Manual

The attached Matlab codes replicate Gustman and Steinmeier (2008) C++ codes. As in the original codes, it could do various jobs by calling different modes. Users could change value of mode to indicate which type of work to do:

test mode (0): test the program by creating a synthetic individual whose characteristics are given in function setdata;

iteration mode (1): do a single q function evaluation, given initial parameter values and the HRS data;

estimation mode (2): estimate preference parametes using the HRS data;

simulation mode (3): simulate moments based on the estimated parameters (must follow mode 2);

As in the paper, this program should be able to examine the effect of various Social Security policies on labor supply decisions (not in the original codes, need double check). Again, users could change value of policy to indicate which type of policy to test:

baseline policy (0): the actual Social Security rules;

counterfactual policy (1): Social Security rules advanced 6 years;

counterfactual policy (2): Social Security rules advanced 12 years;

For the utility function, the program assumes the basic format C ^ (alpha) / alpha, and uses its marginal utility function all the time. If the reader would like to change the utility function, global search “marginal utility” in comments and substitute own marginal utility function.

For budget constrain, the total resource or income includes current assets, pension benefit, social security benefit, other income, and job salary, depending on different time periods (terminal age, after retirement age or before retirement age) and different job status (career job work, part-time work after retirement or return to full-time job). The corresponding equations are in line 89, 314, 1143, 1196, 1300, 1400, 2169 and 2273 in calcmodel.m MATLAB file. Then the consumption calculation based on the assets and income in line 94, 116, 138, 167 and 204in GSModelinc.m MATLAB file, depending on the individual’s low or high time preference, and regular solution or corner solution in interpolation.

Program flow:

1. The entrance program is “GSmodel\_replicate.m”. This program will call other functions to do various calculations and output results.
2. At the beginning, define mode and policy as mentioned above.
3. Define structure variable “gv”.

“gv” stands for global variable. It is a structure variable that enters various functions and provides values of pre-determined, unchanged parameters. For more details, see attached variable list.

1. Create or retrieve structure variable “oridata” from the prepared HRS dataset.

In “test” mode, this program will create a hypothetical household to verify the validity of the codes. In “iteration”, “estimation” and “simulation” modes, this program will read in the actual HRS data.

See attached variable list to find out what variables are included in the calculations.

In the “iteration” and “estimation” modes, also generate variable arglist.nmat. “arglist” is a structure variable captures results of calculations. “arglist.nmat” is a vector of indicators of whether the corresponding moment is observed from the HRS. If it is 1, the corresponding data moment from the HRS is observed, 0 otherwise.

See attached variable list for definitions of other “arglist” variables.

1. After defining variables, the program will enter various functions based on “mode” from user input. I use “test” mode to explain the codes below. Explanations of the differences among various modes are at the end.
2. For the “test” mode, the function used is called “test”. In the function, first initialize “arglist.newalpha”. The variable is reset to 1 every time a new alpha (consumption parameter) is used. If it is 1, the time preference factor “rho” will be re-calculated.
3. Next, enter function “indivmodel”. This function will take “gv” parameters and the created household information “oridata” to calculate retirement moments for the household. It will return “arglist”, the structure variable of retirement moments.
4. In “indivmodel” function, for each household, first gets their firstage and lastage from data. firstage is defined as the earliest possible age to retire from career job. lastage is defined as the latest possible age to work in any type of jobs.
5. Next, enter “calcprep” function. It will create variables necessary for later calculations, such as survival probability, asset grid, asset shock grid, social security grid, DB pension grid, DC pension grid, partial utility grid, leisure utility grid, etc. Social Security benefits in all types of situations are also calculated here in a sub-function “sscalc”.
6. The next step is to calculate household-specific time preference factor “rho” for each household. In this section, an approximated retirement age is used to calculate people’s income, and a simplified backward induction is calculated to find out consumption and saving decisions given current economic state.
7. After having all policy functions from the simplified backward induction, simulate the household to find out his wealth level in 1992.
8. Keep adjusting rho to match simulated 1992 wealth with observed 1992 wealth.
9. After getting rho, insert all existing information into function “calcmodel” to do the main backward induction. Find out policy functions of savings and labor supply decisions given current economic state. See details below.
10. Next is “getmoments” function, which run 10,000 simulations per household (they experience different time preference shocks) using the policy functions just calculated.
11. Summarize and save the average retirement moments to structure variable arglist.
12. “test” will then print the retirement moments after getting arglis results.

Details about main backward induction in “calcmodel”:

The main algorithm is to equate marginal utility current period and discounted expected marginal utility next period to decide optimal consumption. In addition, compare total utility to decide retirement decisions.

At the terminal age:

Individuals are surely retired;

It’s optimal for households to consume all remaining resources.

Calculate the corresponding optimal marginal utility (marguc) given vcat (survival status), pcat (DB pension benefits), scat (Social Security benefits) and acat (asset).

Save the saving (consumption) decisions to savr (saving after retirement).

Save optimal marginal utility to marginal utility next period (when calculating age - 1)

(marguf)

From lastage + 1 to 98:

Individuals are surely retired.

For each age:

Calculate expected marginal utility over asset return shocks, given vcat (survival status), dcat (delayed DB pension), pcat (DB pension benefits), scat (Social Security benefits) and acat (end of period asset).

Calculate expected marginal utility over survival status.

Calculate optimal consumption, savings, marginal utility (marguc) given initial asset, vcat, dcat, pcat, scat and next period discounted expected marginal utilities (marguf).

Save the saving (consumption) decisions to savr (saving after retirement).

Save optimal marginal utility to marginal utility next period (when calculating age - 1)

(marguf)

At lastage:

Separately consider if the husbands are currently working in their career job, or the husbands are currently retired from their career job;

Calculate expected marginal utility (margufw) if currently working in career job at lastage;

He will surely retire at lastage + 1;

His DC pension balance will be transferred to asset at lastage + 1;

Add one dimension, leisure preference, for later calculations;

It is the same for all people at lastage + 1.

Calculate expected marginal utility (margufr) if currently retired from career job at lastage;

No DC balance transfer necessary;

Add one dimension, leisure preference, for later calculations;

It is the same for people at lastage + 1.

From firstage to lastage:

Individuals could in any of the four retirement status: working in career job, working in returned full time work, working in part time work, retired (not working at all).

For each age:

Calculate expectations with respect to asset returns, retired from career job;

Calculate expectations with respect to survival, retired from career job;

Calculate consumption equivalents of future marginal utility, retired from career job;

Calculate expectations with respect to returns, currently working in career job;

Calculate expectations with respect to survival, currently working in career job;

If the husband dies in career job, he is considered as retired at the age of death and his DC pension balance is added to asset immediately;

Calculate consumption equivalents of future marginal utility, currently working in career job;

Calculate consumption, saving, marginal and total utility for staying in main job;

Calculate consumption, saving, marginal and total utility for retirement;

Calculate consumption, saving, marginal and total utility for partial retirement;

Calculate consumption, saving, marginal and utility for return to work;

Choose job types to maximize future total utility if previously retired from career job;

Choose job types to maximize future total if previously working in career job;

From 25 to firstage – 1;

Individuals are surely working in their career job;

For each age:

Calculate expected marginal and total utility with respect to asset returns, husband alive (firstage > 63);

Calculate expected marginal and total utility with respect to survival status, husband alive;

If the husband dies in career job, he is considered as retired at the age of death and his DC pension balance is added to asset immediately;

Calculate optimal consumption, saving, marginal and total utility, currently working in career job, husband living;

Calculate expected marginal and total utility with respect to asset returns, husband deceased (vcat == 3);

Calculate optimal consumption, saving, marginal and total utility, currently working in career job, husband deceased;

Other modes:

1. In “iteration” mode, the function “iter” will read in data from the HRS, calculate retirement moments for each household using initial given parameter values, calculate q statistic and report average retirement moments.
2. In “estimation” mode, the function “qmin” will keep adjusting the parameter values in order to minimize the distance between simulated moments with observed moments.
3. In “simulation” mode, the function “simu” will calculate average retirement moments using the parameter values estimated from “estimation” mode.

Definition and explanations of main variables used in GSmodel:

gv: global variable; it is a structure variable carries pre-determined, unchanged parameters;

gv.mode: program mode; 0. test mode; 1. iteration mode; 2. estimation mode; 3. simulation mode;

gv.policy: 0. actual Social Security rules; 1. Social Security rules advanced 6 years; 2. Social Security rules advanced 12 years;

gv.maxobs: maximum number of observations after treatment from the HRS/to be evaluated using dynamic programs; it is for test purpose only; it doesn’t involve in any calculations;

gv.nobs: number of valid observations after treatment from the HRS/to be evaluated using dynamic programs;

gv.nparms: number of preference parameters to be estimated;

gv.nmoments: number of moments to be matched;

gv.nsim: number of simulations for each household;

gv.nacatlim: grid size of asset;

gv.ntcatlim: grid size of partial retirement preference;

gv.necatlim: grid size of leisure preference shock;

gv.nlcatlim: grid size of DC pension balance;

gv.npcatlim: grid size of DB pension benefits;

gv.nrcatlim: grid size of asset return shock;

gv.nscatlim: grid size of Social Security benefits;

gv.nvcatlim: grid size of survival status;

gv.ndcatlim: indicator of delayed DB pension, 1 is no, 2 is yes;

gv.t0: starting age; everybody enters the model at starting age with $0 in asset;

gv.T: terminal age, the max age of life; everybody dies after this age;

gv.parm.alpha: initial parameter value of consumption parameter; transformation of CRRA coefficient (alpha = 1 – crra coefficient);

gv.parm.beta0: initial parameter value of constant in beta vector; beta vector defines the weight individuals put on leisure as in h(t) = exp(beta\*X(t) + epsilon(t));

gv.parm.beta1: initial parameter value of coefficient of (age - 62) in beta vector;

gv.parm.beta2: initial parameter value of coefficient of health in beta vector; health is a dummy variable takes 1 if the individual has health problem;

gv.parm.sige: initial parameter value of standard deviation of random leisure preference (epsilon); individuals start with a random drawn of epsilon from a distribution with mean 0 and standard deviation sige, keeping it until retires from their career job; then epsilon could be changed following an AR(1) process: epsilon(t) = erho \* epsilon(t-1) + u(sige);;

gv.parm.erho: initial parameter value of autocorrelation of epsilon after retirement;

gv.parm.gamma0: initial parameter value of constant in gamma vector; gamma shows the preference for partial retirement as Vp = L ^ gamma;

gv.parm.gamma1: initial parameter value of coefficient of (age - 65) in gamma vector;

gv.survtable: life table; format survetable(birthyear, gender, age); birthyear is 1 if born in 1900, 2 if born in 1901, and so on; gender is 1 if male, 2 if female; age is 1 indicating age 0 survival numbers, 2 indicating age 1 survival numbers, and so on;

gv.inflrate: inflation rate, fixed at all time;

gv.penadjrate: fraction of inflation that is offset by DB pension increases, so the rate to adjust pension income each year is gv.inflrate \* (1- gv.penadjrate);

gv.realrateg: geometric mean of real returns; rg = (r1 \* r2 \* ... \* rn) ^ (1 / n);

gv.realratea: arithmatic mean of real returns; (r1 + r2 + ... + rn) / n;

gv.realratesd: standard deviation of real returns;

gv.realrate: actual real returns from 1926 to 2000; data from file ibbotson valuation edition yearbook 2001;

oridata/data: data structure variable; oridata contains information of all observations from the HRS; data takes information from one of the observations and does calculations;

data.inputobs: order of observation in the final input file, not used in calculations;

data.birthyear: husband's birth year;

data.spbirthyear: wife's birth year;

data.healthage: initial age of having health problem; people cannot go back as healthy later;

data.penjobend: age at last year of pension job;

data.age: age at six waves of survey time, 6 X 1;

data.ret: retirement status at six waves of survey time, 6 X 1; it is 5 if working full time; 3 if working part time; 1 if completely retired;

data.retv: retirement status vector for ages 50 - 69, 20 X 1; again, 5 if working full time; 3 if working part time; 1 if completely retired;

data.ftwage: husband's career job wages if working at ages 25 - 69, 45 X 1; all income / wealth amounts are in 1992 dollars;

data.prwage: husband's part time work wages if working at ages 50 - 69, 20 X 1;

data.secwage: husband's returned full time work wages if working at ages 50 - 69, 20 X 1;

data.spwage: wife's career job wages if working at spouse ages 25 - 69, 45 X 1; spouse does not have options to do part time job or return to full time job after retirement;

data.pensben: husband's annual DB pension benefit if the husband retires at 50 - 70, 21 X 1;

data.penstart: husband's starting age of collecting DB pension benefits if the husband retires at 50 - 70, 21 X 1; starting age could be later than retirement age;

data.plumpsum: husband's DC pension balance if the husband retires at 50 - 70, 21 X 1;

data.eowamount: husband's early out window lump-sum amounts if the husband retires at 50 - 70, 21 X 1;

data.pia: husband's primary insurance amount if the husband retires at 50 - 70, 21 X 1;

data.ncpens: husband's non-covered pension benefits if the husband retires at 50 - 70, 21 X 1;

data.sppia: wife’s PIA if wife retires at 50 - 70, 21 X 1;

data.spncpens: wife's non-covered pension benefits if wife retires at 50 - 70, 21 X 1;

data.sppensben: wife’s annual DB pension benefits given her retirement age, exogenous and fixed in the model;

data.sppenstart: wife's DB pension starting age given her retirement age;

data.spplumpsum: wife's DC pension balance at her retirement age;

data.assetwealth: household’s total asset, including financial asset and land, business, and other wealth in 1992 (exclude housing);

data.othincome: household’s other income, such as inheritances, at husband’s 25 - 99, 75 X 1;

data.inccat: category of households’ lifetime income; 1 is low, 2 is mid and 3 is high;

calc: household specific structure variable contains variables used in calculations;

calc.obs: order of the household in oridata; row of oridata;

calc.alpha: parameter value of consumption parameter in calculations;

calc.beta0: parameter value of constant in beta vector in calculations;

calc.beta1: parameter value of coefficient of (age - 62) in beta vector in calculations;

calc.beta2: parameter value of coefficient of health in beta vector in calculations;

calc.sige: parameter value of standard deviation of random leisure preference (epsilon) in calculations;

calc.erho: parameter value of autocorrelation of epsilon after retirement in calculations;

calc.gamma0: parameter value of constant in gamma vector in calculations;

calc.gamma1: parameter value of coefficient of (age - 65) in gamma vector in calculations;

calc.nlcatlim: grid size of lump-sum DC pension balance in calculations; it is not necessarily equal to gv.nlcatlim; it is 1 if DC pension is not applicable;

calc.npcatlim: grid size of DB pension benefits in calculations; it is not necessarily equal to gv.npcatlim; it is 1 if DB pension is not applicable;

calc.acats: asset grid, gv.nacatlim X 1;

calc.ecats: leisure preference grid, gv.necatlim X 1;

calc.lcats: lump-sum DC pension balance grid, calc.nlcats X 1;

calc.pcats: DB pension annual benefit grid, calc.npcats X 1;

calc.tcats: utility of partial retirement grid for 50 - 69, gv.ntcatlim X 20;

calc.firstage: earliest possible age to retire from career job;

calc.lastage: oldest possible age to work in any type of jobs;

calc.earlypenage: first age between firstage and lastage that DB pension benefits can be immediately collected after retirement;

calc.delayedpenage: earliest age delayed DB pension benefits can be collected;

calc.contribstartage: DC pension contributions start age;

calc.initialpcat: initial DB pension benefits grid point if starts to receive DB pension at 50 - 70, 21 X 1;

calc.dccontrib: husband’s DC pension annual contribution amount at 25 - 69, 45 X 1;

calc.othincome: other income by age (25 - 99) and survival states (1 - 3), 75 X 3;

calc.sserage: Social Security early entitlement age; the earliest age eligible to claim SS;

calc.nrage: husband’s SS normal retirement age; if he claims at NRA, he receives 100% of his PIA;

calc.spnrage: wife's SS normal retirement age;

calc.delayedret: SS delayed retirement credit rate; percent increase per year after NRA; apply to old worker's own benefits;

calc.spdelayedret: spouse's SS delayed retirement credit rate;

calc.initialscat: initial SS benefits grid point if the husband retires at 50 - 70, 21 X 1;

calc.ssbenf: husband's SS own benefits as if the husband retires at scat ages (50, EEA, NRA, 70);

calc.ssbenm: SS benefits if currently working in career job by age (25 - 69) and survival status (1 - 3);

calc.ssbenr: SS benefits if fully retired by age (50 - 99), survival status (1 - 3) and retirement (scat) ages (1 - gv.nscatlim);

calc.ssbens: SS benefits if working in returned full time work by age (50 - 69), survival status (1 - 2) and retirement (scat) ages (1 - gv.nscatlim);

calc.ssbenp: SS benefits if still working part time by age (50 - 69), survival status (1 - 2) and retirement (scat) ages (1 - gv.nscatlim);

calc.benchanges: SS benefits adjustments if working in returned full time work by age (62 - 69), surv status (1 - 2) and retirement (scat) ages (1 - gv.nscatlim);

calc.benchangep: SS benefits adjustments if working in part time work by age (62 - 69), surv status (1 - 2) and retirement (scat) ages (1 - gv.nscatlim);

calc.srate: survival probabilities by age (25 - 99) and gender (1 is Husband only, 2 is spouse only); Household specific based on husband’s birth year and wife’s birth year, 2 X 75;

calc.transrate: survival status transition probabilities from both spouses alive to (1. both spouses alive, 2. only husband alive, and 3. only wife alive) by age (25 - 99), 3 X 75;

calc.etrans: transition probability matrix of epsilon (the error term in weighting function); AR(1); gv.necatlim X gv.necatlim;

calc.rprob: asset return shock probabilities; iid; gv.nrcatlim X 1;

calc.rtransa: transition of asset grid point after asset return shock; gv.nacatlim X gv.nrcatlim;

calc.rtransl: transition of DC pension balance grid point after asset return shock by age (25 - 69), 45 X calc.nlcatlim X gv.nrcatlim ;

calc.utilr: partial retirement utility by leisure preference (1 – gv.necatlim) and age (50 – 69), gv.necatlim X 20;

calc.income: approximated income by age (25 - 99) and survival status (1 - 3); used in rhocalc and associated functions only, 3 X 75;

calc.cons: approximated optimal consumption by age (25 - 99), survival status (1 - 3) and asset (1 - gv.nacatlim), 75 X 3 X gv.nacatlim;

calc.totuw: expected total utility from current period to terminal age given optimal behaviors in later periods, currently working in career job, by age (25 – 69), DC pension balance (1 – calc.nlcatlim), leisure preference (1- gv.necatlim) and asset (1 – gv.nacatlim), 45 X calc.nlcatlim X gv.necatlim X gv.nacatlim;

calc.totud: expected total utility from current period to terminal age given optimal behaviors in later periods, currently retired from career job, by age (50 – 69), job category besides career job (1 – 3), delayed DB pension (1 – 2), DB pension benefits (1 – calc.npcatlim), SS benefits (1- gv.nscatlim), leisure preference (1- gv.necatlim) and asset (1 – gv.nacatlim), 20 X 3 X gv.ndcatlim X calc.npcatlim X gv.nscatlim X gv.necatlim X gv.nacatlim;

calc.savw: optimal savings, currently working in career job, by age (25 – 69), DC pension balance (1 – calc.nlcatlim), leisure preference (1- gv.necatlim) and asset (1 – gv.nacatlim), 45 X calc.nlcatlim X gv.necatlim X gv.nacatlim;

calc.savd: optimal savings, currently retired from career job, by age (50 – 69), job category besides career job (1 – 3), delayed DB pension (1 – 2), DB pension benefits (1 – calc.npcatlim), SS benefits (1- gv.nscatlim), leisure preference (1- gv.necatlim) and asset (1 – gv.nacatlim), 20 X 3 X gv.ndcatlim X calc.npcatlim X gv.nscatlim X gv.necatlim X gv.nacatlim;

calc.savr: optimal savings, retired, by age (70 – 99), delayed DB pension (1 – 2), DB pension benefits (1 – calc.npcatlim), SS benefits (1- gv.nscatlim), and asset (1 – gv.nacatlim), 30 X gv.ndcatlim X calc.npcatlim X gv.nscatlim X gv.nacatlim;

calc.mjchoice: job (retirement) decision, currently working in career job, by age (50 – 69), DC pension balance (1 – calc.nlcatlim), leisure preference (1- gv.necatlim) and asset (1 – gv.nacatlim), 20 X calc.nlcatlim X gv.necatlim X gv.nacatlim; 1 is staying in career job, 2 is returned full time work, 3 is part time work, 4 is retired;

calc.retchoice: job (retirement) decision, currently retired from career job, by age (50 – 69), delayed DB pension (1 – 2), DB pension benefits (1 – calc.npcatlim), SS benefits (1- gv.nscatlim), leisure preference (1- gv.necatlim) and asset (1 – gv.nacatlim), 20 X gv.ndcatlim X calc.npcatlim X gv.nscatlim X gv.necatlim X gv.nacatlim;

; 2 is returned full time work, 3 is part time work, 4 is retired;

calc.randseed: seed of random number generator;

calc.randresetflag: whether random number generator seed has been reset;

calc.gaussdevreset: whether random number of epsilon has been reset;

calc.ret0vec: intermediate moments for arglist.ret0vec;

calc.ret1vec: intermediate retirement moments for arglist.ret1vec;

calc.ret2vec: intermediate retirement moments for arglist.ret1vec;

calc.returnft: intermediate retirement moments for arglist.returnft;

calc.returnpt: intermediate retirement moments for arglist.returnpt;

calc.newpt: intermediate retirement moments for arglist.newpt;

calc.reversal: intermediate retirement moments for percent of simulated workers who are taking reversal jobs at survey 2 – 6;

calc.everreturnft: intermediate retirement moments for arglist.everreturnft;

calc.everreturnpt: intermediate retirement moments for arglist.everreturnpt;

calc.everreturn: intermediate retirement moments for arglist.everreturn;

moments.ind: structure variable indicating whether the corresponding data moments are observed from the HRS; a total of 43 moments, must equal to gv.nmoments defined above;

moments.ind.age: 13 moments of whether fully retired at ages 54 - 66, 13 X 1;

moments.ind.pr: 5 moments of whether partial retired at ages 55, 58, 60, 62 and 65, 5 X 1;

moments.ind.lowinc: 5 moments of whether fully retired at ages 55, 58, 60, 62 and 65 if in the lower third of lifetime income, 5 X 1;

moments.ind.highinc: 5 moments of whether fully retired at ages 55, 58, 60, 62 and 65 if in the upper third of lifetime income, 5 X 1;

moments.ind.health: 5 moments of whether fully retired at ages 55, 58, 60, 62 and 65 if in poor health, 5 X 1;

moments.ind.healthpr: 5 moments of whether partial retired at ages 55, 58, 60, 62 and 65 if in poor health, 5 X 1;

moments.ind.reversal: 5 moments of whether in reversal job at ages 55, 58, 60, 62 and 65; it is 1 if the husband is working full time at the current age, given they were fully/partially retired in last wave;

moments.dif: structure variable summarizes simulated – observed moments; a total of 43 moments, must equal to gv.nmoments defined above;

moments.dif.age: same as above, only simulated – observed moments;

moments.dif.pr: same as above, only simulated – observed moments;

moments.dif.lowinc: same as above, only simulated – observed moments;

moments.dif.highinc: same as above, only simulated – observed moments;

moments.dif.health: same as above, only simulated – observed moments;

moments.dif.healthpr: same as above, only simulated – observed moments;

moments.dif.reversal: same as above, only simulated – observed moments;

arglist: structure variable captures results of calculations;

arglist.newalpha: indicator of new value of consumption parameter alpha; if it is 1, recalculate time preference factor rho;

arglist.rvec: rho vector; time preference factor (rho) of each household, gv.nobs X 1;

arglist.nmat: indicator matrix, whether the corresponding data moments are observed from the HRS, gv.nobs X gv.nmoments.

arglist.mvec: moments matrix, differences between observed moments and simulated moments, gv.nobs X gv.nmoments;

arglist.wmat: inverse weighting matrix, gv.nobs X gv.nmoments \* (gv.nmoments + 1) / 2;

arglist.ret0mat: percent of simulated workers who retired from career job; or are 1) previously retired, currently in full time job; 2) currently in part time work; and 3) currently retired, by age (50 - 70), gv.nobs X 21;

arglist.ret1mat: percent of simulated workers who are not currently working full time; or are 1) currently in part time work; and 2) currently retired, by age (50 - 70), gv.nobs X 21;

arglist.ret2mat: percent of simulated workers who are currently retired, by age (50 - 70), gv.nobs X 21;

arglist.returnft: percent of simulated workers who are previously retired or in part time work, newly return to full time work, by age (50 - 70), gv.nobs X 21;

arglist.returnpt: percent of simulated workers who are previously retired, newly return to part time work, by age (50 - 70), gv.nobs X 21;

arglist.newpt: percent of simulated workers who are newly enter part time work, by age (50 - 70), gv.nobs X 21;

arglist.everreturnft: percent of simulated workers ever return to full time work after full or partial retirement, gv.nobs X 1;

arglist.everreturnpt: percent of simulated workers ever return to part time work after full retirement, gv.nobs X 1;

arglist.everreturn: percent of simulated workers ever return to full time work after full or partial retirement, or return to part time work after full retirement, gv.nobs X 1;

Extra notes:

The functional form for couples are: sqrt(2) \* (C(t)/sqrt(2)) ^ alpha / alpha;

Spouses do not have options to do part time job or return to full time job after retirement;

people cannot go back to healthy;

full time job includes career job and returned full time job;

part time work = partial retirement;

ret1(age) - ret1(age - 1): increased percent of simulated households who are not in full time job; Reversal jobs may be involved;

ret2(age) - ret2(age - 1): increased percent of simulated households who are retired; Reversal jobs may be involved;

100 - ret0(age): percent of simulated households who are in career job;

ret0(age) - ret1(age): percent of simulated households who previously quit career job but are currently working full time;

ret1(age) - ret2(age): percent of simulated households who are in part time job;

grid size variables could be increased to get more accurate estimates;

variables defined but not used:

data.origobs: order of observation in the original HRS dataset, from 1 to 12685;

data.caseid: hhid of the observation in the original HRS dataset;

data.gender: gender;

data.spretage: husband's age when spouse retires, transfers her DC balance to asset and starts to collect DB/SS benefits if applicable;

data.totalwealth: household’s total wealth in 1992, including housing, not used in calculations;