AMERICAN ECONOMIC REVIEW

Tax-Motivated Trading by Individual Investors[†]

Zoran Ivković
University of Illinois at Urbana-Champaign

James Poterba
MIT and NBER

Scott Weisbenner
University of Illinois at Urbana-Champaign and NBER

MANUSCRIPT 20040061

SUMMARY OF THE DATA AND PROGRAMS EMPLOYED IN THE EMPIRICAL ANALYSES REPORTED IN THE PAPER

March 2005

[†] We thank an anonymous discount broker for providing data on individual investors' trades and Terry Odean for his help in obtaining and understanding the data set.

(I) Data sources

(1) CRSP (Center for Research in Security Prices):

In this paper, we use CRSP to extract security returns.

(2) Propriety data from a discount brokerage house:

The proprietary data consists of around 3,000,000 trades made by around 78,000 households in the period from January 1991 to 1996. Its specifics as they relate to this paper are outlined in the next section. The data may be obtained by contacting Terrance Odean (UC Berkeley, Department of Finance). Signing a non-disclosure agreement is required. The data have been made available to at least several researchers over the past few years.

(II) Basic proprietary data files from the discount brokerage house required for the econometric analyses presented in the paper

(1) Trades file:

The data contain around 3,000,000 trades from January 1991 to November 1996. For the purposes of this paper, the key variables associated with each trade are:

`acctno`	Account identifier
`trade_date`	Trade date
`buy_sell`	Buy (`"B"`) or Sell (`"S"`)
`quantity`	Number of shares
`product_code`	Product code (`"COM"` = common stocks)
`price`	Price
`cusip`	Cusip

(2) Basic household description file:

Summarizes information for around 158,000 accounts, including the identifier for the household the account belongs to (there are around 78,000 households), date on which each account was created, and type of account).

`hh`	Household identifier
`acctno`	Account identifier
`acct_type_1`	Account type (“IR” [IRA] and “KE” [Keogh] tax-deferred; other types taxable)

(III) Preparatory steps: Using the data to generate the sample of trades and link the necessary information

(1) Basic cleanup

(a) Retain only trades in common stocks (`product_code == "COM"`)

(b) Retain only trades from (a) that match with CRSP

(c) Discard trades that stem from short selling
(`buy_sell == "B" & quantity < 0` or `buy_sell == "S" & quantity > 0`;
there are very few such trades—only 0.15% of the total number of trades)

(d) Retain only trades made by households holding both taxable and tax-deferred accounts

(2) Generating the sample of "unambiguous" trades

This stage consists of generating a sample of 414,047 "unambiguous" observations from the basic proprietary data described in Section (II), following the basic cleanup from Section (III, 1). Each observation in this sample consists of a purchase and a potential sale of the same stock in the same account.

(a) The format of the sample of observations (not all the variables are used in the analyses):

`id`	Trade identifier (its index in the initial set of around 3,000,000 trades)
`hh`	Household identifier
`acctno`	Account identifier
`retire`	Retirement account (`0` == taxable; `1` == tax-deferred)
`retire_diff`	Indicator variable describing whether `hh` owns both taxable and tax-deferred accounts (`retire_diff = 1`) or only one kind of account, but not both (`retire_diff = 0`)
`qbuy`	Number of shares purchased
`pbuy`	Stock purchase price
`amount`	Stock purchase amount (`qbuy * pbuy`)
`buydate`	Date of purchase (`yyyymmdd`)
`cusip, permno`	Standard stock identifiers
`sub_sale`	Subsequent sale till `19961130`, the end of sample period (`0` == no; `1` == yes)
`selldate`	Date of sale (`yyyymmdd`; missing if `sub_sale == 0`)
`daytosell`	Number of trading days from purchase to sale (missing if `sub_sale == 0`)
`nsell`	Number of sales of this stock following this purchase till `19961130`
`qsell`	Number of shares sold (missing if `sub_sale == 0`)
`psell`	Stock sale price (missing if `sub_sale == 0`)
`amtsell`	Stock sale amount (`qsell * psell`; missing if `sub_sale == 0`)

(b) Building the set of qualifying observations

Each observation consists of a qualifying purchase and a potential sale. We restrict the sample only to the trades for which the matching between a stock purchase in an account and the potential sale of the stock in the same account can be done unambiguously.

To illustrate the "rules" that govern inclusion into the sample we develop five typical cases. **B** denotes a purchase and **S** denotes a sale of a stock in an account. These five cases generalize readily to more complicated patterns of purchases and sales.

Case 1:	**B, S**	(A buy followed by a sale) ==>	Generates one observation (Buy: **B**, Sale: **S**)
Case 2:	**B**	(A buy followed by no sales) ==>	Generates one observation (Buy: **B**, Sale: **none**)
Case 3:	**B1, B2**	(Two buys followed by no sales) ==>	Generates two observations (Buy: **B1**, Sale: **none**; (Buy: **B2**, Sale: **none**)
Case 4:	**B1, B2, S**	(Two buys followed by a sale) ==>	Generates no observations (**S** cannot be unambiguously linked with the purchase price)
Case 5:	**B, S1, S2**	(A buy followed by two sales) ==>	Generates one observation (Buy: **B**, Sale: **S1**)

- Every qualifying purchase generates an observation. The qualifying purchase may be followed by no sales, one sale, or multiple sales till the end of the sample period (`19961130`) in the same account.
    - If no sales follow the qualifying purchase, `sub_sale` is set to `0`, `nsell` is set to `0`, and other variables associated with the sale are set to `missing`.
    - If one sale follows the qualifying purchase, `sub_sale` is set to `1`, `nsell` is set to `1`, and other variables associated with the sale are filled with the particulars of the sale.
    - If multiple sales follow a qualifying purchase, only the *first* sale is matched with the purchase (93% of sales are such that they completely liquidate the matching purchase), `sub_sale` is set to `1`, `nsell` is set to the number of sales, and other variables associated with the sale are filled with the particulars of the *first* sale.

(IV) Programs and data producing the results reported in the paper

(1) Programs:

All results shown in tables and figures, as well as those mentioned in the text are generated by the main Stata program, `regress_sum_chart_ALL.txt`. The results themselves are displayed in its output file, `regress_sum_chart_ALL.log`.

In the course of its execution, `regress_sum_chart_ALL.txt` invokes the following three Stata programs: `calculate_survival_.txt`, `calculate_survival.txt`, and `program_simulate_cox_tax.txt`.

(2) Data:

The data required by the Stata programs are constructed on the basis of a proprietary data set. Therefore, we cannot provide the data itself. Instead, we provide detailed descriptions of two key data files, buy_sell_return.dta and hazard_all.txt, in this section. The main Stata program `regress_sum_chart_ALL.txt` also calls for four auxiliary data files, `Trades_Original2.dta`, `realize_total_12.dta`, `acct_hh_retire.dta`, and `cusip_permno_01.dta`, the description of which we provide in the Appendix. In all descriptions of the data below, we describe only the variables used in the paper. Also, for each data file we provide a sample observation.

(a) Description of variables from buy_sell_return_all.dta

The set of qualifying observations is the foundation of the data stored in `buy_sell_return_all.dta`. All the variables in it can readily be constructed from the qualifying observations, other information from the proprietary data, and the information contained in the data described in Sections (I-1, I-2, and I-3). In sum, the complete set of variables consists of those described in Section (III-2-a) and those described below:

Purchase and sale-related variables

`sell1 ... sell70`	Indicator variable describing whether the stock was sold during a given month since purchase. For example, `selli = 1` if the stock was sold during the `i`th month since purchase (`i = 1, ... , 70`); if the stock could have been sold, but was not sold, during month `i`, then `selli = 0`. If `selli = 1` (that is, the stock was sold during month `i`), then all subsequent sell variables are set to `missing` (that is, `sellj = missing` for all `j > i`). Finally, `selli = missing` if the covered by `i`th month since purchase goes beyond `19961130`
`r1 ... r71`	Price return variable; `ri` reports the return without dividends over the period of `i` months since the date of purchase (`i = 1, ... , 71`); `ri = missing` if the time period from the date of purchase to `i` months since that date goes beyond `19961130`
`r_bs`	Price return variable; reports the return without dividends from the date of the purchase to the date of sale; `r_bs = missing` if no sale

Variables related to dates

`bdate, b_date, byear, bmonth, bday, yearmonth`	Various ways of expressing the qualifying purchase date `buydate`
`sdate, s_date, syear, smonth, sday`	Various ways of expressing the first sale date `selldate`

Variables related to purchase and sale amounts

`bamount`	`amount` of the qualifying purchase
`samount`	`amount` of the first sale following the qualifying purchase; `samount = missing` if no sale followed the qualifying purchase

Typical observation (please note that it also contains variables not used in this paper):

acctno	1001977	qbuy	30	amount	686
pbuy	22.87	id	4	hh	1001977
retire	0	sub_sale	0	selldate	.
buydate	19920407	psell	.	daytos~l	.
amtsell	.	qsell	.	nsell	0
sell1	0	sell2	0	sell3	0
sell4	0	sell5	0	sell6	0
sell7	0	sell8	0	sell9	0
sell10	0	sell11	0	sell12	0
sell13	0	sell14	0	sell15	0
sell16	0	sell17	0	sell18	0
sell19	0	sell20	0	sell21	0
sell22	0	sell23	0	sell24	0
sell25	0	sell26	0	sell27	0
sell28	0	sell29	0	sell30	0
sell31	0	sell32	0	sell33	0
sell34	0	sell35	0	sell36	0
sell37	0	sell38	0	sell39	0
sell40	0	sell41	0	sell42	0
sell43	0	sell44	0	sell45	0
sell46	0	sell47	0	sell48	0
sell49	0	sell50	0	sell51	0
sell52	0	sell53	0	sell54	0
sell55	0	sell56	.	sell57	.
sell58	.	sell59	.	sell60	.
sell61	.	sell62	.	sell63	.
sell64	.	sell65	.	sell66	.
sell67	.	sell68	.	sell69	.
sell70	.	cusip	02581610	permno	59176
qrat	.	bdate	07apr1992	sdate	.
byear	1992	bmonth	4	bday	7
syear	.	smonth	.	sday	.
b_date	19920407	s_date	.	month_s	.
month_n	55	retire~f	0	sell_91	0
sell_92	0	sell_93	0	sell_94	668
sell_95	515	sell_96	0	r1	-.016304
r2	.0271699	r3	.05435	r4	-.038041
r5	-.043477	r6	-.092391	r7	-.097826
r8	.0163	r9	.0978301	r10	.08696
r11	.11413	r12	.2228301	r13	.23913
r14	.25544	r15	.3478301	r16	.44566
r17	.45653	r18	.55979	r19	.3424
r20	.3587	r21	.27174	r22	.3424
r23	.23914	r24	.29348	r25	.29892
r26	.36375	r27	.2839	r28	.36375
r29	.49275	r30	.4681799	r31	.49889
r32	.4681799	r33	.4558901	r34	.57875
r35	.62789	r36	.73847	r37	.72618
r38	.73847	r39	.7630399	r40	.94119
r41	1.02105	r42	1.15619	r43	1.02719
r44	1.19919	r45	1.0149	r46	1.23605
r47	1.2852	r48	1.48792	r49	1.25449

r50	1.23606	r51	1.13777	r52	1.21149
r53	1.1132	r54	1.33435	r55	1.38349
r56	1.53092	r57	1.70907	r58	2.04693
r59	2.23736	r60	1.92407	r61	2.36022
r62	2.52608	r63	3.07895	r64	3.14651
r65	2.94993	r66	3.2448	r67	2.94686
r68	3.27552	r69	3.27859	r70	3.25709
r71	3.58574	r_bs	.	bamount	686.1
samount	.	sb_rat	.	sellc1	.
sellc2	.	sellc3	.	sellc4	0
sellc5	0	sellc6	0	sellc7	0
sellc8	0	sellc9	0	sellc10	0
sellc11	0	sellc12	0	sellc13	0
sellc14	0	sellc15	0	sellc16	0
sellc17	0	sellc18	0	sellc19	0
sellc20	0	sellc21	0	sellc22	0
sellc23	0	sellc24	0	sellc25	0
sellc26	0	sellc27	0	sellc28	0
sellc29	0	sellc30	0	sellc31	0
sellc32	0	sellc33	0	sellc34	0
sellc35	0	sellc36	0	sellc37	0
sellc38	0	sellc39	0	sellc40	0
sellc41	0	sellc42	0	sellc43	0
sellc44	0	sellc45	0	sellc46	0
sellc47	0	sellc48	0	sellc49	0
sellc50	0	sellc51	0	sellc52	0
sellc53	0	sellc54	0	sellc55	0
sellc56	0	sellc57	0	sellc58	0
sellc59	0	sellc60	.	sellc61	.
sellc62	.	sellc63	.	sellc64	.
sellc65	.	sellc66	.	sellc67	.
sellc68	.	sellc69	.	sellc70	.
sellc71	.	month_sc	.	month_nc	59
rc_bs	.	rc1	.	rc2	.
rc3	.	rc4	-.02174	rc5	-.010869
rc6	.01087	rc7	0	rc8	-.09239
rc9	-.048912	rc10	-.070652	rc11	-.005435
rc12	.08152	rc13	.01087	rc14	.06522
rc15	.17935	rc16	.2173899	rc17	.25
rc18	.40218	rc19	.4184901	rc20	.4130501
rc21	.55979	rc22	.40218	rc23	.36414
rc24	.3424	rc25	.41848	rc26	.27175
rc27	.20653	rc28	.2880501	rc29	.35761
rc30	.26547	rc31	.30232	rc32	.38218
rc33	.49275	rc34	.51732	rc35	.4558901
rc36	.4497499	rc37	.54803	rc38	.65247
rc39	.7139	rc40	.70776	rc41	.74462
rc42	.73233	rc43	.89205	rc44	.98419
rc45	1.18077	rc46	.99647	rc47	1.08862
rc48	1.03333	rc49	1.26062	rc50	1.25448
rc51	1.42649	rc52	1.38349	rc53	1.24834
rc54	1.19306	rc55	1.15006	rc56	1.15006
rc57	1.27292	rc58	1.30977	rc59	1.56778
rc60	1.77664	rc61	2.05307	rc62	2.22507
rc63	1.9425	rc64	2.23121	rc65	2.4155
rc66	2.66122	rc67	3.1158	rc68	2.82093
rc69	3.02365	rc70	2.83322	rc71	2.87623
yearmo~h	199204	capv	10800.8	book_m~t	.9827836
momentum	-.0449217	size1	0	size2	0
size3	0	size4	0	size5	1
bm1	0	bm2	0	bm3	1
bm4	0	bm5	0	mom1	0
mom2	1	mom3	0	mom4	0
mom5	0	size24	0	bm24	1
mom24	1	volati~y	.1076565	vol	.1076565
zip	11230	state_	NY	lat_hh	.7090049
long_hh	-1.29092	contin~l	1	inc	2
job	0	job_sp~e	0	age	34
age_sp~e	76	income	17.5	state_~m	NY
latitude	.7115648	longtude	-1.291082	distance	10.15652
local25	1	local50	1	local100	1
local250	1				

(b) Description of variables from hazard_all.dta

`id`	Trade identifier (its index in the initial set of around 3,000,000 trades)
`hh`	Household identifier
`month_e, month_b`	Number of months since purchase (`month_e`); `month_b = month_e - 1`
`sell_yes`	Indicator variable determining where there was a sale during the current month
`return`	Price return variable; `return` reports the return without dividends over the period from the date of purchase to the beginning of the current month
`dec_yes`	Indicator variable determining whether the current month since purchases encompasses the turn of the year
`gain, loss`	Capital gains/losses since purchase; computed as `gain = max(return, 0)`, `loss = min(0, return)`
`short_term`	Indicator variable determining whether the purchase would qualify for short-term tax treatment (i.e., sold within 12 months since purchase) if it were sold during the current month (`0` == no; `1` == yes)
`retire`	Retirement account (`0` == taxable; `1` == tax-deferred)
`retire_diff`	Indicator variable describing whether `hh` owns both taxable and tax-deferred accounts (`retire_diff = 1`) or only one kind, but not both (`retire_diff = 0`)
`permno`	Standard stock identifier
`b_date` `byear, bmonth`	Purchase date, its year, and month; all obtained from `buydate`
`s_date`	Another name for the first sale date `selldate`
`r_bs`	Price return variable; reports the return without dividends from the date of purchase to the date of sale; `r_bs = missing` if no sale
`bamount, samount`	Purchase amount and the amount of the first sale; alternative names for `amount` and `amtsell`, respectively

Typical observation:

```
        id            4          hh      1001977     month_e            2
  sell_yes            0      return     -.016304     dec_yes            0
      gain            0        loss     -.016304     month_b            1
  short_~m            1      retire            0    sub_sale            0
    permno        59176       byear         1992      bmonth            4
    b_date     19920407      s_date            .    retire~f            0
      r_bs            .     bamount        686.1     samount            .
     size1            0       size5            1         bm1            0
       bm5            0        mom1            0        mom5            0
    size24            0        bm24            1       mom24            1
  volati~y     .1076565         vol     .1076565         inc            2
       age           34    age_sp~e           76      income         17.5
  distance     10.15652     local50            1    local250            1
   age_max           76
```

Appendix

(A) Description of auxiliary data required by `regress_sum_chart_ALL.txt`

(1) `Trades_Original2.dta`

This is a Stata file that results from very minimal processing of the original trades file. Variables relevant for this paper are:

`acctno`	Account identifier
`secno`	Security identifier
`cusip`	CUSIP
`acct_sec`	Unique identifier associated with each `acctno x secno` combination
`buy_sell`	Buy (`"B"`) or Sell (`"S"`)
`year, month, day`	Trade year (`yyyy`), month (`mm`), and day (`dd`); obtained from `trade_date`
`q`	Number of shares; another name for `quantity`
`amount`	Trade amount (`q x p`)
`p`	Price; another name for `price`
`buy`	Buy indicator variable (`buy = 1` if `buy_sell == "B", 0` otherwise)
`sell`	Sell indicator variable (`sell = 1` if `buy_sell == "S", 0` otherwise)
`id`	Trade identifier (its index in the initial set of around 3,000,000 trades)

Typical observation:

```
     acctno      1000706       secno      2203620       cusip     40412010
   acct_sec 100070622036..  buy_sell            B        year         1992
      month            2         day           27           q          100
     amount         2300           p           23         buy            1
       sell            0         b_s            0          id            1
    tr_cnt1            3
```

(2) realize_total_12.dta

This is a Stata file that documents households' total realization patterns through the end of November for each calendar year. The variables used in this paper are:

`hh`	Household identifier
`real_yy`	`hh`'s net capital gains/losses realized through the end of November in year `yy` (`yy = 91, ..., 96`)
`ratio_yy`	Ratio of the total amount of the sales for which the basis is known and the total amount of all sales through the end of November (recall that not all sales can be unambiguously matched with the purchase price)

Typical observation:

```
        hh      1003466    sell_91r         1499    sell_92r            0
  sell_93r        18725    sell_94r        23522    sell_95r     23260.98
  sell_96r        20277     real_91    -363.8388     real_92            0
   real_93     6658.506     real_94    -762.5126     real_95     568.8229
   real_96    -1666.358    gl_lt_91            0    gl_lt_92            0
 gl_lt_93     6658.505    gl_lt_94            0    gl_lt_95            0
 gl_lt_96    -2487.489    gl_st_91    -363.8388    gl_st_92            0
 gl_st_93            0    gl_st_94    -762.5126    gl_st_95      568.823
 gl_st_96    -1236.689     rate_91      st_loss     rate_92         none
  rate_93           lt     rate_94      st_loss     rate_95           st
  rate_96    stlt_loss     sell_91         9319     sell_92            0
  sell_93        18725     sell_94        23525     sell_95        23317
  sell_96        28223    ratio_91     .1608542    ratio_92            1
 ratio_93            1    ratio_94            1    ratio_95            1
 ratio_96     .7184566
```

(3) acct_hh_retire.dta

This is a Stata file that uses the basic household description file to produce the following information about accounts and households:

hh_	Household identifier (hh from the basic household description file)
acctno	Account identifier
ret	Indicator variable describing whether the account is taxable (ret = 0) or tax-deferred (ret = 1); the account type is obtained from acct_type_1
retire_diff	Indicator variable describing whether hh owns both taxable and tax-deferred accounts (retire_diff = 1) or only one kind of account, but not both (retire_diff = 0)

Several typical observations:

	hh_	acctno	ret	retire_~f
1.	1000706	1000706	0	0
2.	1001464	1001464	0	0
3.	1001464	6614640	0	0
4.	1001828	64408183	0	0
5.	1001977	1001977	0	0
6.	100255	100255	1	0
7.	1002686	1002686	0	0
8.	1003466	1003466	0	0
9.	1004317	8885317	0	1
10.	1004317	1004317	0	1
11.	1004317	1092317	1	1
12.	1005486	1005486	0	0

(4) cusip_permno_01.dta

This is a Stata file that provides the mapping between cusip and permno stock identifiers.

Several typical observations:

```
. list

          cusip        permno
    1.   00016510         10015
    2.   00020910         11307
    3.   00035410         10031
    4.   00036010         76868
    5.   00036020         76868
    6.   00036110         54594
```

(B) Stata programs: Code

(1) regress_sum_chart_ALL.txt

```
#delimit ;

set memory 2000m;
set matsize 800;

* sum_buys.txt ;

use /mnt/data2/weisbenn/CG/RAW_DATA/buy_sell_return_all;

count;
keep if retire_diff==1;
count;
count if bamount>=1000;
tab retire if bamount>=1000;
count if bamount>=10000;
tab retire if bamount>=10000;
sum bamount;

gen buy_1=0;
replace buy_1=1 if bamount>=1000;
gen buy_10=0;
replace buy_10=1 if bamount>=10000;

*****************************************************************;
sort byear;
by byear: count;
by byear: sum bamount, detail;
by byear: sum buy_1 buy_10 sub_sale;
by byear: sum buy_10 sub_sale [w=bamount];

count;
sum bamount, detail;
sum buy_1 buy_10 sub_sale;
sum buy_10 sub_sale [w=bamount];
*****************************************************************;
sort byear;
by byear: count if retire==0;
by byear: sum bamount if retire==0, detail;
by byear: sum buy_1 buy_10 sub_sale if retire==0;
by byear: sum buy_10 sub_sale [w=bamount] if retire==0;

count if retire==0;
sum bamount if retire==0, detail;
sum buy_1 buy_10 sub_sale if retire==0;
sum buy_10 sub_sale [w=bamount] if retire==0;
*****************************************************************;
sort byear;
by byear: count if retire==1;
by byear: sum bamount if retire==1, detail;
by byear: sum buy_1 buy_10 sub_sale if retire==1;
by byear: sum buy_10 sub_sale [w=bamount] if retire==1;

count if retire==1;
sum bamount if retire==1, detail;
sum buy_1 buy_10 sub_sale if retire==1;
sum buy_10 sub_sale [w=bamount] if retire==1;
*****************************************************************;

clear;

*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
```

```
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;

* duration_gainloss_chart.txt ;

use /mnt/data2/weisbenn/CG/RAW_DATA/buy_sell_return_all;
count;
drop if bamount==.;

keep if bmonth==1;
keep if retire_diff==1;
count;
gen noretire=1-retire;
tab noretire;
count;

count;
tab sub_sale;
sum month_s r_bs if sub_sale==1;
sum month_s r_bs;
drop rc* sellc*;
sort retire;
by retire: sum sell* r*;


*****************************************************************;
*****************************************************************;
gen ii=-999;
do calculate_survival.txt;

* monthly hazard taxable accounts;
sum ii;
sum hazard*;
sum cpr*;
sum surv;

drop hazard* cpr* pr* surv ii;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
gen ii=0;
do calculate_survival.txt;

*  monthly hazard taxable accounts;
sum ii;
sum hazard*;
sum cpr*;
sum surv;


drop hazard* cpr* pr* surv ii;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
gen ii=0;
do calculate_survival_.txt;

*  monthly hazard taxable accounts;
sum ii;
```

```
sum hazard*;
sum cpr*;
sum surv;

clear;

*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;

* duration_gainloss_chart_10.txt ;

use /mnt/data2/weisbenn/CG/RAW_DATA/buy_sell_return_all;
count;
drop if bamount<10000;
drop if bamount==.;

keep if bmonth==1;
keep if retire_diff==1;
count;
gen noretire=1-retire;
tab noretire;
count;

count;
tab sub_sale;
sum month_s r_bs if sub_sale==1;
sum month_s r_bs;
drop rc* sellc*;
sort retire;
by retire: sum sell* r*;

*********************************************************************;
*********************************************************************;
gen ii=-999;
do calculate_survival.txt;

*  monthly hazard taxable accounts;
sum ii;
sum hazard*;
sum cpr*;
sum surv;

drop hazard* cpr* pr* surv ii;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
gen ii=0;
do calculate_survival.txt;

*  monthly hazard taxable accounts;
sum ii;
sum hazard*;
sum cpr*;
sum surv;

drop hazard* cpr* pr* surv ii;
```

```
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
gen ii=0;
do calculate_survival_.txt;

*  monthly hazard taxable accounts;
sum ii;
sum hazard*;
sum cpr*;
sum surv;

************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;

keep if retire==1;
replace retire=0;

drop hazard* cpr* pr* surv ii;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
gen ii=0;
do calculate_survival.txt;

*  monthly hazard tax-deferred accounts;
sum ii;
sum hazard*;
sum cpr*;
sum surv;

drop hazard* cpr* pr* surv ii;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
gen ii=0;
do calculate_survival_.txt;

* monthly hazard tax-deferred accounts;
sum ii;
sum hazard*;
sum cpr*;
sum surv;

drop hazard* cpr* pr* surv ii;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
************************************************************************;
```

```
clear;

*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;

* regress_chart_10.txt ;

use /mnt/data2/weisbenn/CG/RAW_DATA/buy_sell_return_all;
count;
drop if bamount<10000;
drop if bamount==.;
keep if retire_diff==1;
count;
gen nr=1-retire;
tab nr;
*****************************************************************;
*****************************************************************;
replace sell1=sell1*100;
replace sell2=sell2*100;
replace sell3=sell3*100;
replace sell4=sell4*100;
replace sell5=sell5*100;
replace sell6=sell6*100;
replace sell7=sell7*100;
replace sell8=sell8*100;
replace sell9=sell9*100;
replace sell10=sell10*100;
replace sell11=sell11*100;
replace sell12=sell12*100;
replace sell13=sell13*100;
*****************************************************************;
*****************************************************************;

gen g=0 if r1~=.;
gen l=0 if r1~=.;
replace g=r1 if r1>0 & r1~=.;
replace l=r1 if r1<0 & r1~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell2 g l if retire==0,  robust cluster(hh);
gen c_l2=-.25*_b[l];
gen c_g2=.25*_b[g];
xi: reg sell2 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr2=-.25*_b[l_nr];
gen c_gr2=.25*_b[g_nr];
sum g l;
drop g l g_nr l_nr;

gen g=0 if r2~=.;
gen l=0 if r2~=.;
replace g=r2 if r2>0 & r2~=.;
replace l=r2 if r2<0 & r2~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell3 g l if retire==0,  robust cluster(hh);
gen c_l3=-.25*_b[l];
gen c_g3=.25*_b[g];
xi: reg sell3 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr3=-.25*_b[l_nr];
gen c_gr3=.25*_b[g_nr];
sum g l;
```

```
drop g l g_nr l_nr;

gen g=0 if r3~=.;
gen l=0 if r3~=.;
replace g=r3 if r3>0 & r3~=.;
replace l=r3 if r3<0 & r3~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell4 g l if retire==0,  robust cluster(hh);
gen c_l4=-.25*_b[l];
gen c_g4=.25*_b[g];
xi: reg sell4 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr4=-.25*_b[l_nr];
gen c_gr4=.25*_b[g_nr];
sum g l;
drop g l g_nr l_nr;

gen g=0 if r4~=.;
gen l=0 if r4~=.;
replace g=r4 if r4>0 & r4~=.;
replace l=r4 if r4<0 & r4~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell5 g l if retire==0,  robust cluster(hh);
gen c_l5=-.25*_b[l];
gen c_g5=.25*_b[g];
xi: reg sell5 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr5=-.25*_b[l_nr];
gen c_gr5=.25*_b[g_nr];
sum g l;
drop g l g_nr l_nr;

gen g=0 if r5~=.;
gen l=0 if r5~=.;
replace g=r5 if r5>0 & r5~=.;
replace l=r5 if r5<0 & r5~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell6 g l if retire==0,  robust cluster(hh);
gen c_l6=-.25*_b[l];
gen c_g6=.25*_b[g];
xi: reg sell6 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr6=-.25*_b[l_nr];
gen c_gr6=.25*_b[g_nr];
sum g l;
drop g l g_nr l_nr;

gen g=0 if r6~=.;
gen l=0 if r6~=.;
replace g=r6 if r6>0 & r6~=.;
replace l=r6 if r6<0 & r6~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell7 g l if retire==0,  robust cluster(hh);
gen c_l7=-.25*_b[l];
gen c_g7=.25*_b[g];
xi: reg sell7 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr7=-.25*_b[l_nr];
gen c_gr7=.25*_b[g_nr];
sum g l;
drop g l g_nr l_nr;

gen g=0 if r7~=.;
gen l=0 if r7~=.;
replace g=r7 if r7>0 & r7~=.;
replace l=r7 if r7<0 & r7~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell8 g l if retire==0,  robust cluster(hh);
gen c_l8=-.25*_b[l];
gen c_g8=.25*_b[g];
xi: reg sell8 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr8=-.25*_b[l_nr];
gen c_gr8=.25*_b[g_nr];
sum g l;
```

```
drop g l g_nr l_nr;

gen g=0 if r8~=.;
gen l=0 if r8~=.;
replace g=r8 if r8>0 & r8~=.;
replace l=r8 if r8<0 & r8~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell9 g l if retire==0,  robust cluster(hh);
gen c_l9=-.25*_b[l];
gen c_g9=.25*_b[g];
xi: reg sell9 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr9=-.25*_b[l_nr];
gen c_gr9=.25*_b[g_nr];
sum g l;
drop g l g_nr l_nr;

gen g=0 if r9~=.;
gen l=0 if r9~=.;
replace g=r9 if r9>0 & r9~=.;
replace l=r9 if r9<0 & r9~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell10 g l if retire==0,  robust cluster(hh);
gen c_l10=-.25*_b[l];
gen c_g10=.25*_b[g];
xi: reg sell10 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr10=-.25*_b[l_nr];
gen c_gr10=.25*_b[g_nr];
sum g l;
drop g l g_nr l_nr;

gen g=0 if r10~=.;
gen l=0 if r10~=.;
replace g=r10 if r10>0 & r10~=.;
replace l=r10 if r10<0 & r10~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell11 g l if retire==0,  robust cluster(hh);
gen c_l11=-.25*_b[l];
gen c_g11=.25*_b[g];
xi: reg sell11 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr11=-.25*_b[l_nr];
gen c_gr11=.25*_b[g_nr];
sum g l;
drop g l g_nr l_nr;

gen g=0 if r11~=.;
gen l=0 if r11~=.;
replace g=r11 if r11>0 & r11~=.;
replace l=r11 if r11<0 & r11~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell12 g l if retire==0,  robust cluster(hh);
gen c_l12=-.25*_b[l];
gen c_g12=.25*_b[g];
xi: reg sell12 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr12=-.25*_b[l_nr];
gen c_gr12=.25*_b[g_nr];
sum g l;
drop g l g_nr l_nr;

gen g=0 if r12~=.;
gen l=0 if r12~=.;
replace g=r12 if r12>0 & r12~=.;
replace l=r12 if r12<0 & r12~=.;
gen g_nr=g*nr;
gen l_nr=l*nr;
reg sell13 g l if retire==0,  robust cluster(hh);
gen c_l13=-.25*_b[l];
gen c_g13=.25*_b[g];
xi: reg sell13 nr g l g_nr l_nr,  robust cluster(hh);
gen c_lr13=-.25*_b[l_nr];
gen c_gr13=.25*_b[g_nr];
sum g l;
```

```
drop g l g_nr l_nr;


* first print taxable less tax-deferred effect;
sum c_gr*;
sum c_lr*;
drop c_gr* c_lr*;
* now print effect on taxable accounts;
sum c_g*;
sum c_l*;

drop c_g* c_l*;



*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;

reg sell1 if retire==0,  robust cluster(hh);
xi: reg sell1 nr,  robust cluster(hh);


clear;

*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;

* regress_cox_hh_firm_baseline_10.txt ;

use /mnt/data2/weisbenn/CG/RAW_DATA/hazard_all;
drop b_date s_date samount;
count;
keep if retire_diff==1;
drop retire_diff;
count;
gen noretire=1-retire;
tab noretire;

gen bb=1 if bamount~=.;
replace bb=2 if bamount>=2500 & bamount<5000;
replace bb=3 if bamount>=5000 & bamount<10000;
replace bb=4 if bamount>=10000 & bamount~=.;
tab bb;
tab bb if retire==1;
tab bb if retire==0;

sum month_e month_b;
gen stt=0;
replace stt=1 if month_e<=6;
gen st=short_term;
replace st=0 if stt==1;

tab month_e if stt==1;
tab month_e if st==1;

sum month_e month_b;
replace month_e=month_e-1;
replace month_b=month_b-1;
sum month_e month_b;
```

```
compress;
drop return retire;


keep month_e dec_yes gain loss sell_yes month_b id
noretire hh bb permno;

gen gain_d=gain*dec_yes;
gen loss_d=loss*dec_yes;
gen gain_nr=gain*noretire;
gen loss_nr=loss*noretire;
gen gain_d_nr=gain_d*noretire;
gen loss_d_nr=loss_d*noretire;
gen dec_nr=dec_yes*noretire;
compress;

xi: cox month_e
gain gain_d loss loss_d dec_yes if noretire==1,
dead(sell_yes) t0(month_b) cluster(id);
xi: cox month_e
gain gain_d loss loss_d dec_yes
gain_nr gain_d_nr loss_nr loss_d_nr dec_nr,
dead(sell_yes) t0(month_b) cluster(id) strata(noretire);

xi: cox month_e
gain gain_d loss loss_d dec_yes if noretire==1 & (bb==1 | bb==2),
dead(sell_yes) t0(month_b) cluster(id);
xi: cox month_e
gain gain_d loss loss_d dec_yes
gain_nr gain_d_nr loss_nr loss_d_nr dec_nr if bb==1 | bb==2,
dead(sell_yes) t0(month_b) cluster(id) strata(noretire);

xi: cox month_e
gain gain_d loss loss_d dec_yes if noretire==1 & bb==3,
dead(sell_yes) t0(month_b) cluster(id);
xi: cox month_e
gain gain_d loss loss_d dec_yes
gain_nr gain_d_nr loss_nr loss_d_nr dec_nr if bb==3,
dead(sell_yes) t0(month_b) cluster(id) strata(noretire);

xi: cox month_e
gain gain_d loss loss_d dec_yes if noretire==1 & bb==4,
dead(sell_yes) t0(month_b) cluster(id);
xi: cox month_e
gain gain_d loss loss_d dec_yes
gain_nr gain_d_nr loss_nr loss_d_nr dec_nr if bb==4,
dead(sell_yes) t0(month_b) cluster(id) strata(noretire);

*****************************************************************************************************;
*****************************************************************************************************;
*****************************************************************************************************;
*****************************************************************************************************;

keep if bb==4;

xi: cox month_e
gain gain_d loss loss_d dec_yes if noretire==1,
dead(sell_yes) t0(month_b) cluster(id);
xi: cox month_e
gain gain_d loss loss_d dec_yes
gain_nr gain_d_nr loss_nr loss_d_nr dec_nr,
dead(sell_yes) t0(month_b) cluster(id) strata(noretire);

xi: cox month_e
gain gain_d loss loss_d dec_yes if noretire==1,
dead(sell_yes) t0(month_b) cluster(id) strata(hh);
xi: cox month_e
gain gain_d loss loss_d dec_yes
gain_nr gain_d_nr loss_nr loss_d_nr dec_nr,
dead(sell_yes) t0(month_b) cluster(id) strata(hh noretire);

xi: cox month_e
gain gain_d loss loss_d dec_yes if noretire==1,
dead(sell_yes) t0(month_b) cluster(id) strata(permno);
```

```
xi: cox month_e
gain gain_d loss loss_d dec_yes
gain_nr gain_d_nr loss_nr loss_d_nr dec_nr,
dead(sell_yes) t0(month_b) cluster(id) strata(permno noretire);

xi: cox month_e
gain gain_d loss loss_d dec_yes if noretire==1,
dead(sell_yes) t0(month_b) cluster(id) strata(hh permno);
xi: cox month_e
gain gain_d loss loss_d dec_yes
gain_nr gain_d_nr loss_nr loss_d_nr dec_nr,
dead(sell_yes) t0(month_b) cluster(id) strata(hh permno noretire);

*****************************************************************************************************;
*****************************************************************************************************;
*****************************************************************************************************;
*****************************************************************************************************;

clear;

*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;

use /mnt/data2/weisbenn/CG/RAW_DATA/hazard_all;
drop b_date s_date samount;
count;
keep if retire_diff==1;
drop retire_diff;
count;
gen noretire=1-retire;
tab noretire;

gen bb=1 if bamount~=.;
replace bb=2 if bamount>=2500 & bamount<5000;
replace bb=3 if bamount>=5000 & bamount<10000;
replace bb=4 if bamount>=10000 & bamount~=.;
tab bb;
tab bb if retire==1;
tab bb if retire==0;

sum month_e month_b;
gen stt=0;
replace stt=1 if month_e<=6;
gen st=short_term;
replace st=0 if stt==1;

tab month_e if stt==1;
tab month_e if st==1;

sum month_e month_b;
replace month_e=month_e-1;
replace month_b=month_b-1;
sum month_e month_b;

gen gain_st=gain*st;
gen loss_st=loss*st;
gen gain_stt=gain*stt;
gen loss_stt=loss*stt;
compress;
drop return retire;
```

```
gen dec_st=dec_yes*st;
gen dec_stt=dec_yes*stt;

keep if bb==4;

keep month_e dec_yes gain gain_stt gain_st loss loss_stt loss_st dec_st dec_stt sell_yes month_b
id
noretire hh bb permno;

gen gain_d=gain*dec_yes;
gen loss_d=loss*dec_yes;
gen gain_st_d=gain_st*dec_yes;
gen loss_st_d=loss_st*dec_yes;
gen gain_stt_d=gain_stt*dec_yes;
gen loss_stt_d=loss_stt*dec_yes;
gen gain_d_nr=gain_d*noretire;
gen loss_d_nr=loss_d*noretire;
gen gain_st_d_nr=gain_st_d*noretire;
gen loss_st_d_nr=loss_st_d*noretire;
gen gain_stt_d_nr=gain_stt_d*noretire;
gen loss_stt_d_nr=loss_stt_d*noretire;
gen gain_nr=gain*noretire;
gen loss_nr=loss*noretire;
gen gain_st_nr=gain_st*noretire;
gen loss_st_nr=loss_st*noretire;
gen gain_stt_nr=gain_stt*noretire;
gen loss_stt_nr=loss_stt*noretire;

gen dec_st_nr=dec_st*noretire;
gen dec_stt_nr=dec_stt*noretire;
gen dec_nr=dec_yes*noretire;
compress;

xi: cox month_e
gain gain_stt gain_st
gain_d gain_stt_d gain_st_d
loss loss_stt loss_st
loss_d loss_stt_d loss_st_d
dec_yes dec_stt dec_st if noretire==1,
dead(sell_yes) t0(month_b) cluster(id);

xi: cox month_e
gain gain_stt gain_st
gain_d gain_stt_d gain_st_d
loss loss_stt loss_st
loss_d loss_stt_d loss_st_d
dec_yes dec_stt dec_st
gain_nr gain_stt_nr gain_st_nr
gain_d_nr gain_stt_d_nr gain_st_d_nr
loss_nr loss_stt_nr loss_st_nr
loss_d_nr loss_stt_d_nr loss_st_d_nr
dec_nr dec_stt_nr dec_st_nr,
dead(sell_yes) t0(month_b) cluster(id) strata(noretire);

xi: cox month_e
gain gain_stt gain_st
gain_d gain_stt_d gain_st_d
loss loss_stt loss_st
loss_d loss_stt_d loss_st_d
dec_yes dec_stt dec_st if noretire==1,
dead(sell_yes) t0(month_b) cluster(id) strata(hh);

xi: cox month_e
gain gain_stt gain_st
gain_d gain_stt_d gain_st_d
loss loss_stt loss_st
loss_d loss_stt_d loss_st_d
dec_yes dec_stt dec_st
gain_nr gain_stt_nr gain_st_nr
gain_d_nr gain_stt_d_nr gain_st_d_nr
loss_nr loss_stt_nr loss_st_nr
loss_d_nr loss_stt_d_nr loss_st_d_nr
dec_nr dec_stt_nr dec_st_nr,
dead(sell_yes) t0(month_b) cluster(id) strata(hh noretire);
```

```
xi: cox month_e
gain gain_stt gain_st
gain_d gain_stt_d gain_st_d
loss loss_stt loss_st
loss_d loss_stt_d loss_st_d
dec_yes dec_stt dec_st if noretire==1,
dead(sell_yes) t0(month_b) cluster(id) strata(permno);

xi: cox month_e
gain gain_stt gain_st
gain_d gain_stt_d gain_st_d
loss loss_stt loss_st
loss_d loss_stt_d loss_st_d
dec_yes dec_stt dec_st
gain_nr gain_stt_nr gain_st_nr
gain_d_nr gain_stt_d_nr gain_st_d_nr
loss_nr loss_stt_nr loss_st_nr
loss_d_nr loss_stt_d_nr loss_st_d_nr
dec_nr dec_stt_nr dec_st_nr,
dead(sell_yes) t0(month_b) cluster(id) strata(permno noretire);

clear;

*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;

* regress_cox_realization_10.txt ;

use /mnt/data2/weisbenn/CG/RAW_DATA/hazard_all;
drop b_date s_date samount;
count;
keep if retire_diff==1;
drop retire_diff;
count;
gen noretire=1-retire;
tab noretire;

gen bb=1 if bamount~=.;
replace bb=2 if bamount>=2500 & bamount<5000;
replace bb=3 if bamount>=5000 & bamount<10000;
replace bb=4 if bamount>=10000 & bamount~=.;
tab bb;
tab bb if retire==1;
tab bb if retire==0;

keep if bb==4;
sum bamount, detail;

sum month_e month_b;
gen stt=0;
replace stt=1 if month_e<=6;
gen st=short_term;
replace st=0 if stt==1;

tab month_e if stt==1;
tab month_e if st==1;

sum month_e month_b;
replace month_e=month_e-1;
replace month_b=month_b-1;
sum month_e month_b;
```

```
gen gain_st=gain*st;
gen loss_st=loss*st;
gen gain_stt=gain*stt;
gen loss_stt=loss*stt;
compress;
drop retire;

gen dec_st=dec_yes*st;
gen dec_stt=dec_yes*stt;


tab byear bmonth;
sort byear bmonth;
egen time=group(byear bmonth);
sort byear bmonth;
by byear bmonth: sum byear bmonth time;
tab time;
sum month_e time, detail;
replace time=time+month_e;
sum time, detail;
tab time if dec_yes==1;

keep month_e dec_yes gain gain_stt gain_st loss loss_stt loss_st dec_st dec_stt sell_yes month_b
id noretire hh time st stt return;

keep if dec_yes==1;


**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;

sort hh;
merge hh using /mnt/data2/weisbenn/CG/RAW_DATA/realize_total_12;
tab _merge;
keep if _merge==3 | _merge==1;

replace real_91=0 if _merge==1;
replace real_92=0 if _merge==1;
replace real_93=0 if _merge==1;
replace real_94=0 if _merge==1;
replace real_95=0 if _merge==1;
replace real_96=0 if _merge==1;
replace ratio_91=1 if _merge==1;
replace ratio_92=1 if _merge==1;
replace ratio_93=1 if _merge==1;
replace ratio_94=1 if _merge==1;
replace ratio_95=1 if _merge==1;
replace ratio_96=1 if _merge==1;
drop _merge;

tab noretire;

tab time;

gen real=real_91 if time==12;
replace real=real_92 if time==24;
replace real=real_93 if time==36;
replace real=real_94 if time==48;
replace real=real_95 if time==60;
replace real=real_96 if time==72;

replace real=real/1000;

gen ratio=ratio_91 if time==12;
replace ratio=ratio_92 if time==24;
replace ratio=ratio_93 if time==36;
replace ratio=ratio_94 if time==48;
replace ratio=ratio_95 if time==60;
replace ratio=ratio_96 if time==72;

gen gainc=0 if gain~=.;
replace gainc=1 if return>=0 & gain~=.;
```

```
gen lossc=0 if loss~=.;
replace lossc=1 if loss<0 & loss~=.;

tab gainc;
tab lossc;

gen real_g=real*gainc;
gen real_l=real*lossc;
gen real_g_nr=real_g*noretire;
gen real_l_nr=real_l*noretire;

sum real real_g real_l, detail;

gen mkt1_11=.1740637 if time==12;
replace mkt1_11=.0448931 if time==24;
replace mkt1_11=.0692636 if time==36;
replace mkt1_11=-.0425597 if time==48;
replace mkt1_11=.3076218 if time==60;

gen mkt1_g=mkt1_11*gainc;
gen mkt1_l=mkt1_11*lossc;
gen mkt1_g_nr=mkt1_g*noretire;
gen mkt1_l_nr=mkt1_l*noretire;

**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;

gen gain_nr=gain*noretire;
gen loss_nr=loss*noretire;
gen gain_st_nr=gain_st*noretire;
gen loss_st_nr=loss_st*noretire;
gen gain_stt_nr=gain_stt*noretire;
gen loss_stt_nr=loss_stt*noretire;
gen dec_st_nr=dec_st*noretire;
gen dec_stt_nr=dec_stt*noretire;
gen dec_nr=dec_yes*noretire;

compress;

**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;

xi: cox month_e
gain loss
gain_stt loss_stt
gain_st loss_st
mkt1_g mkt1_l
if noretire==1,
dead(sell_yes) t0(month_b) cluster(id);
test _b[mkt1_g]=_b[mkt1_l];

xi: cox month_e
gain loss
gain_stt loss_stt
gain_st loss_st
gain_nr loss_nr
gain_stt_nr loss_stt_nr
gain_st_nr loss_st_nr
mkt1_g mkt1_l mkt1_g_nr mkt1_l_nr,
dead(sell_yes) t0(month_b) cluster(id) strata(noretire);
test _b[mkt1_g_nr]=_b[mkt1_l_nr];
test _b[mkt1_g]=_b[mkt1_l];

xi: cox month_e
gain loss
gain_stt loss_stt
gain_st loss_st
real_g real_l
if noretire==1 & ratio==1,
dead(sell_yes) t0(month_b) cluster(id);
test _b[real_g]=_b[real_l];
```

```
xi: cox month_e
gain loss
gain_stt loss_stt
gain_st loss_st
gain_nr loss_nr
gain_stt_nr loss_stt_nr
gain_st_nr loss_st_nr
real_g real_l real_g_nr real_l_nr if ratio==1,
dead(sell_yes) t0(month_b) cluster(id) strata(noretire);
test _b[real_g_nr]=_b[real_l_nr];
test _b[real_g]=_b[real_l];

clear;

*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;

* regress_cox_st_1993.txt ;

use /mnt/data2/weisbenn/CG/RAW_DATA/hazard_all;
drop b_date s_date samount;
count;
keep if retire_diff==1;
drop retire_diff;
count;
gen noretire=1-retire;
tab noretire;

gen bb=1 if bamount~=.;
replace bb=2 if bamount>=2500 & bamount<5000;
replace bb=3 if bamount>=5000 & bamount<10000;
replace bb=4 if bamount>=10000 & bamount~=.;
tab bb;
tab bb if retire==1;
tab bb if retire==0;

sum month_e month_b;
gen stt=short_term;
gen st=short_term;

tab month_e if stt==1;
tab month_e if st==1;

sum month_e month_b;
replace month_e=month_e-1;
replace month_b=month_b-1;
sum month_e month_b;

tab byear bmonth;
sort byear bmonth;
egen time=group(byear bmonth);
sort byear bmonth;
by byear bmonth: sum byear bmonth time;
tab time;
sum month_e time, detail;
replace time=time+month_e;
sum time, detail;
tab time if dec_yes==1;

gen s_93=0;
```

```
replace s_93=1 if time>=25 & time~=.;
gen stt_93=0;
replace stt_93=1 if stt==1 & time>=25 & time~=.;
gen st_93=0;
replace st_93=1 if st==1 & time>=25 & time~=.;

tab s_93;
tab stt;
tab stt_93;
tab stt_93 if stt==1;
tab st;
tab st_93;
tab st_93 if st==1;

gen gain_93=gain*s_93;
gen loss_93=loss*s_93;
gen gain_st=gain*st;
gen loss_st=loss*st;
gen gain_stt=gain*stt;
gen loss_stt=loss*stt;
gen gain_st_93=gain*st_93;
gen loss_st_93=loss*st_93;
gen gain_stt_93=gain*stt_93;
gen loss_stt_93=loss*stt_93;

gen gain_d=gain*dec_yes;
gen loss_d=loss*dec_yes;
gen gain_93_d=gain_93*dec_yes;
gen loss_93_d=loss_93*dec_yes;
gen gain_st_d=gain_st*dec_yes;
gen loss_st_d=loss_st*dec_yes;
gen gain_stt_d=gain_stt*dec_yes;
gen loss_stt_d=loss_stt*dec_yes;
gen gain_st_93_d=gain_st_93*dec_yes;
gen loss_st_93_d=loss_st_93*dec_yes;
gen gain_stt_93_d=gain_stt_93*dec_yes;
gen loss_stt_93_d=loss_stt_93*dec_yes;

gen dec_93=dec_yes*s_93;
gen dec_st=dec_yes*st;
gen dec_stt=dec_yes*stt;
gen dec_st_93=dec_yes*st_93;
gen dec_stt_93=dec_yes*stt_93;

compress;

keep month_e dec_yes gain gain_d gain_stt* gain_st* loss loss_d loss_stt* loss_st* dec_st*
dec_stt* gain_93* loss_93* dec_93 s_93
sell_yes month_b id noretire hh bb time;

keep if bb==4;

xi: cox month_e
gain gain_d loss loss_d
gain_st gain_st_d loss_st loss_st_d
gain_93 gain_93_d loss_93 loss_93_d
gain_st_93 gain_st_93_d loss_st_93 loss_st_93_d
dec_yes dec_st dec_93 dec_st_93 s_93 if noretire==1,
dead(sell_yes) t0(month_b) cluster(id);
  test _b[gain]+_b[gain_d]+_b[gain_st]+_b[gain_st_d]=0;
  test
_b[gain]+_b[gain_d]+_b[gain_st]+_b[gain_st_d]+_b[gain_93]+_b[gain_93_d]+_b[gain_st_93]+_b[gain_st
_93_d]=0;
  test _b[gain_93]+_b[gain_93_d]+_b[gain_st_93]+_b[gain_st_93_d]=0;
  test _b[loss]+_b[loss_d]+_b[loss_st]+_b[loss_st_d]=0;
  test
_b[loss]+_b[loss_d]+_b[loss_st]+_b[loss_st_d]+_b[loss_93]+_b[loss_93_d]+_b[loss_st_93]+_b[loss_st
_93_d]=0;
  test _b[loss_93]+_b[loss_93_d]+_b[loss_st_93]+_b[loss_st_93_d]=0;
  test _b[gain]+_b[gain_d]=0;
  test _b[gain]+_b[gain_d]+_b[gain_93]+_b[gain_93_d]=0;
  test _b[gain_93]+_b[gain_93_d]=0;
  test _b[loss]+_b[loss_d]=0;
  test _b[loss]+_b[loss_d]+_b[loss_93]+_b[loss_93_d]=0;
  test _b[loss_93]+_b[loss_93_d]=0;
```

```
  test _b[gain_93]+_b[gain_st_93]=0;
  test _b[gain_93_d]+_b[gain_st_93_d]=0;
  test _b[loss_93]+_b[loss_st_93]=0;
  test _b[loss_93_d]+_b[loss_st_93_d]=0;

clear;

*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;

* wash_buys_10.txt ;

use /mnt/data2/weisbenn/CG/RAW_DATA/Trades_Original2;
sort acctno;
merge acctno using /mnt/data2/weisbenn/CG/RAW_DATA/acct_hh_retire;
tab _merge;
keep if _merge==3;
rename hh_ hh;
rename ret retire;
drop _merge;
sort cusip;
merge cusip using /home/weisbenn/cusip_permno_01;
tab _merge;
keep if _merge==3;
drop _merge;
keep if buy_sell=="B";
sum q p amount, detail;
drop if q<=0;
gen bamount=p*q;
sum bamount, detail;

rename year byear;
rename month bmonth;
rename day bday;
rename p pbuy;
rename q qbuy;

keep hh acctno permno byear bmonth bday bamount pbuy qbuy retire;
gen b_date=mdy(bmonth, bday, byear);
format b_date %d;

compress;
sort hh permno;
save /mnt/data2/weisbenn/CG/RAW_DATA/wash_buys, replace;
sum;
clear;

*****************************************************;
*****************************************************;

use /mnt/data2/weisbenn/CG/RAW_DATA/buy_sell_return_all;

count;
rename retire retirement;
count;
drop if bamount<10000;
count;
keep if sub_sale==1;
drop if s_date>19961099;
drop if r_bs==.;
```

```
sum qrat sb_rat;
replace sb_rat=1 if sb_rat>1 & sb_rat<=1.10;
gen q_rat=sb_rat;
replace q_rat=qrat if sb_rat==0;
replace q_rat=qrat if sb_rat>1 & sb_rat~=.;
replace q_rat=qrat if sb_rat==.;
replace q_rat=1 if q_rat>1 & q_rat~=.;
gen gl=r_bs*bamount*q_rat;

count;
sum q_rat, detail;
count if q_rat>=.95 & q_rat<1.1;
count;

drop sdate;
rename s_date sdate;
keep id hh acctno permno syear smonth sday samount psell qsell retire_diff r_bs retirement sdate;
gen s_date=mdy(smonth, sday, syear);
format s_date %d;
sum;
compress;
sort hh permno;

*****************************************************;
*****************************************************;

joinby hh permno using /mnt/data2/weisbenn/CG/RAW_DATA/wash_buys, unmatched(master);
tab _merge;
sum;
gen date_diff=b_date-s_date;
sum date_diff, detail;
count if date_diff==.;

gen buy1_30=0 if samount~=. & sdate<19961099;
gen buy31_60=0 if samount~=. & sdate<19960999;
gen buy61_90=0 if samount~=. & sdate<19960899;
gen buy91_365=0 if samount~=. & sdate<19951199;
gen buy_never=0 if samount~=. & sdate<19951199;
gen buy1_30_r=0 if samount~=. & sdate<19961099;
gen buy31_60_r=0 if samount~=. & sdate<19960999;
gen buy61_90_r=0 if samount~=. & sdate<19960899;
gen buy91_365_r=0 if samount~=. & sdate<19951199;
gen buy_never_r=0 if samount~=. & sdate<19951199;

replace buy1_30=100 if date_diff>=1 & date_diff<=30 & retire==0;
replace buy31_60=100 if date_diff>=31 & date_diff<=60 & retire==0;
replace buy61_90=100 if date_diff>=61 & date_diff<=90 & retire==0;
replace buy91_365=100 if date_diff>=91 & date_diff<=365 & retire==0;
replace buy_never=100 if buy1_30~=100 & buy31_60~=100 & buy61_90~=100 & buy91_365~=100 &
sdate<19951199;
replace buy1_30_r=100 if date_diff>=1 & date_diff<=30 & retire==1;
replace buy31_60_r=100 if date_diff>=31 & date_diff<=60 & retire==1;
replace buy61_90_r=100 if date_diff>=61 & date_diff<=90 & retire==1;
replace buy91_365_r=100 if date_diff>=91 & date_diff<=365 & retire==1;
replace buy_never_r=100 if buy1_30_r~=100 & buy31_60_r~=100 & buy61_90_r~=100 & buy91_365_r~=100
& sdate<19951199;

keep if _merge==1 | _merge==3;

sum buy1_30 buy31_60 buy61_90 buy1_30_r buy31_60_r buy61_90_r;

egen b1_30=max(buy1_30), by(id);
egen b31_60=max(buy31_60), by(id);
egen b61_90=max(buy61_90), by(id);
egen b91_365=max(buy91_365), by(id);
egen b_never=min(buy_never), by(id);

egen b1_30_r=max(buy1_30_r), by(id);
egen b31_60_r=max(buy31_60_r), by(id);
egen b61_90_r=max(buy61_90_r), by(id);
egen b91_365_r=max(buy91_365_r), by(id);
egen b_never_r=min(buy_never_r), by(id);

gen n=_n;
egen max_n=max(n), by(id);
```

```
count;
keep if max_n==n;
count;
sum b1_30 b31_60 b61_90 b91_365 b_never b1_30_r b31_60_r b61_90_r b91_365_r b_never_r;



gen bs_rat=qbuy/(-qsell);
sum bs_rat, detail;
sum r_bs, detail;


gen gain=0 if r_bs~=.;
gen loss=0 if r_bs~=.;
replace gain=r_bs if r_bs>0 & r_bs~=.;
replace loss=r_bs if r_bs<0;
sum gain loss;

gen loss_dum=0 if r_bs>0;
replace loss_dum=1 if r_bs<0;
sum loss_dum;
sum loss_dum if retirement==0 & retire_diff==1;
sum loss_dum if retirement==1 & retire_diff==1;

count;
count if retire_diff==1;
gen taxable=1-retirement;
tab taxable;
tab retirement;

keep if retire_diff==1;

count;
count if r_bs==0;
tab taxable if r_bs~=0;

gen d=0;
replace d=1 if smonth==12;


*****************************************************************;
*****************************************************************;


reg b1_30 if r_bs<0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
reg b31_60 if r_bs<0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
* reg b61_90 if r_bs<0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
* reg b91_365 if r_bs<0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
* reg b_never if r_bs<0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
sureg (b1_30) (b31_60) if r_bs<0 & retire_diff==1 & retirement==0 & d==1;
test [b1_30]_cons = [b31_60]_cons;


reg b1_30 if r_bs>0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
reg b31_60 if r_bs>0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
* reg b61_90 if r_bs>0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
* reg b91_365 if r_bs>0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
* reg b_never if r_bs>0 & retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
sureg (b1_30) (b31_60) if r_bs>0 & retire_diff==1 & retirement==0 & d==1;
test [b1_30]_cons = [b31_60]_cons;


reg b1_30_r if r_bs<0 & retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
reg b31_60_r if r_bs<0 & retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
* reg b61_90_r if r_bs<0 & retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
* reg b91_365_r if r_bs<0 & retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
* reg b_never_r if r_bs<0 & retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
sureg (b1_30_r) (b31_60_r) if r_bs<0 & retire_diff==1 & retirement==1 & d==1;
test [b1_30_r]_cons = [b31_60_r]_cons;


reg b1_30_r if r_bs>0 & retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
reg b31_60_r if r_bs>0 & retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
* reg b61_90_r if r_bs>0 & retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
```

```
* reg b91_365_r if r_bs>0 & retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
* reg b_never_r if r_bs>0 & retire_dif==1 & retirement==1 & d==1, robust cluster(hh);
sureg (b1_30_r) (b31_60_r) if r_bs>0 & retire_diff==1 & retirement==1 & d==1;
test [b1_30_r]_cons = [b31_60_r]_cons;



*****************************************************************;
*****************************************************************;



reg b1_30 if r_bs<0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
reg b31_60 if r_bs<0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
* reg b61_90 if r_bs<0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
* reg b91_365 if r_bs<0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
* reg b_never if r_bs<0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
sureg (b1_30) (b31_60) if r_bs<0 & retire_diff==1 & retirement==0 & d==0;
test [b1_30]_cons = [b31_60]_cons;


reg b1_30 if r_bs>0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
reg b31_60 if r_bs>0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
* reg b61_90 if r_bs>0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
* reg b91_365 if r_bs>0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
* reg b_never if r_bs>0 & retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
sureg (b1_30) (b31_60) if r_bs>0 & retire_diff==1 & retirement==0 & d==0;
test [b1_30]_cons = [b31_60]_cons;


reg b1_30_r if r_bs<0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
reg b31_60_r if r_bs<0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
* reg b61_90_r if r_bs<0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
* reg b91_365_r if r_bs<0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
* reg b_never_r if r_bs<0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
sureg (b1_30_r) (b31_60_r) if r_bs<0 & retire_diff==1 & retirement==1 & d==0;
test [b1_30_r]_cons = [b31_60_r]_cons;

reg b1_30_r if r_bs>0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
reg b31_60_r if r_bs>0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
* reg b61_90_r if r_bs>0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
* reg b91_365_r if r_bs>0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
* reg b_never_r if r_bs>0 & retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
sureg (b1_30_r) (b31_60_r) if r_bs>0 & retire_diff==1 & retirement==1 & d==0;
test [b1_30_r]_cons = [b31_60_r]_cons;



*****************************************************************;
*****************************************************************;


reg b1_30 loss_dum if retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
reg b31_60 loss_dum if retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
* reg b61_90 loss_dum if retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
* reg b91_365 loss_dum if retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
* reg b_never loss_dum if retire_diff==1 & retirement==0 & d==1, robust cluster(hh);
sureg (b1_30 loss_dum) (b31_60 loss_dum) if retire_diff==1 & retirement==0 & d==1;
test [b1_30]loss_dum = [b31_60]loss_dum;
reg b1_30_r loss_dum if retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
reg b31_60_r loss_dum if retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
* reg b61_90_r loss_dum if retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
* reg b91_365_r loss_dum if retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
* reg b_never_r loss_dum if retire_diff==1 & retirement==1 & d==1, robust cluster(hh);
sureg (b1_30_r loss_dum) (b31_60_r loss_dum) if retire_diff==1 & retirement==1 & d==1;
test [b1_30_r]loss_dum = [b31_60_r]loss_dum;

reg b1_30 loss_dum if retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
reg b31_60 loss_dum if retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
* reg b61_90 loss_dum if retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
* reg b91_365 loss_dum if retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
* reg b_never loss_dum if retire_diff==1 & retirement==0 & d==0, robust cluster(hh);
sureg (b1_30 loss_dum) (b31_60 loss_dum) if retire_diff==1 & retirement==0 & d==0;
test [b1_30]loss_dum = [b31_60]loss_dum;
```

```
reg b1_30_r loss_dum if retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
reg b31_60_r loss_dum if retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
* reg b61_90_r loss_dum if retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
* reg b91_365_r loss_dum if retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
* reg b_never_r loss_dum if retire_diff==1 & retirement==1 & d==0, robust cluster(hh);
sureg (b1_30_r loss_dum) (b31_60_r loss_dum) if retire_diff==1 & retirement==1 & d==0;
test [b1_30_r]loss_dum = [b31_60_r]loss_dum;

*****************************************************************;
*****************************************************************;

reg b1_30 d if r_bs<0 & retire_diff==1 & retirement==0, robust cluster(hh);
reg b31_60 d if r_bs<0 & retire_diff==1 & retirement==0, robust cluster(hh);
* reg b61_90 d if r_bs<0 & retire_diff==1 & retirement==0, robust cluster(hh);
* reg b91_365 d if r_bs<0 & retire_diff==1 & retirement==0, robust cluster(hh);
* reg b_never d if r_bs<0 & retire_diff==1 & retirement==0, robust cluster(hh);
sureg (b1_30 d) (b31_60 d) if r_bs<0 & retire_diff==1 & retirement==0;
test [b1_30]d = [b31_60]d;

reg b1_30 d if r_bs>0 & retire_diff==1 & retirement==0, robust cluster(hh);
reg b31_60 d if r_bs>0 & retire_dif==1 & retirement==0, robust cluster(hh);
* reg b61_90 d if r_bs>0 & retire_diff==1 & retirement==0, robust cluster(hh);
* reg b91_365 d if r_bs>0 & retire_diff==1 & retirement==0, robust cluster(hh);
* reg b_never d if r_bs>0 & retire_diff==1 & retirement==0, robust cluster(hh);
sureg (b1_30 d) (b31_60 d) if r_bs>0 & retire_diff==1 & retirement==0;
test [b1_30]d = [b31_60]d;

reg b1_30_r d if r_bs<0 & retire_diff==1 & retirement==1, robust cluster(hh);
reg b31_60_r d if r_bs<0 & retire_diff==1 & retirement==1, robust cluster(hh);
* reg b61_90_r d if r_bs<0 & retire_diff==1 & retirement==1, robust cluster(hh);
* reg b91_365_r d if r_bs<0 & retire_diff==1 & retirement==1, robust cluster(hh);
* reg b_never_r d if r_bs<0 & retire_diff==1 & retirement==1, robust cluster(hh);
sureg (b1_30_r d) (b31_60_r d) if r_bs<0 & retire_diff==1 & retirement==1;
test [b1_30_r]d = [b31_60_r]d;

reg b1_30_r d if r_bs>0 & retire_diff==1 & retirement==1, robust cluster(hh);
reg b31_60_r d if r_bs>0 & retire_diff==1 & retirement==1, robust cluster(hh);
* reg b61_90_r d if r_bs>0 & retire_diff==1 & retirement==1, robust cluster(hh);
* reg b91_365_r d if r_bs>0 & retire_diff==1 & retirement==1, robust cluster(hh);
* reg b_never_r d if r_bs>0 & retire_diff==1 & retirement==1, robust cluster(hh);
sureg (b1_30_r d) (b31_60_r d) if r_bs>0 & retire_diff==1 & retirement==1;
test [b1_30_r]d = [b31_60_r]d;


*****************************************************************;
*****************************************************************;

gen loss_dum_d=loss_dum*d;

xi: reg b1_30 i.d*loss_dum if retire_diff==1 & retirement==0, robust cluster(hh);
xi: reg b31_60 i.d*loss_dum if retire_diff==1 & retirement==0, robust cluster(hh);
* xi: reg b61_90 i.d*loss_dum if retire_diff==1 & retirement==0, robust cluster(hh);
* xi: reg b91_365 i.d*loss_dum if retire_diff==1 & retirement==0, robust cluster(hh);
* xi: reg b_never i.d*loss_dum if retire_diff==1 & retirement==0, robust cluster(hh);
sureg (b1_30 d loss_dum loss_dum_d) (b31_60 d loss_dum loss_dum_d) if retire_diff==1 &
retirement==0;
test [b1_30]loss_dum_d = [b31_60]loss_dum_d;

xi: reg b1_30_r i.d*loss_dum if retire_diff==1 & retirement==1, robust cluster(hh);
xi: reg b31_60_r i.d*loss_dum if retire_diff==1 & retirement==1, robust cluster(hh);
* xi: reg b61_90_r i.d*loss_dum if retire_diff==1 & retirement==1, robust cluster(hh);
* xi: reg b91_365_r i.d*loss_dum if retire_diff==1 & retirement==1, robust cluster(hh);
* xi: reg b_never_r i.d*loss_dum if retire_diff==1 & retirement==1, robust cluster(hh);
sureg (b1_30_r d loss_dum loss_dum_d) (b31_60_r d loss_dum loss_dum_d) if retire_diff==1 &
retirement==1;
test [b1_30_r]loss_dum_d = [b31_60_r]loss_dum_d;

*****************************************************************;
*****************************************************************;

clear;

*****************************************************************;
*****************************************************************;
*****************************************************************;
```

```
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;
*****************************************************************;

* simulate_cox_tax.txt ;

* Note:  short_term variable is wrong, so drop it;

infile month_e month_b dec_yes short_term retire r0 r25 r50 r75 r_25 r_50 r_75 using
/mnt/data2/weisbenn/CG/RAW_DATA/hazard_simulate.txt;
list if _n==1;
drop if _n==1;
gen n=_n;
gen mult=0 if n<=2400;
replace mult=1 if n>=2401 & n<=4800;
replace mult=2 if n>=4801 & n<=7200;
replace mult=3 if n>=7201 & n<=9600;
replace mult=4 if n>=9601 & n<=12000;
replace mult=5 if n>=12001 & n<=14400;
replace mult=6 if n>=14401 & n<=16800;
replace mult=7 if n>=16801 & n<=19200;
replace mult=8 if n>=19201 & n<=21600;
replace mult=9 if n>=21601 & n<=24000;
tab mult;
drop n;
sum;
replace month_e=(mult*1200)+month_e;
replace month_b=month_e-1;
replace r0=0 if month_e==1;
replace r25=0 if month_e==1;
replace r50=0 if month_e==1;
replace r75=0 if month_e==1;
replace r_25=0 if month_e==1;
replace r_50=0 if month_e==1;
replace r_75=0 if month_e==1;

gen simulate=1;

gen id=0 if retire==0;
replace id=-1 if retire==1;

replace retire=0;

drop short_term;
gen short_term=0;
replace short_term=1 if month_e<=12;

gen return=r50;
gen gain=r50;
gen loss=0;

gen mkt_g=0;
gen mkt_l=0;

gen local50=0 if id==0;
replace local50=1 if id==-1;

gen byear=1991;
gen bmonth=1;
gen retire_diff=1;
gen bamount=50000;

gen paydiv=0;

sum month_b month_e;
```

```
drop if month_e==1;

sort id month_e;
save /mnt/data2/weisbenn/HOLDING/hazard_simulate1, replace;
sum;
sort id;
by id: sum;

clear;

*****************************************************************;
*****************************************************************;
* Note:  short_term variable is wrong, so drop it;

infile month_e month_b dec_yes short_term retire r0 r25 r50 r75 r_25 r_50 r_75 using
/mnt/data2/weisbenn/CG/RAW_DATA/hazard_simulate.txt;
list if _n==1;
drop if _n==1;
gen n=_n;
gen mult=0 if n<=2400;
replace mult=1 if n>=2401 & n<=4800;
replace mult=2 if n>=4801 & n<=7200;
replace mult=3 if n>=7201 & n<=9600;
replace mult=4 if n>=9601 & n<=12000;
replace mult=5 if n>=12001 & n<=14400;
replace mult=6 if n>=14401 & n<=16800;
replace mult=7 if n>=16801 & n<=19200;
replace mult=8 if n>=19201 & n<=21600;
replace mult=9 if n>=21601 & n<=24000;
tab mult;
drop n;
sum;
replace month_e=(mult*1200)+month_e;
replace month_b=month_e-1;
replace r0=0 if month_e==1;
replace r25=0 if month_e==1;
replace r50=0 if month_e==1;
replace r75=0 if month_e==1;
replace r_25=0 if month_e==1;
replace r_50=0 if month_e==1;
replace r_75=0 if month_e==1;

gen simulate=1;

gen id=0 if retire==0;
replace id=-1 if retire==1;

replace retire=0;

drop short_term;
gen short_term=0;
replace short_term=1 if month_e<=12;

gen return=r50;
gen gain=r50;
gen loss=0;

gen mkt_g=0;
gen mkt_l=0;

gen local50=0 if id==0;
replace local50=1 if id==-1;

gen byear=1991;
gen bmonth=1;
gen retire_diff=1;
gen bamount=50000;

gen paydiv=1;
replace id=-2 if id==0;
replace id=-3 if id==-1;

sum month_b month_e;
drop if month_e==1;
```

```
sort id month_e;
save /mnt/data2/weisbenn/HOLDING/hazard_simulate2, replace;
sum;
sort id;
by id: sum;

clear;

*****************************************************************;
*****************************************************************;
* Note:  short_term variable is wrong, so drop it;

infile month_e month_b dec_yes short_term retire r0 r25 r50 r75 r_25 r_50 r_75 using
/mnt/data2/weisbenn/CG/RAW_DATA/hazard_simulate.txt;
list if _n==1;
drop if _n==1;
gen n=_n;
gen mult=0 if n<=2400;
replace mult=1 if n>=2401 & n<=4800;
replace mult=2 if n>=4801 & n<=7200;
replace mult=3 if n>=7201 & n<=9600;
replace mult=4 if n>=9601 & n<=12000;
replace mult=5 if n>=12001 & n<=14400;
replace mult=6 if n>=14401 & n<=16800;
replace mult=7 if n>=16801 & n<=19200;
replace mult=8 if n>=19201 & n<=21600;
replace mult=9 if n>=21601 & n<=24000;
tab mult;
drop n;
sum;
replace month_e=(mult*1200)+month_e;
replace month_b=month_e-1;
replace r0=0 if month_e==1;
replace r25=0 if month_e==1;
replace r50=0 if month_e==1;
replace r75=0 if month_e==1;
replace r_25=0 if month_e==1;
replace r_50=0 if month_e==1;
replace r_75=0 if month_e==1;

gen simulate=1;

gen id=0 if retire==0;
replace id=-1 if retire==1;

replace retire=0;

drop short_term;
gen short_term=0;
replace short_term=1 if month_e<=12;

gen return=r50;
gen gain=r50;
gen loss=0;

gen mkt_g=.25;
gen mkt_l=0;

gen local50=0;

gen byear=1991;
gen bmonth=1;
gen retire_diff=1;
gen bamount=50000;

gen paydiv=0 if id==0;
replace paydiv=1 if id==-1;
replace id=-4 if id==0;
replace id=-5 if id==-1;

sum month_b month_e;
drop if month_e==1;

sort id month_e;
save /mnt/data2/weisbenn/HOLDING/hazard_simulate3, replace;
```

```
sum;
sort id;
by id: sum;

clear;

*****************************************************************;
*****************************************************************;
* Note:  short_term variable is wrong, so drop it;

infile month_e month_b dec_yes short_term retire r0 r25 r50 r75 r_25 r_50 r_75 using
/mnt/data2/weisbenn/CG/RAW_DATA/hazard_simulate.txt;
list if _n==1;
drop if _n==1;
gen n=_n;
gen mult=0 if n<=2400;
replace mult=1 if n>=2401 & n<=4800;
replace mult=2 if n>=4801 & n<=7200;
replace mult=3 if n>=7201 & n<=9600;
replace mult=4 if n>=9601 & n<=12000;
replace mult=5 if n>=12001 & n<=14400;
replace mult=6 if n>=14401 & n<=16800;
replace mult=7 if n>=16801 & n<=19200;
replace mult=8 if n>=19201 & n<=21600;
replace mult=9 if n>=21601 & n<=24000;
tab mult;
drop n;
sum;
replace month_e=(mult*1200)+month_e;
replace month_b=month_e-1;
replace r0=0 if month_e==1;
replace r25=0 if month_e==1;
replace r50=0 if month_e==1;
replace r75=0 if month_e==1;
replace r_25=0 if month_e==1;
replace r_50=0 if month_e==1;
replace r_75=0 if month_e==1;

gen simulate=1;

gen id=0 if retire==0;
replace id=-1 if retire==1;

replace retire=0;

drop short_term;
gen short_term=0;
replace short_term=1 if month_e<=12;

gen return=r50;
gen gain=r50;
gen loss=0;

gen mkt_g=0;
gen mkt_l=-.25;

gen local50=0;

gen byear=1991;
gen bmonth=1;
gen retire_diff=1;
gen bamount=50000;

gen paydiv=0 if id==0;
replace paydiv=1 if id==-1;
replace id=-6 if id==0;
replace id=-7 if id==-1;

sum month_b month_e;
drop if month_e==1;

sort id month_e;
save /mnt/data2/weisbenn/HOLDING/hazard_simulate4, replace;
sum;
sort id;
```

```
by id: sum;

clear;

*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;

use /mnt/data2/weisbenn/CG/RAW_DATA/hazard_all;
drop b_date s_date samount;
count;
keep if retire_diff==1;
drop if bamount<10000;
count;
sum bamount;
tab retire;

gen simulate=0;
append using /mnt/data2/weisbenn/HOLDING/hazard_simulate1;
append using /mnt/data2/weisbenn/HOLDING/hazard_simulate2;
append using /mnt/data2/weisbenn/HOLDING/hazard_simulate3;

sum month_e month_b if simulate==0;
sum month_e month_b if simulate==1;

gen stt=0;
replace stt=1 if month_e<=6;
gen st=short_term;
replace st=0 if stt==1;
compress;

keep month_e month_b sell_yes dec_yes return gain loss stt st id simulate retire;
compress;

sum month_e month_b if simulate==0;
sum month_e month_b if simulate==1;
drop if month_e>80 & month_e~=.;
sum month_e month_b if simulate==0;
sum month_e month_b if simulate==1;
replace month_e=month_e-1;
replace month_b=month_b-1;
sum month_e month_b if simulate==0;
sum month_e month_b if simulate==1;

gen month_ee=month_e;
sum month_ee month_e month_b if simulate==0;
sum month_ee month_e month_b if simulate==1;

tab id if id<=0;

gen r=0 if id==0;
replace r=0.005 if id==-1;
replace r=0.010 if id==-2;
replace r=0.015 if id==-3;
replace r=0.020 if id==-4;
replace r=0.025 if id==-5;

replace return=0 if id==0;
replace return=((1.005)^month_e)-1 if id==-1;
replace return=((1.010)^month_e)-1 if id==-2;
replace return=((1.015)^month_e)-1 if id==-3;
replace return=((1.020)^month_e)-1 if id==-4;
```

```
replace return=((1.025)^month_e)-1 if id==-5;

replace loss=0 if simulate==1;
replace gain=0 if id==0;
replace gain=((1.005)^month_e)-1 if id==-1;
replace gain=((1.010)^month_e)-1 if id==-2;
replace gain=((1.015)^month_e)-1 if id==-3;
replace gain=((1.020)^month_e)-1 if id==-4;
replace gain=((1.025)^month_e)-1 if id==-5;

gen gain_st=gain*st;
gen loss_st=loss*st;
gen gain_stt=gain*stt;
gen loss_stt=loss*stt;
gen dec_st=dec_yes*st;
gen dec_stt=dec_yes*stt;

gen gain_d=gain*dec_yes;
gen loss_d=loss*dec_yes;
gen gain_st_d=gain_st*dec_yes;
gen loss_st_d=loss_st*dec_yes;
gen gain_stt_d=gain_stt*dec_yes;
gen loss_stt_d=loss_stt*dec_yes;

**********************************************************************;
**********************************************************************;

compress;

gen id2=id if id<=0;
sort id2 month_e;
sort id2;
by id2: list gain loss month_e if id<=0 & month_e<=12;

sum if simulate~=1;
sum if simulate==1;


* estimate on non-retirement (taxable) accounts only;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;


**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;

* taxable accounts;

sum month_ee month_e month_b;
sum month_e month_b if stt==1;
sum month_e month_b if st==1;
gen keep=1 if retire==0;

do program_simulate_cox_tax.txt;

**********************************************************************;
**********************************************************************;
**********************************************************************;
**********************************************************************;
```

```
* tax-deferred accounts;

sum month_ee month_e month_b;
sum month_e month_b if stt==1;
sum month_e month_b if st==1;
gen keep=1 if retire==1;

do program_simulate_cox_tax.txt;

*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;

* taxable accounts;

sum month_ee month_e month_b;
sum month_e month_b if stt==1;
sum month_e month_b if st==1;
drop if month_e<=5;
replace month_e=month_e-5;
replace month_b=month_b-5;
sum month_ee month_e month_b;
sum month_e month_b if stt==1;
sum month_e month_b if st==1;
gen keep=1 if retire==0;

do program_simulate_cox_tax.txt;

*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;

* tax-deferred accounts;

sum month_ee month_e month_b;
sum month_e month_b if stt==1;
sum month_e month_b if st==1;
gen keep=1 if retire==1;

do program_simulate_cox_tax.txt;

*********************************************************************;
*********************************************************************;
*********************************************************************;
*********************************************************************;

clear;
exit;
```

(2) calculate_survival_.txt

```
# delimit ;

gen keep=1;
reg sell1 if retire==0 & keep==1,  robust cluster(hh);
gen hazard1=_b[_cons];  gen surv=(1-_b[_cons]);
gen cpr1=1-surv;
drop keep;

gen rr=r1;
gen keep=1 if rr<ii & rr~=.;
reg sell2 if retire==0 & keep==1,  robust cluster(hh);
gen hazard2=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr2=1-surv;
drop rr keep;

gen rr=r2;
gen keep=1 if rr<ii & rr~=.;
reg sell3 if retire==0 & keep==1,  robust cluster(hh);
gen hazard3=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr3=1-surv;
drop rr keep;

gen rr=r3;
gen keep=1 if rr<ii & rr~=.;
reg sell4 if retire==0 & keep==1,  robust cluster(hh);
gen hazard4=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr4=1-surv;
drop rr keep;

gen rr=r4;
gen keep=1 if rr<ii & rr~=.;
reg sell5 if retire==0 & keep==1,  robust cluster(hh);
gen hazard5=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr5=1-surv;
drop rr keep;

gen rr=r5;
gen keep=1 if rr<ii & rr~=.;
reg sell6 if retire==0 & keep==1,  robust cluster(hh);
gen hazard6=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr6=1-surv;
drop rr keep;

gen rr=r6;
gen keep=1 if rr<ii & rr~=.;
reg sell7 if retire==0 & keep==1,  robust cluster(hh);
gen hazard7=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr7=1-surv;
drop rr keep;

gen rr=r7;
gen keep=1 if rr<ii & rr~=.;
reg sell8 if retire==0 & keep==1,  robust cluster(hh);
gen hazard8=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr8=1-surv;
drop rr keep;

gen rr=r8;
gen keep=1 if rr<ii & rr~=.;
reg sell9 if retire==0 & keep==1,  robust cluster(hh);
gen hazard9=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr9=1-surv;
drop rr keep;

gen rr=r9;
gen keep=1 if rr<ii & rr~=.;
reg sell10 if retire==0 & keep==1,  robust cluster(hh);
gen hazard10=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr10=1-surv;
drop rr keep;

;
```

```
;
;
;
gen rr=r10;
gen keep=1 if rr<ii & rr~=.;
reg sell11 if retire==0 & keep==1,  robust cluster(hh);
gen hazard11=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr11=1-surv;
drop rr keep;

gen rr=r11;
gen keep=1 if rr<ii & rr~=.;
reg sell12 if retire==0 & keep==1,  robust cluster(hh);
gen hazard12=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr12=1-surv;
drop rr keep;

gen rr=r12;
gen keep=1 if rr<ii & rr~=.;
reg sell13 if retire==0 & keep==1,  robust cluster(hh);
gen hazard13=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr13=1-surv;
drop rr keep;

gen rr=r13;
gen keep=1 if rr<ii & rr~=.;
reg sell14 if retire==0 & keep==1,  robust cluster(hh);
gen hazard14=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr14=1-surv;
drop rr keep;

gen rr=r14;
gen keep=1 if rr<ii & rr~=.;
reg sell15 if retire==0 & keep==1,  robust cluster(hh);
gen hazard15=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr15=1-surv;
drop rr keep;

gen rr=r15;
gen keep=1 if rr<ii & rr~=.;
reg sell16 if retire==0 & keep==1,  robust cluster(hh);
gen hazard16=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr16=1-surv;
drop rr keep;

gen rr=r16;
gen keep=1 if rr<ii & rr~=.;
reg sell17 if retire==0 & keep==1,  robust cluster(hh);
gen hazard17=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr17=1-surv;
drop rr keep;

gen rr=r17;
gen keep=1 if rr<ii & rr~=.;
reg sell18 if retire==0 & keep==1,  robust cluster(hh);
gen hazard18=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr18=1-surv;
drop rr keep;

gen rr=r18;
gen keep=1 if rr<ii & rr~=.;
reg sell19 if retire==0 & keep==1,  robust cluster(hh);
gen hazard19=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr19=1-surv;
drop rr keep;

gen rr=r19;
gen keep=1 if rr<ii & rr~=.;
reg sell20 if retire==0 & keep==1,  robust cluster(hh);
gen hazard20=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr20=1-surv;
drop rr keep;

;
;
```

```
;
;

gen rr=r20;
gen keep=1 if rr<ii & rr~=.;
reg sell21 if retire==0 & keep==1,  robust cluster(hh);
gen hazard21=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr21=1-surv;
drop rr keep;

gen rr=r21;
gen keep=1 if rr<ii & rr~=.;
reg sell22 if retire==0 & keep==1,  robust cluster(hh);
gen hazard22=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr22=1-surv;
drop rr keep;

gen rr=r22;
gen keep=1 if rr<ii & rr~=.;
reg sell23 if retire==0 & keep==1,  robust cluster(hh);
gen hazard23=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr23=1-surv;
drop rr keep;

gen rr=r23;
gen keep=1 if rr<ii & rr~=.;
reg sell24 if retire==0 & keep==1,  robust cluster(hh);
gen hazard24=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr24=1-surv;
drop rr keep;

gen rr=r24;
gen keep=1 if rr<ii & rr~=.;
reg sell25 if retire==0 & keep==1,  robust cluster(hh);
gen hazard25=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr25=1-surv;
drop rr keep;

gen rr=r25;
gen keep=1 if rr<ii & rr~=.;
reg sell26 if retire==0 & keep==1,  robust cluster(hh);
gen hazard26=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr26=1-surv;
drop rr keep;

gen rr=r26;
gen keep=1 if rr<ii & rr~=.;
reg sell27 if retire==0 & keep==1,  robust cluster(hh);
gen hazard27=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr27=1-surv;
drop rr keep;

gen rr=r27;
gen keep=1 if rr<ii & rr~=.;
reg sell28 if retire==0 & keep==1,  robust cluster(hh);
gen hazard28=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr28=1-surv;
drop rr keep;

gen rr=r28;
gen keep=1 if rr<ii & rr~=.;
reg sell29 if retire==0 & keep==1,  robust cluster(hh);
gen hazard29=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr29=1-surv;
drop rr keep;

gen rr=r29;
gen keep=1 if rr<ii & rr~=.;
reg sell30 if retire==0 & keep==1,  robust cluster(hh);
gen hazard30=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr30=1-surv;
drop rr keep;

;
;
```

```
;
;

gen rr=r30;
gen keep=1 if rr<ii & rr~=.;reg sell31 if retire==0 & keep==1,  robust cluster(hh);
gen hazard31=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr31=1-surv;
drop rr keep;


gen rr=r31;
gen keep=1 if rr<ii & rr~=.;reg sell32 if retire==0 & keep==1,  robust cluster(hh);
gen hazard32=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr32=1-surv;
drop rr keep;

gen rr=r32;
gen keep=1 if rr<ii & rr~=.;
reg sell33 if retire==0 & keep==1,  robust cluster(hh);
gen hazard33=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr33=1-surv;
drop rr keep;

gen rr=r33;
gen keep=1 if rr<ii & rr~=.;
reg sell34 if retire==0 & keep==1,  robust cluster(hh);
gen hazard34=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr34=1-surv;
drop rr keep;

gen rr=r34;
gen keep=1 if rr<ii & rr~=.;
reg sell35 if retire==0 & keep==1,  robust cluster(hh);
gen hazard35=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr35=1-surv;
drop rr keep;

gen rr=r35;
gen keep=1 if rr<ii & rr~=.;
reg sell36 if retire==0 & keep==1,  robust cluster(hh);
gen hazard36=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr36=1-surv;
drop rr keep;

gen rr=r36;
gen keep=1 if rr<ii & rr~=.;
reg sell37 if retire==0 & keep==1,  robust cluster(hh);
gen hazard37=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr37=1-surv;
drop rr keep;

gen rr=r37;
gen keep=1 if rr<ii & rr~=.;
reg sell38 if retire==0 & keep==1,  robust cluster(hh);
gen hazard38=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr38=1-surv;
drop rr keep;

gen rr=r38;
gen keep=1 if rr<ii & rr~=.;
reg sell39 if retire==0 & keep==1,  robust cluster(hh);
gen hazard39=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr39=1-surv;
drop rr keep;

gen rr=r39;
gen keep=1 if rr<ii & rr~=.;
reg sell40 if retire==0 & keep==1,  robust cluster(hh);
gen hazard40=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr40=1-surv;
drop rr keep;

;
;
;
```

```
;

gen rr=r40;
gen keep=1 if rr<ii & rr~=.;
reg sell41 if retire==0 & keep==1,  robust cluster(hh);
gen hazard41=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr41=1-surv;
drop rr keep;

gen rr=r41;
gen keep=1 if rr<ii & rr~=.;
reg sell42 if retire==0 & keep==1,  robust cluster(hh);
gen hazard42=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr42=1-surv;
drop rr keep;

gen rr=r42;
gen keep=1 if rr<ii & rr~=.;
reg sell43 if retire==0 & keep==1,  robust cluster(hh);
gen hazard43=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr43=1-surv;
drop rr keep;

gen rr=r43;
gen keep=1 if rr<ii & rr~=.;
reg sell44 if retire==0 & keep==1,  robust cluster(hh);
gen hazard44=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr44=1-surv;
drop rr keep;

gen rr=r44;
gen keep=1 if rr<ii & rr~=.;
reg sell45 if retire==0 & keep==1,  robust cluster(hh);
gen hazard45=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr45=1-surv;
drop rr keep;

gen rr=r45;
gen keep=1 if rr<ii & rr~=.;
reg sell46 if retire==0 & keep==1,  robust cluster(hh);
gen hazard46=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr46=1-surv;
drop rr keep;

gen rr=r46;
gen keep=1 if rr<ii & rr~=.;
reg sell47 if retire==0 & keep==1,  robust cluster(hh);
gen hazard47=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr47=1-surv;
drop rr keep;

gen rr=r47;
gen keep=1 if rr<ii & rr~=.;
reg sell48 if retire==0 & keep==1,  robust cluster(hh);
gen hazard48=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr48=1-surv;
drop rr keep;

gen rr=r48;
gen keep=1 if rr<ii & rr~=.;
reg sell49 if retire==0 & keep==1,  robust cluster(hh);
gen hazard49=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr49=1-surv;
drop rr keep;

gen rr=r49;
gen keep=1 if rr<ii & rr~=.;
reg sell50 if retire==0 & keep==1,  robust cluster(hh);
gen hazard50=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr50=1-surv;
drop rr keep;

;
;
;
```

```
;

gen rr=r50;
gen keep=1 if rr<ii & rr~=.;
reg sell51 if retire==0 & keep==1,  robust cluster(hh);
gen hazard51=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr51=1-surv;
drop rr keep;

gen rr=r51;
gen keep=1 if rr<ii & rr~=.;
reg sell52 if retire==0 & keep==1,  robust cluster(hh);
gen hazard52=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr52=1-surv;
drop rr keep;

gen rr=r52;
gen keep=1 if rr<ii & rr~=.;
reg sell53 if retire==0 & keep==1,  robust cluster(hh);
gen hazard53=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr53=1-surv;
drop rr keep;

gen rr=r53;
gen keep=1 if rr<ii & rr~=.;
reg sell54 if retire==0 & keep==1,  robust cluster(hh);
gen hazard54=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr54=1-surv;
drop rr keep;

gen rr=r54;
gen keep=1 if rr<ii & rr~=.;
reg sell55 if retire==0 & keep==1,  robust cluster(hh);
gen hazard55=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr55=1-surv;
drop rr keep;

gen rr=r55;
gen keep=1 if rr<ii & rr~=.;
reg sell56 if retire==0 & keep==1,  robust cluster(hh);
gen hazard56=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr56=1-surv;
drop rr keep;

gen rr=r56;
gen keep=1 if rr<ii & rr~=.;
reg sell57 if retire==0 & keep==1,  robust cluster(hh);
gen hazard57=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr57=1-surv;
drop rr keep;

gen rr=r57;
gen keep=1 if rr<ii & rr~=.;
reg sell58 if retire==0 & keep==1,  robust cluster(hh);
gen hazard58=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr58=1-surv;
drop rr keep;

gen rr=r58;
gen keep=1 if rr<ii & rr~=.;
reg sell59 if retire==0 & keep==1,  robust cluster(hh);
gen hazard59=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr59=1-surv;
drop rr keep;

gen rr=r59;
gen keep=1 if rr<ii & rr~=.;
reg sell60 if retire==0 & keep==1,  robust cluster(hh);
gen hazard60=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr60=1-surv;
drop rr keep;


gen pr1=cpr1;
gen pr2=cpr2-cpr1;
```

```
gen pr3=cpr3-cpr2;
gen pr4=cpr4-cpr3;
gen pr5=cpr5-cpr4;
gen pr6=cpr6-cpr5;
gen pr7=cpr7-cpr6;
gen pr8=cpr8-cpr7;
gen pr9=cpr9-cpr8;

gen pr10=cpr10-cpr9;
gen pr11=cpr11-cpr10;
gen pr12=cpr12-cpr11;
gen pr13=cpr13-cpr12;
gen pr14=cpr14-cpr13;
gen pr15=cpr15-cpr14;
gen pr16=cpr16-cpr15;
gen pr17=cpr17-cpr16;
gen pr18=cpr18-cpr17;
gen pr19=cpr19-cpr18;

gen pr20=cpr20-cpr19;
gen pr21=cpr21-cpr20;
gen pr22=cpr22-cpr21;
gen pr23=cpr23-cpr22;
gen pr24=cpr24-cpr23;
gen pr25=cpr25-cpr24;
gen pr26=cpr26-cpr25;
gen pr27=cpr27-cpr26;
gen pr28=cpr28-cpr27;
gen pr29=cpr29-cpr28;

gen pr30=cpr30-cpr29;
gen pr31=cpr31-cpr30;
gen pr32=cpr32-cpr31;
gen pr33=cpr33-cpr32;
gen pr34=cpr34-cpr33;
gen pr35=cpr35-cpr34;
gen pr36=cpr36-cpr35;
gen pr37=cpr37-cpr36;
gen pr38=cpr38-cpr37;
gen pr39=cpr39-cpr38;

gen pr40=cpr40-cpr39;
gen pr41=cpr41-cpr40;
gen pr42=cpr42-cpr41;
gen pr43=cpr43-cpr42;
gen pr44=cpr44-cpr43;
gen pr45=cpr45-cpr44;
gen pr46=cpr46-cpr45;
gen pr47=cpr47-cpr46;
gen pr48=cpr48-cpr47;
gen pr49=cpr49-cpr48;

gen pr50=cpr50-cpr49;
gen pr51=cpr51-cpr50;
gen pr52=cpr52-cpr51;
gen pr53=cpr53-cpr52;
gen pr54=cpr54-cpr53;
gen pr55=cpr55-cpr54;
gen pr56=cpr56-cpr55;
gen pr57=cpr57-cpr56;
gen pr58=cpr58-cpr57;
gen pr59=cpr59-cpr58;
gen pr60=cpr60-cpr59;

;
;



replace hazard1=hazard1*100;
replace hazard2=hazard2*100;
replace hazard3=hazard3*100;
replace hazard4=hazard4*100;
replace hazard5=hazard5*100;
```

```
replace hazard6=hazard6*100;
replace hazard7=hazard7*100;
replace hazard8=hazard8*100;
replace hazard9=hazard9*100;

replace hazard10=hazard10*100;
replace hazard11=hazard11*100;
replace hazard12=hazard12*100;
replace hazard13=hazard13*100;
replace hazard14=hazard14*100;
replace hazard15=hazard15*100;
replace hazard16=hazard16*100;
replace hazard17=hazard17*100;
replace hazard18=hazard18*100;
replace hazard19=hazard19*100;

replace hazard20=hazard20*100;
replace hazard21=hazard21*100;
replace hazard22=hazard22*100;
replace hazard23=hazard23*100;
replace hazard24=hazard24*100;
replace hazard25=hazard25*100;
replace hazard26=hazard26*100;
replace hazard27=hazard27*100;
replace hazard28=hazard28*100;
replace hazard29=hazard29*100;

replace hazard30=hazard30*100;
replace hazard31=hazard31*100;
replace hazard32=hazard32*100;
replace hazard33=hazard33*100;
replace hazard34=hazard34*100;
replace hazard35=hazard35*100;
replace hazard36=hazard36*100;
replace hazard37=hazard37*100;
replace hazard38=hazard38*100;
replace hazard39=hazard39*100;

replace hazard40=hazard40*100;
replace hazard41=hazard41*100;
replace hazard42=hazard42*100;
replace hazard43=hazard43*100;
replace hazard44=hazard44*100;
replace hazard45=hazard45*100;
replace hazard46=hazard46*100;
replace hazard47=hazard47*100;
replace hazard48=hazard48*100;
replace hazard49=hazard49*100;

replace hazard50=hazard50*100;
replace hazard51=hazard51*100;
replace hazard52=hazard52*100;
replace hazard53=hazard53*100;
replace hazard54=hazard54*100;
replace hazard55=hazard55*100;
replace hazard56=hazard56*100;
replace hazard57=hazard57*100;
replace hazard58=hazard58*100;
replace hazard59=hazard59*100;
replace hazard60=hazard60*100;



replace cpr1=cpr1*100;
replace cpr2=cpr2*100;
replace cpr3=cpr3*100;
replace cpr4=cpr4*100;
replace cpr5=cpr5*100;
replace cpr6=cpr6*100;
replace cpr7=cpr7*100;
replace cpr8=cpr8*100;
replace cpr9=cpr9*100;

replace cpr10=cpr10*100;
```

```
replace cpr11=cpr11*100;
replace cpr12=cpr12*100;
replace cpr13=cpr13*100;
replace cpr14=cpr14*100;
replace cpr15=cpr15*100;
replace cpr16=cpr16*100;
replace cpr17=cpr17*100;
replace cpr18=cpr18*100;
replace cpr19=cpr19*100;

replace cpr20=cpr20*100;
replace cpr21=cpr21*100;
replace cpr22=cpr22*100;
replace cpr23=cpr23*100;
replace cpr24=cpr24*100;
replace cpr25=cpr25*100;
replace cpr26=cpr26*100;
replace cpr27=cpr27*100;
replace cpr28=cpr28*100;
replace cpr29=cpr29*100;

replace cpr30=cpr30*100;
replace cpr31=cpr31*100;
replace cpr32=cpr32*100;
replace cpr33=cpr33*100;
replace cpr34=cpr34*100;
replace cpr35=cpr35*100;
replace cpr36=cpr36*100;
replace cpr37=cpr37*100;
replace cpr38=cpr38*100;
replace cpr39=cpr39*100;

replace cpr40=cpr40*100;
replace cpr41=cpr41*100;
replace cpr42=cpr42*100;
replace cpr43=cpr43*100;
replace cpr44=cpr44*100;
replace cpr45=cpr45*100;
replace cpr46=cpr46*100;
replace cpr47=cpr47*100;
replace cpr48=cpr48*100;
replace cpr49=cpr49*100;

replace cpr50=cpr50*100;
replace cpr51=cpr51*100;
replace cpr52=cpr52*100;
replace cpr53=cpr53*100;
replace cpr54=cpr54*100;
replace cpr55=cpr55*100;
replace cpr56=cpr56*100;
replace cpr57=cpr57*100;
replace cpr58=cpr58*100;
replace cpr59=cpr59*100;
replace cpr60=cpr60*100;


replace pr1=pr1*100;
replace pr2=pr2*100;
replace pr3=pr3*100;
replace pr4=pr4*100;
replace pr5=pr5*100;
replace pr6=pr6*100;
replace pr7=pr7*100;
replace pr8=pr8*100;
replace pr9=pr9*100;

replace pr10=pr10*100;
replace pr11=pr11*100;
replace pr12=pr12*100;
replace pr13=pr13*100;
replace pr14=pr14*100;
replace pr15=pr15*100;
replace pr16=pr16*100;
replace pr17=pr17*100;
replace pr18=pr18*100;
```

```
replace pr19=pr19*100;

replace pr20=pr20*100;
replace pr21=pr21*100;
replace pr22=pr22*100;
replace pr23=pr23*100;
replace pr24=pr24*100;
replace pr25=pr25*100;
replace pr26=pr26*100;
replace pr27=pr27*100;
replace pr28=pr28*100;
replace pr29=pr29*100;

replace pr30=pr30*100;
replace pr31=pr31*100;
replace pr32=pr32*100;
replace pr33=pr33*100;
replace pr34=pr34*100;
replace pr35=pr35*100;
replace pr36=pr36*100;
replace pr37=pr37*100;
replace pr38=pr38*100;
replace pr39=pr39*100;

replace pr40=pr40*100;
replace pr41=pr41*100;
replace pr42=pr42*100;
replace pr43=pr43*100;
replace pr44=pr44*100;
replace pr45=pr45*100;
replace pr46=pr46*100;
replace pr47=pr47*100;
replace pr48=pr48*100;
replace pr49=pr49*100;

replace pr50=pr50*100;
replace pr51=pr51*100;
replace pr52=pr52*100;
replace pr53=pr53*100;
replace pr54=pr54*100;
replace pr55=pr55*100;
replace pr56=pr56*100;
replace pr57=pr57*100;
replace pr58=pr58*100;
replace pr59=pr59*100;
replace pr60=pr60*100;

replace surv=surv*100;
```

(3) calculate_survival.txt

```
# delimit ;

gen keep=1;
reg sell1 if retire==0 & keep==1,  robust cluster(hh);
gen hazard1=_b[_cons];  gen surv=(1-_b[_cons]);
gen cpr1=1-surv;
drop keep;

gen rr=r1;
gen keep=1 if rr>ii & rr~=.;
reg sell2 if retire==0 & keep==1,  robust cluster(hh);
gen hazard2=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr2=1-surv;
drop rr keep;

gen rr=r2;
gen keep=1 if rr>ii & rr~=.;
reg sell3 if retire==0 & keep==1,  robust cluster(hh);
gen hazard3=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr3=1-surv;
drop rr keep;

gen rr=r3;
gen keep=1 if rr>ii & rr~=.;
reg sell4 if retire==0 & keep==1,  robust cluster(hh);
gen hazard4=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr4=1-surv;
drop rr keep;

gen rr=r4;
gen keep=1 if rr>ii & rr~=.;
reg sell5 if retire==0 & keep==1,  robust cluster(hh);
gen hazard5=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr5=1-surv;
drop rr keep;

gen rr=r5;
gen keep=1 if rr>ii & rr~=.;
reg sell6 if retire==0 & keep==1,  robust cluster(hh);
gen hazard6=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr6=1-surv;
drop rr keep;

gen rr=r6;
gen keep=1 if rr>ii & rr~=.;
reg sell7 if retire==0 & keep==1,  robust cluster(hh);
gen hazard7=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr7=1-surv;
drop rr keep;

gen rr=r7;
gen keep=1 if rr>ii & rr~=.;
reg sell8 if retire==0 & keep==1,  robust cluster(hh);
gen hazard8=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr8=1-surv;
drop rr keep;

gen rr=r8;
gen keep=1 if rr>ii & rr~=.;
reg sell9 if retire==0 & keep==1,  robust cluster(hh);
gen hazard9=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr9=1-surv;
drop rr keep;

gen rr=r9;
gen keep=1 if rr>ii & rr~=.;
reg sell10 if retire==0 & keep==1,  robust cluster(hh);
gen hazard10=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr10=1-surv;
drop rr keep;

;
```

```
;
;
;
gen rr=r10;
gen keep=1 if rr>ii & rr~=.;
reg sell11 if retire==0 & keep==1,  robust cluster(hh);
gen hazard11=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr11=1-surv;
drop rr keep;

gen rr=r11;
gen keep=1 if rr>ii & rr~=.;
reg sell12 if retire==0 & keep==1,  robust cluster(hh);
gen hazard12=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr12=1-surv;
drop rr keep;

gen rr=r12;
gen keep=1 if rr>ii & rr~=.;
reg sell13 if retire==0 & keep==1,  robust cluster(hh);
gen hazard13=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr13=1-surv;
drop rr keep;

gen rr=r13;
gen keep=1 if rr>ii & rr~=.;
reg sell14 if retire==0 & keep==1,  robust cluster(hh);
gen hazard14=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr14=1-surv;
drop rr keep;

gen rr=r14;
gen keep=1 if rr>ii & rr~=.;
reg sell15 if retire==0 & keep==1,  robust cluster(hh);
gen hazard15=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr15=1-surv;
drop rr keep;

gen rr=r15;
gen keep=1 if rr>ii & rr~=.;
reg sell16 if retire==0 & keep==1,  robust cluster(hh);
gen hazard16=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr16=1-surv;
drop rr keep;

gen rr=r16;
gen keep=1 if rr>ii & rr~=.;
reg sell17 if retire==0 & keep==1,  robust cluster(hh);
gen hazard17=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr17=1-surv;
drop rr keep;

gen rr=r17;
gen keep=1 if rr>ii & rr~=.;
reg sell18 if retire==0 & keep==1,  robust cluster(hh);
gen hazard18=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr18=1-surv;
drop rr keep;

gen rr=r18;
gen keep=1 if rr>ii & rr~=.;
reg sell19 if retire==0 & keep==1,  robust cluster(hh);
gen hazard19=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr19=1-surv;
drop rr keep;

gen rr=r19;
gen keep=1 if rr>ii & rr~=.;
reg sell20 if retire==0 & keep==1,  robust cluster(hh);
gen hazard20=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr20=1-surv;
drop rr keep;

;
;
```

```
;
;

gen rr=r20;
gen keep=1 if rr>ii & rr~=.;
reg sell21 if retire==0 & keep==1,  robust cluster(hh);
gen hazard21=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr21=1-surv;
drop rr keep;

gen rr=r21;
gen keep=1 if rr>ii & rr~=.;
reg sell22 if retire==0 & keep==1,  robust cluster(hh);
gen hazard22=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr22=1-surv;
drop rr keep;

gen rr=r22;
gen keep=1 if rr>ii & rr~=.;
reg sell23 if retire==0 & keep==1,  robust cluster(hh);
gen hazard23=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr23=1-surv;
drop rr keep;

gen rr=r23;
gen keep=1 if rr>ii & rr~=.;
reg sell24 if retire==0 & keep==1,  robust cluster(hh);
gen hazard24=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr24=1-surv;
drop rr keep;

gen rr=r24;
gen keep=1 if rr>ii & rr~=.;
reg sell25 if retire==0 & keep==1,  robust cluster(hh);
gen hazard25=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr25=1-surv;
drop rr keep;

gen rr=r25;
gen keep=1 if rr>ii & rr~=.;
reg sell26 if retire==0 & keep==1,  robust cluster(hh);
gen hazard26=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr26=1-surv;
drop rr keep;

gen rr=r26;
gen keep=1 if rr>ii & rr~=.;
reg sell27 if retire==0 & keep==1,  robust cluster(hh);
gen hazard27=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr27=1-surv;
drop rr keep;

gen rr=r27;
gen keep=1 if rr>ii & rr~=.;
reg sell28 if retire==0 & keep==1,  robust cluster(hh);
gen hazard28=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr28=1-surv;
drop rr keep;

gen rr=r28;
gen keep=1 if rr>ii & rr~=.;
reg sell29 if retire==0 & keep==1,  robust cluster(hh);
gen hazard29=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr29=1-surv;
drop rr keep;

gen rr=r29;
gen keep=1 if rr>ii & rr~=.;
reg sell30 if retire==0 & keep==1,  robust cluster(hh);
gen hazard30=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr30=1-surv;
drop rr keep;

;
;
```

```
;
;

gen rr=r30;
gen keep=1 if rr>ii & rr~=.;reg sell31 if retire==0 & keep==1,  robust cluster(hh);
gen hazard31=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr31=1-surv;
drop rr keep;


gen rr=r31;
gen keep=1 if rr>ii & rr~=.;reg sell32 if retire==0 & keep==1,  robust cluster(hh);
gen hazard32=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr32=1-surv;
drop rr keep;

gen rr=r32;
gen keep=1 if rr>ii & rr~=.;
reg sell33 if retire==0 & keep==1,  robust cluster(hh);
gen hazard33=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr33=1-surv;
drop rr keep;

gen rr=r33;
gen keep=1 if rr>ii & rr~=.;
reg sell34 if retire==0 & keep==1,  robust cluster(hh);
gen hazard34=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr34=1-surv;
drop rr keep;

gen rr=r34;
gen keep=1 if rr>ii & rr~=.;
reg sell35 if retire==0 & keep==1,  robust cluster(hh);
gen hazard35=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr35=1-surv;
drop rr keep;

gen rr=r35;
gen keep=1 if rr>ii & rr~=.;
reg sell36 if retire==0 & keep==1,  robust cluster(hh);
gen hazard36=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr36=1-surv;
drop rr keep;

gen rr=r36;
gen keep=1 if rr>ii & rr~=.;
reg sell37 if retire==0 & keep==1,  robust cluster(hh);
gen hazard37=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr37=1-surv;
drop rr keep;

gen rr=r37;
gen keep=1 if rr>ii & rr~=.;
reg sell38 if retire==0 & keep==1,  robust cluster(hh);
gen hazard38=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr38=1-surv;
drop rr keep;

gen rr=r38;
gen keep=1 if rr>ii & rr~=.;
reg sell39 if retire==0 & keep==1,  robust cluster(hh);
gen hazard39=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr39=1-surv;
drop rr keep;

gen rr=r39;
gen keep=1 if rr>ii & rr~=.;
reg sell40 if retire==0 & keep==1,  robust cluster(hh);
gen hazard40=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr40=1-surv;
drop rr keep;

;
;
;
```

```
;

gen rr=r40;
gen keep=1 if rr>ii & rr~=.;
reg sell41 if retire==0 & keep==1,  robust cluster(hh);
gen hazard41=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr41=1-surv;
drop rr keep;

gen rr=r41;
gen keep=1 if rr>ii & rr~=.;
reg sell42 if retire==0 & keep==1,  robust cluster(hh);
gen hazard42=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr42=1-surv;
drop rr keep;

gen rr=r42;
gen keep=1 if rr>ii & rr~=.;
reg sell43 if retire==0 & keep==1,  robust cluster(hh);
gen hazard43=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr43=1-surv;
drop rr keep;

gen rr=r43;
gen keep=1 if rr>ii & rr~=.;
reg sell44 if retire==0 & keep==1,  robust cluster(hh);
gen hazard44=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr44=1-surv;
drop rr keep;

gen rr=r44;
gen keep=1 if rr>ii & rr~=.;
reg sell45 if retire==0 & keep==1,  robust cluster(hh);
gen hazard45=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr45=1-surv;
drop rr keep;

gen rr=r45;
gen keep=1 if rr>ii & rr~=.;
reg sell46 if retire==0 & keep==1,  robust cluster(hh);
gen hazard46=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr46=1-surv;
drop rr keep;

gen rr=r46;
gen keep=1 if rr>ii & rr~=.;
reg sell47 if retire==0 & keep==1,  robust cluster(hh);
gen hazard47=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr47=1-surv;
drop rr keep;

gen rr=r47;
gen keep=1 if rr>ii & rr~=.;
reg sell48 if retire==0 & keep==1,  robust cluster(hh);
gen hazard48=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr48=1-surv;
drop rr keep;

gen rr=r48;
gen keep=1 if rr>ii & rr~=.;
reg sell49 if retire==0 & keep==1,  robust cluster(hh);
gen hazard49=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr49=1-surv;
drop rr keep;

gen rr=r49;
gen keep=1 if rr>ii & rr~=.;
reg sell50 if retire==0 & keep==1,  robust cluster(hh);
gen hazard50=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr50=1-surv;
drop rr keep;

;
;
;
```

```
;

gen rr=r50;
gen keep=1 if rr>ii & rr~=.;
reg sell51 if retire==0 & keep==1,  robust cluster(hh);
gen hazard51=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr51=1-surv;
drop rr keep;

gen rr=r51;
gen keep=1 if rr>ii & rr~=.;
reg sell52 if retire==0 & keep==1,  robust cluster(hh);
gen hazard52=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr52=1-surv;
drop rr keep;

gen rr=r52;
gen keep=1 if rr>ii & rr~=.;
reg sell53 if retire==0 & keep==1,  robust cluster(hh);
gen hazard53=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr53=1-surv;
drop rr keep;

gen rr=r53;
gen keep=1 if rr>ii & rr~=.;
reg sell54 if retire==0 & keep==1,  robust cluster(hh);
gen hazard54=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr54=1-surv;
drop rr keep;

gen rr=r54;
gen keep=1 if rr>ii & rr~=.;
reg sell55 if retire==0 & keep==1,  robust cluster(hh);
gen hazard55=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr55=1-surv;
drop rr keep;

gen rr=r55;
gen keep=1 if rr>ii & rr~=.;
reg sell56 if retire==0 & keep==1,  robust cluster(hh);
gen hazard56=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr56=1-surv;
drop rr keep;

gen rr=r56;
gen keep=1 if rr>ii & rr~=.;
reg sell57 if retire==0 & keep==1,  robust cluster(hh);
gen hazard57=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr57=1-surv;
drop rr keep;

gen rr=r57;
gen keep=1 if rr>ii & rr~=.;
reg sell58 if retire==0 & keep==1,  robust cluster(hh);
gen hazard58=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr58=1-surv;
drop rr keep;

gen rr=r58;
gen keep=1 if rr>ii & rr~=.;
reg sell59 if retire==0 & keep==1,  robust cluster(hh);
gen hazard59=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr59=1-surv;
drop rr keep;

gen rr=r59;
gen keep=1 if rr>ii & rr~=.;
reg sell60 if retire==0 & keep==1,  robust cluster(hh);
gen hazard60=_b[_cons];  replace surv=surv*(1-_b[_cons]);
gen cpr60=1-surv;
drop rr keep;


gen pr1=cpr1;
gen pr2=cpr2-cpr1;
```

```
gen pr3=cpr3-cpr2;
gen pr4=cpr4-cpr3;
gen pr5=cpr5-cpr4;
gen pr6=cpr6-cpr5;
gen pr7=cpr7-cpr6;
gen pr8=cpr8-cpr7;
gen pr9=cpr9-cpr8;

gen pr10=cpr10-cpr9;
gen pr11=cpr11-cpr10;
gen pr12=cpr12-cpr11;
gen pr13=cpr13-cpr12;
gen pr14=cpr14-cpr13;
gen pr15=cpr15-cpr14;
gen pr16=cpr16-cpr15;
gen pr17=cpr17-cpr16;
gen pr18=cpr18-cpr17;
gen pr19=cpr19-cpr18;

gen pr20=cpr20-cpr19;
gen pr21=cpr21-cpr20;
gen pr22=cpr22-cpr21;
gen pr23=cpr23-cpr22;
gen pr24=cpr24-cpr23;
gen pr25=cpr25-cpr24;
gen pr26=cpr26-cpr25;
gen pr27=cpr27-cpr26;
gen pr28=cpr28-cpr27;
gen pr29=cpr29-cpr28;

gen pr30=cpr30-cpr29;
gen pr31=cpr31-cpr30;
gen pr32=cpr32-cpr31;
gen pr33=cpr33-cpr32;
gen pr34=cpr34-cpr33;
gen pr35=cpr35-cpr34;
gen pr36=cpr36-cpr35;
gen pr37=cpr37-cpr36;
gen pr38=cpr38-cpr37;
gen pr39=cpr39-cpr38;

gen pr40=cpr40-cpr39;
gen pr41=cpr41-cpr40;
gen pr42=cpr42-cpr41;
gen pr43=cpr43-cpr42;
gen pr44=cpr44-cpr43;
gen pr45=cpr45-cpr44;
gen pr46=cpr46-cpr45;
gen pr47=cpr47-cpr46;
gen pr48=cpr48-cpr47;
gen pr49=cpr49-cpr48;

gen pr50=cpr50-cpr49;
gen pr51=cpr51-cpr50;
gen pr52=cpr52-cpr51;
gen pr53=cpr53-cpr52;
gen pr54=cpr54-cpr53;
gen pr55=cpr55-cpr54;
gen pr56=cpr56-cpr55;
gen pr57=cpr57-cpr56;
gen pr58=cpr58-cpr57;
gen pr59=cpr59-cpr58;
gen pr60=cpr60-cpr59;

;
;



replace hazard1=hazard1*100;
replace hazard2=hazard2*100;
replace hazard3=hazard3*100;
replace hazard4=hazard4*100;
replace hazard5=hazard5*100;
replace hazard6=hazard6*100;
```

```
replace hazard7=hazard7*100;
replace hazard8=hazard8*100;
replace hazard9=hazard9*100;

replace hazard10=hazard10*100;
replace hazard11=hazard11*100;
replace hazard12=hazard12*100;
replace hazard13=hazard13*100;
replace hazard14=hazard14*100;
replace hazard15=hazard15*100;
replace hazard16=hazard16*100;
replace hazard17=hazard17*100;
replace hazard18=hazard18*100;
replace hazard19=hazard19*100;

replace hazard20=hazard20*100;
replace hazard21=hazard21*100;
replace hazard22=hazard22*100;
replace hazard23=hazard23*100;
replace hazard24=hazard24*100;
replace hazard25=hazard25*100;
replace hazard26=hazard26*100;
replace hazard27=hazard27*100;
replace hazard28=hazard28*100;
replace hazard29=hazard29*100;

replace hazard30=hazard30*100;
replace hazard31=hazard31*100;
replace hazard32=hazard32*100;
replace hazard33=hazard33*100;
replace hazard34=hazard34*100;
replace hazard35=hazard35*100;
replace hazard36=hazard36*100;
replace hazard37=hazard37*100;
replace hazard38=hazard38*100;
replace hazard39=hazard39*100;

replace hazard40=hazard40*100;
replace hazard41=hazard41*100;
replace hazard42=hazard42*100;
replace hazard43=hazard43*100;
replace hazard44=hazard44*100;
replace hazard45=hazard45*100;
replace hazard46=hazard46*100;
replace hazard47=hazard47*100;
replace hazard48=hazard48*100;
replace hazard49=hazard49*100;

replace hazard50=hazard50*100;
replace hazard51=hazard51*100;
replace hazard52=hazard52*100;
replace hazard53=hazard53*100;
replace hazard54=hazard54*100;
replace hazard55=hazard55*100;
replace hazard56=hazard56*100;
replace hazard57=hazard57*100;
replace hazard58=hazard58*100;
replace hazard59=hazard59*100;
replace hazard60=hazard60*100;



replace cpr1=cpr1*100;
replace cpr2=cpr2*100;
replace cpr3=cpr3*100;
replace cpr4=cpr4*100;
replace cpr5=cpr5*100;
replace cpr6=cpr6*100;
replace cpr7=cpr7*100;
replace cpr8=cpr8*100;
replace cpr9=cpr9*100;

replace cpr10=cpr10*100;
replace cpr11=cpr11*100;
```

```
replace cpr12=cpr12*100;
replace cpr13=cpr13*100;
replace cpr14=cpr14*100;
replace cpr15=cpr15*100;
replace cpr16=cpr16*100;
replace cpr17=cpr17*100;
replace cpr18=cpr18*100;
replace cpr19=cpr19*100;

replace cpr20=cpr20*100;
replace cpr21=cpr21*100;
replace cpr22=cpr22*100;
replace cpr23=cpr23*100;
replace cpr24=cpr24*100;
replace cpr25=cpr25*100;
replace cpr26=cpr26*100;
replace cpr27=cpr27*100;
replace cpr28=cpr28*100;
replace cpr29=cpr29*100;

replace cpr30=cpr30*100;
replace cpr31=cpr31*100;
replace cpr32=cpr32*100;
replace cpr33=cpr33*100;
replace cpr34=cpr34*100;
replace cpr35=cpr35*100;
replace cpr36=cpr36*100;
replace cpr37=cpr37*100;
replace cpr38=cpr38*100;
replace cpr39=cpr39*100;

replace cpr40=cpr40*100;
replace cpr41=cpr41*100;
replace cpr42=cpr42*100;
replace cpr43=cpr43*100;
replace cpr44=cpr44*100;
replace cpr45=cpr45*100;
replace cpr46=cpr46*100;
replace cpr47=cpr47*100;
replace cpr48=cpr48*100;
replace cpr49=cpr49*100;

replace cpr50=cpr50*100;
replace cpr51=cpr51*100;
replace cpr52=cpr52*100;
replace cpr53=cpr53*100;
replace cpr54=cpr54*100;
replace cpr55=cpr55*100;
replace cpr56=cpr56*100;
replace cpr57=cpr57*100;
replace cpr58=cpr58*100;
replace cpr59=cpr59*100;
replace cpr60=cpr60*100;


replace pr1=pr1*100;
replace pr2=pr2*100;
replace pr3=pr3*100;
replace pr4=pr4*100;
replace pr5=pr5*100;
replace pr6=pr6*100;
replace pr7=pr7*100;
replace pr8=pr8*100;
replace pr9=pr9*100;

replace pr10=pr10*100;
replace pr11=pr11*100;
replace pr12=pr12*100;
replace pr13=pr13*100;
replace pr14=pr14*100;
replace pr15=pr15*100;
replace pr16=pr16*100;
replace pr17=pr17*100;
replace pr18=pr18*100;
replace pr19=pr19*100;
```

```
replace pr20=pr20*100;
replace pr21=pr21*100;
replace pr22=pr22*100;
replace pr23=pr23*100;
replace pr24=pr24*100;
replace pr25=pr25*100;
replace pr26=pr26*100;
replace pr27=pr27*100;
replace pr28=pr28*100;
replace pr29=pr29*100;

replace pr30=pr30*100;
replace pr31=pr31*100;
replace pr32=pr32*100;
replace pr33=pr33*100;
replace pr34=pr34*100;
replace pr35=pr35*100;
replace pr36=pr36*100;
replace pr37=pr37*100;
replace pr38=pr38*100;
replace pr39=pr39*100;

replace pr40=pr40*100;
replace pr41=pr41*100;
replace pr42=pr42*100;
replace pr43=pr43*100;
replace pr44=pr44*100;
replace pr45=pr45*100;
replace pr46=pr46*100;
replace pr47=pr47*100;
replace pr48=pr48*100;
replace pr49=pr49*100;

replace pr50=pr50*100;
replace pr51=pr51*100;
replace pr52=pr52*100;
replace pr53=pr53*100;
replace pr54=pr54*100;
replace pr55=pr55*100;
replace pr56=pr56*100;
replace pr57=pr57*100;
replace pr58=pr58*100;
replace pr59=pr59*100;
replace pr60=pr60*100;

replace surv=surv*100;
```

(4) program_simulate_cox_tax.txt

```
#delimit ;

xi: cox month_e
gain gain_d loss loss_d
gain_stt gain_stt_d loss_stt loss_stt_d
gain_st gain_st_d loss_st loss_st_d
dec_yes dec_st dec_stt
if simulate==0 & keep==1,
dead(sell_yes) t0(month_b) cluster(id) nolog basehc(ha);
predict exb if simulate==1 & month_e<=60, hr;

egen haz=mean(ha), by(month_e);
sort month_e;
by month_e: sum ha haz if month_e<=75;

gen hazard=exb*haz if simulate==1 & month_e<=60;
replace hazard=0 if haz==. & simulate==1 & month_e<=60;
sort month_e;
by month_e: sum ha haz hazard if simulate==1 & month_e<=60;
drop ha haz exb;

sort id month_e;
gen chazard=hazard if simulate==1 & month_e<=60;
replace chazard=chazard[_n-1]+(1-chazard[_n-1])*hazard if month_e>1
& id==id[_n-1] & month_e==month_e[_n-1]+1 & simulate==1 & month_e<=60;

sort id month_e;
gen hazard_=hazard if simulate==1 & month_e<=60;
replace hazard_=(chazard-chazard[_n-1]) if month_e>1
& id==id[_n-1] & month_e==month_e[_n-1]+1 & simulate==1 & month_e<=60;

sort id month_e;
gen expect=month_e*hazard_ if simulate==1 & month_e<=60;
replace expect=expect[_n-1]+month_e*hazard_ if month_e>1
& id==id[_n-1] & month_e==month_e[_n-1]+1 & simulate==1 & month_e<=60;

egen ch=max(chazard) if month_e<=61, by(id);
replace expect=expect/ch if month_e<=60;

sort id2;
by id2: sum ch chazard if id<=0;
by id2: sum chazard if id<=0;
gen expect5=( (expect)*ch + ((1-ch)*60) ) if month_e<=60;
gen expect20=( (expect)*ch + ((1-ch)*240) ) if month_e<=60;

gen tt=.40 if month_ee<=11 & month_e<=60;
replace tt=.28 if month_ee>=12 & month_ee~=. & month_e<=60;
gen A=(return)*(1-tt)+1 if month_e<=60;
replace A=ln(A)/month_ee if month_e<=60;
gen ttt= 1 - ((exp(A)-1)/r) if month_e<=60;
sort id month_e;
gen tax=hazard_*ttt if simulate==1 & month_e<=60;
replace tax=tax[_n-1]+hazard_*ttt if month_e>1
& id==id[_n-1] & month_e==month_e[_n-1]+1 & simulate==1 & month_e<=60;
replace tax=tax/ch if month_e<=60;

gen r5=((1+r)^60)-1 if month_e<=61;
gen A5=(r5)*(0.72)+1 if month_e<=61;
replace A5=ln(A5)/60 if month_e<=61;
gen t5= 1 - ((exp(A5)-1)/r) if month_e<=61;

gen r20=((1+r)^240)-1 if month_e<=61;
gen A20=(r20)*(0.72)+1 if month_e<=61;
replace A20=ln(A20)/240 if month_e<=61;
gen t20= 1 - ((exp(A20)-1)/r) if month_e<=61;

gen tax100=( tax*ch + ((1-ch)*0) ) if month_e<=60;
gen tax5= ( tax*ch + ((1-ch)*t5) ) if month_e<=60;
gen tax20= ( tax*ch + ((1-ch)*t20) ) if month_e<=60;
drop tt A A5 A20;

gen hazardw5=hazard_ if month_e<=60;
```

```
replace hazardw5=(1-ch) if month_e==61;
gen hazardw20=hazard_ if month_e<=60;
replace hazardw20=(1-ch) if month_e==61;
gen taxw5=ttt if month_e<=60;
replace taxw5=t5 if month_e==61;
gen taxw20=ttt if month_e<=60;
replace taxw20=t20 if month_e==61;
gen taxw100=ttt if month_e<=60;
replace taxw100=0 if month_e==61;

sort id2;
by id2: list hazard_ ch return hazardw5 hazardw20 if id<=0 & month_e<=61;
by id2: sum taxw5 taxw100 [w=hazardw5];
by id2: sum taxw20 taxw100 [w=hazardw20];
drop taxw5 taxw20 taxw100 hazardw*;

gen hazardw5=hazard_*return if month_e<=60;
replace hazardw5=r5*(1-ch) if month_e==61;
gen hazardw20=hazard_*return if month_e<=60;
replace hazardw20=r20*(1-ch) if month_e==61;
gen taxw5=ttt if month_e<=60;
replace taxw5=t5 if month_e==61;
gen taxw20=ttt if month_e<=60;
replace taxw20=t20 if month_e==61;
gen taxw100=ttt if month_e<=60;
replace taxw100=0 if month_e==61;
drop r5 r20 t5 t20;

sort id2;
by id2: list hazard_ ch return hazardw5 hazardw20 if id<=0 & month_e<=61;
by id2: sum taxw5 taxw100 [w=hazardw5];
by id2: sum taxw20 taxw100 [w=hazardw20];
drop taxw5 taxw20 taxw100 hazardw*;

sort id2;
by id2: sum gain loss r month_e if id<=0;
sort id month_e;
sort id2;
by id2: sum hazard hazard_ chazard expect expect5 expect20 tax* if id<=0 & month_e==60;
by id2: list month_e hazard hazard_ chazard ttt if id<=0 & month_e<=60;
drop hazard hazard_ chazard expect expect5 expect20 tax* ttt keep;
drop ch;
```