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.

Test SNW_A4CHK_WRK_BISEC_VEC Defaults Dense

Call the function with default parameters. Solve first for non-covid value and policy. Then depending on 2020 taxes, solve for 2020 policy and value.

```
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);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=490.7157
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                i    idx    ndim     numel      rowN      colN           sum         mean       std      coefvari
                _    ___    ____    ________    ____    _________    ___________    _______    ______    ________

    V_VFI       1     1      6      4.37e+07     83     5.265e+05    -1.5339e+08    -3.5101    26.119     -7.441
    ap_VFI      2     2      6      4.37e+07     83     5.265e+05     1.4159e+09     32.402    36.798     1.1357
    cons_VFI    3     3      6      4.37e+07     83     5.265e+05     2.1402e+08     4.8975    8.3294     1.7007

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
              c1         c2         c3         c4         c5       c526496    c526497    c526498    c526499    c52650
           _______    _______    _______    _______    _______    _______    _______    _______    _______    ______

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

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
           c1    c2        c3             c4            c5        c526496    c526497    c526498    c526499    c526500
           __    __    __________    _________    ________    _______    _______    _______    _______    _______

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

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
```

```
             c1          c2          c3          c4          c5        c526496    c526497    c526498    c526499    c
          ________    ________    ________    ________    ________    _______    _______    _______    _______    _

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

```
welf_checks = 2; % 2 checks is $200 dollar of welfare checks
xi=1; % xi=0 full income loss from covid shock, xi=1, no covid income losses
b=1; % when xi=1, b does not matter, no income losses
TR = 100/58056;
mp_params('TR') = TR;
mp_params('xi') = xi;
mp_params('b') = b;
% if = mp_params('a2_covidyr_manna_heaven'), V_emp_2020 same as V_ss if b=1
% or xi=1.
% if = mp_params('a2_covidyr_tax_fully_pay'), V_emp_2020 differ due to 2020
% tax differences
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
% mp_params('a2_covidyr') = mp_params('a2_covidyr_tax_fully_pay');
[V_emp_2020,~,cons_emp_2020,~] = 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_docdense;SNW_MP_CONTROL=default_test;time
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_outcomes ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                 i    idx    ndim     numel      rowN      colN          sum         mean      std      coefvari
                 _    ___    ____    ________    ____    _________    __________    ______    ______    ________

    V_VFI        1     1      6      4.37e+07     83     5.265e+05    -1.5339e+08   -3.5101   26.119     -7.441
    ap_VFI       2     2      6      4.37e+07     83     5.265e+05     1.4159e+09    32.402   36.798     1.1357
    cons_VFI     3     3      6      4.37e+07     83     5.265e+05     2.1402e+08    4.8975   8.3294     1.7007

xxx TABLE:V_VFI xxxxxxxxxxxxxxxxxx
            c1         c2         c3         c4         c5       c526496    c526497    c526498    c526499    c52650
          _______    _______    _______    _______    _______    _______    _______    _______    _______    ______

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

xxx TABLE:ap_VFI xxxxxxxxxxxxxxxxxx
          c1    c2        c3            c4           c5       c526496    c526497    c526498    c526499    c526500
          __    __    __________    _________    ________    _______    _______    _______    _______    _______

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

xxx TABLE:cons_VFI xxxxxxxxxxxxxxxxxx
              c1          c2          c3          c4          c5       c526496    c526497    c526498    c526499    c
           ________    ________    ________    ________    ________    _______    _______    _______    _______    _

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

```
[V_W_2020, C_W_2020] = snw_a4chk_wrk_bisec_vec(welf_checks, V_emp_2020, cons_emp_2020, mp_param
```

```
Completed SNW_A4CHK_WRK_BISEC_VEC;welf_checks=2;TR=0.0017225;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_te
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_container_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                      i    idx    ndim     numel     rowN      colN          sum          mean          std
                      _    ___    ____    ________    ____    _________    __________    __________    __________

    C_W               1     1      6      4.37e+07     83     5.265e+05    2.1404e+08        4.8981        8.3296
    C_W_minus_C_ss    2     2      6      4.37e+07     83     5.265e+05         23044    0.00052732    0.00076007
    V_W               3     3      6      4.37e+07     83     5.265e+05   -1.5281e+08       -3.4969         26.07
    V_W_minus_V_ss    4     4      6      4.37e+07     83     5.265e+05    5.7996e+05      0.013271      0.069912
    mn_MPC            5     5      6      4.37e+07     83     5.265e+05    6.6891e+06       0.15307       0.22063
```

```
mn_V_W_gain_check = V_W_2020 - V_emp_2020;
mn_MPC_W_gain_share_check = (C_W_2020 - cons_emp_2020)./(welf_checks*mp_params('TR'));
```

Dense Param Results Define Frames

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

```
% Grids:
age_grid = 18:100;
agrid = mp_params('agrid')';
eta_H_grid = mp_params('eta_H_grid')';
eta_S_grid = mp_params('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_params('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
```

```
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

Analyze Difference in V and C with Check

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

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

MEAN(MN_V_GAIN_CHECK(A,Z))

Tabulate value and policies along savings and shocks:

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

```
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    mean_eta_6    mean_
    _____    __________    __________    __________    __________    __________    __________    __________    ____

      1               0       0.93933        0.8402       0.75179       0.67285       0.60245       0.53977       0.
      2      0.00051498       0.92869       0.83136       0.74443       0.66672       0.59732       0.53546        0
      3       0.0041199       0.77596       0.70579       0.63957       0.57819       0.52215       0.47146       0.
      4        0.013905       0.59506       0.54978       0.50512       0.46222       0.42198       0.38483       0.
      5        0.032959        0.4333       0.40626       0.37812       0.35006       0.32301        0.2975       0.
      6        0.064373       0.31482       0.29826       0.28025       0.26173       0.24347       0.22597       0.
```

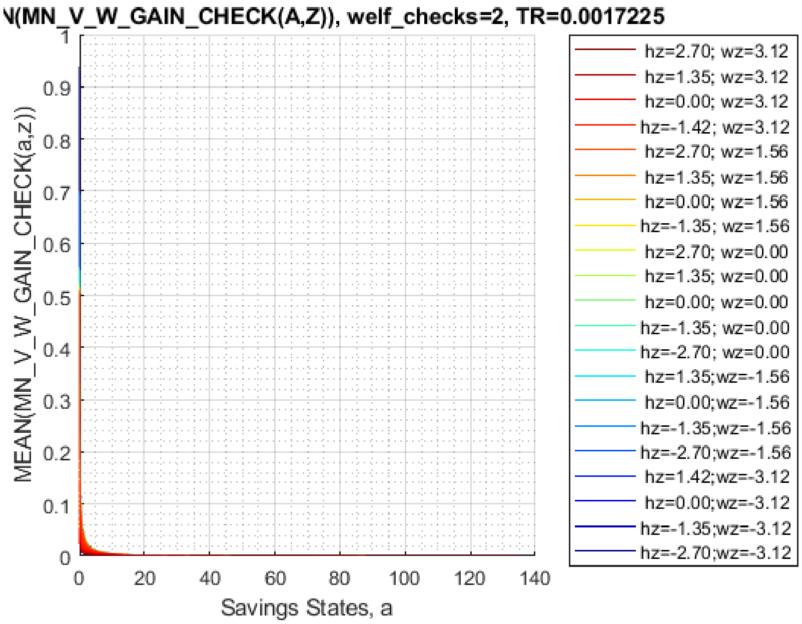
```
% Consumption
st_title = ['MEAN(MN_MPC_W_GAIN_CHECK(A,Z)), welf_checks=' num2str(welf_checks) ', TR=' num2str
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_W_gain_share_check, true, ["mean"], 4, 1, cl_mp_dat
```

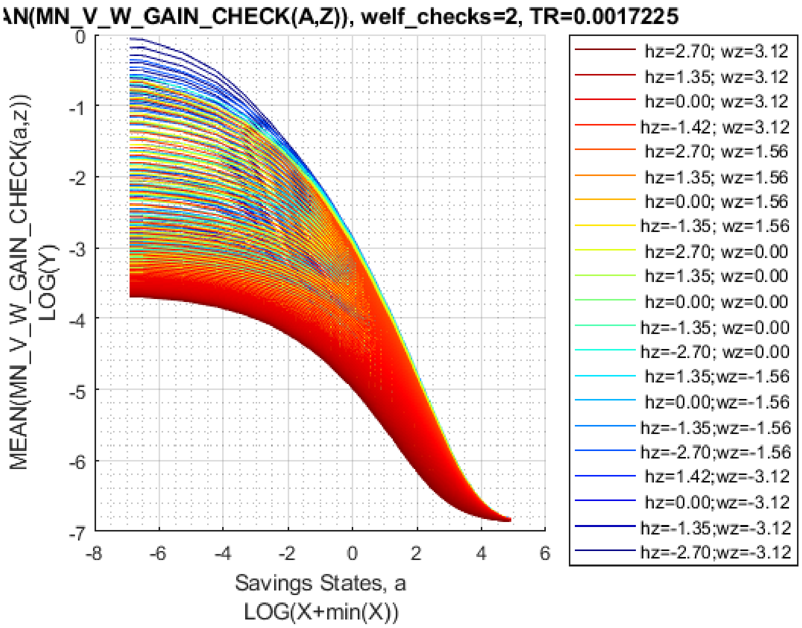
```
xxx  MEAN(MN_MPC_W_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group     savings      mean_eta_1    mean_eta_2    mean_eta_3    mean_eta_4    mean_eta_5    mean_eta_6    mean_
    _____    __________    __________    __________    __________    __________    __________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

      1               0     0.99507       0.98976       0.98416       0.98178       0.98145       0.98218       0.98
      2      0.00051498     0.99417       0.98789       0.98126       0.97855       0.97836       0.97937       0.98
      3       0.0041199     0.87132       0.86834       0.86648       0.86492        0.8641       0.86378       0.86
      4        0.013905     0.78572       0.77918       0.77443       0.77198       0.77167       0.77258        0.7
      5        0.032959     0.70053       0.69561        0.6935       0.69298       0.69308        0.6937       0.69
      6        0.064373     0.63139       0.63141        0.6321       0.63315       0.63436       0.63579       0.63
```

Graph Mean Values:

```
st_title = ['MEAN(MN\_V\_W\_GAIN\_CHECK(A,Z)), welf\_checks=' num2str(welf_checks) ', TR=' num2
```

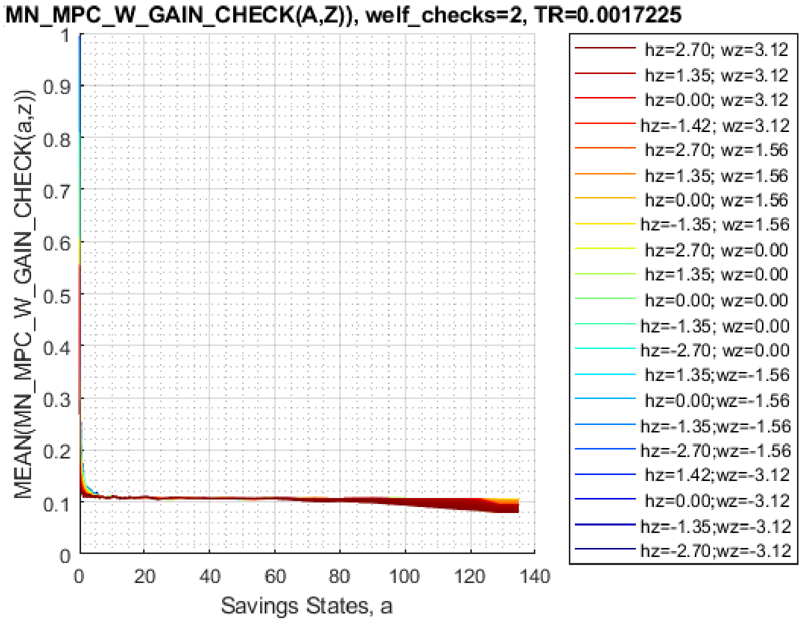
```
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_W\_GAIN\_CHECK(a,z))'};
ff_graph_grid((tb_az_v{1:end, 3:end})', ar_st_eta_HS_grid, agrid, mp_support_graph);
```

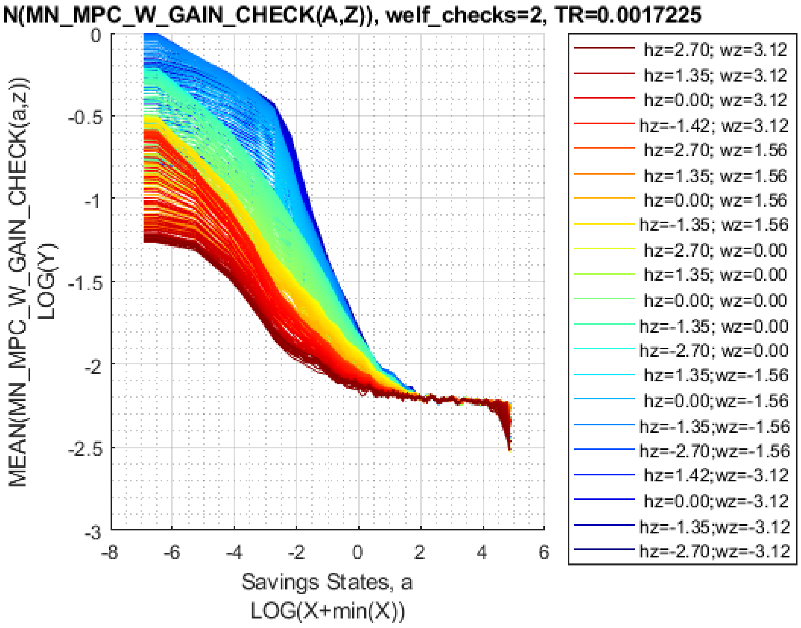




Graph Mean Consumption (***MPC: Share of Check Consumed***):

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


MN_MPC_W_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225


N(MN_MPC_W_GAIN_CHECK(A,Z)), welf_checks=2, TR=0.0017225

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(
tb_az_v = ff_summ_nd_array(st_title, mn_V_W_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc
```

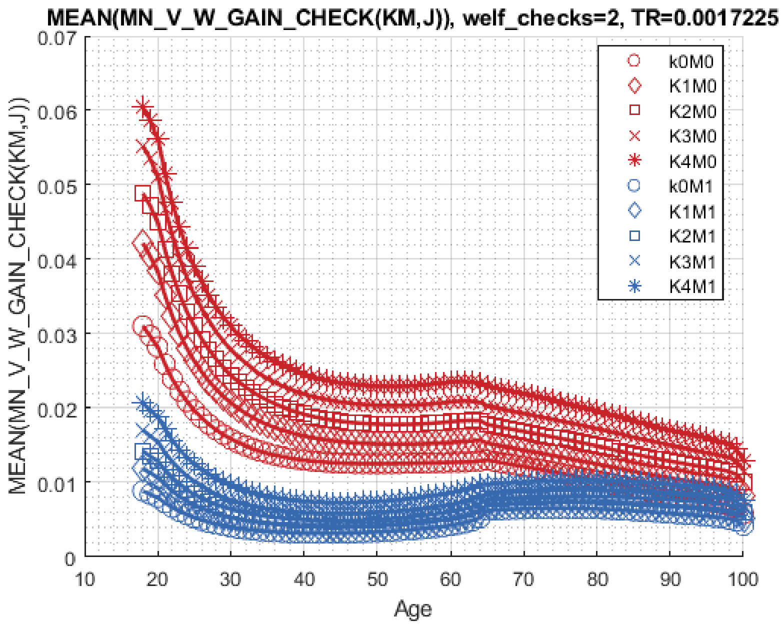
```
xxx  MEAN(MN_V_W_GAIN_CHECK(KM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    kids    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22    mean_age_23
    _____    ____    _____    ___________    ___________    ___________    ___________    ___________    ___________

      1       1        0        0.031034       0.029745       0.028239       0.025902        0.02395       0.022308
      2       2        0        0.042182       0.040477         0.0384       0.035121       0.032371       0.030047
      3       3        0        0.048861       0.047196       0.045053       0.041229       0.038024       0.035318
      4       4        0        0.055272       0.053518       0.051182       0.046853       0.043227       0.040165
      5       5        0        0.060434         0.0587       0.056288       0.051575       0.047632       0.044306
      6       1        1       0.0088974      0.0084565      0.0080356       0.007287      0.0066592      0.0061271
      7       2        1        0.011887       0.011299       0.010733      0.0097237      0.0088734      0.0081583
      8       3        1        0.014254       0.013578        0.01292       0.011706       0.010686      0.0098251
      9       4        1        0.017048       0.016271       0.015496       0.014054       0.012839        0.01181
     10       5        1        0.020638       0.019777       0.018893       0.017162       0.015705       0.014472
```

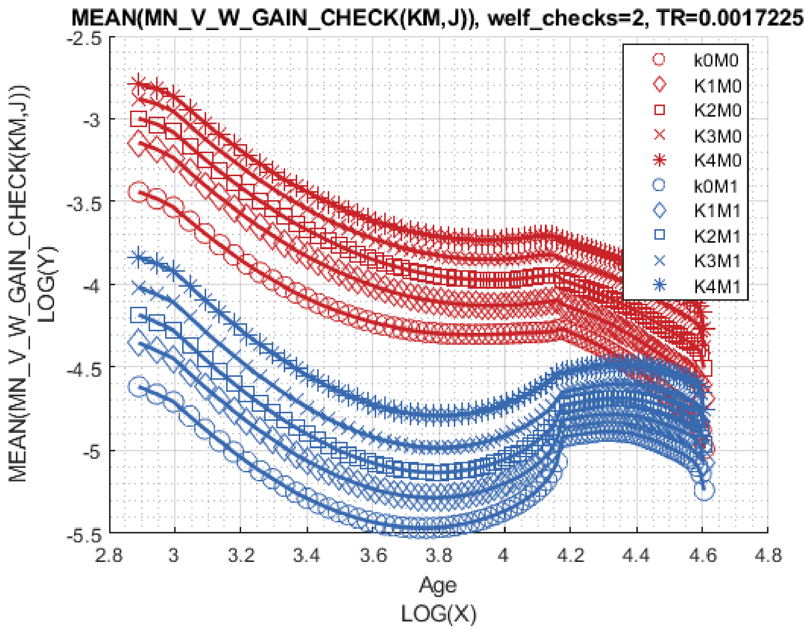
```
% Consumption Function
st_title = ['MEAN(MN_MPC_W_GAIN_CHECK(KM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2st
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_W_gain_share_check, true, ["mean"], 3, 1, cl_mp_dat
```

```
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_22    mean_age_23
    _____    ____    _____    ___________    ___________    ___________    ___________    ___________    ___________

      1       1        0       0.050486       0.055015       0.066697       0.065256       0.063327        0.06259
      2       2        0       0.057524       0.062751       0.075955       0.074114       0.073156       0.071278
      3       3        0       0.065349        0.07164        0.08726       0.084785       0.083817       0.080976
      4       4        0       0.069224       0.077159       0.092501       0.089833       0.087737       0.085697
      5       5        0        0.07472       0.082465       0.097494       0.094842       0.091941       0.089663
      6       1        1        0.08249       0.085999       0.088435       0.087541       0.086694       0.086355
      7       2        1       0.084681       0.087712        0.09101       0.089941       0.088666       0.087902
      8       3        1        0.08783       0.092431        0.09692       0.095481       0.094512       0.094028
      9       4        1       0.090426       0.093954       0.098752       0.097968       0.096993       0.095558
     10       5        1       0.095109       0.099933         0.1075        0.10332        0.10194        0.10128
```

Graph Mean Values:

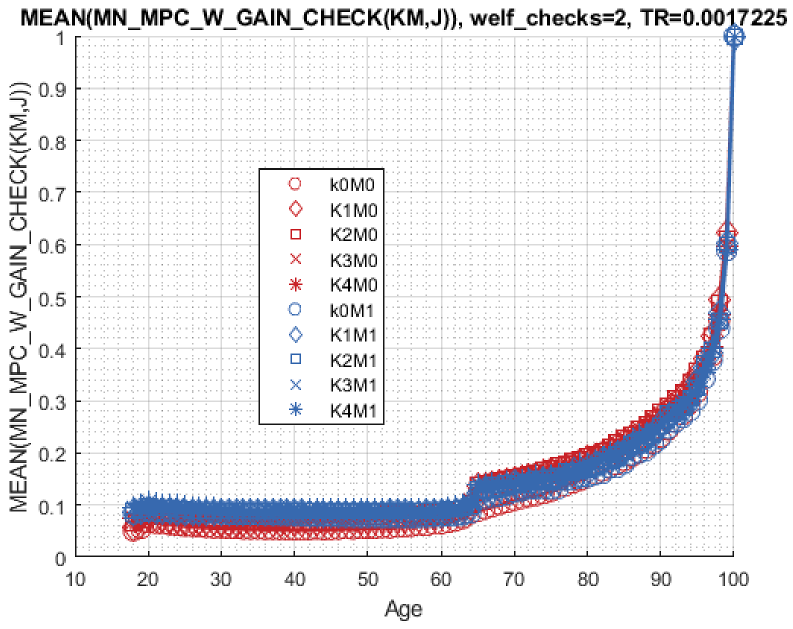
```
st_title = ['MEAN(MN\_V\_W\_GAIN\_CHECK(KM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_W\_GAIN\_CHECK(KM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

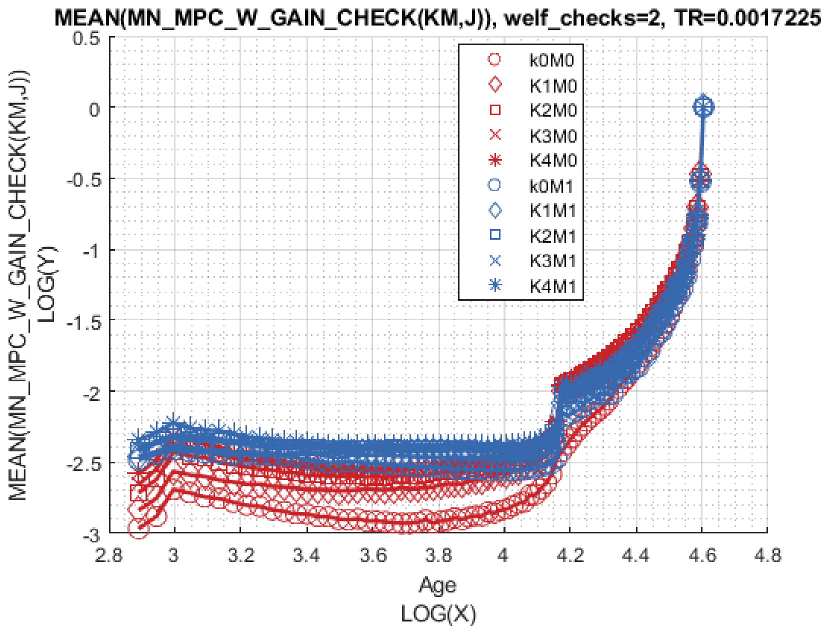




Graph Mean Consumption (***MPC: Share of Check Consumed***):

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





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(
tb_az_v = ff_summ_nd_array(st_title, mn_V_W_gain_check, true, ["mean"], 3, 1, cl_mp_datasetdesc
```

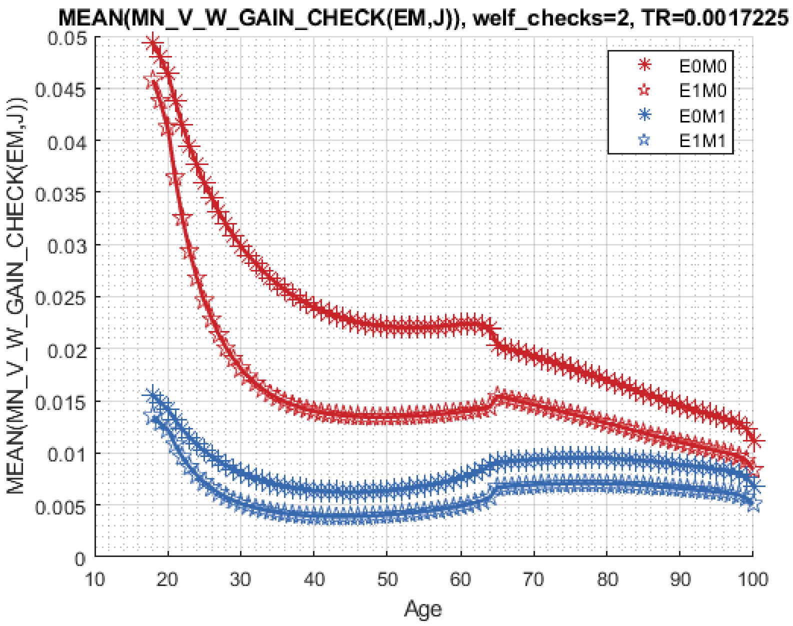
```
xxx  MEAN(MN_V_W_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225  xxxxxxxxxxxxxxxxxxxxxxxxxxx
    group    edu    marry    mean_age_18    mean_age_19    mean_age_20    mean_age_21    mean_age_22    mean_age_23
    _____    ___    _____    ___________    ___________    ___________    ___________  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

      1       0       0       0.049345       0.048033         0.0464        0.04381       0.041506       0.039451
      2       1       0       0.045768       0.043821       0.041264       0.036462       0.032576       0.029407
      3       0       1       0.015526       0.014854       0.014197       0.013177        0.01228       0.011498
      4       1       1       0.013563       0.012899       0.012234       0.010796      0.0096246      0.0086591
```

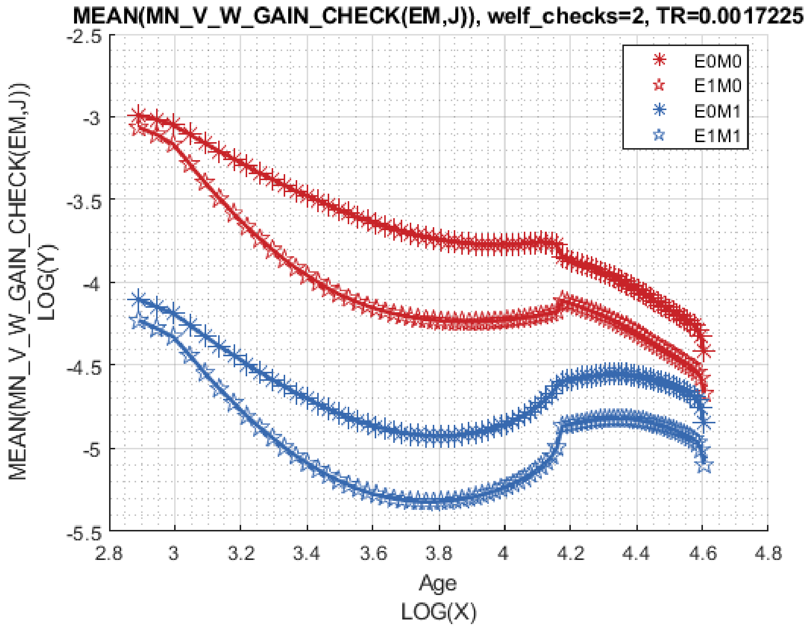
```
% Consumption
st_title = ['MEAN(MN_MPC_W_GAIN_CHECK(EM,J)), welf_checks=' num2str(welf_checks) ', TR=' num2st
tb_az_c = ff_summ_nd_array(st_title, mn_MPC_W_gain_share_check, true, ["mean"], 3, 1, cl_mp_dat
```

```
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    mean_age_23
    _____    ___    _____    ___________    ___________    ___________    ___________    ___________    ___________

      1       0       0       0.056783       0.060466       0.069707       0.069219       0.069323       0.068475
      2       1       0       0.070138       0.079146       0.098256       0.094312       0.090668       0.087607
      3       0       1        0.08145       0.084246       0.086686       0.086881       0.086662       0.086707
      4       1       1       0.094765       0.099766        0.10636        0.10282        0.10086       0.099342
```

Graph Mean Values:

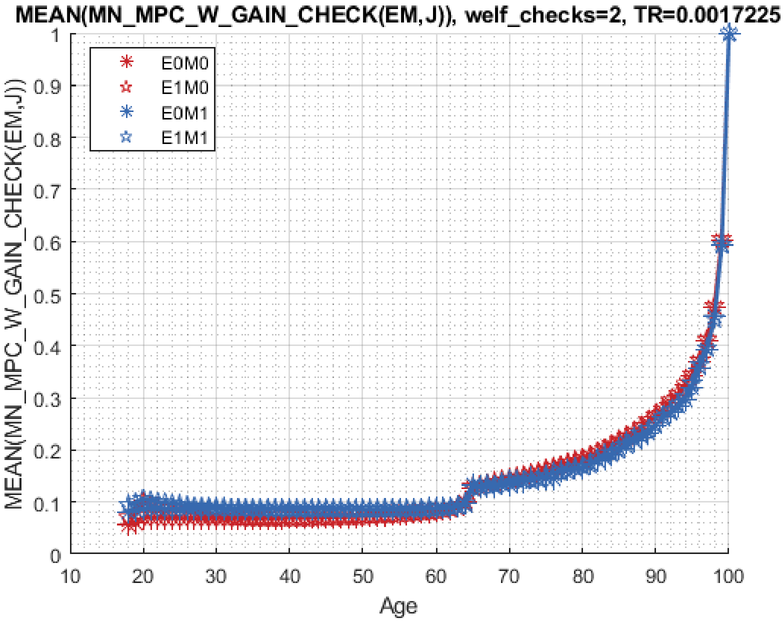
```
st_title = ['MEAN(MN\_V\_W\_GAIN\_CHECK(EM,J)), welf\_checks=' num2str(welf_checks) ', TR=' num
mp_support_graph('cl_st_graph_title') = {st_title};
mp_support_graph('cl_st_ytitle') = {'MEAN(MN\_V\_W\_GAIN\_CHECK(EM,J))'};
ff_graph_grid((tb_az_v{1:end, 4:end}), ar_row_grid, age_grid, mp_support_graph);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

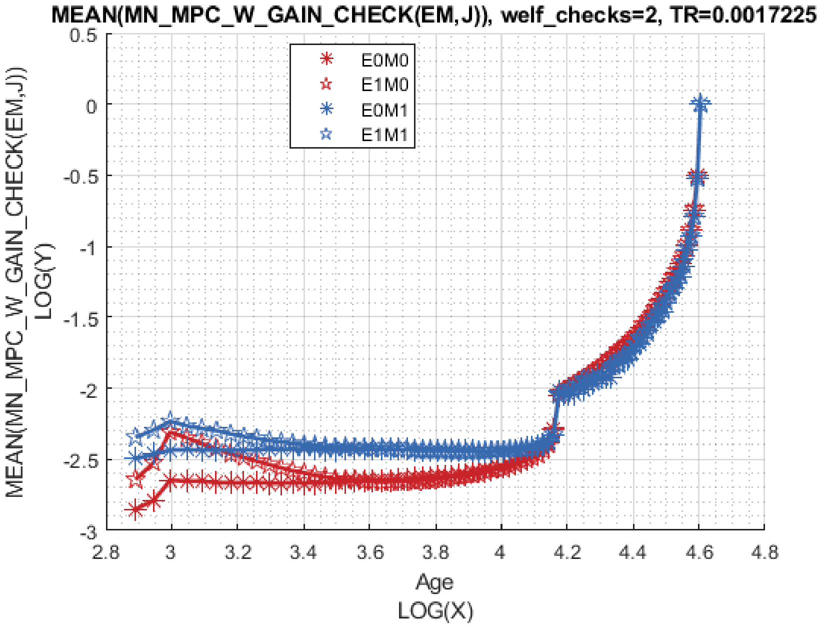




Graph Mean Consumption (***MPC: Share of Check Consumed***):

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


MEAN(MN_MPC_W_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225
MEAN(MN_MPC_W_GAIN_CHECK(EM,J))
Age
E0M0
E1M0
E0M1
E1M1
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_W_GAIN_CHECK(EM,J)), welf_checks=2, TR=0.0017225
MEAN(MN_MPC_W_GAIN_CHECK(EM,J))
LOG(Y)
Age
LOG(X)
E0M0
E1M0
E0M1
E1M1
-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
4.8