

```
clear
*cd "F:\My Documents"
*cd "/Users/victorsimonsen/Documents/Master"
cd "/Users/Espen/Documents/BI/Master Thesis"
sysuse EV2
ssc install trimmean
```

```
*RENAME
{
rename item_9 Revenue
rename item_15 Depr
rename item_30 OtherIntExp
rename item_39 NetIncome
rename item_63 TotFixAss
rename item_76 Cash
rename item_78 TotCurAss
rename item_86 RetEarn
rename item_87 TotEq
rename item_105 DivPay
rename item_503 County
rename item_11102 Industry
rename item_13420 Age
rename item_14018 UltPerInst
rename item_15302 LargestFamOwn
rename item_15304 FamCeo
rename item_15307 LargestFamNumOwn
rename item_15311 UltOwnFam
rename item_18011 CeoOwn
rename item_50109 Empl
rename pcid Id
rename yr Year
rename item_17002 Listed
rename item_6 Enterprise
}
```

```
xtset Id Year
```

```
*DESTRING EMPLOYEES, COUNTY, INDUSTRY
{
destring Empl, generate(Empl_n) force
drop Empl
rename Empl_n Empl
drop if Empl==0
drop if mi(Empl)
```

```
destring County, replace
split Industry, p(",")
destring Industry1, replace
drop Industry2 Industry3
```

```
drop Industry
rename Industry1 Industry
}
```

```
*DROPS
```

```
{
drop if mi(Id)
drop if mi(Year)
drop if mi(Enterprise)
drop if mi(Revenue)
drop if mi(Depr)
drop if mi(OtherIntExp)
drop if mi(NetIncome)
drop if mi(TotFixAss)
drop if mi(TotCurAss)
drop if mi(Cash)
drop if mi(RetEarn)
drop if mi(TotEq)
drop if mi(DivPay)
drop if mi(County)
drop if mi(Industry)
drop if mi(Age)
drop if mi(UltPercInst)
drop if mi(LargestFamOwn)
drop if mi(FamCeo)
drop if mi(LargestFamNumOwn)
drop if mi(UltOwnFam)
drop if mi(Listed)
```

```
drop if Enterprise=="ANS"
drop if Listed==1
drop Listed
drop if TotFixAss<=0
drop if TotCurAss<=0
}
```

```
*INDUSTRY CHANGE MERGER
```

```
{
*merge 1:1 Id Year using "F:\My Documents\AfterIndChgNew.DTA"
*merge 1:1 Id Year using "/Users/victorsimonsen/Documents/Master/IND.DTA"
merge 1:1 Id Year using "/Users/Espen/Documents/BI/Master
Thesis/AfterIndChgNew.DTA"
drop _merge
drop Industry
drop if NewInd ==0
}
```

```
*INDUSTRY//
```

```
{
```

```

gen A = 0
replace A = 1 if inrange(NewInd,1001,3223)
gen B = 0
replace B = 1 if inrange(NewInd,5100,9900)
gen C = 0
replace C = 1 if inrange(NewInd,10110,33200)
gen D = 0
replace D = 1 if inrange(NewInd,35111,35300)
gen E = 0
replace E = 1 if inrange(NewInd,36000,39000)
gen F = 0
replace F = 1 if inrange(NewInd,41101,43990)
gen G = 0
replace G = 1 if inrange(NewInd,45111,47990)
gen H = 0
replace H = 1 if inrange(NewInd,49100,53200)
gen I = 0
replace I = 1 if inrange(NewInd,55101,56309)
gen J = 0
replace J = 1 if inrange(NewInd,58110,63990)
gen K = 0
replace K = 1 if inrange(NewInd,64100,66300)
gen L = 0
replace L = 1 if inrange(NewInd,68100,68320)
gen M = 0
replace M = 1 if inrange(NewInd,69100,75000)
gen N = 0
replace N = 1 if inrange(NewInd,77100,82990)
gen O = 0
replace O = 1 if inrange(NewInd,84110,84300)
gen P = 0
replace P = 1 if inrange(NewInd,85100,85609)
gen Q = 0
replace Q = 1 if inrange(NewInd,86101,88999)
gen R = 0
replace R = 1 if inrange(NewInd,90011,93299)
gen S = 0
replace S = 1 if inrange(NewInd,94100,96090)
gen T = 0
replace T = 1 if inrange(NewInd,97000,97000)
gen U = 0
replace U = 1 if inrange(NewInd,99000,99000)
drop if K == 1
drop if L == 1
drop K L
rename NewInd Industry
}
* INDUSTRIES EXCL. G
global Industries "A B C D E F H I J M N O P Q R S T"

```

*COUNTY

{

gen Oestfold = 0

replace Oestfold = 1 if inrange(County,100,199)

gen Akershus = 0

replace Akershus = 1 if inrange(County,200,299)

gen Oslo = 0

replace Oslo = 1 if inrange(County,300,399)

gen Hedmark = 0

replace Hedmark = 1 if inrange(County,400,499)

gen Oppland = 0

replace Oppland = 1 if inrange(County,500,599)

gen Buskerud = 0

replace Buskerud = 1 if inrange(County,600,699)

gen Vestfold = 0

replace Vestfold = 1 if inrange(County,700,799)

gen Telemark = 0

replace Telemark = 1 if inrange(County,800,899)

gen AustAgder = 0

replace AustAgder = 1 if inrange(County,900,999)

gen VestAgder = 0

replace VestAgder = 1 if inrange(County,1000,1099)

gen Rogaland = 0

replace Rogaland = 1 if inrange(County,1100,1199)

gen Hordaland = 0

replace Hordaland = 1 if inrange(County,1200,1299)

gen SognFjordane = 0

replace SognFjordane = 1 if inrange(County,1400,1499)

gen MoereRomsdal = 0

replace MoereRomsdal = 1 if inrange(County,1500,1599)

gen SoerTroendelag = 0

replace SoerTroendelag = 1 if inrange(County,1600,1699)

gen NordTroendelag = 0

replace NordTroendelag = 1 if inrange(County,1700,1799)

gen Nordland = 0

replace Nordland = 1 if inrange(County,1800,1899)

gen Troms = 0

replace Troms = 1 if inrange(County,1900,1999)

gen Finnmark = 0

replace Finnmark = 1 if inrange(County,2000,2099)

gen Unknown = 0

replace Unknown = 1 if inrange(County,9900,9099)

drop Unknown

gen Check = Oestfold + Akershus + Oslo + Hedmark + Oppland + Buskerud +
Vestfold + Telemark + AustAgder + VestAgder + Rogaland + Hordaland +
SognFjordane + MoereRomsdal + SoerTroendelag + NordTroendelag + Nordland
+ Troms + Finnmark

```
drop if Check == 0
drop Check
}
*Counties excl. Oslo
global Counties "Oestfold Akershus Hedmark Oppland Buskerud Vestfold
Telemark AustAgder VestAgder Rogaland Hordaland SognFjordane
MoereRomsdal SoerTroendelag NordTroendelag Nordland Troms Finnmark"
```

```
*BNP MERGER(?)
{
*merge 1:1 Id Year using "F:\My Documents\AfterBNPNew.dta"
*merge 1:1 Id Year using
"/Users/victorsimonsen/Documents/Master/BNP.DTA"
merge 1:1 Id Year using "/Users/Espen/Documents/BI/Master
Thesis/AfterBNPNew.DTA"
drop _merge
}
```

```
*INFLATION ADJUSTMENT
{
gen infl = 0
replace infl = 3.14 if Year == 2000
replace infl = 2.91 if Year == 2001
replace infl = 1.29 if Year == 2002
replace infl = 2.54 if Year == 2003
replace infl = 0.37 if Year == 2004
replace infl = 1.60 if Year == 2005
replace infl = 2.31 if Year == 2006
replace infl = 0.71 if Year == 2007
replace infl = 3.77 if Year == 2008
replace infl = 2.16 if Year == 2009
replace infl = 2.45 if Year == 2010
replace infl = 1.30 if Year == 2011
replace infl = 0.64 if Year == 2012
replace infl = 2.13 if Year == 2013
replace infl = 2.09 if Year == 2014
replace infl = 2.15 if Year == 2015
```

```
gen RevenueInfl = Revenue
gen TotAss = TotCurAss + TotFixAss
gen TotAssInfl = TotAss
gen NetIncomeInfl = NetIncome
gen DeprInfl = Depr
gen OtherIntExpInfl = OtherIntExp
gen CashInfl = Cash
gen RetEarnInfl = RetEarn
gen TotEqInfl = TotEq
gen DivPayInfl = DivPay
gen TotFixAssInfl = TotFixAss
```

```
gen TotCurAssInfl = TotCurAss
```

```
foreach x of varlist RevenueInfl TotAssInfl NetIncomeInfl DeprInfl  
OtherIntExpInfl CashInfl RetEarnInfl TotEqInfl DivPayInfl TotFixAssInfl  
TotCurAssInfl{  
    format `x' %12.0g  
    replace `x' = `x'*1.3374 if Year == 2000  
    replace `x' = `x'*1.2967 if Year == 2001  
    replace `x' = `x'*1.2600 if Year == 2002  
    replace `x' = `x'*1.2440 if Year == 2003  
    replace `x' = `x'*1.2131 if Year == 2004  
    replace `x' = `x'*1.2086 if Year == 2005  
    replace `x' = `x'*1.1896 if Year == 2006  
    replace `x' = `x'*1.1627 if Year == 2007  
    replace `x' = `x'*1.1545 if Year == 2008  
    replace `x' = `x'*1.1125 if Year == 2009  
    replace `x' = `x'*1.0890 if Year == 2010  
    replace `x' = `x'*1.0630 if Year == 2011  
    replace `x' = `x'*1.0493 if Year == 2012  
    replace `x' = `x'*1.0426 if Year == 2013  
    replace `x' = `x'*1.0209 if Year == 2014  
}  
}
```

```
bysort Id (Year): egen avgRev = mean(RevenueInfl)  
drop if RevenueInfl <= 0  
drop if avgRev <=1000000  
format avgRev %12.0g
```

```
bysort Id (Year) : drop if _N < (Year[_N] - Year[1] + 1)  
*Removing gaps
```

```
bysort Id (Year) : drop if _N==1  
*Removing firms with 1 obs
```

```
////////////////////////////////////  
////////////////////////////////VARIABLER////////////////////////////////  
////////////////////////////////////  
{
```

```
*RETENTION RATE
```

```
bysort Id (Year): gen RetRate = (RetEarnInfl[_n]-RetEarnInfl[_n-  
1])/NetIncomeInfl[_n]*100
```

```
*CASH RATE
```

bysort Id (Year): gen CashRate = CashInfl/TotAssInfl*100
drop if CashRate > 100

***AGE SQUARED**

gen AgeSq = Age*Age

*The distribution of age is believed to be exponential (Coad and Tamvada, 2008).
Hence, we use the squared value of age.

***SIZE**

gen LnRevenue = ln(RevenueInfl)

gen LnAssets = ln(TotAssInfl)

gen LnGrossFixAssets = ln(TotFixAssInfl-DeprInfl)

***ROA**

gen RoA = (NetIncomeInfl-OtherIntExpInfl)/TotAssInfl*100

* Note: Some investors add interest expense back into net income when
performing this calculation because they'd like to use operating returns before
cost of borrowing

* We subtract Int exp above because of the sign (-).

***ROE**

gen RoE = NetIncomeInfl/TotEqInfl*100

***FAMILY DUMMY**

gen Family = 0

replace Family = 1 if LargestFamOwn > 50

***PURE FAMILY**

gen PureFamily = 0

replace PureFamily = 1 if LargestFamOwn == 100

gen PureFamEntr = 0

replace PureFamEntr = 1 if PureFamily == 1 & LargestFamNumOwn == 1

***DEBT RATIO**

gen DebtRatio = (1-(TotEqInfl/TotAssInfl))*100

***MARGINAL MAJORITY OWNER**

gen MargMajority = 0

replace MargMajority = 1 if LargestFamOwn<=55

replace MargMajority = 0 if LargestFamOwn<=50

***SUPERMAJORITY OWNER 2/3**

gen Supermajority = 0

replace Supermajority = 1 if LargestFamOwn>= 66.67

*SALES GROWTH

bys Id: gen SalesGrowthPerc = ((RevenueInfl[_n]/RevenueInfl[_n-1])-1)*100 if Year[_n]-Year[_n-1]==1

*ASSETS GROWTH

bys Id: gen LnAssetsGrowth = (LnAssets[_n]-LnAssets[_n-1])*100 if Year[_n]-Year[_n-1]==1

*EMPLOYEE GROWTH

bys Id: gen EmplGrowthPerc = ((Empl[_n]/Empl[_n-1])-1)*100 if Year[_n]-Year[_n-1]==1

*INVESTMENTS PROXY

bys Id: gen InvestProxy = ((TotFixAssInfl[_n]-DeprInfl[_n])-(TotFixAssInfl[_n-1]))/TotFixAssInfl*100

*LARGE/SMALL FIRMS

qui sum avgRev, detail
gen Large = 1 if avgRev > r(p95)
gen Small = 1 if avgRev < r(p25)

}

*DESCRIPTIVE STATISTICS

{
/*

*TRIMMING

*ALL FIRMS

foreach x of varlist TotAssInfl RevenueInfl Empl Age RoA RoE SalesGrowthPerc
DebtRatio RetRate OtherIntExpInfl CashRate DeprInfl TotEqInfl NetIncomeInfl {

gen `x'TrimFamilyNo = `x' if Family == 0

gen `x'TrimEntr = `x' if PureFamEntr == 1

gen `x'TrimNotEntr = `x' if PureFamily == 1 & PureFamEntr == 0

gen `x'TrimPureNo = `x' if PureFamily == 0 & Family == 1

gen `x'TrimCeo = `x' if FamCeo == 1 & Family == 1

gen `x'TrimCeoNo = `x' if FamCeo == 0 & Family == 1

tabstat `x', statistics(mean median)

trimmean `x', percent(1)

tabstat `x'TrimFamilyNo, statistics(mean median)

trimmean `x'TrimFamilyNo, percent(1)

tabstat `x'TrimEntr, statistics(mean median)


```

trimmean `x'TrimEntr, percent(1)
tabstat `x'TrimNotEntr, statistics( mean median )
trimmean `x'TrimNotEntr, percent(1)
tabstat `x'TrimPureNo, statistics( mean median )
trimmean `x'TrimPureNo, percent(1)

tabstat `x'TrimCeo, statistics( mean median )
trimmean `x'TrimCeo, percent(1)
tabstat `x'TrimCeoNo, statistics( mean median )
trimmean `x'TrimCeoNo, percent(1)

qui drop `x'TrimFamilyNo `x'TrimEntr `x'TrimNotEntr
`x'TrimPureNo `x'TrimCeo `x'TrimCeoNo

}

*SMALL FIRMS
foreach x of varlist TotAssInfl RevenueInfl Empl Age RoA RoE SalesGrowthPerc
DebtRatio {

gen `x'TrimFamilyNo = `x' if Family == 0 & Small == 1

gen `x'TrimEntr = `x' if PureFamEntr == 1 & Small == 1
gen `x'TrimNotEntr = `x' if PureFamily == 1 & PureFamEntr == 0 &
Small == 1
gen `x'TrimPureNo = `x' if PureFamily == 0 & Family ==1 & Small
== 1

gen `x'TrimCeo = `x' if FamCeo == 1 & Family ==1 & Small == 1
gen `x'TrimCeoNo = `x' if FamCeo == 0 & Family ==1 & Small == 1

tabstat `x', statistics( mean median )
trimmean `x', percent(1)

tabstat `x'TrimFamilyNo, statistics( mean median )
trimmean `x'TrimFamilyNo, percent(1)

tabstat `x'TrimEntr, statistics( mean median )
trimmean `x'TrimEntr, percent(1)
tabstat `x'TrimNotEntr, statistics( mean median )
trimmean `x'TrimNotEntr, percent(1)
tabstat `x'TrimPureNo, statistics( mean median )
trimmean `x'TrimPureNo, percent(1)

tabstat `x'TrimCeo, statistics( mean median )
trimmean `x'TrimCeo, percent(1)
tabstat `x'TrimCeoNo, statistics( mean median )
trimmean `x'TrimCeoNo, percent(1)

```

```

        qui drop `x'TrimFamilyNo `x'TrimEntr `x'TrimNotEntr
`x'TrimPureNo `x'TrimCeo `x'TrimCeoNo
    }
    *LARGE FIRMS
    foreach x of varlist TotAssInfl RevenueInfl Empl Age RoA RoE SalesGrowthPerc
    DebtRatio {

        gen `x'TrimFamilyNo = `x' if Family == 0 & Large == 1

        gen `x'TrimEntr = `x' if PureFamEntr == 1 & Large == 1
        gen `x'TrimNotEntr = `x' if PureFamily == 1 & PureFamEntr == 0 &
Large == 1
        gen `x'TrimPureNo = `x' if PureFamily == 0 & Family ==1 & Large
== 1

        gen `x'TrimCeo = `x' if FamCeo == 1 & Family ==1 & Large == 1
        gen `x'TrimCeoNo = `x' if FamCeo == 0 & Family ==1 & Large == 1

        tabstat `x', statistics( mean median )
        trimmean `x', percent(1)

        tabstat `x'TrimFamilyNo, statistics( mean median )
        trimmean `x'TrimFamilyNo, percent(1)

        tabstat `x'TrimEntr, statistics( mean median )
        trimmean `x'TrimEntr, percent(1)
        tabstat `x'TrimNotEntr, statistics( mean median )
        trimmean `x'TrimNotEntr, percent(1)
        tabstat `x'TrimPureNo, statistics( mean median )
        trimmean `x'TrimPureNo, percent(1)

        tabstat `x'TrimCeo, statistics( mean median )
        trimmean `x'TrimCeo, percent(1)
        tabstat `x'TrimCeoNo, statistics( mean median )
        trimmean `x'TrimCeoNo, percent(1)

        qui drop `x'TrimFamilyNo `x'TrimEntr `x'TrimNotEntr
`x'TrimPureNo `x'TrimCeo `x'TrimCeoNo
    }

    //YEARLY//

    *TRIMMED SALES GROWTH
    foreach i of num 2001/2015{
        disp `i' " Sales Growth"
        disp "Family:"
        trimmean SalesGrowthPerc if Year == `i' & Family == 1,
percent(1)
        disp "Non-family:"
    }

```

```

        trimmean SalesGrowthPerc if Year == `i' & Family == 0,
percent(1)
    }
*TRIMMED ROA
foreach i of num 2001/2015{
    disp `i' " RoA"
    disp "Family:"
    trimmean RoA if Year == `i' & Family == 1, percent(1)
    disp "Non-family:"
    trimmean RoA if Year == `i' & Family == 0, percent(1)
}
*TRIMMED DEBT RATIO
foreach i of num 2000/2015{
    disp `i' " Debt Ratio"
    disp "Family:"
    trimmean DebtRatio if Year == `i' & Family == 1, percent(1)
    disp "Non-family:"
    trimmean DebtRatio if Year == `i' & Family == 0, percent(1)
}
*TRIMMED TOTAL ASSETS GROWTH
foreach i of num 2001/2015{
    disp `i' " Total Assets Growth"
    disp "Family:"
    trimmean LnAssetsGrowth if Year == `i' & Family == 1,
percent(1)
    disp "Non-family:"
    trimmean LnAssetsGrowth if Year == `i' & Family == 0,
percent(1)
}
*EMPLOYEE GROWTH
foreach i of num 2001/2015{
    disp `i' " Employee Growth"
    disp "Family:"
    trimmean EmplGrowthPerc if Year == `i' & Family == 1,
percent(1)
    disp "Non-family:"
    trimmean EmplGrowthPerc if Year == `i' & Family == 0,
percent(1)
}

*SMALL / LARGE
{
*SMALL TRIMMED SALES GROWTH
foreach i of num 2001/2015{
    disp `i' " Sales Growth Small"
    disp "Family:"
    trimmean SalesGrowthPerc if Year == `i' & Family == 1 &
Small == 1, percent(1)
}

```

```

disp "Non-family:"
trimmean SalesGrowthPerc if Year == `i' & Family == 0 &
Small == 1, percent(1)
}
*SMALL TRIMMED ROA
foreach i of num 2001/2015{
disp `i' " RoA Small"
disp "Family:"
trimmean RoA if Year == `i' & Family == 1 & Small == 1,
percent(1)

disp "Non-family:"
trimmean RoA if Year == `i' & Family == 0 & Small == 1,
percent(1)
}
*SMALL TRIMMED DEBT RATIO
foreach i of num 2000/2015{
disp `i' " Debt Ratio Small"
disp "Family:"
trimmean DebtRatio if Year == `i' & Family == 1 & Small ==
1, percent(1)

disp "Non-family:"
trimmean DebtRatio if Year == `i' & Family == 0 & Small ==
1, percent(1)
}
*SMALL TRIMMED TOTAL ASSETS GROWTH
foreach i of num 2001/2015{
disp `i' " Total Assets Growth Small"
disp "Family:"
trimmean LnAssetsGrowth if Year == `i' & Family == 1 &
Small == 1, percent(1)
disp "Non-family:"
trimmean LnAssetsGrowth if Year == `i' & Family == 0 &
Small == 1, percent(1)
}
*SMALL EMPLOYEE GROWTH
foreach i of num 2001/2015{
disp `i' " Employee Growth Small"
disp "Family:"
trimmean EmplGrowthPerc if Year == `i' & Family == 1 &
Small == 1, percent(1)
disp "Non-family:"
trimmean EmplGrowthPerc if Year == `i' & Family == 0 &
Small == 1, percent(1)
}
*LARGE TRIMMED SALES GROWTH
foreach i of num 2001/2015{
disp `i' " Sales Growth Large"
disp "Family:"

```

```

        trimmean SalesGrowthPerc if Year == `i' & Family == 1 &
Large == 1, percent(1)
        disp "Non-family:"
        trimmean SalesGrowthPerc if Year == `i' & Family == 0 &
Large == 1, percent(1)
    }
    *LARGE TRIMMED ROA
    foreach i of num 2001/2015{
        disp `i' " RoA Large"
        disp "Family:"
        trimmean RoA if Year == `i' & Family == 1 & Large == 1,
percent(1)
        disp "Non-family:"
        trimmean RoA if Year == `i' & Family == 0 & Large == 1,
percent(1)
    }
    *LARGE TRIMMED DEBT RATIO
    foreach i of num 2000/2015{
        disp `i' " Debt Ratio Large"
        disp "Family:"
        trimmean DebtRatio if Year == `i' & Family == 1 & Large ==
1, percent(1)
        disp "Non-family:"
        trimmean DebtRatio if Year == `i' & Family == 0 & Large ==
1, percent(1)
    }
    *LARGE TRIMMED TOTAL ASSETS GROWTH
    foreach i of num 2001/2015{
        disp `i' " Total Assets Growth Large"
        disp "Family:"
        trimmean LnAssetsGrowth if Year == `i' & Family == 1 &
Large == 1, percent(1)
        disp "Non-family:"
        trimmean LnAssetsGrowth if Year == `i' & Family == 0 &
Large == 1, percent(1)
    }
    *LARGE EMPLOYEE GROWTH
    foreach i of num 2001/2015{
        disp `i' " Employee Growth Large"
        disp "Family:"
        trimmean EmplGrowthPerc if Year == `i' & Family == 1 &
Large == 1, percent(1)
        disp "Non-family:"
        trimmean EmplGrowthPerc if Year == `i' & Family == 0 &
Large == 1, percent(1)
    }
}
*/
}

```

```

//TRIMMING FOR REGRESSIONS
// SalesGrowthPerc
gen TSalesGrowthPerc = .
sum SalesGrowthPerc, d
replace TSalesGrowthPerc = SalesGrowthPerc if SalesGrowthPerc < r(p99)

// ROBUST TESTING
{
gen TLnAssetsGrowth = .
sum LnAssetsGrowth, d
replace TLnAssetsGrowth = LnAssetsGrowth if LnAssetsGrowth < r(p99)

// Regression
xtreg TLnAssetsGrowth Family PureFamily RoA Age AgeSq BNP L.InvestProxy
L.RetRate L.CashRate L.DebtRatio Empl $Industries $Counties i.Year

// FIXED EFFECTS & RANDOM EFFECTS
xtreg TLnAssetsGrowth Family BNP L.InvestProxy L.RetRate L.CashRate RoA
L.DebtRatio Age AgeSq Empl i.Year, fe
estimate store fe

xtreg TLnAssetsGrowth Family BNP L.InvestProxy L.RetRate L.CashRate RoA
L.DebtRatio Age AgeSq Empl i.Year , re
estimate store re
hausman fe re

// PROPENSITY SCORE MATCHING
//Full Sample
capture teffects psmatch (TLnAssetsGrowth) (Family InvestProxy CashRate RoA
DebtRatio Age Empl), osample(error)
drop if error==1
teffects psmatch (TLnAssetsGrowth) (Family InvestProxy CashRate RoA
DebtRatio Age Empl)

//HECKMAN
heckman TLnAssetsGrowth InvestProxy CashRate RoA DebtRatio Age Empl,
select(Family = InvestProxy CashRate RoA DebtRatio Age Empl)
}

rename LnAssets Size

gen FamOwn = Family*LargestFamOwn

//REGRESSION

xtreg TSalesGrowthPerc Family
xtreg TSalesGrowthPerc Family RoA L.Size Age AgeSq BNP

```

```

xtreg TSalesGrowthPerc Family RoA L.Size Age AgeSq BNP L.InvestProxy
L.RetRate L.CashRate L.DebtRatio Empl
xtreg TSalesGrowthPerc Family RoA L.Size Age AgeSq BNP L.InvestProxy
L.RetRate L.CashRate L.DebtRatio Empl $Industries
xtreg TSalesGrowthPerc Family RoA L.Size Age AgeSq BNP L.InvestProxy
L.RetRate L.CashRate L.DebtRatio Empl $Industries $Counties
xtreg TSalesGrowthPerc Family RoA L.Size Age AgeSq BNP L.InvestProxy
L.RetRate L.CashRate L.DebtRatio Empl $Industries $Counties i.Year
xtreg TSalesGrowthPerc Family Supermajority RoA L.Size Age AgeSq BNP
L.InvestProxy L.RetRate L.CashRate L.DebtRatio Empl $Industries $Counties
i.Year
xtreg TSalesGrowthPerc Family PureFamily RoA L.Size Age AgeSq BNP
L.InvestProxy L.RetRate L.CashRate L.DebtRatio Empl $Industries $Counties
i.Year

```

* FIXED EFFECTS & RANDOM EFFECTS

```

xtreg TSalesGrowthPerc Family BNP L.Size L.InvestProxy L.RetRate L.CashRate
RoA L.DebtRatio Age AgeSq Empl i.Year, fe
estimate store fe
xtreg TSalesGrowthPerc L.Family BNP L.Size L.InvestProxy L.RetRate L.CashRate
RoA L.DebtRatio Age AgeSq Empl i.Year, fe

```

```

xtreg TSalesGrowthPerc Family BNP L.Size L.InvestProxy L.RetRate L.CashRate
RoA L.DebtRatio Age AgeSq Empl i.Year , re
estimate store re
hausman fe re

```

* PROPENSITY SCORE MATCHING

```

{
/*
//Full Sample
capture teffects psmatch (TSalesGrowthPerc) (Family Size InvestProxy CashRate
RoA DebtRatio Age Empl), osample(error)
drop if error==1
teffects psmatch (TSalesGrowthPerc) (Family Size InvestProxy CashRate RoA
DebtRatio Age Empl)

//Small Sample
capture teffects psmatch (TSalesGrowthPerc) (Family Size InvestProxy CashRate
RoA DebtRatio Age Empl) if Small==1, osample(error1)
drop if error1==1
teffects psmatch (TSalesGrowthPerc) (Family Size InvestProxy CashRate RoA
DebtRatio Age Empl) if Small==1

//Large Sample

```

```
capture teffects psmatch (TSalesGrowthPerc) (Family Size InvestProxy CashRate
RoA DebtRatio Age Empl) if Large==1, osample(error2)
drop if error2==1
teffects psmatch (TSalesGrowthPerc) (Family Size InvestProxy CashRate RoA
DebtRatio Age Empl) if Large==1
*/
}
```

```
/*
*HECKMAN
heckman TSalesGrowthPerc Size InvestProxy CashRate RoA DebtRatio Age Empl,
select(Family = Size InvestProxy CashRate RoA DebtRatio Age Empl)
*/
```

****COHORT****

```
drop if Year < 2010
drop if Age > 7
drop if Year == 2010 & Age != 2
drop if Year == 2011 & Age != 3
drop if Year == 2012 & Age != 4
drop if Year == 2013 & Age != 5
drop if Year == 2014 & Age != 6
drop if Year == 2015 & Age != 7
```

```
{
/*
*Family in every year 2010-2015
```

```
bysort Id (Year) : drop if Family !=1 & Year == 2010 | Family != 1 & Year == 2011
| Family != 1 & Year == 2012 | Family != 1 & Year == 2013 | Family != 1 & Year ==
2014 | Family != 1 & Year == 2015
```

```
bysort Id (Year) : drop if _N==1
sum TSalesGrowthPerc if Year == 2010, d
sum TSalesGrowthPerc if Year == 2011, d
sum TSalesGrowthPerc if Year == 2012, d
sum TSalesGrowthPerc if Year == 2013, d
sum TSalesGrowthPerc if Year == 2014, d
sum TSalesGrowthPerc if Year == 2015, d
gen Cohort = 1
```

```
export excel Id Year TSalesGrowthPerc Age Cohort using
"/Users/Espen/Documents/BI/Master Thesis/CohortFamily.xlsx",
firstrow(variables) replace
*export excel Id Year TSalesGrowthPerc Age Cohort using "F:\My
Documents\CohortFamily.xlsx", firstrow(variables) replace
```

```
*Non-Family in every year 2010-2015
```



```

bysort Id (Year) : drop if Family ==1 & Year == 2010 | Family == 1 & Year ==
2011 | Family == 1 & Year == 2012 | Family == 1 & Year == 2013 | Family == 1 &
Year == 2014 | Family == 1 & Year == 2015
bysort Id (Year) : drop if _N==1
sum TSalesGrowthPerc if Year == 2010, d
sum TSalesGrowthPerc if Year == 2011, d
sum TSalesGrowthPerc if Year == 2012, d
sum TSalesGrowthPerc if Year == 2013, d
sum TSalesGrowthPerc if Year == 2014, d
sum TSalesGrowthPerc if Year == 2015, d
gen Cohort = 0
export excel Id Year TSalesGrowthPerc Age Cohort using
"/Users/Espen/Documents/BI/Master Thesis/CohortNonFamily.xlsx",
firstrow(variables) replace
*export excel Id Year TSalesGrowthPerc Age Cohort using "F:\My
Documents\CohortNonFamily.xlsx", firstrow(variables) replace
*/

/*
*TESTING FAMILY IN EVERY YEAR VERSUS NON-FAMILY IN EVERY YEAR
*N.B. Must have created Cohort.dta from CohortFamily and CohortNonFamily
first
clear all
cd "/Users/Espen/Documents/BI/Master Thesis"
*cd "F:\My Documents\
sysuse Cohort
ttest TSalesGrowthPerc, by(Cohort)
*/
}

*Family --> Non-Family
keep if Id == 769038729 | Id == 769139349| Id == 769259379| Id ==
769394479 | Id == 769473519| Id == 769553399| Id == 769655629| Id
== 769695139 | Id == 770072289| Id == 770379959| Id ==
869006371| Id == 869025349| Id == 869044486| Id ==
869060163| Id == 869100665| Id == 869102052| Id ==
869140772| Id == 869147935| Id == 869162160| Id ==
869204894| Id == 869277769| Id == 869287856| Id ==
869301211| Id == 869309009| Id == 869332625| Id ==
869349046| Id == 869367327| Id == 869381188| Id ==
869393000| Id == 869411055| Id == 869416893| Id ==
869458333| Id == 869462358| Id == 869472554| Id ==
869474229| Id == 869480294| Id == 869480759| Id ==
869481938| Id == 869497753| Id == 869526472| Id ==
869543223| Id == 869599301| Id == 869624812| Id ==
869634621| Id == 869652157| Id == 869662187| Id ==
869685892| Id == 869691497| Id == 869696567| Id ==
869697044| Id == 869707871| Id == 869708542| Id ==
869716352| Id == 869718916| Id == 869721372| Id ==

```

869726479| Id == 869786763| Id == 869825783| Id ==
869896642| Id == 869983170| Id == 870029409| Id ==
870044347| Id == 870052936| Id == 870058253| Id ==
870061795| Id == 870118958| Id == 870128934| Id ==
870134389| Id == 870156277| Id == 870202636| Id ==
870248840| Id == 870282016| Id == 870298491| Id ==
870322347| Id == 870331152| Id == 870334167| Id ==
870435025| Id == 870454894| Id == 870469107

sum TSalesGrowthPerc if Family == 1
sum TSalesGrowthPerc if Family == 0
ttest TSalesGrowthPerc, by(Family)

*Non-Family --> Family

keep if Id == 769161839| Id == 769215129| Id == 769318689| Id ==
769394479| Id == 769679149| Id == 769913029| Id == 769970389| Id ==
770084419| Id == 770154909| Id == 770216669| Id == 770345239| Id ==
869006371| Id == 869037857| Id == 869056287| Id == 869100665| Id ==
869106349| Id == 869170036| Id == 869196464| Id == 869235281| Id ==
869243041| Id == 869266632| Id == 869279993| Id == 869292258| Id ==
869313128| Id == 869320189| Id == 869323998| Id == 869324808| Id ==
869329048| Id == 869341434| Id == 869360450| Id == 869373177| Id ==
869378886| Id == 869383509| Id == 869391186| Id == 869393000| Id ==
869423596| Id == 869428818| Id == 869447175| Id == 869457713| Id ==
869464315| Id == 869466615| Id == 869480294| Id == 869481938| Id ==
869495880| Id == 869546384| Id == 869553880| Id == 869561523| Id ==
869567539| Id == 869583957| Id == 869584000| Id == 869630548| Id ==
869634218| Id == 869636731| Id == 869665632| Id == 869670330| Id ==
869671000| Id == 869685892| Id == 869696173| Id == 869710870| Id ==
869712800| Id == 869713458| Id == 869717634| Id == 869724855| Id ==
869729178| Id == 869738615| Id == 869746178| Id == 869751806| Id ==
869758345| Id == 869777272| Id == 869778346| Id == 869786763| Id ==
869794288| Id == 869822454| Id == 869823985| Id == 869825783| Id ==
869829323| Id == 869925640| Id == 869930147| Id == 869931140| Id ==
869958326| Id == 869967244| Id == 869972249| Id == 869976093| Id ==
869977447| Id == 869984701| Id == 869997886| Id == 870022937| Id ==
870025428| Id == 870041480| Id == 870044347| Id == 870044882| Id ==
870050572| Id == 870053071| Id == 870058304| Id == 870071930| Id ==
870076389| Id == 870123863| Id == 870125212| Id == 870129158| Id ==
870134389| Id == 870148392| Id == 870176003| Id == 870179085| Id ==
870198211| Id == 870200163| Id == 870202860| Id == 870208368| Id ==
870217805| Id == 870242354| Id == 870259718| Id == 870309428| Id ==
870322347| Id == 870331152| Id == 870334167| Id == 870353417| Id ==
870363799| Id == 870396726| Id == 870469107

sum TSalesGrowthPerc if Family == 1
sum TSalesGrowthPerc if Family == 0
ttest TSalesGrowthPerc, by(Family)

/*

*These become non-fam in 2012 (#35):

```
keep if Id == 769139349 | Id == 769655629 | Id == 869006371 | Id ==  
869025349 | Id == 869044486 | Id == 869147935 | Id == 869204894 | Id ==  
869287856 | Id == 869332625 | Id == 869367327 | Id == 869381188 | Id ==  
869411055 | Id == 869458333 | Id == 869462358 | Id == 869474229 | Id ==  
869480759 | Id == 869526472 | Id == 869543223 | Id == 869599301 | Id ==  
869624812 | Id == 869652157 | Id == 869662187 | Id == 869716352 | Id ==  
869718916 | Id == 869896642 | Id == 869983170 | Id == 870029409 | Id ==  
870052936 | Id == 870156277 | Id == 870202636 | Id == 870322347 | Id ==  
870331152 | Id == 870435025 | Id == 870454894 | Id == 870469107  
*/
```