

Risk Aversion and Wealth: Evidence from Person-to-Person Lending Portfolios
by Daniel Paravisini, Veronica Rappoport, and Enrichetta Ravina

Definition of the variables in our dataset. Additional variables are created and defined in the code.

Variable name	Definition
investor_id	investor numeric identifier
gender	investor's gender
marital_status	investor's marital status
home_owner_renter	investor's home ownership status; "Renter", "Home Owner"
net_worth	investor's net worth; see code for brackets
age	investor's age
zip_code	zip code where the investor lives
zhvi_current	median house price for the zip code where the investor lives
creation_time	time and date the investment in a given loan is made by the investor
creation_method	"Automatic" vs. "Manual" and "Hybrid" loan selection method
amount_invested	amount invested in a given loan at each creation date
nfirms	number of loan available for investment at a given date
assumed_default_rate	historical default rate for loans with a give credit grade based on a long-run validation study conducted on 10 years of credit bureau data
r_y3	3 yr US treasury rate
loan_id	loan numeric identifier
credit_grade	one of 35 risk buckets a loan is assigned to, based on FICO score, credit history, amount requested, ...
interest_rate	interest rate charged to loans assigned a specific credit grade
amount_requested	loan amount requested
amount_borrowed	loan amount granted (same as amount_requested if loan is granted, 0 otherwise)
monthly_payment	monthly installment payment
application_date	date the borrower applied for a loan on LC
fico_score	borrower's FICO score
debt_to_income_ratio	borrower's debt to income ratio from credit bureau
monthly_income	borrower's monthly income from credit bureau

***RISK AVERSION AND WEALTH: EVIDENCE FROM PERSON-TO-PERSON
LENDING PORTFOLIOS**

***by Daniel Paravisini, Veronica Rappoport, and Enrichetta
Ravina**

```
clear all
set more off
```

```
*Put the original datasets in this directory
global datapath0 "/Volumes/Seagate Backup Plus
Drive/AAA/Lending Club/RISK AVERSION and WEALTH/Revision/New
Table 7"
```

```
*This directory will contain tables and figures generated by
the code
global datapathoutreg "/Volumes/Seagate Backup Plus
Drive/AAA/Lending Club/RISK AVERSION and
WEALTH/Revision/Management Science/REPLICATE/outreg"
```

```
cd "$datapath0"
```

```
*Returns, ns and ps for every choice
*A choice is defined as an individual making an investment
within a time interval (day)
```

```
use "investments_complete.dta", clear
gen investment_date=dofc(creation_time)
encode creation_method, gen(creationmethod)
replace creationmethod=0 if creationmethod==2
drop creation_method
collapse (sum) amount_invested (max) creationmethod, by
(investor_id loan_id investment_date)
sort loan_id
merge loan_id using "cu_loans.dta", nok keep( interest_rate
credit_grade amount_requested monthly_payment)
drop if _merge==1 /*this data is missing because
the cu_loans sample ends earlier*/
drop _merge
sort credit_grade investment_date
merge credit_grade investment_date using
"available_projects_daily", nok keep(nfirms_)
drop if _merge==1 /*same thing*/
drop _merge
rename nfirms_ nfirms_available
sort credit_grade
merge credit_grade using "default_probability_modif.dta"
drop _merge
gen inv_month=mofd(investment_date)
gen pre=inv_month<580
drop if inv_month>=580 & inv_month<=584
drop if inv_month<=572
sort inv_month credit_grade
merge inv_month credit_grade using
"nfirmsavailable_monthgrade.dta", nok keep(nfirms)
drop if _merge==1 /*Censoring at the beginning
```

```

of the sample: 178 obs*/
drop _merge
destring monthly_payment, replace
rename amount_invested investment_amount
egen nz_i_actual=count(investment_amount), by(investor_id
inv_month credit_grade )
compress

/*Keep the "rational" investments + the ones chosen with
portfolio tool.
Drop this command to keep all investments, including cases
where the investor only lends to one borrower */
keep if investment_amount<=50 | creationmethod==0

collapse (sum) investment_amount amount_requested
monthly_payment (mean) creationmethod pre nz_i_actual
nfirms_available nfirms interest_rate assumed_default_rate, by
(investor_id inv_month credit_grade )
sort inv_month
merge inv_month using "riskfreeratesby_inv_month.dta", nok
keep(r_y3)
drop _merge
rename r_y3 rf
replace rf=(1+rf/100)^(1/12)-1

*****Change default rate here
gen def_rate_1y=assumed_default_rate

gen def_rate_1m=(1+def_rate_1y)^(1/12)-1
gen paymnt=amount_requested*((interest_rate/12)*(1+
(interest_rate/12))^36)/(-1+(1+(interest_rate/12))^36)
gen Rz=1*(paymnt/amount_requested)
gen Piz=(1-def_rate_1m)/(1+rf)
gen Xz=Piz*(1-Piz^36)/(1-Piz)
gen ERz=Rz*Piz*(1-Piz^36)/(1-Piz)
gen PihatZ=Piz/(1+rf)
gen Sum_Rz=Rz/(1+rf)
gen VRz= def_rate_1m*(1-def_rate_1m)*(Sum_Rz)^2
set more off
foreach i of numlist 2(1)35 {
    replace Sum_Rz= Sum_Rz + Rz/(1+rf)^`i'
    replace VRz=VRz+def_rate_1m*(1-def_rate_1m)^`i'*(Sum_Rz)^
2
}
set more on
replace VRz=VRz - ERz^2 + (1-def_rate_1m)^36*(Sum_Rz+Rz/(1+rf)
^36)^2
egen W=sum(investment_amount), by(investor_id inv_month)

*****Here we choose how to measure the number of projects
available:
gen nz_i=nfirms

gen VrZ_actual= (1/nz_i_actual) * VRz
gen VrZ= (1/nz_i) * VRz
gen xVrZ_actual=investment_amount*VrZ_actual/W
gen xVrZ=investment_amount*VrZ/W

```

```

gen WxVrz_actual=investment_amount*Vrz_actual
gen WxVrz=investment_amount*Vrz
egen ER=sum(ERz), by(investor_id inv_month)
gen lW=ln(W)
egen C=sum(investment_amount*ERz), by(investor_id inv_month)
gen lC=ln(C)

keep if inv_month<=579

*****Here choose minimum number of buckets in beta
calculation:
egen count_investmonth=count(WxVrz), by(investor_id inv_month)
egen invmonth_id=group(investor_id inv_month) if WxVrz<. &
count_investmonth>=2
egen count_investmonth_auto=count(WxVrz) if creationmethod==0,
by(investor_id inv_month)
egen invmonth_id_auto=group(investor_id inv_month) if WxVrz<.
& count_investmonth_auto>=2 & creationmethod==0
egen count_investmonth_twk=count(WxVrz) if creationmethod>0,
by(investor_id inv_month)
egen invmonth_id_twk=group(investor_id inv_month) if WxVrz<. &
count_investmonth_twk>=2 & creationmethod>0

encode credit_grade, gen(creditgrade)
gen byte highgrade=creditgrade>=19
egen byte tag_invmonth=tag(investor_id inv_month)
egen byte tag_invmonthgrade=tag(investor_id inv_month
highgrade)
compress
keep if invmonth_id<.
sum invmonth_id invmonth_id_auto invmonth_id_twk

*****ARA and THETA ESTIMATION*****
*Beta = Absolute risk aversion
*Alpha = Theta
gen betaim_act=.
gen betaim_act_sd=.
gen alphaim_act=.
gen alphaim_act_sd=.

set more off

*All: Rational Investments with 25 cutoff subsample +
portfolio assignment
forvalues i = 1(1)3745 {
    display `i' "/"3745"
    quietly reg ERz WxVrz_actual if invmonth_id==`i', r
    quietly replace betaim_act=_b[WxVrz_actual] if
invmonth_id==`i'
    quietly replace alphaim_act=_b[_cons] if invmonth_id==`i'
    quietly replace betaim_act_sd=_se[WxVrz_actual] if
invmonth_id==`i'
    quietly replace alphaim_act_sd=_se[_cons] if
invmonth_id==`i'
}

gen betaim_aut=.

```

```

gen betaim_aut_sd=.
gen alphaim_aut=.
gen alphaim_aut_sd=.
*Subset: Automatic Portfolios
forvalues i = 1(1)1363 {
    display `i' "/"1363"
    quietly reg ERz WxVrz_actual if invmonth_id_auto==`i', r
    quietly replace betaim_aut=_b[WxVrz_actual] if
invmonth_id_auto==`i'
    quietly replace alphaim_aut=_b[_cons] if
invmonth_id_auto==`i'
    quietly replace betaim_aut_sd=_se[WxVrz_actual] if
invmonth_id_auto==`i'
    quietly replace alphaim_aut_sd=_se[_cons] if
invmonth_id_auto==`i'
}

```

```

gen betaim_twk=.
gen betaim_twk_sd=.
gen alphaim_twk=.
gen alphaim_twk_sd=.
*Subset: Manual Portfolios
forvalues i = 1(1)2871 {
    display `i' "/"2871"
    quietly reg ERz WxVrz_actual if invmonth_id_twk==`i', r
    quietly replace betaim_twk=_b[WxVrz_actual] if
invmonth_id_twk==`i'
    quietly replace alphaim_twk=_b[_cons] if
invmonth_id_twk==`i'
    quietly replace betaim_twk_sd=_se[WxVrz_actual] if
invmonth_id_twk==`i'
    quietly replace alphaim_twk_sd=_se[_cons] if
invmonth_id_twk==`i'
}
set more on
compress

```

```

sort investor_id
merge investor_id using "cu_investors.dta", nok keep(zip_code)
drop if _merge==2
drop _merge

```

```

**Obtain wealth data from acxiom.dta and zillow
*****

```

```

merge m:1 investor_id using acxiom.dta
drop if _merge==2
drop _merge
gen inc=C-W
gen betaim_act_inc=betaim_act*inc

```

```

encode net_worth, gen(networth)
replace networth=networth+100
replace networth=0 if networth==111
replace networth=2500 if networth==101
replace networth=7500 if networth==108
replace networth=(10000+25000)/2 if networth==103
replace networth=(25000+50000)/2 if networth==106

```

```

replace networth=(50000+100000)/2 if networth==109
replace networth=(250000+500000)/2 if networth==107
replace networth=(500000+1000000)/2 if networth==110
replace networth=(1000000+2000000)/2 if networth==102
replace networth=3000000 if networth==105
gen lnetworth=ln(networth)

sort zip_code inv_month
merge zip_code inv_month using zillow_zhvi_for_monthly.dta
tab _merge
drop if _merge==2
drop _merge

save "$datapath0/investment_riskaversion10
_rationalinv25portfolio.dta", replace

***** FIGURE 2 *****
twoway (sc ERz WxVrz_actual, mlabel(credit_grade) mlabposition
(6)) (lfit ERz WxVrz_actual) if investor_id==285983, title
("Theta = 1.082, ARA = .0661")
graph save "$datapathoutreg/Fig2a.gph", replace
twoway (sc ERz WxVrz_actual, mlabel(credit_grade) mlabposition
(6)) (lfit ERz WxVrz_actual) if betaim_act_inc==
8.542567253112793, title("Theta = 1.068, ARA = .0120")
graph save "$datapathoutreg/Fig2b.gph", replace

twoway (sc ERz WxVrz_act, mlabel(credit_grade) mlabposition
(6)) (lfit ERz WxVrz_act) if betaim_aut_inc<. & investor_id==
70654 & inv_month==576 & creationmethod==0, title("Theta =
1.075, ARA = 0.048")
graph save "$datapathoutreg/Fig2c.gph", replace
twoway (sc ERz WxVrz_act, mlabel(credit_grade) mlabposition
(6)) (lfit ERz WxVrz_act) if betaim_twk_inc<. & investor_id==
70654 & inv_month==576 & creationmethod>0, title("Theta =
1.068, ARA = 0.051")
graph save "$datapathoutreg/Fig2d.gph", replace
clear

***** TABLE 1 *****
* Section A - Borrowers Characteristics
cd "$datapath0"
use "$datapath0/cu_loans.dta", clear
sort loan_id
merge loan_id using "$datapath0/credit_scores.dta"
drop if _merge==2
drop _merge
gen sample=mofd( application_date)>=573 & mofd(
application_date)<=579
gen sample1=mofd( issued_date)>=573 & mofd( issued_date)<=579
tabstat fico_score debt_to_income_ratio monthly_income
amount_borrowed if sample1==1, s(mean sd p50) c(s) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table1_BorChars.xls)
replace

* Section B - Investors' Characteristics
cd "$datapath0"
use "$datapath0/investment_riskaversion10

```

```

_rationalinv25portfolio.dta"

gen byte male=gender=="Male"
gen married=marital_status=="Inferred Married" |
marital_status=="Married"
gen homeowner=home_owner_renter=="Home Owner"
replace networth=networth/1000

preserve
drop if zhvi_current==.
tabstat male age married homeowner networth, s(mean sd p50) c
(s) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table1
_InvChars_firstrows.xls) replace
restore

keep zip_code zhvi_current inv_month
duplicates drop
drop if inv_month>579
keep if inv_month==573 | inv_month==579
sort zip_code inv_month
by zip_code: gen change_in_house_prices=
(zhvi_current/zhvi_current[_n-1])-1
replace zhvi_current=zhvi_current/1000
tabstat zhvi_current change_in_house_prices, s(mean sd p50) c
(s) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table1
_InvChars_last2rows.xls) replace
clear

***** TABLE 2 - Diversified Investments
and "With Real Estate Data" *****
*See the end of the code for teh commands to generate TABLE 2
All Investments

cd "$datapath0"
use "$datapath0/investment_riskaversion10
_rationalinv25portfolio.dta"

*by investor-bucket-month
tabstat investment_amount nz_i_actual interest_rate def_rate_
ly ERz Vrz_actual if betaim_act_inc~=. , s(mean sd p50 n) c(s)
save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table2_DivInv_SecA.xls)
replace

tabstat investment_amount nz_i_actual interest_rate def_rate_
ly ERz Vrz_actual if betaim_act_inc~=. & zhvi_current~=. , s
(mean sd p50 n) c(s) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table2_RealEst_SecA.xls)
replace

*by investor-month

```

```

gen avg_return=C/W
egen avg_variance=sum((investment_amount/W)^2*Vrz_actual), by
(inv_ investor_id inv_month)
egen nbuckets=count(investment_amount), by(inv_ investor_id
inv_month)
egen nprojects=sum(nz_i_actual), by(inv_ investor_id inv_month)
keep investor_id zip_code W nbuckets nprojects avg_return
avg_variance betaim_act_inc zhvi_current inv_month
duplicates drop

```

```

tabstat W nbuckets nprojects avg_return avg_variance if
betaim_act_inc~=., s(mean sd p50 n) c(s) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table2_DivInv_SecB.xls)
replace

```

```

tabstat W nbuckets nprojects avg_return avg_variance if
betaim_act_inc~=. & zhvi_current~=., s(mean sd p50 n) c(s)
save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table2_RealEst_SecB.xls)
replace

```

```
clear
```

```

cd "$datapath0"
use "$datapath0/investment_riskaversion10
_rationalinv25portfolio.dta"

```

```

**We drop duplicates to obtain values that are investor x
month specific
keep investor_id inv_month inc betaim_act betaim_act_inc
alphaim_act W lnetworth zip home_owner_renter zhvi_current
duplicates drop
duplicates tag investor_id inv_month betaim_act, gen(dupl)
drop if dupl==1
drop dupl
egen byte tag_investor=tag(investor_id)

```

***** TABLE 3 *****

```

preserve
keep if zhvi_current~=. & betaim_act~=.
collapse (mean) betaim_act betaim_act_inc inc alphaim_act
zhvi_current, by(inv_ investor_id inv_month)
tabstat betaim_act alphaim_act inc betaim_act_inc, c(v) stat
(mean sd p1 p10 p25 p50 p75 p90 p99 n) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table3.xls) replace
restore

```

***** TABLE 4 *****

```

preserve
keep if zhvi_current~=. & betaim_act~=.
tabstat betaim_act alphaim_act inc betaim_act_inc, c(v) s(mean
sd) by(inv_month) nototal save
tabstatmat mat1

```



```

xml_tab mat1, save($datapathoutreg/table4.xls) replace
restore
clear

***** TABLE 5 *****

cd "$datapath0"
use investment_riskaversion10_rationalinv25portfolio.dta

keep if zhvi_current~=. & betaim_act~=.

keep alphaim_twk alphaim_aut betaim_aut betaim_twk investor_id
inv_month
duplicates drop
sort investor_id inv_month
by investor_id inv_month: egen betaim_autN=mean(betaim_aut)
by investor_id inv_month: egen betaim_twkN=mean(betaim_twk)
by investor_id inv_month: egen alphaim_autN=mean(alphaim_aut)
by investor_id inv_month: egen alphaim_twkN=mean(alphaim_twk)
keep investor_id inv_month *N
duplicates drop
gen alpha_diff=alphaim_twkN-alphaim_autN
gen beta_diff=betaim_twkN-betaim_autN
keep if alpha_diff~=. & beta_diff~=.

tabstat alphaim_twkN alphaim_autN alpha_diff betaim_twkN
betaim_autN beta_diff, s(mean sd p1 p5 p25 p50 p75 p95 p99 n)
c(s) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table5_all.xls) replace
ttest alphaim_twkN=alphaim_autN
ttest betaim_twkN=betaim_autN

tabstat alphaim_twkN alphaim_autN alpha_diff betaim_twkN
betaim_autN beta_diff if inv_month<=575, s(mean sd p1 p5 p25
p50 p75 p95 p99 n) c(s) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table5_pre.xls) replace
ttest alphaim_twkN=alphaim_autN if inv_month<=575
ttest betaim_twkN=betaim_autN if inv_month<=575

tabstat alphaim_twkN alphaim_autN alpha_diff betaim_twkN
betaim_autN beta_diff if inv_month>575, s(mean sd p1 p5 p25
p50 p75 p95 p99 n) c(s) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table5_post.xls) replace
ttest alphaim_twkN=alphaim_autN if inv_month>575
ttest betaim_twkN=betaim_autN if inv_month>575

*Figure 3a and Figure 3b
kdensity beta_diff, nor title ("") xtitle("ARA Non-Automatic -
ARA Automatic")
graph save Graph "$datapathoutreg/ARA_Fig3a.gph"
kdensity alpha_diff, nor title ("") xtitle("Theta Non-
Automatic - Theta Automatic")
graph save Graph "$datapathoutreg/Theta_Fig3b.gph"
clear

```

***** FIGURE 4a and FIGURE 4b

```

cd "$datapathoutreg"
use "$datapath/investment_riskaversion10
_rationalinv25portfolio.dta"

gen lhouseprice=ln(zhvi_current)

winsor betaim_act_inc, gen(betaim_act_inc2) p(0.01)
winsor betaim_act, gen(betaim_act2) p(0.01)

winsor betaim_act_inc, gen(betaim_act_inc3) p(0.05)
winsor betaim_act, gen(betaim_act3) p(0.05)

*Bandwidth =0.75
twoway (lpolyci betaim_act2 lnetworth if lnetworth>10 &
tag_invmonth==1, degree(2) bwidth(0.75) ciplot(rline) )
(lpolyci betaim_act2 lhouseprice if lhouseprice>10 &
lhouseprice <14.5 & tag_invmonth==1, degree(0) bwidth(0.75)
ciplot(rline)) if home_owner_renter=="Home Owner", ytitle
("ARA") xtitle("log Net Worth/ log Median House Prices")
legend(label(1 "95% C.I.") label(2 "ARA and Net Worth") label
(3 "ARA and House Prices") order (2 3 1))
graph save "$datapathoutreg/Fig4a_ARA_HousePrices.gph",
replace
twoway (lpolyci betaim_act_inc2 lnetworth if lnetworth>10 &
tag_invmonth==1, degree(2) bwidth(0.75) ciplot(rline) )
(lpolyci betaim_act_inc2 lhouseprice if lhouseprice>10 &
lhouseprice <14.5 & tag_invmonth==1, degree(0) bwidth(0.75)
ciplot(rline)) if home_owner_renter=="Home Owner", ytitle
("RRA") xtitle("log Net Worth/ log Median House Prices")
legend(label(1 "95% C.I.") label(2 "RRA and Net Worth") label
(3 "RRA and House Prices") order (2 3 1))
graph save "$datapathoutreg/Fig4b_RRA_HousePrices.gph",
replace
clear

```

***** TABLE 6 *****

```

cd "$datapath0"
use "$datapath0/investment_riskaversion10
_rationalinv25portfolio.dta"
**We drop duplicates to obtain values that are investor x
month specific
keep investor_id inv_month inc betaim_act betaim_act_inc
alphaim_act W lnetworth zip home_owner_renter zhvi_current
duplicates drop
duplicates tag investor_id inv_month betaim_act, gen(dupl)
drop if dupl==1
drop dupl
egen byte tag_investor=tag(investor_id)

preserve
egen zip_g=group(zip)
**Aggregating at the investor level
collapse (mean) betaim_act betaim_act_inc W zhvi_current
lnetworth zip_g, by(investor_id)

```

```
foreach i of varlist betaim_act betaim_act_inc W zhvi_current
{
gen l`i'=ln(`i')
}
```

```
reg lbetaim_act lnetworth if zhvi_current!=., r cl(zip)
outreg2 lnetworth using "$datapathoutreg/table6.xls", replace
ct(lARA) bdec(3) aster(coef) se e( r2_a r2_o r2_b r2_w )
reg lbetaim_act_inc lnetworth if lbetaim_act!=. &
zhvi_current!=., r cl(zip)
outreg2 lnetworth using "$datapathoutreg/table6.xls", append
ct(lRRA) bdec(3) aster(coef) se e( r2_a r2_o r2_b r2_w )
reg lW lnetworth if lbetaim_act!=. & zhvi_current!=., r cl
(zip)
outreg2 lnetworth using "$datapathoutreg/table6.xls", append
ct(lW) bdec(3) aster(coef) se e( r2_a r2_o r2_b r2_w )
ivregress 2sls lbetaim_act (lnetworth=lnzhvi_current), r cl
(zip)
outreg2 lnetworth using "$datapathoutreg/table6.xls", append
ct(IV: lARA) bdec(3) aster(coef) se e( r2_a r2_o r2_b r2_w )
ivregress 2sls lbetaim_act_inc (lnetworth=lnzhvi_current) if
lbetaim_act!=., r cl(zip)
outreg2 lnetworth using "$datapathoutreg/table6.xls", append
ct(IV: lRRA) bdec(3) aster(coef) se e( r2_a r2_o r2_b r2_w )
ivregress 2sls lW (lnetworth=lnzhvi_current) if lbetaim_act!=.
, r cl(zip)
outreg2 lnetworth using "$datapathoutreg/table6.xls", append
ct(IV: lW) bdec(3) aster(coef) se e( r2_a r2_o r2_b r2_w )
reg lnetworth lzhvi_current if lbetaim_act!=., r cl(zip)
outreg2 lzhvi_current using "$datapathoutreg/table6.xls",
append ct(FS: lnetworth) bdec(3) aster(coef) se e( r2_a r2_o
r2_b r2_w )
```

restore

***** TABLE 7

```
preserve
gen trend=0
forvalues i=573(1)591 {
replace trend=`i'-570 if inv_month==`i'
}
```

```
foreach i of varlist betaim_act betaim_act_inc W zhvi_current
trend {
gen l`i'=ln(`i')
}
```

**Tag investors that appear at least twice

```
bys investor_id: egen n_invest=count(lbetaim_act)
```

```
xtset investor_id inv_month
```

```
reg lbetaim_act lzhvi_current ltrend if n_invest>1, r cl(zip)
outreg2 lzhvi_current using "$datapathoutreg/table7tr1
_Old.xls", replace ct(lARA tr1) bdec(3) aster(coef) se e( r2_a
```

```

r2_o r2_b r2_w )
reg lbetaim_act_inc lzhvi_current ltrend if n_invest>1 &
lbetaim_act!=., r cl(zip)
outreg2 lzhvi_current using "$datapathoutreg/table7tr1
_Old.xls", append ct(lRRA tr1) bdec(3) aster(coef) se e( r2_a
r2_o r2_b r2_w )
reg lW lzhvi_current ltrend if n_invest>1 & lbetaim_act!=., r
cl(zip)
outreg2 lzhvi_current using "$datapathoutreg/table7tr1
_Old.xls", append ct(lW tr1) bdec(3) aster(coef) se e( r2_a r2
_o r2_b r2_w )
xtreg lbetaim_act lzhvi_current ltrend if n_invest>1, fe r cl
(zip)
outreg2 lzhvi_current using "$datapathoutreg/table7tr1
_Old.xls", append ct(FE + lARA + tr1) bdec(3) aster(coef) se e
( r2_a r2_o r2_b r2_w )
xtreg lbetaim_act_inc lzhvi_current ltrend if n_invest>1 &
lbetaim_act!=., fe r cl(zip)
outreg2 lzhvi_current using "$datapathoutreg/table7tr1
_Old.xls", append ct(FE + lRRA + tr1) bdec(3) aster(coef) se e
( r2_a r2_o r2_b r2_w )
xtreg lW lzhvi_current ltrend if n_invest>1 & lbetaim_act!=. ,
fe r cl(zip)
outreg2 lzhvi_current using "$datapathoutreg/table7tr1
_Old.xls", append ct(FE + lW + tr1) bdec(3) aster(coef) se e(
r2_a r2_o r2_b r2_w )

```

```
clear
```

```

*****
*** APPENDIX
*****

```

```

***** Amount Invested in LC
cd "$datapath0"
use "$datapath0/investment_riskaversion10
_rationalinv25portfolio.dta"
drop if zhvi_current==.
*Averaging by investor and IV:
foreach var of varlist betaim_act W{
    egen m_`var'=mean(`var'), by(investor_id)
    gen lm_`var'=ln(m_`var')
}
egen byte tag_investor=tag(investor_id)
reg lm_W lm_betaim_act if home_owner_renter=="Home Owner" &
inv_month<579 & tag_investor==1, r cl(zip) level(99)
clear
***** Figure 7
winsor betaim_act, gen(betaim_act2) p(0.01)
winsor betaim_act, gen(betaim_act3) p(0.05)

egen byte tag_invmonth=tag(investor_id inv_month)
gen INV=ln(investment_amount)
twoway (lpolyci lm_W betaim_act3, degree(2) bwidth(0.75)
ciplot(rline) ) if tag_invmonth==1 & home_owner_renter=="Home
Owner", ytitle("ln(Investment)") xtitle("ARA") legend(label(1
"95% C.I.") label(2 "ln(Investment)") order (1 2))

```

```
graph save "$datapathoutreg/FigC1.gph", replace
```

*See the outreg for table 6 and table 7 for the first two and last two columns of Table C.1, respectively.

```
***** Rationality Test *****
```

```
program make_matrices
    tempvar tag_bucketmonth
    egen `tag_bucketmonth'=tag(credit_grade inv_month)
    preserve
    keep if `tag_bucketmonth'==1
    keep credit_grade inv_month ERz
    sort inv_month credit_grade
    reshape wide ERz, i(credit_grade) j(inv_month)
    mkmat ERz573 - ERz591, matrix(RET)
    restore
    preserve
    keep if `tag_bucketmonth'==1
    keep credit_grade inv_month VRz
    sort inv_month credit_grade
    reshape wide VRz, i(credit_grade) j(inv_month)
    mkmat VRz573 - VRz591, matrix(VAR)
    restore
end
```

```
cd "$datapath"
use "$datapath0/investment_riskaversion10
_rationalinv25portfolio.dta", clear
make_matrices
keep investor_id inv_month invmonth_id creditgrade
investment_amount creationmethod betaim_act betaim_act_sd
alphaim_act alphaim_act_sd
drop if invmonth_id==.

egen cm=sum(creationmethod), by(invmonth_id )
keep if cm>0
drop creationmethod cm

reshape wide investment_amount, i(invmonth_id) j(creditgrade)
forvalues h = 1(1)35 {
    gen t`h'= RET[`h',inv_month-572] - alphaim_act -
    betaim_act*25*VAR[`h',inv_month-572] if
    investment_amount`h'==.
}
forvalues h = 1(1)35 {
    gen tsd`h'= ( alphaim_act_sd^2 + betaim_act_sd^2*( 25*VAR
    [`h',inv_month-572])^2 )^.5 if investment_amount`h'==.
}

drop investment_amount*
reshape long t tsd, i(invmonth_id) j(creditgrade)
keep if t<.
keep if inv_month<=579
replace tsd=. if tsd==0
```

```

gen tneg=t<0
sum tneg, d
gen tmin=t-1.31*tsd
gen tneg_min=tmin<0
egen n=count(t), by(invmonth_id)
gen t_RET=t/(RET[creditgrade,inv_month-572]-1)

*Desc Stats
gen ERz=RET[creditgrade,inv_month-572]
gen VRz=VAR[creditgrade,inv_month-572]
tabstat ERz VRz , s(mean sd p1 p25 p50 p75 p99 n) c(s)

sum tneg , d

sum t , d
ttest t=0
sum t if t>0, d

*Investment-by-investment:
egen invmonth_id_test=group(invmonth_id)
sum invmonth_id_test
gen tstat=.
set more off
forvalues i = 1(1)2938 {
    di `i'
    quietly ttest t=0 if invmonth_id_test==`i'
    quietly replace tstat=r(t) if invmonth_id_test==`i'
}
set more on
egen tag_test=tag(invmonth_id_test)
sum tstat if tag_test==1, d
ins tstat if tag_test==1

ins tmin
sum tmin , d
sum t if tmin>0 & tmin<., d
sum t_RET if tmin>0 & tmin<., d

```

***** Table 2 - All

```

Investments *****
*This section requires to recreate the dataset keeping all
investments, not only the diversified ones.
use "investments_complete.dta", clear
gen investment_date=dofc(creation_time)
encode creation_method, gen(creationmethod)
replace creationmethod=0 if creationmethod==2
drop creation_method
collapse (sum) amount_invested (max) creationmethod, by
(investor_id loan_id investment_date)
sort loan_id
merge loan_id using "cu_loans.dta", nok keep( interest_rate
credit_grade amount_requested monthly_payment)
drop if _merge==1 /*this data is missing because
the cu_loans sample ends earlier*/
drop _merge
sort credit_grade investment_date
merge credit_grade investment_date using

```

```

"available_projects_daily", nok keep(nfirms_)
drop if _merge==1 /*same thing*/
drop _merge
rename nfirms_ nfirms_available
sort credit_grade
merge credit_grade using "default_probability_modif.dta"
drop _merge
gen inv_month=mofd(investment_date)
gen pre=inv_month<580
drop if inv_month>=580 & inv_month<=584
drop if inv_month<=572
sort inv_month credit_grade
merge inv_month credit_grade using
"nfirmsavailable_monthgrade.dta", nok keep(nfirms)
drop if _merge==1 /*Censoring at the beginning
of the sample: 178 obs*/
drop _merge
destring monthly_payment, replace
rename amount_invested investment_amount
egen nz_i_actual=count(investment_amount), by(investor_id
inv_month credit_grade )
compress

/*Keep the "rational" investments + the ones chosen with
portfolio tool.
Drop this command to keep all investments, including cases
where the investor only lends to one borrower */
*keep if investment_amount<=50 | creationmethod==0

collapse (sum) investment_amount amount_requested
monthly_payment (mean) creationmethod pre nz_i_actual
nfirms_available nfirms interest_rate assumed_default_rate
lc_base_rate default_rate_1y interest_rate_table, by
(investor_id inv_month credit_grade )
sort inv_month
merge inv_month using "riskfreeratesby_inv_month.dta", nok
keep(r_y3)
drop _merge
rename r_y3 rf
replace rf=(1+rf/100)^(1/12)-1

*****Change default rate here
gen def_rate_1y=assumed_default_rate

gen def_rate_1m=(1+def_rate_1y)^(1/12)-1
gen paymnt=amount_requested*((interest_rate/12)*(1+
(interest_rate/12))^36)/(-1+(1+(interest_rate/12))^36)
gen Rz=1*(paymnt/amount_requested)
gen Piz=(1-def_rate_1m)/(1+rf)
gen Xz=Piz*(1-Piz^36)/(1-Piz)
gen ERz=Rz*Piz*(1-Piz^36)/(1-Piz)
gen PihatZ=Piz/(1+rf)
gen Sum_Rz=Sum_Rz/(1+rf)
gen VRz= def_rate_1m*(1-def_rate_1m)*(Sum_Rz)^2
set more off
foreach i of numlist 2(1)35 {
    replace Sum_Rz= Sum_Rz + Rz/(1+rf)^^i'

```

```

        replace VRz=VRz+def_rate_lm*(1-def_rate_lm)^(i)*(Sum_Rz)^
2
    }
set more on
replace VRz=VRz - ERz^2 + (1-def_rate_lm)^36*(Sum_Rz+Rz/(1+rf)
^36)^2
egen W=sum(investment_amount), by(investor_id inv_month)

*****Here we choose how to measure the number of projects
available:
gen nz_i=nfirms

gen Vrz_actual= (1/nz_i_actual) * VRz
gen Vrz= (1/nz_i) * VRz
gen xVrz_actual=investment_amount*Vrz_actual/W
gen xVrz=investment_amount*Vrz/W
gen WxVrz_actual=investment_amount*Vrz_actual
gen WxVrz=investment_amount*Vrz
egen ER=sum(ERz), by(investor_id inv_month)
gen lW=ln(W)
egen C=sum(investment_amount*ERz), by(investor_id inv_month)
gen lC=ln(C)

keep if inv_month<=579

*****Here choose minimum number of buckets in beta
calculation:
egen count_investmonth=count(WxVrz), by(investor_id inv_month)
egen invmonth_id=group(investor_id inv_month) if WxVrz<. &
count_investmonth>=2
egen count_investmonth_auto=count(WxVrz) if creationmethod==0,
by(investor_id inv_month)
egen invmonth_id_auto=group(investor_id inv_month) if WxVrz<.
& count_investmonth_auto>=2 & creationmethod==0
egen count_investmonth_twk=count(WxVrz) if creationmethod>0,
by(investor_id inv_month)
egen invmonth_id_twk=group(investor_id inv_month) if WxVrz<. &
count_investmonth_twk>=2 & creationmethod>0

encode credit_grade, gen(creditgrade)
gen byte highgrade=creditgrade>=19
egen byte tag_invmonth=tag(investor_id inv_month)
egen byte tag_invmonthgrade=tag(investor_id inv_month
highgrade)
compress
keep if invmonth_id<.
sum invmonth_id invmonth_id_auto invmonth_id_twk

*****ARA and THETA ESTIMATION*****
*Beta = Absolute risk aversion
*Alpha = Theta
gen betaim_act=.
gen betaim_act_sd=.
gen alphaim_act=.
gen alphaim_act_sd=.

set more off

```



```

*All: Rational Investments with 25 cutoff subsample +
portfolio assignment
forvalues i = 1(1)4256 {
    display `i' "/"4256"
    quietly reg ERz WxVrz_actual if invmonth_id==`i', r
    quietly replace betaim_act=_b[WxVrz_actual] if
invmonth_id==`i'
    quietly replace alphaim_act=_b[_cons] if invmonth_id==`i'
    quietly replace betaim_act_sd=_se[WxVrz_actual] if
invmonth_id==`i'
    quietly replace alphaim_act_sd=_se[_cons] if
invmonth_id==`i'
}

gen betaim_aut=.
gen betaim_aut_sd=.
gen alphaim_aut=.
gen alphaim_aut_sd=.
*Subset: Automatic Portfolios
forvalues i = 1(1)1359 {
    display `i' "/"1359"
    quietly reg ERz WxVrz_actual if invmonth_id_auto==`i', r
    quietly replace betaim_aut=_b[WxVrz_actual] if
invmonth_id_auto==`i'
    quietly replace alphaim_aut=_b[_cons] if
invmonth_id_auto==`i'
    quietly replace betaim_aut_sd=_se[WxVrz_actual] if
invmonth_id_auto==`i'
    quietly replace alphaim_aut_sd=_se[_cons] if
invmonth_id_auto==`i'
}

gen betaim_twk=.
gen betaim_twk_sd=.
gen alphaim_twk=.
gen alphaim_twk_sd=.
*Subset: Manual Portfolios
forvalues i = 1(1)3460 {
    display `i' "/"3460"
    quietly reg ERz WxVrz_actual if invmonth_id_twk==`i', r
    quietly replace betaim_twk=_b[WxVrz_actual] if
invmonth_id_twk==`i'
    quietly replace alphaim_twk=_b[_cons] if
invmonth_id_twk==`i'
    quietly replace betaim_twk_sd=_se[WxVrz_actual] if
invmonth_id_twk==`i'
    quietly replace alphaim_twk_sd=_se[_cons] if
invmonth_id_twk==`i'
}
set more on
compress

sort investor_id
merge investor_id using "cu_investors.dta", nok keep(zip_code)
drop if _merge==2
drop _merge

```

```

**Obtain wealth data from acxiom.dta and zillow
*****
merge m:1 investor_id using acxiom.dta
drop if _merge==2
drop _merge
gen inc=C-W
gen betaim_act_inc=betaim_act*inc

encode net_worth, gen(networth)
replace networth=networth+100
replace networth=0 if networth==111
replace networth=2500 if networth==101
replace networth=7500 if networth==108
replace networth=(10000+25000)/2 if networth==103
replace networth=(25000+50000)/2 if networth==106
replace networth=(50000+100000)/2 if networth==109
replace networth=(250000+500000)/2 if networth==107
replace networth=(500000+1000000)/2 if networth==110
replace networth=(1000000+2000000)/2 if networth==102
replace networth=3000000 if networth==105
gen lnetworth=ln(networth)

sort zip_code inv_month
merge zip_code inv_month using zillow_zhvi_for_monthly.dta
tab _merge
drop if _merge==2
drop _merge

save "$datapath0/investment_riskaversion10
_rationalinv25portfolioBIS.dta", replace

*by investor-bucket-month
tabstat investment_amount nz_i_actual interest_rate def_rate_
ly ERz Vrz_actual if betaim_act_inc~=. , s(mean sd p50 n) c(s)
save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table2_A1Inv_SecA.xls)
replace

*by investor-month
gen avg_return=C/W
egen avg_variance=sum((investment_amount/W)^2*Vrz_actual), by
(investor_id inv_month)
egen nbuckets=count(investment_amount), by(investor_id
inv_month)
egen nprojects=sum(nz_i_actual), by(investor_id inv_month)
keep investor_id zip_code W nbuckets nprojects avg_return
avg_variance betaim_act_inc inv_month
duplicates drop

tabstat W nbuckets nprojects avg_return avg_variance if
betaim_act_inc~=. , s(mean sd p50 n) c(s) save
tabstatmat mat1
xml_tab mat1, save($datapathoutreg/table2_AllInv_SecB.xls)
replace

```

clear