Results

August 30, 2018

1 Obtaining Estimation and Simulation Results

```
In [1]: import os
        from matplotlib import pyplot as plt
        from tools import models
```

We now create a base instance of the model (constant discount rates). This is useful for starting other specifications

```
In [2]: sp = models.sp
        base = sp('ref')
        base.chgoption('iload','T')
```

```
Default options:
isurvival  =  T
iload  =  F
iestimate  =  T
data  =  hrs_final_ref.csv
info  =  info_ref.dat
includeratings  =  F
icomplement  =  T
ihetero  =  T
icorr  =  T
iunitary  =  F
idiscount  =  F
idirect  =  F
arf  =  0.06
drc  =  0.08
reprate  =  0.6
ishufheter  =  F
ishufwages  =  F
iblockcomp  =  F
changed item  iload  with value F  to  T  ...
```

We estimate parameters and simulate data post estimation

```
In [3]: base.estimate()
        base.loglike
```

```
Out[3]: -2846.719602855049

In [4]: base.params

Out[4]:                                          par                      se
        alpha_c                   0.66633373417180419  5.6122670691351358E-002
        alpha_lm_cons             -3.0011251508001253      0.90449244271695561
        alpha_lm_age              0.69651389820926191      0.11056314226474352
        alpha_lm_hlim              2.9463469185950322       1.0225535662574339
        alpha_lm_college          0.35886169711893634      0.55527179177383701
        alpha_lm_male             0.44031879426277948      0.68699947514119597
        alpha_lm_job              0.20787813231521524      0.68948447312626215
        alpha_lf                   0.0000000000000000       0.0000000000000000
        alpha_lm_lf               0.30981760501449634  3.8083770834153111E-002
        beta_c                    0.42943333470586809  4.2108751479773922E-002
        beta_lf_cons              -1.0268780702287061      0.69129338175150190
        beta_lf_age               0.90656799405118504      0.12554348079751254
        beta_lf_hlim               2.5284402973200382      0.94499320947253129
        beta_lf_college          -0.44275407423888347      0.46670677513903536
        beta_lf_male               1.7532283661526034      0.64109115481058654
        beta_lf_job              -0.61245625259324421      0.51096473475580906
        beta_lm                    0.0000000000000000       0.0000000000000000
        beta_lm_lf                0.16779176648036098  3.3464716365136843E-002
        mu                    6.2343344910625103E-003      0.32768342063301836
        wageratio                 0.52072685658633933      0.29421109815920543
        log_rho_m            -5.1293294387550578E-002       0.0000000000000000
        log_rho_f            -5.1293294387550578E-002       0.0000000000000000
        tau_m_0                   -3.0000000000000000       0.0000000000000000
        log_tau_m                  1.0000000000000000       0.0000000000000000
        log_tau_m                  1.0000000000000000       0.0000000000000000
        log_tau_m                  1.0000000000000000       0.0000000000000000
        log_tau_m                  1.0000000000000000       0.0000000000000000
        log_tau_m                  1.0000000000000000       0.0000000000000000
        log_tau_m                  1.0000000000000000       0.0000000000000000
        log_tau_m                  1.0000000000000000       0.0000000000000000
        log_tau_m                  1.0000000000000000       0.0000000000000000
        tau_f_0                   -3.0000000000000000       0.0000000000000000
        log_tau_f                  1.0000000000000000       0.0000000000000000
        log_tau_f                  1.0000000000000000       0.0000000000000000
        log_tau_f                  1.0000000000000000       0.0000000000000000
        log_tau_f                  1.0000000000000000       0.0000000000000000
        log_tau_f                  1.0000000000000000       0.0000000000000000
        log_tau_f                  1.0000000000000000       0.0000000000000000
        log_tau_f                  1.0000000000000000       0.0000000000000000
        log_tau_f                  1.0000000000000000       0.0000000000000000
        log_sig_m                  0.0000000000000000       0.0000000000000000
        log_sig_f                  0.0000000000000000       0.0000000000000000
        L_nm_nm                    5.1786890765854103      0.45632695764220993
```

```
L_nf_nm                3.7585532220113662      0.34959386488950961
L_nf_nf                0.18930154935525897     0.80190150709707797
scale_m                1.0000000000000000      0.0000000000000000
scale_f                1.0000000000000000      0.0000000000000000
scale_h                1.0000000000000000      0.0000000000000000
```

Here are some stats on the simulations:

In [5]: `base.sim.describe()`

Out[5]:

```
           hhidpn  insim      jprob  rexpret_sim  sexpret_sim  leisure_m  \
count  464.000000  464.0  464.000000  464.000000  464.000000  464.000000
mean   303.915948    1.0    0.084052   65.974138   63.661638    0.094745
std    174.229335    0.0    0.277765    4.989771    3.134796    4.308279
min      1.000000    1.0    0.000000   54.000000   54.000000   -9.093125
25%    155.750000    1.0    0.000000   62.000000   62.000000   -3.401245
50%    300.500000    1.0    0.000000   67.000000   64.000000   -0.072925
75%    454.250000    1.0    0.000000   70.000000   66.000000    3.325566
max    604.000000    1.0    1.000000   80.000000   78.000000   15.364392

        leisure_f   rage    rcollege      hwho_m     ...        rhlth62  \
count  464.000000  464.0  464.000000  464.000000     ...     464.000000
mean     0.070270    0.0    0.441810    0.538793     ...       0.400431
std      3.133871    0.0    0.497139    0.499031     ...       0.277485
min     -6.581606    0.0    0.000000    0.000000     ...       0.000000
25%     -2.461458    0.0    0.000000    0.000000     ...       0.200000
50%     -0.024315    0.0    0.000000    1.000000     ...       0.400000
75%      2.391943    0.0    1.000000    1.000000     ...       0.500000
max     11.174369    0.0    1.000000    1.000000     ...       1.000000

          shlth62     rliv75r     sliv75r       rwage       swage      rhours  \
count  464.000000  464.000000  464.000000  464.000000  464.000000  464.000000
mean     0.395948    0.962090    0.912592   32.615540   25.620789   42.728448
std      0.265845    0.322436    0.245628   35.582619   60.577724   10.666330
min      0.000000    0.137999    0.062038    0.000000    0.000000    2.000000
25%      0.200000    0.720568    0.744456   16.347501   12.967500   40.000000
50%      0.400000    1.012476    0.968663   25.092495   18.000000   40.000000
75%      0.500000    1.175370    1.065559   36.758401   27.415630   50.000000
max      1.000000    1.491040    1.291550  365.000000  916.666687   80.000000

           shours      rexpret      sexpret
count  464.000000   200.000000   257.000000
mean    35.831897  2019.150000  2019.669261
std     12.125414     5.325345     5.157249
min      1.000000  2010.000000  2010.000000
25%     30.000000  2015.000000  2016.000000
50%     40.000000  2019.000000  2019.000000
75%     40.000000  2022.000000  2023.000000
```

```
max     75.000000  2040.000000  2034.000000

[8 rows x 147 columns]
```

We will now create a model that estimates discount rates

```
In [6]: disc = sp('discount')
```

```
Default options:
isurvival  =  T
iload  =  F
iestimate  =  T
data  =  hrs_final_ref.csv
info  =  info_ref.dat
includeratings  =  F
icomplement  =  T
ihetero  =  T
icorr  =  T
iunitary  =  F
idiscount  =  F
idirect  =  F
arf  =  0.06
drc  =  0.08
reprate  =  0.6
ishufheter  =  F
ishufwages  =  F
iblockcomp  =  F
```

We need to flag that we want to estimate discount rates

```
In [7]: disc.chgoption('idiscount','T')
        disc.chgoption('iload','T')

changed item  idiscount  with value F  to  T  ...
changed item  iload  with value F  to  T  ...
```

We now estimate parameters

```
In [8]: disc.estimate()
        disc.loglike

Out[8]: -2818.3448708879732

In [9]: disc.params
```

Out[9]:

	par	se
alpha_c	0.15381251390660106	3.6869215116319499E-002
alpha_lm_cons	-1.6856245225335307	0.47626887892585534
alpha_lm_age	0.43392941685436398	6.4713050539400324E-002
alpha_lm_hlim	1.6092484441292974	0.58108601683515682
alpha_lm_college	0.16824997567175534	0.29412243048848907
alpha_lm_male	0.18781045293783144	0.35008985490206834
alpha_lm_job	9.4708005159917244E-002	0.34833985097243253
alpha_lf	0.0000000000000000	0.0000000000000000
alpha_lm_lf	0.13429416621238721	2.2565647993691720E-002
beta_c	0.11485127235506647	3.6719465085222336E-002
beta_lf_cons	-0.73837181779287142	0.35302862062213530
beta_lf_age	0.47235191825346212	8.4960703057416681E-002
beta_lf_hlim	1.4110382629231570	0.50161809249363210
beta_lf_college	-0.19834138811438137	0.23843295986957000
beta_lf_male	0.98612606625759591	0.35689987927312755
beta_lf_job	-0.20854353827235886	0.25751032640971322
beta_lm	0.0000000000000000	0.0000000000000000
beta_lm_lf	8.0452567730378247E-002	2.1847676823783656E-002
mu	1.4218698111350370E-002	0.33004677361013668
wageratio	0.51733841891843568	0.30799050211924672
log_rho_m	2.9898035462288353E-002	9.8538846325579230E-003
log_rho_f	1.8250032086581467E-002	1.2313067799274637E-002
tau_m_0	-3.0000000000000000	0.0000000000000000
log_tau_m	1.0000000000000000	0.0000000000000000
log_tau_m	1.0000000000000000	0.0000000000000000
log_tau_m	1.0000000000000000	0.0000000000000000
log_tau_m	1.0000000000000000	0.0000000000000000
log_tau_m	1.0000000000000000	0.0000000000000000
log_tau_m	1.0000000000000000	0.0000000000000000
log_tau_m	1.0000000000000000	0.0000000000000000
log_tau_m	1.0000000000000000	0.0000000000000000
tau_f_0	-3.0000000000000000	0.0000000000000000
log_tau_f	1.0000000000000000	0.0000000000000000
log_tau_f	1.0000000000000000	0.0000000000000000
log_tau_f	1.0000000000000000	0.0000000000000000
log_tau_f	1.0000000000000000	0.0000000000000000
log_tau_f	1.0000000000000000	0.0000000000000000
log_tau_f	1.0000000000000000	0.0000000000000000
log_tau_f	1.0000000000000000	0.0000000000000000
log_tau_f	1.0000000000000000	0.0000000000000000
log_sig_m	0.0000000000000000	0.0000000000000000
log_sig_f	0.0000000000000000	0.0000000000000000
L_nm_nm	2.8009577500976697	0.33090153486995239
L_nf_nm	1.9209082908103476	0.29135905373494486
L_nf_nf	0.19384219823655499	0.38542195318519701
scale_m	1.0000000000000000	0.0000000000000000
scale_f	1.0000000000000000	0.0000000000000000

```
scale_h                 1.0000000000000000       0.0000000000000000
```

Some stats again on types

In [10]: disc.sim.describe()

```
Out[10]:           hhidpn  insim       jprob  rexpret_sim  sexpret_sim  leisure_m  \
         count  464.000000  464.0  464.000000   464.000000   464.000000 464.000000
         mean   303.915948    1.0    0.101293    65.728448    63.829741   0.002458
         std    174.229335    0.0    0.302042     4.670523     3.130547   2.323502
         min      1.000000    1.0    0.000000    55.000000    54.000000  -4.755283
         25%    155.750000    1.0    0.000000    62.000000    62.000000  -1.981291
         50%    300.500000    1.0    0.000000    65.000000    64.000000  -0.072990
         75%    454.250000    1.0    0.000000    69.000000    66.000000   1.905043
         max    604.000000    1.0    1.000000    80.000000    80.000000   7.262594

                 leisure_f   rage    rcollege      hwho_m     ...          rhlth62  \
         count  464.000000  464.0  464.000000  464.000000     ...       464.000000
         mean     0.003629    0.0    0.441810    0.538793     ...         0.400431
         std      1.603933    0.0    0.497139    0.499031     ...         0.277485
         min     -3.287411    0.0    0.000000    0.000000     ...         0.000000
         25%     -1.329474    0.0    0.000000    0.000000     ...         0.200000
         50%     -0.036837    0.0    0.000000    1.000000     ...         0.400000
         75%      1.306682    0.0    1.000000    1.000000     ...         0.500000
         max      5.025972    0.0    1.000000    1.000000     ...         1.000000

                   shlth62     rliv75r     sliv75r       rwage       swage      rhours  \
         count  464.000000  464.000000  464.000000  464.000000  464.000000  464.000000
         mean     0.395948    0.962090    0.912592   32.615540   25.620789   42.728448
         std      0.265845    0.322436    0.245628   35.582619   60.577724   10.666330
         min      0.000000    0.137999    0.062038    0.000000    0.000000    2.000000
         25%      0.200000    0.720568    0.744456   16.347501   12.967500   40.000000
         50%      0.400000    1.012476    0.968663   25.092495   18.000000   40.000000
         75%      0.500000    1.175370    1.065559   36.758401   27.415630   50.000000
         max      1.000000    1.491040    1.291550  365.000000  916.666687   80.000000

                    shours      rexpret      sexpret
         count  464.000000   200.000000   257.000000
         mean    35.831897  2019.150000  2019.669261
         std     12.125414     5.325345     5.157249
         min      1.000000  2010.000000  2010.000000
         25%     30.000000  2015.000000  2016.000000
         50%     40.000000  2019.000000  2019.000000
         75%     40.000000  2022.000000  2023.000000
         max     75.000000  2040.000000  2034.000000

         [8 rows x 147 columns]
```

1.1 Robustness of Specification

We will collect a few loglikelihoods to do LR tests (Table in main text). We will also present some results in the appendix for these alternative models.

1.1.1 Model with uncorrelated types

```
In [11]: uncor = sp('nocorr')
```

```
Default options:
isurvival  =  T
iload  =  F
iestimate  =  T
data  =  hrs_final_ref.csv
info  =  info_ref.dat
includeratings  =  F
icomplement  =  T
ihetero  =  T
icorr  =  T
iunitary  =  F
idiscount  =  F
idirect  =  F
arf  =  0.06
drc  =  0.08
reprate  =  0.6
ishufheter  =  F
ishufwages  =  F
iblockcomp  =  F
```

```
In [12]: uncor.chgoption('icorr','F')
         uncor.chgoption('idiscount','T')
         uncor.chgoption('iload','T')
```

```
changed item  icorr  with value T  to  F  ...
changed item  idiscount  with value F  to  T  ...
changed item  iload  with value F  to  T  ...
```

```
In [13]: uncor.estimate()
         uncor.loglike
```

```
Out[13]: -2899.3527710927633
```

```
In [14]: uncor.params
```

```
Out[14]:
                                          par                       se
         alpha_c              0.12731889866088533  3.0793642209440010E-002
         alpha_lm_cons       -1.5340243266839209       0.47945605981721146
         alpha_lm_age         0.39959661401166141  5.8654178709795030E-002
```

```
alpha_lm_hlim          1.4796758136882329      0.60728068532669932
alpha_lm_college       0.21884688256894691     0.29863194186991587
alpha_lm_male          0.22277046305399947     0.35158817513263235
alpha_lm_job           0.12915273291347937     0.34171748043423017
alpha_lf               0.0000000000000000      0.0000000000000000
alpha_lm_lf            0.11893090796036433   2.1113741139981653E-002
beta_c               9.4144062988514118E-002 3.1454041356208925E-002
beta_lf_cons          -0.84182049609971144     0.37781248924237981
beta_lf_age            0.44524137818253967   8.1350730522594977E-002
beta_lf_hlim           1.2684062614991500      0.49471347401235893
beta_lf_college     -7.5722918536815828E-002   0.22820658644093911
beta_lf_male           0.80703656001555546     0.33902077474246961
beta_lf_job           -0.22046245986825083     0.26158173418095548
beta_lm                0.0000000000000000      0.0000000000000000
beta_lm_lf           7.5134992166190268E-002 2.0575888540385206E-002
mu                   9.1925408253605317E-002   0.33514908430768803
wageratio              0.39074813730495450     0.29067227658565292
log_rho_m            3.7794156247160038E-002 9.3437135167809913E-003
log_rho_f            2.5829822321673510E-002 1.2149064440757213E-002
tau_m_0               -3.0000000000000000      0.0000000000000000
log_tau_m              1.0000000000000000      0.0000000000000000
log_tau_m              1.0000000000000000      0.0000000000000000
log_tau_m              1.0000000000000000      0.0000000000000000
log_tau_m              1.0000000000000000      0.0000000000000000
log_tau_m              1.0000000000000000      0.0000000000000000
log_tau_m              1.0000000000000000      0.0000000000000000
log_tau_m              1.0000000000000000      0.0000000000000000
log_tau_m              1.0000000000000000      0.0000000000000000
tau_f_0               -3.0000000000000000      0.0000000000000000
log_tau_f              1.0000000000000000      0.0000000000000000
log_tau_f              1.0000000000000000      0.0000000000000000
log_tau_f              1.0000000000000000      0.0000000000000000
log_tau_f              1.0000000000000000      0.0000000000000000
log_tau_f              1.0000000000000000      0.0000000000000000
log_tau_f              1.0000000000000000      0.0000000000000000
log_tau_f              1.0000000000000000      0.0000000000000000
log_tau_f              1.0000000000000000      0.0000000000000000
log_sig_m              0.0000000000000000      0.0000000000000000
log_sig_f              0.0000000000000000      0.0000000000000000
L_nm_nm                2.6889485082915456      0.32260868339714660
L_nf_nm                0.0000000000000000      0.0000000000000000
L_nf_nf                1.7604243970895372      0.27809241139621660
scale_m                1.0000000000000000      0.0000000000000000
scale_f                1.0000000000000000      0.0000000000000000
scale_h                1.0000000000000000      0.0000000000000000
```

In [15]: uncor.sim.describe()

```
Out[15]:          hhidpn  insim       jprob  rexpret_sim  sexpret_sim   leisure_m  \
```

```
count  464.000000  464.0  464.000000  464.000000  464.000000  464.000000
mean   303.915948    1.0    0.088362   65.646552   63.909483   -0.079614
std    174.229335    0.0    0.284127    4.561089    2.697971    2.043431
min      1.000000    1.0    0.000000   54.000000   58.000000   -4.379481
25%    155.750000    1.0    0.000000   62.000000   62.000000   -1.756941
50%    300.500000    1.0    0.000000   65.000000   64.000000   -0.070071
75%    454.250000    1.0    0.000000   70.000000   65.000000    1.500670
max    604.000000    1.0    1.000000   80.000000   80.000000    5.754107

        leisure_f   rage    rcollege     hwho_m       ...          rhlth62  \
count  464.000000  464.0  464.000000  464.000000      ...       464.000000
mean     0.060195    0.0    0.441810    0.538793      ...         0.400431
std      1.103602    0.0    0.497139    0.499031      ...         0.277485
min     -2.488377    0.0    0.000000    0.000000      ...         0.000000
25%     -0.687988    0.0    0.000000    0.000000      ...         0.200000
50%      0.045366    0.0    0.000000    1.000000      ...         0.400000
75%      0.794043    0.0    1.000000    1.000000      ...         0.500000
max      2.775941    0.0    1.000000    1.000000      ...         1.000000

          shlth62     rliv75r     sliv75r       rwage       swage      rhours  \
count  464.000000  464.000000  464.000000  464.000000  464.000000  464.000000
mean     0.395948    0.962090    0.912592   32.615540   25.620789   42.728448
std      0.265845    0.322436    0.245628   35.582619   60.577724   10.666330
min      0.000000    0.137999    0.062038    0.000000    0.000000    2.000000
25%      0.200000    0.720568    0.744456   16.347501   12.967500   40.000000
50%      0.400000    1.012476    0.968663   25.092495   18.000000   40.000000
75%      0.500000    1.175370    1.065559   36.758401   27.415630   50.000000
max      1.000000    1.491040    1.291550  365.000000  916.666687   80.000000

           shours      rexpret      sexpret
count  464.000000   200.000000   257.000000
mean    35.831897  2019.150000  2019.669261
std     12.125414     5.325345     5.157249
min      1.000000  2010.000000  2010.000000
25%     30.000000  2015.000000  2016.000000
50%     40.000000  2019.000000  2019.000000
75%     40.000000  2022.000000  2023.000000
max     75.000000  2040.000000  2034.000000

[8 rows x 147 columns]
```

1.1.2 Model without Survival Risk

```
In [16]: nosurv = sp('nosurv')
```

```
Default options:
isurvival  =  T
iload  =  F
```

```
iestimate  =  T
data  =  hrs_final_ref.csv
info  =  info_ref.dat
includeratings  =  F
icomplement  =  T
ihetero  =  T
icorr  =  T
iunitary  =  F
idiscount  =  F
idirect  =  F
arf  =  0.06
drc  =  0.08
reprate  =  0.6
ishufheter  =  F
ishufwages  =  F
iblockcomp  =  F
```

In [17]:
```
nosurv.chgoption('isurvival','F')
nosurv.chgoption('idiscount','T')
nosurv.chgoption('iload','T')
```

```
changed item  isurvival  with value T  to  F  ...
changed item  idiscount  with value F  to  T  ...
changed item  iload  with value F  to  T  ...
```

In [18]:
```
nosurv.estimate()
nosurv.loglike
```

Out[18]: -2818.120226635457

In [19]:
```
nosurv.params
```

Out[19]:

	par	se
alpha_c	5.7937437701925977E-002	1.8685959777645762E-002
alpha_lm_cons	-1.5569891225270480	0.44541774572065673
alpha_lm_age	0.39753938236519670	6.3788681427956087E-002
alpha_lm_hlim	1.3336471853986134	0.52879474065546739
alpha_lm_college	0.18752700990462692	0.26597546774532815
alpha_lm_male	0.14628562356290978	0.30977343355771275
alpha_lm_job	0.12593483425053939	0.31756201519433302
alpha_lf	0.0000000000000000	0.0000000000000000
alpha_lm_lf	0.10550926148941667	1.9630021307509786E-002
beta_c	6.7583277293491339E-002	2.5578055813741288E-002
beta_lf_cons	-0.68082625194673874	0.33382649050123009
beta_lf_age	0.44117277786824077	8.1720893523168789E-002
beta_lf_hlim	1.2820647657970310	0.47500928823973809
beta_lf_college	-0.15853861591137058	0.22226425154498003

```
beta_lf_male          0.90430251699161335      0.32951909189810158
beta_lf_job          -0.19157453827138776      0.24123765560475949
beta_lm               0.0000000000000000       0.0000000000000000
beta_lm_lf          7.3899559015865909E-002  1.9620054787783094E-002
mu                  1.3585435371352695E-002   0.32614823410856181
wageratio             0.45996365684047502      0.29604261692535994
log_rho_m           3.0342494520804571E-002  1.0385765026975080E-002
log_rho_f           1.6703383253027140E-002  1.2436078705210020E-002
tau_m_0              -3.0000000000000000       0.0000000000000000
log_tau_m             1.0000000000000000       0.0000000000000000
log_tau_m             1.0000000000000000       0.0000000000000000
log_tau_m             1.0000000000000000       0.0000000000000000
log_tau_m             1.0000000000000000       0.0000000000000000
log_tau_m             1.0000000000000000       0.0000000000000000
log_tau_m             1.0000000000000000       0.0000000000000000
log_tau_m             1.0000000000000000       0.0000000000000000
log_tau_m             1.0000000000000000       0.0000000000000000
tau_f_0              -3.0000000000000000       0.0000000000000000
log_tau_f             1.0000000000000000       0.0000000000000000
log_tau_f             1.0000000000000000       0.0000000000000000
log_tau_f             1.0000000000000000       0.0000000000000000
log_tau_f             1.0000000000000000       0.0000000000000000
log_tau_f             1.0000000000000000       0.0000000000000000
log_tau_f             1.0000000000000000       0.0000000000000000
log_tau_f             1.0000000000000000       0.0000000000000000
log_tau_f             1.0000000000000000       0.0000000000000000
log_sig_m             0.0000000000000000       0.0000000000000000
log_sig_f             0.0000000000000000       0.0000000000000000
L_nm_nm               2.5521059687009942       0.31077235161910743
L_nf_nm               1.8028811124829329       0.27673469133903150
L_nf_nf               0.18438574202742528      0.35149314714567520
scale_m               1.0000000000000000       0.0000000000000000
scale_f               1.0000000000000000       0.0000000000000000
scale_h               1.0000000000000000       0.0000000000000000
```

In [20]:
```
nosurv.sim.describe()
```

Out[20]:
```
           hhidpn  insim       jprob  rexpret_sim  sexpret_sim  leisure_m  \
count  464.000000  464.0  464.000000   464.000000   464.000000 464.000000
mean   303.915948    1.0    0.135776    65.659483    63.698276  -0.000398
std    174.229335    0.0    0.342920     4.757215     3.162195   2.121325
min      1.000000    1.0    0.000000    54.000000    55.000000  -4.258304
25%    155.750000    1.0    0.000000    62.000000    62.000000  -1.785382
50%    300.500000    1.0    0.000000    65.000000    63.500000  -0.073973
75%    454.250000    1.0    0.000000    70.000000    66.000000   1.709255
max    604.000000    1.0    1.000000    80.000000    80.000000   7.040929

        leisure_f   rage    rcollege     hwho_m      ...         rhlth62  \
```

```
count  464.000000  464.0  464.000000  464.000000     ...       464.000000
mean     0.001453    0.0    0.441810    0.538793     ...         0.400431
std      1.507879    0.0    0.497139    0.499031     ...         0.277485
min     -3.059676    0.0    0.000000    0.000000     ...         0.000000
25%     -1.244012    0.0    0.000000    0.000000     ...         0.200000
50%     -0.073523    0.0    0.000000    1.000000     ...         0.400000
75%      1.216423    0.0    1.000000    1.000000     ...         0.500000
max      5.008634    0.0    1.000000    1.000000     ...         1.000000

          shlth62     rliv75r     sliv75r       rwage       swage      rhours  \
count  464.000000  464.000000  464.000000  464.000000  464.000000  464.000000
mean     0.395948    0.962090    0.912592   32.615540   25.620789   42.728448
std      0.265845    0.322436    0.245628   35.582619   60.577724   10.666330
min      0.000000    0.137999    0.062038    0.000000    0.000000    2.000000
25%      0.200000    0.720568    0.744456   16.347501   12.967500   40.000000
50%      0.400000    1.012476    0.968663   25.092495   18.000000   40.000000
75%      0.500000    1.175370    1.065559   36.758401   27.415630   50.000000
max      1.000000    1.491040    1.291550  365.000000  916.666687   80.000000

           shours      rexpret      sexpret
count  464.000000   200.000000   257.000000
mean    35.831897  2019.150000  2019.669261
std     12.125414     5.325345     5.157249
min      1.000000  2010.000000  2010.000000
25%     30.000000  2015.000000  2016.000000
50%     40.000000  2019.000000  2019.000000
75%     40.000000  2022.000000  2023.000000
max     75.000000  2040.000000  2034.000000

[8 rows x 147 columns]
```

1.1.3 Model without Complementarity

```
In [21]: nocomp = sp('nocomp')
```

```
Default options:
isurvival  =  T
iload  =  F
iestimate  =  T
data  =  hrs_final_ref.csv
info  =  info_ref.dat
includeratings  =  F
icomplement  =  T
ihetero  =  T
icorr  =  T
iunitary  =  F
idiscount  =  F
idirect  =  F
```

```
arf  =  0.06
drc  =  0.08
reprate  =  0.6
ishufheter  =  F
ishufwages  =  F
iblockcomp  =  F
```

In [22]:
```python
nocomp.chgoption('idiscount','T')
nocomp.chgoption('icomplement','F')
nocomp.chgoption('iload','T')
```

```
changed item  idiscount  with value F  to  T  ...
changed item  icomplement  with value T  to  F  ...
changed item  iload  with value F  to  T  ...
```

In [23]:
```python
nocomp.estimate()
```

In [24]:
```python
nocomp.loglike
```

Out[24]:
```
-2855.7661896271484
```

In [25]:
```python
nocomp.params
```

Out[25]:

	par	se
alpha_c	6.7684114800481102E-002	2.1125315408699219E-002
alpha_lm_cons	-1.2170811179200167	0.40079968589044540
alpha_lm_age	0.45481435809495274	6.7766162367306010E-002
alpha_lm_hlim	1.3604653358285910	0.52278416175670195
alpha_lm_college	9.3951859449528441E-002	0.25667368427848036
alpha_lm_male	0.54276941604783990	0.30397804670483097
alpha_lm_job	0.12313002674575364	0.30475058014360912
alpha_lf	0.0000000000000000	0.0000000000000000
alpha_lm_lf	0.0000000000000000	0.0000000000000000
beta_c	6.4138939825550037E-002	2.0747345557093864E-002
beta_lf_cons	-0.43329039766274519	0.29745350194114933
beta_lf_age	0.45705866120729566	8.0737929200789801E-002
beta_lf_hlim	1.2415022846946957	0.44742368823001244
beta_lf_college	-0.18990553568380528	0.21077746783058879
beta_lf_male	1.1698775542079083	0.32330953633726717
beta_lf_job	-0.15210009976387934	0.22640318484424771
beta_lm	0.0000000000000000	0.0000000000000000
beta_lm_lf	0.0000000000000000	0.0000000000000000
mu	-6.3526427342565167E-002	0.32045305829599025
wageratio	0.53306770597258557	0.29010031828782473
log_rho_m	4.4053589133654925E-002	1.1749315117099739E-002
log_rho_f	2.9382550901632078E-002	1.2279142500165945E-002
tau_m_0	-3.0000000000000000	0.0000000000000000

```
log_tau_m              1.0000000000000000        0.0000000000000000
log_tau_m              1.0000000000000000        0.0000000000000000
log_tau_m              1.0000000000000000        0.0000000000000000
log_tau_m              1.0000000000000000        0.0000000000000000
log_tau_m              1.0000000000000000        0.0000000000000000
log_tau_m              1.0000000000000000        0.0000000000000000
log_tau_m              1.0000000000000000        0.0000000000000000
log_tau_m              1.0000000000000000        0.0000000000000000
tau_f_0               -3.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_sig_m              0.0000000000000000        0.0000000000000000
log_sig_f              0.0000000000000000        0.0000000000000000
L_nm_nm                2.4620823263829235       0.33736996351434723
L_nf_nm                1.7112125845760651       0.26203155150501378
L_nf_nf               0.15818941692517388       0.33801943004323448
scale_m                1.0000000000000000        0.0000000000000000
scale_f                1.0000000000000000        0.0000000000000000
scale_h                1.0000000000000000        0.0000000000000000
```

In [26]:
```
nocomp.sim.describe()
```

Out[26]:
```
           hhidpn  insim       jprob  rexpret_sim  sexpret_sim  leisure_m  \
count  464.000000  464.0  464.000000   464.000000   464.000000 464.000000
mean   303.915948    1.0    0.116379    65.790948    64.478448  -0.006056
std    174.229335    0.0    0.321025     4.161617     2.939191   2.041977
min      1.000000    1.0    0.000000    57.000000    56.000000  -4.148323
25%    155.750000    1.0    0.000000    62.000000    62.000000  -1.690364
50%    300.500000    1.0    0.000000    66.000000    65.000000  -0.080175
75%    454.250000    1.0    0.000000    69.000000    67.000000   1.687099
max    604.000000    1.0    1.000000    80.000000    79.000000   5.436692

        leisure_f   rage    rcollege      hwho_m      ...        rhlth62  \
count  464.000000  464.0  464.000000  464.000000      ...     464.000000
mean    -0.002523    0.0    0.441810    0.538793      ...       0.400431
std      1.427030    0.0    0.497139    0.499031      ...       0.277485
min     -2.877898    0.0    0.000000    0.000000      ...       0.000000
25%     -1.162311    0.0    0.000000    0.000000      ...       0.200000
50%     -0.075648    0.0    0.000000    1.000000      ...       0.400000
75%      1.162369    0.0    1.000000    1.000000      ...       0.500000
max      3.831154    0.0    1.000000    1.000000      ...       1.000000
```

```
           shlth62     rliv75r     sliv75r       rwage       swage      rhours  \
count   464.000000  464.000000  464.000000  464.000000  464.000000  464.000000
mean      0.395948    0.962090    0.912592   32.615540   25.620789   42.728448
std       0.265845    0.322436    0.245628   35.582619   60.577724   10.666330
min       0.000000    0.137999    0.062038    0.000000    0.000000    2.000000
25%       0.200000    0.720568    0.744456   16.347501   12.967500   40.000000
50%       0.400000    1.012476    0.968663   25.092495   18.000000   40.000000
75%       0.500000    1.175370    1.065559   36.758401   27.415630   50.000000
max       1.000000    1.491040    1.291550  365.000000  916.666687   80.000000

            shours      rexpret      sexpret
count   464.000000   200.000000   257.000000
mean     35.831897  2019.150000  2019.669261
std      12.125414     5.325345     5.157249
min       1.000000  2010.000000  2010.000000
25%      30.000000  2015.000000  2016.000000
50%      40.000000  2019.000000  2019.000000
75%      40.000000  2022.000000  2023.000000
max      75.000000  2040.000000  2034.000000

[8 rows x 147 columns]
```

1.1.4 Unitary Model

```
In [27]: unitary = sp('unitary')
         unitary.chgoption('iunitary','T')
         unitary.chgoption('idiscount','T')
         unitary.chgoption('iload','T')
```

```
Default options:
isurvival  =  T
iload  =  F
iestimate  =  T
data  =  hrs_final_ref.csv
info  =  info_ref.dat
includeratings  =  F
icomplement  =  T
ihetero  =  T
icorr  =  T
iunitary  =  F
idiscount  =  F
idirect  =  F
arf  =  0.06
drc  =  0.08
reprate  =  0.6
ishufheter  =  F
ishufwages  =  F
iblockcomp  =  F
```

```
changed item  iunitary  with value F  to  T  ...
changed item  idiscount  with value F  to  T  ...
changed item  iload  with value F  to  T  ...
```

In [28]:
```
unitary.estimate()
unitary.loglike
```

Out[28]:
```
-2821.128273333861
```

In [29]:
```
unitary.params
```

Out[29]:
```
                                     par                       se
alpha_c               0.15182406589585989  3.6515442807415519E-002
alpha_lm_cons         -1.6332190463212242      0.47920257905411873
alpha_lm_age          0.43978975451722974  6.5674456817966145E-002
alpha_lm_hlim          1.6016682028999958      0.57991010763889972
alpha_lm_college      0.18890444304578757      0.29514226604767002
alpha_lm_male         0.17241043919956345      0.35310262979713164
alpha_lm_job      3.1270538303150297E-002      0.35129118557519690
alpha_lf               0.0000000000000000       0.0000000000000000
alpha_lm_lf           0.13156460331229886  2.2219316196333432E-002
beta_c                0.11644184827948612  3.7494000762092340E-002
beta_lf_cons         -0.76121050924083900      0.35431469034514063
beta_lf_age           0.47550737984740393  8.7280864075439293E-002
beta_lf_hlim           1.3998530431902385      0.50242822559667200
beta_lf_college      -0.18990389663978482      0.24037187392418863
beta_lf_male          0.95489005492934931      0.35745048652950767
beta_lf_job          -0.19990931749879925      0.25972725877600911
beta_lm                0.0000000000000000       0.0000000000000000
beta_lm_lf        8.3377570799921752E-002  2.2062729677311068E-002
mu                5.4465499281290214E-002      0.31442616883468111
wageratio              0.0000000000000000       0.0000000000000000
log_rho_m         3.0513754669652459E-002  9.7806943730015285E-003
log_rho_f         1.7548533291637688E-002  1.2426394396677639E-002
tau_m_0               -3.0000000000000000       0.0000000000000000
log_tau_m              1.0000000000000000       0.0000000000000000
log_tau_m              1.0000000000000000       0.0000000000000000
log_tau_m              1.0000000000000000       0.0000000000000000
log_tau_m              1.0000000000000000       0.0000000000000000
log_tau_m              1.0000000000000000       0.0000000000000000
log_tau_m              1.0000000000000000       0.0000000000000000
log_tau_m              1.0000000000000000       0.0000000000000000
log_tau_m              1.0000000000000000       0.0000000000000000
tau_f_0               -3.0000000000000000       0.0000000000000000
log_tau_f              1.0000000000000000       0.0000000000000000
log_tau_f              1.0000000000000000       0.0000000000000000
log_tau_f              1.0000000000000000       0.0000000000000000
log_tau_f              1.0000000000000000       0.0000000000000000
```

```
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_tau_f              1.0000000000000000        0.0000000000000000
log_sig_m              0.0000000000000000        0.0000000000000000
log_sig_f              0.0000000000000000        0.0000000000000000
L_nm_nm                2.8174463204995299       0.33258627578260586
L_nf_nm                1.9342076271723192       0.29597088083342304
L_nf_nf               0.19511085082811300       0.38799314915048322
scale_m                1.0000000000000000        0.0000000000000000
scale_f                1.0000000000000000        0.0000000000000000
scale_h                1.0000000000000000        0.0000000000000000
```

In [30]: unitary.sim.describe()

Out[30]:

```
           hhidpn  insim       jprob  rexpret_sim  sexpret_sim  leisure_m  \
count  464.000000  464.0  464.000000   464.000000   464.000000 464.000000
mean   303.915948    1.0    0.127155    65.635776    63.903017  -0.000685
std    174.229335    0.0    0.333506     4.632369     3.098329   2.335276
min      1.000000    1.0    0.000000    55.000000    55.000000  -4.758435
25%    155.750000    1.0    0.000000    62.000000    62.000000  -1.965214
50%    300.500000    1.0    0.000000    65.000000    64.000000  -0.073420
75%    454.250000    1.0    0.000000    69.000000    66.000000   1.901728
max    604.000000    1.0    1.000000    80.000000    78.000000   7.349665

        leisure_f   rage    rcollege      hwho_m      ...          rhlth62  \
count  464.000000  464.0  464.000000  464.000000      ...       464.000000
mean     0.001419    0.0    0.441810    0.538793      ...         0.400431
std      1.613392    0.0    0.497139    0.499031      ...         0.277485
min     -3.293268    0.0    0.000000    0.000000      ...         0.000000
25%     -1.347942    0.0    0.000000    0.000000      ...         0.200000
50%     -0.055019    0.0    0.000000    1.000000      ...         0.400000
75%      1.310731    0.0    1.000000    1.000000      ...         0.500000
max      5.090378    0.0    1.000000    1.000000      ...         1.000000

          shlth62     rliv75r     sliv75r       rwage       swage      rhours  \
count  464.000000  464.000000  464.000000  464.000000  464.000000  464.000000
mean     0.395948    0.962090    0.912592   32.615540   25.620789   42.728448
std      0.265845    0.322436    0.245628   35.582619   60.577724   10.666330
min      0.000000    0.137999    0.062038    0.000000    0.000000    2.000000
25%      0.200000    0.720568    0.744456   16.347501   12.967500   40.000000
50%      0.400000    1.012476    0.968663   25.092495   18.000000   40.000000
75%      0.500000    1.175370    1.065559   36.758401   27.415630   50.000000
max      1.000000    1.491040    1.291550  365.000000  916.666687   80.000000

           shours      rexpret      sexpret
count  464.000000   200.000000   257.000000
mean    35.831897  2019.150000  2019.669261
```

```
std     12.125414     5.325345     5.157249
min      1.000000  2010.000000  2010.000000
25%     30.000000  2015.000000  2016.000000
50%     40.000000  2019.000000  2019.000000
75%     40.000000  2022.000000  2023.000000
max     75.000000  2040.000000  2034.000000

[8 rows x 147 columns]
```

1.1.5 Table with LR Tests

In [31]:
```python
import pandas as pd
data = [disc.loglike,base.loglike,uncor.loglike,nocomp.loglike,unitary.loglike]
lr = [0]
for i in range(1,len(data)):
    lr.append(-2.0*(data[i]-data[0]))
names = ['Baseline','Fixed Discount Rates (2)','No Correlation UH (1)','No Complementa
table = pd.DataFrame(data=list(zip(data,lr)),index=names,columns=['Loglikehood Value'
def f(x):
    return '{:1.3f}'.format(x)
with open('../tex/tables/lrtests.tex','w') as tf:
    tf.write(table.to_latex(formatters=[f,f]))
table
```

Out[31]:

	Loglikehood Value	LR Statistic
Baseline	-2818.344871	0.000000
Fixed Discount Rates (2)	-2846.719603	56.749464
No Correlation UH (1)	-2899.352771	162.015800
No Complementarity (2)	-2855.766190	74.842637
Unitary (1)	-2821.128273	5.566805

Critical values at 5% for these tests:

In [32]:
```python
from scipy.stats import chi2
[chi2(1).ppf(0.95),chi2(2).ppf(0.95)]
```

Out[32]: [3.8414588206941236, 5.99146454710798]

1.2 Correlation with Expected Retirement in HRS

In [33]:
```python
disc.sim['rexpret'] = disc.sim['rage_mod'] + (disc.sim['rexpret']-2011)
disc.sim['sexpret'] = disc.sim['sage_mod'] + (disc.sim['sexpret']-2011)
```

In [34]:
```python
disc.sim[['rexpret','sexpret']].describe()
```

Out[34]:

```
          rexpret     sexpret
count  200.000000  257.000000
mean    64.870000   64.459144
```

```
std       4.203743    4.221328
min      54.000000   54.000000
25%      62.000000   62.000000
50%      65.000000   64.000000
75%      68.000000   66.000000
max      84.000000   80.000000
```

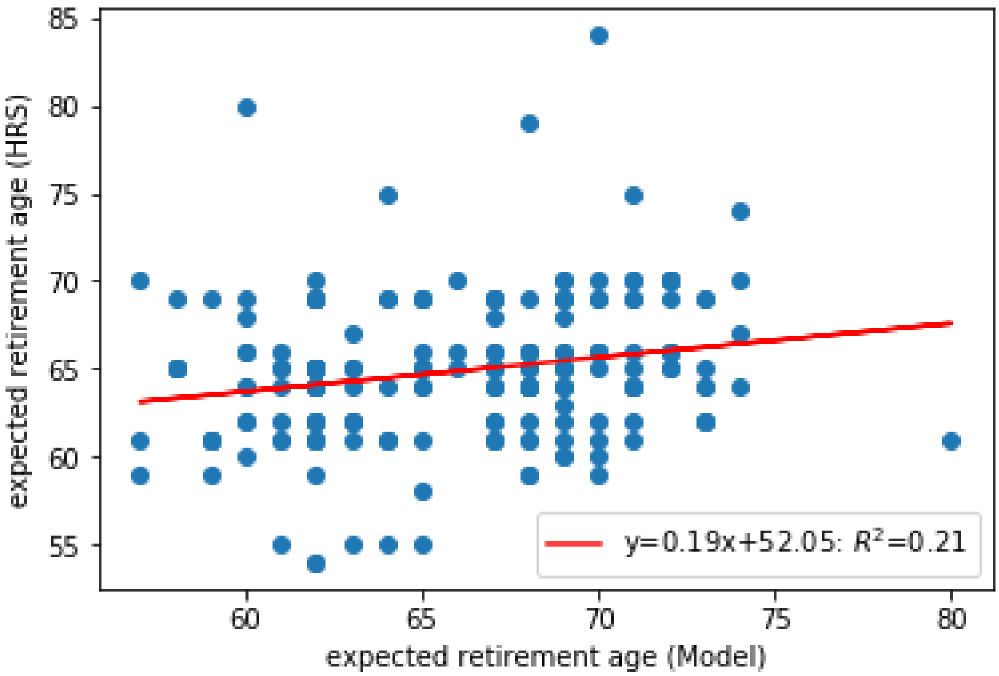
Correlations with simulated expected retirement age

```
In [35]: print('Males   : ',disc.sim[['rexpret','rexpret_sim']].corr())
         print('Females : ',disc.sim[['sexpret','sexpret_sim']].corr())
```

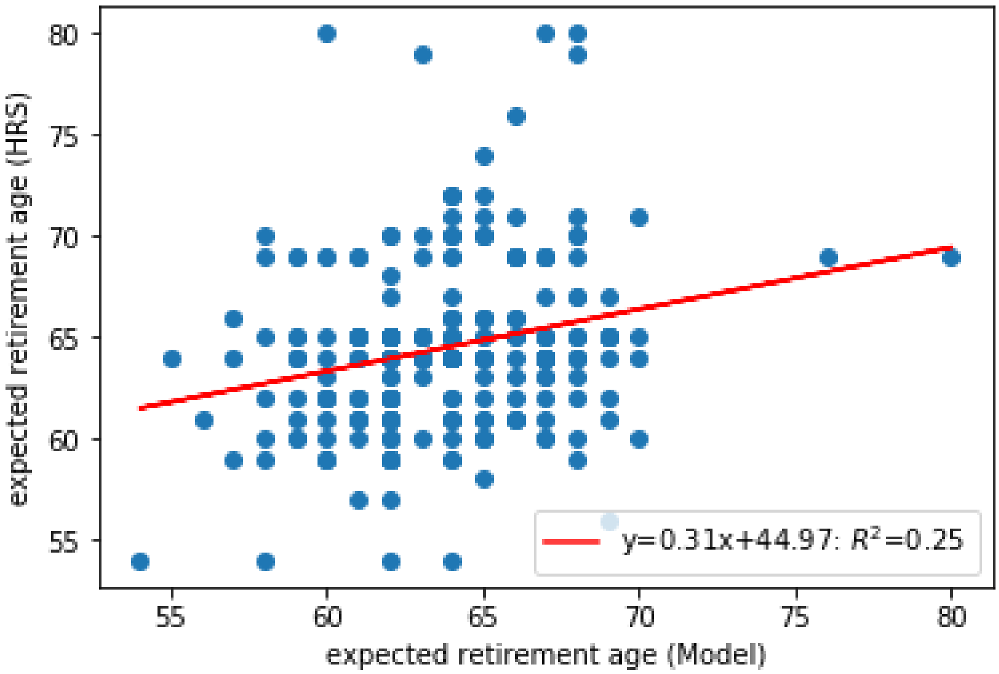
```
Males   :                  rexpret  rexpret_sim
rexpret      1.000000     0.208299
rexpret_sim  0.208299     1.000000
Females :                 sexpret  sexpret_sim
sexpret      1.000000     0.245102
sexpret_sim  0.245102     1.000000
```

We will do a figure to check correlation

```
In [36]: from scipy.stats import linregress
         data = disc.sim[['rexpret_sim','rexpret']].dropna()
         x = data['rexpret_sim']
         y = data['rexpret']
         slope, intercept, r_value, p_value, std_err = linregress(x,y)
         line = slope*x+intercept
         plt.figure()
         plt.scatter(x,y,label='')
         plt.plot(x, line, 'r', label='y={:.2f}x+{:.2f}: $R^2$={:.2f}'.format(slope,intercept,r
         plt.xlabel('expected retirement age (Model)')
         plt.ylabel('expected retirement age (HRS)')
         plt.legend(loc=4)
         plt.savefig('../tex/figures/match_males.eps')
         plt.show()
```




```
In [37]: from scipy.stats import linregress
         data = disc.sim[['sexpret_sim','sexpret']].dropna()
         x = data['sexpret_sim']
         y = data['sexpret']
         slope, intercept, r_value, p_value, std_err = linregress(x,y)
         line = slope*x+intercept
         plt.figure()
         plt.scatter(x,y,label='')
         plt.plot(x, line, 'r', label='y={:.2f}x+{:.2f}: $R^2$={:.2f}'.format(slope,intercept,r
         plt.xlabel('expected retirement age (Model)')
         plt.ylabel('expected retirement age (HRS)')
         plt.legend(loc=4)
         plt.savefig('../tex/figures/match_females.eps')
         plt.show()
```

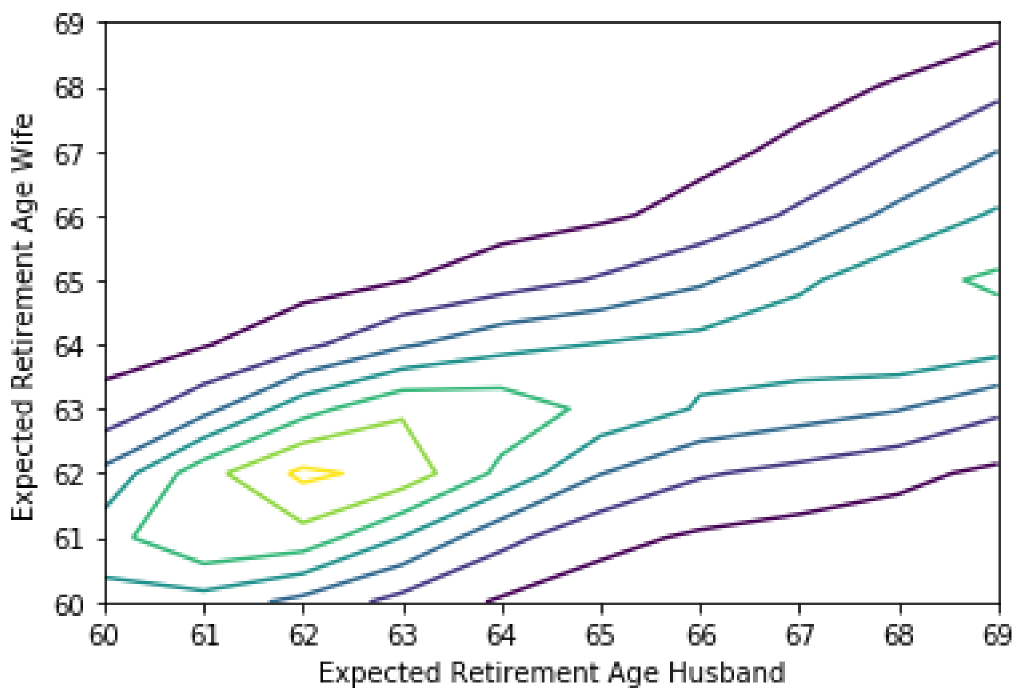


1.3 Distribution of Retirement Ages

```
In [38]: %matplotlib inline
         from scipy.stats import gaussian_kde
         from numba import jit
         import numpy as np
         x = disc.sim['rexpret_sim'].tolist()
         y = disc.sim['sexpret_sim'].tolist()
         print(disc.sim[['rexpret_sim','sexpret_sim']].describe())
         print('correlation : ', disc.sim[['rexpret_sim','sexpret_sim']].corr())
         X, Y = np.mgrid[60:69:10j, 60:69:10j]
         positions = np.vstack([X.ravel(), Y.ravel()])
         values = np.vstack([x, y])
         kernel = gaussian_kde(values)
         Z = np.reshape(kernel(positions).T, X.shape)
         plt.figure()
         plt.contour(X,Y,Z)
         plt.xlabel('Expected Retirement Age Husband')
         plt.ylabel('Expected Retirement Age Wife')
         plt.savefig('../tex/figures/retages.eps')
```

```
       rexpret_sim  sexpret_sim
count   464.000000   464.000000
mean     65.728448    63.829741
std       4.670523     3.130547
```

```
min      55.000000    54.000000
25%      62.000000    62.000000
50%      65.000000    64.000000
75%      69.000000    66.000000
max      80.000000    80.000000
correlation :              rexpret_sim  sexpret_sim
rexpret_sim     1.000000     0.741182
sexpret_sim     0.741182     1.000000
```



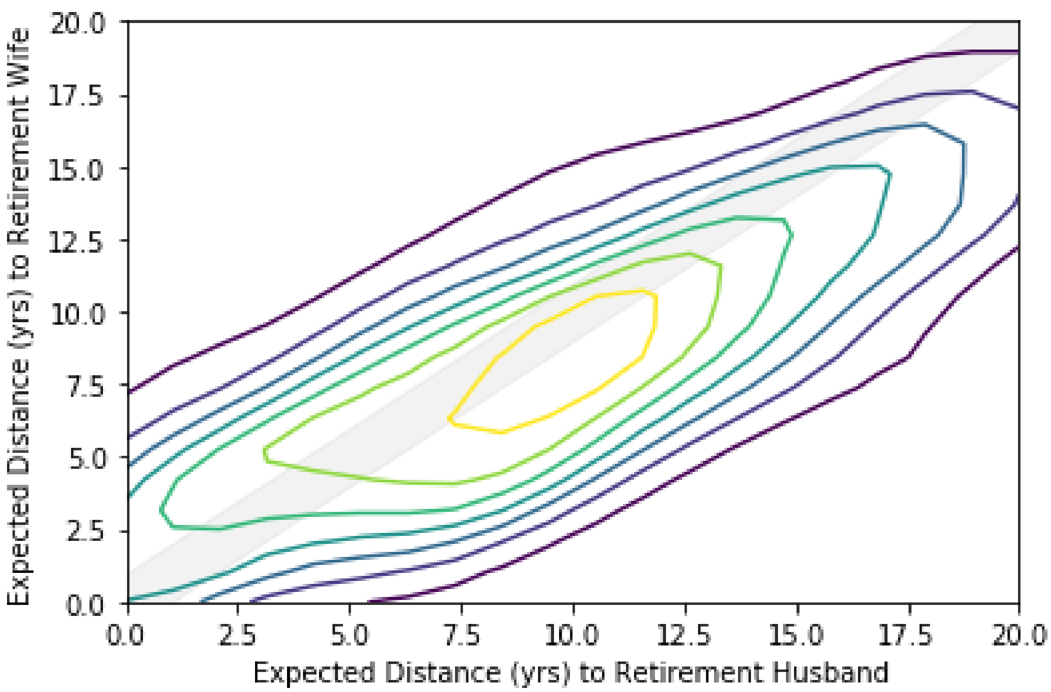

```
In [39]: disc.sim['distance_m'] = disc.sim['rexpret_sim'] - disc.sim['rage_mod']
         disc.sim['distance_f'] = disc.sim['sexpret_sim'] - disc.sim['sage_mod']
         print(disc.sim[['distance_m','distance_f']].describe())
         print('correlation: ', disc.sim[['distance_m','distance_f']].corr())

         %matplotlib inline
         from scipy.stats import gaussian_kde
         from numba import jit
         import numpy as np
         x = disc.sim['distance_m'].tolist()
         y = disc.sim['distance_f'].tolist()

         X, Y = np.mgrid[0:20:20j, 0:20:20j]
         positions = np.vstack([X.ravel(), Y.ravel()])
         values = np.vstack([x, y])
```

```
kernel = gaussian_kde(values)
Z = np.reshape(kernel(positions).T, X.shape)
plt.figure()
plt.contour(X,Y,Z)
xx = np.linspace(0,20,20)
plt.fill_between(xx, xx-1, xx+1, color='grey', alpha='0.1')
plt.xlim([0,20])
plt.ylim([0,20])
plt.xlabel('Expected Distance (yrs) to Retirement Husband')
plt.ylabel('Expected Distance (yrs) to Retirement Wife')
plt.savefig('../tex/figures/distances.eps',dpi=600)
```

```
       distance_m  distance_f
count  464.000000  464.000000
mean     9.092672    8.168103
std      5.640990    4.582549
min      0.000000    0.000000
25%      5.000000    5.000000
50%      9.000000    8.000000
75%     13.000000   11.000000
max     23.000000   23.000000
correlation:              distance_m  distance_f
distance_m    1.000000    0.766069
distance_f    0.766069    1.000000
```



1.4 Joint Retirement

We will now re-rerun the discount model (baseline) with three scenarios: reshuffling heterogeneity, reshuffling wages and shutting down complementarity.

```
In [40]: wages = sp('discount')
         wages.chgoption('iload','T')
         wages.chgoption('idiscount','T')
         wages.chgoption('ishufwages','T')
         wages.estimate()
```

```
Default options:
isurvival  =  T
iload  =  F
iestimate  =  T
data  =  hrs_final_ref.csv
info  =  info_ref.dat
includeratings  =  F
icomplement  =  T
ihetero  =  T
icorr  =  T
iunitary  =  F
idiscount  =  F
idirect  =  F
arf  =  0.06
drc  =  0.08
reprate  =  0.6
ishufheter  =  F
ishufwages  =  F
iblockcomp  =  F
changed item  iload  with value F  to  T  ...
changed item  idiscount  with value F  to  T  ...
changed item  ishufwages  with value F  to  T  ...
```

```
In [41]: heter = sp('discount')
         heter.chgoption('iload','T')
         heter.chgoption('idiscount','T')
         heter.chgoption('ishufheter','T')
         heter.estimate()
```

```
Default options:
isurvival  =  T
iload  =  F
iestimate  =  T
data  =  hrs_final_ref.csv
info  =  info_ref.dat
includeratings  =  F
icomplement  =  T
```

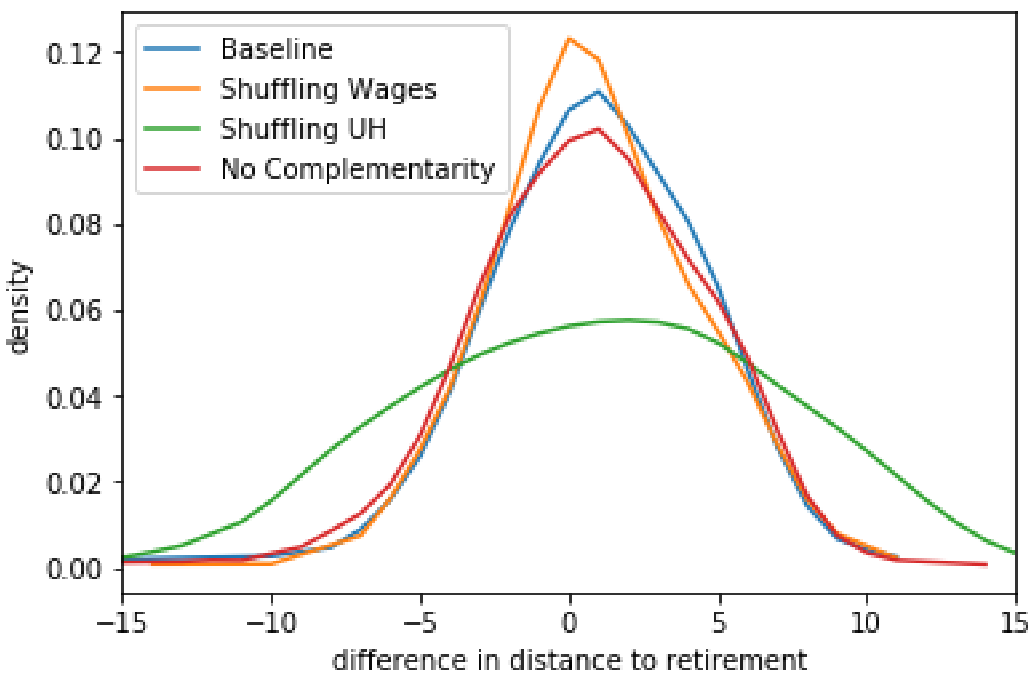
```
ihetero  =  T
icorr  =  T
iunitary  =  F
idiscount  =  F
idirect  =  F
arf  =  0.06
drc  =  0.08
reprate  =  0.6
ishufheter  =  F
ishufwages  =  F
iblockcomp  =  F
changed item  iload  with value F  to  T  ...
changed item  idiscount  with value F  to  T  ...
changed item  ishufheter  with value F  to  T  ...
```

```
In [46]: comp = sp('discount')
         comp.chgoption('iload','T')
         comp.chgoption('idiscount','T')
         comp.chgoption('iblockcomp','T')
         comp.estimate()
```

```
Default options:
isurvival  =  T
iload  =  F
iestimate  =  T
data  =  hrs_final_ref.csv
info  =  info_ref.dat
includeratings  =  F
icomplement  =  T
ihetero  =  T
icorr  =  T
iunitary  =  F
idiscount  =  F
idirect  =  F
arf  =  0.06
drc  =  0.08
reprate  =  0.6
ishufheter  =  F
ishufwages  =  F
iblockcomp  =  F
changed item  iload  with value F  to  T  ...
changed item  idiscount  with value F  to  T  ...
changed item  iblockcomp  with value F  to  T  ...
```

```
In [47]: specs = [disc,wages,heter,comp]
         names =  ['Baseline','Shuffling Wages','Shuffling UH','No Complementarity']
```

```
data = []
plt.figure()
for i,s in enumerate(specs):
    s.sim['distance_m'] = s.sim['rexpret_sim'] - s.sim['rage_mod']
    s.sim['distance_f'] = s.sim['sexpret_sim'] - s.sim['sage_mod']
    s.sim['joint'] = np.abs(s.sim['distance_m'] - s.sim['distance_f'])<=1.0
    s.sim['joint_yrs'] = s.sim['distance_m'] - s.sim['distance_f']
    x = s.sim['joint_yrs'].values
    x.sort()
    ff = gaussian_kde(x)
    plt.plot(x,ff(x),label=names[i])
    data.append([s.sim['rexpret_sim'].mean(),s.sim['sexpret_sim'].mean(),s.sim['joint
table = pd.DataFrame(data=data,index=names,columns=['Ret Age Males','Ret Age Females'
def f(x):
    return '{:1.3f}'.format(x)
with open('../tex/tables/joint.tex','w') as tf:
    tf.write(table.to_latex(formatters=[f,f,f]))
plt.legend(loc=2)
plt.xlabel('difference in distance to retirement')
plt.ylabel('density')
plt.xlim([-15,15])
plt.savefig('../tex/figures/compare_distances.eps')
plt.show()
table
```



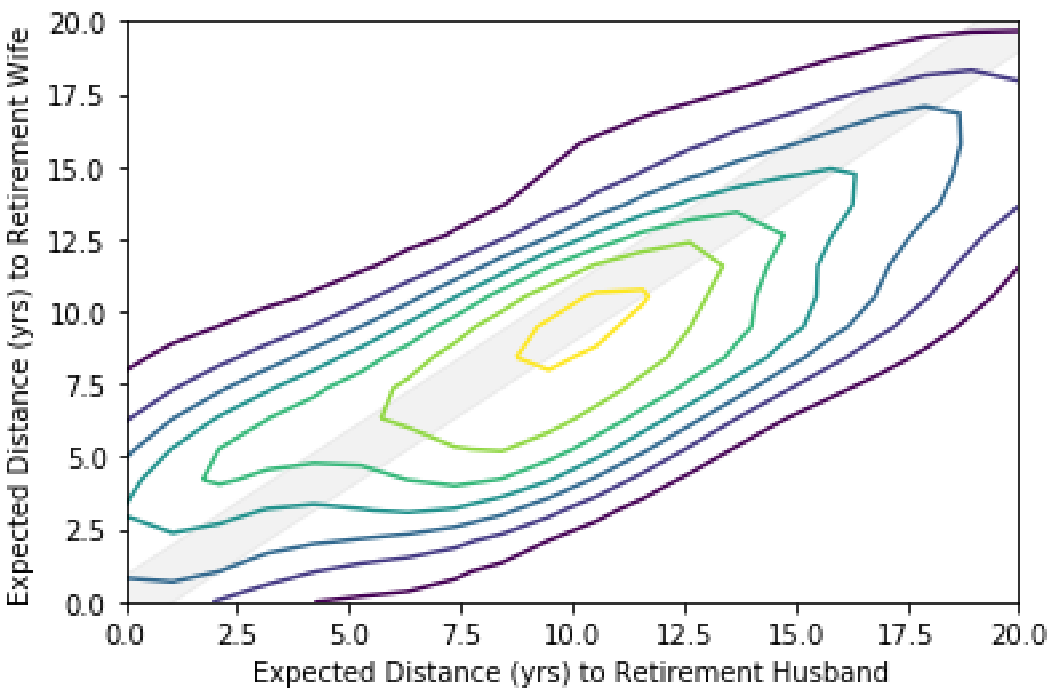
Out[47]:

	Ret Age Males	Ret Age Females	Fraction Joint
Baseline	65.728448	63.829741	0.336207
Shuffling Wages	65.629310	63.803879	0.368534
Shuffling UH	65.765086	63.892241	0.176724
No Complementarity	66.142241	64.471983	0.303879

In [48]:

```
%matplotlib inline
from scipy.stats import gaussian_kde
from numba import jit
import numpy as np
x = comp.sim['distance_m'].tolist()
y = comp.sim['distance_f'].tolist()

X, Y = np.mgrid[0:20:20j, 0:20:20j]
positions = np.vstack([X.ravel(), Y.ravel()])
values = np.vstack([x, y])
kernel = gaussian_kde(values)
Z = np.reshape(kernel(positions).T, X.shape)
plt.figure()
plt.contour(X,Y,Z)
xx = np.linspace(0,20,20)
plt.fill_between(xx, xx-1, xx+1, color='grey', alpha='0.1')
plt.xlim([0,20])
plt.ylim([0,20])
plt.xlabel('Expected Distance (yrs) to Retirement Husband')
plt.ylabel('Expected Distance (yrs) to Retirement Wife')
plt.savefig('../tex/figures/distances_nocomp.eps',dpi=600)
```



```
In [45]: disc.sim['agediff'] = disc.sim['rage_mod']- disc.sim['sage_mod']
         print(comp.sim[['distance_m','distance_f']].describe())
         print(disc.sim[['distance_m','distance_f']].describe())
```

```
       distance_m  distance_f
count  464.000000  464.000000
mean     9.810345    9.519397
std      5.755889    4.703711
min      0.000000    0.000000
25%      6.000000    6.000000
50%     10.000000    9.000000
75%     14.000000   13.000000
max     24.000000   23.000000
       distance_m  distance_f
count  464.000000  464.000000
mean     9.092672    8.168103
std      5.640990    4.582549
min      0.000000    0.000000
25%      5.000000    5.000000
50%      9.000000    8.000000
75%     13.000000   11.000000
max     23.000000   23.000000
```

2 Policy Simulation

Finally, we have to check what happens when we do policy simulations. We will do policy simulations over two parameters, the ARF (actuarial reduction factor) and the generosity of the pension (replacement rate).

```
In [ ]: arf = [0,0.09]
        drc = [0,0.11]
        rep = [0.4,0.8]
        factors = zip(arf,drc)
        experiments = [disc]
        for a,d in factors:
            this = sp('discount')
            this.chgoption('iload','T')
            this.chgoption('idiscount','T')
            this.chgoption('arf',a)
            this.chgoption('drc',d)
            this.estimate()
            this.sim['distance_m'] = this.sim['rexpret_sim'] - this.sim['rage_mod']
            this.sim['distance_f'] = this.sim['sexpret_sim'] - this.sim['sage_mod']
            this.sim['joint'] = np.abs(this.sim['distance_m'] - this.sim['distance_f'])<=1.0
```

```
            experiments.append(this)
        for r in rep:
            this = sp('discount')
            this.chgoption('iload','T')
            this.chgoption('idiscount','T')
            this.chgoption('reprate',r)
            this.estimate()
            this.sim['distance_m'] = this.sim['rexpret_sim'] - this.sim['rage_mod']
            this.sim['distance_f'] = this.sim['sexpret_sim'] - this.sim['sage_mod']
            this.sim['joint'] = np.abs(this.sim['distance_m'] - this.sim['distance_f'])<=1.0
            experiments.append(this)
```

```
In [ ]: data = [[e.sim['rexpret_sim'].mean(),e.sim['sexpret_sim'].mean(),e.sim['joint'].mean()]
        names = ['Baseline','No Penalty','High Penalty','Low Generosity','High Generosity']
        table = pd.DataFrame(data=data,columns=['Ret Age Males','Ret Age Females','Fraction Re
        print(table)
        def f(x):
            return '{:1.3f}'.format(x)
        with open('../tex/tables/policy.tex','w') as tf:
            tf.write(table.to_latex(formatters=[f,f,f]))
```