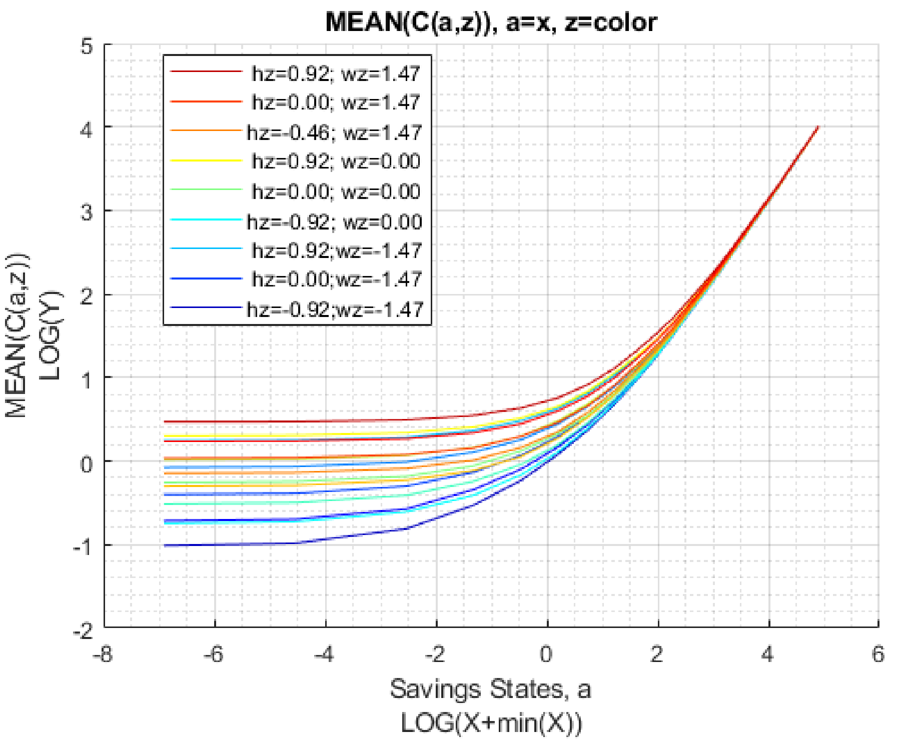




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





3.4.4 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", "k0M1", "K1M1", "K2M1"];
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', 'o', 'd', 's'};
mp_support_graph('cl_colors') = {'red', 'red', 'red', '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
tb_az_v = ff_summ_nd_array("MEAN(VAL(KM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_per
```

```
xxx  MEAN(VAL(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0        2.6201         2.7665         2.8454         2.8242         2.7343
      2        2       0        1.5887         1.8727         2.0791         2.1577         2.1527
      3        3       0        1.0708         1.3439         1.5546         1.6415         1.6452
      4        1       1        2.6802         2.8229         2.9077         2.9056         2.8417
      5        2       1        1.9461          2.164         2.3286         2.3959         2.3955
      6        3       1         1.615         1.8161         1.9731         2.0371         2.0352
```

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(KM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe
```

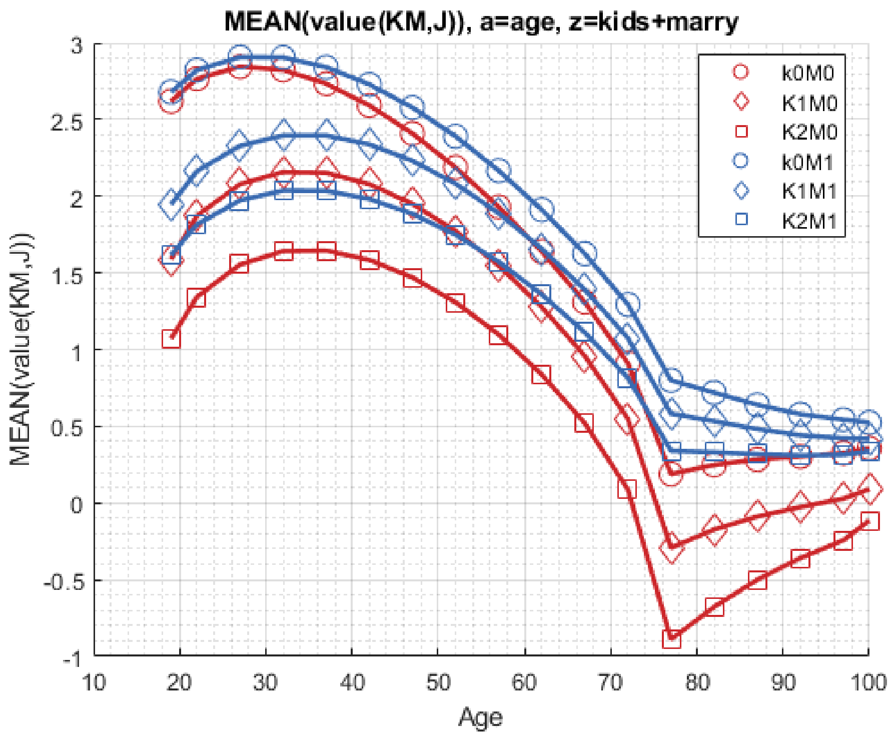
```
xxx  MEAN(AP(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0         34.74         34.523         34.415         34.265         34.042
      2        2       0        34.413         34.138         33.952         33.709         33.376
      3        3       0        34.001         33.777         33.635         33.423         33.115
      4        1       1        34.626         34.419         34.323         34.185         33.976
      5        2       1        34.477         34.253         34.116         33.923         33.642
      6        3       1        34.101         33.913         33.809         33.641         33.384
```

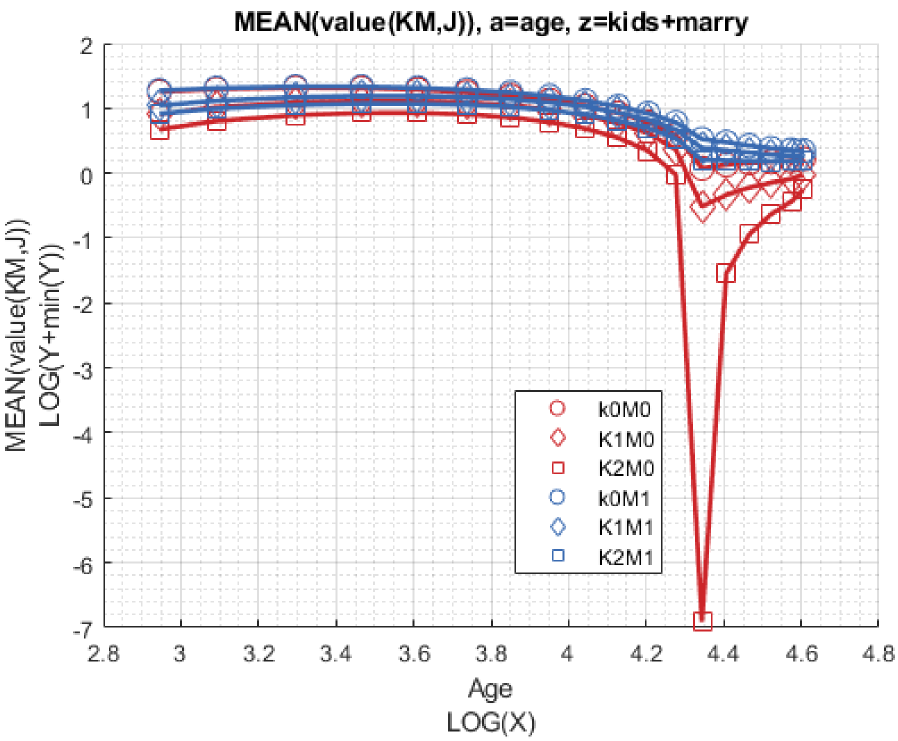
% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(C(KM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permute);

```
xxx  MEAN(C(KM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_3
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0         6.778         7.0757         7.3697         7.6652         7.9826
      2        2       0        7.1055         7.4611         7.8334         8.2212         8.6487
      3        3       0        7.5174         7.8216         8.1497         8.5069           8.91
      4        1       1        7.3063         7.6329         7.9639         8.2916          8.638
      5        2       1        7.3917         7.7297         8.0937         8.4696         8.8812
      6        3       1        7.7233         8.0217         8.3477         8.6932         9.0766
```

Graph Mean Values:

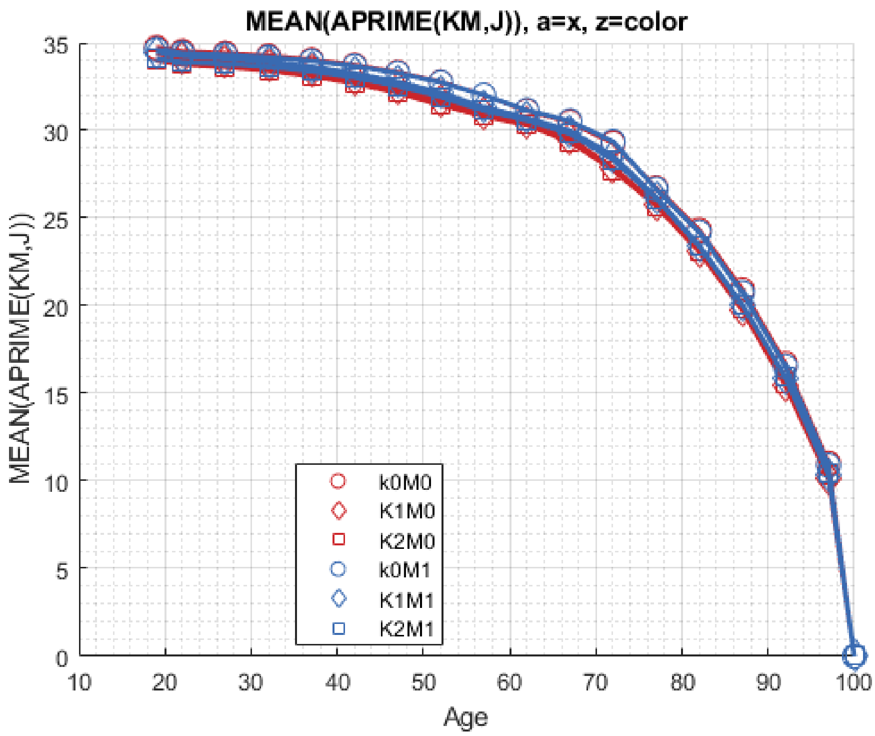
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(KM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

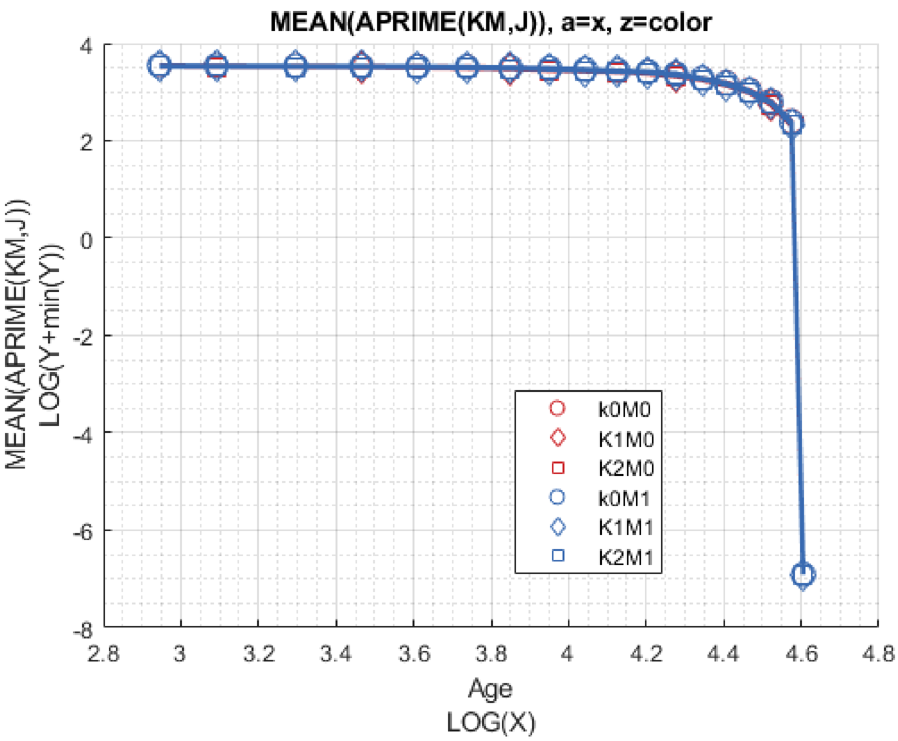




Graph Mean Savings Choices:

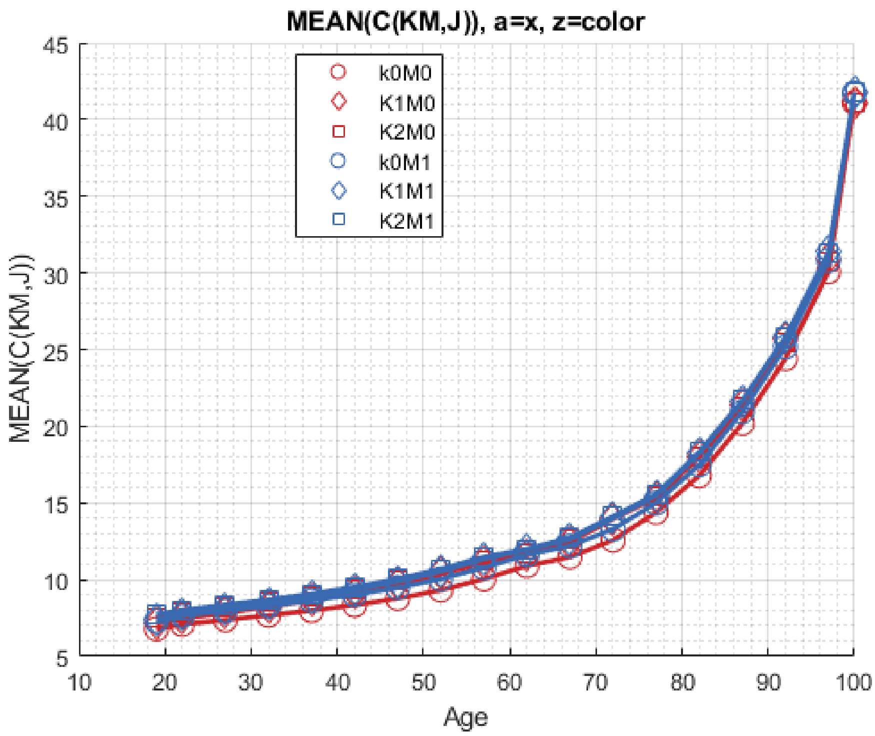
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(KM,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

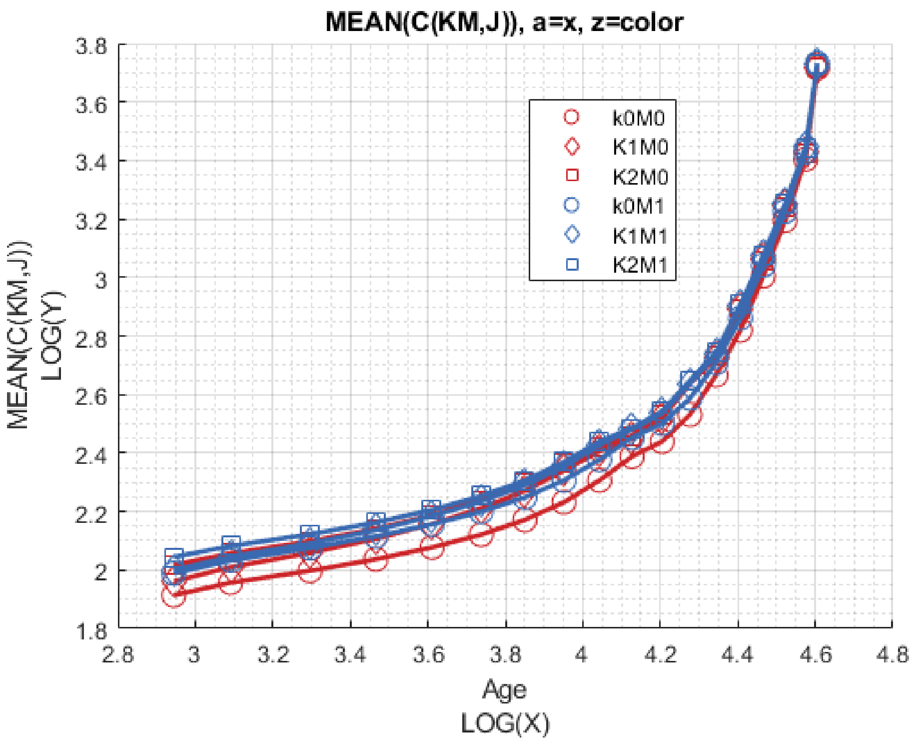




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(KM,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





3.4.5 Analyze Education 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 = ["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'};
```

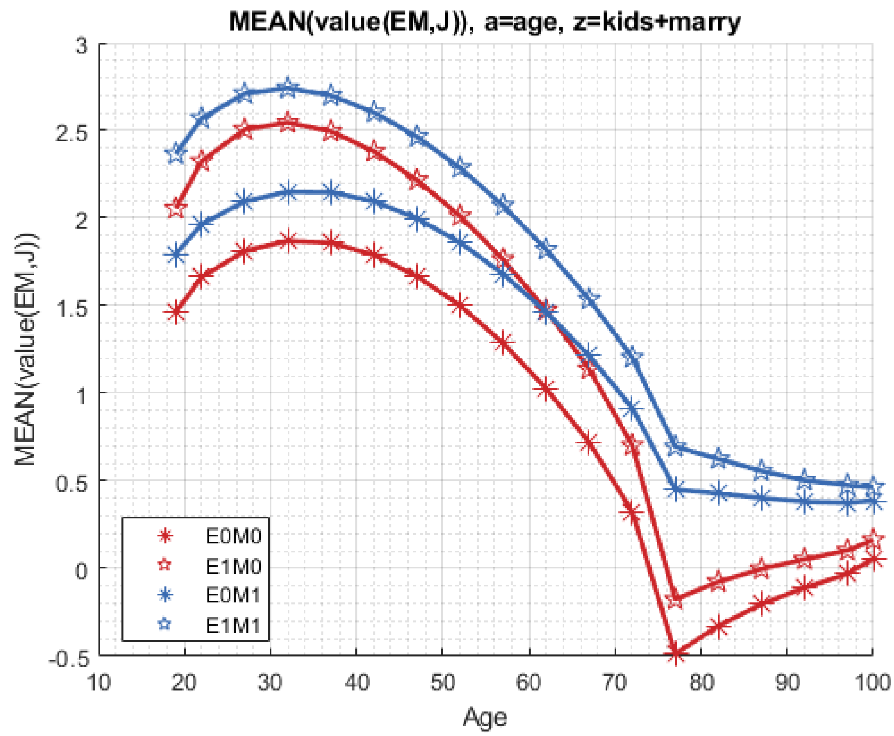
MEAN(VAL(EKM,J)), MEAN(AP(EKM,J)), MEAN(C(EKM,J))

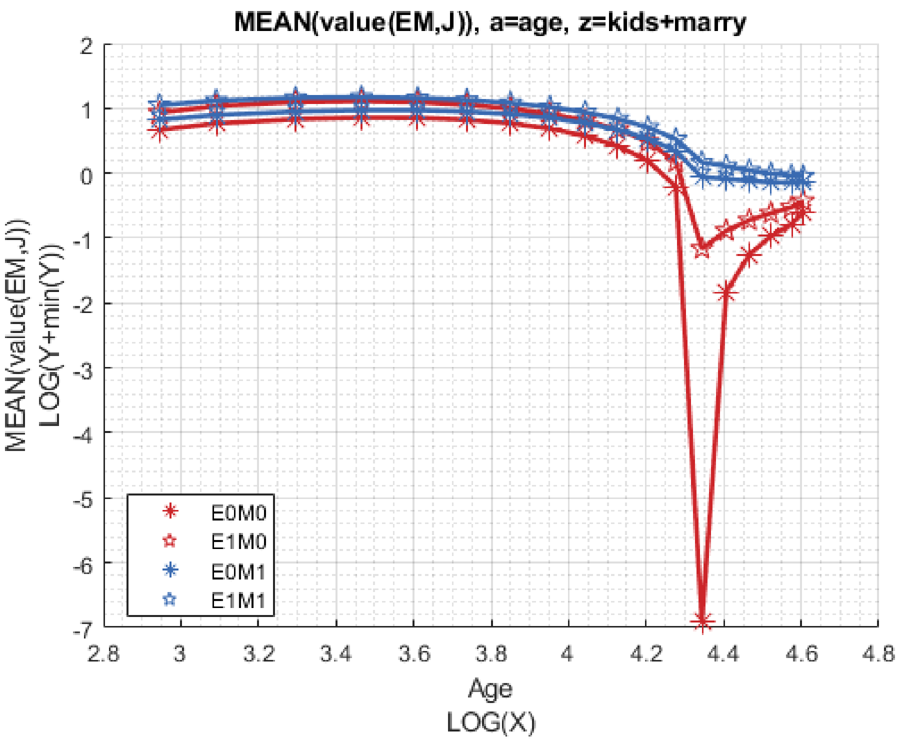
Tabulate value and policies:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(EKM,J))", V_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_pe

xxx  MEAN(VAL(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        1.4646         1.6636         1.8129         1.8698         1.8591
      2       1       0        2.0551         2.3251         2.5065         2.5458         2.4958
      3       0       1         1.795         1.9657         2.0955         2.1504         2.1481
      4       1       1        2.3659         2.5697         2.7107         2.7421         2.7001

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(EKM,J))", ap_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(AP(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        34.471         34.227         34.028         33.781         33.465
      2       1       0        34.298         34.065         33.974         33.817         33.557
      3       0       1        34.499         34.288         34.123          33.91         33.632
      4       1       1        34.304         34.102         34.042         33.923         33.703

% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(EKM,J))", cons_VFI, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_p

xxx  MEAN(C(EKM,J))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_19    mean_age_22    mean_age_27    mean_age_32    mean_age_37
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0         7.047         7.3391          7.647          7.982         8.3628
      2       1       0        7.2203         7.5665         7.9215         8.2802         8.6647
      3       0       1        7.3401         7.6271         7.9338         8.2659         8.6401
      4       1       1        7.6075         7.9624         8.3364         8.7038         9.0904
```

Graph Mean Values:

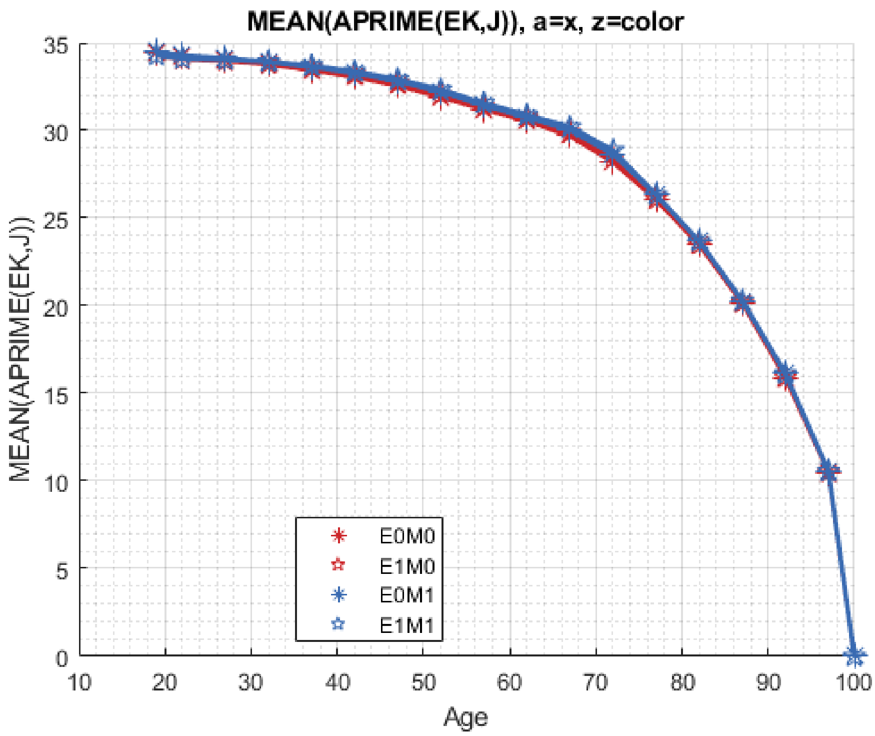
```
mp_support_graph('cl_st_graph_title') = {'MEAN(value(EM,J)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(value(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

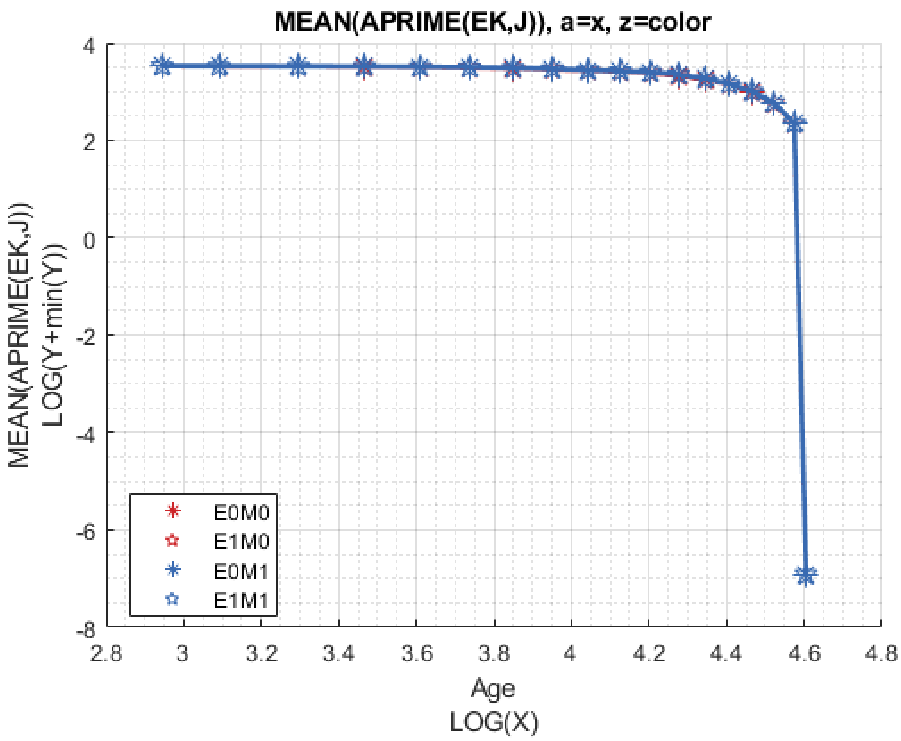




Graph Mean Savings Choices:

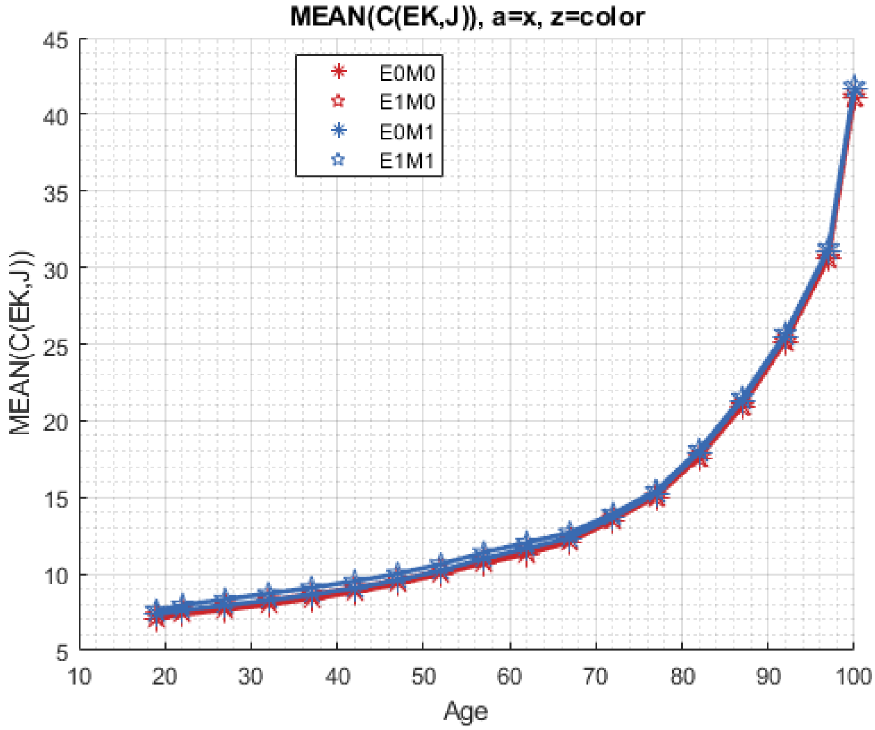
```
mp_support_graph('cl_st_graph_title') = {'MEAN(APRIME(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(APRIME(EK,J))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

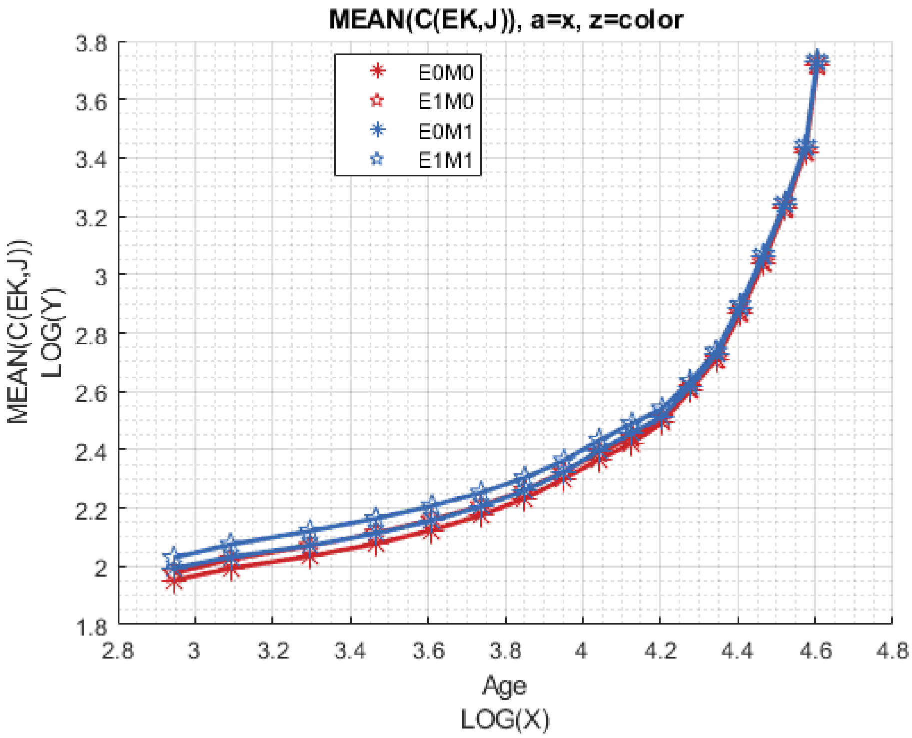




Graph Mean Consumption:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(EK,J)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(EK,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```


MEAN(C(EK,J)), a=x, z=color
E0M0
E1M0
E0M1
E1M1
MEAN(C(EK,J))
Age
45
40
35
30
25
20
15
10
5
10
20
30
40
50
60
70
80
90
100


MEAN(C(EK,J)), a=x, z=color
E0M0
E1M0
E0M1
E1M1
MEAN(C(EK,J))
LOG(Y)
Age
LOG(X)
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
4.8

Chapter 4

Solution with Unemployment

4.1 Life Cycle Dynamic Programming under Unemployment Shock

This is the example vignette for function: **snw_vfi_main_bisec_vec** from the **PrjOptiSNW Package.** This function solves for policy function using Exact Vectorized Solution. Dense Solution Analysis. Unemployment Shock. The file focuses on the change in value function, asset choice, and consumption choice given a one period unemployment shock (that does not reappear in the future again).

4.1.1 Test SNW_VFI_UNEMP Defaults Dense

Solve the Regular Value and Also the Unemployment Value.

First, solve for value without unemployment issue (use the vectorized code that was previously tested):

```
mp_params = snw_mp_param('default_docdense');
mp_controls = snw_mp_control('default_test');
[V_VFI_ss,ap_VFI_ss,cons_VFI_ss,mp_valpol_more_ss] = ...
    snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:83 of 82, time-this-age:1.8205
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:82 of 82, time-this-age:3.1788
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:81 of 82, time-this-age:3.0847
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:80 of 82, time-this-age:3.0421
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:79 of 82, time-this-age:3.0229
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:78 of 82, time-this-age:3.075
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:77 of 82, time-this-age:3.0089
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:76 of 82, time-this-age:3.0223
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:75 of 82, time-this-age:3.0628
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:74 of 82, time-this-age:2.9162
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:73 of 82, time-this-age:2.8128
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:72 of 82, time-this-age:2.8287
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:71 of 82, time-this-age:2.8302
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:70 of 82, time-this-age:2.8056
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:69 of 82, time-this-age:2.7875
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:68 of 82, time-this-age:2.9285
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:67 of 82, time-this-age:2.9005
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:66 of 82, time-this-age:2.8196
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:65 of 82, time-this-age:2.791
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:64 of 82, time-this-age:2.8039
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:63 of 82, time-this-age:2.7649
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:62 of 82, time-this-age:2.8592
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:61 of 82, time-this-age:2.7595
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:60 of 82, time-this-age:2.7984
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:59 of 82, time-this-age:2.8238
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:58 of 82, time-this-age:2.8372
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:57 of 82, time-this-age:2.847
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:56 of 82, time-this-age:2.8287
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:55 of 82, time-this-age:2.8489
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:54 of 82, time-this-age:2.8663
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:53 of 82, time-this-age:2.8091
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:52 of 82, time-this-age:2.8397
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:51 of 82, time-this-age:2.8433
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:50 of 82, time-this-age:2.8445
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:49 of 82, time-this-age:2.7143
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:48 of 82, time-this-age:2.6729
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:47 of 82, time-this-age:2.9111
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:46 of 82, time-this-age:2.8268
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:45 of 82, time-this-age:2.8303
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:44 of 82, time-this-age:2.8012
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:43 of 82, time-this-age:2.7892
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:42 of 82, time-this-age:2.8242
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:41 of 82, time-this-age:2.9846
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:40 of 82, time-this-age:2.8347
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:39 of 82, time-this-age:2.8005
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:38 of 82, time-this-age:2.7936
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:37 of 82, time-this-age:2.8617
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:36 of 82, time-this-age:2.8021
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:35 of 82, time-this-age:2.821
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:34 of 82, time-this-age:2.8082
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:33 of 82, time-this-age:2.7967
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:32 of 82, time-this-age:2.8976
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:31 of 82, time-this-age:2.8975
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:30 of 82, time-this-age:3.0965
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:29 of 82, time-this-age:3.169
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:28 of 82, time-this-age:3.148
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:27 of 82, time-this-age:3.1235
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:26 of 82, time-this-age:3.1654
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:25 of 82, time-this-age:3.0846
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:24 of 82, time-this-age:3.2605
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:23 of 82, time-this-age:2.8673
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:22 of 82, time-this-age:2.8366
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:21 of 82, time-this-age:2.9026
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:20 of 82, time-this-age:2.8865
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:19 of 82, time-this-age:2.8479
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:18 of 82, time-this-age:2.8406
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:17 of 82, time-this-age:2.9797
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:16 of 82, time-this-age:3.2286
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:15 of 82, time-this-age:3.0905
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:14 of 82, time-this-age:2.9062
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:13 of 82, time-this-age:2.8862
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:12 of 82, time-this-age:2.948
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:11 of 82, time-this-age:2.9553
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:10 of 82, time-this-age:2.9076
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:9 of 82, time-this-age:2.9098
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:8 of 82, time-this-age:2.8502
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:7 of 82, time-this-age:2.8649
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:6 of 82, time-this-age:2.8968
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:5 of 82, time-this-age:2.8606
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:4 of 82, time-this-age:2.835
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:3 of 82, time-this-age:2.8988
```

```
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:2 of 82, time-this-age:2.8594
SNW_VFI_MAIN_BISEC_VEC: Finished Age Group:1 of 82, time-this-age:2.867
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=240.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN           sum         mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -1.5339e+08    -3.5101    26.11
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.4159e+09     32.402    36.79
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05     2.1402e+08     4.8975    8.329

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5       c526496    c526497    c526498    c
           -------    -------    -------    -------    -------    -------    -------    -------    -

    r1     -346.51    -346.12    -343.63    -337.86    -328.51     21.702     21.852     22.003
    r2     -334.38    -333.99    -331.51    -325.83    -316.83     21.724     21.869     22.015
    r3     -322.45    -322.06     -319.6    -314.14     -305.6     21.745     21.885     22.027
    r4     -310.63    -310.27    -307.99    -302.88    -294.87     21.767     21.903     22.041
    r5     -299.94     -299.6    -297.46    -292.67    -285.12     21.775     21.907     22.042
    r79    -9.9437    -9.9325    -9.8557    -9.6597    -9.3232     2.5394     2.5501     2.5602
    r80    -8.9023    -8.8911    -8.8143    -8.6183    -8.2818     2.3039     2.3121     2.3198
    r81    -7.6363    -7.6251    -7.5484    -7.3524    -7.0159     2.0068     2.0124     2.0176
    r82    -5.9673    -5.9561    -5.8793    -5.6833    -5.3468     1.5958     1.5989     1.6018
    r83    -3.5892     -3.578    -3.5012    -3.3052    -2.9687    0.97904    0.98004    0.98097    0

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2        c3            c4           c5        c526496    c526497    c526498    c5264
           --    --    ----------    ---------    ---------    -------    -------    -------    -----

    r1     0     0      0.0005656    0.0075134    0.022901     114.75     120.41     126.27     132.3
    r2     0     0     0.00051498    0.0065334    0.021549     114.86     120.53     126.41     132.5
    r3     0     0     0.00051498    0.0049294    0.019875     114.97     120.65     126.56      132.
    r4     0     0     0.00051498    0.0047937    0.019672     115.73     121.42     127.34     133.5
    r5     0     0     0.00048517    0.0046683    0.019484      116.5     122.21     128.15     134.3
    r79    0     0              0            0           0     81.091      85.68     90.335     94.37
    r80    0     0              0            0           0     76.669     80.563     84.304      88.0
    r81    0     0              0            0           0     68.313     71.534     74.475     77.83
    r82    0     0              0            0           0     50.126     53.467     56.953     58.74
    r83    0     0              0            0           0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.036717    0.037251    0.040426     0.04363    0.048012     9.6491      9.817     9.9649
    r2     0.036717    0.037251    0.040477     0.04461    0.049364     9.8118     9.9685     10.101
    r3     0.036717    0.037251    0.040477    0.046214    0.051039     9.9779      10.12     10.234
    r4     0.038144    0.038678    0.041903    0.047776    0.052666     10.131     10.258     10.354
    r5     0.039534    0.040068    0.043323     0.04929    0.054241     10.272     10.384     10.463
    r79      0.2179     0.21844     0.22216     0.23228     0.25197     35.858     37.092     38.455
    r80      0.2179     0.21844     0.22216     0.23228     0.25197     40.253     42.183     44.459
    r81      0.2179     0.21844     0.22216     0.23228     0.25197     48.587      51.19     54.266
    r82      0.2179     0.21844     0.22216     0.23228     0.25197     66.755     69.238      71.77
    r83      0.2179     0.21844     0.22216     0.23228     0.25197     116.87     122.69     128.71
```

Second, solve for the unemployment value, use the exact-bisec result code, call the snw_vfi_main_bisec_vec.m function with a third input of existing value:

```
mp_params('xi') = 0.5;
mp_params('b') = 0;
[V_VFI_unemp,ap_VFI_unemp,cons_VFI_unemp,mp_valpol_more_unemp] = ...
    snw_vfi_main_bisec_vec(mp_params, mp_controls, V_VFI_ss);
```

```
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 1 of 82, time-this-age:2.8563
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 2 of 82, time-this-age:3.0648
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 3 of 82, time-this-age:2.7929
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 4 of 82, time-this-age:2.8051
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 5 of 82, time-this-age:2.8264
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 6 of 82, time-this-age:2.836
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 7 of 82, time-this-age:2.8277
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 8 of 82, time-this-age:2.7941
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 9 of 82, time-this-age:2.8232
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 10 of 82, time-this-age:2.7723
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 11 of 82, time-this-age:2.8291
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 12 of 82, time-this-age:2.7968
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 13 of 82, time-this-age:2.848
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 14 of 82, time-this-age:2.8093
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 15 of 82, time-this-age:2.8312
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 16 of 82, time-this-age:2.8063
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 17 of 82, time-this-age:2.8041
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 18 of 82, time-this-age:2.8034
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 19 of 82, time-this-age:2.796
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 20 of 82, time-this-age:2.8926
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 21 of 82, time-this-age:3.4005
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 22 of 82, time-this-age:3.0663
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 23 of 82, time-this-age:2.839
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 24 of 82, time-this-age:3.0613
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 25 of 82, time-this-age:3.1482
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 26 of 82, time-this-age:2.9432
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 27 of 82, time-this-age:2.964
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 28 of 82, time-this-age:2.9991
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 29 of 82, time-this-age:2.7389
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 30 of 82, time-this-age:2.8259
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 31 of 82, time-this-age:2.7929
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 32 of 82, time-this-age:2.7872
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 33 of 82, time-this-age:2.8341
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 34 of 82, time-this-age:2.9876
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 35 of 82, time-this-age:3.0836
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 36 of 82, time-this-age:2.7955
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 37 of 82, time-this-age:2.9634
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 38 of 82, time-this-age:2.7022
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 39 of 82, time-this-age:2.6388
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 40 of 82, time-this-age:2.6754
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 41 of 82, time-this-age:2.6157
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 42 of 82, time-this-age:2.7739
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 43 of 82, time-this-age:3.0513
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 44 of 82, time-this-age:2.6043
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 45 of 82, time-this-age:2.6203
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 46 of 82, time-this-age:2.9099
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 47 of 82, time-this-age:2.8328
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 48 of 82, time-this-age:2.5949
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 49 of 82, time-this-age:2.5623
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 50 of 82, time-this-age:2.4733
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 51 of 82, time-this-age:2.4966
```

```
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 52 of 82, time-this-age:2.4712
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 53 of 82, time-this-age:2.5194
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 54 of 82, time-this-age:2.5973
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 55 of 82, time-this-age:2.6167
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 56 of 82, time-this-age:2.8954
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 57 of 82, time-this-age:2.6303
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 58 of 82, time-this-age:2.5062
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 59 of 82, time-this-age:2.8318
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 60 of 82, time-this-age:3.1076
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 61 of 82, time-this-age:2.4185
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 62 of 82, time-this-age:2.5062
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 63 of 82, time-this-age:3.0883
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 64 of 82, time-this-age:2.715
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 65 of 82, time-this-age:2.5774
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 66 of 82, time-this-age:2.6634
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 67 of 82, time-this-age:2.4816
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 68 of 82, time-this-age:2.4953
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 69 of 82, time-this-age:2.7169
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 70 of 82, time-this-age:2.5446
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 71 of 82, time-this-age:2.6734
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 72 of 82, time-this-age:2.5802
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 73 of 82, time-this-age:2.5602
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 74 of 82, time-this-age:2.5524
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 75 of 82, time-this-age:2.5331
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 76 of 82, time-this-age:2.5831
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 77 of 82, time-this-age:2.5422
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 78 of 82, time-this-age:2.5236
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 79 of 82, time-this-age:2.5017
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 80 of 82, time-this-age:2.4577
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 81 of 82, time-this-age:2.7372
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 82 of 82, time-this-age:2.71
SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock: Age 83 of 82, time-this-age:1.6049
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=d
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN         sum        mean       std
                -    ---    ----    --------    ----    ---------   -----------   -------    -----

    V_VFI       1     1      6      4.37e+07     83     5.265e+05   -1.7805e+08   -4.0743    27.11
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05    1.3789e+09    31.553    36.67
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05    2.1097e+08    4.8277    8.328

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5        c526496    c526497    c526498    c
           -------    -------    -------    -------    -------    -------    -------    -------    -

    r1     -372.97    -371.47    -362.94    -349.52    -336.96     21.573     21.728     21.882
    r2     -360.84    -359.34    -350.81    -337.39    -324.98     21.595     21.745     21.894
    r3     -348.91    -347.41    -338.88    -325.46    -313.34     21.617     21.762     21.906
    r4     -336.09     -334.7    -326.73    -314.01    -302.44     21.633     21.772     21.913
    r5     -324.48    -323.18    -315.72    -303.62    -292.54     21.634      21.77     21.907
    r79    -9.9437    -9.9325    -9.8557    -9.6597    -9.3232     2.5374     2.5482     2.5584
    r80    -8.9023    -8.8911    -8.8143    -8.6183    -8.2818     2.3024     2.3107     2.3185
    r81    -7.6363    -7.6251    -7.5484    -7.3524    -7.0159     2.0057     2.0114     2.0168
    r82    -5.9673    -5.9561    -5.8793    -5.6833    -5.3468     1.5952     1.5984     1.6014
    r83    -3.5892     -3.578    -3.5012    -3.3052    -2.9687    0.97886    0.97987    0.98082    0
```

```
xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2    c3    c4       c5        c526496    c526497    c526498    c526499    c526500
           --    --    --    --    ---------    -------    -------    -------    -------    -------

    r1     0     0     0     0     0.0092181    110.06     115.71     121.55     127.62     133.93
    r2     0     0     0     0      0.008238    110.03     115.68     121.54     127.62     133.95
    r3     0     0     0     0     0.0066341    109.99     115.65     121.53     127.63     133.97
    r4     0     0     0     0     0.0058019    110.28     115.95     121.84     127.96     134.33
    r5     0     0     0     0      0.004998    110.58     116.27     122.17     128.31     134.69
    r79    0     0     0     0             0    81.091     85.229     89.297     93.341     97.382
    r80    0     0     0     0             0    75.865     79.539      83.28     87.016     90.669
    r81    0     0     0     0             0    67.781     70.521     73.462     76.819     81.091
    r82    0     0     0     0             0    50.126     53.467     56.108     57.742     60.587
    r83    0     0     0     0             0         0          0          0          0          0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5        c526496    c526497    c526498
           --------    --------    --------    --------    --------    -------    -------    -------

    r1     0.018623    0.019158    0.022901    0.033062     0.04363    9.4708     9.6491      9.817
    r2     0.018623    0.019158    0.022901    0.033062     0.04461    9.6414     9.8118     9.9685
    r3     0.018623    0.019158    0.022901    0.033062    0.046214    9.8179     9.9779      10.12
    r4     0.019354    0.019888    0.023632    0.033792    0.047776    9.9825     10.131     10.258
    r5     0.020066    0.020601    0.024344    0.034504     0.04929    10.135     10.272     10.384
    r79      0.2179     0.21844     0.22216     0.23228     0.25197     34.82     36.506     38.455
    r80      0.2179     0.21844     0.22216     0.23228     0.25197    40.033     42.183     44.459
    r81      0.2179     0.21844     0.22216     0.23228     0.25197    48.106      51.19     54.266
    r82      0.2179     0.21844     0.22216     0.23228     0.25197    65.751     68.234     71.611
    r83      0.2179     0.21844     0.22216     0.23228     0.25197    115.87     121.69     127.71
```

Difference Between Value and Choices In Unemployment and Future Periods

```
V_VFI_unemp_drop = V_VFI_ss - V_VFI_unemp;
ap_VFI_unemp_drop = ap_VFI_ss - ap_VFI_unemp;
cons_VFI_unemp_drop = cons_VFI_ss - cons_VFI_unemp;
```

4.1.2 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=%3.2
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});
```

4.1.3 Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% 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') = 15; % how many shock legends to show
mp_support_graph('cl_colors') = 'jet';
```

MEAN(VAL(A,Z) - VAL(A,Z|unemp)), MEAN(AP(A,Z) - AP(A,Z|unemp)), MEAN(C(A,Z) - C(A,Z|unemp))

Tabulate value and policies along savings and shocks:

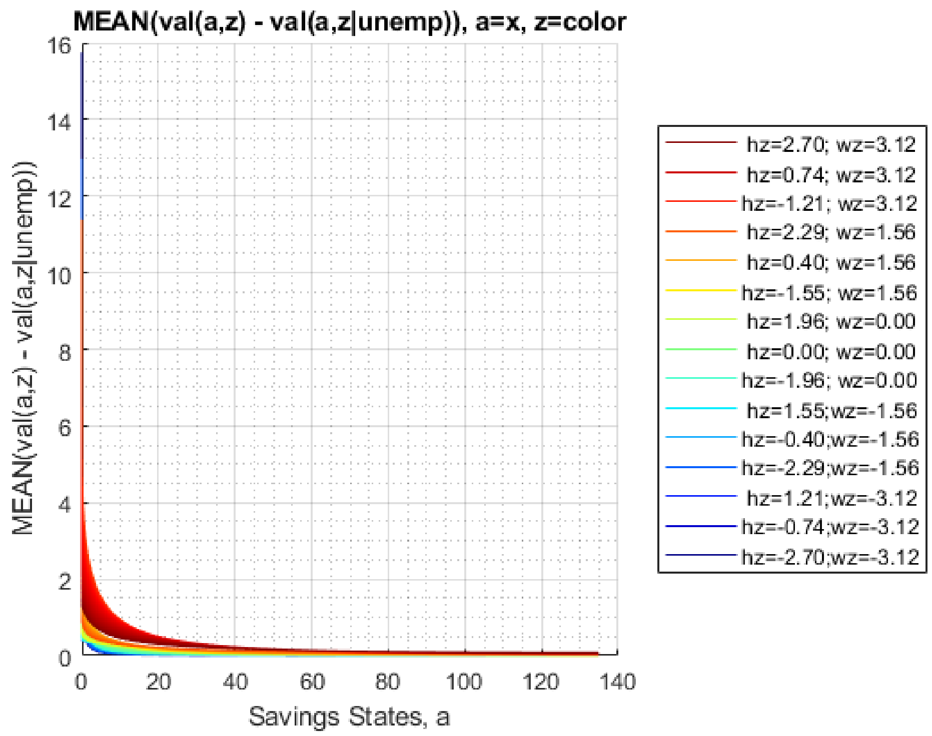
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(v(A,Z) - v(A,Z|unemp))", V_VFI_unemp_drop, true, ["mean"], 4, 1, cl
```

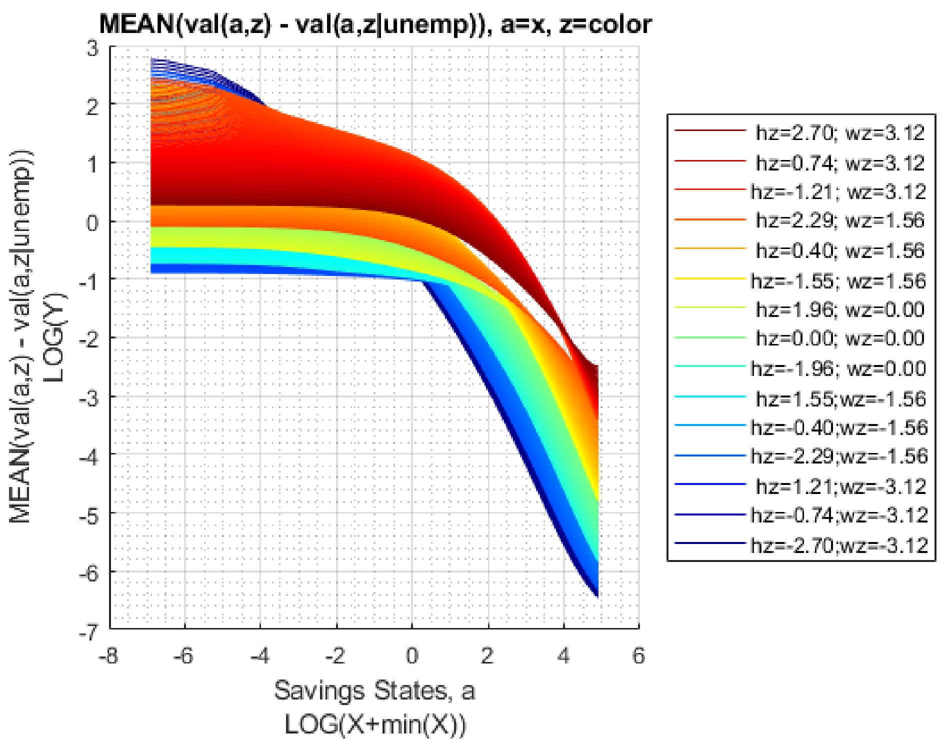
```
xxx  MEAN(v(A,Z) - v(A,Z|unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings       mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5    mea
    -----    ----------    ----------     ----------     ----------     ----------     ----------    ---

      1            0         15.753         14.805         13.912         13.072         12.281
```

```
xxx  MEAN(C(A,Z) - C(A,Z|unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings       mean_eta_1     mean_eta_2     mean_eta_3     mean_eta_4     mean_eta_5    mea
    -----    ----------    ----------     ----------     ----------     ----------     ----------    ---

      1            0       0.019317       0.020449       0.021654       0.022935       0.024299
```

Graph Mean Values Change:

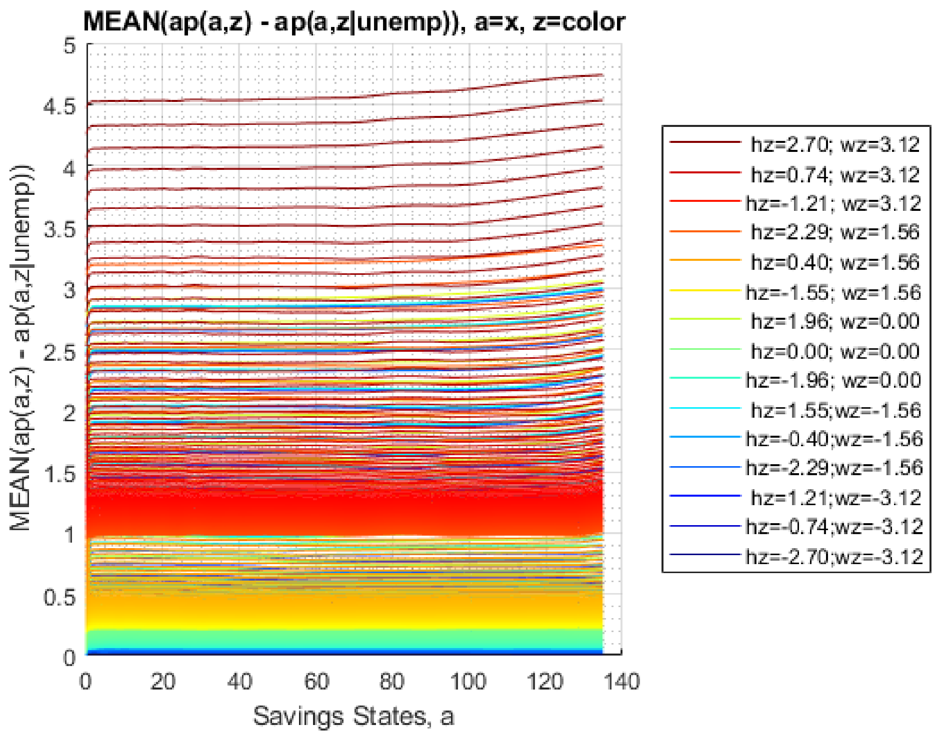
```
mp_support_graph('cl_st_graph_title') = {'MEAN(val(a,z) - val(a,z|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(val(a,z) - val(a,z|unemp))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

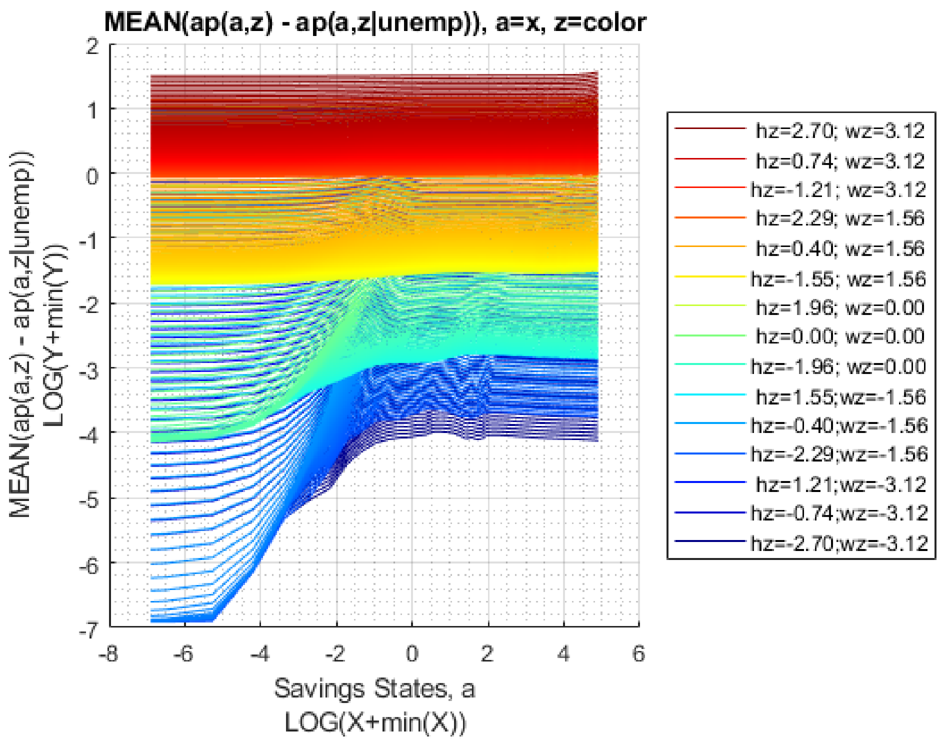




Graph Mean Savings Choices Change:

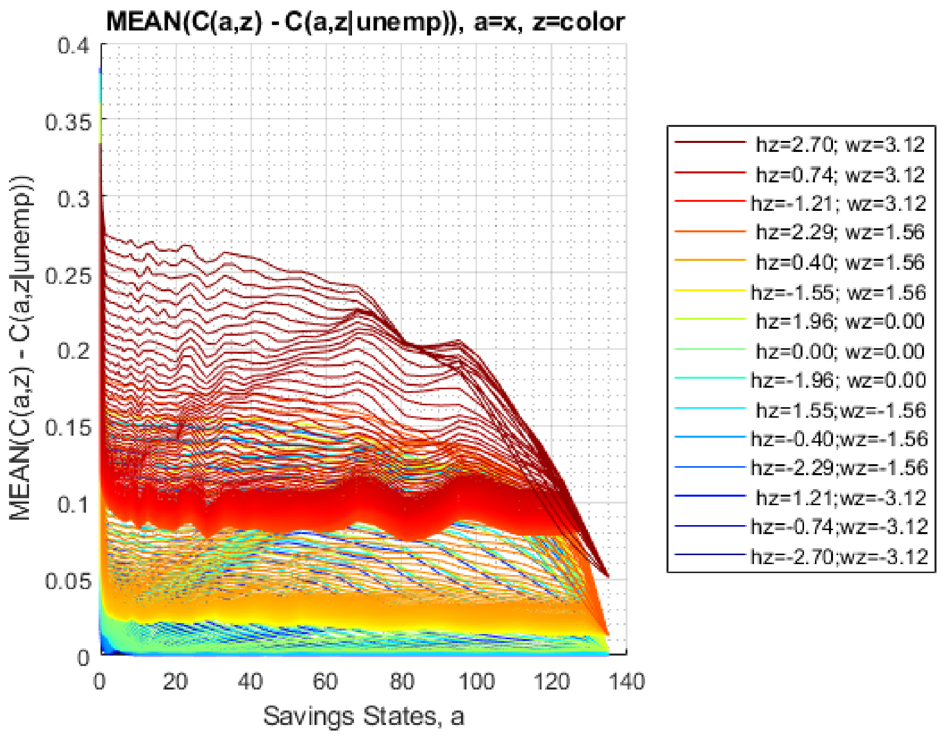
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(a,z) - ap(a,z|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(a,z) - ap(a,z|unemp))'};
ff_graph_grid((tb_az_ap{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

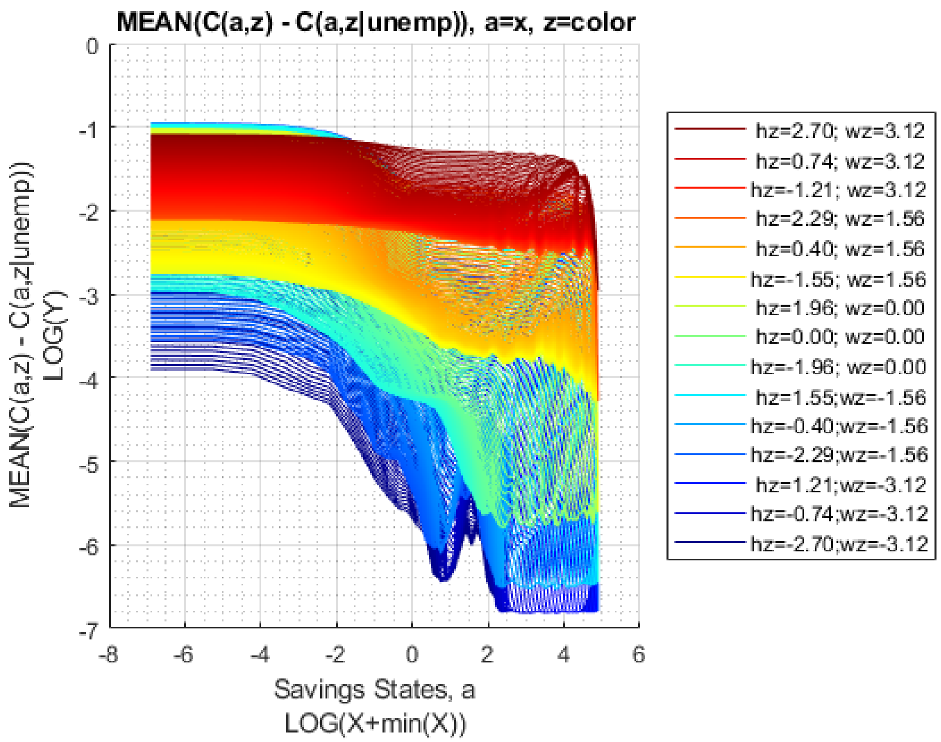




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(C(a,z) - C(a,z|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(C(a,z) - C(a,z|unemp))'};
ff_graph_grid((tb_az_c{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```





4.1.4 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(V(KM,J) - V(KM,J | unemp)), MEAN(ap(KM,J) - ap(KM,J | unemp)), MEAN(c(KM,J) - c(KM,J | unemp))

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
tb_az_v = ff_summ_nd_array("MEAN(V(KM,J) - V(KM,J | unemp))", V_VFI_unemp_drop, true, ["mean"], 3, 1

xxx  MEAN(V(KM,J) - V(KM,J | unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        0.61637        0.59885        0.58106        0.56498        0.55117
      2        2       0        0.82734        0.80489        0.78136        0.75704        0.73572
      3        3       0        0.96755        0.94502        0.92045        0.89136        0.86587
      4        4       0         1.0948         1.0713          1.045         1.0118         0.9827
      5        5       0         1.2011         1.1779          1.151         1.1149         1.0833
      6        1       1        0.76784        0.74924        0.73091        0.71544        0.70238
      7        2       1        0.93021        0.90698        0.88323        0.86203        0.84347
      8        3       1         1.0185         0.9941        0.96877        0.94495        0.92408
      9        4       1         1.1171         1.0915         1.0645         1.0382         1.0151
     10        5       1         1.1585         1.1346         1.1083         1.0807         1.0569

% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(ap(KM,J) - ap(KM,J | unemp))", ap_VFI_unemp_drop, true, ["mean"],

xxx  MEAN(ap(KM,J) - ap(KM,J | unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        0.54429        0.54157        0.53838        0.57688        0.61527
      2        2       0        0.53828        0.53451        0.53011        0.56791        0.60562
      3        3       0        0.53173        0.52734        0.52253        0.55991        0.59734
      4        4       0         0.5276          0.523        0.51797        0.55513        0.59235
      5        5       0        0.52354        0.51894        0.51381        0.55085        0.58805
      6        1       1         1.1323         1.1757         1.2198         1.3119         1.4048
      7        2       1         1.0396         1.0753         1.1115         1.1942         1.2777
      8        3       1        0.97097          1.002         1.0331         1.1097          1.187
      9        4       1        0.89591        0.92257        0.94909         1.0212         1.0937
     10        5       1        0.78017        0.79798        0.81575        0.87811        0.94079
```

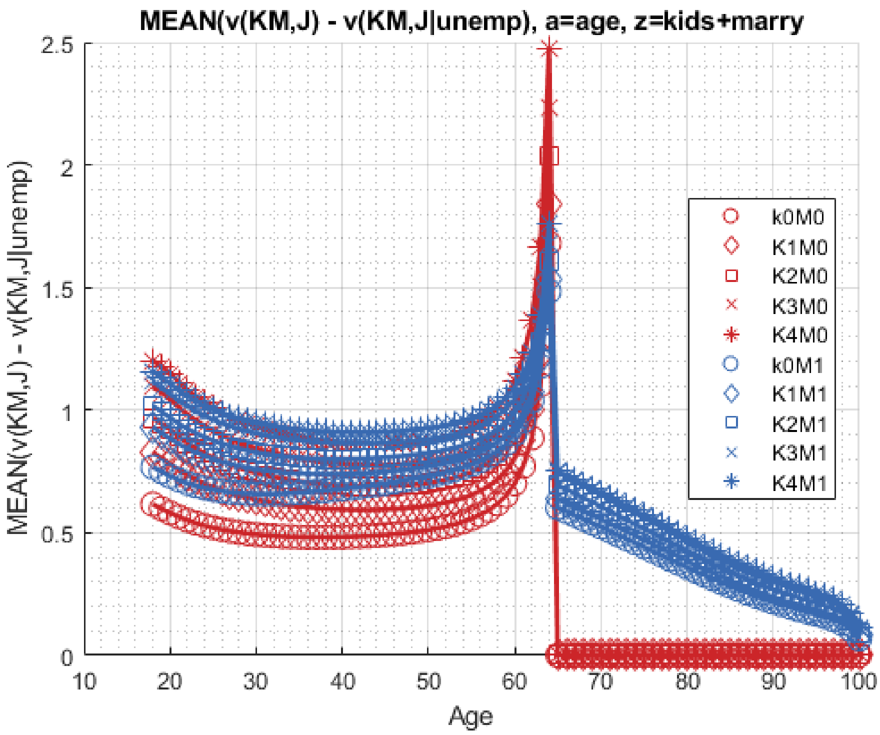
% Consumption Choices tb_az_c = ff_summ_nd_array("MEAN(c(KM,J) - c(KM,J | unemp))", cons_VFI_unemp_drop, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_permute);

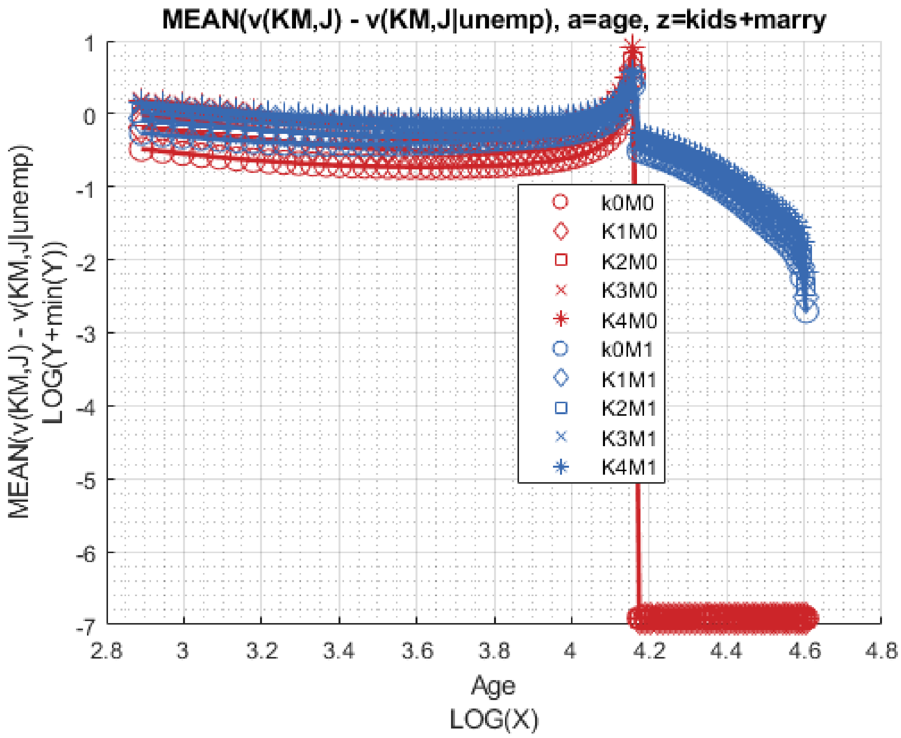
```
xxx  MEAN(c(KM,J) - c(KM,J | unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________
```

1	1	0	0.050084	0.052801	0.055995	0.056344	0.056497
2	2	0	0.056094	0.059866	0.064267	0.065317	0.06615
3	3	0	0.062643	0.067034	0.071841	0.073312	0.074434
4	4	0	0.06677	0.071371	0.076406	0.078097	0.079421
5	5	0	0.07083	0.075431	0.080561	0.082377	0.083719
6	1	1	0.091654	0.09722	0.1029	0.10693	0.11041
7	2	1	0.087426	0.093165	0.099035	0.10362	0.10765
8	3	1	0.089332	0.094467	0.10022	0.10478	0.10884
9	4	1	0.095488	0.099656	0.10451	0.10733	0.10981
10	5	1	0.1018	0.10631	0.11124	0.11381	0.11605

Graph Mean Values Change:

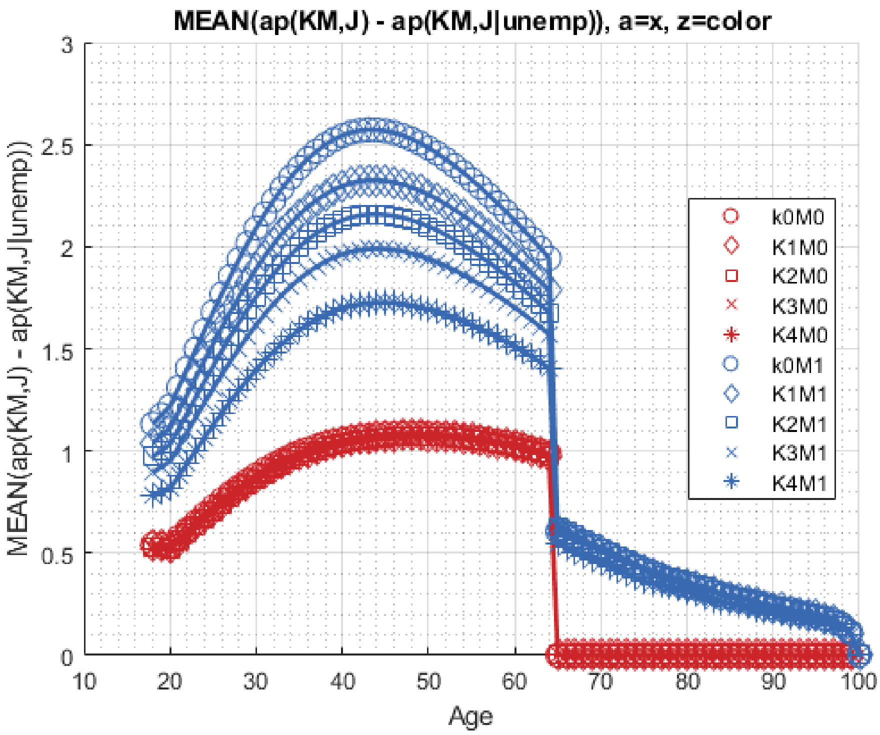
```
mp_support_graph('cl_st_graph_title') = {'MEAN(v(KM,J) - v(KM,J|unemp), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(v(KM,J) - v(KM,J|unemp)'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

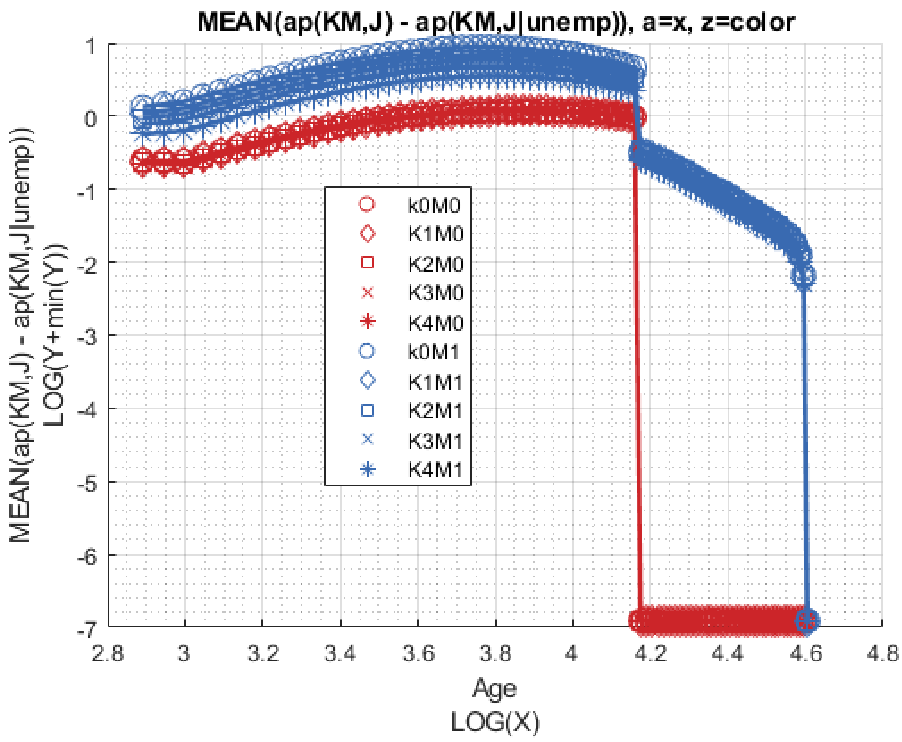




Graph Mean Savings Choices Change:

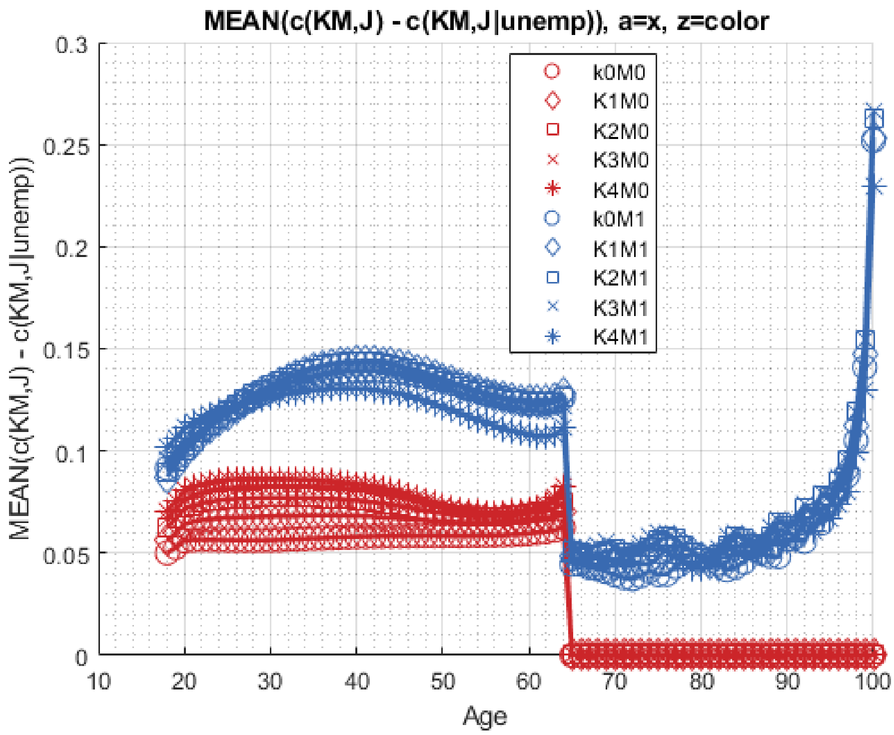
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(KM,J) - ap(KM,J|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(KM,J) - ap(KM,J|unemp))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

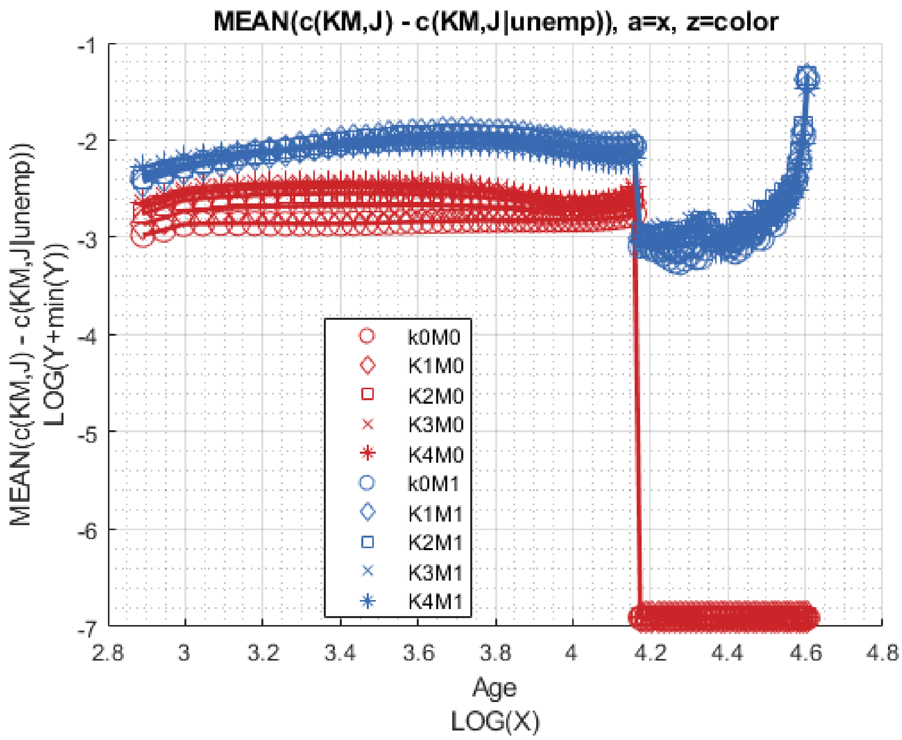




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(c(KM,J) - c(KM,J|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(c(KM,J) - c(KM,J|unemp))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





4.1.5 Analyze Education 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 = ["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(v(EKM,J) - v(EKM,J|unemp)), MEAN(ap(EM,J) - ap(EM,J|unemp)), MEAN(c(EM,J) - c(EM,J|unemp))

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
tb_az_v = ff_summ_nd_array("MEAN(v(EM,J) - v(EM,J|unemp))", V_VFI_unemp_drop, true, ["mean"], 3, 1,
```

```
xxx  MEAN(v(EM,J) - v(EM,J|unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        0.98303        0.96405        0.94385        0.92458        0.90689
      2       1       0        0.89982        0.87513        0.84768        0.81144        0.78062
      3       0       1         1.0503         1.0306         1.0104        0.99222        0.97585
      4       1       1        0.94657        0.91993        0.89191        0.86431        0.84092
```

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(ap(EM,J) - ap(EM,J|unemp))", ap_VFI_unemp_drop, true, ["mean"], 3,
```

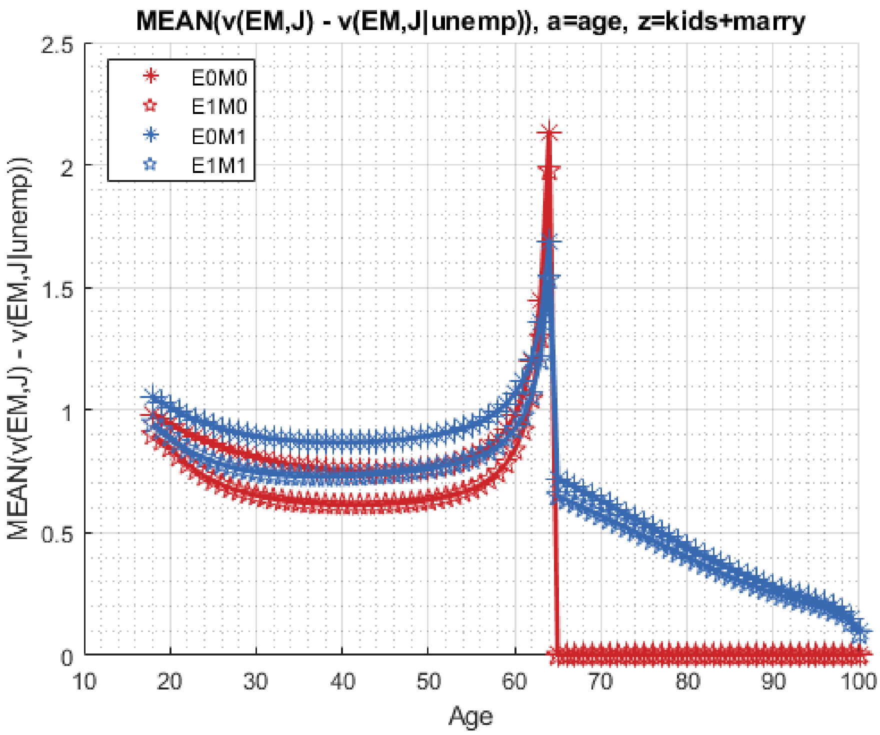
```
xxx  MEAN(ap(EM,J) - ap(EM,J|unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        0.54395        0.54191        0.53951        0.56214        0.58423
      2       1       0        0.52222        0.51623        0.50961        0.56213        0.61523
      3       0       1        0.93033        0.95904        0.98801         1.0446         1.1011
      4       1       1        0.99726         1.0304         1.0637         1.1614         1.2605
```

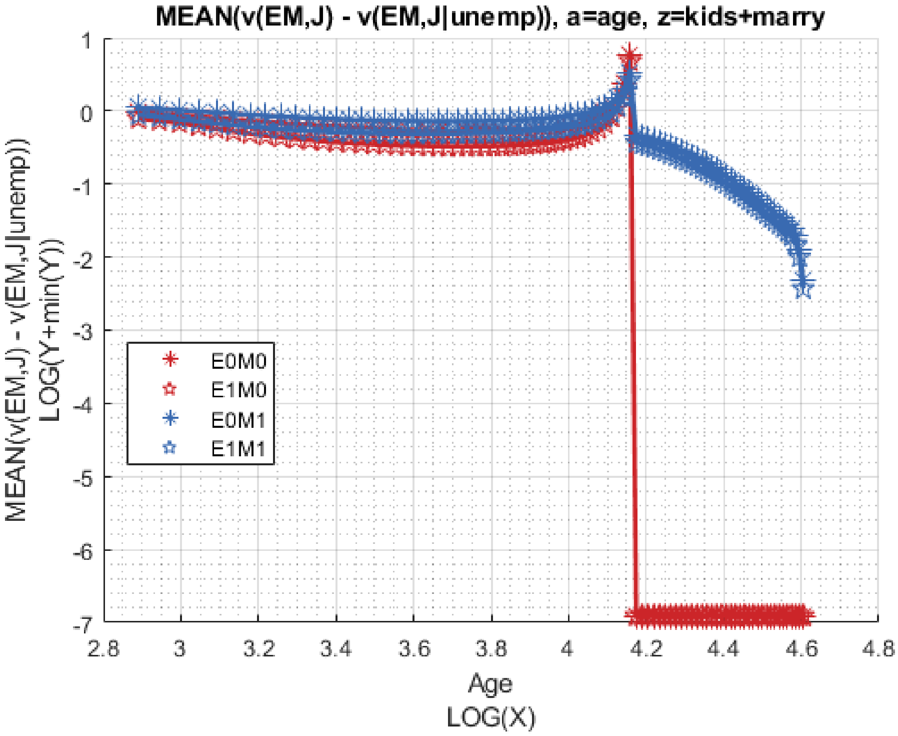
```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(c(EM,J) - c(EM,J|unemp))", cons_VFI_unemp_drop, true, ["mean"], 3,
```

```
xxx  MEAN(c(EM,J) - c(EM,J|unemp))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        0.05042       0.052463       0.054861       0.055684       0.056488
      2       1       0       0.072148       0.078138       0.084767       0.086495         0.0876
      3       0       1       0.079245       0.082789       0.086633       0.089336       0.091941
      4       1       1        0.10704        0.11354        0.12053        0.12525        0.12917
```

Graph Mean Values Change:

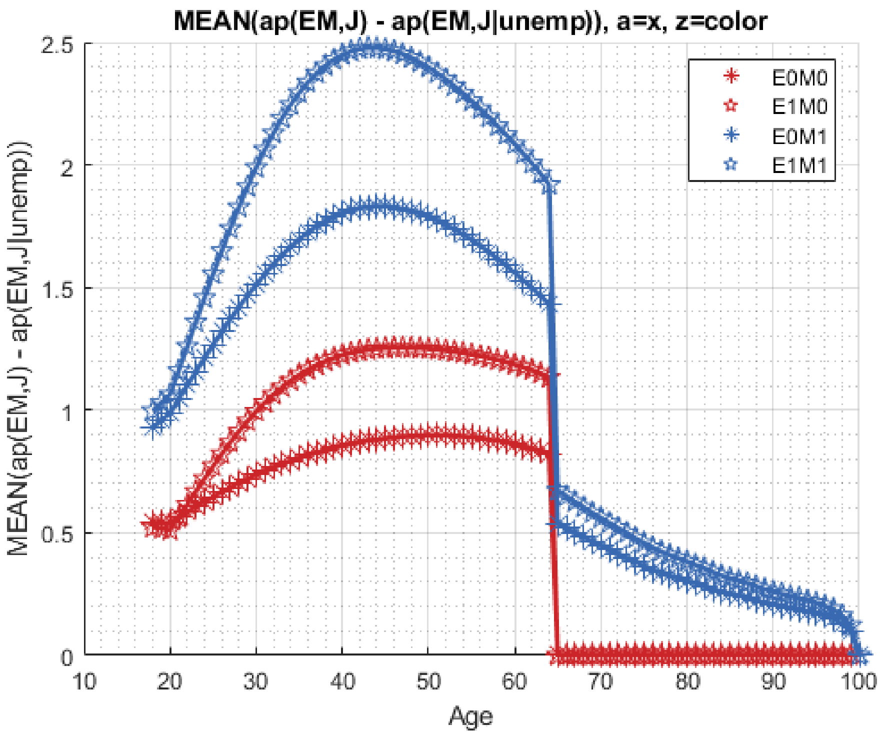
```
mp_support_graph('cl_st_graph_title') = {'MEAN(v(EM,J) - v(EM,J|unemp)), a=age, z=kids+marry'};
mp_support_graph('cl_st_ytitle') = {'MEAN(v(EM,J) - v(EM,J|unemp))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

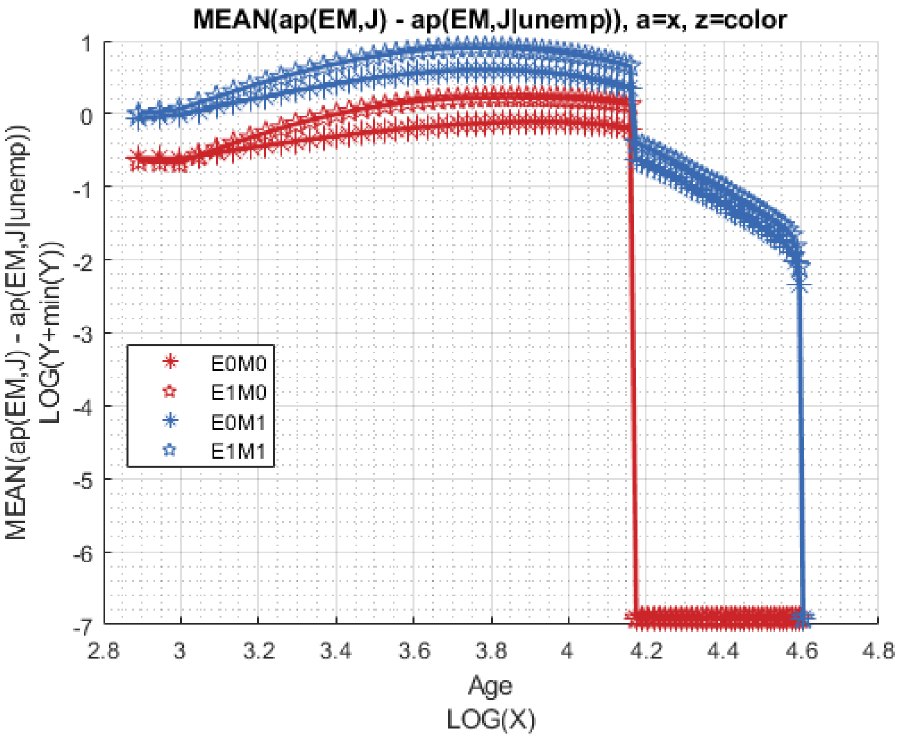




Graph Mean Savings Choices Change:

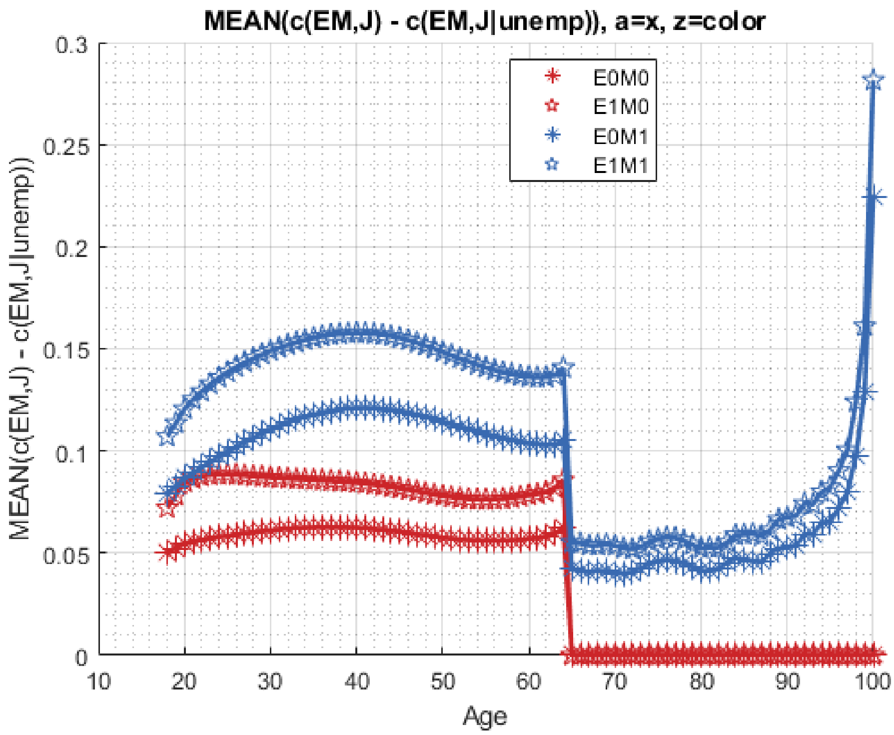
```
mp_support_graph('cl_st_graph_title') = {'MEAN(ap(EM,J) - ap(EM,J|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(ap(EM,J) - ap(EM,J|unemp))'};
ff_graph_grid((tb_az_ap{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```

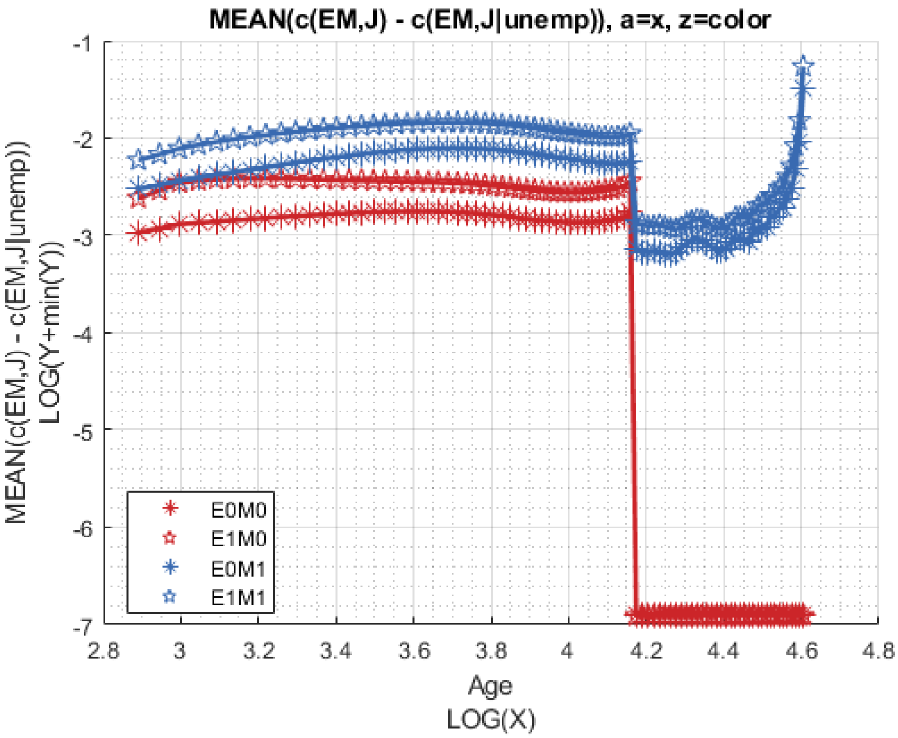




Graph Mean Consumption Change:

```
mp_support_graph('cl_st_graph_title') = {'MEAN(c(EM,J) - c(EM,J|unemp)), a=x, z=color'};
mp_support_graph('cl_st_ytitle') = {'MEAN(c(EM,J) - c(EM,J|unemp))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```




MEAN(c(EM,J) - c(EM,J|unemp)), a=x, z=color
MEAN(c(EM,J) - c(EM,J|unemp))
LOG(Y+min(Y))
-1
-2
-3
-4
-5
-6
-7
E0M0
E1M0
E0M1
E1M1
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
4.8
Age
LOG(X)

Chapter 5

Household Life Cycle Distribution

5.1 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.More Dense Simulation. **Looped** to get distribution, but uses **bisect vec** for VFI.

5.1.1 Test SNW_DS_MAIN Defaults Dense

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=253.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN     colN         sum          mean       std
                -    ---    ----    --------    ----    ---------    -----------    -------    -----

    V_VFI       1     1       6     4.37e+07     83     5.265e+05    -1.5339e+08    -3.5101    26.11
    ap_VFI      2     2       6     4.37e+07     83     5.265e+05     1.4159e+09     32.402    36.79
    cons_VFI    3     3       6     4.37e+07     83     5.265e+05     2.1402e+08     4.8975    8.329

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5       c526496    c526497    c526498    c
           -------    -------    -------    -------    -------    -------    -------    -------    -

    r1     -346.51    -346.12    -343.63    -337.86    -328.51    21.702     21.852     22.003
    r2     -334.38    -333.99    -331.51    -325.83    -316.83    21.724     21.869     22.015
    r3     -322.45    -322.06     -319.6    -314.14     -305.6    21.745     21.885     22.027
    r4     -310.63    -310.27    -307.99    -302.88    -294.87    21.767     21.903     22.041
    r5     -299.94     -299.6    -297.46    -292.67    -285.12    21.775     21.907     22.042
    r79    -9.9437    -9.9325    -9.8557    -9.6597    -9.3232    2.5394     2.5501     2.5602
    r80    -8.9023    -8.8911    -8.8143    -8.6183    -8.2818    2.3039     2.3121     2.3198
    r81    -7.6363    -7.6251    -7.5484    -7.3524    -7.0159    2.0068     2.0124     2.0176
    r82    -5.9673    -5.9561    -5.8793    -5.6833    -5.3468    1.5958     1.5989     1.6018
```

```
    r83    -3.5892     -3.578     -3.5012     -3.3052     -2.9687     0.97904     0.98004     0.98097     0

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1     c2        c3           c4           c5        c526496     c526497     c526498     c5264
           --     --     ----------    ---------    --------    -------     -------     -------     -----

    r1      0      0       0.0005656    0.0075134    0.022901    114.75      120.41      126.27      132.3
    r2      0      0      0.00051498    0.0065334    0.021549    114.86      120.53      126.41      132.5
    r3      0      0      0.00051498    0.0049294    0.019875    114.97      120.65      126.56       132.
    r4      0      0      0.00051498    0.0047937    0.019672    115.73      121.42      127.34      133.5
    r5      0      0      0.00048517    0.0046683    0.019484     116.5      122.21      128.15      134.3
    r79     0      0               0            0           0    81.091       85.68      90.335      94.37
    r80     0      0               0            0           0    76.669      80.563      84.304       88.0
    r81     0      0               0            0           0    68.313      71.534      74.475      77.83
    r82     0      0               0            0           0    50.126      53.467      56.953      58.74
    r83     0      0               0            0           0         0           0           0

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5        c526496     c526497     c526498
           --------    --------    --------    --------    --------    -------     -------     -------

    r1     0.036717    0.037251    0.040426     0.04363    0.048012     9.6491       9.817      9.9649
    r2     0.036717    0.037251    0.040477     0.04461    0.049364     9.8118      9.9685      10.101
    r3     0.036717    0.037251    0.040477    0.046214    0.051039     9.9779       10.12      10.234
    r4     0.038144    0.038678    0.041903    0.047776    0.052666     10.131      10.258      10.354
    r5     0.039534    0.040068    0.043323     0.04929    0.054241     10.272      10.384      10.463
    r79      0.2179     0.21844     0.22216     0.23228     0.25197     35.858      37.092      38.455
    r80      0.2179     0.21844     0.22216     0.23228     0.25197     40.253      42.183      44.459
    r81      0.2179     0.21844     0.22216     0.23228     0.25197     48.587       51.19      54.266
    r82      0.2179     0.21844     0.22216     0.23228     0.25197     66.755      69.238       71.77
    r83      0.2179     0.21844     0.22216     0.23228     0.25197     116.87      122.69      128.71

Completed SNW_DS_MAIN;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1804.8494
```

```
% [Phi_true,Phi_adj] = snw_ds_main(mp_params, mp_controls);
Phi_true = Phi_true/sum(Phi_true(:));
```

5.1.2 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
                      --------    -------    ---------    --------    -------    ---------    -------

    a_ss                4.2486     6.7963       1.5996           0        135       0.1223           0
    ap_ss               4.3473      6.834        1.572           0      163.7      0.10225           0
    cons_ss             1.0676    0.69454      0.65055    0.036717     141.66            0           0
    v_ss               -15.745      21.68      -1.3769     -586.22      24.63            0      0.8122
    n_ss                2.3554     1.4375      0.61029           1          6            0           0
    y_all                1.415     1.4926       1.0548           0     50.873    0.0072908           0
    y_head_inc          1.1087     1.0092      0.91029    0.038108     24.357            0           0
    y_head_earn        0.88655    0.92804       1.0468           0     18.957       0.2016           0
    y_spouse_inc       0.35849    0.95494       2.6638           0     26.627      0.52499           0
    yshr_interest      0.12214    0.16806       1.3759           0    0.99299       0.1223           0
    yshr_wage          0.77513    0.33759      0.43553           0          1      0.10584           0
    yshr_SS            0.10273    0.23637       2.3009           0          1       0.7984           0
    yshr_tax           0.17862    0.03519      0.19701    0.036506     0.2552            0           0
```

```
    yshr_nttxss       0.075896     0.25563       3.3681      -0.89184      0.2552             0    0.17845
```

5.1.3 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=%3.2
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});
```

5.1.4 Analyze Probability Mass Along Age Dimensions

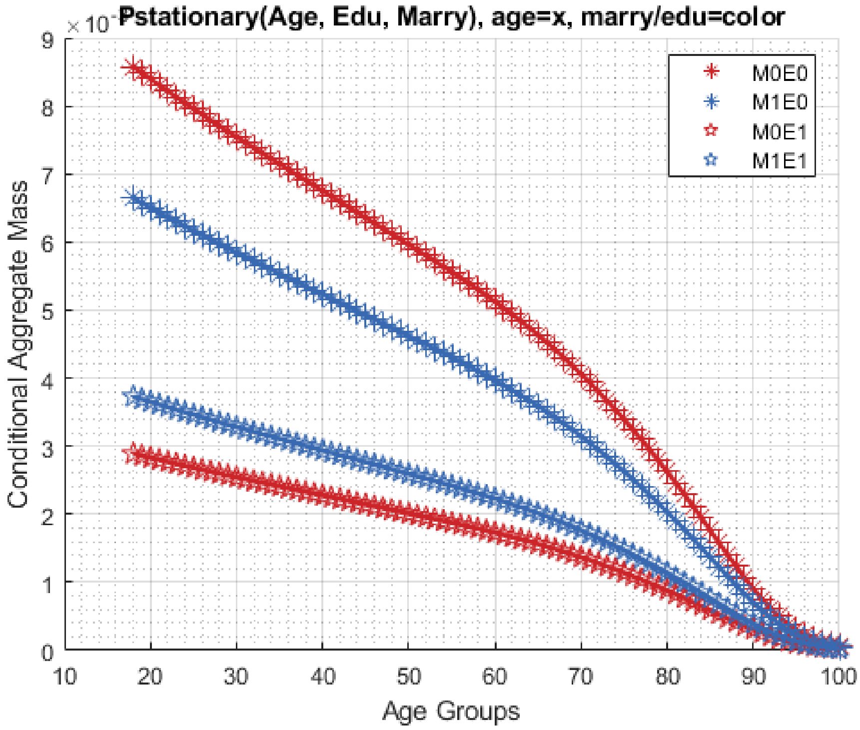
Where are the mass at? Analyze mass given state space components.

```
% 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:

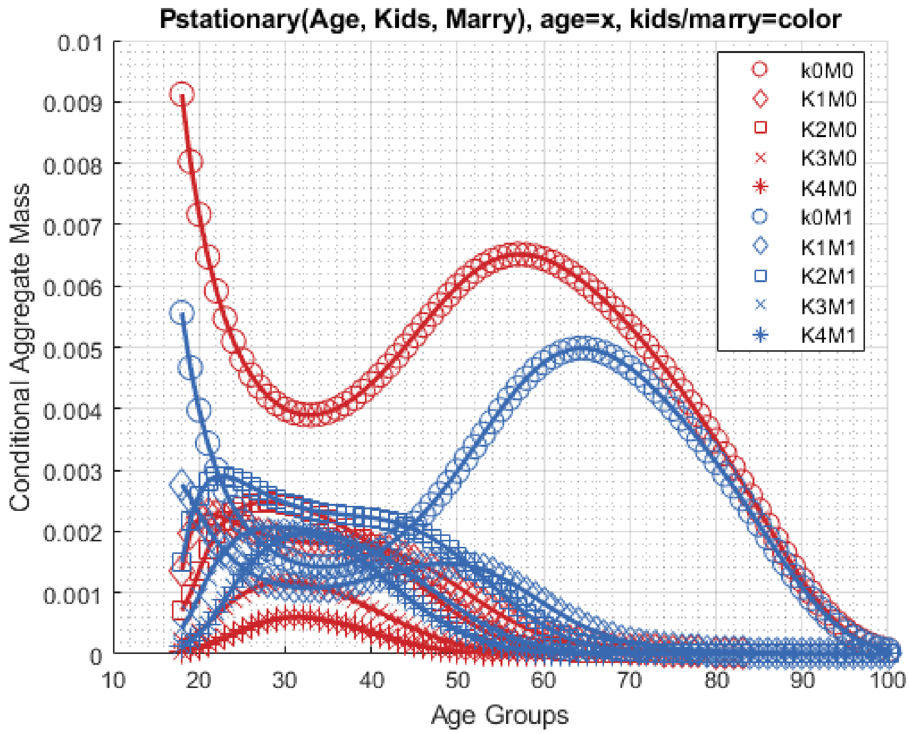
```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,5,4];
% Value Function
tb_prob_aem = ff_summ_nd_array("P(Age, EDU, MARRY))", Phi_true, true, ["sum"], 3, 1, cl_mp_datasetde

xxx  P(Age, EDU, MARRY))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    marry    edu    sum_age_18    sum_age_19    sum_age_20    sum_age_21    sum_age_22    s
    -----    -----    ---    ----------    ----------    ----------    ----------    ----------    -

      1        0       0     0.0085768     0.0084866     0.0083969     0.0083078     0.0082194     0
      2        1       0     0.0066438     0.0065739     0.0065044     0.0064354     0.0063669
      3        0       1     0.0028875     0.0028571      0.002827      0.002797     0.0027672     0
      4        1       1     0.0037292     0.0036899     0.0036509     0.0036122     0.0035738     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_mp_d

xxx  P(Age, Kids, Marry))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    sum_age_18    sum_age_19    sum_age_20    sum_age_21    sum_age_22
    -----    ----    -----    ----------    ----------    ----------    ----------    ----------

      1        1       0       0.0091249     0.0080278     0.0071652     0.0064765     0.0059205
      2        2       0       0.0013699     0.0019743     0.0022187     0.0022858     0.0022687
      3        3       0      0.00071266    0.00098425     0.0013537     0.0016929     0.0019639
      4        4       0      0.00020622    0.00027865    0.00037326    0.00049476    0.00062818
      5        5       0      5.0761e-05    7.8715e-05      0.000113    0.00015485    0.00020534
      6        1       1       0.0055624     0.0046679     0.0039774     0.0034368     0.0030088
      7        2       1       0.0027682     0.0025539     0.0023005     0.0020611     0.0018525
      8        3       1       0.0014982     0.0021823     0.0025943     0.0028096      0.002896
      9        4       1      0.00041197    0.00064648    0.00095224     0.0012491     0.0015009
     10        5       1      0.00013221     0.0002132    0.00033097    0.00049097    0.00068255

mp_support_graph('cl_st_graph_title') = {'Pstationary(Age, Kids, Marry), age=x, kids/marry=color'};
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);
```



5.1.5 Analyze Probability Mass Asset and Shock Dimensions

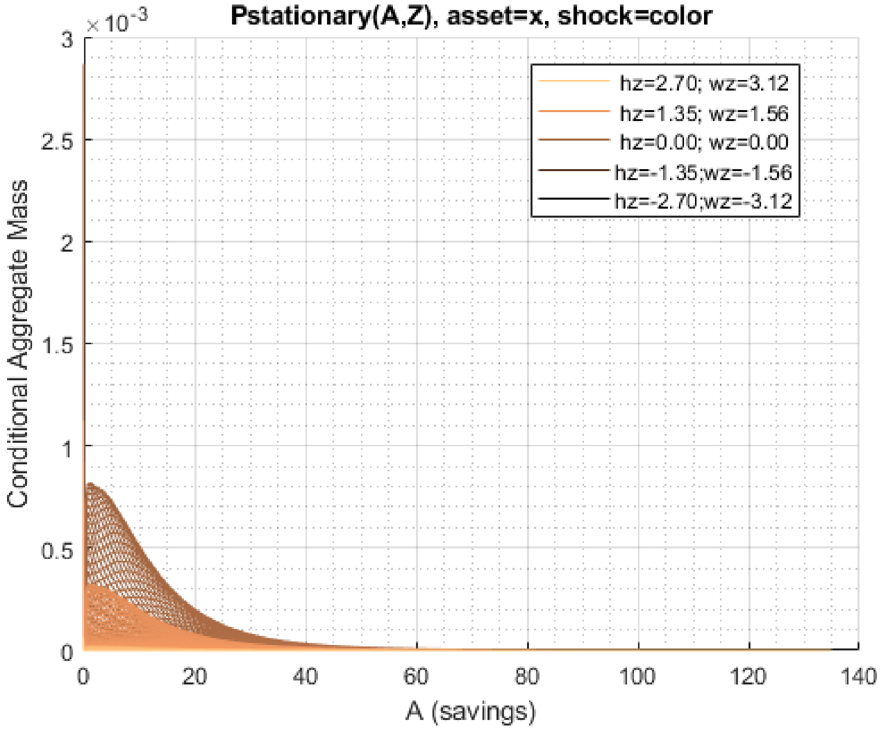
Where are the mass at?

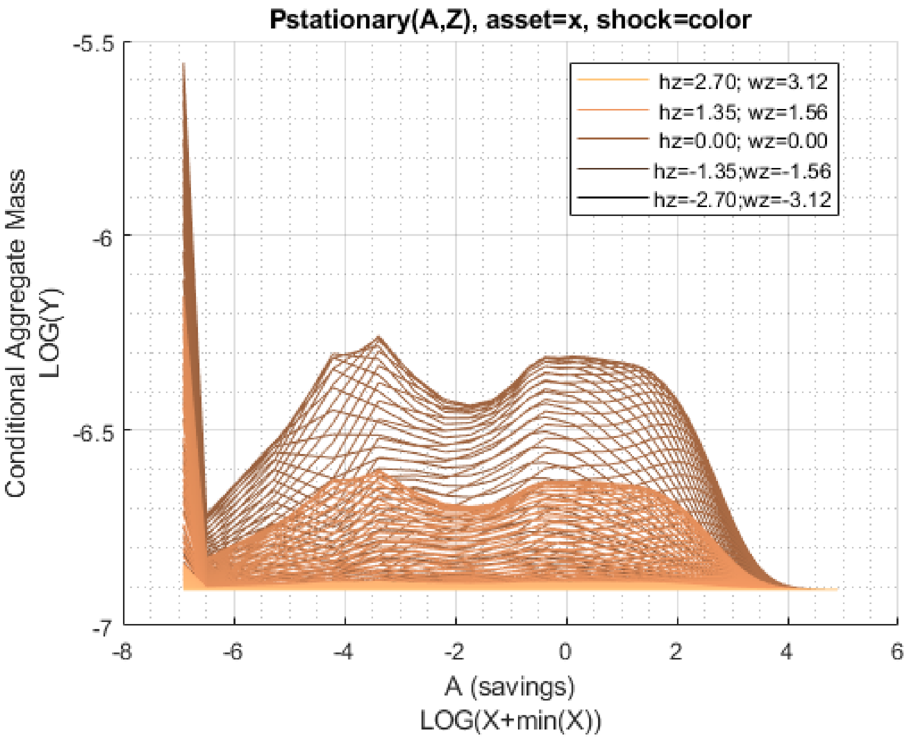
```
% Define Graph Inputs
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = false; % do not log
```

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_permut

xxx  P(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings       sum_eta_1     sum_eta_2     sum_eta_3     sum_eta_4     sum_eta_5     sum
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1               0    1.6824e-07    1.4406e-07    2.1911e-07    3.1913e-07    4.5491e-07    6.4
      2      0.00051498    3.4279e-10    3.2632e-10    5.6501e-10    1.0203e-09    1.9975e-09    4.1
      3       0.0041199    7.1369e-10    6.2373e-10    9.7246e-10    1.4702e-09    2.2039e-09    3.2
      4        0.013905     1.573e-09    1.3633e-09    2.1044e-09    3.1331e-09    4.6025e-09    6.7
      5        0.032959     5.494e-09    4.7235e-09      7.23e-09    1.0641e-08    1.5401e-08     2.
      6        0.064373    6.5788e-09    5.6779e-09     8.702e-09    1.2804e-08    1.8492e-08    2.6

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);% Consumption
```





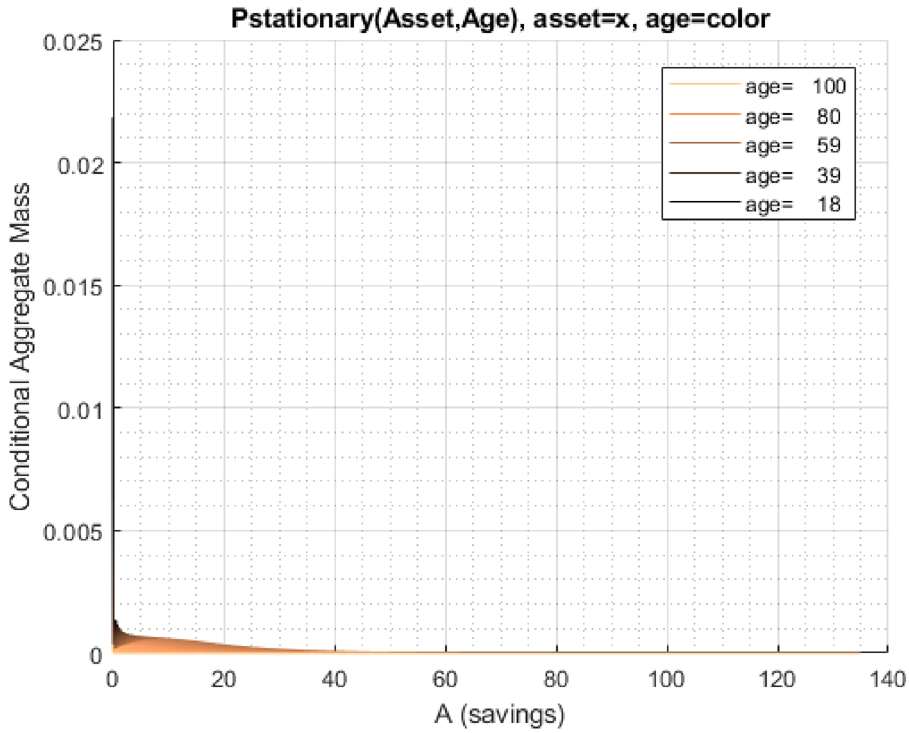
Asset Mass by Age

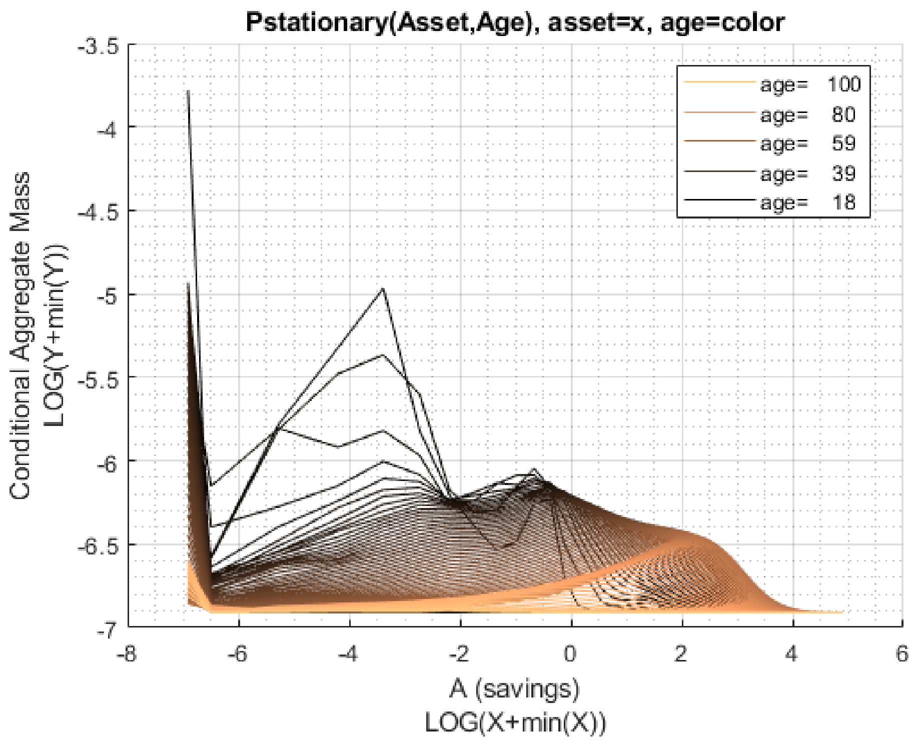
```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [3,4,5,6,1,2];
% Value Function
tb_prob_aage = ff_summ_nd_array("P(A,Z))", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar_perm
```

```
xxx  P(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      sum_age_18    sum_age_19    sum_age_20    sum_age_21    sum_age_22    sum
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0      0.021837     0.0023507     0.0017993     0.0039371     0.0058435     0.
      2     0.00051498             0    0.00039608    0.00037932     0.0011301    0.00066626     0.0
      3      0.0041199             0     0.0020816     0.0019888      0.002009    0.00088325     0.0
      4       0.013905             0     0.0038656     0.0031682      0.001688     0.0011334     0.0
      5       0.032959             0     0.0059678     0.0036757     0.0019686     0.0014691     0.
      6       0.064373             0      0.001968     0.0026857     0.0015598     0.0012805     0.
      7        0.11124             0     0.0010155     0.0010772    0.00089495    0.00094737     0.0
      8        0.17664             0    0.00066497    0.00081578     0.0009608     0.0010548     0.
      9        0.26367             0    0.00045021    0.00085579     0.0011593     0.0011712     0.
     10        0.37542             0    0.00053095     0.0011218     0.0012745     0.0011467     0.
     11        0.51498             0    0.00090691     0.0013663     0.0012758     0.0012278     0
     12        0.68544             0    0.00097523      0.001111     0.0010957     0.0011325     0.
     13        0.88989             0    0.00023441    0.00050314    0.00074645     0.0009432     0.
     14         1.1314             0    4.5279e-05    0.00027467    0.00049029    0.00060869     0.0
     15         1.4131             0    1.7339e-05    0.00019476    0.00030104    0.00040391     0.0
     16         1.7381             0    8.1464e-06    6.6555e-05    0.00014925    0.00025602     0.0
     17         2.1094             0    6.1188e-06    3.5994e-05    9.5417e-05      0.000162     0.0
     18         2.5301             0    1.3448e-05    3.7101e-05    7.3464e-05    0.00012006     0.0
     19         3.0034             0    2.2537e-05    4.8195e-05    7.7883e-05    0.00011025     0.0
     20         3.5323             0    2.9909e-05    5.5599e-05    8.0928e-05    0.00010452     0.0
     21         4.1199             0    3.0433e-05     5.458e-05    7.2693e-05    9.1664e-05     0.0
     22         4.7693             0    2.0391e-05    3.7793e-05    5.5429e-05    7.2296e-05     8.9
     23         5.4836             0    5.1199e-06    1.8361e-05     3.277e-05    4.8259e-05     6.4
     24         6.2658             0    7.2528e-07    5.2955e-06    1.4093e-05    2.6887e-05     4.
     25         7.1191             0    1.0524e-07    1.2817e-06    4.9228e-06    1.2149e-05     2.2
     26         8.0466             0    1.7628e-08    5.0295e-07    2.0294e-06    5.2782e-06     1.1
     27         9.0514             0    3.0056e-09    3.0395e-07    1.0911e-06    2.7755e-06     5.7
     28         10.136             0    1.1825e-10    1.6421e-07    5.5086e-07    1.5801e-06     3.2
     29         11.305             0             0    4.8037e-08    2.2122e-07    8.0726e-07     1.8
     30          12.56             0             0    9.2865e-09    6.9448e-08    3.1086e-07     1.0
     31         13.905             0             0     1.789e-09     2.077e-08    9.8086e-08     4.7
     32         15.342             0             0    4.0984e-10    6.2012e-09    3.4485e-08     1.8
     33         16.875             0             0    9.8855e-11    1.6718e-09    1.2956e-08     6.
     34         18.507             0             0    2.1171e-11    4.7002e-10    4.2475e-09     2.1
     35         20.241             0             0    8.4937e-13    1.3772e-10    1.2013e-09     8.
     36          22.08             0             0             0    2.9206e-11     3.623e-10     2.7
     37         24.027             0             0             0    3.6378e-12    1.1269e-10     8.3
     38         26.085             0             0             0    7.7367e-13    2.3608e-11     2.7
     39         28.258             0             0             0    1.7753e-13    3.9993e-12     8.0
     40         30.548             0             0             0    8.3602e-15    1.0518e-12     1.7
     41         32.959             0             0             0             0    1.9415e-13     3.6
     42         35.493             0             0             0             0    1.4615e-14     9.1
     43         38.154             0             0             0             0    2.3455e-15     1.4
     44         40.945             0             0             0             0    2.9499e-16     1.7
     45         43.868             0             0             0             0    6.0398e-18     3.2
     46         46.928             0             0             0             0             0     3.3
     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 Choice
```





5.1.6 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_of_s
    end
end

age =18
xxx tb_outcomes: all stats xxx
    OriginalVariableNames      ap_ss          c_ss          v_ss           n_ss        y_head_inc
    ______________________    __________    __________    ___________    ___________    ___________
```

```
    {'mean'               }      0.13166       0.63405        -31.11        1.9854       0.71265
    {'sd'                 }      0.34823       0.37905        29.813        1.0848       0.54567
    {'coefofvar'          }        2.645       0.59783      -0.95831       0.54639       0.76569
    {'min'                }            0      0.036717       -586.22             1      0.038108
    {'max'                }       145.07        10.212         24.63             6        13.784
    {'pYis0'              }      0.10805             0             0             0             0
    {'pYls0'              }            0             0       0.93414             0             0
    {'pYgr0'              }      0.89195             1      0.065859             1             1
    {'pYisMINY'           }      0.10805    1.3288e-05    5.8837e-08       0.41786    2.5312e-05
    {'pYisMAXY'           }            0             0             0     0.0060544             0
    {'p0_01'              }            0      0.047727       -322.58             1      0.046651
    {'p10'                }            0       0.24819       -67.491             1       0.23528
    {'p25'                }     0.012186       0.36957       -41.871             1       0.35258
    {'p50'                }     0.032959       0.55272       -24.354             2       0.56523
    {'p75'                }      0.07477       0.80089        -11.18             3       0.90612
    {'p90'                }      0.47812        1.1198       -2.6906             4        1.3579
    {'p99_99'             }       5.4504        3.6593        17.393             6        6.8484
    {'fl_cov_ap_ss'       }      0.12126      0.055072        2.4507      0.026881          0.05
    {'fl_cor_ap_ss'       }            1       0.41721       0.23606      0.071158       0.26313
    {'fl_cov_c_ss'        }     0.055072       0.14368        8.0391       0.07643       0.18689
    {'fl_cor_c_ss'        }      0.41721             1       0.71138       0.18587       0.90355
    {'fl_cov_v_ss'        }       2.4507        8.0391         888.8       0.38384        10.004
    {'fl_cor_v_ss'        }      0.23606       0.71138             1      0.011868       0.61498
    {'fl_cov_n_ss'        }     0.026881       0.07643       0.38384        1.1768    -8.185e-18
    {'fl_cor_n_ss'        }     0.071158       0.18587      0.011868             1   -1.3827e-17
    {'fl_cov_y_head_inc' }          0.05       0.18689        10.004    -8.185e-18       0.29776
    {'fl_cor_y_head_inc' }       0.26313       0.90355       0.61498   -1.3827e-17             1
    {'fl_cov_y_spouse'   }       0.18249      0.071644        3.4658       0.13323      0.010455
    {'fl_cor_y_spouse'   }       0.92021       0.33189       0.20413       0.21565      0.033645
    {'fl_cov_yshr_wage'  }    3.6882e-32    2.4079e-31   -9.3307e-30    1.6917e-30    1.2042e-31
    {'fl_cor_yshr_wage'  }    2.3849e-16    1.4304e-15   -7.0476e-16    3.5116e-15    4.9692e-16
    {'fl_cov_yshr_SS'    }             0             0             0             0             0
    {'fl_cor_yshr_SS'    }           NaN           NaN           NaN           NaN           NaN
    {'fl_cov_yshr_nttxss'}     0.0057457      0.011176       0.85848      0.007516       0.01319
    {'fl_cor_yshr_nttxss'}       0.48632       0.86907       0.84874       0.20421       0.71249
    {'fracByP0_01'       }             0    7.1684e-06     0.0013012       0.21046     7.788e-06
    {'fracByP10'         }             0      0.030643       0.32088       0.21046      0.027495
    {'fracByP25'         }     0.0067356       0.10365       0.58193       0.21046      0.092606
    {'fracByP50'         }       0.04689       0.29058       0.83099       0.53024       0.26377
    {'fracByP75'         }       0.13162       0.54875       0.97426       0.77109        0.5245
    {'fracByP90'         }       0.35822       0.76944        1.0077       0.92834       0.74403
    {'fracByP99_99'      }       0.99575       0.99938        1.0001             1       0.99912
age =59
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       ap_ss          c_ss          v_ss         n_ss       y_head_inc
    _____________________    __________    __________    __________    _________    __________    _

    {'mean'               }       9.4506        1.2067       -9.9431        1.7239        1.6033
    {'sd'                 }       9.4598       0.76797        14.834       0.90777        1.2742
    {'coefofvar'          }        1.001       0.63643       -1.4919       0.52659       0.79474
    {'min'                }            0       0.05663       -208.18             1      0.059541
    {'max'                }       158.43        12.311        14.965             6         23.47
    {'pYis0'              }    0.0059691             0             0             0             0
    {'pYls0'              }            0             0       0.73383             0             0
    {'pYgr0'              }      0.99403             1       0.26617             1             1
    {'pYisMINY'           }    0.0059691    9.8324e-06    2.9687e-09       0.48835    9.8989e-06
    {'pYisMAXY'           }   9.0457e-09    3.8325e-11    5.2662e-07     0.0036816    1.4683e-06    3
```

```
    {'p0_01'              }             0      0.07838          -101            1      0.08341
    {'p10'                }        1.0833      0.41297        -30.14            1      0.49019
    {'p25'                }        3.0034      0.65765        -16.23            1       0.7717
    {'p50'                }        6.7818       1.0568        -6.363            2       1.2612
    {'p75'                }        12.812       1.5534       0.45344            2       2.0256
    {'p90'                }          20.8       2.1542        4.9139            3       3.0996
    {'p99_99'             }        112.23       8.4857        13.926            6       15.937
    {'fl_cov_ap_ss'       }        89.487       6.8831        97.649       0.8159       10.409
    {'fl_cor_ap_ss'       }             1      0.94746       0.69588     0.095013      0.86354
    {'fl_cov_c_ss'        }        6.8831      0.58977        8.5503      0.23192      0.85197
    {'fl_cor_c_ss'        }       0.94746            1       0.75055      0.33267      0.87063
    {'fl_cov_v_ss'        }        97.649       8.5503        220.04       2.4373       12.623
    {'fl_cor_v_ss'        }       0.69588      0.75055             1        0.181      0.66782
    {'fl_cov_n_ss'        }        0.8159      0.23192        2.4373      0.82404     0.055267
    {'fl_cor_n_ss'        }      0.095013      0.33267         0.181            1      0.04778
    {'fl_cov_y_head_inc'  }        10.409      0.85197        12.623     0.055267       1.6237
    {'fl_cor_y_head_inc'  }       0.86354      0.87063       0.66782      0.04778            1
    {'fl_cov_y_spouse'    }        2.2143      0.24542        3.4887      0.27625        0.116
    {'fl_cor_y_spouse'    }        0.2103      0.28712       0.21131      0.27342      0.08179
    {'fl_cov_yshr_wage'   }      -0.54196    -0.036396      -0.86915    0.0011758    -0.038212
    {'fl_cor_yshr_wage'   }      -0.56735     -0.46933      -0.58024     0.012827     -0.29697
    {'fl_cov_yshr_SS'     }             0            0             0            0            0
    {'fl_cor_yshr_SS'     }           NaN          NaN           NaN          NaN          NaN
    {'fl_cov_yshr_nttxss'}        0.19452     0.017952       0.42036    0.0075501     0.027003
    {'fl_cor_yshr_nttxss'}        0.67266       0.7647       0.92699      0.27208      0.69323
    {'fracByP0_01'        }             0   6.8812e-06     0.0011212      0.28329   5.8341e-06
    {'fracByP10'          }      0.004897     0.026408       0.43931      0.28329     0.022426
    {'fracByP25'          }      0.037048     0.092569       0.77208      0.28329     0.081818
    {'fracByP50'          }       0.16368      0.27051        1.0414      0.72028      0.23952
    {'fracByP75'          }       0.41532      0.53706        1.1137      0.72028      0.48823
    {'fracByP90'          }       0.67288      0.76168         1.075      0.85389      0.72007
    {'fracByP99_99'       }       0.99866      0.99926        1.0001            1      0.99889
age =100
xxx tb_outcomes: all stats xxx
    OriginalVariableNames     ap_ss      c_ss         v_ss          n_ss        y_head_inc      y_sp
    ______________________    _____    _________    __________    __________    ___________    _____

    {'mean'               }      0      0.34868       -3.0033        1.4797         0.2604      0.
    {'sd'                 }      0      0.23392         1.043       0.50567        0.02289      0.
    {'coefofvar'          }    NaN      0.67088      -0.34728       0.34173       0.087904       2
    {'min'                }      0       0.2179       -10.065             1        0.24433
    {'max'                }      0       141.66       0.99282             6         5.6926
    {'pYis0'              }      1            0             0             0              0      0.
    {'pYls0'              }      0            0       0.99285             0              0
    {'pYgr0'              }      0            1     0.0071501             1              1      0.
    {'pYisMINY'           }      1      0.36483    1.5455e-10        0.5232        0.52813      0.
    {'pYisMAXY'           }      1            0             0    4.2206e-08              0    1.033
    {'p0_01'              }      0       0.2179       -6.3349             1        0.24433
    {'p10'                }      0       0.2179       -3.6603             1        0.24433
    {'p25'                }      0       0.2179       -3.5892             1        0.24433
    {'p50'                }      0      0.25824       -3.5892             1        0.24433
    {'p75'                }      0      0.36458       -2.8095             2        0.29263      0.
    {'p90'                }      0       0.6134       -1.3055             2        0.29279      0.
    {'p99_99'             }      0       2.8989       0.51215             4        0.33789       2
    {'fl_cov_ap_ss'       }      0            0             0             0              0
    {'fl_cor_ap_ss'       }    NaN          NaN           NaN           NaN            NaN
    {'fl_cov_c_ss'        }      0     0.054721       0.19746      0.059476      0.0015551      0.
    {'fl_cor_c_ss'        }    NaN            1       0.80934       0.50281        0.29042      0.
```

```
    {'fl_cov_v_ss'       }      0       0.19746        1.0878       0.16711       0.01031       0
    {'fl_cor_v_ss'       }    NaN       0.80934             1       0.31686       0.43183       0.
    {'fl_cov_n_ss'       }      0      0.059476       0.16711        0.2557     0.0019105       0
    {'fl_cor_n_ss'       }    NaN       0.50281       0.31686             1       0.16506       0
    {'fl_cov_y_head_inc' }      0     0.0015551       0.01031     0.0019105    0.00052397    0.000
    {'fl_cor_y_head_inc' }    NaN       0.29042       0.43183       0.16506             1       0.
    {'fl_cov_y_spouse'   }      0       0.05178        0.1649        0.0533    0.00067518     0.0
    {'fl_cor_y_spouse'   }    NaN       0.89356       0.63823        0.4255       0.11907
    {'fl_cov_yshr_wage'  }      0      0.039513       0.15927      0.083913    0.00067571     0.0
    {'fl_cor_yshr_wage'  }    NaN        0.7643       0.69097       0.75087       0.13357       0.
    {'fl_cov_yshr_SS'    }      0     -0.040547      -0.16461     -0.085285   -0.00072523    -0.0
    {'fl_cor_yshr_SS'    }    NaN      -0.77966      -0.70991      -0.75864      -0.14251      -0.
    {'fl_cov_yshr_nttxss'}      0      0.044511       0.18091      0.091879    0.00087698     0.0
    {'fl_cor_yshr_nttxss'}    NaN       0.78763       0.71798       0.75212       0.15859       0.
    {'fracByP0_01'       }    NaN       0.22799    0.00053042       0.35357       0.49553
    {'fracByP10'         }    NaN       0.22799       0.22059       0.35357       0.49553
    {'fracByP25'         }    NaN       0.22799        0.6552       0.35357       0.49553
    {'fracByP50'         }    NaN       0.35394        0.6552       0.35357       0.49553
    {'fracByP75'         }    NaN       0.55083       0.87677       0.99419       0.88359       0.
    {'fracByP90'         }    NaN        0.7612       0.97549       0.99419       0.89158       0.
    {'fracByP99_99'      }    NaN       0.99927             1       0.99999       0.99991       0
```

5.2 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.

5.2.1 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_contr

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(:));
```

5.2.2 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
                    --------    --------    ---------    --------    -------    -------    -------

    a_ss             3.6126      6.4914      1.7969            0        135     0.17087          0
    ap_ss            12.366       8.591     0.69474            1         55           0          0
```

```
cons_ss            1.1622     0.80935     0.69639    0.036857     140.65          0          0
v_ss              -15.043      17.999     -1.1965      -597.7     23.892          0     0.8378
n_ss               2.3554      1.4375     0.61029           1          6          0          0
y_all              1.5684      1.4453     0.92149    0.038325     47.427          0          0
y_head_inc         1.2411      1.1553      0.9309    0.038325     31.844          0          0
y_head_earn        1.0444      1.0725      1.0269           0     26.444     0.2016          0
y_spouse_inc      0.32734     0.73631      2.2494           0     15.702    0.52499          0
yshr_interest    0.096139     0.15385      1.6002           0    0.99295    0.17087          0
yshr_wage         0.79228     0.33742     0.42588           0          1    0.10584          0
yshr_SS           0.11158     0.25418       2.278           0          1     0.7984          0
yshr_tax          0.18447    0.034469     0.18686    0.038299    0.25519          0          0
yshr_nttxss      0.072887     0.27591      3.7854    -0.88844    0.25519          0    0.18437
```

5.2.3 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=%3.2
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});
```

5.2.4 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

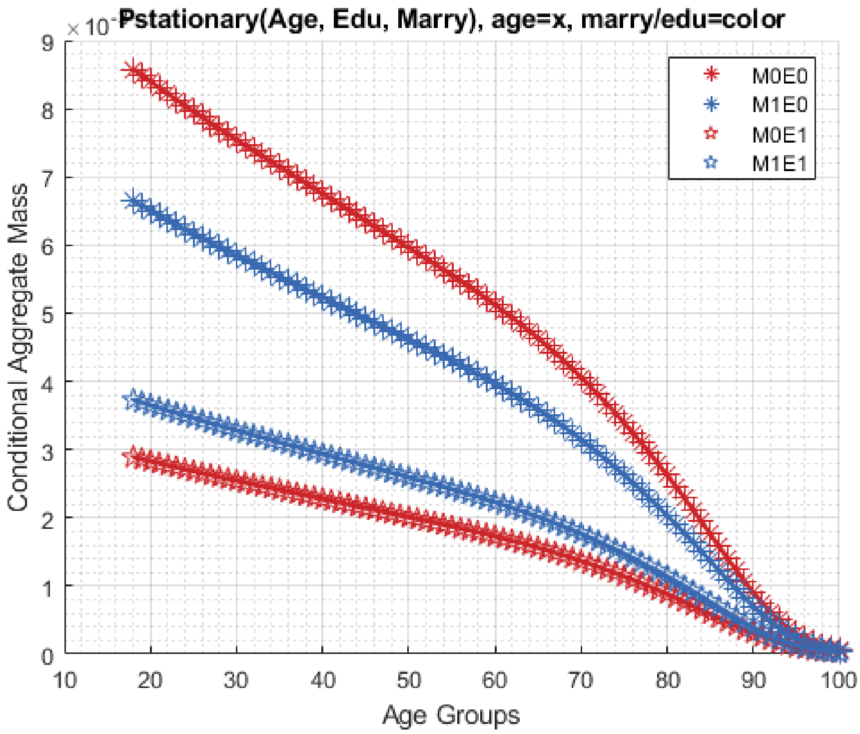
Tabulate value and policies along savings and shocks:

```
% 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_datasetde
```

```
xxx  P(Age, EDU, MARRY))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    marry    edu    sum_age_18    sum_age_19    sum_age_20    sum_age_21    sum_age_22    s
    -----    -----    ---    ----------    ----------    ----------    ----------    ----------    -

      1        0       0     0.0085768     0.0084866     0.0083969     0.0083078     0.0082194     0
      2        1       0     0.0066438     0.0065739     0.0065044     0.0064354     0.0063669
```

3	0	1	0.0028875	0.0028571	0.002827	0.002797	0.0027672	0
4	1	1	0.0037292	0.0036899	0.0036509	0.0036122	0.0035738	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

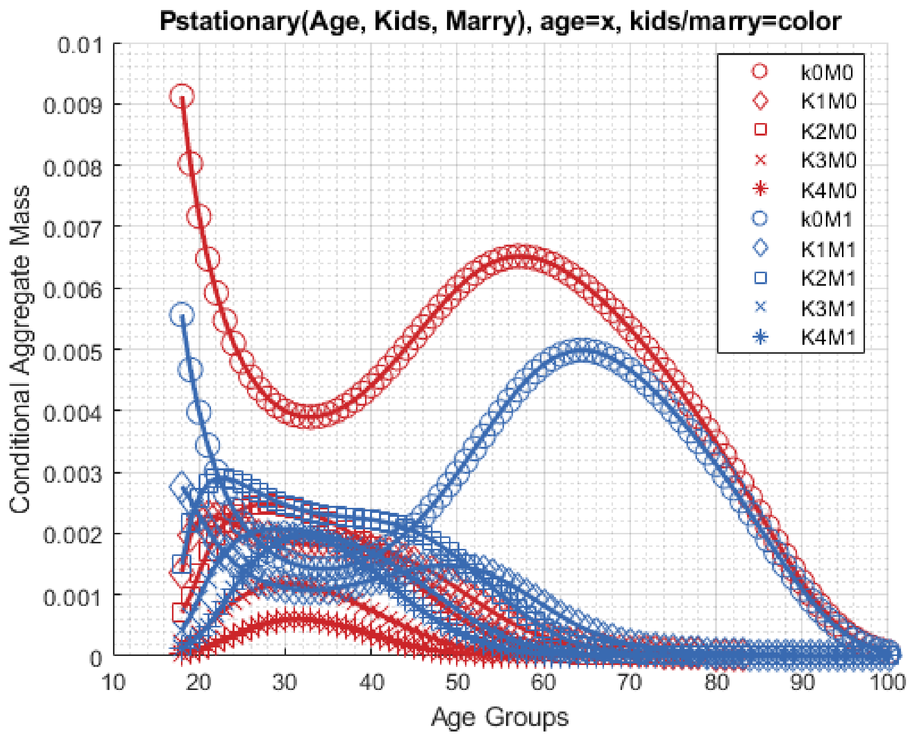
```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
% Value Function
tb_prob_amarrykids = ff_summ_nd_array("P(Age, Kids, Marry))", Phi_true, true, ["sum"], 3, 1, cl_mp_d
```

xxx P(Age, Kids, Marry)) xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	kids	marry	sum_age_18	sum_age_19	sum_age_20	sum_age_21	sum_age_22
1	1	0	0.0091249	0.0080278	0.0071652	0.0064765	0.0059205
2	2	0	0.0013699	0.0019743	0.0022187	0.0022858	0.0022687
3	3	0	0.00071266	0.00098425	0.0013537	0.0016929	0.0019639
4	4	0	0.00020622	0.00027865	0.00037326	0.00049476	0.00062818
5	5	0	5.0761e-05	7.8715e-05	0.000113	0.00015485	0.00020534
6	1	1	0.0055624	0.0046679	0.0039774	0.0034368	0.0030088
7	2	1	0.0027682	0.0025539	0.0023005	0.0020611	0.0018525
8	3	1	0.0014982	0.0021823	0.0025943	0.0028096	0.002896
9	4	1	0.00041197	0.00064648	0.00095224	0.0012491	0.0015009
10	5	1	0.00013221	0.0002132	0.00033097	0.00049097	0.00068255

```
mp_support_graph('cl_st_graph_title') = {'Pstationary(Age, Kids, Marry), age=x, kids/marry=color'};
mp_support_graph('cl_st_ytitle') = {'Conditional Aggregate Mass'};
```

```
ar_row_grid = [...
    "kOM0", "K1M0", "K2M0", "K3M0", "K4M0", ...
    "kOM1", "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);
```



5.2.5 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
```

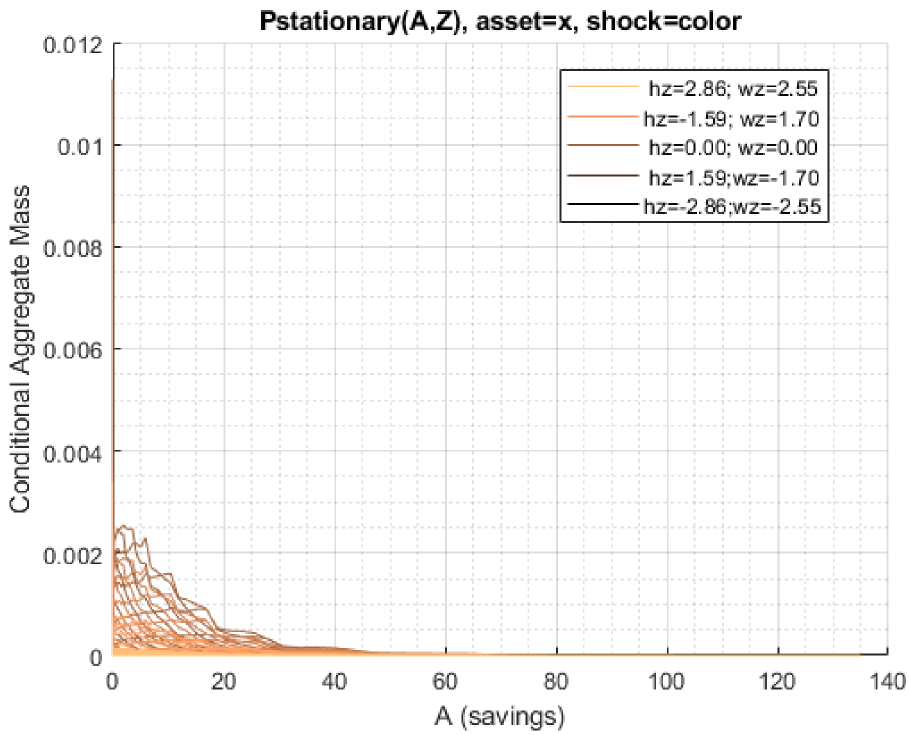
Asset and Shock Mass

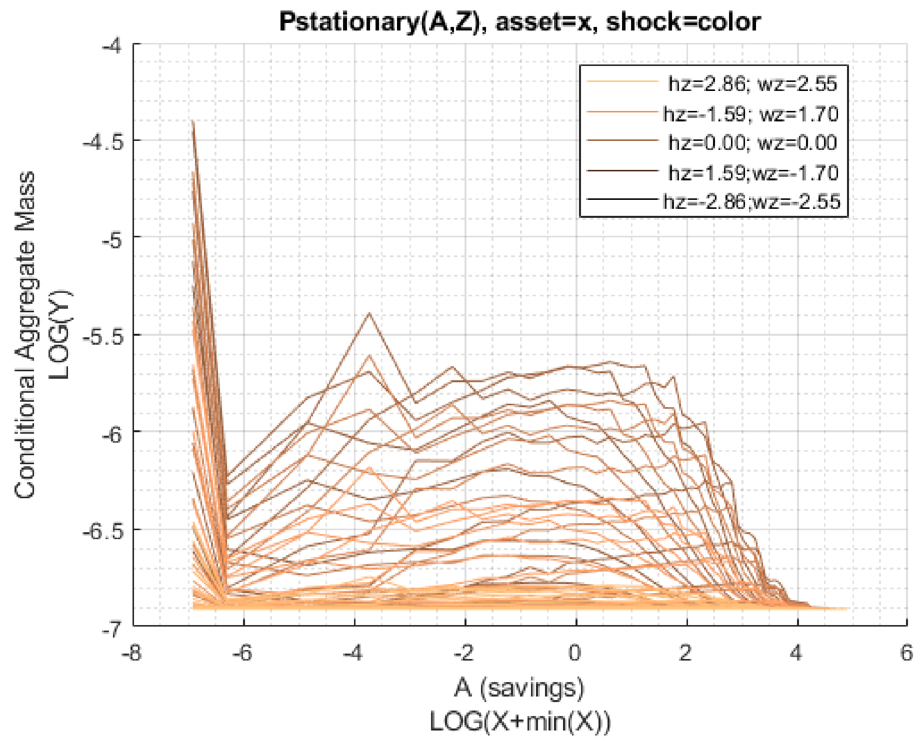
```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_prob_az = ff_summ_nd_array("P(A,Z))", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar_permut
```

```
xxx  P(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings       sum_eta_1     sum_eta_2     sum_eta_3     sum_eta_4     sum_eta_5     sum
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0     3.3248e-08     5.983e-07    5.0468e-06    2.6071e-05    8.9773e-05    0.0
      2     0.00085734     1.0185e-10    1.5738e-09    3.3484e-08    6.0665e-07    5.2814e-06    1.7
      3      0.0068587     3.5085e-10    6.5198e-09    4.9979e-08    3.1065e-07    2.0919e-06    1.2
```

4	0.023148	9.1216e-10	1.727e-08	1.5853e-07	9.1924e-07	4.228e-06	1.6
5	0.05487	1.4512e-09	2.6722e-08	2.3417e-07	1.3076e-06	5.2407e-06	1.8
6	0.10717	1.6379e-09	2.9149e-08	2.4125e-07	1.2884e-06	5.1297e-06	1.7
7	0.18519	2.294e-09	4.0502e-08	3.4422e-07	1.7987e-06	6.7964e-06	2.0
8	0.29407	2.7467e-09	4.5812e-08	3.5554e-07	1.8118e-06	6.8723e-06	2.0
9	0.43896	2.8104e-09	4.8243e-08	3.9121e-07	1.9756e-06	7.1951e-06	2.0
10	0.625	2.7355e-09	4.7422e-08	3.8625e-07	1.9709e-06	7.2487e-06	2.0
11	0.85734	2.6045e-09	4.5648e-08	3.7801e-07	1.9446e-06	7.1135e-06	2.0
12	1.1411	2.3091e-09	4.118e-08	3.427e-07	1.7857e-06	6.6881e-06	1.9
13	1.4815	1.8391e-09	3.3807e-08	2.9225e-07	1.5886e-06	6.0926e-06	1.7
14	1.8836	1.4638e-09	2.7339e-08	2.4063e-07	1.3481e-06	5.3035e-06	1.5
15	2.3525	1.0398e-09	2.042e-08	1.9039e-07	1.1046e-06	4.5148e-06	1.3
16	2.8935	7.6831e-10	1.5168e-08	1.4306e-07	8.5678e-07	3.6129e-06	1.1
17	3.5117	5.0779e-10	1.0306e-08	1.0222e-07	6.4935e-07	2.7868e-06	9.0
18	4.2121	3.3725e-10	7.1325e-09	7.2301e-08	4.5701e-07	2.0605e-06	7.
19	5	2.0339e-10	4.5055e-09	4.7886e-08	3.1798e-07	1.4784e-06	5.2
20	5.8805	1.1697e-10	2.6614e-09	2.899e-08	2.04e-07	1.0085e-06	3.8
21	6.8587	6.4191e-11	1.5386e-09	1.7458e-08	1.2852e-07	6.6971e-07	2.7
22	7.9398	3.223e-11	8.2009e-10	1.023e-08	7.992e-08	4.4241e-07	1.8
23	9.1289	1.5935e-11	4.2547e-10	5.5799e-09	4.7148e-08	2.8701e-07	1.3
24	10.431	7.6602e-12	2.1449e-10	3.021e-09	2.7923e-08	1.8511e-07	9.7
25	11.852	3.4707e-12	1.1014e-10	1.6503e-09	1.6439e-08	1.1882e-07	6.7
26	13.396	1.5171e-12	5.0241e-11	8.4768e-10	9.2919e-09	7.4842e-08	4
27	15.069	6.5407e-13	2.3567e-11	4.244e-10	4.9998e-09	4.3238e-08	2.8

```
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);% Consumption
```





Asset Mass by Age

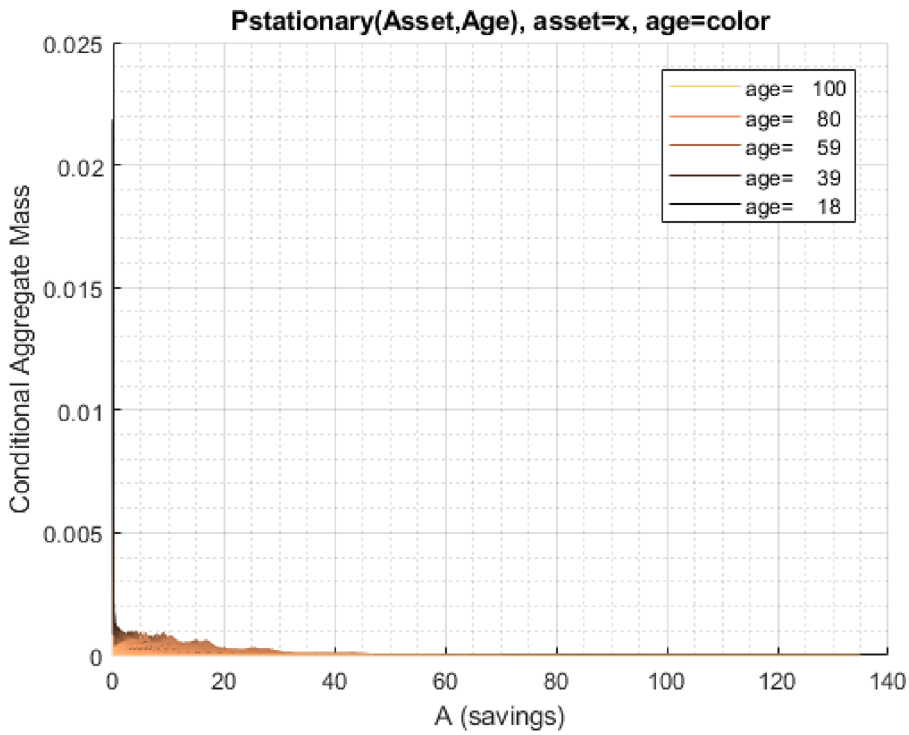
```
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [3,4,5,6,1,2];
% Value Function
tb_prob_aage = ff_summ_nd_array("P(A,Z))", Phi_true, true, ["sum"], 4, 1, cl_mp_datasetdesc, ar_perm
```

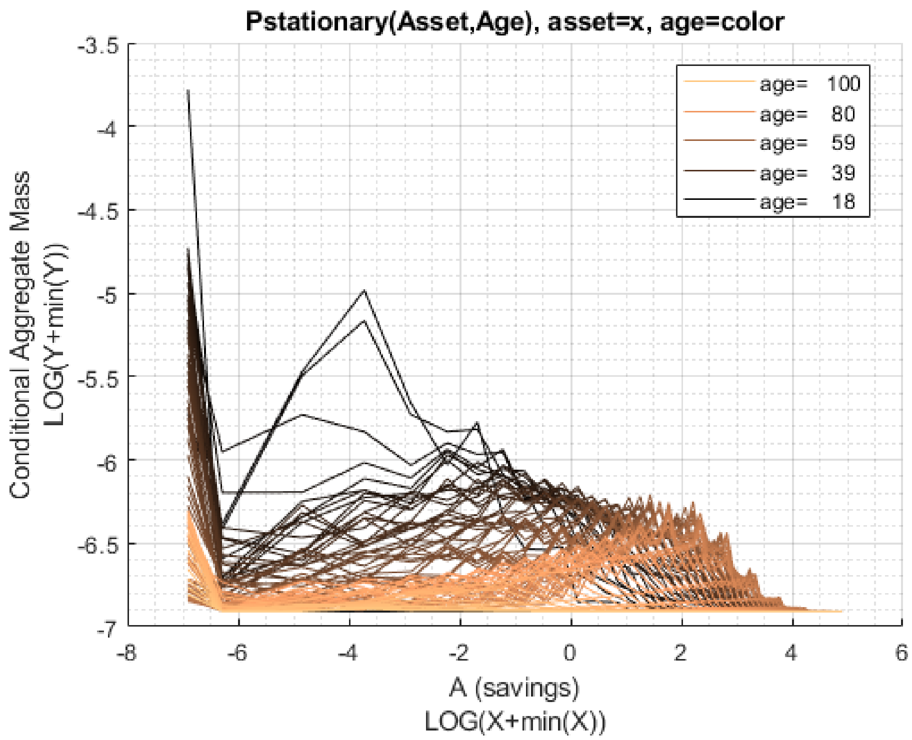
xxx P(A,Z)) xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	savings	sum_age_18	sum_age_19	sum_age_20	sum_age_21	sum_age_22	sum
1	0	0.021837	0.0034489	0.0022777	0.0051619	0.0070701	0.
2	0.00085734	0	0.00068848	0.00063521	0.0016036	0.0010378	0.0
3	0.0068587	0	0.0032017	0.0031082	0.0022491	0.0010432	0.0
4	0.023148	0	0.0058625	0.0047177	0.0019351	0.0014397	0.
5	0.05487	0	0.0024915	0.0022549	0.0014027	0.001227	0.
6	0.10717	0	0.0014086	0.0019373	0.0017422	0.0016421	
7	0.18519	0	0.0021091	0.0019779	0.0015628	0.0013724	0.
8	0.29407	0	0.00077614	0.0012753	0.0015936	0.0016414	0.
9	0.43896	0	0.00045252	0.001062	0.0009547	0.00098397	0.
10	0.625	0	0.00044667	0.0006628	0.0010416	0.00096275	0.0
11	0.85734	0	0.00047477	0.00074294	0.0006549	0.00088876	0.0
12	1.1411	0	5.5654e-05	0.00024727	0.00052409	0.00050156	0.0
13	1.4815	0	6.2855e-05	0.00019563	0.00024402	0.00047382	0.0
14	1.8836	0	5.9284e-05	0.00012121	0.00019477	0.00022476	0.0
15	2.3525	0	5.4786e-05	9.1062e-05	0.00015255	0.0001713	0.
16	2.8935	0	1.2562e-05	5.2495e-05	7.1526e-05	0.00013824	0.0
17	3.5117	0	1.3674e-06	1.1359e-05	3.0456e-05	4.9417e-05	0.0
18	4.2121	0	1.7163e-07	5.944e-06	2.1633e-05	2.4249e-05	3.9
19	5	0	7.8629e-08	1.35e-06	5.2915e-06	1.3252e-05	2.1
20	5.8805	0	6.0581e-09	7.6218e-07	4.4269e-06	1.3802e-05	1.2
21	6.8587	0	4.4396e-10	9.1165e-08	6.9508e-07	3.4569e-06	1.
22	7.9398	0	0	3.4291e-08	4.1481e-07	2.9624e-06	4.2
23	9.1289	0	0	2.5522e-08	3.1203e-07	5.0602e-07	2.4

```
    24        10.431          0          0    6.5868e-10    4.3387e-08    3.9922e-07    1.8
    25        11.852          0          0    2.4326e-10    1.8803e-08    2.6275e-07    4.5
    26        13.396          0          0    4.8838e-12    1.8589e-08     3.869e-08    2.8
    27        15.069          0          0     1.813e-12    4.7634e-10    2.1906e-08    2.1
    28        16.875          0          0             0    6.2066e-11    1.9079e-08    3.5
    29         18.82          0          0             0      6.42e-12    6.8737e-10    2.0
    30         20.91          0          0             0    1.6411e-13     9.995e-11    1.6
    31        23.148          0          0             0    1.9878e-14    1.2652e-11    7.6
    32        25.541          0          0             0             0     7.895e-13    1.3
    33        28.093          0          0             0             0    1.2464e-14      1
    34         30.81          0          0             0             0    2.5551e-16     5.
    35        33.697          0          0             0             0             0    1.2
    36        36.758          0          0             0             0             0    2.0
    37            40          0          0             0             0             0    2.4
    38        43.427          0          0             0             0             0
    39        47.044          0          0             0             0             0
    40        50.856          0          0             0             0             0
    41         54.87          0          0             0             0             0
    42        59.089          0          0             0             0             0
    43        63.519          0          0             0             0             0
    44        68.164          0          0             0             0             0
    45        73.032          0          0             0             0             0
    46        78.125          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_aage{1:end, 3:end})', age_grid, agrid, mp_support_graph);% Consumption Choice
```





5.2.6 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_of_s
    end
end

age =18
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       ap_ss         c_ss          v_ss          n_ss       y_head_inc
    ______________________    ___________    ___________    __________    ___________    ___________

    {'mean'              }         4.4841        0.73608       -24.737         1.9854        0.83951
    {'sd'                }         2.5942        0.43457        24.426         1.0848        0.62548
    {'coefofvar'         }        0.57854        0.59038      -0.98745        0.54639        0.74505
    {'min'               }              1       0.036857        -597.7              1       0.038325
    {'max'               }             55          17.35        23.892              6         17.095
    {'pYis0'             }              0              0             0              0              0
    {'pYls0'             }              0              0       0.91009              0              0
    {'pYgr0'             }              1              1      0.089914              1              1
    {'pYisMINY'          }        0.15961     2.0027e-06    8.8672e-09        0.41786     3.8147e-06
    {'pYisMAXY'          }              0              0             0      0.0060544              0
    {'p0_01'             }              1         0.0682        -251.7              1       0.072367
    {'p10'               }              1        0.30119       -51.639              1        0.25802
    {'p25'               }              3        0.40819       -36.859              1         0.4872
    {'p50'               }              4        0.65163       -19.949              2        0.66948
    {'p75'               }              6        0.91178       -7.7939              3        0.91995
    {'p90'               }              8         1.2709       -1.1471              4         1.7371
    {'p99_99'            }             16         4.1437        17.666              6         6.1934
    {'fl_cov_ap_ss'      }           6.73        0.61926        25.315      -0.086009        0.71874
    {'fl_cor_ap_ss'      }              1         0.5493        0.3995      -0.030562        0.44295
    {'fl_cov_c_ss'       }        0.61926        0.18885        7.7987        0.07295        0.25218
    {'fl_cor_c_ss'       }         0.5493              1        0.7347        0.15474        0.92778
    {'fl_cov_v_ss'       }         25.315         7.7987        596.64        -1.0333         10.003
    {'fl_cor_v_ss'       }         0.3995         0.7347             1      -0.038995        0.65476
    {'fl_cov_n_ss'       }      -0.086009        0.07295       -1.0333         1.1768     2.5745e-18
    {'fl_cor_n_ss'       }      -0.030562        0.15474     -0.038995              1     3.7942e-18
    {'fl_cov_y_head_inc' }        0.71874        0.25218        10.003     2.5745e-18        0.39122
    {'fl_cor_y_head_inc' }        0.44295        0.92778       0.65476     3.7942e-18              1
```

```
    {'fl_cov_y_spouse'   }       0.77387       0.060176        2.2091        0.12195       0.010942
    {'fl_cor_y_spouse'   }       0.68113        0.31618       0.20651        0.25667       0.039945
    {'fl_cov_yshr_wage'  }   -1.9999e-30    -3.1429e-31    1.1155e-29    -5.0228e-31    -4.8634e-31
    {'fl_cor_yshr_wage'  }   -1.7359e-15    -1.6286e-15    1.0284e-15    -1.0426e-15    -1.7509e-15
    {'fl_cov_yshr_SS'    }             0              0             0              0              0
    {'fl_cor_yshr_SS'    }           NaN            NaN           NaN            NaN            NaN
    {'fl_cov_yshr_nttxss'}       0.05377       0.011984       0.67106      0.0063843       0.014635
    {'fl_cor_yshr_nttxss'}       0.65501        0.87148        0.8682        0.18598        0.73941
    {'fracByP0_01'       }      0.035596     3.1006e-05     0.0014244        0.21046     5.4793e-05
    {'fracByP10'         }      0.035596       0.046234       0.31706        0.21046       0.031399
    {'fracByP25'         }       0.14894        0.11395       0.58865        0.21046        0.17954
    {'fracByP50'         }       0.39097        0.28419       0.84937        0.53024        0.32745
    {'fracByP75'         }       0.60677        0.54782       0.99563        0.77109        0.51036
    {'fracByP90'         }       0.82323        0.77805        1.0163        0.92834         0.8397
    {'fracByP99_99'      }       0.99971        0.99944             1              1        0.99925

age =59
xxx tb_outcomes: all stats xxx
    OriginalVariableNames       ap_ss          c_ss          v_ss          n_ss        y_head_inc
    ______________________    __________    __________    __________    __________    __________

    {'mean'              }        19.816         1.359       -12.596        1.7239        1.7902
    {'sd'                }        7.5997       0.94263        14.529       0.90777        1.4384
    {'coefofvar'         }        0.3835       0.69362       -1.1534       0.52659       0.80349
    {'min'               }             1      0.056816       -211.91             1       0.05988
    {'max'               }            55        31.643        14.416             6        30.606
    {'pYis0'             }             0             0             0             0             0
    {'pYls0'             }             0             0       0.80596             0             0
    {'pYgr0'             }             1             1       0.19404             1             1
    {'pYisMINY'          }     0.0097508    1.4955e-06    4.4767e-10       0.48835    1.5128e-06
    {'pYisMAXY'          }    3.9575e-05    3.6923e-09      2.35e-07     0.0036816    5.7591e-07
    {'p0_01'             }             1        0.1046       -83.156             1       0.11307
    {'p10'               }            10        0.4435       -33.387             1       0.56354
    {'p25'               }            15       0.68544       -20.857             1       0.82346
    {'p50'               }            20        1.1501       -9.5768             2        1.4112
    {'p75'               }            25        1.7983       -1.8874             2        2.3004
    {'p90'               }            29        2.5653        3.2283             3        3.3891
    {'p99_99'            }            52        10.514        13.528             6        17.142
    {'fl_cov_ap_ss'      }        57.755        5.9089        102.91         0.833        8.8962
    {'fl_cor_ap_ss'      }             1       0.82483       0.93205       0.12075       0.81382
    {'fl_cov_c_ss'       }        5.9089       0.88856        10.148       0.19066        1.1168
    {'fl_cor_c_ss'       }       0.82483             1       0.74094       0.22282       0.82371
    {'fl_cov_v_ss'       }        102.91        10.148         211.1        2.8206          14.8
    {'fl_cor_v_ss'       }       0.93205       0.74094             1       0.21386       0.70816
    {'fl_cov_n_ss'       }         0.833       0.19066        2.8206       0.82404      0.051157
    {'fl_cor_n_ss'       }       0.12075       0.22282       0.21386             1      0.039179
    {'fl_cov_y_head_inc' }        8.8962        1.1168          14.8      0.051157         2.069
    {'fl_cor_y_head_inc' }       0.81382       0.82371       0.70816      0.039179             1
    {'fl_cov_y_spouse'   }        1.5795       0.19114        3.2552       0.25285       0.10701
    {'fl_cor_y_spouse'   }       0.24292       0.23699       0.26185       0.32555       0.08695
    {'fl_cov_yshr_wage'  }      -0.49211     -0.043084      -0.82571    -0.0010612     -0.051178
    {'fl_cor_yshr_wage'  }      -0.75357      -0.53189      -0.66137     -0.013605      -0.41406
    {'fl_cov_yshr_SS'    }             0             0             0             0             0
    {'fl_cor_yshr_SS'    }           NaN           NaN           NaN           NaN           NaN
    {'fl_cov_yshr_nttxss'}       0.18595      0.018907       0.38617     0.0064928      0.028554
    {'fl_cor_yshr_nttxss'}       0.86513       0.70922       0.93979        0.2529        0.7019
    {'fracByP0_01'       }    0.00049206    1.8935e-05     0.0018764       0.28329    1.6223e-05
    {'fracByP10'         }      0.038429      0.027522       0.35161       0.28329      0.022756
```

```
    {'fracByP25'        }      0.15239      0.088557       0.65681       0.28329      0.081638
    {'fracByP50'        }      0.38532       0.25604       0.94508       0.72028       0.23879
    {'fracByP75'        }      0.65807       0.51752        1.0572       0.72028        0.4917
    {'fracByP90'        }      0.83502       0.75552        1.0505       0.85389       0.71833
    {'fracByP99_99'     }      0.99977       0.99911        1.0001             1         0.999

age =100
xxx tb_outcomes: all stats xxx
    OriginalVariableNames        ap_ss          c_ss          v_ss          n_ss       y_head_inc
    _____________________    ___________    ___________    __________    ___________    ___________

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

Chapter 6

Value of Each Check

6.1 2020 V and C without Unemployment

This is the example vignette for function: **snw_a4chk_wrk_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.

6.1.1 Test SNW_A4CHK_WRK_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') = true;
mp_controls('bl_timer') = true;
[V_ss,~,cons_ss,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
SNW_VFI_MAIN: Finished Age Group:83 of 83
SNW_VFI_MAIN: Finished Age Group:82 of 83
SNW_VFI_MAIN: Finished Age Group:81 of 83
SNW_VFI_MAIN: Finished Age Group:80 of 83
SNW_VFI_MAIN: Finished Age Group:79 of 83
SNW_VFI_MAIN: Finished Age Group:78 of 83
SNW_VFI_MAIN: Finished Age Group:77 of 83
SNW_VFI_MAIN: Finished Age Group:76 of 83
SNW_VFI_MAIN: Finished Age Group:75 of 83
SNW_VFI_MAIN: Finished Age Group:74 of 83
SNW_VFI_MAIN: Finished Age Group:73 of 83
SNW_VFI_MAIN: Finished Age Group:72 of 83
SNW_VFI_MAIN: Finished Age Group:71 of 83
SNW_VFI_MAIN: Finished Age Group:70 of 83
SNW_VFI_MAIN: Finished Age Group:69 of 83
SNW_VFI_MAIN: Finished Age Group:68 of 83
SNW_VFI_MAIN: Finished Age Group:67 of 83
SNW_VFI_MAIN: Finished Age Group:66 of 83
SNW_VFI_MAIN: Finished Age Group:65 of 83
SNW_VFI_MAIN: Finished Age Group:64 of 83
SNW_VFI_MAIN: Finished Age Group:63 of 83
SNW_VFI_MAIN: Finished Age Group:62 of 83
SNW_VFI_MAIN: Finished Age Group:61 of 83
SNW_VFI_MAIN: Finished Age Group:60 of 83
SNW_VFI_MAIN: Finished Age Group:59 of 83
```

```
SNW_VFI_MAIN: Finished Age Group:58 of 83
SNW_VFI_MAIN: Finished Age Group:57 of 83
SNW_VFI_MAIN: Finished Age Group:56 of 83
SNW_VFI_MAIN: Finished Age Group:55 of 83
SNW_VFI_MAIN: Finished Age Group:54 of 83
SNW_VFI_MAIN: Finished Age Group:53 of 83
SNW_VFI_MAIN: Finished Age Group:52 of 83
SNW_VFI_MAIN: Finished Age Group:51 of 83
SNW_VFI_MAIN: Finished Age Group:50 of 83
SNW_VFI_MAIN: Finished Age Group:49 of 83
SNW_VFI_MAIN: Finished Age Group:48 of 83
SNW_VFI_MAIN: Finished Age Group:47 of 83
SNW_VFI_MAIN: Finished Age Group:46 of 83
SNW_VFI_MAIN: Finished Age Group:45 of 83
SNW_VFI_MAIN: Finished Age Group:44 of 83
SNW_VFI_MAIN: Finished Age Group:43 of 83
SNW_VFI_MAIN: Finished Age Group:42 of 83
SNW_VFI_MAIN: Finished Age Group:41 of 83
SNW_VFI_MAIN: Finished Age Group:40 of 83
SNW_VFI_MAIN: Finished Age Group:39 of 83
SNW_VFI_MAIN: Finished Age Group:38 of 83
SNW_VFI_MAIN: Finished Age Group:37 of 83
SNW_VFI_MAIN: Finished Age Group:36 of 83
SNW_VFI_MAIN: Finished Age Group:35 of 83
SNW_VFI_MAIN: Finished Age Group:34 of 83
SNW_VFI_MAIN: Finished Age Group:33 of 83
SNW_VFI_MAIN: Finished Age Group:32 of 83
SNW_VFI_MAIN: Finished Age Group:31 of 83
SNW_VFI_MAIN: Finished Age Group:30 of 83
SNW_VFI_MAIN: Finished Age Group:29 of 83
SNW_VFI_MAIN: Finished Age Group:28 of 83
SNW_VFI_MAIN: Finished Age Group:27 of 83
SNW_VFI_MAIN: Finished Age Group:26 of 83
SNW_VFI_MAIN: Finished Age Group:25 of 83
SNW_VFI_MAIN: Finished Age Group:24 of 83
SNW_VFI_MAIN: Finished Age Group:23 of 83
SNW_VFI_MAIN: Finished Age Group:22 of 83
SNW_VFI_MAIN: Finished Age Group:21 of 83
SNW_VFI_MAIN: Finished Age Group:20 of 83
SNW_VFI_MAIN: Finished Age Group:19 of 83
SNW_VFI_MAIN: Finished Age Group:18 of 83
SNW_VFI_MAIN: Finished Age Group:17 of 83
SNW_VFI_MAIN: Finished Age Group:16 of 83
SNW_VFI_MAIN: Finished Age Group:15 of 83
SNW_VFI_MAIN: Finished Age Group:14 of 83
SNW_VFI_MAIN: Finished Age Group:13 of 83
SNW_VFI_MAIN: Finished Age Group:12 of 83
SNW_VFI_MAIN: Finished Age Group:11 of 83
SNW_VFI_MAIN: Finished Age Group:10 of 83
SNW_VFI_MAIN: Finished Age Group:9 of 83
SNW_VFI_MAIN: Finished Age Group:8 of 83
SNW_VFI_MAIN: Finished Age Group:7 of 83
SNW_VFI_MAIN: Finished Age Group:6 of 83
SNW_VFI_MAIN: Finished Age Group:5 of 83
SNW_VFI_MAIN: Finished Age Group:4 of 83
SNW_VFI_MAIN: Finished Age Group:3 of 83
SNW_VFI_MAIN: Finished Age Group:2 of 83
SNW_VFI_MAIN: Finished Age Group:1 of 83
```

```
Elapsed time is 139.984500 seconds.
Completed SNW_VFI_MAIN;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test

welf_checks = 2;
[V_W, C_W] = snw_a4chk_wrk_bisec_vec(welf_checks, V_ss, cons_ss, mp_params, mp_controls);

Elapsed time is 76.079485 seconds.
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=2;TR=0.0017225;SNW_MP_PARAM=default_dense;SNW_MP_CONTR
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_container_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                      i    idx    ndim      numel       rowN    colN        sum           mean
                      -    ---    ----    ----------    ----    -----    -----------    ----------

    C_W               1     1      6      1.9173e+06     83     23100     9.1863e+06        4.7913
    C_W_minus_C_ss    2     2      6      1.9173e+06     83     23100         1018.4    0.00053118
    V_W               3     3      6      1.9173e+06     83     23100    -4.2855e+06       -2.2352
    V_W_minus_V_ss    4     4      6      1.9173e+06     83     23100          15640     0.0081571
    mn_MPC            5     5      6      1.9173e+06     83     23100     2.9563e+05       0.15419

mn_V_W_gain_check = V_W - V_ss;
mn_MPC_W_gain_share_check = (C_W - cons_ss)./(welf_checks*mp_params('TR'));
```

6.1.2 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=%3.2
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});
```

6.1.3 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_W_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_par
tb_az_v = ff_summ_nd_array(st_title, mn_V_W_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_W_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    mea
    _____    __________    __________    __________    __________    __________    __________    ___

      1               0        0.1253      0.074981      0.055673      0.048117      0.044975      0
      2      0.00085734       0.12419      0.074491      0.055326      0.047802      0.044666      0
      3       0.0068587        0.1134      0.069223      0.051492      0.044298      0.041233      0
      4        0.023148      0.097789      0.061019      0.045415      0.038762      0.035824      0
      5         0.05487      0.080591      0.051512      0.038393      0.032456      0.029694      0
      6         0.10717      0.066209       0.04312      0.032206      0.026999      0.024439
      7         0.18519      0.055273      0.036722      0.027464      0.022816      0.020445      0
      8         0.29407      0.046205      0.031335      0.023475      0.019332      0.017115      0
      9         0.43896      0.038899      0.026901      0.020203      0.016487      0.014432      0
     10           0.625      0.032914      0.023206       0.01748      0.014151      0.012235      0
     11         0.85734      0.027964      0.020082      0.015185      0.012196      0.010425      0.
     12          1.1411      0.023831      0.017415      0.013228       0.01055     0.0089155      0.
     13          1.4815      0.020354      0.015123      0.011544      0.009154     0.0076473      0.
     14          1.8836      0.017411      0.013142      0.010084     0.0079624     0.0065822      0.
     15          2.3525       0.01491      0.011422     0.0088155     0.0069411     0.0056799      0.
     16          2.8935       0.01278     0.0099292     0.0077111     0.0060627     0.0049139
     17          3.5117      0.010963     0.0086325      0.006748     0.0053037     0.0042629      0.
     18          4.2121     0.0094097     0.0075049     0.0059067     0.0046452      0.003708      0.
     19               5     0.0080801     0.0065236     0.0051712     0.0040728     0.0032336      0.
     20          5.8805     0.0069408      0.005669     0.0045263     0.0035738     0.0028265      0.
     21          6.8587     0.0059645     0.0049254     0.0039619     0.0031383     0.0024763      0.
     22          7.9398     0.0051283     0.0042793     0.0034693     0.0027581     0.0021743      0.
     23          9.1289     0.0044123     0.0037185     0.0030385     0.0024262     0.0019129      0.
     24          10.431     0.0037995     0.0032324     0.0026622     0.0021364     0.0016863      0.
     25          11.852      0.003275     0.0028112      0.002334     0.0018832     0.0014891      0.
     26          13.396      0.002826     0.0024466     0.0020476     0.0016617     0.0013172      0.
     27          15.069     0.0024417      0.002131     0.0017977     0.0014674      0.001167      0.0
     28          16.875     0.0021125     0.0018578     0.0015793      0.001297     0.0010354      0.0
     29           18.82     0.0018302     0.0016212     0.0013885     0.0011478    0.00092002      0.0
     30           20.91     0.0015879     0.0014162     0.0012217     0.0010167    0.00081861      0.0
     31          23.148     0.0013798     0.0012385     0.0010759     0.0009012    0.00072932      0.0
     32          25.541     0.0012009     0.0010843     0.0009484    0.00079962     0.0006506      0.0
     33          28.093     0.0010468     0.0009506    0.00083682    0.00071027    0.00058112      0.0
     34           30.81    0.00091399    0.00083444    0.00073914    0.00063151    0.00051959      0.
     35          33.697    0.00079937    0.00073346    0.00065356      0.000562    0.00046497      0.0
     36          36.758    0.00070031     0.0006456    0.00057854    0.00050061    0.00041657      0.0
     37              40    0.00061458    0.00056907    0.00051272    0.00044637    0.00037369      0.0
     38          43.427    0.00054027    0.00050235    0.00045493    0.00039841    0.00033554      0.0
     39          47.044    0.00047578     0.0004441    0.00040415    0.00035595    0.00030154      0.0
     40          50.856     0.0004197     0.0003932    0.00035947    0.00031835    0.00027123      0.0
     41           54.87    0.00037088    0.00034866    0.00032015    0.00028502    0.00024425      0.0
     42          59.089    0.00032832    0.00030963    0.00028549    0.00025545    0.00022017      0.0
     43          63.519    0.00029112    0.00027538    0.00025491     0.0002292    0.00019866      0.
     44          68.164    0.00025858     0.0002453    0.00022791    0.00020588    0.00017942      0.0
     45          73.032    0.00023009    0.00021884    0.00020403    0.00018514    0.00016219      0.0
     46          78.125    0.00020506    0.00019552     0.0001829    0.00016667    0.00014676      0.0
     47           83.45    0.00018305    0.00017495    0.00016417    0.00015021    0.00013292      0.0
```

48	89.011	0.00016367	0.00015678	0.00014756	0.00013553	0.00012051	0.0
49	94.815	0.00014658	0.00014071	0.00013281	0.00012243	0.00010935	9.3
50	100.87	0.00013149	0.00012646	0.00011968	0.00011071	9.9328e-05	8.5
51	107.17	0.00011812	0.00011382	0.00010799	0.00010023	9.0308e-05	7.
52	113.73	0.00010628	0.00010259	9.756e-05	9.0844e-05	8.2186e-05	7.
53	120.55	9.5767e-05	9.2592e-05	8.8254e-05	8.2429e-05	7.4868e-05	6.
54	127.64	8.642e-05	8.3687e-05	7.9939e-05	7.4885e-05	6.8292e-05	6.0
55	135	8.642e-05	8.3687e-05	7.9939e-05	7.4886e-05	6.8293e-05	6.0

```
% Consumption
st_title = ['MEAN(MN_MPC_W_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_p
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_W_gain_share_check, true, ["mean"], 4, 1, cl_mp_datasetd
```

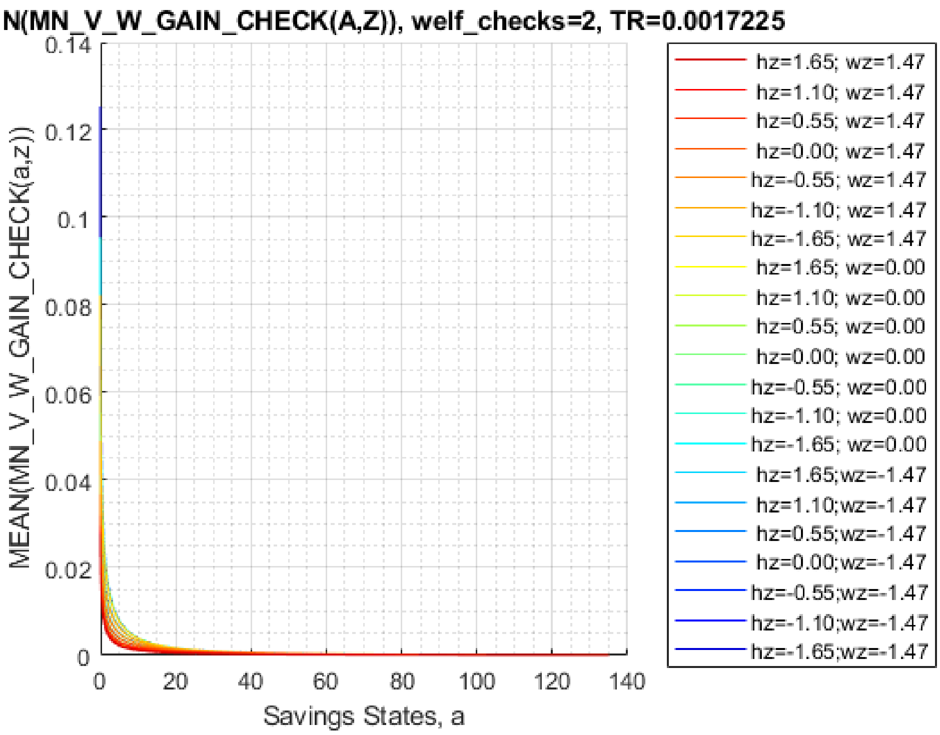
xxx MEAN(MN_MPC_W_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225 xxxxxxxxxxxxxxxxxxxxxxxxxxx

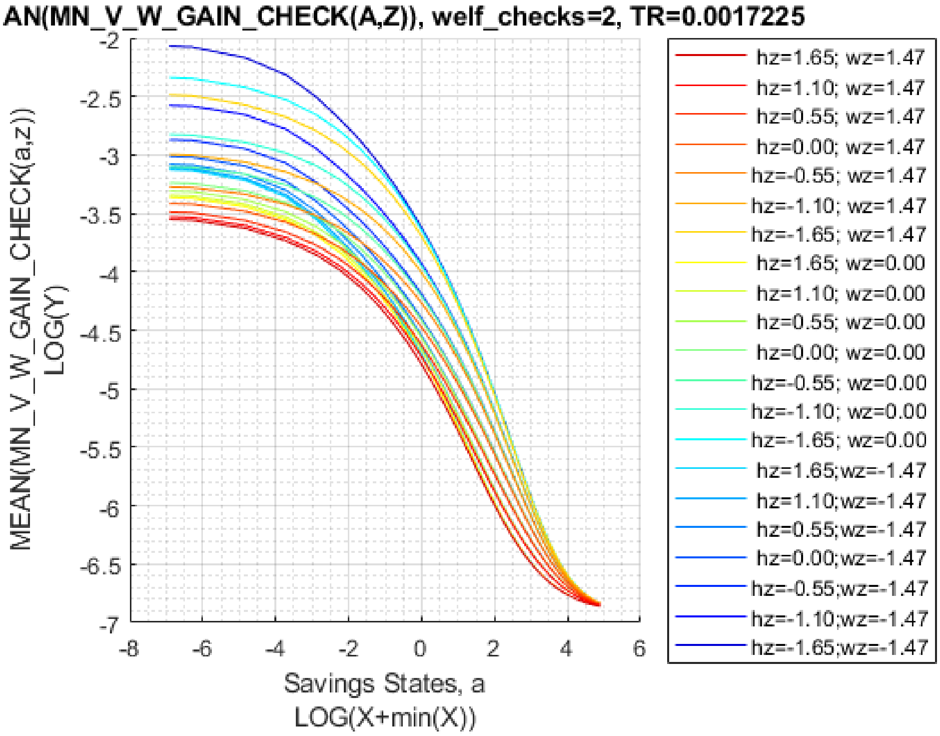
group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5	mea
-----	----------	----------	----------	----------	----------	----------	---
1	0	0.98526	0.94217	0.8526	0.74198	0.63783	0.
2	0.00085734	0.98083	0.93786	0.84881	0.73884	0.63523	0.
3	0.0068587	0.84368	0.82032	0.76961	0.69489	0.60245	0.
4	0.023148	0.74921	0.72805	0.68324	0.61779	0.54083	0.
5	0.05487	0.65324	0.6462	0.61132	0.55665	0.48187	0.
6	0.10717	0.44202	0.45645	0.45163	0.41094	0.33575	0.
7	0.18519	0.33034	0.32383	0.31456	0.29031	0.25187	0.
8	0.29407	0.2813	0.27271	0.25807	0.23769	0.21574	0.
9	0.43896	0.2249	0.21944	0.2088	0.19411	0.17106	0.
10	0.625	0.18874	0.18492	0.17862	0.16796	0.15453	0.
11	0.85734	0.16236	0.16065	0.15616	0.14905	0.14279	0.
12	1.1411	0.14885	0.14631	0.14134	0.13638	0.13247	0.
13	1.4815	0.14162	0.13888	0.13477	0.13128	0.12868	0
14	1.8836	0.13892	0.1348	0.13159	0.12972	0.12474	0
15	2.3525	0.1332	0.13062	0.12766	0.12451	0.12159	0
16	2.8935	0.12454	0.12268	0.11998	0.1182	0.11676	0
17	3.5117	0.11709	0.11733	0.11648	0.11442	0.11309	0.
18	4.2121	0.11435	0.11378	0.11298	0.11334	0.11262	0.
19	5	0.1145	0.11409	0.11341	0.11308	0.11352	0.
20	5.8805	0.11285	0.11256	0.11208	0.11172	0.11225	0.
21	6.8587	0.11083	0.11068	0.11047	0.10985	0.11078	0.
22	7.9398	0.10949	0.1094	0.10929	0.10873	0.10975	0.
23	9.1289	0.11049	0.11045	0.1103	0.11001	0.11084	0.
24	10.431	0.10943	0.10944	0.10941	0.10911	0.10976	0
25	11.852	0.10714	0.10715	0.10724	0.10692	0.10733	0.
26	13.396	0.10662	0.10663	0.1067	0.10651	0.10659	0.
27	15.069	0.10898	0.10898	0.10905	0.10886	0.10872	0.
28	16.875	0.11044	0.11045	0.11051	0.11053	0.11005	0.
29	18.82	0.10911	0.10911	0.10917	0.10934	0.10873	0.
30	20.91	0.10635	0.10635	0.1064	0.10632	0.10602	0
31	23.148	0.10594	0.10595	0.106	0.1059	0.10562	0.
32	25.541	0.10778	0.10778	0.10784	0.10792	0.10752	0.
33	28.093	0.10799	0.10799	0.10804	0.10814	0.10789	0.
34	30.81	0.10767	0.10768	0.10771	0.1078	0.1075	0.
35	33.697	0.10815	0.10815	0.10818	0.10827	0.1081	0.
36	36.758	0.10925	0.10926	0.10928	0.10937	0.10947	0.
37	40	0.10756	0.10757	0.10759	0.10766	0.10784	0.
38	43.427	0.1062	0.10621	0.10623	0.1063	0.10628	0.
39	47.044	0.10582	0.10583	0.10586	0.10592	0.1058	0.
40	50.856	0.10829	0.10831	0.10833	0.1084	0.1084	
41	54.87	0.10898	0.10899	0.10902	0.10908	0.10916	0.

42	59.089	0.10774	0.10775	0.10778	0.10783	0.10792	0.
43	63.519	0.10666	0.10668	0.1067	0.10674	0.10683	0.
44	68.164	0.1073	0.10732	0.10734	0.10738	0.10746	0.
45	73.032	0.1085	0.10851	0.10853	0.10857	0.10864	0.
46	78.125	0.10779	0.1078	0.10782	0.10785	0.10792	0.
47	83.45	0.10631	0.10632	0.10634	0.10637	0.10643	0.
48	89.011	0.10666	0.10667	0.10668	0.10671	0.10677	0.
49	94.815	0.10809	0.1081	0.10812	0.10814	0.1082	0
50	100.87	0.10807	0.10808	0.10809	0.10811	0.10816	0.
51	107.17	0.10728	0.10729	0.1073	0.10732	0.10737	0.
52	113.73	0.10761	0.10762	0.10763	0.10764	0.10768	0.
53	120.55	0.10824	0.10825	0.10825	0.10827	0.10829	0.
54	127.64	0.10739	0.10738	0.10738	0.10736	0.10728	0.
55	135	0.10739	0.10738	0.10738	0.10736	0.10728	0.

Graph Mean Values:

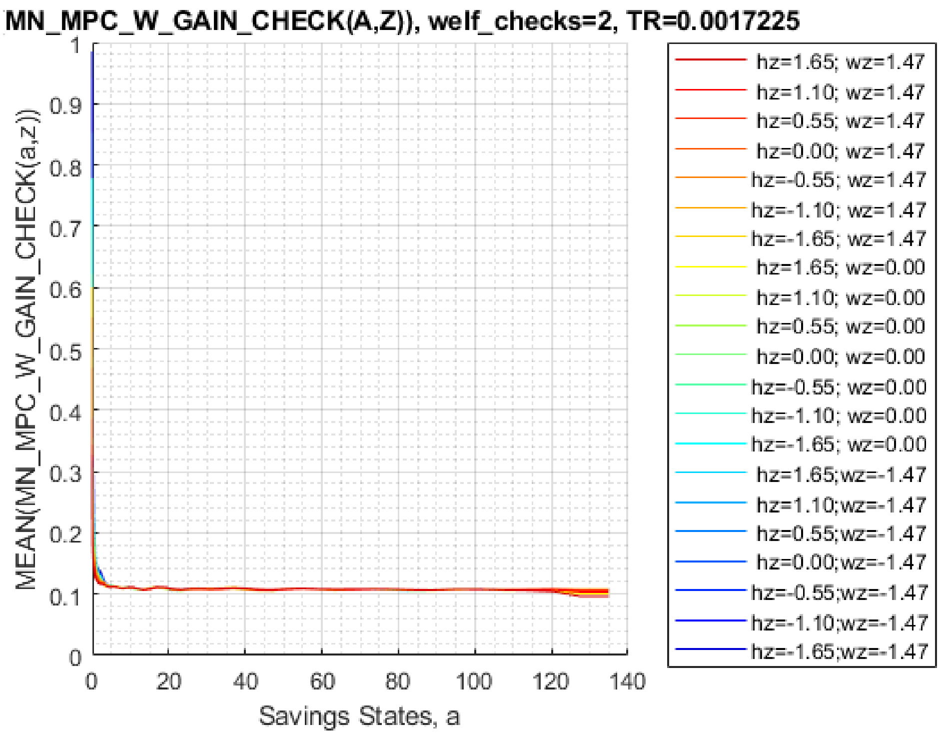
```
st_title = ['MEAN(MN\_V\_W\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_W\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);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

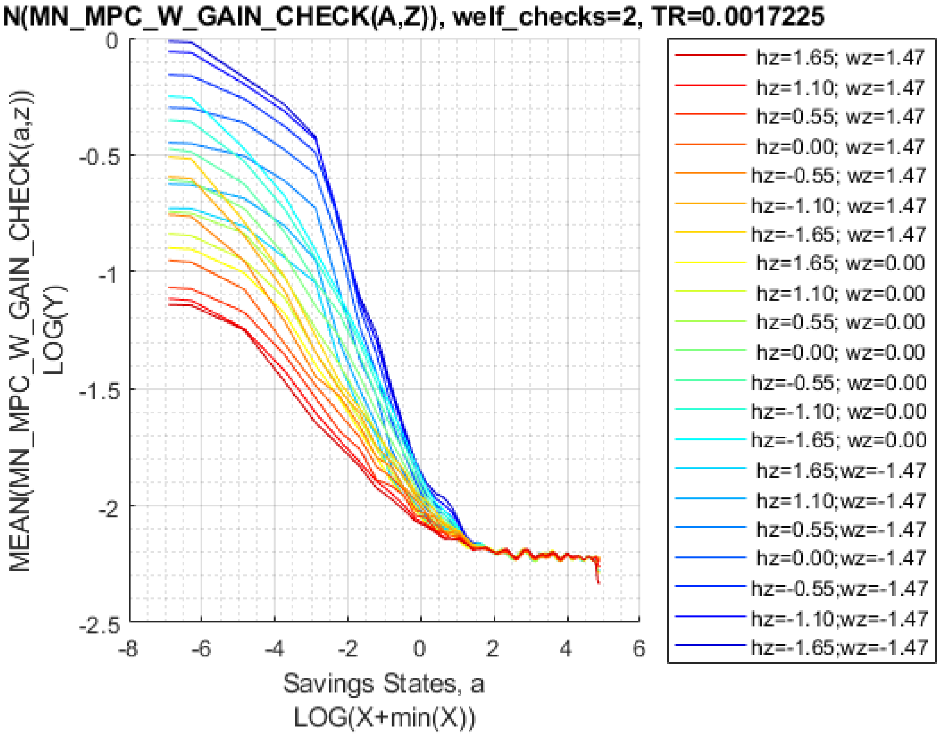




Graph Mean Consumption (***MPC: Share of Check Consumed***):

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





6.1.4 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_W_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_W_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_W_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0         0.010124      0.0096664      0.0091647      0.0084563       0.007865
```

2	2	0	0.013555	0.012956	0.012265	0.01127	0.010434
3	3	0	0.015552	0.014968	0.014264	0.013108	0.012138
4	4	0	0.017504	0.016894	0.01613	0.014825	0.01373
5	5	0	0.019059	0.018461	0.017679	0.016263	0.015076
6	1	1	0.0042913	0.0040316	0.003771	0.0034429	0.003166
7	2	1	0.0057322	0.0053869	0.0050374	0.0045932	0.0042175
8	3	1	0.0067661	0.0063759	0.0059804	0.0054519	0.0050047
9	4	1	0.0080474	0.007599	0.0071398	0.0065203	0.0059876
10	5	1	0.0095567	0.0090861	0.0085927	0.0078601	0.0072337

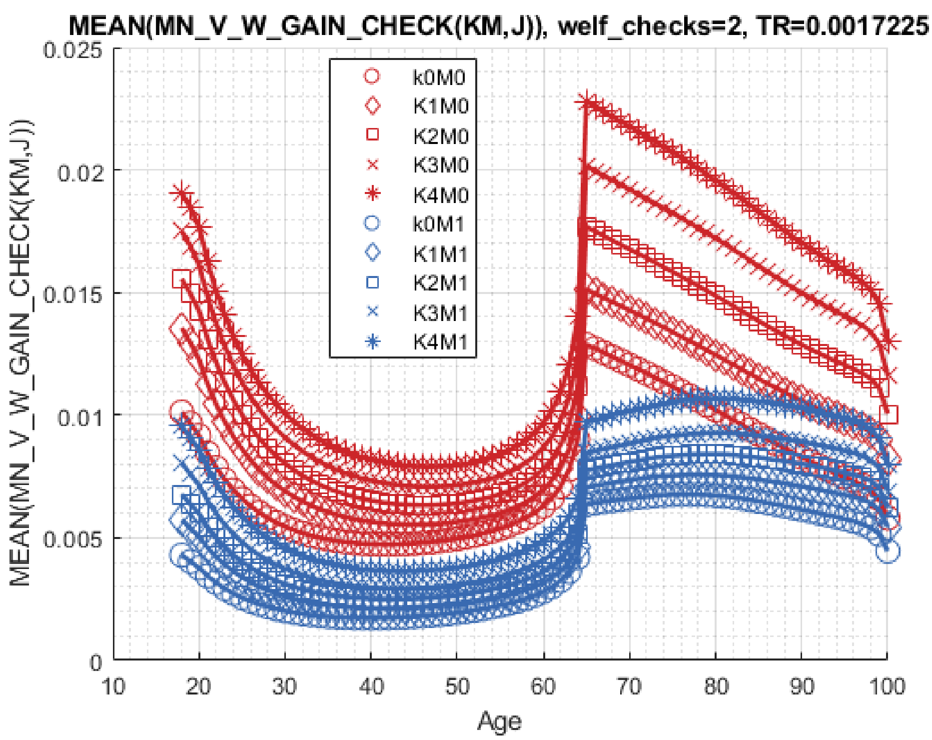
```
% Consumption Function
st_title = ['MEAN(MN_MPC_W_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_W_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd
```

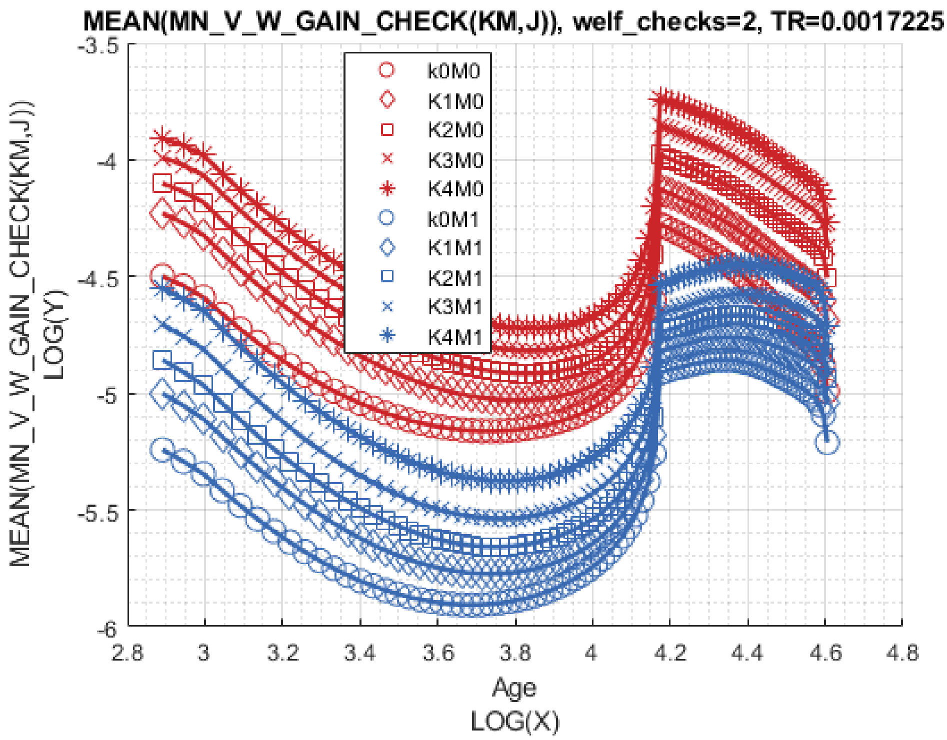
xxx MEAN(MN_MPC_W_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225 xxxxxxxxxxxxxxxxxxxxxxxxxxx

group	kids	marry	mean_age_18	mean_age_19	mean_age_20	mean_age_21	mean_age_2
1	1	0	0.056055	0.062152	0.076579	0.072049	0.069081
2	2	0	0.062949	0.07061	0.08564	0.082109	0.080745
3	3	0	0.068855	0.079381	0.10061	0.099067	0.095297
4	4	0	0.073038	0.087775	0.10721	0.10193	0.10055
5	5	0	0.086493	0.089191	0.11627	0.10754	0.10561
6	1	1	0.087018	0.092174	0.10003	0.094839	0.091444
7	2	1	0.087939	0.094393	0.10299	0.10006	0.098626
8	3	1	0.10007	0.10127	0.11098	0.10622	0.10685
9	4	1	0.099978	0.1049	0.11207	0.11365	0.11202
10	5	1	0.10876	0.11206	0.12109	0.11868	0.1267

Graph Mean Values:

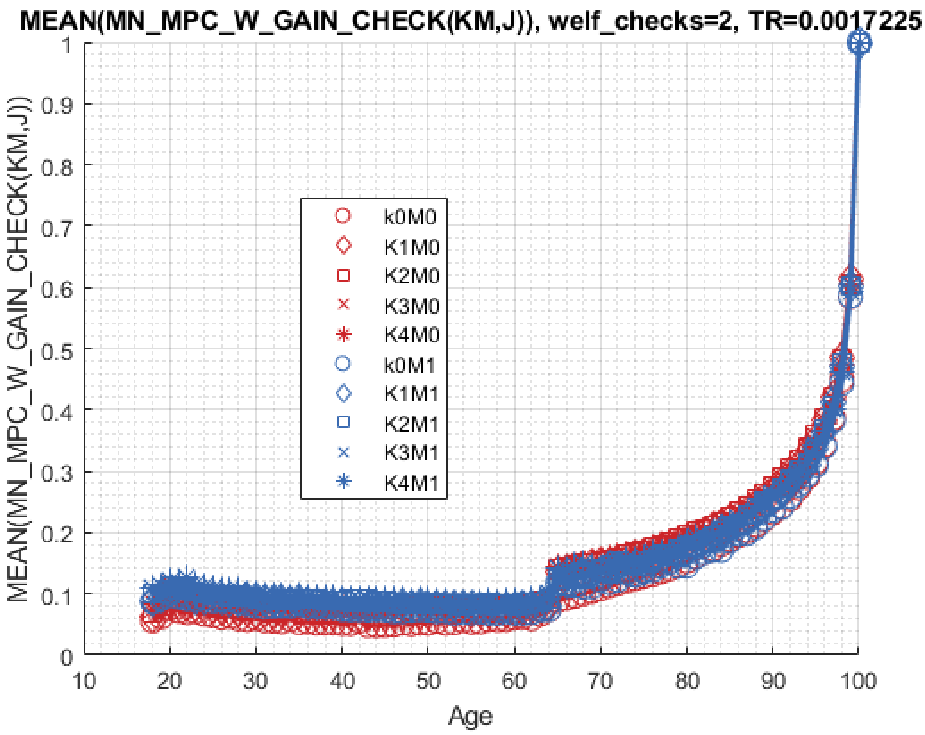
```
st_title = ['MEAN(MN\_V\_W\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_W\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

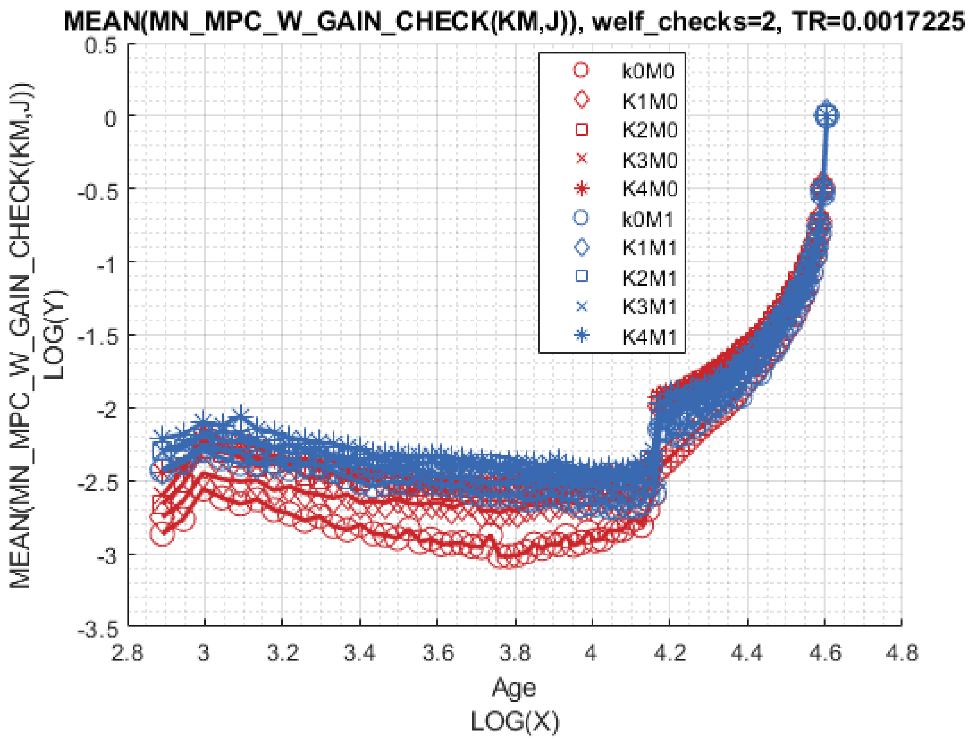




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_W\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_W\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





6.1.5 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_W_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_W_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_
```

```
xxx  MEAN(MN_V_W_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0       0.015898       0.015428        0.01488       0.014086        0.01338
      2       1       0        0.01442       0.013751       0.012921       0.011482       0.010317
      3       0       1      0.0074423      0.0070662      0.0066934       0.006237      0.0058341
      4       1       1      0.0063152      0.0059256       0.005515      0.0049104      0.0044097
```

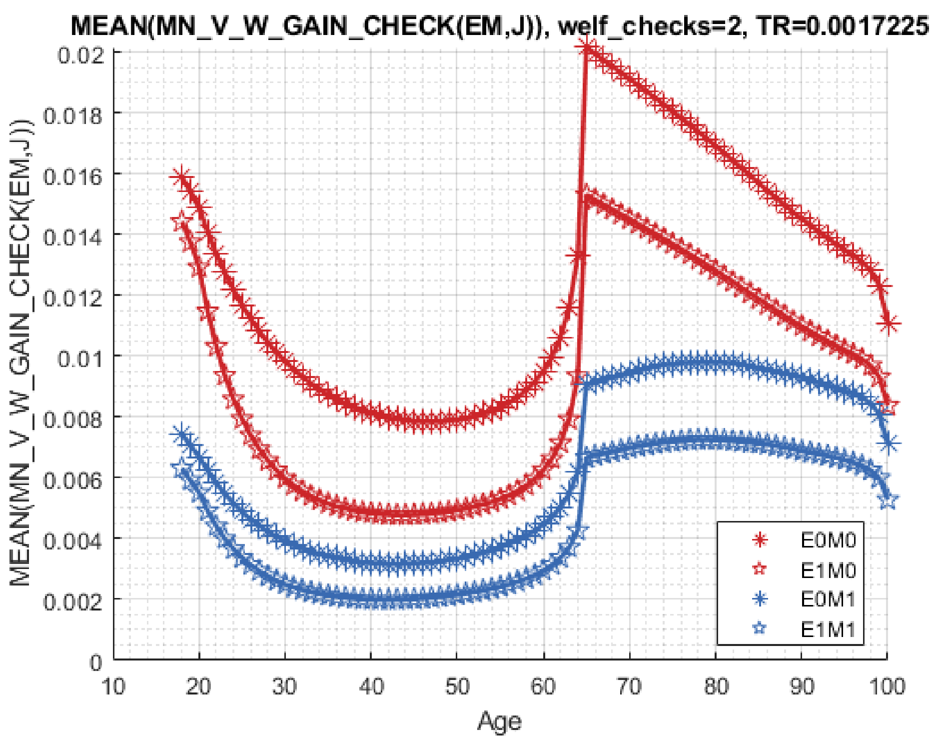
```
% Consumption
st_title = ['MEAN(MN_MPC_W_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
```

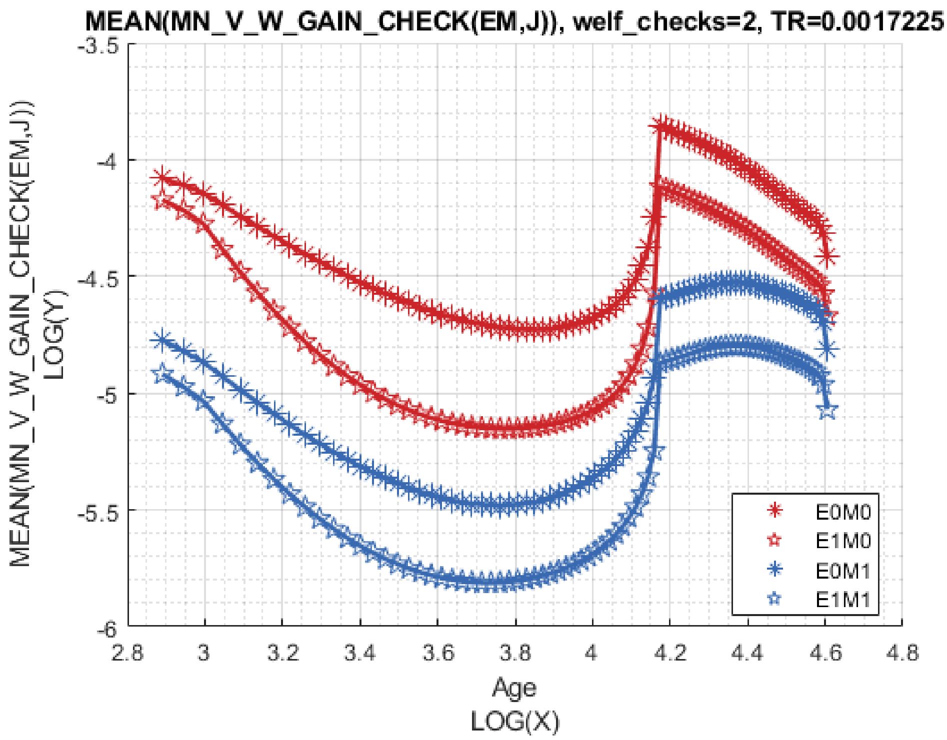
```
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_W_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd
```

```
xxx  MEAN(MN_MPC_W_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
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0       0.059933       0.065486       0.078043       0.076386       0.076149
      2       1       0       0.079022       0.090158        0.11648        0.10869        0.10436
      3       0       1       0.088005       0.089818       0.094532       0.093334       0.094468
      4       1       1         0.1055         0.1121        0.12433        0.12005        0.11979
```

Graph Mean Values:

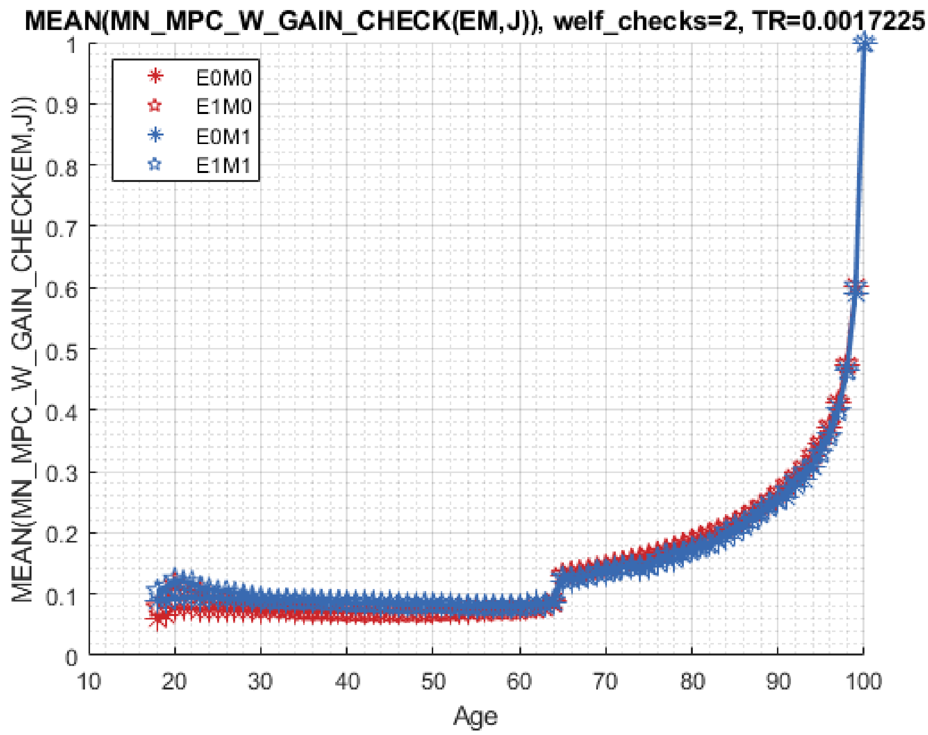
```
st_title = ['MEAN(MN\_V\_W\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_W\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

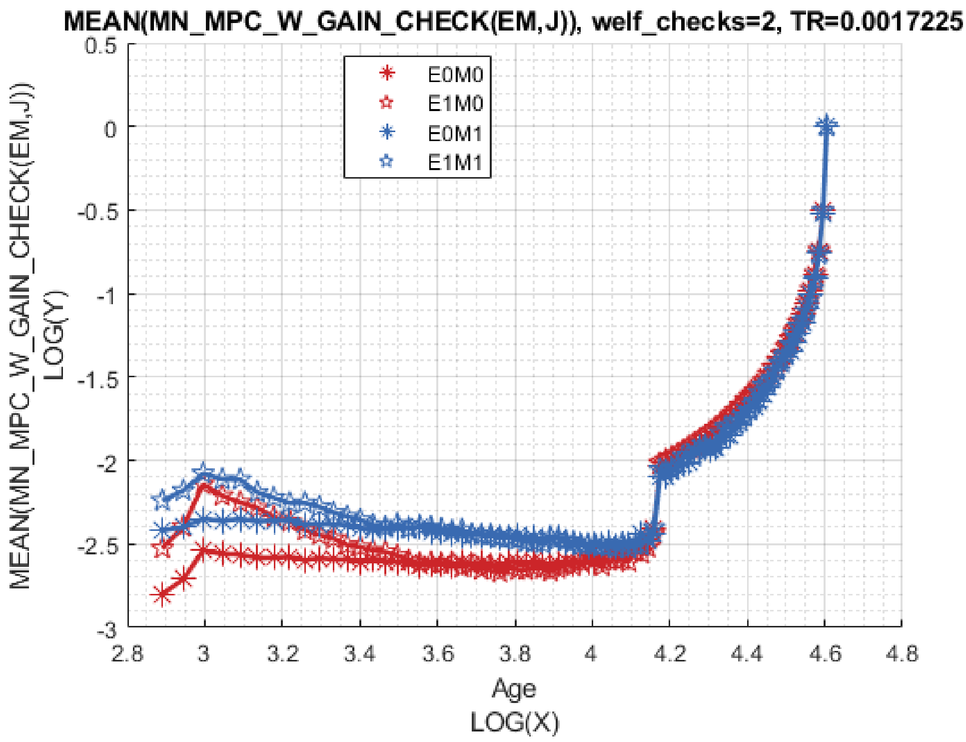




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_W\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_MPC\_W\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_c{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);
```





6.2 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.

6.2.1 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
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_container_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                        i    idx    ndim      numel       rowN    colN         sum          mean
                        -    ---    ----    ----------    ----    -----    -----------    ---------

    C_U                 1     1      6      1.9173e+06     83     23100     9.1143e+06        4.753
    C_U_minus_C_unemp   2     2      6      1.9173e+06     83     23100         1264.8    0.0006596
    V_U                 3     3      6      1.9173e+06     83     23100    -4.8023e+06       -2.504
    V_U_minus_V_unemp   4     4      6      1.9173e+06     83     23100          20117      0.01049
    mn_MPC_unemp        5     5      6      1.9173e+06     83     23100     3.6714e+05       0.1914
```

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

6.2.2 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=%3.2
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});
```

6.2.3 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(mp_par
```

```
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_
```

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

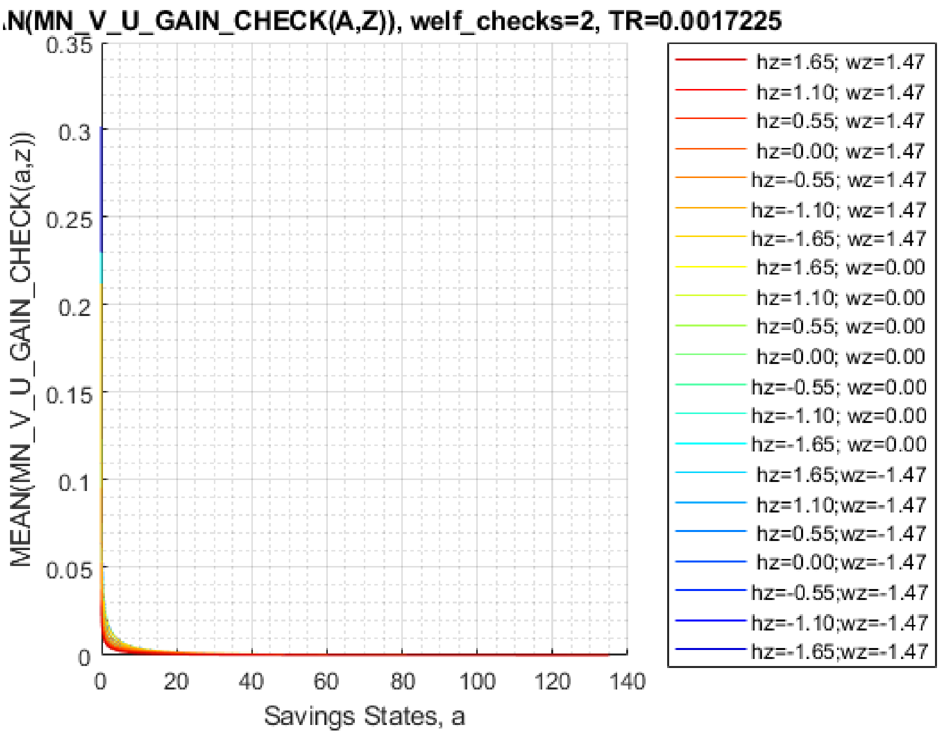
group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5	mea
1	0	0.30204	0.14591	0.083016	0.058265	0.048632	0
2	0.00085734	0.29687	0.14443	0.08243	0.057893	0.04831	0
3	0.0068587	0.24517	0.12826	0.075796	0.053707	0.044719	
4	0.023148	0.17496	0.10278	0.064774	0.046886	0.039	0
5	0.05487	0.11411	0.075197	0.05168	0.038779	0.032381	0
6	0.10717	0.07945	0.053698	0.03981	0.031363	0.026553	0
7	0.18519	0.061381	0.041221	0.031035	0.025388	0.021951	0
8	0.29407	0.049672	0.033665	0.025164	0.020645	0.018059	0
9	0.43896	0.04097	0.028356	0.021184	0.017207	0.014985	
10	0.625	0.03421	0.024175	0.018158	0.014629	0.012593	0
11	0.85734	0.028804	0.020748	0.015685	0.012564	0.010694	0.
12	1.1411	0.024415	0.017898	0.013612	0.010849	0.0091391	0.
13	1.4815	0.020781	0.015487	0.011845	0.009402	0.0078443	0.
14	1.8836	0.017732	0.013429	0.01033	0.0081694	0.0067529	0
15	2.3525	0.015155	0.011653	0.0090184	0.0071161	0.005829	0.
16	2.8935	0.012965	0.010113	0.0078806	0.0062113	0.0050446	0.
17	3.5117	0.011102	0.0087778	0.0068893	0.0054304	0.0043768	0.
18	4.2121	0.0095159	0.0076201	0.0060242	0.0047539	0.0038073	0.
19	5	0.0081617	0.0066157	0.0052682	0.0041656	0.0033198	0.
20	5.8805	0.0070048	0.0057428	0.0046071	0.003653	0.002901	0.
21	6.8587	0.0060148	0.004985	0.0040287	0.0032059	0.0025406	0.
22	7.9398	0.0051679	0.0043275	0.0035243	0.0028158	0.0022296	0.
23	9.1289	0.0044436	0.0037576	0.0030842	0.0024752	0.0019606	0.
24	10.431	0.0038243	0.0032641	0.0027001	0.002178	0.0017273	0.
25	11.852	0.0032946	0.0028369	0.0023652	0.0019184	0.0015245	0.
26	13.396	0.0028417	0.0024674	0.0020734	0.0016914	0.0013477	0.
27	15.069	0.0024542	0.0021479	0.001819	0.0014925	0.0011933	0.0
28	16.875	0.0021225	0.0018716	0.0015969	0.0013182	0.0010581	0.0
29	18.82	0.0018382	0.0016324	0.0014031	0.0011655	0.00093959	0.0
30	20.91	0.0015944	0.0014253	0.0012338	0.0010316	0.00083546	0.0
31	23.148	0.001385	0.0012459	0.001086	0.00091386	0.00074382	0.0
32	25.541	0.0012051	0.0010904	0.00095674	0.00081028	0.00066306	0.0
33	28.093	0.0010502	0.00095561	0.00084375	0.00071923	0.0005918	0.0
34	30.81	0.00091678	0.00083855	0.00074491	0.00063906	0.00052877	0.0
35	33.697	0.00080165	0.00073685	0.00065837	0.00056838	0.00047287	0.0
36	36.758	0.00070216	0.0006484	0.00058255	0.000506	0.00042333	0.
37	40	0.0006161	0.00057139	0.00051608	0.00045093	0.00037946	0.0
38	43.427	0.00054153	0.00050427	0.00045774	0.00040227	0.0003405	0.0
39	47.044	0.00047681	0.0004457	0.0004065	0.00035923	0.00030581	0.0
40	50.856	0.00042057	0.00039453	0.00036145	0.00032113	0.0002749	0.0
41	54.87	0.00037159	0.00034976	0.00032182	0.00028739	0.0002474	0.
42	59.089	0.00032891	0.00031056	0.00028689	0.00025747	0.00022288	0.0
43	63.519	0.00029162	0.00027616	0.00025609	0.00023092	0.00020099	0.0
44	68.164	0.000259	0.00024595	0.00022891	0.00020735	0.00018142	0.0
45	73.032	0.00023043	0.0002194	0.00020489	0.00018639	0.00016392	0.0
46	78.125	0.00020535	0.00019599	0.00018363	0.00016774	0.00014826	0.0
47	83.45	0.0001833	0.00017535	0.00016479	0.00015113	0.00013422	0.0
48	89.011	0.00016388	0.00015711	0.00014808	0.00013632	0.00012163	0.0
49	94.815	0.00014675	0.00014099	0.00013325	0.00012311	0.00011033	9.5
50	100.87	0.00013164	0.00012671	0.00012006	0.0001113	0.00010017	8.6
51	107.17	0.00011825	0.00011403	0.00010832	0.00010074	9.1041e-05	7.9
52	113.73	0.00010639	0.00010277	9.7844e-05	9.1281e-05	8.2823e-05	7.2

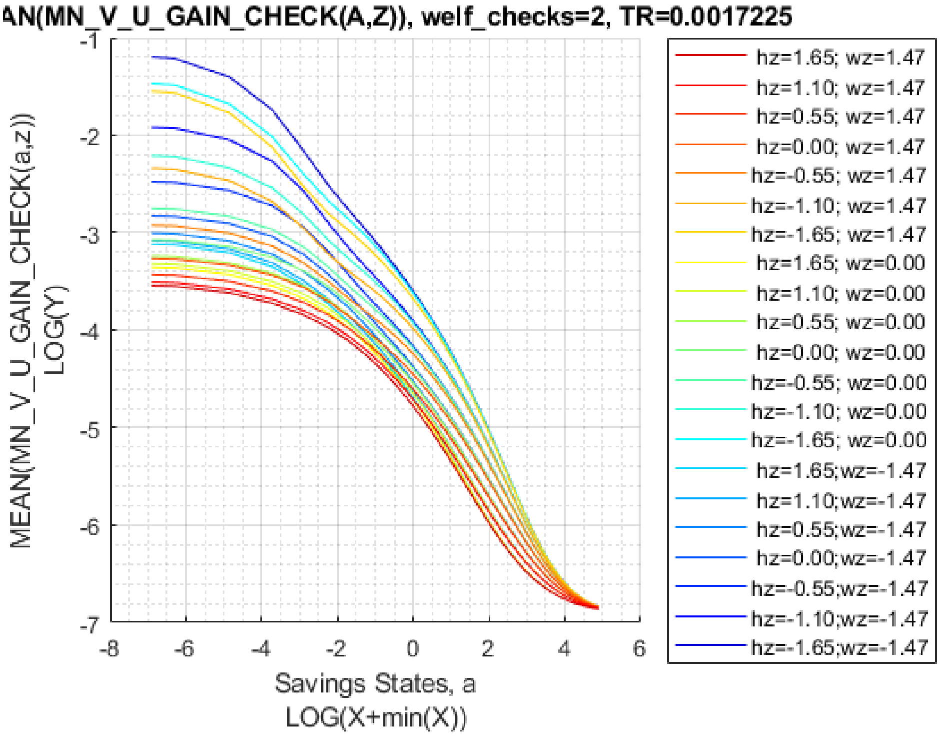
```
    53          120.55      9.586e-05     9.2745e-05     8.8498e-05     8.2806e-05     7.5422e-05     6.6
    54          127.64       8.65e-05     8.3818e-05     8.0149e-05     7.5211e-05      6.877e-05     6.0
    55             135       8.65e-05     8.3818e-05     8.0149e-05     7.5211e-05      6.877e-05     6.0

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

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     mea
    -----    ----------     ----------     ----------     ----------     ----------     ----------     ---

      1               0      0.99676        0.99676        0.98968        0.96861          0.936        0.
      2      0.00085734      0.99564        0.99564         0.9885        0.96775        0.93479        0.
      3       0.0068587      0.96608        0.96608        0.96111        0.93769        0.90391         0
      4        0.023148       0.9144        0.91442        0.90273        0.87976        0.84669        0.
      5         0.05487      0.83014         0.8675        0.85172        0.82953        0.79856        0.
      6         0.10717      0.57928         0.6794        0.72166        0.70603        0.67574        0.
      7         0.18519      0.40677         0.4866        0.57883        0.62471        0.61322        0.
      8         0.29407      0.30075        0.32748        0.40471        0.48963        0.53386        0.
      9         0.43896      0.23989        0.24678        0.26581        0.32826        0.40159        0.
     10           0.625      0.20277        0.20016        0.20528        0.22484        0.27796        0.
     11         0.85734      0.17483        0.17446        0.16482        0.16705        0.18499        0.
     12          1.1411      0.15256        0.15408        0.15362        0.14656        0.14374        0.
     13          1.4815      0.13901        0.14193         0.1396        0.13759        0.13213        0.
     14          1.8836      0.13445        0.13173        0.13532        0.13183        0.12918        0.
     15          2.3525      0.13073        0.12804        0.12598        0.12513        0.12402        0.
     16          2.8935      0.12611        0.12259        0.11944        0.11919        0.11786        0.
     17          3.5117      0.11866        0.11921        0.11564        0.11369        0.11418        0.
     18          4.2121      0.11801        0.11572        0.11569        0.11296        0.11197         0
     19               5      0.11495         0.1165        0.11547        0.11464        0.11333        0.
     20          5.8805      0.11283        0.11351        0.11397        0.11371        0.11268        0.
     21          6.8587      0.11082        0.11067        0.11238        0.11122        0.11137        0.
     22          7.9398      0.10947        0.10939        0.10939        0.11073         0.1094         0
     23          9.1289      0.11048        0.11044        0.11055        0.11147         0.1111        0.
     24          10.431      0.10942        0.10943        0.10956        0.10966         0.1103        0.
     25          11.852      0.10713        0.10714        0.10723         0.1074        0.10779        0.
     26          13.396      0.10662        0.10662        0.10669        0.10689        0.10696        0.
     27          15.069      0.10897        0.10898        0.10904        0.10931        0.10944        0.
     28          16.875      0.11044        0.11044         0.1105        0.11079        0.11094        0.
     29           18.82       0.1091        0.10911        0.10916        0.10934        0.10959        0.
     30           20.91      0.10634        0.10635        0.10639        0.10655        0.10686        0.
     31          23.148      0.10594        0.10594        0.10599        0.10622        0.10646        0.
     32          25.541      0.10777        0.10778        0.10783        0.10802        0.10823        0.
     33          28.093      0.10798        0.10799        0.10803        0.10814        0.10838        0.
     34           30.81      0.10767        0.10768        0.10771         0.1078        0.10808        0.
     35          33.697      0.10814        0.10815        0.10818        0.10827        0.10861        0.
     36          36.758      0.10925        0.10925        0.10928        0.10936        0.10964        0.
     37              40      0.10756        0.10757        0.10759        0.10766        0.10783        0.
     38          43.427       0.1062        0.10621        0.10623        0.10629        0.10646         0
     39          47.044      0.10582        0.10583        0.10586        0.10592        0.10616        0.
     40          50.856      0.10829         0.1083        0.10833        0.10839        0.10861        0.
     41           54.87      0.10898        0.10899        0.10902        0.10908        0.10921        0.
     42          59.089      0.10774        0.10775        0.10777        0.10782        0.10792         0
     43          63.519      0.10666        0.10668         0.1067        0.10674        0.10683        0.
     44          68.164       0.1073        0.10731        0.10734        0.10738        0.10746        0.
     45          73.032       0.1085        0.10851        0.10853        0.10857        0.10864        0.
     46          78.125      0.10779         0.1078        0.10782        0.10785        0.10792        0.
```

47	83.45	0.10631	0.10632	0.10634	0.10637	0.10643	0.
48	89.011	0.10666	0.10667	0.10668	0.10671	0.10677	0.
49	94.815	0.10809	0.1081	0.10811	0.10814	0.1082	0.
50	100.87	0.10807	0.10808	0.10809	0.10811	0.10816	0.
51	107.17	0.10728	0.10729	0.1073	0.10732	0.10737	0.
52	113.73	0.10761	0.10762	0.10763	0.10764	0.10768	0.
53	120.55	0.10824	0.10825	0.10825	0.10827	0.10829	0.
54	127.64	0.10739	0.10738	0.10738	0.10736	0.10728	0.
55	135	0.10739	0.10738	0.10738	0.10736	0.10728	0.

Graph Mean Values:

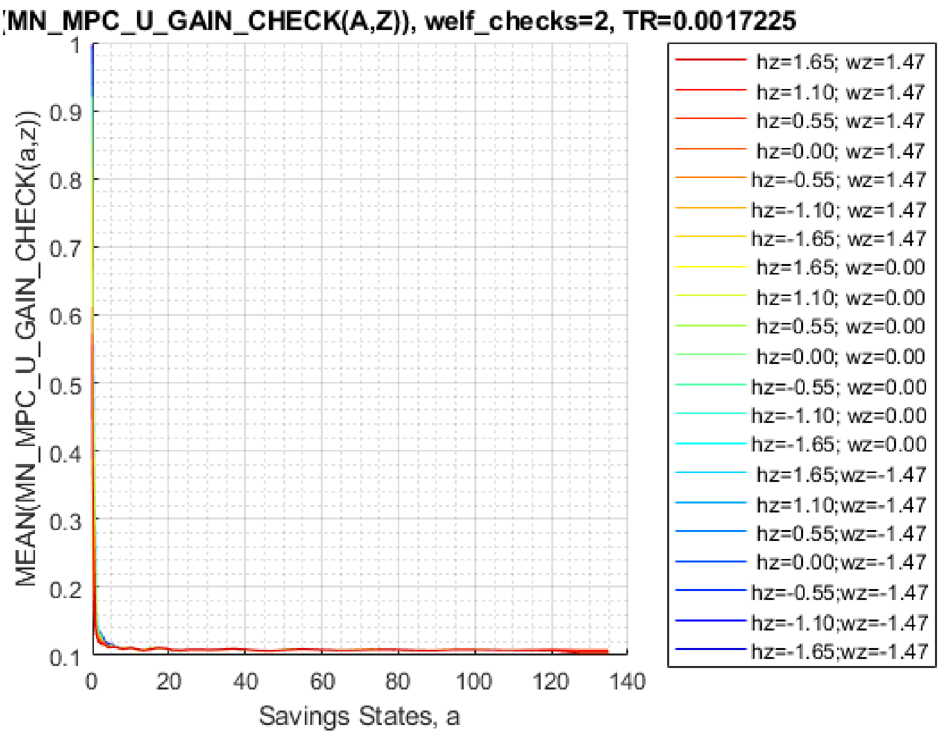
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);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

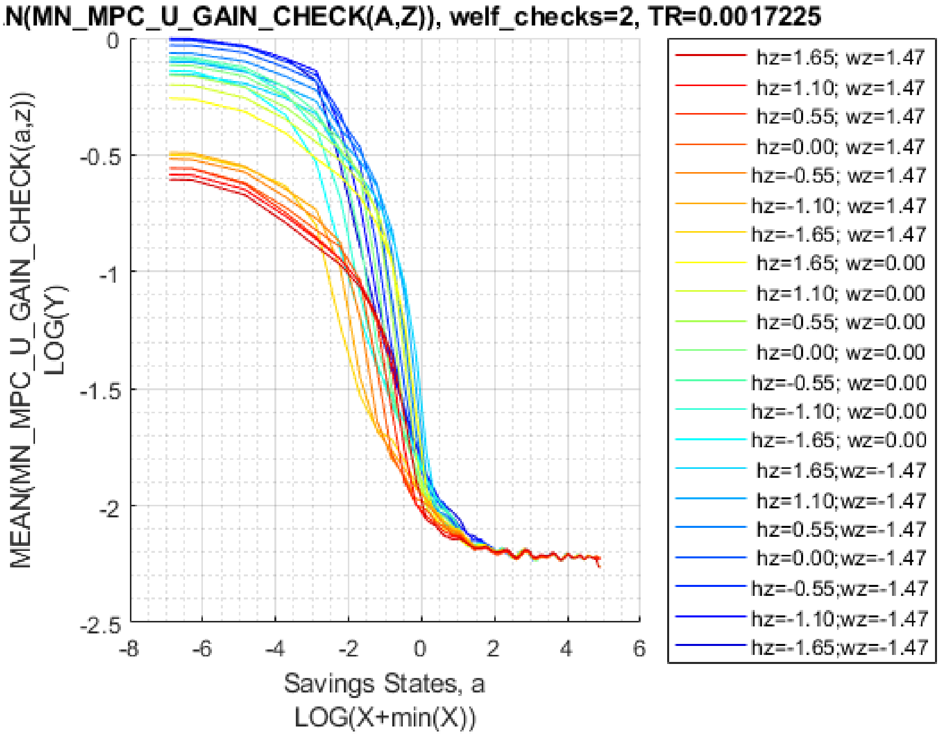




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str
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);
```





6.2.4 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(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

xxx  MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0       0.018051       0.017729       0.017395       0.015889       0.014636
```

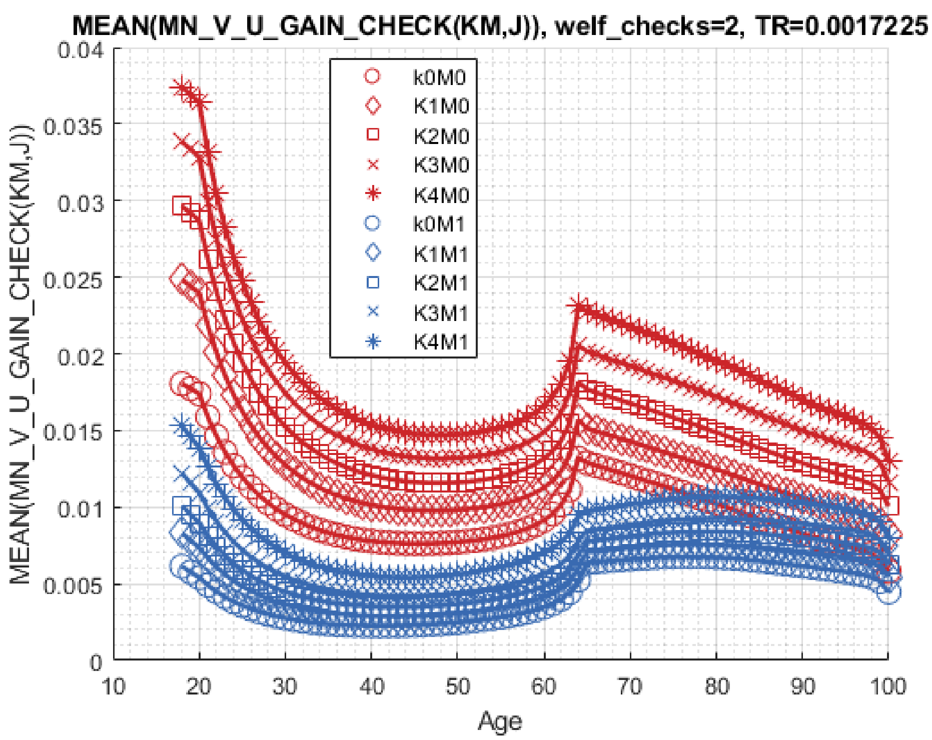
```
         2         2        0        0.024841       0.024418       0.023963       0.021844       0.020076
         3         3        0        0.029625       0.029186       0.028697       0.026156       0.024037
         4         4        0        0.033827       0.033361        0.03283       0.029923         0.0275
         5         5        0        0.037408        0.03694       0.036392       0.033181       0.030504
         6         1        1       0.0061332      0.0057595      0.0054053      0.0049063       0.004487
         7         2        1       0.0083066      0.0078178       0.007352      0.0066702      0.0060896
         8         3        1        0.010082      0.0095157      0.0089749      0.0081404       0.007436
         9         4        1        0.012214       0.011567       0.010941      0.0099258      0.0090678
        10         5        1        0.015297       0.014584       0.013875       0.012596       0.011524
```

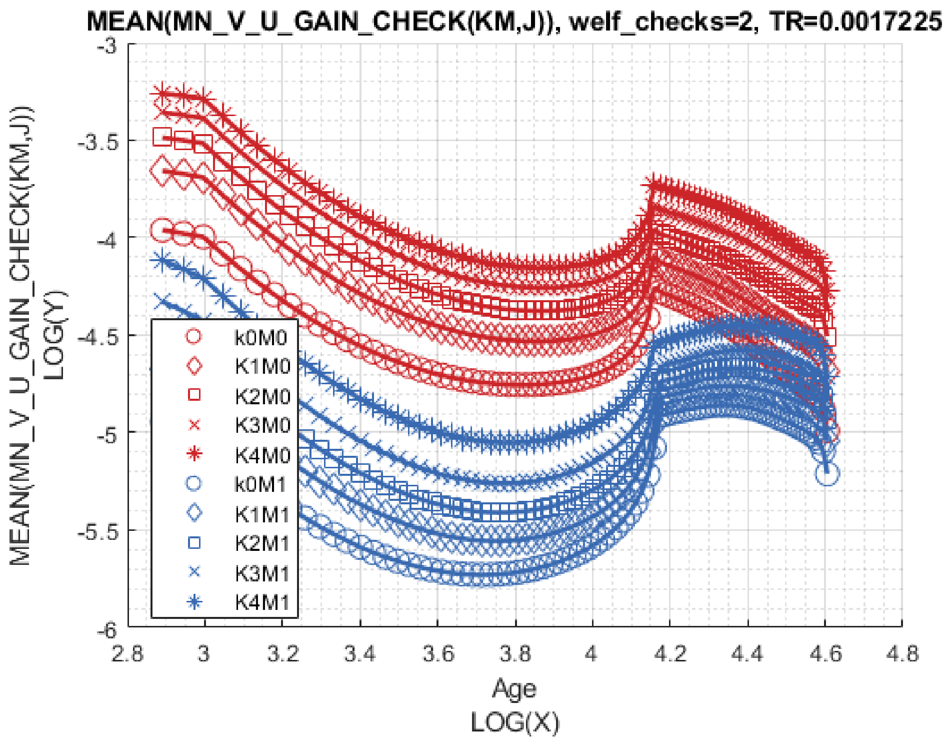
```
% Consumption Function
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd
```

```
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_2
    -----     ----     -----     -----------    -----------    -----------    -----------    ----------

      1         1        0        0.15281        0.15636        0.16001        0.16042        0.16057
      2         2        0        0.16175         0.1653        0.16916        0.17012        0.17091
      3         3        0        0.16911        0.17246         0.1763        0.17733        0.17815
      4         4        0        0.17289        0.17613        0.17991        0.18092        0.18173
      5         5        0         0.1764        0.17944        0.18311        0.18398        0.18469
      6         1        1        0.13726        0.14236        0.14341        0.14404        0.14148
      7         2        1        0.14642        0.14777        0.14963        0.15056        0.14937
      8         3        1        0.15601        0.15841         0.1601        0.16253        0.15944
      9         4        1        0.16238         0.1649        0.16712        0.16745         0.1673
     10         5        1        0.17026         0.1742        0.17738        0.17809        0.17684
```

Graph Mean Values:

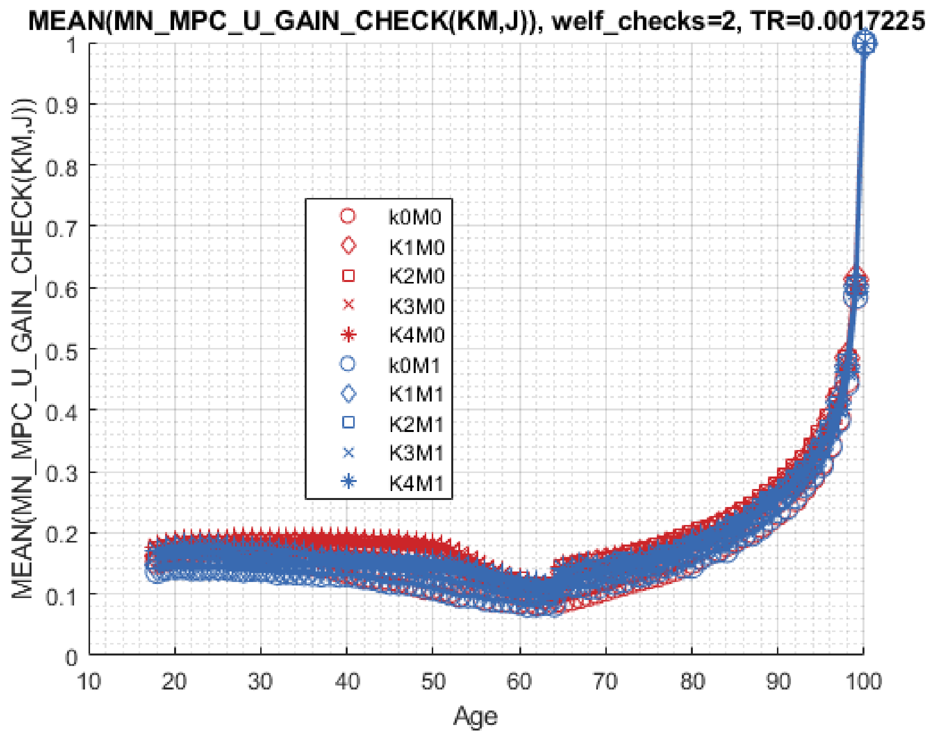
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

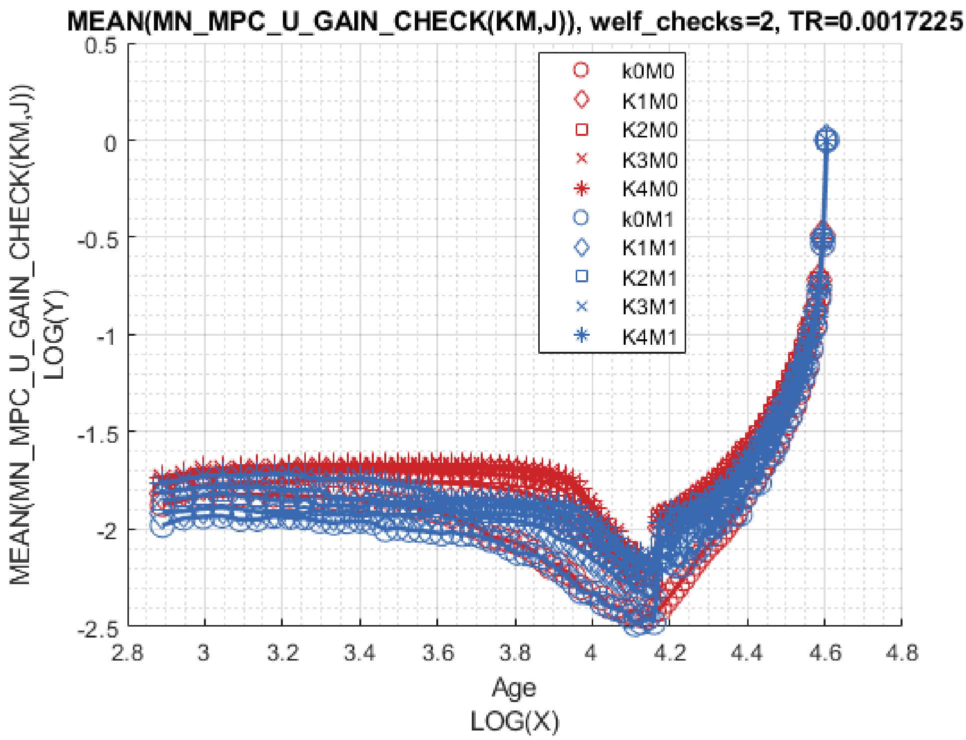




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
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);
```





6.2.5 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:

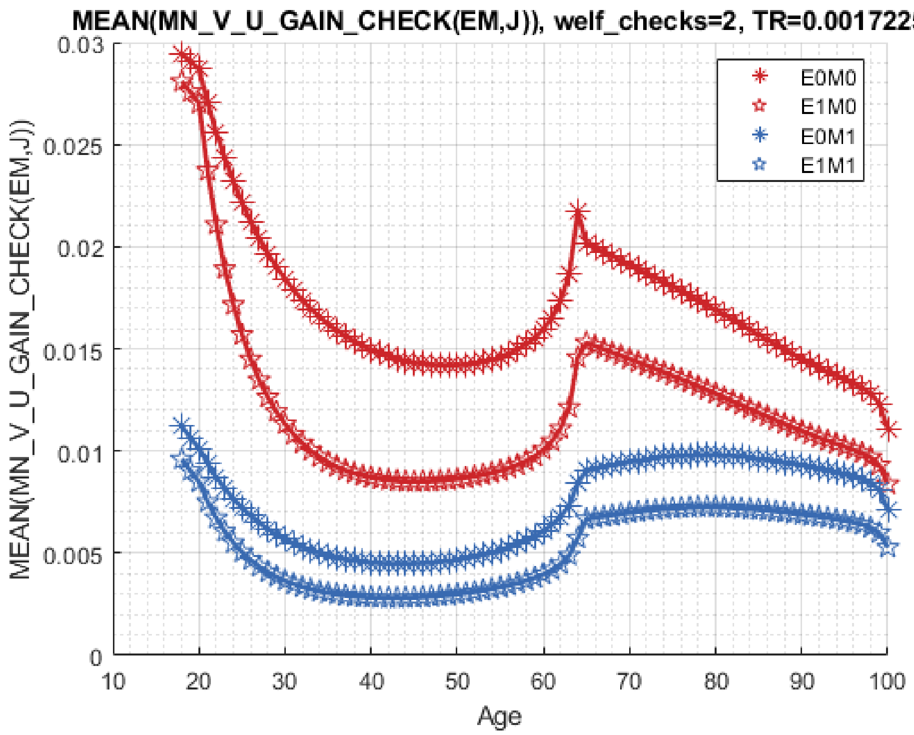
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,6,1,4,5];
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3,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

xxx  MEAN(MN_V_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

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

% Consumption
st_title = ['MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_
```

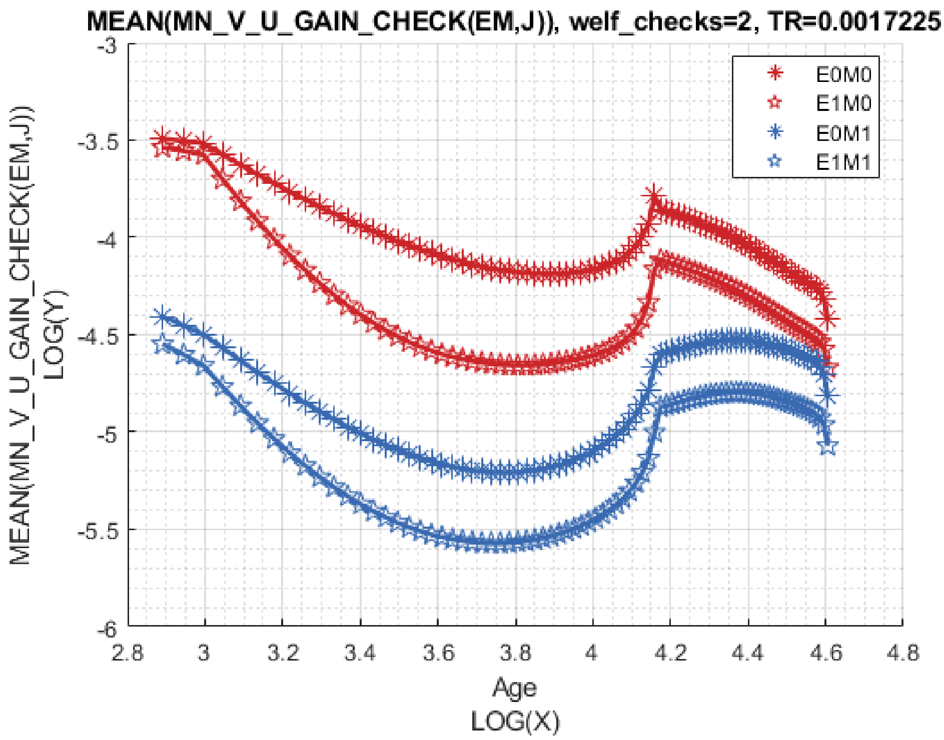
```
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_U_gain_share_check, true, ["mean"], 3, 1, cl_mp_datasetd
```

```
xxx  MEAN(MN_MPC_U_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

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

Graph Mean Values:

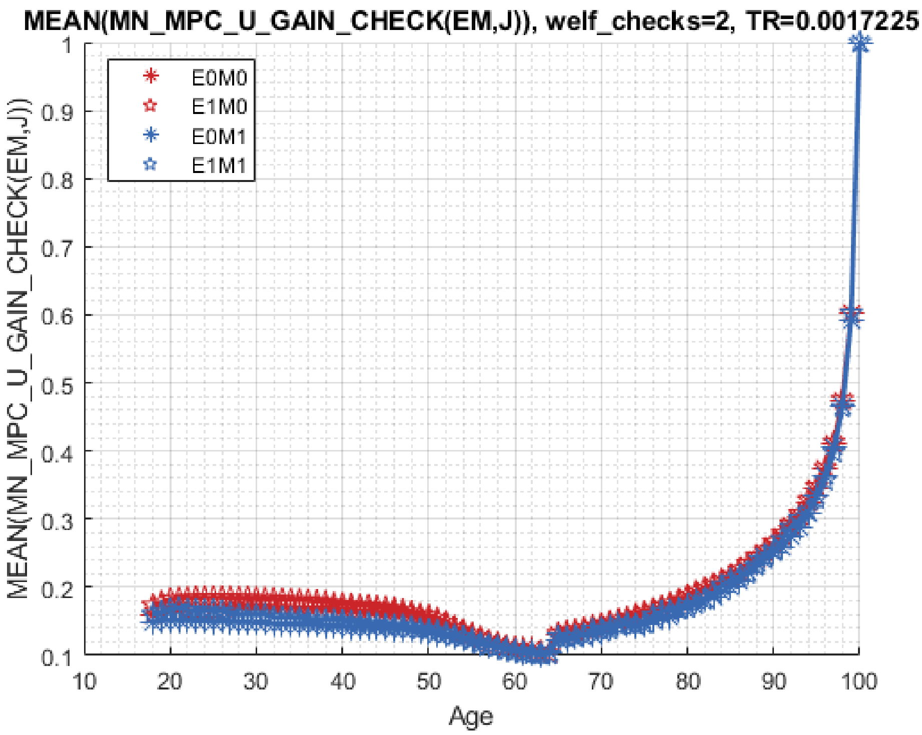
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

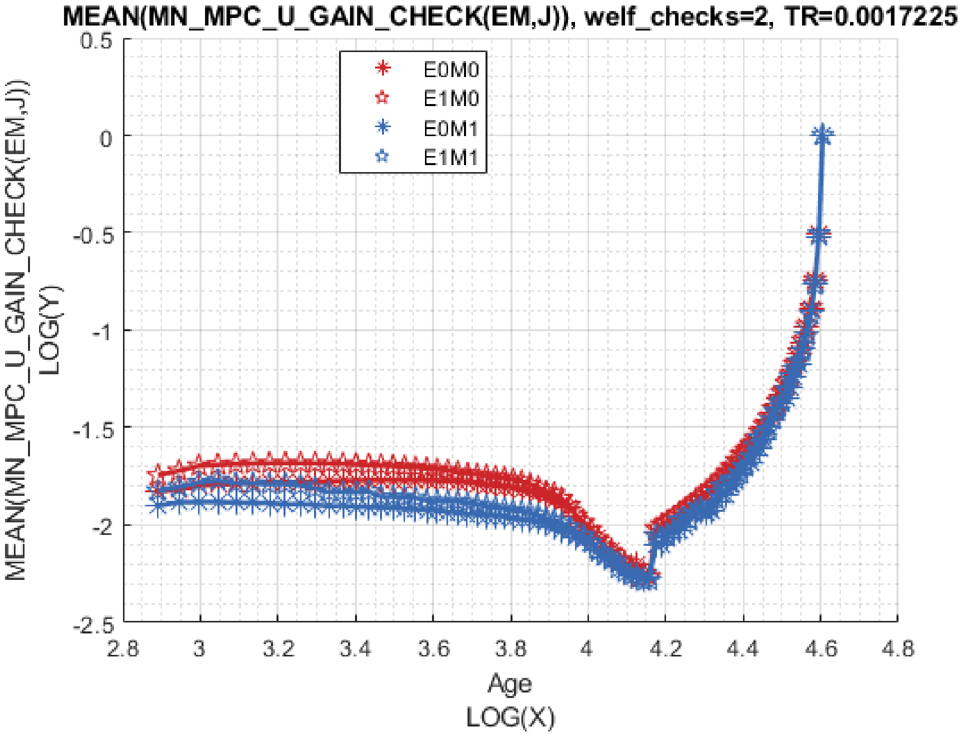




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
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))
LOG(Y)
Age
LOG(X)
0.5
0
-0.5
-1
-1.5
-2
-2.5
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
4.8

Chapter 7

2020 Outcomes Full State Space with Savings, Shocks and Education

7.1 2020 Full States EV and EC of One Check

This is the example vignette for function: **snw_evuvw20_jaeemk** from the **PrjOptiSNW Package.** 2020 integrated over VU and VW

7.1.1 Test SNW_EVUVW20_JAEEMK Defaults Dense

VFI and Distribution

Call the function with defaults.

```
clear all;
st_solu_type = 'bisec_vec';

% Solve the VFI Problem and get Value Function
mp_params = snw_mp_param('default_docdense');
mp_controls = snw_mp_control('default_test');

% set Unemployment Related Variables
xi=0.5; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=1
b=0; % Unemployment insurance replacement rate (b=0 refers to no UI benefits; b=1 refers to 100 perc
TR=100/58056; % Value of a welfare check (can receive multiple checks). TO DO: Update with alternati

mp_params('xi') = xi;
mp_params('b') = b;
mp_params('TR') = TR;

% Solve for Unemployment Values
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;

% Solve the Model to get V working and unemployed
[V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time=20.6119
```

```
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;
% Solve unemployment
[V_unemp,~,cons_unemp,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=defa
```

```
[Phi_true] = snw_ds_main(mp_params, mp_controls, ap_ss, cons_ss, mp_valpol_more_ss);
```

```
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time=51.8429
```

```
% Get Matrixes
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_ss, Phi_tr
```

```
Wage quintile cutoffs=0.49295     0.79302      1.3138      2.1063
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time cost=23.0315
```

7.1.2 Solve for 2020 Evuvw With 0 and 2 Checks

```
% Call Function
welf_checks = 0;
[ev20_jaeemk_check0, ec20_jaeemk_check0] = snw_evuvw20_jaeemk(...
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss, cons_ss, V_unemp, cons_unemp, mp_precompute_res);
```

```
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=0;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_dense
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=0;TR=0.0017225;SNW_MP_PARAM=default_dense;SNW_MP_CONTR
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;timeEUEC=0.64818
```

```
% Call Function
welf_checks = 2;
[ev20_jaeemk_check2, ec20_jaeemk_check2] = snw_evuvw20_jaeemk(...
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss, cons_ss, V_unemp, cons_unemp, mp_precompute_res);
```

```
Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=2;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_dense
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=2;TR=0.0017225;SNW_MP_PARAM=default_dense;SNW_MP_CONTR
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;timeEUEC=0.44581
```

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check = ev20_jaeemk_check2 - ev20_jaeemk_check0;
mn_MPC_U_gain_share_check = (ec20_jaeemk_check2 - ec20_jaeemk_check0)./(welf_checks*mp_params('TR'))
```

7.1.3 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=%3.2
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});
```

7.1.4 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(mp_par
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_
```

```
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    mea
    _____    __________    __________    __________    __________    __________    __________    ___

      1               0       0.27912       0.11616      0.067433      0.053393      0.049158      0
      2      0.00085734       0.27485       0.11517      0.066984      0.053023      0.048799      0
      3       0.0068587       0.23632       0.10481      0.062008      0.048905      0.044801
      4        0.023148        0.1863       0.08898      0.054024      0.042348      0.038458      0
      5         0.05487       0.14109      0.071528      0.044733      0.034862      0.031271      0
      6         0.10717       0.10829      0.056442      0.036039      0.027915      0.024677      0
      7         0.18519      0.087242      0.046352       0.02987      0.023041      0.020157      0
      8         0.29407      0.072208      0.039367      0.025433      0.019424      0.016782      0
      9         0.43896      0.060221      0.033761      0.021874      0.016469      0.014026      0
     10           0.625      0.050442      0.029129      0.018949      0.014075      0.011802      0
     11         0.85734      0.042316      0.025218      0.016484      0.012083     0.0099683      0.
     12          1.1411      0.035502      0.021857      0.014374      0.010409     0.0084519      0.
     13          1.4815      0.029772       0.01894      0.012549     0.0089945     0.0071841      0.
     14          1.8836      0.024956      0.016396      0.010963     0.0077932     0.0061208      0.
     15          2.3525      0.020913      0.014175     0.0095793     0.0067677     0.0052312      0.
     16          2.8935       0.01753      0.012238      0.008369     0.0058896     0.0044819      0
     17          3.5117      0.014702      0.010552     0.0073101     0.0051351     0.0038505      0.
     18          4.2121      0.012342     0.0090878      0.006382     0.0044849     0.0033193      0.
     19               5      0.010373     0.0078203     0.0055676     0.0039231     0.0028711      0.
     20          5.8805     0.0087301     0.0067252     0.0048575     0.0034361      0.002492      0.
```

21	6.8587	0.0073585	0.0057817	0.004238	0.0030132	0.0021701	0.
22	7.9398	0.0062131	0.0049708	0.0036966	0.0026454	0.001896	0.
23	9.1289	0.0052558	0.0042752	0.0032257	0.0023256	0.0016619	0.
24	10.431	0.0044552	0.0036794	0.0028167	0.002047	0.0014609	0.
25	11.852	0.0037847	0.0031694	0.0024609	0.0018039	0.0012875	0.0
26	13.396	0.0032224	0.0027332	0.0021515	0.0015915	0.0011375	0.0
27	15.069	0.0027501	0.00236	0.0018824	0.0014053	0.0010072	0.0
28	16.875	0.0023527	0.0020407	0.0016483	0.001242	0.00089377	0.0
29	18.82	0.0020176	0.0017673	0.0014445	0.0010993	0.0007946	0.0
30	20.91	0.0017343	0.0015329	0.0012672	0.00097404	0.00070765	0.
31	23.148	0.0014944	0.0013318	0.0011128	0.00086375	0.00063126	0.0
32	25.541	0.0012908	0.0011591	0.00097818	0.00076679	0.00056405	0.0
33	28.093	0.0011176	0.0010105	0.0008609	0.00068159	0.00050477	0.
34	30.81	0.00096988	0.00088251	0.00075856	0.00060649	0.00045237	0.0
35	33.697	0.0008436	0.00077212	0.00066925	0.00054025	0.00040595	0.0
36	36.758	0.00073542	0.00067675	0.0005912	0.00048173	0.00036464	0.0
37	40	0.00064253	0.00059424	0.00052296	0.00043001	0.00032792	0.0
38	43.427	0.00056261	0.00052272	0.00046322	0.00038423	0.00029534	0.0
39	47.044	0.00049368	0.00046065	0.00041086	0.00034368	0.00026633	0.0
40	50.856	0.00043412	0.00040667	0.00036494	0.00030774	0.0002404	0.0
41	54.87	0.00038251	0.00035966	0.00032459	0.00027584	0.00021718	0.0
42	59.089	0.00033773	0.00031865	0.00028912	0.00024751	0.00019642	0.0
43	63.519	0.00029877	0.00028279	0.00025789	0.00022233	0.00017785	0.0
44	68.164	0.00026481	0.0002514	0.00023035	0.00019993	0.00016119	0.0
45	73.032	0.00023517	0.00022388	0.00020604	0.00017998	0.00014624	0.0
46	78.125	0.00020923	0.00019969	0.00018455	0.0001622	0.00013279	0.0
47	83.45	0.00018648	0.00017841	0.00016553	0.00014633	0.00012068	9.1
48	89.011	0.00016649	0.00015965	0.00014868	0.00013216	0.00010978	8.4
49	94.815	0.00014891	0.0001431	0.00013373	0.00011949	9.995e-05	7.7
50	100.87	0.00013342	0.00012846	0.00012044	0.00010816	9.1084e-05	7.0
51	107.17	0.00011973	0.0001155	0.00010862	9.8007e-05	8.308e-05	6.5
52	113.73	0.00010762	0.000104	9.8089e-05	8.8914e-05	7.5856e-05	5.9
53	120.55	9.6907e-05	9.3809e-05	8.873e-05	8.0788e-05	6.9365e-05	5.5
54	127.64	8.7567e-05	8.4918e-05	8.0546e-05	7.3655e-05	6.364e-05	5.1
55	135	8.7567e-05	8.4918e-05	8.0546e-05	7.3656e-05	6.3641e-05	5.1

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

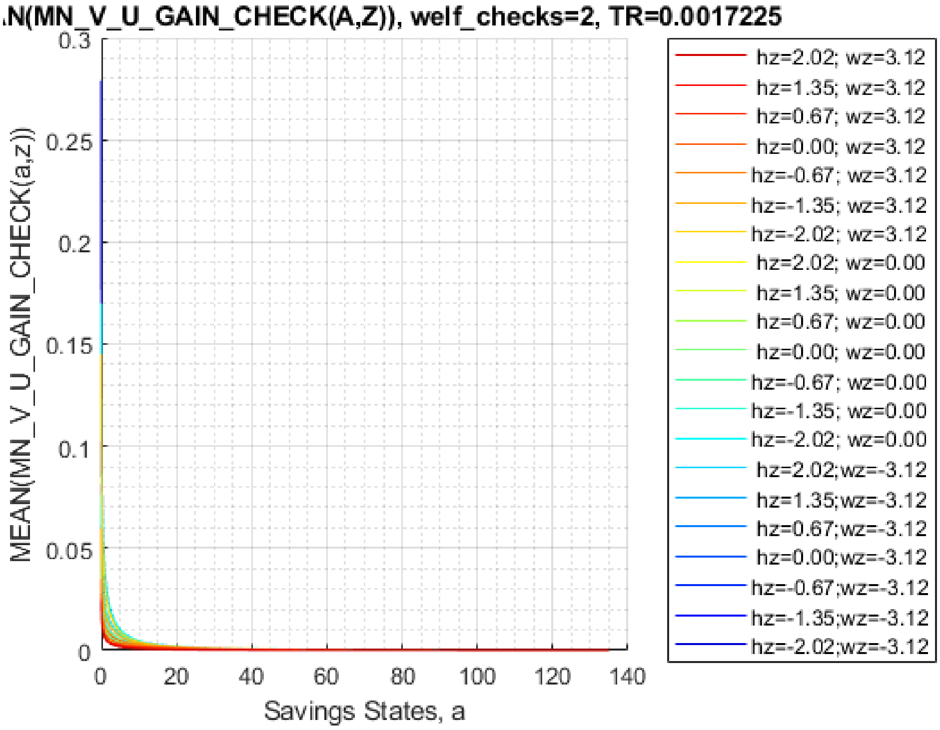
```
xxx  MEAN(MN_MPC_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
```

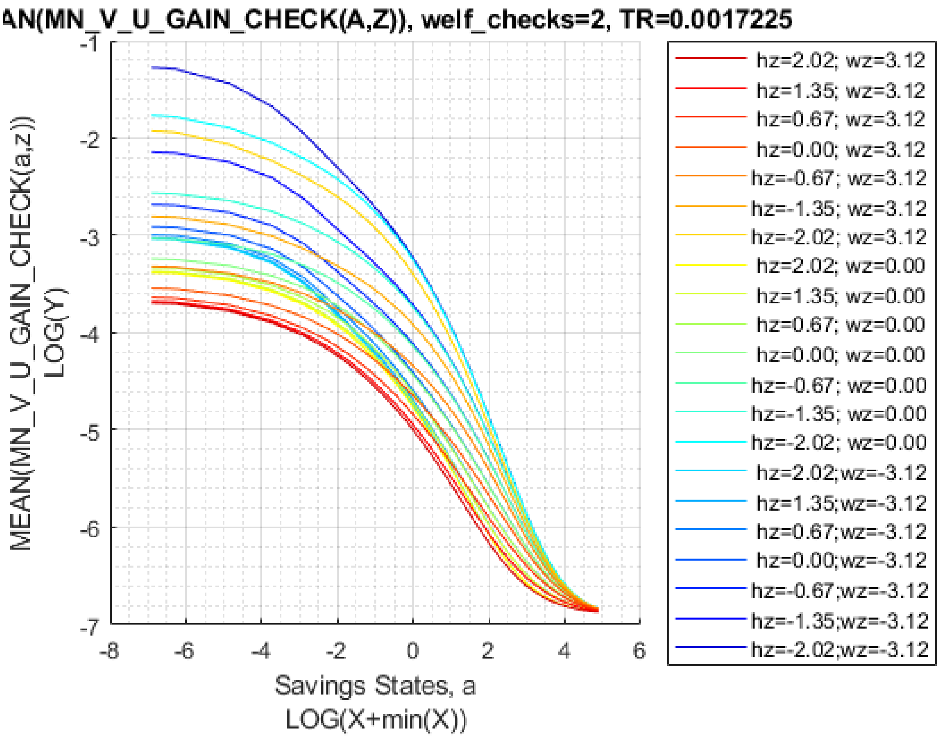
group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5	mea
1	0	0.9469	0.94276	0.8717	0.79605	0.7222	0
2	0.00085734	0.93411	0.9349	0.86947	0.79402	0.72155	0.
3	0.0068587	0.81862	0.82623	0.78945	0.74561	0.6858	0
4	0.023148	0.71821	0.72464	0.70291	0.65987	0.61284	0.
5	0.05487	0.65702	0.66511	0.64788	0.60732	0.55769	0.
6	0.10717	0.54356	0.56685	0.56196	0.52654	0.48344	0.
7	0.18519	0.34762	0.37134	0.3935	0.38764	0.35935	0
8	0.29407	0.27014	0.27012	0.26968	0.2781	0.27577	0.
9	0.43896	0.22635	0.22379	0.21808	0.21552	0.21153	0.
10	0.625	0.18643	0.18323	0.18257	0.18047	0.17918	0.
11	0.85734	0.16247	0.15989	0.15903	0.15917	0.1587	0.
12	1.1411	0.14602	0.14447	0.14433	0.1439	0.14242	0.
13	1.4815	0.13936	0.13684	0.13479	0.13517	0.13698	0.
14	1.8836	0.13686	0.13554	0.13207	0.13035	0.13296	0.

15	2.3525	0.13507	0.13247	0.12893	0.12723	0.12724	0
16	2.8935	0.12673	0.12656	0.12382	0.12237	0.12168	0.
17	3.5117	0.12256	0.12043	0.11953	0.11889	0.11806	0.
18	4.2121	0.11793	0.11791	0.11793	0.11601	0.11689	0.
19	5	0.11739	0.11678	0.11684	0.11637	0.11577	0.
20	5.8805	0.11528	0.11487	0.11456	0.11528	0.11413	0.
21	6.8587	0.11285	0.11244	0.11208	0.11157	0.11191	0.
22	7.9398	0.11153	0.11118	0.11088	0.1103	0.11091	0.
23	9.1289	0.1122	0.11192	0.11174	0.11112	0.11192	0.
24	10.431	0.11098	0.11077	0.11058	0.11009	0.11068	0.
25	11.852	0.10854	0.10833	0.10817	0.10779	0.10805	0.
26	13.396	0.10801	0.10783	0.10765	0.10745	0.10738	
27	15.069	0.11027	0.11014	0.10997	0.10974	0.1096	0
28	16.875	0.11158	0.11146	0.11134	0.11129	0.11095	0.
29	18.82	0.11001	0.1099	0.1098	0.10996	0.1093	0.
30	20.91	0.1067	0.10661	0.10651	0.10645	0.10603	0.
31	23.148	0.10684	0.10677	0.10669	0.10658	0.10631	0.
32	25.541	0.10847	0.10842	0.10835	0.1084	0.108	0.
33	28.093	0.1086	0.10855	0.10851	0.10857	0.10826	0.
34	30.81	0.10826	0.10822	0.10817	0.10821	0.10802	0.
35	33.697	0.10867	0.10864	0.1086	0.10861	0.10855	0.
36	36.758	0.10953	0.10951	0.10947	0.10947	0.10941	0.
37	40	0.10783	0.1078	0.10777	0.10776	0.10793	0.
38	43.427	0.10628	0.10627	0.10624	0.10623	0.1065	0.
39	47.044	0.10618	0.10618	0.10615	0.10614	0.1063	0.
40	50.856	0.10875	0.10873	0.10871	0.1087	0.10863	0.
41	54.87	0.1092	0.10918	0.10917	0.10916	0.10914	0.
42	59.089	0.10782	0.1078	0.10779	0.10778	0.10797	0.
43	63.519	0.10676	0.10675	0.10673	0.10673	0.10684	0.
44	68.164	0.10765	0.10764	0.10763	0.10762	0.1077	0.
45	73.032	0.10869	0.10868	0.10867	0.10866	0.10871	0.
46	78.125	0.10779	0.10778	0.10777	0.10776	0.1078	0.
47	83.45	0.10629	0.10629	0.10628	0.10627	0.1063	0.
48	89.011	0.10682	0.10682	0.10681	0.1068	0.10681	0.
49	94.815	0.10832	0.10832	0.10831	0.1083	0.1083	0.
50	100.87	0.10817	0.10817	0.10816	0.10815	0.10814	0.
51	107.17	0.10742	0.10742	0.10741	0.10739	0.10736	0.
52	113.73	0.10763	0.10763	0.10761	0.10757	0.1075	0.
53	120.55	0.10824	0.10821	0.10815	0.10805	0.10784	0
54	127.64	0.10659	0.10646	0.10629	0.10605	0.10564	0.
55	135	0.10659	0.10646	0.10629	0.10605	0.10564	0.

Graph Mean Values:

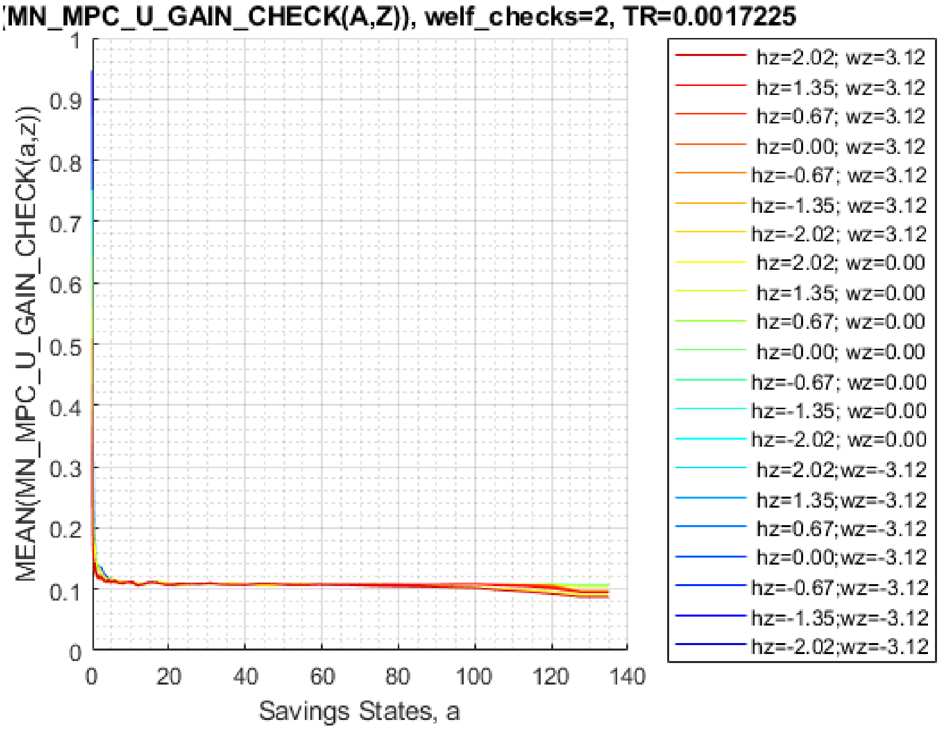
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);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

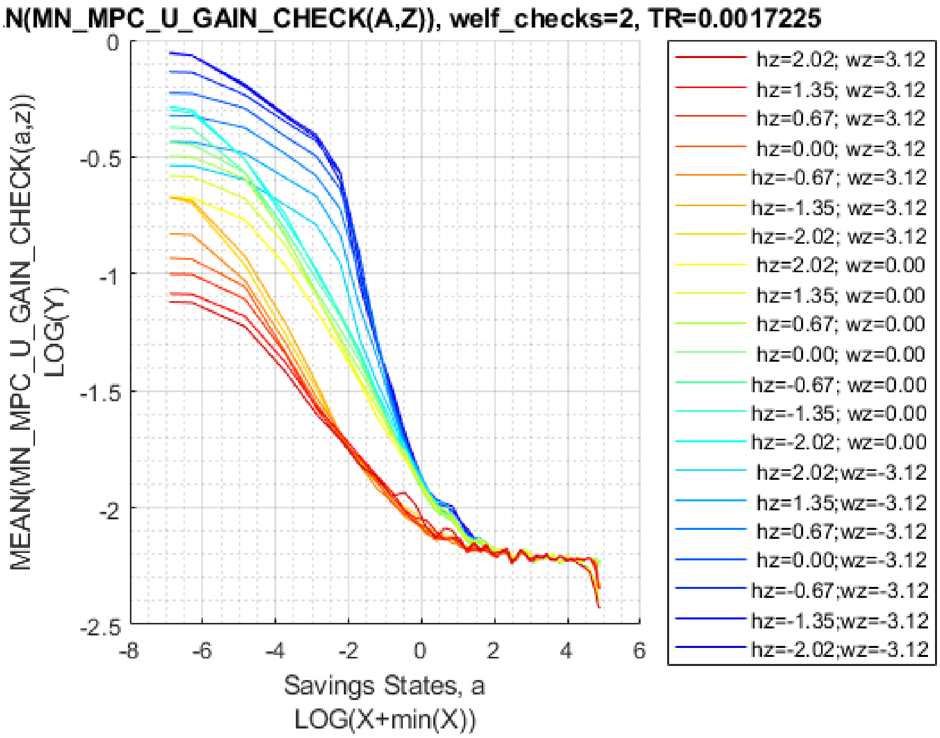




Graph Mean Consumption (MPC: Share of Check Consumed):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str
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);
```





7.1.5 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 38, 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(20,:,1:mp_params('n_eta_H_grid'),1,1,1);
mn_V_W_gain_check_use = ev20_jaeemk_check2 - ev20_jaeemk_check0;
mn_C_W_gain_check_use = ec20_jaeemk_check2 - ec20_jaeemk_check0;
```

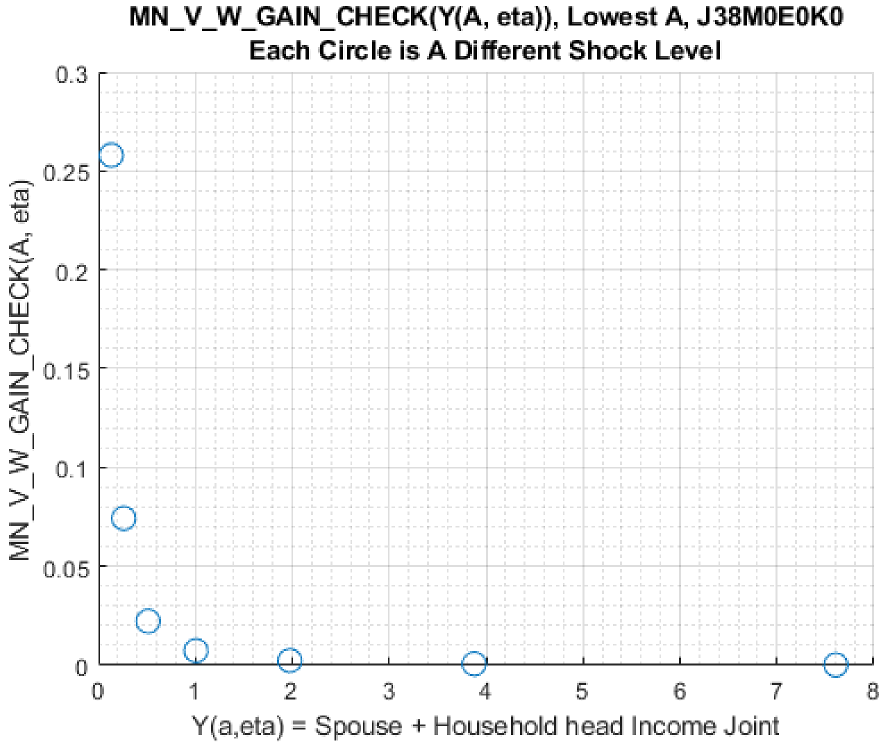
Select Age, Education, Marital, Kids Count:s

```
% 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, it_m
mn_V_W_gain_check_jemk = mn_V_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ, it_m
% 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);
```

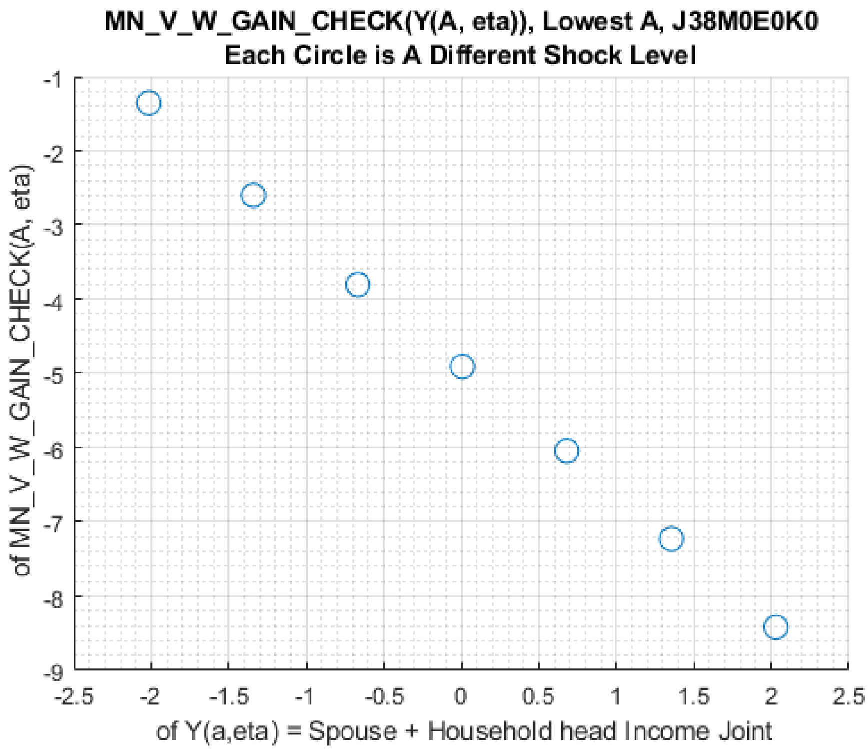
7.1.6 Marginal Value Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

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;
```

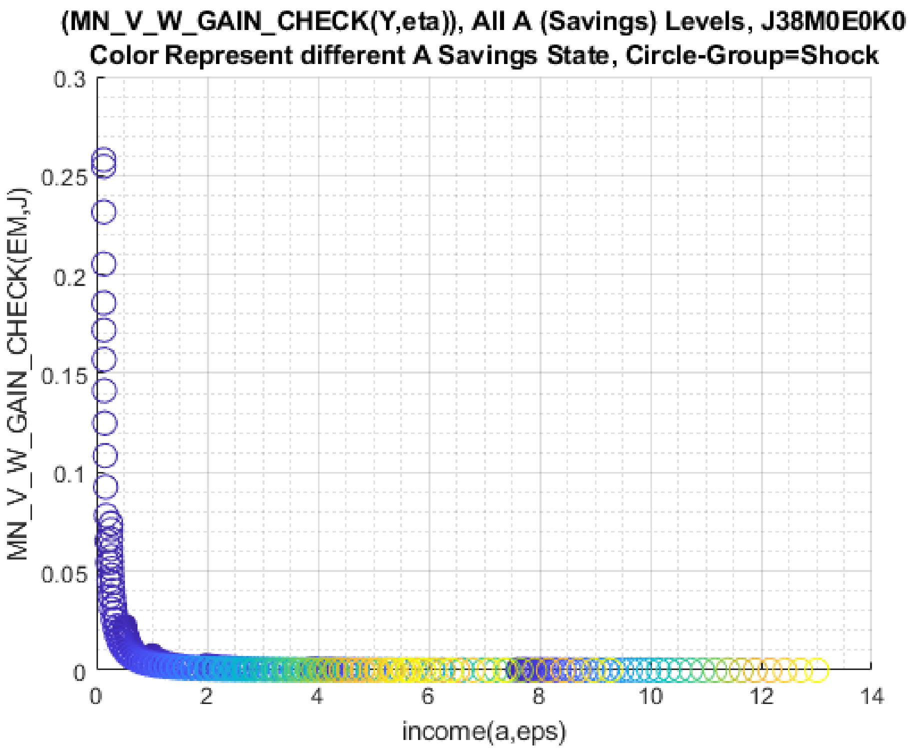



```
figure();
it_shock = 1;
scatter(log(mt_total_inc_jemk(:,it_a)), log(mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel(' of Y(a,eta) = Spouse + Household head Income Joint');
ylabel(' of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

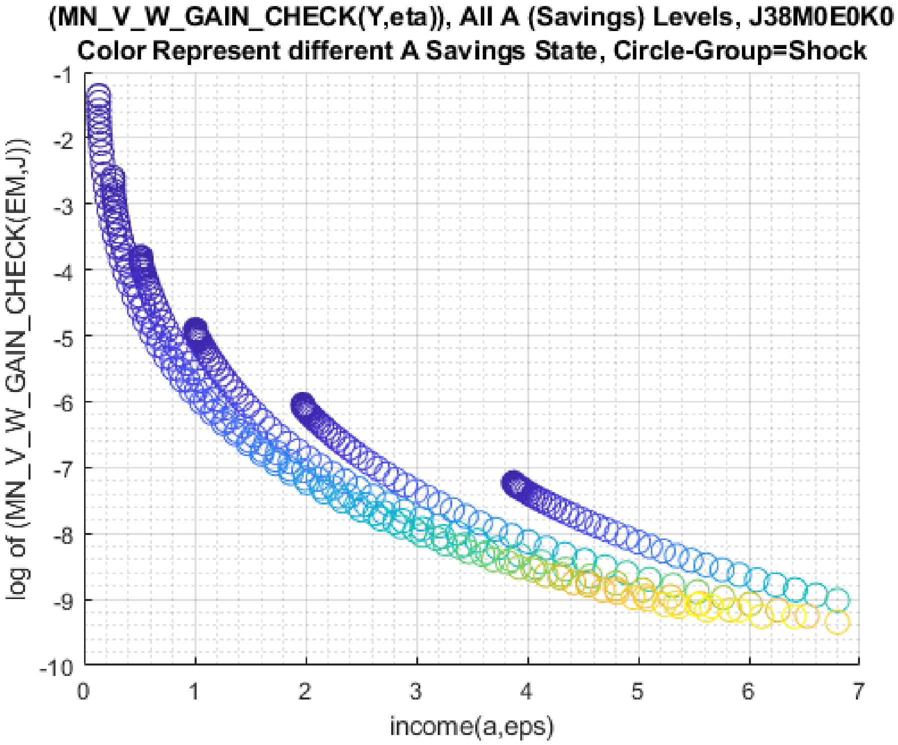


Plot all asset levels:

```
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;
```



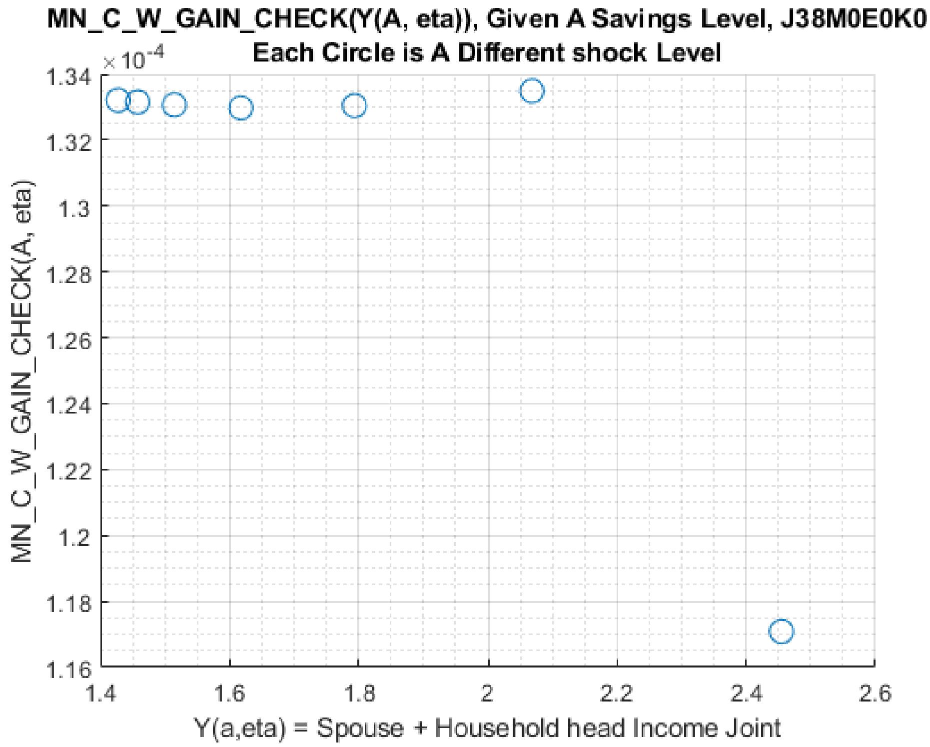

```
figure();
scatter((mt_total_inc_jemk(:)), log(mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');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



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

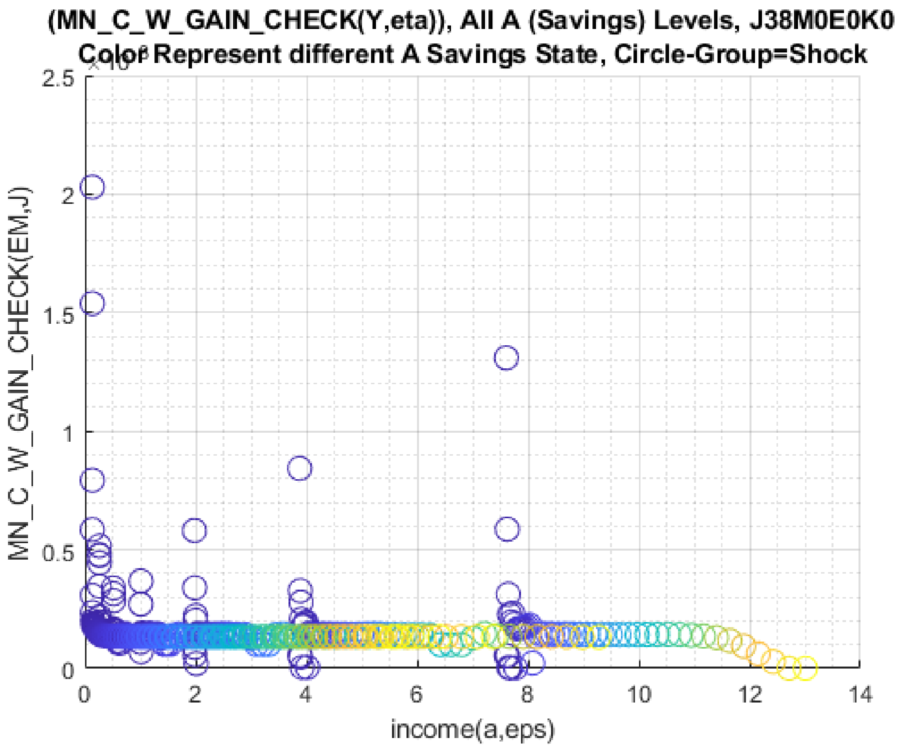
How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
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;
```

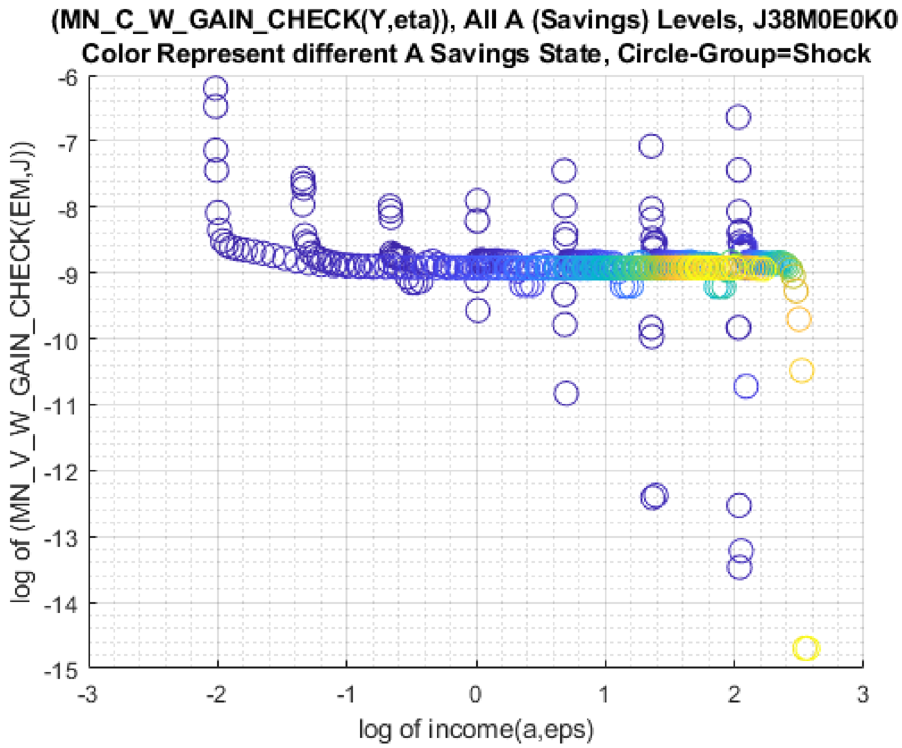


Plot all asset levels:

```
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;
```



7.1.8 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))

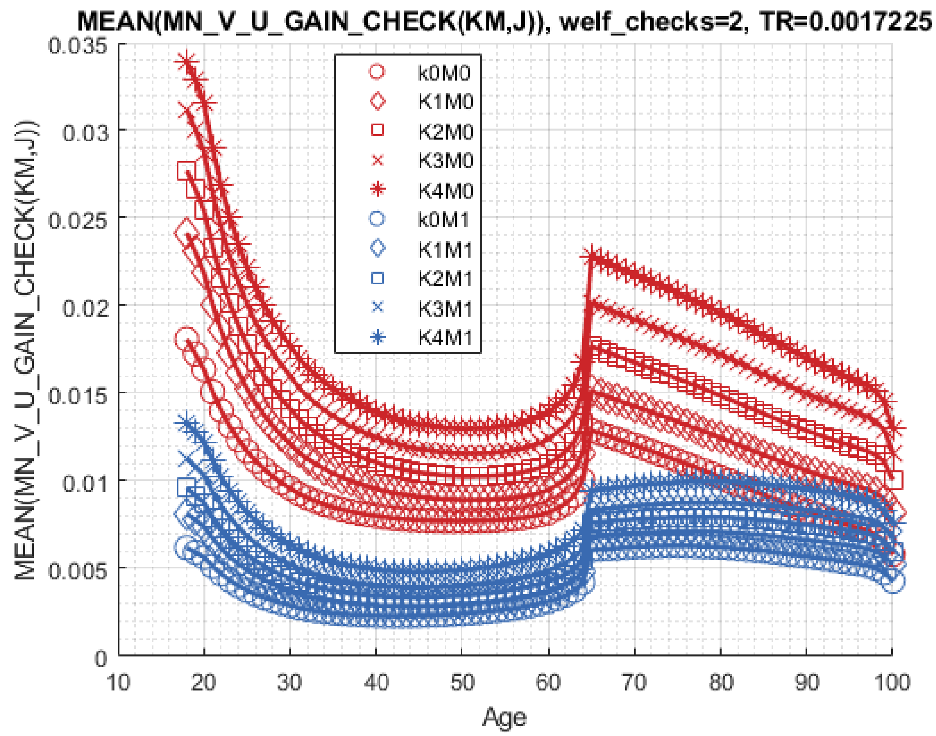
Tabulate value and policies:

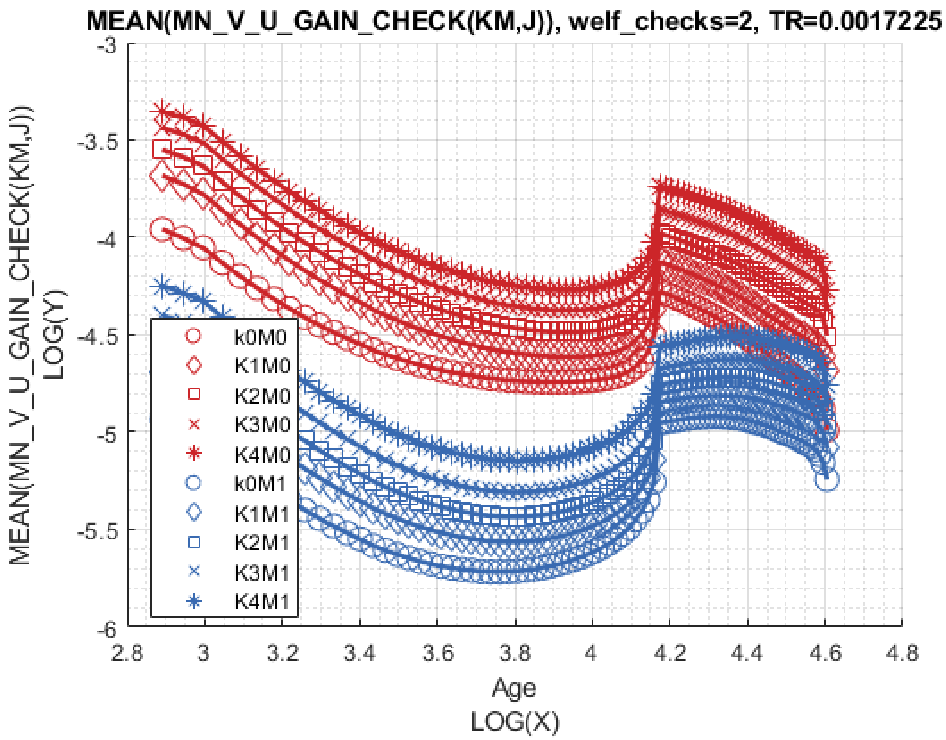
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
```

```
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

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_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0        0.018089       0.017252       0.016367       0.015089       0.014021
      2        2       0        0.024167       0.023077       0.021867       0.020077       0.018573
      3        3       0        0.027732       0.026669       0.025441       0.023362       0.021616
      4        4       0        0.031193        0.03009       0.028767       0.026421       0.024452
      5        5       0        0.033947       0.032875       0.031534        0.02899       0.026856
      6        1       1       0.0061943      0.0059191      0.0056503      0.0051336      0.0047002
      7        2       1       0.0081386      0.0077695       0.007405      0.0067275      0.0061552
      8        3       1       0.0095966      0.0091734      0.0087634      0.0079591      0.0072803
      9        4       1        0.011304       0.010815       0.010337      0.0093981      0.0086038
     10        5       1        0.013294       0.012771       0.012229       0.011142       0.010228

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

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_2
    -----    ----    -----    -----------    -----------    -----------    -----------    ----------

      1        1       0       0.064794       0.073581        0.10118       0.097925       0.093502
      2        2       0       0.072702       0.085068        0.11591        0.11391        0.11124
      3        3       0       0.087366        0.10442        0.13386        0.13266         0.1309
      4        4       0       0.088926        0.10344        0.13895          0.138        0.13683
      5        5       0        0.10276        0.11528        0.14404        0.14305        0.14187
      6        1       1       0.098235        0.10626        0.11555        0.11505        0.11337
      7        2       1        0.10346        0.10657        0.11743        0.11639        0.11571
      8        3       1        0.10975        0.11642        0.12686        0.12437        0.12447
      9        4       1        0.11043        0.11635        0.12843        0.12825        0.13154
     10        5       1        0.12035        0.12608        0.13427          0.136        0.13418
```

Graph Mean Values:

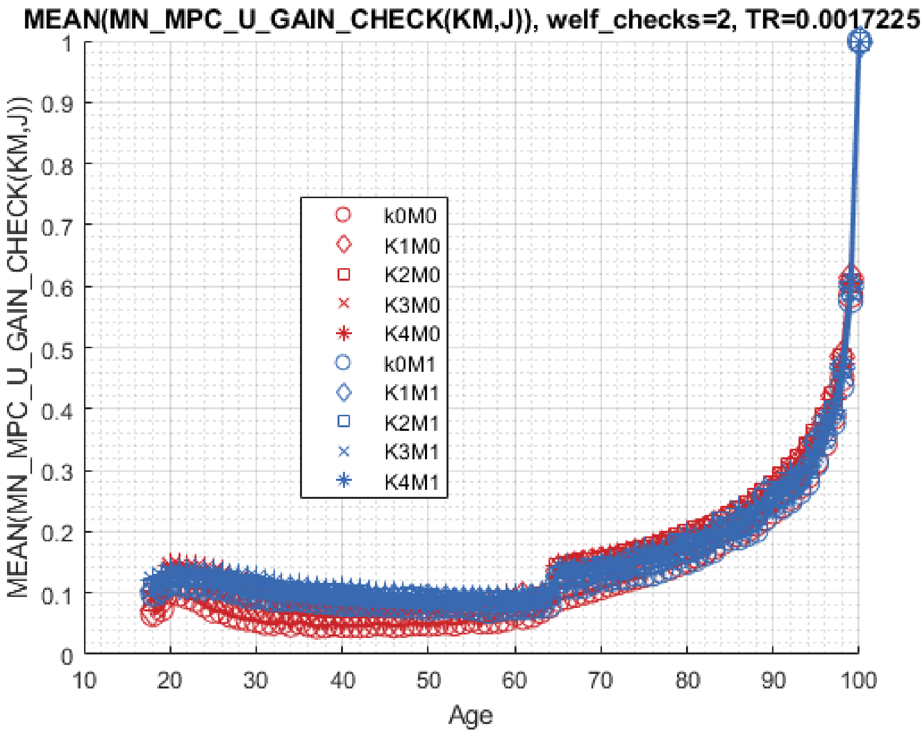
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

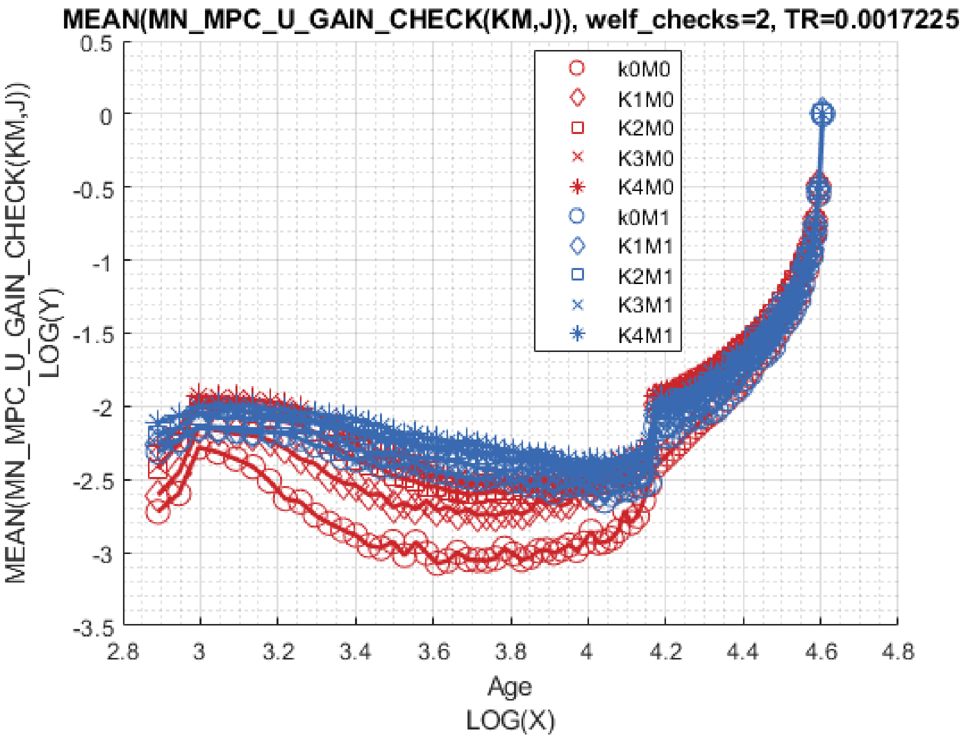




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
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);
```





7.1.9 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'};
```

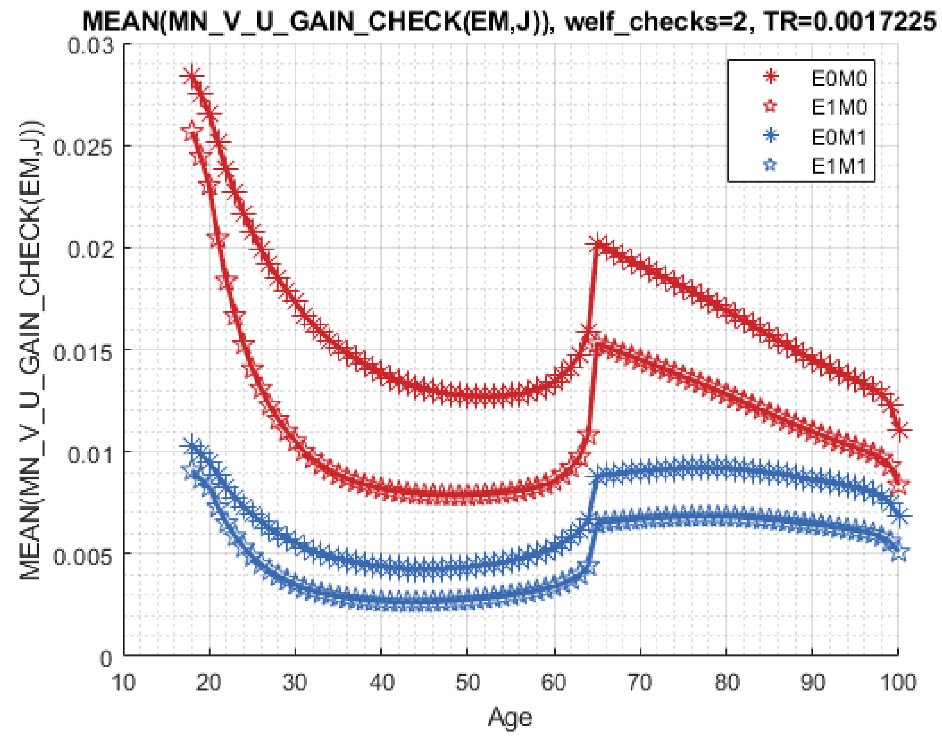
MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

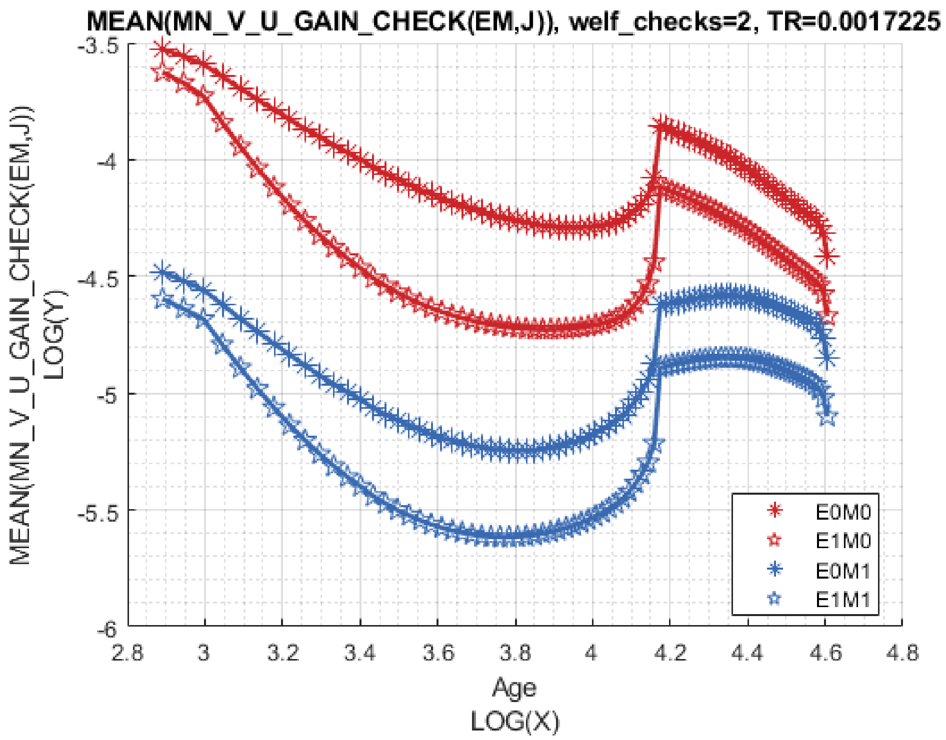
Tabulate value and policies:

```
% 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(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

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
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0          0.0284       0.027529       0.026552       0.025115       0.023836
      2       1       0        0.025651       0.024457       0.023039       0.020461       0.018371
      3       0       1         0.01032      0.0099016      0.0094945      0.0088461      0.0082733
      4       1       1       0.0090905      0.0086775      0.0082596      0.0072981      0.0065136

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

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
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________

      1       0       0        0.073692       0.083211         0.1068        0.10496        0.10269
      2       1       0        0.092928        0.10951        0.14677        0.14526        0.14305
      3       0       1         0.10086         0.1066        0.11448        0.11496        0.11372
      4       1       1         0.11603        0.12207        0.13453        0.13306        0.13399
```

Graph Mean Values:

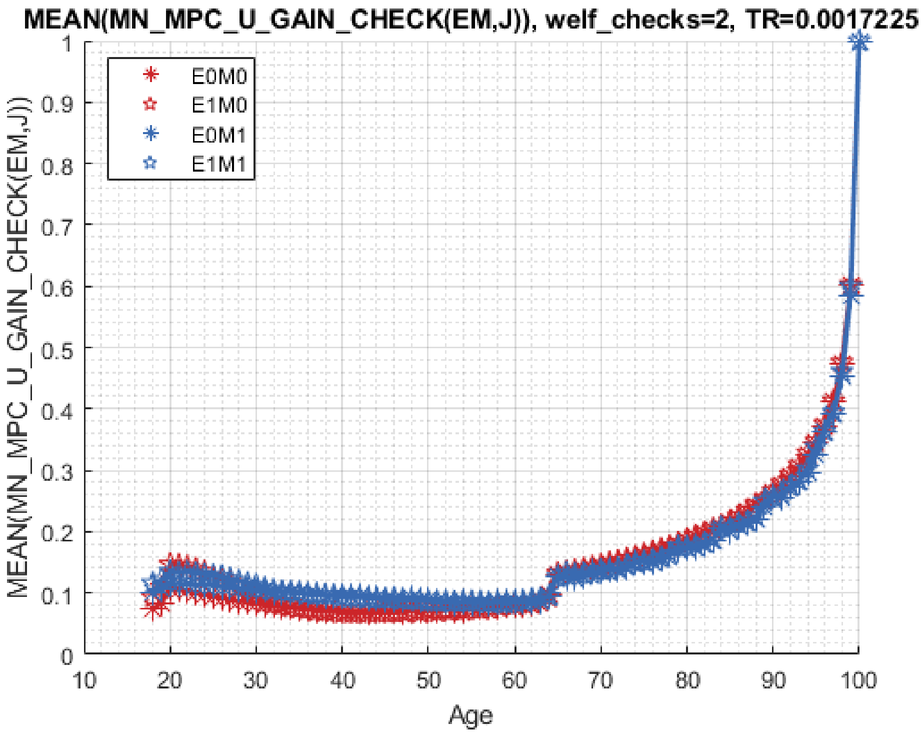
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

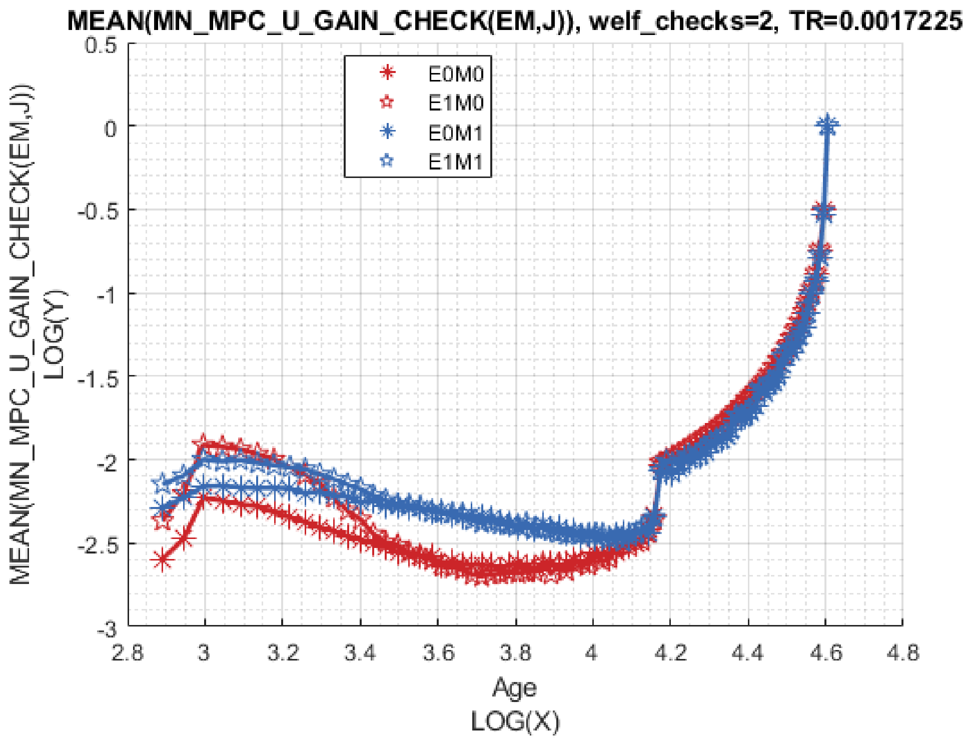




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
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);
```





7.2 2019 Full States EV and EC of One Check

This is the example vignette for function: **snw_evuvw20_jaeemk** from the **PrjOptiSNW Package.** 2019 integrated over VU and VW, given optimal savings choices, unemployment shocks and various expectations.

7.2.1 Test SNW_EVUVW19_JAEEMK Defaults Dense

VFI and Distribution

Call the function with defaults.

```
clear all;
st_solu_type = 'bisec_vec';

% Solve the VFI Problem and get Value Function
mp_params = snw_mp_param('default_docdense');
mp_controls = snw_mp_control('default_test');

% set Unemployment Related Variables
xi=0.5; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=1
b=0; % Unemployment insurance replacement rate (b=0 refers to no UI benefits; b=1 refers to 100 perc
TR=100/58056; % Value of a welfare check (can receive multiple checks). TO DO: Update with alternati

mp_params('xi') = xi;
mp_params('b') = b;
mp_params('TR') = TR;

% Solve for Unemployment Values
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;

% Solve the Model to get V working and unemployed
[V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time=16.9824
```

```
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;
% Solve unemployment
[V_unemp,~,cons_unemp,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=defa
```

```
[Phi_true] = snw_ds_main(mp_params, mp_controls, ap_ss, cons_ss, mp_valpol_more_ss);
```

```
Completed SNW_DS_MAIN;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time=61.7368
```

```
% Get Matrixes
cl_st_precompute_list = {'a', ...
    'inc', 'inc_unemp', 'spouse_inc', 'spouse_inc_unemp', 'ref_earn_wageind_grid',...
    'ap_idx_lower_ss', 'ap_idx_higher_ss', 'ap_idx_lower_weight_ss'};
mp_controls('bl_print_precompute_verbose') = false;
[mp_precompute_res] = snw_hh_precompute(mp_params, mp_controls, cl_st_precompute_list, ap_ss, Phi_tr
```

```
Wage quintile cutoffs=0.49295     0.79302     1.3138     2.1063
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time cost=35.3213
```

7.2.2 Solve for 2019 Evuvw With 0 and 2 Checks

```
% Call Function
```

```
welf_checks = 0;
[ev19_jaeemk_check0, ec19_jaeemk_check0, ev20_jaeemk_check0, ec20_jaeemk_check0] = snw_evuvw19_jaeem
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss, cons_ss, V_unemp, cons_unemp, mp_precompute_res);

Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=0;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_dense
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=0;TR=0.0017225;SNW_MP_PARAM=default_dense;SNW_MP_CONTR
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;timeEUEC=0.67039
Completed SNW_EVUVW19_JAEEMK;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time=8.1727
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                  i    idx    ndim       numel       rowN    colN        sum           mean       std
                  -    ---    ----    ----------     ----    -----    -----------    -------    ----

    ec19_jaeemk   1     1      6      1.8942e+06      82     23100     8.2855e+06     4.3742    5.23
    ec20_jaeemk   2     2      6      1.9173e+06      83     23100     9.7703e+06     5.0959    8.37
    ev19_jaeemk   3     3      6      1.8942e+06      82     23100    -3.7288e+06    -1.9685    22.2
    ev20_jaeemk   4     4      6      1.9173e+06      83     23100    -4.1377e+06    -2.1581    22.8

xxx TABLE:ec19_jaeemk xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5        c23096    c23097    c23098
           --------    --------    --------    --------    --------    ------    ------    ------

    r1      0.08064    0.080744    0.082559    0.086958    0.092876      10.7    10.935    11.171
    r2     0.081432    0.081432    0.084644    0.089174    0.095493     10.86    11.091     11.32
    r3     0.083622    0.083622    0.085501    0.091223    0.097903    11.024    11.249    11.471
    r4     0.086619    0.086619    0.088508    0.094197     0.10085    11.168    11.388    11.601
    r5     0.089528    0.089528    0.091431    0.097077      0.1037    11.309    11.521    11.724
    r78     0.21707     0.21707     0.21707     0.21707     0.22416    26.837    28.052    29.222
    r79     0.21707     0.21707     0.21707     0.21707     0.22416    28.992    31.165    32.888
    r80     0.21707     0.21707     0.21707     0.21707     0.22416    32.266    33.961    36.121
    r81     0.21707     0.21707     0.21707     0.21707     0.22416    38.348    39.931     42.54
    r82     0.21707     0.21707     0.21707     0.21707     0.22361    51.027    52.913    57.047

xxx TABLE:ec20_jaeemk xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5        c23096    c23097    c23098
           --------    --------    --------    --------    --------    ------    ------    ------

    r1     0.078786    0.079375    0.080372    0.084318    0.090081    10.601    10.841    11.082
    r2     0.078786    0.079674    0.081595    0.086622     0.09285    10.755    10.992    11.228
    r3     0.079575    0.080463    0.084561    0.089582    0.096059    10.912    11.145    11.375
    r4     0.082616    0.083505    0.087581    0.092591    0.099121    11.281    11.504    11.718
    r5     0.085578    0.086467    0.090513    0.095505     0.10201    11.412    11.627     11.83
    r79     0.21707     0.21796     0.22416       0.241     0.26692    34.627    36.056    37.896
    r80     0.21707     0.21796     0.22416       0.241     0.26692    39.694    41.355    43.554
    r81     0.21707     0.21796     0.22416       0.241     0.26692    47.978    51.293    53.451
    r82     0.21707     0.21796     0.22416       0.241     0.26745    63.659    68.583    71.594
    r83     0.21707     0.21796     0.22416       0.241     0.27378    112.67    119.43    126.45

xxx TABLE:ev19_jaeemk xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5       c23096    c23097     c23098     c2
           -------    -------    -------    -------    -------    ------    -------    -------    --

    r1     -231.16     -231.1    -230.08     -227.9    -224.17    20.989      21.13      21.27    2
    r2     -222.14    -222.14    -221.33    -219.39    -215.84    20.973      21.11     21.246    2
    r3      -213.6     -213.6    -213.21    -211.33    -207.93    20.953     21.086     21.219    2
```

```
    r4     -205.67    -205.67    -205.31    -203.55    -200.36    20.925     21.055     21.185     2
    r5     -198.45    -198.45    -198.11    -196.45    -193.44    20.891     21.018     21.145     2
    r78    -9.9698    -9.9698    -9.9698    -9.9698     -9.824    2.4708     2.4858     2.5001     2
    r79    -8.9313    -8.9313    -8.9313    -8.9313    -8.7855    2.2408     2.2553     2.2677
    r80    -7.6669    -7.6669    -7.6669    -7.6669    -7.5211    1.9523     1.9632     1.9722     1
    r81    -5.9967    -5.9967    -5.9967    -5.9967    -5.8508    1.5534     1.5588     1.5654     1
    r82    -3.6113    -3.6113    -3.6113    -3.6113    -3.4767    0.9548    0.95648    0.95963    0.

xxx TABLE:ev20_jaeemk xxxxxxxxxxxxxxxxxx
             c1         c2         c3          c4         c5       c23096     c23097     c23098     c
           -------    -------    -------    -------    -------    -------    -------    -------    -

    r1     -235.08    -234.92    -233.84    -231.32    -227.14     21.095     21.236     21.376
    r2     -226.34    -226.18    -225.14     -222.7    -218.72     21.082     21.219     21.356
    r3     -217.94    -217.78    -216.77    -214.45    -210.67     21.065     21.198     21.331
    r4     -209.81    -209.67    -208.73    -206.55    -203.01     21.045     21.174     21.304
    r5     -202.41    -202.27     -201.4    -199.35    -196.01     21.014      21.14     21.267
    r79    -9.9634    -9.9447    -9.8178    -9.5061    -8.9993     2.5241     2.5372     2.5494
    r80    -8.9252    -8.9064    -8.7795    -8.4679    -7.9637     2.2926     2.3026     2.3118
    r81     -7.661    -7.6423    -7.5154    -7.2037    -6.7021     1.9992     2.0061     2.0125
    r82    -5.9911    -5.9724    -5.8455    -5.5339    -5.0346     1.5919     1.5958     1.5994
    r83    -3.6067     -3.588    -3.4611    -3.1494    -2.6526    0.97826    0.97949    0.98063    0

% Call Function
welf_checks = 2;
[ev19_jaeemk_check2, ec19_jaeemk_check2, ev20_jaeemk_check2, ec20_jaeemk_check2] = snw_evuvw19_jaeem
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss, cons_ss, V_unemp, cons_unemp, mp_precompute_res);

Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=2;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_dense
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=2;TR=0.0017225;SNW_MP_PARAM=default_dense;SNW_MP_CONTR
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;timeEUEC=0.4338
Completed SNW_EVUVW19_JAEEMK;SNW_MP_PARAM=default_dense;SNW_MP_CONTROL=default_test;time=9.4255
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                   i    idx    ndim      numel       rowN    colN         sum          mean       std
                   -    ---    ----    ----------    ----    -----    -----------    -------    ----

    ec19_jaeemk    1     1      6      1.8942e+06     82     23100     8.2866e+06     4.3747     5.23
    ec20_jaeemk    2     2      6      1.9173e+06     83     23100     9.7714e+06     5.0964     8.37
    ev19_jaeemk    3     3      6      1.8942e+06     82     23100    -3.7109e+06    -1.9591     22.
    ev20_jaeemk    4     4      6      1.9173e+06     83     23100    -4.1188e+06    -2.1482     22.8

xxx TABLE:ec19_jaeemk xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5        c23096    c23097    c23098
           --------    --------    --------    --------    --------    ------    ------    ------

    r1     0.081703    0.081806      0.0836    0.087675     0.09332      10.7    10.935    11.171
    r2     0.083719    0.083719    0.085961    0.089966    0.095955     10.86    11.091     11.32
    r3     0.086201    0.086201    0.087481    0.092085    0.098395    11.024     11.25    11.471
    r4     0.089189    0.089189    0.090472    0.095056     0.10134    11.168    11.388    11.601
    r5     0.092085    0.092085    0.093374    0.097931     0.10419    11.309    11.521    11.724
    r78     0.22052     0.22052     0.22052     0.22052     0.22761    26.838    28.053    29.223
    r79     0.22052     0.22052     0.22052     0.22052     0.22761    28.993    31.166    32.889
    r80     0.22052     0.22052     0.22052     0.22052     0.22761    32.267    33.963    36.122
    r81     0.22052     0.22052     0.22052     0.22052     0.22761    38.349    39.933    42.542
```

```
    r82     0.22052      0.22052      0.22052      0.22052      0.22706      51.03     52.916      57.05

xxx TABLE:ec20_jaeemk xxxxxxxxxxxxxxxxxx
              c1           c2           c3           c4           c5       c23096     c23097     c23098
           --------     --------     --------     --------     --------     ------     ------     ------

    r1     0.079785     0.079927     0.081177     0.084922      0.09046     10.601     10.841     11.082
    r2     0.080464     0.080738     0.082621     0.087274     0.093237     10.755     10.992     11.228
    r3     0.082148     0.082733     0.085586     0.090261     0.096428     10.913     11.145     11.375
    r4     0.085181     0.085763     0.088604     0.093275     0.099485     11.281     11.504     11.718
    r5     0.088131     0.088709     0.091532     0.096187      0.10238     11.412     11.627      11.83
    r79     0.22052       0.2214      0.22761      0.24372      0.26885     34.627     36.057     37.898
    r80     0.22052       0.2214      0.22761      0.24372      0.26914     39.694     41.356     43.555
    r81     0.22052       0.2214      0.22761      0.24372      0.26933      47.98     51.294     53.452
    r82     0.22052       0.2214      0.22761      0.24378      0.26982     63.662     68.584     71.596
    r83     0.22052       0.2214      0.22761      0.24444      0.27722     112.67     119.43     126.45

xxx TABLE:ev19_jaeemk xxxxxxxxxxxxxxxxxx
             c1          c2          c3          c4          c5        c23096     c23097      c23098      c2
           -------     -------     -------     -------     -------     ------     -------     -------     --

    r1     -230.56     -230.51     -229.56     -227.46     -223.78     20.989       21.13       21.27     2
    r2     -221.56     -221.56     -220.82     -218.96     -215.47     20.973       21.11      21.246     2
    r3     -213.05     -213.05      -212.7     -210.91     -207.58     20.953      21.086      21.219     2
    r4     -205.16     -205.16     -204.83     -203.16     -200.03     20.926      21.055      21.185     2
    r5     -197.97     -197.97     -197.66     -196.08     -193.13     20.891      21.018      21.145     2
    r78    -9.8986     -9.8986     -9.8986     -9.8986     -9.7602     2.4708      2.4859      2.5001     2
    r79    -8.8601     -8.8601     -8.8601     -8.8601     -8.7216     2.2408      2.2553      2.2677
    r80    -7.5957     -7.5957     -7.5957     -7.5957     -7.4572     1.9523      1.9632      1.9722     1
    r81    -5.9255     -5.9255     -5.9255     -5.9255      -5.787     1.5534      1.5588      1.5654     1
    r82    -3.5401     -3.5401     -3.5401     -3.5401     -3.4123     0.9548     0.95648     0.95963     0.

xxx TABLE:ev20_jaeemk xxxxxxxxxxxxxxxxxx
             c1          c2          c3          c4          c5        c23096      c23097      c23098      c
           -------     -------     -------     -------     -------     -------     -------     -------     -

    r1     -234.48     -234.32     -233.33     -230.88     -226.76      21.095      21.236      21.377
    r2     -225.75      -225.6     -224.64     -222.29     -218.36      21.082      21.219      21.356
    r3     -217.37     -217.22      -216.3     -214.05     -210.33      21.065      21.198      21.331
    r4     -209.28     -209.15     -208.29     -206.18     -202.69      21.045      21.174      21.304
    r5     -201.91     -201.79     -200.98        -199     -195.71      21.014       21.14      21.267
    r79    -9.8923     -9.8742      -9.754     -9.4529     -8.9555      2.5241      2.5372      2.5494
    r80    -8.8541     -8.8359     -8.7157     -8.4149     -7.9203      2.2926      2.3026      2.3118
    r81    -7.5899     -7.5718     -7.4516      -7.151     -6.6591      1.9992      2.0061      2.0125
    r82    -5.9201     -5.9019     -5.7817     -5.4814     -4.9918      1.5919      1.5958      1.5994
    r83    -3.5356     -3.5175     -3.3973     -3.0972     -2.6142     0.97826     0.97949     0.98063     0
```

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check = ev19_jaeemk_check2 - ev19_jaeemk_check0;
mn_MPC_U_gain_share_check = (ec19_jaeemk_check2 - ec19_jaeemk_check0)./(welf_checks*mp_params('TR'))
```

7.2.3 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 = 18:99;
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=%3.2
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});
```

7.2.4 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(mp_par
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar_
```

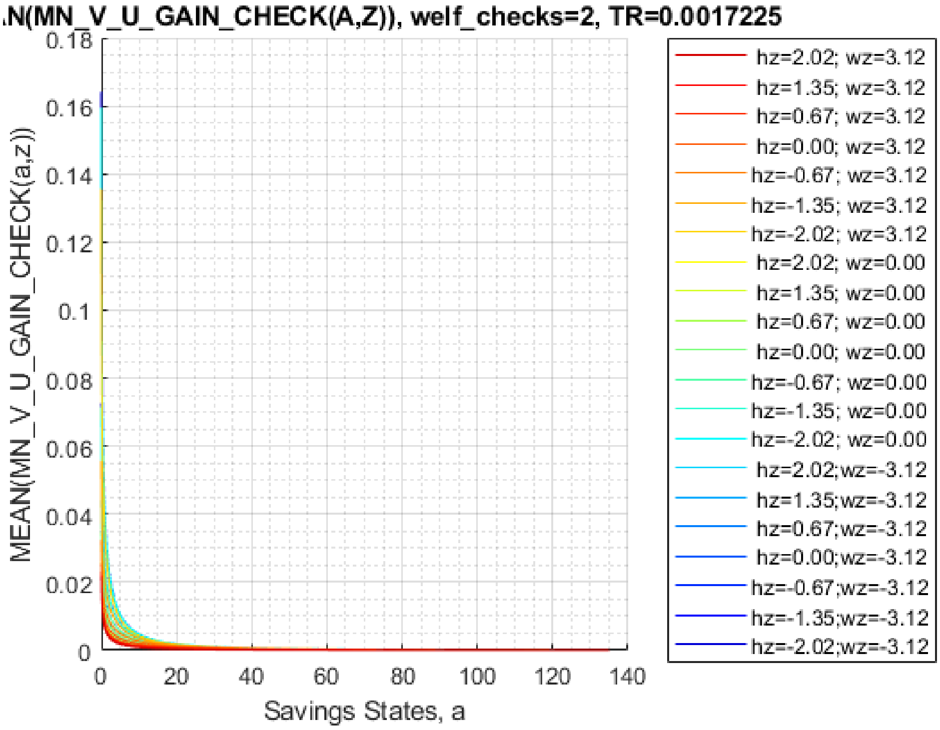
```
xxx  MEAN(MN_V_U_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1               0       0.16422      0.072711      0.044261      0.035185      0.031921      0
      2      0.00085734       0.16411      0.072685      0.044254      0.035183      0.031921      0
      3       0.0068587       0.16089      0.072203      0.044174       0.03516      0.031906      0
      4        0.023148       0.14347      0.068722      0.043233      0.034539      0.031319      0
      5         0.05487       0.12489      0.062983      0.040587      0.032437      0.029301      0
      6         0.10717       0.10956      0.056872      0.037152      0.029572      0.026538      0
      7         0.18519      0.093145      0.050122      0.032855      0.025873      0.022971
      8         0.29407      0.076208      0.042336      0.027832      0.021566      0.018844      0
      9         0.43896      0.063079      0.035973      0.023726      0.018152      0.015638      0
     10           0.625      0.052632      0.030859      0.020414       0.01541      0.013081      0
     11         0.85734      0.044007      0.026566      0.017638      0.013131      0.010971      0.
     12          1.1411      0.036806      0.022912      0.015283      0.011232     0.0092306      0.
     13          1.4815      0.030786      0.019772      0.013268     0.0096429     0.0077925      0.
     14          1.8836      0.025765      0.017067      0.011541     0.0083128      0.006606      0
     15          2.3525      0.021575      0.014731      0.010054     0.0071926     0.0056241      0.
     16          2.8935      0.018077      0.012701     0.0087613      0.006238     0.0048017      0.
     17          3.5117      0.015151      0.010936     0.0076339     0.0054202       0.00411      0.
     18          4.2121      0.012711     0.0094052     0.0066501     0.0047191     0.0035306      0.
     19               5      0.010676      0.008083     0.0057903     0.0041163     0.0030444      0.
```

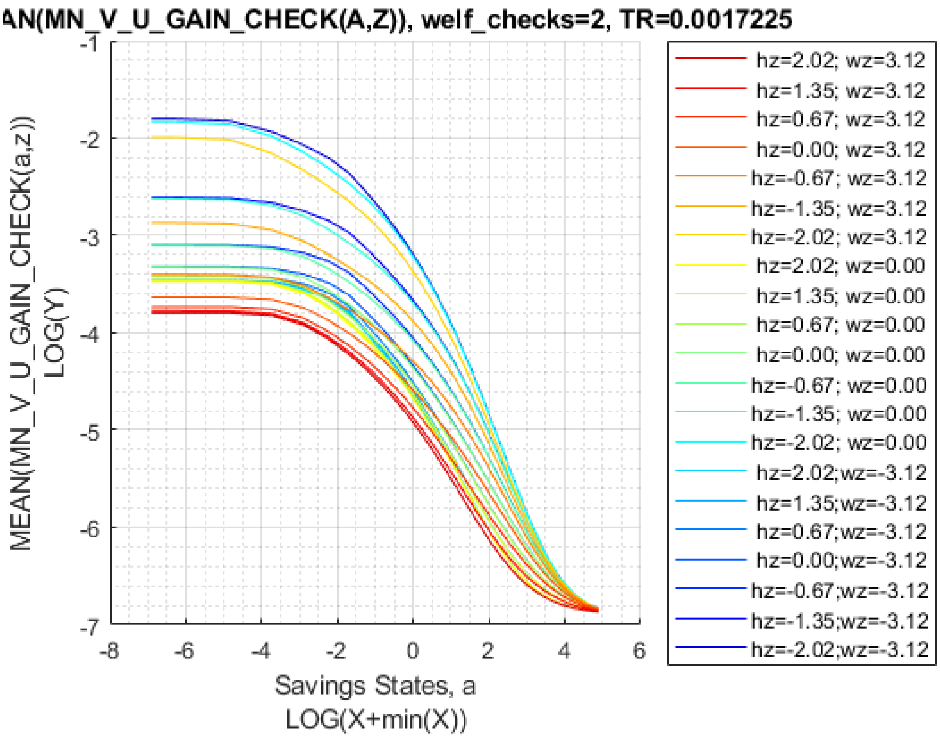
```
20        5.8805      0.008979     0.0069435     0.0050432      0.003597     0.0026352     0.
21        6.8587     0.0075642     0.0059636     0.0043935     0.0031477      0.002289     0.
22        7.9398     0.0063835     0.0051226      0.003827     0.0027577      0.001995     0.
23        9.1289     0.0053974     0.0044023     0.0033352     0.0024197     0.0017445     0.
24        10.431     0.0045736     0.0037863     0.0029093     0.0021264     0.0015304     0.
25        11.852     0.0038839     0.0032595     0.0025394     0.0018711     0.0013461    0.0
26        13.396     0.0033057     0.0028093     0.0022181     0.0016485      0.001187    0.0
27        15.069     0.0028204     0.0024245     0.0019392     0.0014539     0.0010493    0.0
28        16.875     0.0024123     0.0020957     0.0016969     0.0012837    0.00092975    0.0
29         18.82     0.0020683     0.0018143     0.0014864     0.0011352    0.00082552    0.0
30         20.91     0.0017776     0.0015732     0.0013032     0.0010051    0.00073426    0.0
31        23.148     0.0015314     0.0013663     0.0011438    0.00089056    0.00065417    0.0
32        25.541     0.0013225     0.0011887      0.001005    0.00079002    0.00058386    0.0
33        28.093     0.0011448     0.0010361    0.00088415    0.00070181    0.00052199    0.0
34         30.81    0.00099337    0.00090469    0.00077879    0.00062417     0.0004674    0.0
35        33.697    0.00086393    0.00079137    0.00068687    0.00055572    0.00041911    0.0
36        36.758    0.00075305     0.0006935     0.0006066    0.00049531     0.0003762    0.0
37            40    0.00065787    0.00060885    0.00053645    0.00044195    0.00033809    0.0
38        43.427    0.00057598     0.0005355    0.00047505    0.00039475    0.00030432    0.0
39        47.044    0.00050536    0.00047182    0.00042126    0.00035297    0.00027428    0.0
40        50.856    0.00044434    0.00041648    0.00037408    0.00031594    0.00024745    0.0
41         54.87    0.00039149    0.00036829    0.00033267    0.00028312    0.00022344    0.0
42        59.089    0.00034563    0.00032626    0.00029627    0.00025398    0.00020201     0.
43        63.519    0.00030574    0.00028952    0.00026422    0.00022808    0.00018284    0.0
44        68.164    0.00027097    0.00025735    0.00023597    0.00020506    0.00016566    0.0
45        73.032    0.00024062    0.00022915    0.00021104    0.00018456    0.00015024    0.0
46        78.125    0.00021406    0.00020439    0.00018901    0.00016629    0.00013638    0.0
47         83.45    0.00019078    0.00018259    0.00016951       0.00015    0.00012392    9.4
48        89.011    0.00017032    0.00016338    0.00015223    0.00013545    0.00011269    8.6
49        94.815    0.00015233    0.00014643    0.00013691    0.00012244    0.00010258    7.9
50        100.87    0.00013647    0.00013144    0.00012329    0.00011081    9.3454e-05    7.2
51        107.17    0.00012246    0.00011817    0.00011118     0.0001004    8.5225e-05    6.6
52        113.73    0.00011007     0.0001064     0.0001004    9.1075e-05      7.78e-05    6.1
53        120.55    9.9117e-05    9.5971e-05    9.0813e-05    8.2743e-05     7.113e-05    5.6
54        127.64    8.9565e-05    8.6875e-05    8.2434e-05    7.5433e-05    6.5251e-05    5.2
55           135    8.5718e-05    8.3153e-05    7.8927e-05    7.2281e-05    6.2631e-05    5.0

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

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    mea
    -----    ----------    ----------    ----------    ----------    ----------    ----------    ---

      1              0       0.74098       0.72618       0.66567       0.62344       0.58432     0.
      2     0.00085734        0.7405       0.72615       0.66568       0.62348       0.58437     0.
      3      0.0068587       0.72669       0.71733       0.66211       0.62125       0.58261     0.
      4       0.023148       0.61478       0.61585       0.59388       0.58469       0.55844     0.
      5        0.05487       0.52118       0.52002       0.51447       0.51652       0.50302     0.
      6        0.10717        0.4677       0.46139       0.46307       0.46985       0.45963     0.
      7        0.18519        0.3973       0.37778       0.38896       0.39697       0.38775     0.
      8        0.29407       0.32167       0.31637       0.29863       0.31313       0.31844     0.
      9        0.43896       0.25268       0.24569        0.2474       0.24208       0.24978     0.
     10          0.625       0.20528       0.20521       0.20529       0.20426       0.20225     0.
     11        0.85734        0.1787       0.17548         0.175       0.17606       0.17597     0.
     12         1.1411       0.16031        0.1574       0.15581       0.15574       0.15674     0.
     13         1.4815       0.14645       0.14407       0.14238       0.14223       0.14334
```

14	1.8836	0.13586	0.1339	0.13226	0.13233	0.13318	0.
15	2.3525	0.12905	0.12748	0.12583	0.12631	0.12672	0.
16	2.8935	0.1248	0.12332	0.12186	0.12224	0.12237	0.
17	3.5117	0.12316	0.12217	0.12089	0.121	0.12082	0.
18	4.2121	0.1215	0.12046	0.11937	0.11949	0.11943	0.
19	5	0.11643	0.11587	0.1151	0.1147	0.11496	0.
20	5.8805	0.11287	0.11225	0.11191	0.11124	0.11141	0.
21	6.8587	0.112	0.11157	0.11084	0.11056	0.11096	0.
22	7.9398	0.11185	0.11153	0.11102	0.11039	0.11115	0.
23	9.1289	0.11312	0.1128	0.11262	0.11186	0.11264	0.
24	10.431	0.1119	0.11162	0.11143	0.11076	0.11149	0.
25	11.852	0.10944	0.10926	0.10903	0.10841	0.10905	0.
26	13.396	0.10862	0.10843	0.10826	0.10771	0.10818	0.
27	15.069	0.10932	0.10916	0.109	0.1085	0.10883	0.
28	16.875	0.1099	0.10978	0.10963	0.10944	0.10935	0.
29	18.82	0.10798	0.10788	0.10777	0.10777	0.10737	0.
30	20.91	0.10705	0.10696	0.10687	0.10661	0.10637	0.
31	23.148	0.10761	0.10755	0.10746	0.10724	0.10705	0.
32	25.541	0.1096	0.10954	0.10947	0.1095	0.10909	0.
33	28.093	0.11004	0.10999	0.10994	0.11	0.10956	0
34	30.81	0.10764	0.1076	0.10756	0.10759	0.10721	0.
35	33.697	0.10612	0.10609	0.10605	0.10605	0.10566	0.
36	36.758	0.10599	0.10597	0.10593	0.10592	0.10557	0.
37	40	0.10721	0.1072	0.10716	0.10717	0.10715	0.
38	43.427	0.10865	0.10862	0.10859	0.10859	0.10875	0.
39	47.044	0.10816	0.10813	0.1081	0.10808	0.1081	0.
40	50.856	0.10782	0.10782	0.10778	0.10777	0.10749	0.
41	54.87	0.10726	0.10727	0.10724	0.10722	0.10708	0.
42	59.089	0.10707	0.10706	0.10706	0.10703	0.10722	0.
43	63.519	0.10759	0.10756	0.10757	0.10755	0.10766	0.
44	68.164	0.1073	0.10729	0.10727	0.10728	0.10734	0.
45	73.032	0.10665	0.10665	0.10662	0.10663	0.10666	0.
46	78.125	0.10663	0.10662	0.10661	0.1066	0.10665	0.
47	83.45	0.10789	0.10788	0.10787	0.10785	0.10789	0.
48	89.011	0.10873	0.10873	0.10872	0.10871	0.10873	0.
49	94.815	0.10779	0.10779	0.10779	0.10778	0.10777	0.
50	100.87	0.10679	0.10679	0.10678	0.10677	0.10675	0.
51	107.17	0.10631	0.1063	0.10629	0.10627	0.10623	0.
52	113.73	0.10697	0.10696	0.10694	0.10689	0.10681	0.
53	120.55	0.1072	0.10716	0.10708	0.10697	0.10673	0.
54	127.64	0.10698	0.10688	0.10673	0.10652	0.10612	0.
55	135	0.10607	0.10591	0.10572	0.10546	0.10504	0.

Graph Mean Values:

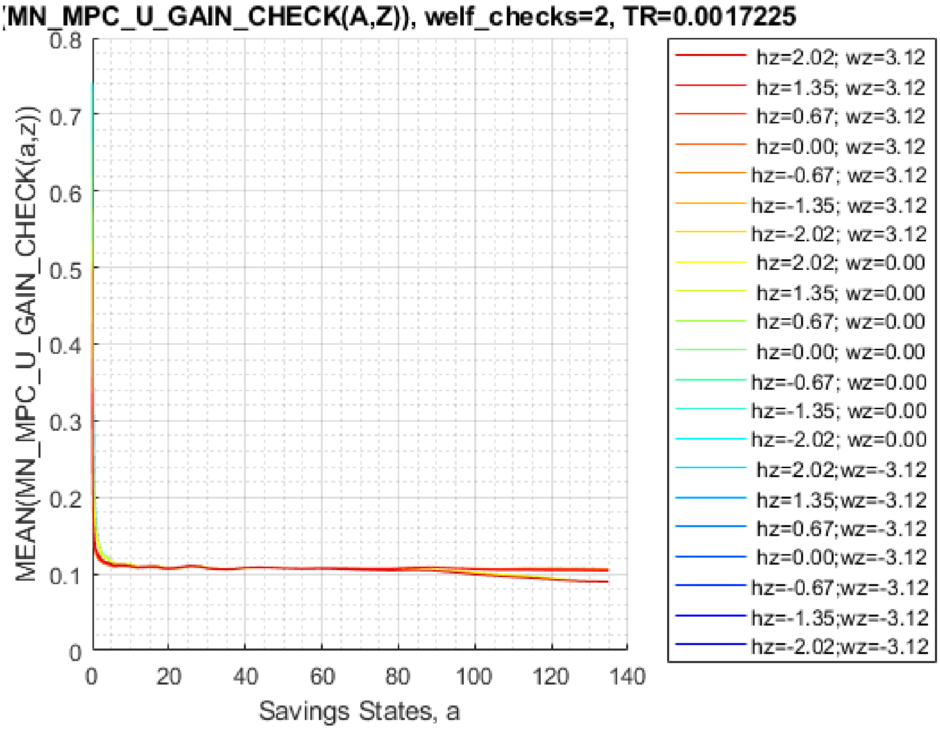
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(m
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);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

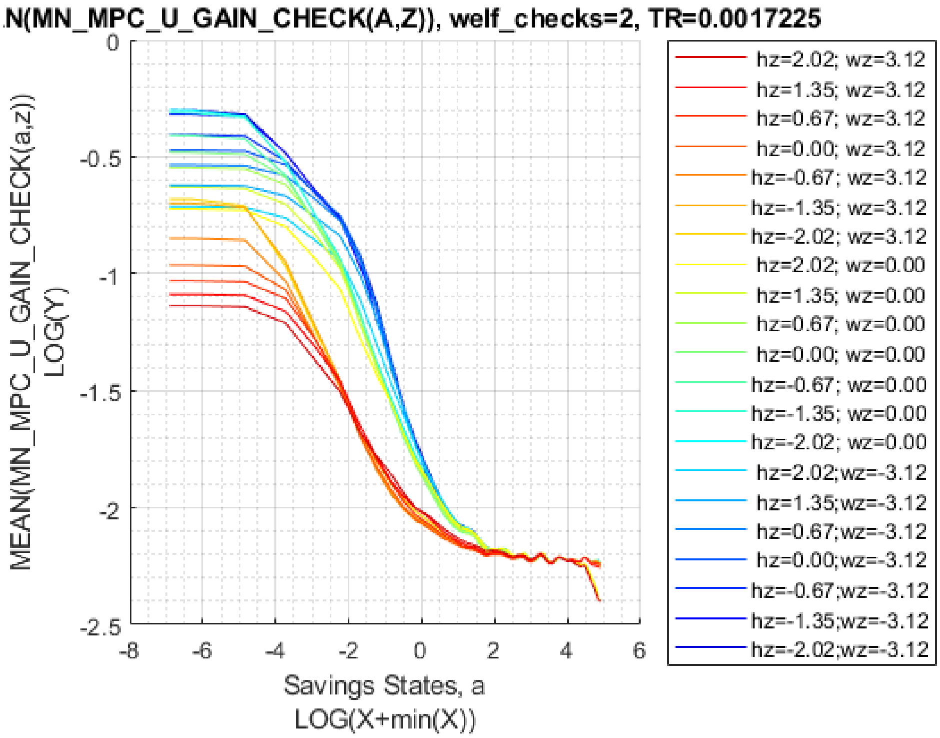




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2str
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);
```





7.2.5 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 37vn

, 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 = ev19_jaeemk_check2 - ev19_jaeemk_check0;
mn_C_W_gain_check_use = ec19_jaeemk_check2 - ec19_jaeemk_check0;
```

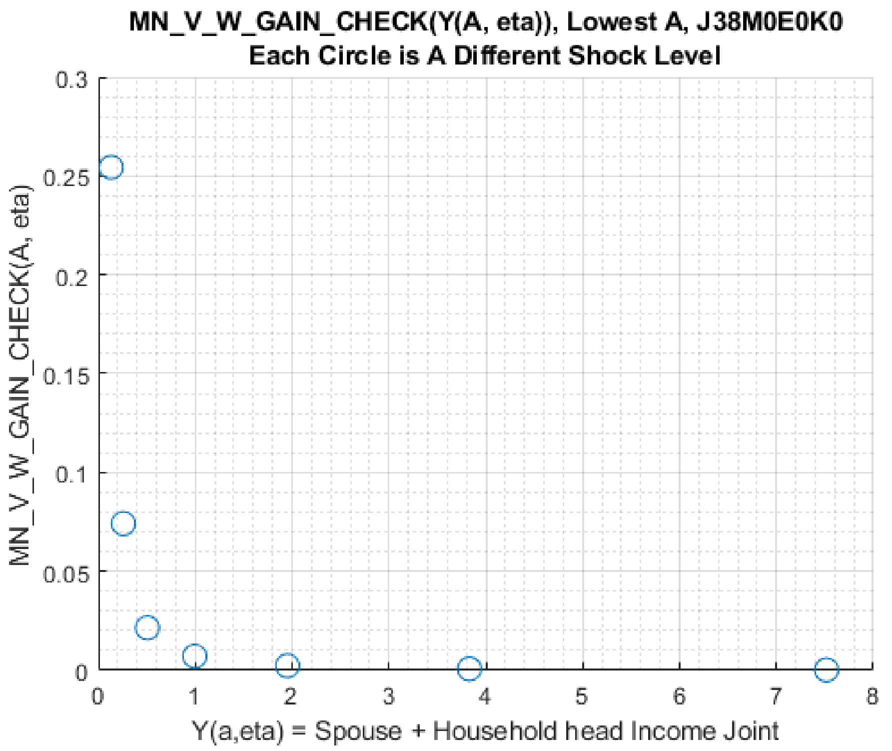
Select Age, Education, Marital, Kids Count:s

```
% 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, it_m
mn_V_W_gain_check_jemk = mn_V_W_gain_check_use(it_age, :, 1:mp_params('n_eta_H_grid'), it_educ, it_m
% 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);
```

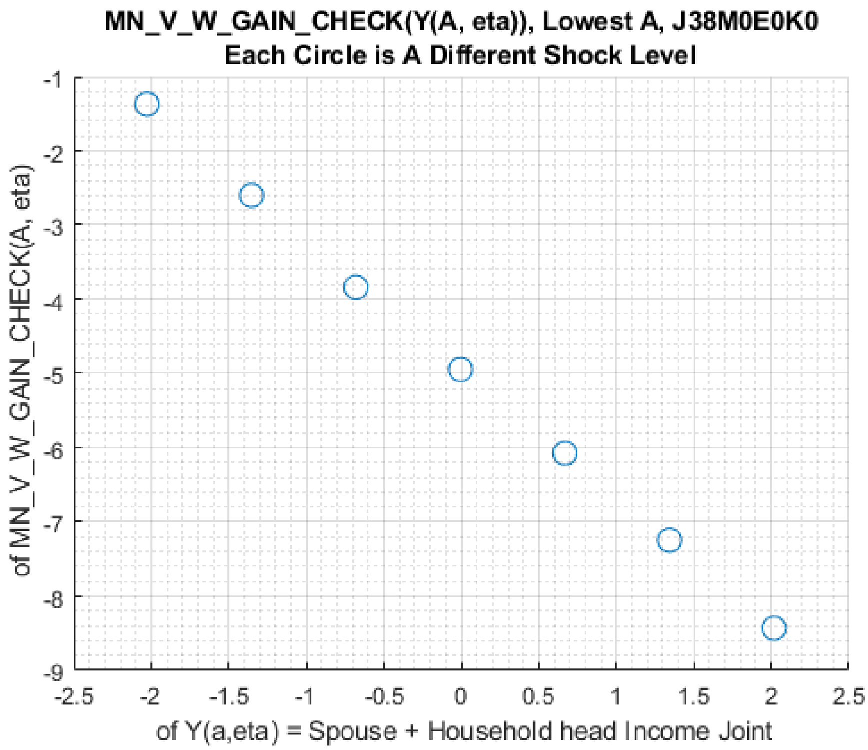
7.2.6 Marginal Value Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

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;
```

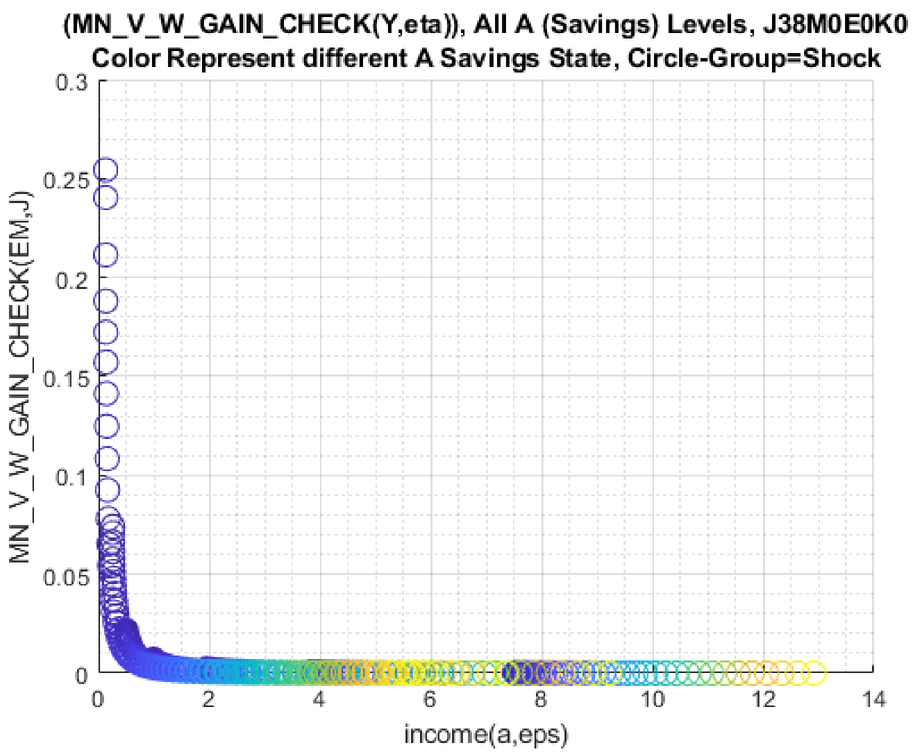



```
figure();
it_shock = 1;
scatter(log(mt_total_inc_jemk(:,it_a)), log(mt_V_W_gain_check_jemk(:,it_a)), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel(' of Y(a,eta) = Spouse + Household head Income Joint');
ylabel(' of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

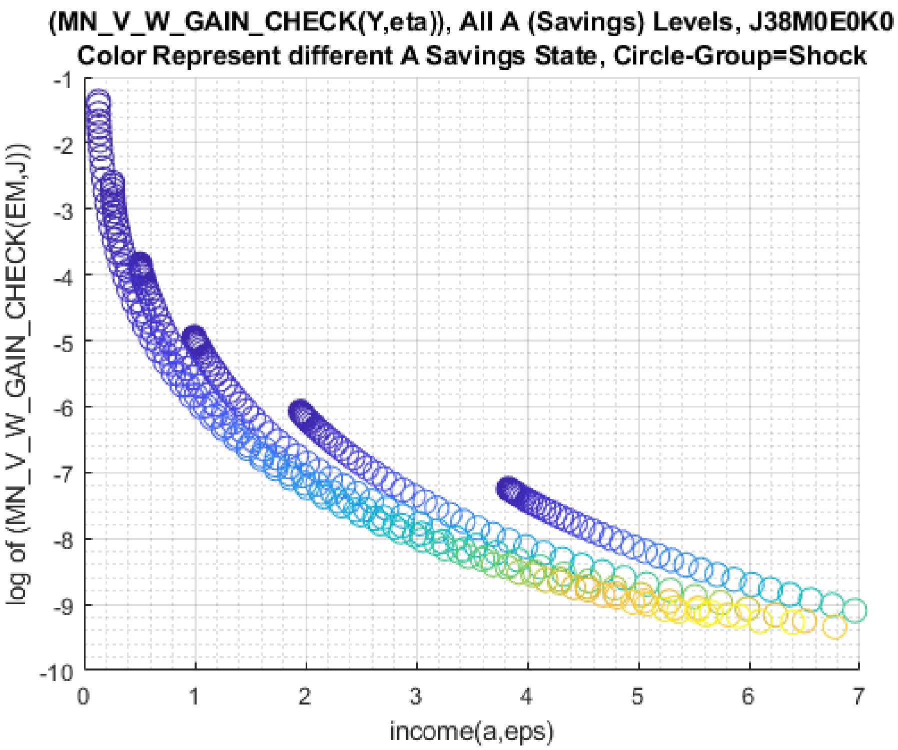


Plot all asset levels:

```
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;
```



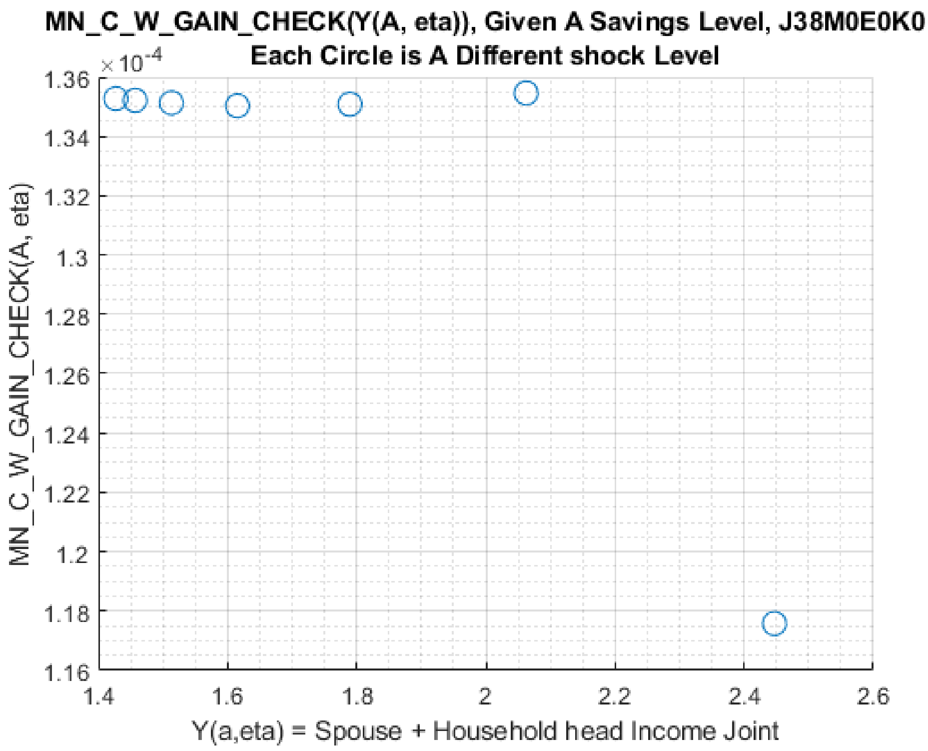

```
figure();
scatter((mt_total_inc_jemk(:)), log(mt_V_W_gain_check_jemk(:)), 100, mt_a(:));
title({'(MN\_V\_W\_GAIN\_CHECK(Y,eta)), All A (Savings) Levels, J38M0E0K0', ...
    'Color Represent different A Savings State, Circle-Group=Shock'});
xlabel('income(a,eps)');
ylabel('log of (MN\_V\_W\_GAIN\_CHECK(EM,J))');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



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

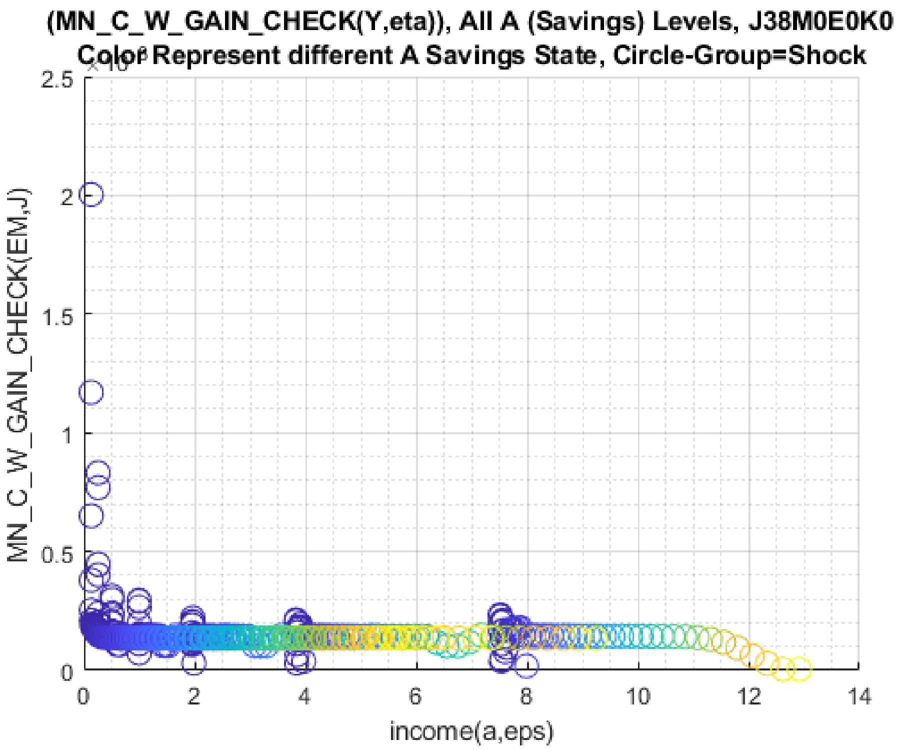
How do shocks and a impact marginal value. First plot one asset level, variation comes only from increasingly higher shocks:

```
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;
```

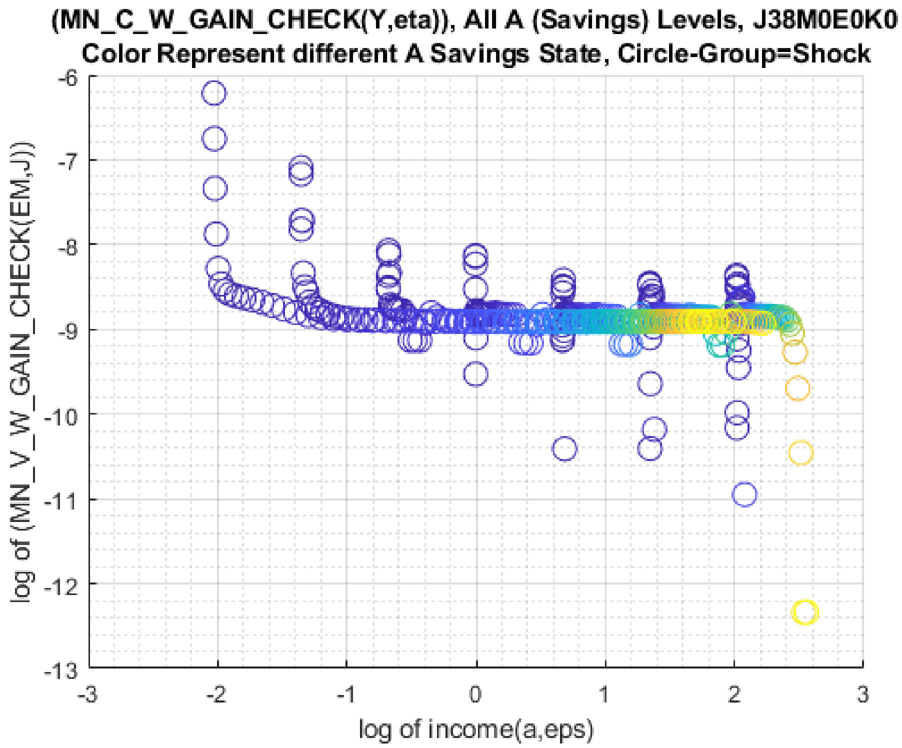


Plot all asset levels:

```
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;
```



7.2.8 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:

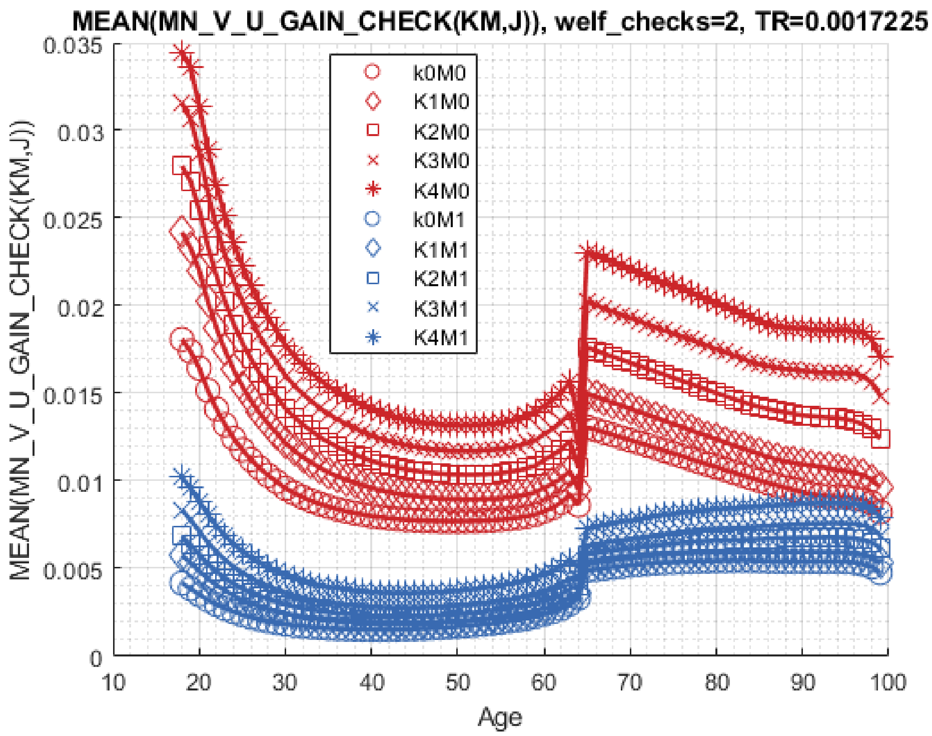
```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [2,3,4,1,6,5];
```

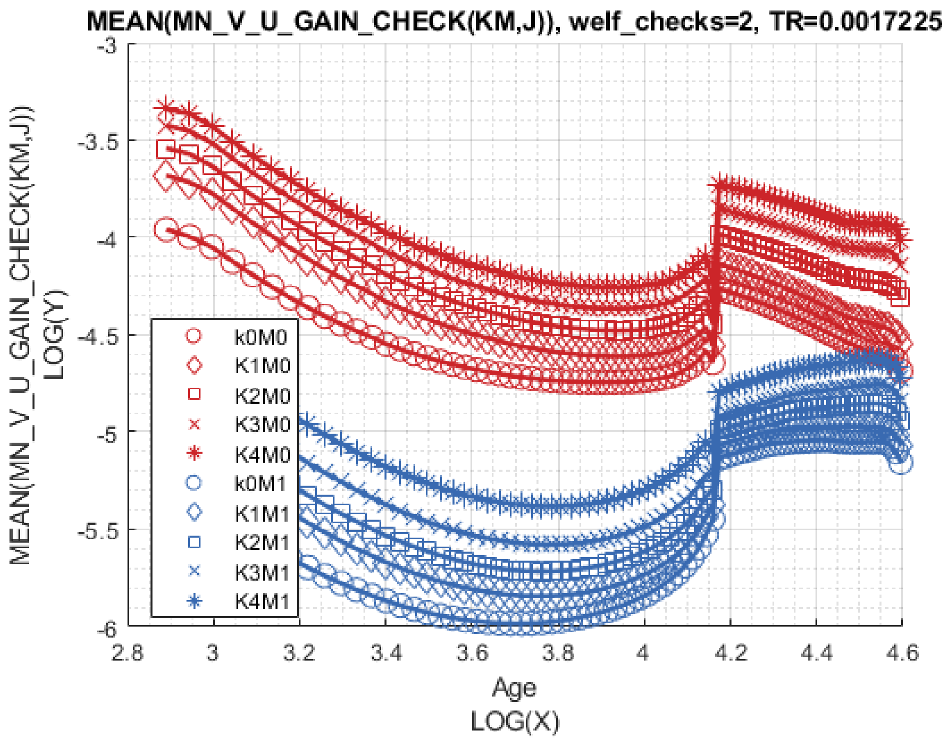
```
% Value Function
st_title = ['MEAN(MN_V_U_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2str(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

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_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0        0.018081       0.017399       0.016444       0.015175       0.01411
      2        2       0        0.024198       0.023312       0.022022       0.020238      0.018732
      3        3       0        0.027972       0.027119       0.025477       0.023434       0.02171
      4        4       0        0.031555       0.030668       0.028763       0.026469      0.024532
      5        5       0         0.03448       0.033614       0.031423       0.028959      0.026879
      6        1       1       0.0041769      0.0038579      0.0035234      0.0032138     0.0029477
      7        2       1       0.0057601      0.0053292       0.004867      0.0044266     0.0040531
      8        3       1       0.0068486      0.0063558      0.0058113      0.0052918     0.0048508
      9        4       1       0.0083574       0.007782      0.0071369       0.006497     0.0059602
     10        5       1        0.010213      0.0095931      0.0088366      0.0080803     0.0074291
```

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

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_2
    _____    ____    _____    ___________    ___________    ___________    ___________    __________

      1        1       0       0.074031        0.10225        0.11599        0.11147        0.10577
      2        2       0       0.089094        0.11803        0.13543          0.132        0.12978
      3        3       0        0.10426        0.14094        0.15528         0.1534        0.15067
      4        4       0         0.1102        0.15019        0.16182        0.16053         0.1588
      5        5       0        0.12088        0.15838        0.16724        0.16598        0.16431
      6        1       1       0.088885        0.10829        0.11259        0.10847        0.10521
      7        2       1       0.093837        0.11495         0.1194        0.11493        0.11003
      8        3       1        0.10258          0.126         0.1301        0.12812        0.12491
      9        4       1        0.10896         0.1303         0.1351        0.13564        0.13402
     10        5       1        0.12962        0.14112        0.14504        0.14418        0.14327
```

Graph Mean Values:

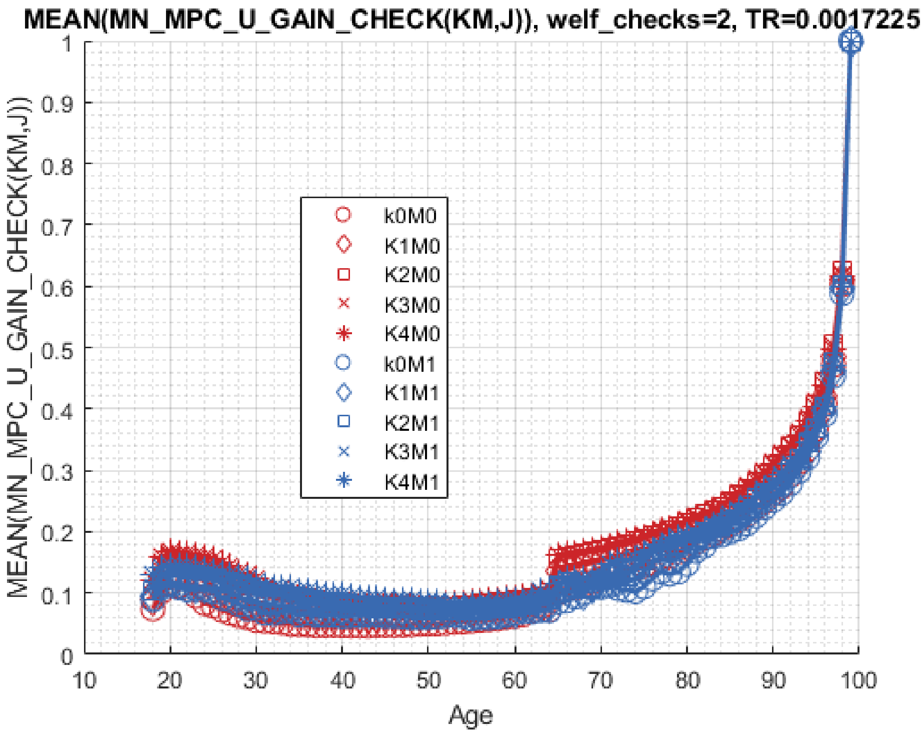
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

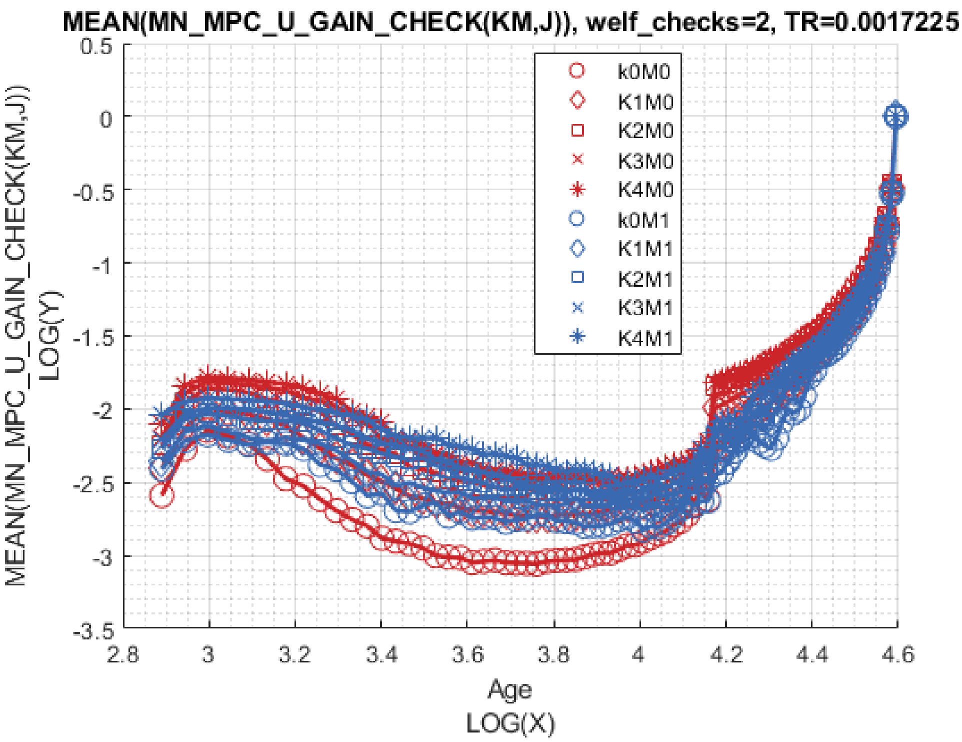




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
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);
```





7.2.9 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'};
```

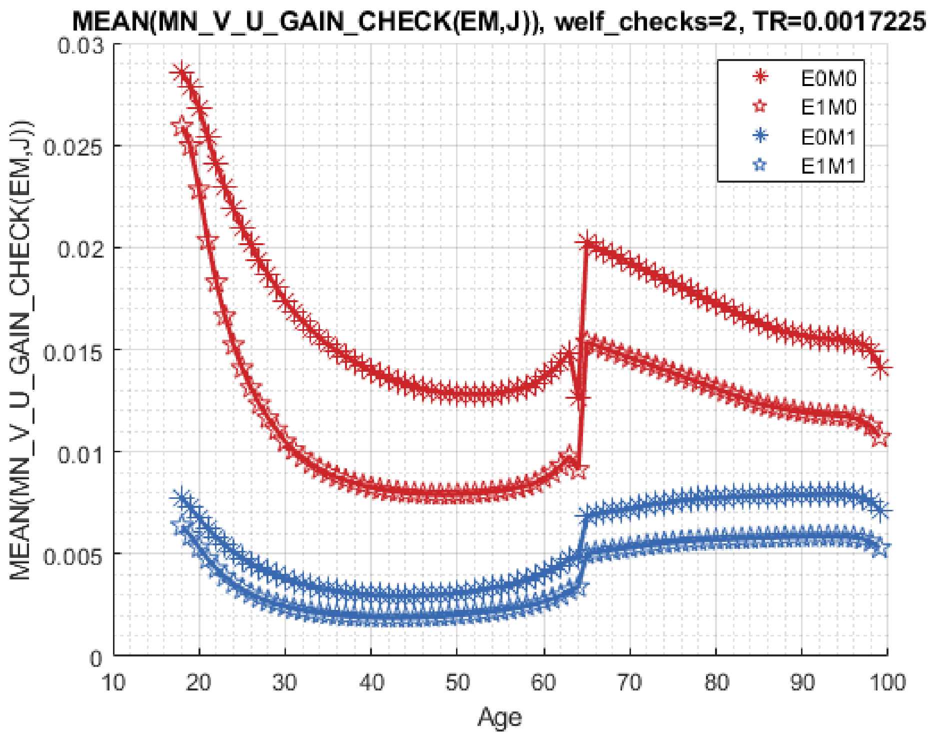
MEAN(VAL(EM,J)), MEAN(AP(EM,J)), MEAN(C(EM,J))

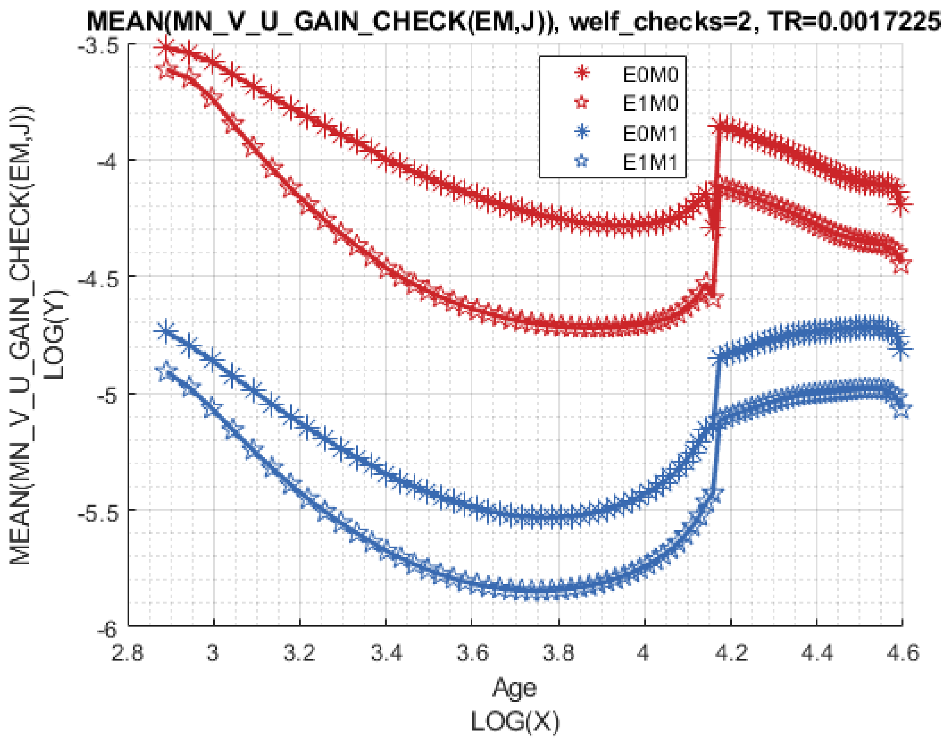
Tabulate value and policies:

```
% 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(mp_pa
tb_az_v = ff_summ_nd_array(st_title, mn_V_U_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc, ar_

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
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0        0.028598       0.027879       0.026816        0.02537       0.024081
      2       1       0        0.025917       0.024966       0.022835        0.02034       0.018305
      3       0       1       0.0077546      0.0072639      0.0067503      0.0062497      0.0058142
      4       1       1       0.0063878      0.0059033      0.0053197      0.0047542      0.0042822

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

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
    -----    ---    -----    -----------    -----------    -----------    -----------    -----------

      1       0       0       0.083839       0.10872        0.12214        0.11934        0.11723
      2       1       0        0.11555        0.1592        0.17216        0.17002         0.1665
      3       0       1       0.096421       0.11218        0.11688        0.11343        0.11043
      4       1       1        0.11313       0.13609        0.14001        0.13911        0.13655
```

Graph Mean Values:

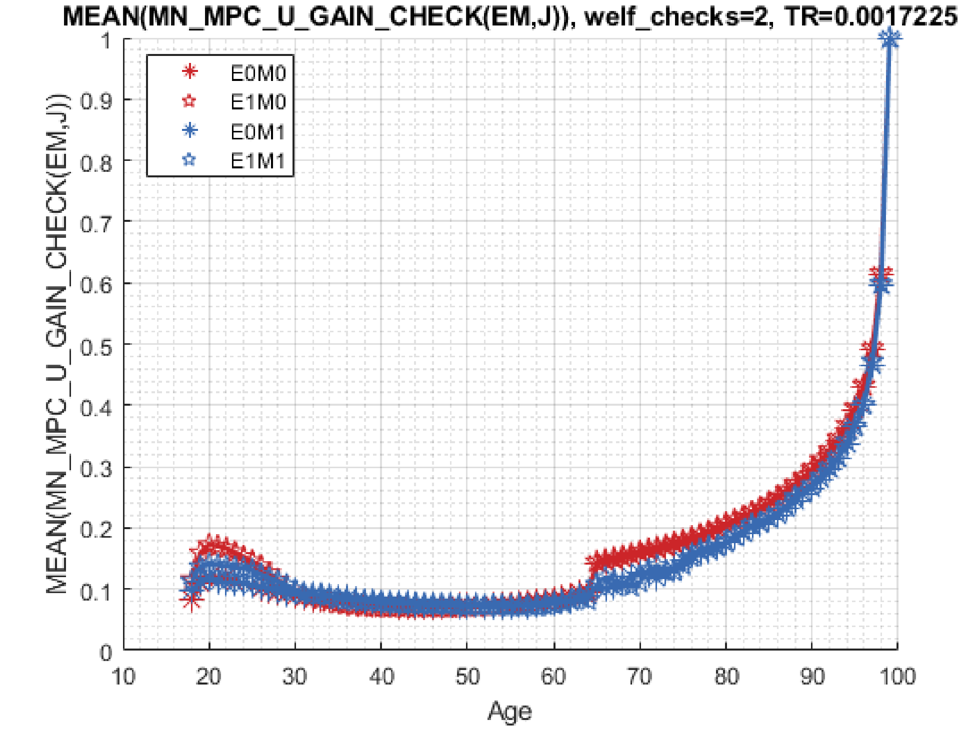
```
st_title = ['MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2str(
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_U\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

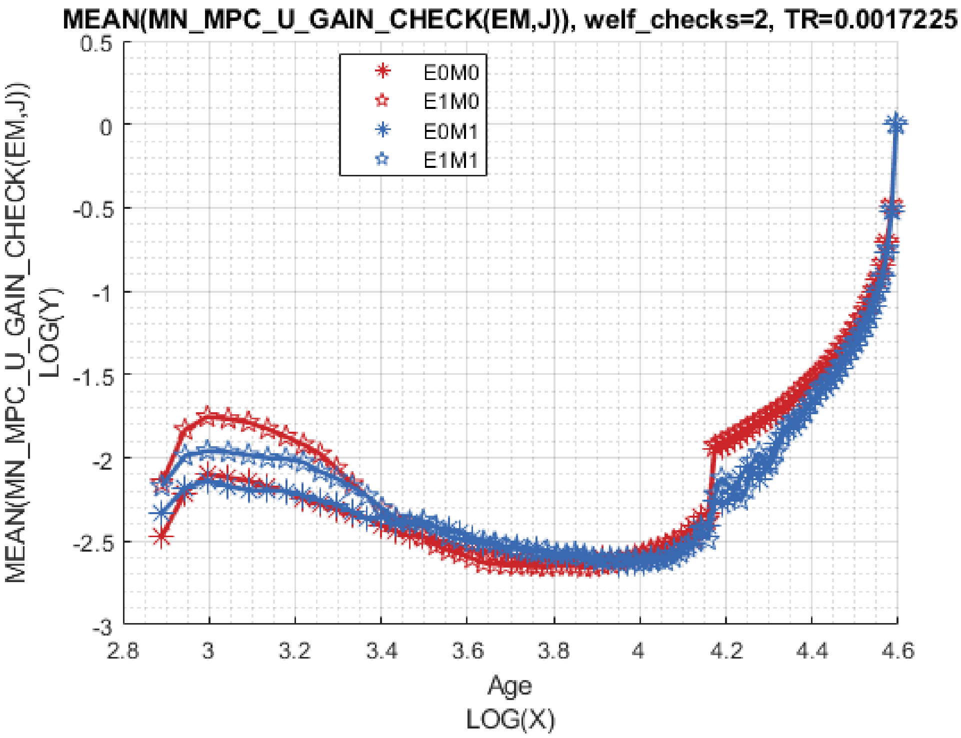




Graph Mean Consumption (***MPC: Share of Check Consumed***):

```
st_title = ['MEAN(MN\_MPC\_U\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num2st
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
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
10
20
30
40
50
60
70
80
90
100


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)
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
2.8
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6

Chapter 8

2019 Expectations Given Income, Age, Kids and Marital Status

8.1 2019 Age, Income, Kids, Marry EV and EC of One Check

This is the example vignette for function: **snw_evuvw20_jaeemk** from the **PrjOptiSNW Package.**
2019 integrated over VU and VW

8.1.1 Test SNW_EVUVW19_JMKY Defaults Dense

Set Parameters

Call the function with defaults.

```
clear all;
st_solu_type = 'bisec_vec';

% Solve the VFI Problem and get Value Function
% mp_params = snw_mp_param('default_moredense_a100z266_e0m0');
% mp_params = snw_mp_param('default_tiny');
% mp_params = snw_mp_param('default_dense');
mp_params = snw_mp_param('default_docdense');
mp_controls = snw_mp_control('default_test');

% set Unemployment Related Variables
xi=0.5; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=1
b=0; % Unemployment insurance replacement rate (b=0 refers to no UI benefits; b=1 refers to 100 perc
TR=100/58056; % Value of a welfare check (can receive multiple checks). TO DO: Update with alternati

mp_params('xi') = xi;
mp_params('b') = b;
mp_params('TR') = TR;

% Check Numbers
% n_incgrid=201; % Number of income groups
% n_incgrid_aux=round(0.75*n_incgrid);
% inc_grid1=linspace(0,4,n_incgrid_aux)'; % 4 refers to 4*58056=232224 dollars in 2012USD
% inc_grid=[inc_grid1;linspace(4+((7-4)/(n_incgrid-n_incgrid_aux)),7,n_incgrid-n_incgrid_aux)']; % 7
n_incgrid=201; % Number of income groups
inc_grid=linspace(0,7,n_incgrid)';
mp_params('n_incgrid') = n_incgrid;
mp_params('inc_grid') = inc_grid;
```

```
% Solve for Unemployment Values
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
mp_controls('bl_print_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jaeemk') = false;
mp_controls('bl_print_evuvw19_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jmky') = false;
```

8.1.2 Solve VFI and Distributon

```
% Solve the Model to get V working and unemployed
[V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);

Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;time=116

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;
% Solve unemployment
[V_unemp,~,cons_unemp,~] = snw_vfi_main_bisec_vec(mp_params, mp_controls, V_ss);

Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=

[Phi_true] = snw_ds_main(mp_params, mp_controls, ap_ss, cons_ss, mp_valpol_more_ss);

Completed SNW_DS_MAIN;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;time=192.6858

% Get Matrixes
cl_st_precompute_list = {'a', ...
    'inc', 'inc_unemp', 'spouse_inc', 'spouse_inc_unemp', 'ref_earn_wageind_grid', ...
    'ap_idx_lower_ss', 'ap_idx_higher_ss', 'ap_idx_lower_weight_ss', ...
    'inc_tot_ygroup_grid'};
mp_controls('bl_print_precompute_verbose') = false;
```

8.1.3 Pre-Compute Matrixes and YMKY Mass

```
% Pre-compute
[mp_precompute_res] = snw_hh_precompute(mp_params, mp_controls, cl_st_precompute_list, ap_ss, Phi_tr

Wage quintile cutoffs=0.48006     0.84085      1.2804      2.2175
Completed SNW_HH_PRECOMPUTE;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;time cost=123

inc_tot_ygroup_grid = mp_precompute_res('inc_tot_ygroup_grid');
% YMKY Mass
[Phi_true_jmky] = snw_evuvw19_jmky_mass(mp_params, mp_controls, Phi_true, inc_tot_ygroup_grid);

SNW_EVUVW19_JMKY_MASS Start
Completed SNW_EVUVW19_JMKY_MASS;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;time=2.66
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
                    i     idx    ndim      numel       rowN      colN         sum         mean
                    -     ---    ----    ----------    ----    ---------    ------    ----------

    Phi_true        1      1      6       1.128e+07     83     1.359e+05    45.793    4.0598e-06
    Phi_true_jmky   2      2      4      1.6482e+05     82          2010    45.787     0.0002778
```

8.1.4 Solve for 2019 Evuvw With 0 and 2 Checks

Zero checks:

```
% Solve ev 19 JAEEMK
welf_checks = 0;
[ev19_jaeemk_check0, ec19_jaeemk_check0, ev20_jaeemk_check0, ec20_jaeemk_check0] = ...
    snw_evuvw19_jaeemk(...
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss, cons_ss, V_unemp, cons_unemp, mp_precompute_res);

Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=0;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_mored
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=0;TR=0.0017225;SNW_MP_PARAM=default_moredense;SNW_MP_C
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;timeEUEC=2.0
Completed SNW_EVUVW19_JAEEMK;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;time=77.3603

% Solve ev 19 JMKY
[ev19_jmky_check0, ec19_jmky_check0] = snw_evuvw19_jmky(...
    mp_params, mp_controls, ...
    ev19_jaeemk_check0, ec19_jaeemk_check0, ...
    Phi_true, Phi_true_jmky, inc_tot_ygroup_grid);

Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;time=5.0081
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                    i     idx    ndim      numel       rowN      colN            sum          mean
                    -     ---    ----    ----------    ----    ---------    -----------    ---------

    Phi_true        1      1      6       1.128e+07     83     1.359e+05         45.793    4.0598e-0
    Phi_true_jmky   2      2      4      1.6482e+05     82          2010         45.787     0.000277
    ec19_jaeemk     3      3      6      1.1144e+07     82     1.359e+05     4.7069e+07        4.223
    ec19_jmky       4      4      4      1.6482e+05     82          2010     3.2335e+05        1.961
    ev19_jaeemk     5      5      6      1.1144e+07     82     1.359e+05    -2.1277e+07       -1.909
    ev19_jmky       6      6      4      1.6482e+05     82          2010         -16603      -0.1007
```

Two checks:

```
% Solve ev 19 JAEEMK
welf_checks = 1;
[ev19_jaeemk_check2, ec19_jaeemk_check2, ev20_jaeemk_check2, ec20_jaeemk_check2] = ...
    snw_evuvw19_jaeemk(...
    welf_checks, st_solu_type, mp_params, mp_controls, ...
    V_ss, cons_ss, V_unemp, cons_unemp, mp_precompute_res);

Completed SNW_A4CHK_UNEMP_BISEC_VEC;welf_checks=1;TR=0.0017225;xi=0.5;b=0;SNW_MP_PARAM=default_mored
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=1;TR=0.0017225;SNW_MP_PARAM=default_moredense;SNW_MP_C
Completed SNW_EVUVW20_JAEEMK;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;timeEUEC=1.9
Completed SNW_EVUVW19_JAEEMK;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;time=85.8135

% Solve ev 19 JMKY
[ev19_jmky_check2, ec19_jmky_check2] = snw_evuvw19_jmky(...
    mp_params, mp_controls, ...
```

```
    ev19_jaeemk_check2, ec19_jaeemk_check2, ...
    Phi_true, Phi_true_jmky, inc_tot_ygroup_grid);
```

```
Completed SNW_EVUVW19_JMKY;SNW_MP_PARAM=default_moredense;SNW_MP_CONTROL=default_test;time=4.5581
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                     i    idx    ndim      numel       rowN      colN          sum           mean
                     -    ---    ----    ----------    ----    ---------    -----------    ---------

    Phi_true         1     1      6       1.128e+07     83     1.359e+05         45.793    4.0598e-0
    Phi_true_jmky    2     2      4      1.6482e+05     82          2010         45.787     0.000277
    ec19_jaeemk      3     3      6      1.1144e+07     82     1.359e+05     4.7072e+07        4.224
    ec19_jmky        4     4      4      1.6482e+05     82          2010     3.2336e+05        1.961
    ev19_jaeemk      5     5      6      1.1144e+07     82     1.359e+05    -2.1225e+07       -1.904
    ev19_jmky        6     6      4      1.6482e+05     82          2010         -16281     -0.09877
```

Differences between Checks in Expected Value and Expected Consumption

```
mn_V_U_gain_check = ev19_jmky_check2 - ev19_jmky_check0;
mn_MPC_U_gain_share_check = (ec19_jmky_check2 - ec19_jmky_check0)./(welf_checks*mp_params('TR'));
```

8.1.5 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 = 18:99;
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
inc_grid = mp_params('inc_grid');
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'ylower', inc_grid});
```

8.1.6 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 37, 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_V_W_gain_check_use = ev19_jmky_check2 - ev19_jmky_check0;
mn_C_W_gain_check_use = ec19_jmky_check2 - ec19_jmky_check0;
```

Select Age, Education, Marital, Kids Count:s

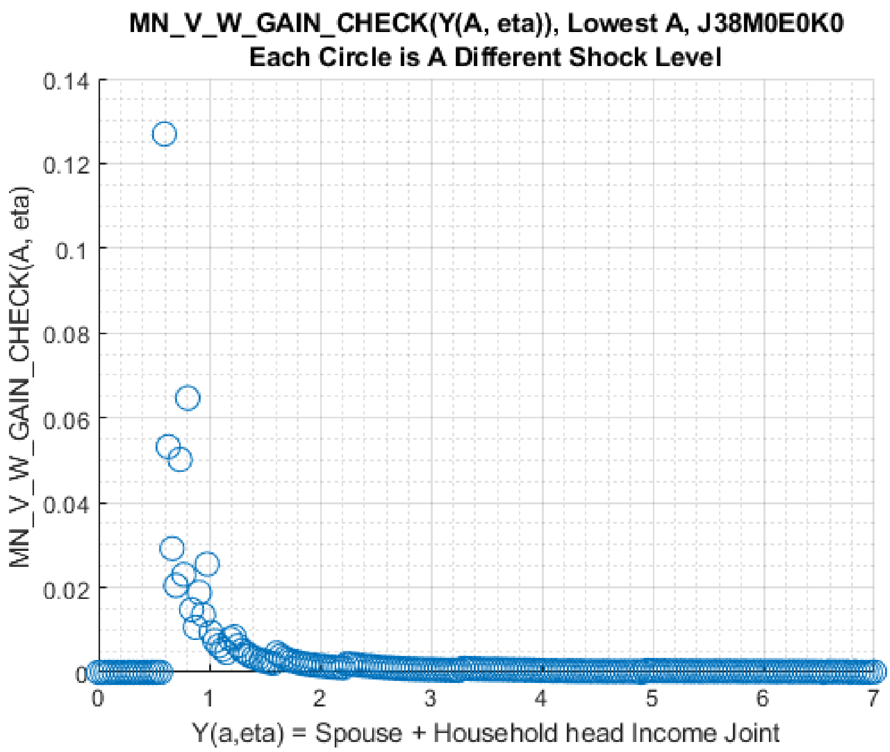
```
% Selections
it_age = 21; % +18
it_marital = 1; % 1 = unmarried
it_kids = 1; % 1 = kids is zero
% Select: NaN(n_jgrid-1,n_marriedgrid,n_kidsgrid,n_incgrid);
```

```
mn_C_W_gain_check_jemk = mn_C_W_gain_check_use(it_age, it_marital, it_kids, :);
mn_V_W_gain_check_jemk = mn_V_W_gain_check_use(it_age, it_marital, it_kids, :);
% Reshape, so shock is the first dim, a is the second
ar_C_W_gain_check_jemk = mn_C_W_gain_check_jemk(:);
ar_V_W_gain_check_jemk = mn_V_W_gain_check_jemk(:);
```

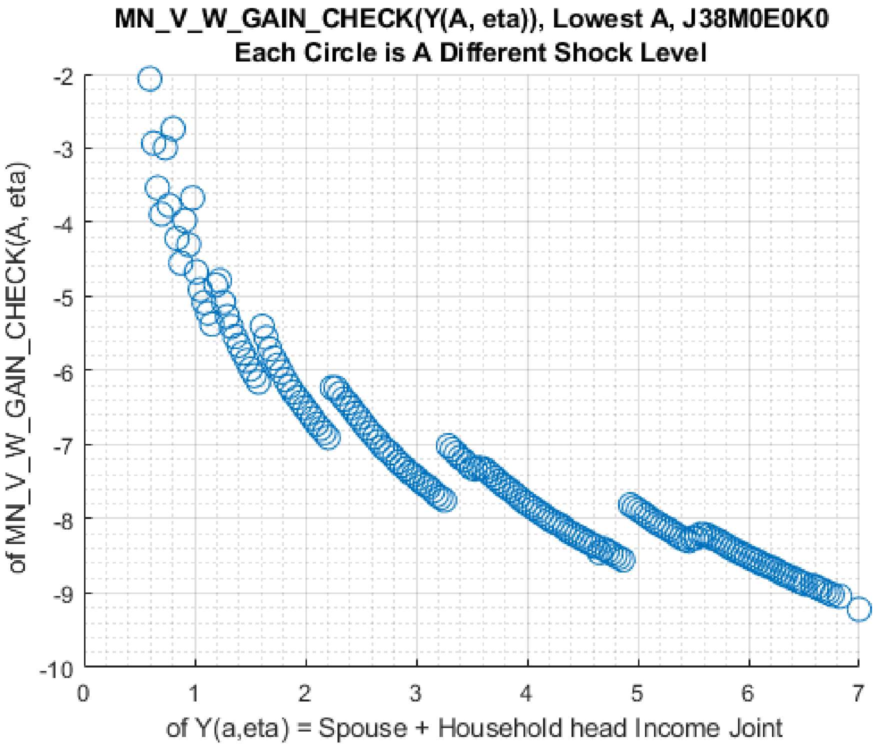
Marginal Value Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

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

```
figure();
scatter(inc_grid, ar_V_W_gain_check_jemk, 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;
```



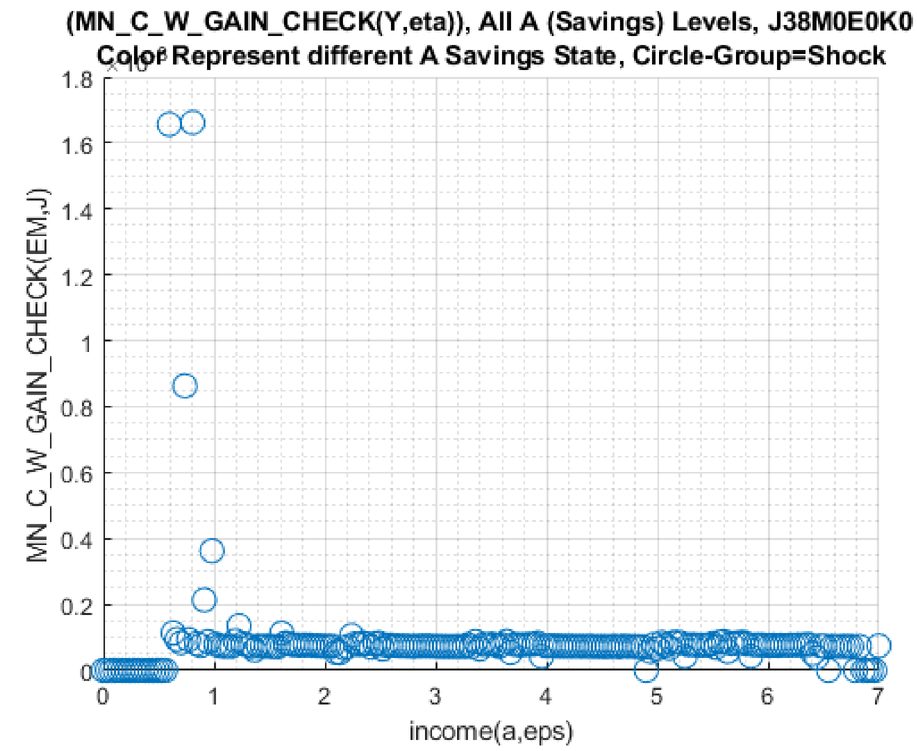

```
figure();
it_shock = 1;
scatter((inc_grid), log(ar_V_W_gain_check_jemk), 100);
title({'MN\_V\_W\_GAIN\_CHECK(Y(A, eta)), Lowest A, J38M0E0K0', ...
    'Each Circle is A Different Shock Level'});
xlabel(' of Y(a,eta) = Spouse + Household head Income Joint');
ylabel(' of MN\_V\_W\_GAIN\_CHECK(A, eta)');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



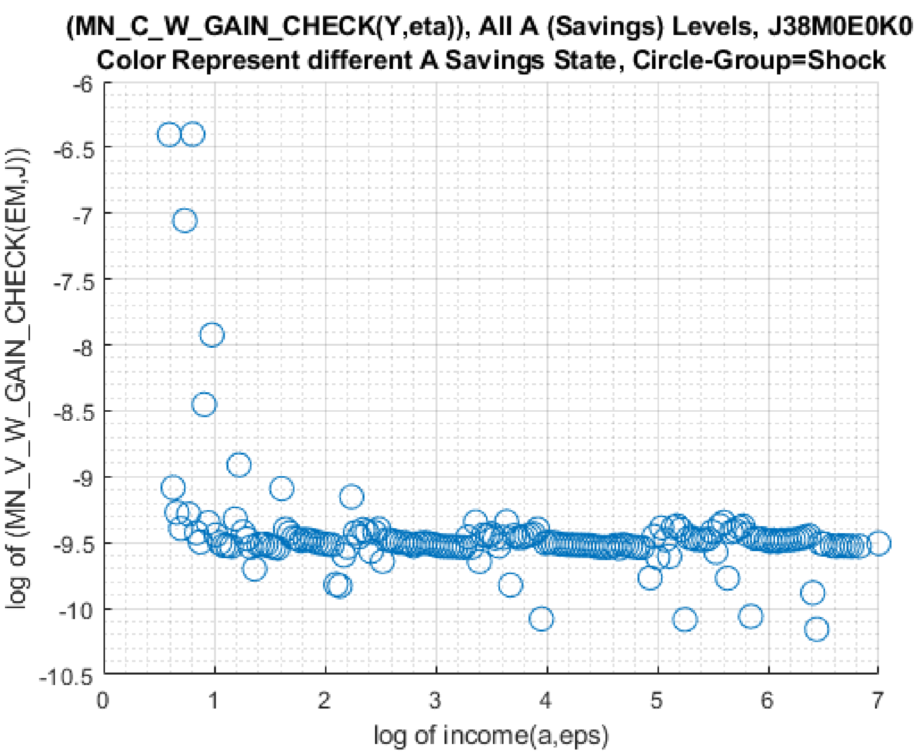
Marginal Consumption Gains, Color as Shock, Conditional on Age, Marital, Kids, and Education

Plot all asset levels:

```
figure();
scatter(inc_grid, ar_C_W_gain_check_jemk, 100);
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((inc_grid), log(ar_C_W_gain_check_jemk), 100);
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;
```



8.2 2019 Age, Income, Kids, Marry EV and EC All Checks

This is the example vignette for function: **snw_evuvw20_jaeemk** from the **PrjOptiSNW Package.** 2019 integrated over VU and VW

8.2.1 Test SNW_EVUVW19_JMKY_ALLCHECKS Parameters

Save a result that is low in memory cost so that it can be loaded quickly for various allocation tests. Turn off Various Printing Controls. Call function with wide income bins to reduce memory storage and retrievel costs

```
clear all;
% Start mp contorls
mp_controls = snw_mp_control('default_test');
% Solve for Unemployment Values
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_precompute') = false;
mp_controls('bl_print_precompute_verbose') = false;
mp_controls('bl_print_a4chk') = false;
mp_controls('bl_print_a4chk_verbose') = false;
mp_controls('bl_print_evuvw20_jaeemk') = false;
mp_controls('bl_print_evuvw20_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jaeemk') = false;
mp_controls('bl_print_evuvw19_jaeemk_verbose') = false;
mp_controls('bl_print_evuvw19_jmky') = false;
mp_controls('bl_print_evuvw19_jmky_verbose') = false;
```

Dense default, and unemployment parameters:

```
% default dense load
mp_params = snw_mp_param('default_docdense');
% Unemployment
xi=0.5; % Proportional reduction in income due to unemployment (xi=0 refers to 0 labor income; xi=1
b=0; % Unemployment insurance replacement rate (b=0 refers to no UI benefits; b=1 refers to 100 perc
TR=100/58056; % Value of a wezlfare check (can receive multiple checks). TO DO: Update with alternat
mp_params('xi') = xi;
mp_params('b') = b;
mp_params('TR') = TR;
% Check Count: 89 checks to allow for both the first and the second round
n_welfchecksgrid = 3;
mp_params('n_welfchecksgrid') = n_welfchecksgrid;
```

Income bins:

```
% Income Grid
% 4 refers to 4*58056=232224 dollars in 2012USD
% max 7 refers to 7*58056=406392 dollars in 2012USD
% all phase out = (4400/5)*100 + 150000 = 238000
% if 500 dollar interval, need 476 inc groups before 238000
% if have 85 percent of points betwen 238000,
fl_max_phaseout = 238000;
fl_multiple = 58056;
it_bin_dollar_before_phaseout = 5000;
it_bin_dollar_after_phaseout = 25000;
fl_thres = fl_max_phaseout/fl_multiple;
inc_grid1 = linspace(0,fl_thres,(fl_max_phaseout)/it_bin_dollar_before_phaseout);
inc_grid2 = linspace(fl_thres, 7, (7*fl_multiple-fl_max_phaseout)/it_bin_dollar_after_phaseout);
inc_grid=sort(unique([inc_grid1 inc_grid2]'));
```

```
mp_params('n_incgrid') = length(inc_grid);
mp_params('inc_grid') = inc_grid;
```

8.2.2 SNW_EVUVW19_JMKY_ALLCHECKS Low Storage Invoke

The simulation here (dense) requires less than 10 GB of memory with 8 workers (8 threads needed), simulating over 88 checks takes with 8 workers

```
st_solu_type = 'bisec_vec';
bl_parfor = false;
it_workers = 1;
bl_export = false;
snm_suffix = ['_ybin' num2str(it_bin_dollar_before_phaseout)];
[ev19_jmky_allchecks, ec19_jmky_allchecks, output] = ...
    snw_evuvw19_jmky_allchecks(mp_params, mp_controls, ...
    st_solu_type, bl_parfor, it_workers, bl_export, snm_suffix);
```

Chapter 9

Calibration

9.1 Calibrate Beta and Normalize GDP

Taking advantage of **snw_ds_main** from the **PrjOptiSNW Package**, this function calibrates the discount factor and also solves for the normalizing constant.

9.1.1 Calibrate Parameter Controls for SNW Functions

Set up controls for shock process and tiny/small/dense/densemore

```
clear all;
bl_print_mp_params = false;
% st_shock_method = 'rouwenhorst';
st_shock_method = 'tauchen';
% st_param_group = 'default_tiny';
% st_param_group = 'default_small';
% st_param_group = 'default_base';
% st_param_group = 'default_dense';
% st_param_group = 'default_moredense';
st_param_group = 'default_docdense';
mp_params = snw_mp_param(st_param_group, bl_print_mp_params, st_shock_method);
Pop = mp_params('Pop');
```

Set up print defaults

```
mp_controls = snw_mp_control('default_test');
mp_controls('bl_timer') = timer;
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;
```

9.1.2 Calibrate Routine

Test this for 3 iterations

```
%% Calibration
err=1;
tol=0.005;

Start calibration

it_counter = 1;
while err>tol && it_counter <= 3
    disp('');
```

```
it=1;

while it>0

    % Solve optimization problem and get the distribution
    tm_start_a2 = tic;
    a2_old = mp_params('a2');
    [Phi_true,~,A_agg,Y_inc_agg,it,mp_dsvfi_results, a2] = snw_ds_main(mp_params, mp_controls);
    mp_params('a2') = a2;
    tm_end_a2 = toc(tm_start_a2);
    disp(['a2_old:' num2str(a2_old) ', a2_new:' num2str(a2) ', tm_end_a2:' num2str(tm_end_a2)])
end

% Get Stats
mp_cl_mt_xyz_of_s = mp_dsvfi_results('mp_cl_mt_xyz_of_s');
tb_outcomes = mp_cl_mt_xyz_of_s('tb_outcomes');
A_agg_alt = tb_outcomes{'a_ss', 'mean'}*sum(Pop);
Aprime_agg_alt = tb_outcomes{'ap_ss', 'mean'}*sum(Pop);
Y_inc_agg_alt = tb_outcomes{'y_all', 'mean'}*sum(Pop);
Y_inc_median = tb_outcomes{'y_all', 'p50'};

% Comparison
name='Median household income (target=1.0)=';
name2=[name,num2str(Y_inc_median)];
disp(name2);
name='Aggregate wealth to aggregate income (target=3.0)=';
name2=[name,num2str(A_agg/Y_inc_agg)];
disp(name2);

err1=abs(Y_inc_median-1.0); % Target: Median household income (normalized to 1 in the model)
err2=abs((A_agg/Y_inc_agg)-3.0); % Target: Annual capital/income ratio of 3

err=max(err1,err2);

% Beta and Theta
theta = mp_params('theta');
beta = mp_params('beta');
param_update=[theta;beta];

if err>tol

    theta=theta*((1.0/Y_inc_median)^0.2); % Normalize theta such that median household income eq
    beta=beta*((3.0/(A_agg/Y_inc_agg))^0.025); % Calibrate beta such that annual capital/income

end
mp_params('theta') = theta;
mp_params('beta') = beta;

param_update=[param_update(1,1),theta;param_update(2,1),beta];

it_counter = it_counter + 1;
name='Old/updated theta:';
st_theta=[name, num2str(param_update(1,:))];
name='Old/updated beta:';
st_beta=[name,num2str(param_update(2,:))];
disp(['counter=' num2str(it_counter) ...
    ';beta=' num2str(beta) ...
    ';theta=' num2str(theta)]);
```

```
end
```

```
a2_old:1.5286, a2_new:1.5286, tm_end_a2:4134.1075
Median household income (target=1.0)=0.99853
Aggregate wealth to aggregate income (target=3.0)=3.0026
counter=2;beta=0.97116;theta=0.56523
```

Appendix A

Index and Code Links

A.1 Parameters links

1. Model Parameters: **mlx** | **m** | **pdf** | **html**
   - Model parameters, transition matrices, permanent heterogeneities.
   - **PrjOptiSNW**: *snw_mp_param()*
2. Model Controls Parameters: **mlx** | **m** | **pdf** | **html**
   - Parameters to control display options etc.
   - **PrjOptiSNW**: *snw_mp_control()*

A.2 Solving the Dynamic Life Cycle Problem links

1. Policy and Value Functions Dynamic Life Cycle Vectorized Bisection: **mlx** | **m** | **pdf** | **html**
   - Solving for policy and value functions from 18 to 100 years of age, at 1 year interval.
   - Households face persistent productivity shocks for household heads, stochastic shocks for spousal income, exogenous children under age 17 transition probability, and age-specific household-head survival probabilities.
   - The household can have up to four children under age 17, and has permanent heterogeneity in marital status and education types.
   - Problem solved for exact savings choices using vectorized bisection from from MEconTools.
   - **PrjOptiSNW**: *snwx_vfi_bisec_vec()*

A.3 Alternative Value Function Solution Testing links

1. Small Test Looped Minimizer Routine to Solve for Exact Savings Choices: **mlx** | **m** | **pdf** | **html**
   - Solve for the exact savings choices using matlab minimizer in an iterative loop.
   - The code demonstrates the solution structure. We use snwx_vfi_bisec_vec() with vectorized bisection for working implementations.
   - Due to speed, only show testing results at small grid without spousal shocks.
   - **PrjOptiSNW**: *snw_vfi_main()*
2. Small Test Looped over States Grid Search Solution: **mlx** | **m** | **pdf** | **html**
   - The savings choice grid is the same as the savings states grid. Solve for optimal savings choices using grid-search. Loop over the state space, at each state-space point, vectorized optimization.
   - Our problem requires very high precision to solve for the marginal gains to households from each increment of welfare checks. We rely on the exact solution method from snwx_vfi_bisec_vec() for the working code.
   - Due to speed, only show testing results at small grid without spousal shocks.
   - **PrjOptiSNW**: *snw_vfi_main_grid_search()*
3. Small Test Vectorized Bisection Solve for Exact Savings Choices: **mlx** | **m** | **pdf** | **html**

- Vectorized bisection exact solution code tested with small grid to compare to alternative solutiom methods.
- Small grid without spousal shocks.
- **PrjOptiSNW**: *snwx_vfi_bisec_vec()*

4. Small Test Spousal Shocks Test Vectorized Bisection Solve for Exact Savings Choices: **mlx** | **m** | **pdf** | **html**
   - Vectorized bisection exact solution code tested with small grid to compare to alternative solutiom methods.
   - Small grid with spousal shocks. There are three shocks: persistent household head income shock, i.i.d. spousal income shock, and persistent kids count transition shocks.
   - **PrjOptiSNW**: *snwx_vfi_bisec_vec()*

A.4 Solution with Unemployment links

1. Policy and Value Functions Dynamic Life Cycle if Unemployed: **mlx** | **m** | **pdf** | **html**
   - Solving the dynamic programming problem conditional on having an one period unemployment shock.
   - There is an unemployment shock in 2020. We first solve for the policy and value functions without the unemployment shock.
   - Using the value function from the world without the 2020 covid unemployment shock as future values, we solve for optimal choices in 2020 given a COVID unemployment shock.
   - The COVID shock lowers the realization of household's stochastic income process proportionally, but the lost income might be replenished by unemployment benefits up to 100 percent. Unemployment benefits have to be paid for by taxes.
   - **PrjOptiSNW**: *snwx_vfi_bisec_vec()*

A.5 Household Life Cycle Distribution links

1. Assets and Demographic Distributions with Continuous Exact Savings Choices: **mlx** | **m** | **pdf** | **html**
   - Simulate the life cycle distribution of assets, consumptions, and demographic patterns up to age 100, given exogenous initial distributions at age 18.
   - Solves for budget clearing tax rates given distributional results.
   - Uses vectorized bisection to solve for exact savings choices, looped distribution code.
   - **PrjOptiSNW**: *snw_ds_main()*
2. Assets and Demographic Distributions with Grid Search: **mlx** | **m** | **pdf** | **html**
   - Grid search solution using grid search for savings choices, the savings state-space grid is the same as the savings choice-grid.
   - Exact choice solution from snw_ds_main() generates significantly smoother distributions.
   - **PrjOptiSNW**: *snw_ds_main_grid_search()*

A.6 Value of Each Check links

1. Marginal Gain Per Check 2020 Employed: **mlx** | **m** | **pdf** | **html**
   - Evaluate the marginal gain per check in 2020 if household head is employed.
   - Solve for the increase in savings that is equivalent to the impact of an additional check on a household's resource available in 2020, given tax and interest rates considerations.
   - **PrjOptiSNW**: *snw_a4chk_wrk_bisec_vec()*
2. Marginal Gain Per Check 2020 Unemployed: **mlx** | **m** | **pdf** | **html**
   - Evaluate the marginal gain per check in 2020 if household head is unemployed.
   - Solve for the increase in savings that is equivalent to the impact of an additional check on a household's resource available in 2020, given tax and interest rates considerations.
   - **PrjOptiSNW**: *snw_a4chk_unemp_bisec_vec()*

A.7 Value of Each Check links

1. Marginal Gain Per Check 2020 Employed: **mlx** | **m** | **pdf** | **html**
   - Evaluate the marginal gain per check in 2020 if household head is employed.
   - Solve for the increase in savings that is equivalent to the impact of an additional check on a household's resource available in 2020, given tax and interest rates considerations.
   - **PrjOptiSNW**: *snw_a4chk_wrk_bisec_vec()*
2. Marginal Gain Per Check 2020 Unemployed: **mlx** | **m** | **pdf** | **html**
   - Evaluate the marginal gain per check in 2020 if household head is unemployed.
   - Solve for the increase in savings that is equivalent to the impact of an additional check on a household's resource available in 2020, given tax and interest rates considerations.
   - **PrjOptiSNW**: *snw_a4chk_unemp_bisec_vec()*

A.8 Outcomes Full State Space with Savings, Shocks and Education links

1. Value in 2020 Given Age, Savings, Shocks, Kids, Educaiton and Marriage: **mlx** | **m** | **pdf** | **html**
   - Expected value and expected consumption from 2020 for a household given at a particular age (18-100), with a particular savings level, at a particular combination of household head and spouse income shocks, with 0 to 4 children, high or low Education status, and married or not married.
   - This uses the unemployment probability and generates the average value given the probability of the unemployment state that is dependent on the state-space.
   - **PrjOptiSNW**: *snw_evuvw20_jaeemk()*
2. Expected Value in 2019 Given Age, Savings, Shocks, Kids, Educaiton and Marriage: **mlx** | **m** | **pdf** | **html**
   - Expected value and expected consumption from 2019 for a household at a particular age (18-99), savings level, shocks combinations, kids/education/marriage status, given 2019 optimal savings choices, income shock transition probability as well as household children count transition probabilities.
   - **PrjOptiSNW**: *snw_evuvw19_jaeemk()*

A.9 Expectations Given Income, Age, Kids and Marital Status links

1. Expected Value from 2019 Given Age, Kids, Income and Marriage: **mlx** | **m** | **pdf** | **html**
   - Expected Value from 2019 Given Age, Kids, Income and Marriage.
   - Each 2019 income group consists of individuals with varying productivity shocks, savings, and from lower and higher education groups.
   - **PrjOptiSNW**: *snw_evuvw19_jmky()*
2. Expected Value from 2019 Given Age, Kids, Income and Marriage for All Checks: **mlx** | **m** | **pdf** | **html**
   - Expected Value from 2019 Given Age, Kids, Income and Marriage for All Checks.
   - This is the gateway function that solves policy functions, derive distributions, computes value in 2020 with and without unemployment shocks with varying check levels, derives 2019 planner expected values given household optimization and shocks, and finds the mass of individuals in different income/age/marital-status bins, and saves the simulated value of check results for the planner.
   - **PrjOptiSNW**: *snw_evuvw19_jmky_allchecks()*

A.10 Calibration links

1. Calibrate Discount Factor and Normalize GDP: **mlx** | **m** | **pdf** | **html**
   - We calibrate the model so that the Asset/Savings/Capital to GDP/Income ratio is 3.
   - We normalize the model so that median household income is equal to 1 in the model.
   - **PrjOptiSNW**: *snw_calibrate_beta_norm_gdp()*

Bibliography

The MathWorks Inc (2019). *MATLAB*. Matlab package version 2019b.

Vegard M. Nygaard, Bent E. Sorensen, F. W. (2020). Optimal allocation of the covid-19 stimulus checks.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.18.