CODE – INDIVIDUAL STOCK MOMENTUM WITHIN INDUSTRIES

Table of Contents

//Covid-19 - Disruption

use "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\Only data.dta"

//Set up the Import Options and import the data//
//small letters on all variables
rename PERMNO permno
rename SHRCD shrcd
rename EXCHCD exchcd
rename SICCD siccd
rename PRC prc
rename RET ret
rename SHROUT shrout

//Create date variable mofd, and make data readable.
gen mofd = mofd(date)
sort date
keep if inrange(mofd, 714, 736)
format mofd %tm
xtset permno mofd, monthly
unique permno

//Data cleaning//
//Remove top 1% return and bottom 1% return
winsor2 ret, cuts(1 99) trim by(mofd)
replace ret=ret_tr
drop ret_tr

//keep exchange code if it is in NYSE, NASDAQ, NYSE American
keep if exchcd == 1 | exchcd == 2 | exchcd == 3

//keep share code if they are ordinary common securities
keep if shrcd == 10 | shrcd == 11

//Remove share price below 5
drop if prc < 5

//Create new variable for two-digit SIC codes
gen sic = substr(string(siccd),1,2)

//Convert sic to numeric variable
destring sic, replace

//recode sic to numeric industry
recode sic (10/14=1) (20=2) (22/23=3) (26=4) (28=5) (29=6) (32=7) (33=8) (34=9) (35=10) (36=11) (37=12) (38/39=13) (40=14) (41/44=15) (46/47=15) (45=16) (49=17) (53=18) (50/52=19) (54/59=19) (60/69=20) (73=21) (70/72=22) (74/79=22) (80/86=23) (87/99=24)

//Only keep industries we want to analyze
keep if sic == 1 | sic == 2 | sic == 3 | sic == 4 | sic == 5 | sic == 6 | sic == 7| sic == 8 | sic == 9 | sic == 10 | sic == 11 | sic == 12 | sic == 13 | sic == 14 | sic == 15 | sic == 16 | sic == 17 | sic == 18 | sic == 19 | sic == 20 | sic == 21 | sic == 22 | sic == 23 | sic == 24

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret1 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==1
by permno: gen preret2 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==2
by permno: gen preret3 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==3
by permno: gen preret4 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==4
by permno: gen preret5 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==5
by permno: gen preret6 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==6
by permno: gen preret7 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==7
by permno: gen preret8 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==8
by permno: gen preret9 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==9
by permno: gen preret10 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==10
by permno: gen preret11 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==11
by permno: gen preret12 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==12
by permno: gen preret13 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==13
by permno: gen preret14 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==14
by permno: gen preret15 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==15
by permno: gen preret16 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==16
by permno: gen preret17 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==17
by permno: gen preret18 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==18
by permno: gen preret19 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==19
by permno: gen preret20 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==20
by permno: gen preret21 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==21
by permno: gen preret22 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==22
by permno: gen preret23 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==23
by permno: gen preret24 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==24

sort mofd

//Holding period: Make returns for 6 months, expressed returns in percentage points
sort permno mofd
by permno: gen fret1 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==1
by permno: gen fret2 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==2
by permno: gen fret3 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==3
by permno: gen fret4 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==4
by permno: gen fret5 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==5
by permno: gen fret6 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==6
by permno: gen fret7 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==7
by permno: gen fret8 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==8
by permno: gen fret9 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==9
by permno: gen fret10 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==10
by permno: gen fret11 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==11
by permno: gen fret12 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==12
by permno: gen fret13 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==13
by permno: gen fret14 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==14

```
by permno: gen fret15 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==15
by permno: gen fret16 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==16
by permno: gen fret17 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==17
by permno: gen fret18 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==18
by permno: gen fret19 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==19
by permno: gen fret20 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==20
by permno: gen fret21 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==21
by permno: gen fret22 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==22
by permno: gen fret23 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==23
by permno: gen fret24 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==24

//Rank stocks based on past 6-month return: the command creates the variable p_6 which ranks each
stock based on their past 6 months return every month, into 10 quantiles//
astile port1 = preret1, by(mofd) nq(10), if sic==1
astile port2 = preret2, by(mofd) nq(10), if sic==2
astile port3 = preret3, by(mofd) nq(10), if sic==3
astile port4 = preret4, by(mofd) nq(10), if sic==4
astile port5 = preret5, by(mofd) nq(10), if sic==5
astile port6 = preret6, by(mofd) nq(10), if sic==6
astile port7 = preret7, by(mofd) nq(10), if sic==7
astile port8 = preret8, by(mofd) nq(10), if sic==8
astile port9 = preret9, by(mofd) nq(10), if sic==9
astile port10 = preret10, by(mofd) nq(10), if sic==10
astile port11 = preret11, by(mofd) nq(10), if sic==11
astile port12 = preret12, by(mofd) nq(10), if sic==12
astile port13 = preret13, by(mofd) nq(10), if sic==13
astile port14 = preret14, by(mofd) nq(10), if sic==14
astile port15 = preret15, by(mofd) nq(10), if sic==15
astile port16 = preret16, by(mofd) nq(10), if sic==16
astile port17 = preret17, by(mofd) nq(10), if sic==17
astile port18 = preret18, by(mofd) nq(10), if sic==18
astile port19 = preret19, by(mofd) nq(10), if sic==19
astile port20 = preret20, by(mofd) nq(10), if sic==20
astile port21 = preret21, by(mofd) nq(10), if sic==21
astile port22 = preret22, by(mofd) nq(10), if sic==22
astile port23 = preret23, by(mofd) nq(10), if sic==23
astile port24 = preret24, by(mofd) nq(10), if sic==24

//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen p1_1=fret1 if port1==1
bysort mofd: gen p10_1=fret1 if port1==10
bysort mofd: gen p1_2=fret2 if port2==1
bysort mofd: gen p10_2=fret2 if port2==10
bysort mofd: gen p1_3=fret3 if port3==1
bysort mofd: gen p10_3=fret3 if port3==10
bysort mofd: gen p1_4=fret4 if port4==1
bysort mofd: gen p10_4=fret4 if port4==10
bysort mofd: gen p1_5=fret5 if port5==1
bysort mofd: gen p10_5=fret5 if port5==10
```

```
bysort mofd: gen p1_6=fret6 if port6==1
bysort mofd: gen p10_6=fret6 if port6==10
bysort mofd: gen p1_7=fret7 if port7==1
bysort mofd: gen p10_7=fret7 if port7==10
bysort mofd: gen p1_8=fret8 if port8==1
bysort mofd: gen p10_8=fret8 if port8==10
bysort mofd: gen p1_9=fret9 if port9==1
bysort mofd: gen p10_9=fret9 if port9==10
bysort mofd: gen p1_10=fret10 if port10==1
bysort mofd: gen p10_10=fret10 if port10==10
bysort mofd: gen p1_11=fret11 if port11==1
bysort mofd: gen p10_11=fret11 if port11==10
bysort mofd: gen p1_12=fret12 if port12==1
bysort mofd: gen p10_12=fret12 if port12==10
bysort mofd: gen p1_13=fret13 if port13==1
bysort mofd: gen p10_13=fret13 if port13==10
bysort mofd: gen p1_14=fret14 if port14==1
bysort mofd: gen p10_14=fret14 if port14==10
bysort mofd: gen p1_15=fret15 if port15==1
bysort mofd: gen p10_15=fret15 if port15==10
bysort mofd: gen p1_16=fret16 if port16==1
bysort mofd: gen p10_16=fret16 if port16==10
bysort mofd: gen p1_17=fret17 if port17==1
bysort mofd: gen p10_17=fret17 if port17==10
bysort mofd: gen p1_18=fret18 if port18==1
bysort mofd: gen p10_18=fret18 if port18==10
bysort mofd: gen p1_19=fret19 if port19==1
bysort mofd: gen p10_19=fret19 if port19==10
bysort mofd: gen p1_20=fret20 if port20==1
bysort mofd: gen p10_20=fret20 if port20==10
bysort mofd: gen p1_21=fret21 if port21==1
bysort mofd: gen p10_21=fret21 if port21==10
bysort mofd: gen p1_22=fret22 if port22==1
bysort mofd: gen p10_22=fret22 if port22==10
bysort mofd: gen p1_23=fret23 if port23==1
bysort mofd: gen p10_23=fret23 if port23==10
bysort mofd: gen p1_24=fret24 if port24==1
bysort mofd: gen p10_24=fret24 if port24==10
```

```
//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
bysort mofd: egen meanp1_1=mean(p1_1)
bysort mofd: egen meanp10_1=mean(p10_1)
bysort mofd: egen meanp1_2=mean(p1_2)
bysort mofd: egen meanp10_2=mean(p10_2)
bysort mofd: egen meanp1_3=mean(p1_3)
bysort mofd: egen meanp10_3=mean(p10_3)
bysort mofd: egen meanp1_4=mean(p1_4)
bysort mofd: egen meanp10_4=mean(p10_4)
bysort mofd: egen meanp1_5=mean(p1_5)
bysort mofd: egen meanp10_5=mean(p10_5)
```

bysort mofd: egen meanp1_6=mean(p1_6)
bysort mofd: egen meanp10_6=mean(p10_6)
bysort mofd: egen meanp1_7=mean(p1_7)
bysort mofd: egen meanp10_7=mean(p10_7)
bysort mofd: egen meanp1_8=mean(p1_8)
bysort mofd: egen meanp10_8=mean(p10_8)
bysort mofd: egen meanp1_9=mean(p1_9)
bysort mofd: egen meanp10_9=mean(p10_9)
bysort mofd: egen meanp1_10=mean(p1_10)
bysort mofd: egen meanp10_10=mean(p10_10)
bysort mofd: egen meanp1_11=mean(p1_11)
bysort mofd: egen meanp10_11=mean(p10_11)
bysort mofd: egen meanp1_12=mean(p1_12)
bysort mofd: egen meanp10_12=mean(p10_12)
bysort mofd: egen meanp1_13=mean(p1_13)
bysort mofd: egen meanp10_13=mean(p10_13)
bysort mofd: egen meanp1_14=mean(p1_14)
bysort mofd: egen meanp10_14=mean(p10_14)
bysort mofd: egen meanp1_15=mean(p1_15)
bysort mofd: egen meanp10_15=mean(p10_15)
bysort mofd: egen meanp1_16=mean(p1_16)
bysort mofd: egen meanp10_16=mean(p10_16)
bysort mofd: egen meanp1_17=mean(p1_17)
bysort mofd: egen meanp10_17=mean(p10_17)
bysort mofd: egen meanp1_18=mean(p1_18)
bysort mofd: egen meanp10_18=mean(p10_18)
bysort mofd: egen meanp1_19=mean(p1_19)
bysort mofd: egen meanp10_19=mean(p10_19)
bysort mofd: egen meanp1_20=mean(p1_20)
bysort mofd: egen meanp10_20=mean(p10_20)
bysort mofd: egen meanp1_21=mean(p1_21)
bysort mofd: egen meanp10_21=mean(p10_21)
bysort mofd: egen meanp1_22=mean(p1_22)
bysort mofd: egen meanp10_22=mean(p10_22)
bysort mofd: egen meanp1_23=mean(p1_23)
bysort mofd: egen meanp10_23=mean(p10_23)
bysort mofd: egen meanp1_24=mean(p1_24)
bysort mofd: egen meanp10_24=mean(p10_24)

***//* Create momentum return, winners-losers**
gen mom1=meanp10_1-meanp1_1
gen mom2=meanp10_2-meanp1_2
gen mom3=meanp10_3-meanp1_3
gen mom4=meanp10_4-meanp1_4
gen mom5=meanp10_5-meanp1_5
gen mom6=meanp10_6-meanp1_6
gen mom7=meanp10_7-meanp1_7
gen mom8=meanp10_8-meanp1_8
gen mom9=meanp10_9-meanp1_9
gen mom10=meanp10_10-meanp1_10

```
gen mom11=meanp10_11-meanp1_11
gen mom12=meanp10_12-meanp1_12
gen mom13=meanp10_13-meanp1_13
gen mom14=meanp10_14-meanp1_14
gen mom15=meanp10_15-meanp1_15
gen mom16=meanp10_16-meanp1_16
gen mom17=meanp10_17-meanp1_17
gen mom18=meanp10_18-meanp1_18
gen mom19=meanp10_19-meanp1_19
gen mom20=meanp10_20-meanp1_20
gen mom21=meanp10_21-meanp1_21
gen mom22=meanp10_22-meanp1_22
gen mom23=meanp10_23-meanp1_23
gen mom24=meanp10_24-meanp1_24


//drop duplicates, one average return for each month
duplicates drop mofd, force


// Table of descriptive statistics
//mean return is expressed as monthly return ,in percentage form
local varlist mom1 mom2 mom3 mom4 mom5 mom6 mom7 mom8 mom9 mom10 mom11 mom12
mom13 mom14 mom15 mom16 mom17 mom18 mom19 mom20 mom21 mom22 mom23 mom24
local n : word count `varlist'
matrix define A = J(`n',3,.)
mat rownames A = `varlist'
mat colnames A = Mean SD T
local row = 1
foreach var of varlist `varlist' {
   qui summarize `var'
   mat A[`row',1] = r(mean)/6*100
   mat A[`row',2] = r(sd)
   qui reg `var'
   mat A[`row',3] = _b[_cons] / _se[_cons]
   local ++row
}
mat list A, format(%9.2f)
```

//Covid-19 – Recession Period

use "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\Only data.dta"

//Set up the Import Options and import the data//
//small letters on all variables
rename PERMNO permno
rename SHRCD shrcd
rename EXCHCD exchcd
rename SICCD siccd
rename PRC prc
rename RET ret
rename SHROUT shrout

//Create date variable mofd, and make data readable.
gen mofd = mofd(date)
sort date
keep if inrange(mofd, 715, 728)
format mofd %tm
xtset permno mofd, monthly
unique permno

//Data cleaning//
//Remove top 1% return and bottom 1% return
winsor2 ret, cuts(1 99) trim by(mofd)
replace ret=ret_tr
drop ret_tr

//keep exchange code if it is in NYSE, NASDAQ, NYSE American
keep if exchcd == 1 | exchcd == 2 | exchcd == 3

//keep share code if they are ordinary common securities
keep if shrcd == 10 | shrcd == 11

//Remove share price below 5
drop if prc < 5

//Create new variable for two-digit SIC codes
gen sic = substr(string(siccd),1,2)

//Convert sic to numeric variable
destring sic, replace

//recode sic to numeric industry

recode sic (10/14=1) (20=2) (22/23=3) (26=4) (28=5) (29=6) (32=7) (33=8) (34=9) (35=10) (36=11) (37=12) (38/39=13) (40=14) (41/44=15) (46/47=15) (45=16) (49=17) (53=18) (50/52=19) (54/59=19) (60/69=20) (73=21) (70/72=22) (74/79=22) (80/86=23) (87/99=24)

//Only keep industries we want to analyze
keep if sic == 1 | sic == 2 | sic == 3 | sic == 4 | sic == 5 | sic == 6 | sic == 7| sic == 8 | sic == 9 | sic == 10 | sic == 11 | sic == 12 | sic == 13 | sic == 14 | sic == 15 | sic == 16 | sic == 17 | sic == 18 | sic == 19 | sic == 20 | sic == 21 | sic == 22 | sic == 23 | sic == 24

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret1 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==1
by permno: gen preret2 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==2
by permno: gen preret3 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==3
by permno: gen preret4 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==4
by permno: gen preret5 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==5
by permno: gen preret6 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==6
by permno: gen preret7 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==7
by permno: gen preret8 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==8
by permno: gen preret9 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==9
by permno: gen preret10 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==10
by permno: gen preret11 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==11
by permno: gen preret12 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==12
by permno: gen preret13 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==13
by permno: gen preret14 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==14
by permno: gen preret15 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==15
by permno: gen preret16 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==16
by permno: gen preret17 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==17
by permno: gen preret18 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==18
by permno: gen preret19 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==19
by permno: gen preret20 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==20
by permno: gen preret21 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==21
by permno: gen preret22 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==22
by permno: gen preret23 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==23
by permno: gen preret24 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==24

sort mofd

//Holding period: Make returns for 6 months, expressed returns in percentage points
sort permno mofd
by permno: gen fret1 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==1
by permno: gen fret2 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==2
by permno: gen fret3 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==3
by permno: gen fret4 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==4
by permno: gen fret5 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==5
by permno: gen fret6 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==6
by permno: gen fret7 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==7
by permno: gen fret8 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==8
by permno: gen fret9 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==9
by permno: gen fret10 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==10
by permno: gen fret11 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==11
by permno: gen fret12 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==12

```
by permno: gen fret13 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==13
by permno: gen fret14 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==14
by permno: gen fret15 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==15
by permno: gen fret16 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==16
by permno: gen fret17 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==17
by permno: gen fret18 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==18
by permno: gen fret19 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==19
by permno: gen fret20 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==20
by permno: gen fret21 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==21
by permno: gen fret22 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==22
by permno: gen fret23 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==23
by permno: gen fret24 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==24

//Rank stocks based on past 6-month return: the command creates the variable p_6 which ranks each
stock based on their past 6 months return every month, into 10 quantiles//
astile port1 = preret1, by(mofd) nq(10), if sic==1
astile port2 = preret2, by(mofd) nq(10), if sic==2
astile port3 = preret3, by(mofd) nq(10), if sic==3
astile port4 = preret4, by(mofd) nq(10), if sic==4
astile port5 = preret5, by(mofd) nq(10), if sic==5
astile port6 = preret6, by(mofd) nq(10), if sic==6
astile port7 = preret7, by(mofd) nq(10), if sic==7
astile port8 = preret8, by(mofd) nq(10), if sic==8
astile port9 = preret9, by(mofd) nq(10), if sic==9
astile port10 = preret10, by(mofd) nq(10), if sic==10
astile port11 = preret11, by(mofd) nq(10), if sic==11
astile port12 = preret12, by(mofd) nq(10), if sic==12
astile port13 = preret13, by(mofd) nq(10), if sic==13
astile port14 = preret14, by(mofd) nq(10), if sic==14
astile port15 = preret15, by(mofd) nq(10), if sic==15
astile port16 = preret16, by(mofd) nq(10), if sic==16
astile port17 = preret17, by(mofd) nq(10), if sic==17
astile port18 = preret18, by(mofd) nq(10), if sic==18
astile port19 = preret19, by(mofd) nq(10), if sic==19
astile port20 = preret20, by(mofd) nq(10), if sic==20
astile port21 = preret21, by(mofd) nq(10), if sic==21
astile port22 = preret22, by(mofd) nq(10), if sic==22
astile port23 = preret23, by(mofd) nq(10), if sic==23
astile port24 = preret24, by(mofd) nq(10), if sic==24


//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen p1_1=fret1 if port1==1
bysort mofd: gen p10_1=fret1 if port1==10
bysort mofd: gen p1_2=fret2 if port2==1
bysort mofd: gen p10_2=fret2 if port2==10
bysort mofd: gen p1_3=fret3 if port3==1
bysort mofd: gen p10_3=fret3 if port3==10
bysort mofd: gen p1_4=fret4 if port4==1
bysort mofd: gen p10_4=fret4 if port4==10
bysort mofd: gen p1_5=fret5 if port5==1
bysort mofd: gen p10_5=fret5 if port5==10
bysort mofd: gen p1_6=fret6 if port6==1
```

```
bysort mofd: gen p10_6=fret6 if port6==10
bysort mofd: gen p1_7=fret7 if port7==1
bysort mofd: gen p10_7=fret7 if port7==10
bysort mofd: gen p1_8=fret8 if port8==1
bysort mofd: gen p10_8=fret8 if port8==10
bysort mofd: gen p1_9=fret9 if port9==1
bysort mofd: gen p10_9=fret9 if port9==10
bysort mofd: gen p1_10=fret10 if port10==1
bysort mofd: gen p10_10=fret10 if port10==10
bysort mofd: gen p1_11=fret11 if port11==1
bysort mofd: gen p10_11=fret11 if port11==10
bysort mofd: gen p1_12=fret12 if port12==1
bysort mofd: gen p10_12=fret12 if port12==10
bysort mofd: gen p1_13=fret13 if port13==1
bysort mofd: gen p10_13=fret13 if port13==10
bysort mofd: gen p1_14=fret14 if port14==1
bysort mofd: gen p10_14=fret14 if port14==10
bysort mofd: gen p1_15=fret15 if port15==1
bysort mofd: gen p10_15=fret15 if port15==10
bysort mofd: gen p1_16=fret16 if port16==1
bysort mofd: gen p10_16=fret16 if port16==10
bysort mofd: gen p1_17=fret17 if port17==1
bysort mofd: gen p10_17=fret17 if port17==10
bysort mofd: gen p1_18=fret18 if port18==1
bysort mofd: gen p10_18=fret18 if port18==10
bysort mofd: gen p1_19=fret19 if port19==1
bysort mofd: gen p10_19=fret19 if port19==10
bysort mofd: gen p1_20=fret20 if port20==1
bysort mofd: gen p10_20=fret20 if port20==10
bysort mofd: gen p1_21=fret21 if port21==1
bysort mofd: gen p10_21=fret21 if port21==10
bysort mofd: gen p1_22=fret22 if port22==1
bysort mofd: gen p10_22=fret22 if port22==10
bysort mofd: gen p1_23=fret23 if port23==1
bysort mofd: gen p10_23=fret23 if port23==10
bysort mofd: gen p1_24=fret24 if port24==1
bysort mofd: gen p10_24=fret24 if port24==10
```

```
//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
bysort mofd: egen meanp1_1=mean(p1_1)
bysort mofd: egen meanp10_1=mean(p10_1)
bysort mofd: egen meanp1_2=mean(p1_2)
bysort mofd: egen meanp10_2=mean(p10_2)
bysort mofd: egen meanp1_3=mean(p1_3)
bysort mofd: egen meanp10_3=mean(p10_3)
bysort mofd: egen meanp1_4=mean(p1_4)
bysort mofd: egen meanp10_4=mean(p10_4)
bysort mofd: egen meanp1_5=mean(p1_5)
bysort mofd: egen meanp10_5=mean(p10_5)
bysort mofd: egen meanp1_6=mean(p1_6)
bysort mofd: egen meanp10_6=mean(p10_6)
```

bysort mofd: egen meanp1_7=mean(p1_7)
bysort mofd: egen meanp10_7=mean(p10_7)
bysort mofd: egen meanp1_8=mean(p1_8)
bysort mofd: egen meanp10_8=mean(p10_8)
bysort mofd: egen meanp1_9=mean(p1_9)
bysort mofd: egen meanp10_9=mean(p10_9)
bysort mofd: egen meanp1_10=mean(p1_10)
bysort mofd: egen meanp10_10=mean(p10_10)
bysort mofd: egen meanp1_11=mean(p1_11)
bysort mofd: egen meanp10_11=mean(p10_11)
bysort mofd: egen meanp1_12=mean(p1_12)
bysort mofd: egen meanp10_12=mean(p10_12)
bysort mofd: egen meanp1_13=mean(p1_13)
bysort mofd: egen meanp10_13=mean(p10_13)
bysort mofd: egen meanp1_14=mean(p1_14)
bysort mofd: egen meanp10_14=mean(p10_14)
bysort mofd: egen meanp1_15=mean(p1_15)
bysort mofd: egen meanp10_15=mean(p10_15)
bysort mofd: egen meanp1_16=mean(p1_16)
bysort mofd: egen meanp10_16=mean(p10_16)
bysort mofd: egen meanp1_17=mean(p1_17)
bysort mofd: egen meanp10_17=mean(p10_17)
bysort mofd: egen meanp1_18=mean(p1_18)
bysort mofd: egen meanp10_18=mean(p10_18)
bysort mofd: egen meanp1_19=mean(p1_19)
bysort mofd: egen meanp10_19=mean(p10_19)
bysort mofd: egen meanp1_20=mean(p1_20)
bysort mofd: egen meanp10_20=mean(p10_20)
bysort mofd: egen meanp1_21=mean(p1_21)
bysort mofd: egen meanp10_21=mean(p10_21)
bysort mofd: egen meanp1_22=mean(p1_22)
bysort mofd: egen meanp10_22=mean(p10_22)
bysort mofd: egen meanp1_23=mean(p1_23)
bysort mofd: egen meanp10_23=mean(p10_23)
bysort mofd: egen meanp1_24=mean(p1_24)
bysort mofd: egen meanp10_24=mean(p10_24)

//Create momentum return, winners-losers
gen mom1=meanp10_1-meanp1_1
gen mom2=meanp10_2-meanp1_2
gen mom3=meanp10_3-meanp1_3
gen mom4=meanp10_4-meanp1_4
gen mom5=meanp10_5-meanp1_5
gen mom6=meanp10_6-meanp1_6
gen mom7=meanp10_7-meanp1_7
gen mom8=meanp10_8-meanp1_8
gen mom9=meanp10_9-meanp1_9
gen mom10=meanp10_10-meanp1_10
gen mom11=meanp10_11-meanp1_11
gen mom12=meanp10_12-meanp1_12
gen mom13=meanp10_13-meanp1_13
gen mom14=meanp10_14-meanp1_14

```
gen mom15=meanp10_15-meanp1_15
gen mom16=meanp10_16-meanp1_16
gen mom17=meanp10_17-meanp1_17
gen mom18=meanp10_18-meanp1_18
gen mom19=meanp10_19-meanp1_19
gen mom20=meanp10_20-meanp1_20
gen mom21=meanp10_21-meanp1_21
gen mom22=meanp10_22-meanp1_22
gen mom23=meanp10_23-meanp1_23
gen mom24=meanp10_24-meanp1_24


//drop duplicates, one average return for each month
duplicates drop mofd, force
```

//Table of descriptive statistics

```
//mean return is expressed as monthly return ,in percentage form
local varlist mom1 mom2 mom3 mom4 mom5 mom6 mom7 mom8 mom9 mom10 mom11 mom12
mom13 mom14 mom15 mom16 mom17 mom18 mom19 mom20 mom21 mom22 mom23 mom24
local n : word count `varlist'
matrix define A = J(`n',3,.)
mat rownames A = `varlist'
mat colnames A = Mean SD T
local row = 1
foreach var of varlist `varlist' {
   qui summarize `var'
   mat A[`row',1] = r(mean)/6*100
   mat A[`row',2] = r(sd)
   qui reg `var'
   mat A[`row',3] = _b[_cons] / _se[_cons]
   local ++row
}
mat list A, format(%9.2f)
```

//Covid-19 - Post-Recession Period

```
use "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\Only data.dta"


//Set up the Import Options and import the data//
//small letters on all variables
rename PERMNO permno
rename SHRCD shrcd
rename EXCHCD exchcd
rename SICCD siccd
rename PRC prc
rename RET ret
rename SHROUT shrout

//Create date variable mofd, and make data readable.
gen mofd = mofd(date)
sort date
keep if inrange(mofd, 718, 740)

format mofd %tm
xtset permno mofd, monthly
unique permno


//Data cleaning//
//Remove top 1% return and bottom 1% return
winsor2 ret, cuts(1 99) trim by(mofd)
replace ret=ret_tr
drop ret_tr

//keep exchange code if it is in NYSE, NASDAQ, NYSE American
keep if exchcd == 1 | exchcd == 2 | exchcd == 3

//keep share code if they are ordinary common securities
keep if shrcd == 10 | shrcd == 11

//Remove share price below 5
drop if prc < 5



//Create new variable for two-digit SIC codes
gen sic = substr(string(siccd),1,2)

//Convert sic to numeric variable
destring sic, replace
```

```
//recode sic to numeric industry
recode sic (10/14=1) (20=2) (22/23=3) (26=4) (28=5) (29=6) (32=7) (33=8) (34=9) (35=10) (36=11)
(37=12) (38/39=13) (40=14) (41/44=15) (46/47=15) (45=16) (49=17) (53=18) (50/52=19) (54/59=19)
(60/69=20) (73=21) (70/72=22) (74/79=22) (80/86=23) (87/99=24)


//Only keep industries we want to analyze
keep if sic == 1 | sic == 2 | sic == 3 | sic == 4 | sic == 5 | sic == 6 | sic == 7| sic == 8 | sic == 9 | sic ==
10 | sic == 11 | sic == 12 | sic == 13 | sic == 14 | sic == 15 | sic == 16 | sic == 17 | sic == 18 | sic == 19
| sic == 20 | sic == 21 | sic == 22 | sic == 23 | sic == 24
```

//Formation period: Generate past 6 months return, and skip 1 month – J=6

```
by permno: gen preret1 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==1
by permno: gen preret2 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==2
by permno: gen preret3 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==3
by permno: gen preret4 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==4
by permno: gen preret5 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==5
by permno: gen preret6 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==6
by permno: gen preret7 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==7
by permno: gen preret8 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==8
by permno: gen preret9 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==9
by permno: gen preret10 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==10
by permno: gen preret11 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==11
by permno: gen preret12 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==12
by permno: gen preret13 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==13
by permno: gen preret14 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==14
by permno: gen preret15 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==15
by permno: gen preret16 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==16
by permno: gen preret17 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==17
by permno: gen preret18 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==18
by permno: gen preret19 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==19
by permno: gen preret20 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==20
by permno: gen preret21 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==21
by permno: gen preret22 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==22
by permno: gen preret23 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==23
by permno: gen preret24 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==24

sort mofd
```

//Holding period: Make returns for 6 months, expressed returns in percentage points

```
sort permno mofd
by permno: gen fret1 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==1
by permno: gen fret2 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==2
by permno: gen fret3 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==3
by permno: gen fret4 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==4
by permno: gen fret5 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==5
by permno: gen fret6 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==6
by permno: gen fret7 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==7
by permno: gen fret8 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==8
by permno: gen fret9 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==9
by permno: gen fret10 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==10
by permno: gen fret11 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==11
```

```
by permno: gen fret12 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==12
by permno: gen fret13 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==13
by permno: gen fret14 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==14
by permno: gen fret15 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==15
by permno: gen fret16 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==16
by permno: gen fret17 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==17
by permno: gen fret18 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==18
by permno: gen fret19 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==19
by permno: gen fret20 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==20
by permno: gen fret21 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==21
by permno: gen fret22 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==22
by permno: gen fret23 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==23
by permno: gen fret24 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==24

//Rank stocks based on past 6-month return: the command creates the variable p_6 which ranks each
stock based on their past 6 months return every month, into 10 quantiles//
astile port1 = preret1, by(mofd) nq(10), if sic==1
astile port2 = preret2, by(mofd) nq(10), if sic==2
astile port3 = preret3, by(mofd) nq(10), if sic==3
astile port4 = preret4, by(mofd) nq(10), if sic==4
astile port5 = preret5, by(mofd) nq(10), if sic==5
astile port6 = preret6, by(mofd) nq(10), if sic==6
astile port7 = preret7, by(mofd) nq(10), if sic==7
astile port8 = preret8, by(mofd) nq(10), if sic==8
astile port9 = preret9, by(mofd) nq(10), if sic==9
astile port10 = preret10, by(mofd) nq(10), if sic==10
astile port11 = preret11, by(mofd) nq(10), if sic==11
astile port12 = preret12, by(mofd) nq(10), if sic==12
astile port13 = preret13, by(mofd) nq(10), if sic==13
astile port14 = preret14, by(mofd) nq(10), if sic==14
astile port15 = preret15, by(mofd) nq(10), if sic==15
astile port16 = preret16, by(mofd) nq(10), if sic==16
astile port17 = preret17, by(mofd) nq(10), if sic==17
astile port18 = preret18, by(mofd) nq(10), if sic==18
astile port19 = preret19, by(mofd) nq(10), if sic==19
astile port20 = preret20, by(mofd) nq(10), if sic==20
astile port21 = preret21, by(mofd) nq(10), if sic==21
astile port22 = preret22, by(mofd) nq(10), if sic==22
astile port23 = preret23, by(mofd) nq(10), if sic==23
astile port24 = preret24, by(mofd) nq(10), if sic==24


//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen p1_1=fret1 if port1==1
bysort mofd: gen p10_1=fret1 if port1==10
bysort mofd: gen p1_2=fret2 if port2==1
bysort mofd: gen p10_2=fret2 if port2==10
bysort mofd: gen p1_3=fret3 if port3==1
bysort mofd: gen p10_3=fret3 if port3==10
bysort mofd: gen p1_4=fret4 if port4==1
bysort mofd: gen p10_4=fret4 if port4==10
bysort mofd: gen p1_5=fret5 if port5==1
bysort mofd: gen p10_5=fret5 if port5==10
```

```
bysort mofd: gen p1_6=fret6 if port6==1
bysort mofd: gen p10_6=fret6 if port6==10
bysort mofd: gen p1_7=fret7 if port7==1
bysort mofd: gen p10_7=fret7 if port7==10
bysort mofd: gen p1_8=fret8 if port8==1
bysort mofd: gen p10_8=fret8 if port8==10
bysort mofd: gen p1_9=fret9 if port9==1
bysort mofd: gen p10_9=fret9 if port9==10
bysort mofd: gen p1_10=fret10 if port10==1
bysort mofd: gen p10_10=fret10 if port10==10
bysort mofd: gen p1_11=fret11 if port11==1
bysort mofd: gen p10_11=fret11 if port11==10
bysort mofd: gen p1_12=fret12 if port12==1
bysort mofd: gen p10_12=fret12 if port12==10
bysort mofd: gen p1_13=fret13 if port13==1
bysort mofd: gen p10_13=fret13 if port13==10
bysort mofd: gen p1_14=fret14 if port14==1
bysort mofd: gen p10_14=fret14 if port14==10
bysort mofd: gen p1_15=fret15 if port15==1
bysort mofd: gen p10_15=fret15 if port15==10
bysort mofd: gen p1_16=fret16 if port16==1
bysort mofd: gen p10_16=fret16 if port16==10
bysort mofd: gen p1_17=fret17 if port17==1
bysort mofd: gen p10_17=fret17 if port17==10
bysort mofd: gen p1_18=fret18 if port18==1
bysort mofd: gen p10_18=fret18 if port18==10
bysort mofd: gen p1_19=fret19 if port19==1
bysort mofd: gen p10_19=fret19 if port19==10
bysort mofd: gen p1_20=fret20 if port20==1
bysort mofd: gen p10_20=fret20 if port20==10
bysort mofd: gen p1_21=fret21 if port21==1
bysort mofd: gen p10_21=fret21 if port21==10
bysort mofd: gen p1_22=fret22 if port22==1
bysort mofd: gen p10_22=fret22 if port22==10
bysort mofd: gen p1_23=fret23 if port23==1
bysort mofd: gen p10_23=fret23 if port23==10
bysort mofd: gen p1_24=fret24 if port24==1
bysort mofd: gen p10_24=fret24 if port24==10


//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
bysort mofd: egen meanp1_1=mean(p1_1)
bysort mofd: egen meanp10_1=mean(p10_1)
bysort mofd: egen meanp1_2=mean(p1_2)
bysort mofd: egen meanp10_2=mean(p10_2)
bysort mofd: egen meanp1_3=mean(p1_3)
bysort mofd: egen meanp10_3=mean(p10_3)
bysort mofd: egen meanp1_4=mean(p1_4)
bysort mofd: egen meanp10_4=mean(p10_4)
bysort mofd: egen meanp1_5=mean(p1_5)
bysort mofd: egen meanp10_5=mean(p10_5)
bysort mofd: egen meanp1_6=mean(p1_6)
bysort mofd: egen meanp10_6=mean(p10_6)
```

bysort mofd: egen meanp1_7=mean(p1_7)
bysort mofd: egen meanp10_7=mean(p10_7)
bysort mofd: egen meanp1_8=mean(p1_8)
bysort mofd: egen meanp10_8=mean(p10_8)
bysort mofd: egen meanp1_9=mean(p1_9)
bysort mofd: egen meanp10_9=mean(p10_9)
bysort mofd: egen meanp1_10=mean(p1_10)
bysort mofd: egen meanp10_10=mean(p10_10)
bysort mofd: egen meanp1_11=mean(p1_11)
bysort mofd: egen meanp10_11=mean(p10_11)
bysort mofd: egen meanp1_12=mean(p1_12)
bysort mofd: egen meanp10_12=mean(p10_12)
bysort mofd: egen meanp1_13=mean(p1_13)
bysort mofd: egen meanp10_13=mean(p10_13)
bysort mofd: egen meanp1_14=mean(p1_14)
bysort mofd: egen meanp10_14=mean(p10_14)
bysort mofd: egen meanp1_15=mean(p1_15)
bysort mofd: egen meanp10_15=mean(p10_15)
bysort mofd: egen meanp1_16=mean(p1_16)
bysort mofd: egen meanp10_16=mean(p10_16)
bysort mofd: egen meanp1_17=mean(p1_17)
bysort mofd: egen meanp10_17=mean(p10_17)
bysort mofd: egen meanp1_18=mean(p1_18)
bysort mofd: egen meanp10_18=mean(p10_18)
bysort mofd: egen meanp1_19=mean(p1_19)
bysort mofd: egen meanp10_19=mean(p10_19)
bysort mofd: egen meanp1_20=mean(p1_20)
bysort mofd: egen meanp10_20=mean(p10_20)
bysort mofd: egen meanp1_21=mean(p1_21)
bysort mofd: egen meanp10_21=mean(p10_21)
bysort mofd: egen meanp1_22=mean(p1_22)
bysort mofd: egen meanp10_22=mean(p10_22)
bysort mofd: egen meanp1_23=mean(p1_23)
bysort mofd: egen meanp10_23=mean(p10_23)
bysort mofd: egen meanp1_24=mean(p1_24)
bysort mofd: egen meanp10_24=mean(p10_24)

//Create momentum return, winners-losers
gen mom1=meanp10_1-meanp1_1
gen mom2=meanp10_2-meanp1_2
gen mom3=meanp10_3-meanp1_3
gen mom4=meanp10_4-meanp1_4
gen mom5=meanp10_5-meanp1_5
gen mom6=meanp10_6-meanp1_6
gen mom7=meanp10_7-meanp1_7
gen mom8=meanp10_8-meanp1_8
gen mom9=meanp10_9-meanp1_9
gen mom10=meanp10_10-meanp1_10
gen mom11=meanp10_11-meanp1_11
gen mom12=meanp10_12-meanp1_12
gen mom13=meanp10_13-meanp1_13

```
gen mom14=meanp10_14-meanp1_14
gen mom15=meanp10_15-meanp1_15
gen mom16=meanp10_16-meanp1_16
gen mom17=meanp10_17-meanp1_17
gen mom18=meanp10_18-meanp1_18
gen mom19=meanp10_19-meanp1_19
gen mom20=meanp10_20-meanp1_20
gen mom21=meanp10_21-meanp1_21
gen mom22=meanp10_22-meanp1_22
gen mom23=meanp10_23-meanp1_23
gen mom24=meanp10_24-meanp1_24


//drop duplicates, one average return for each month
duplicates drop mofd, force
```

//Table of descriptive statistics

```
//mean return is expressed as monthly return ,in percentage form
local varlist mom1 mom2 mom3 mom4 mom5 mom6 mom7 mom8 mom9 mom10 mom11 mom12
mom13 mom15 mom16 mom17 mom18 mom19 mom20 mom21 mom22 mom23 mom24
local n : word count `varlist'
matrix define A = J(`n',3,.)
mat rownames A = `varlist'
mat colnames A = Mean SD T
local row = 1
foreach var of varlist `varlist' {
   qui summarize `var'
   mat A[`row',1] = r(mean)/6*100
   mat A[`row',2] = r(sd)
   qui reg `var'
   mat A[`row',3] = _b[_cons] / _se[_cons]
   local ++row
}
mat list A, format(%9.2f)
```

//The Financial Crisis - Recession Period

use "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\Only data.dta"

//Set up the Import Options and import the data//
//small letters on all variables
rename PERMNO permno
rename SHRCD shrcd
rename EXCHCD exchcd
rename SICCD siccd
rename PRC prc
rename RET ret
rename SHROUT shrout

//Create date variable mofd, and make data readable.
gen mofd = mofd(date)
sort date
keep if inrange(mofd, 569, 598)
format mofd %tm
xtset permno mofd, monthly
unique permno

//Data cleaning//
//Remove top 1% return and bottom 1% return
winsor2 ret, cuts(1 99) trim by(mofd)
replace ret=ret_tr
drop ret_tr

//keep exchange code if it is in NYSE, NASDAQ, NYSE American
keep if exchcd == 1 | exchcd == 2 | exchcd == 3

//keep share code if they are ordinary common securities
keep if shrcd == 10 | shrcd == 11

//Remove share price below 5
drop if prc < 5

//Create new variable for two-digit SIC codes
gen sic = substr(string(siccd),1,2)

//Convert sic to numeric variable
destring sic, replace

```
//recode sic to numeric industry
recode sic (10/14=1) (20=2) (22/23=3) (26=4) (28=5) (29=6) (32=7) (33=8) (34=9) (35=10) (36=11)
(37=12) (38/39=13) (40=14) (41/44=15) (46/47=15) (45=16) (49=17) (53=18) (50/52=19) (54/59=19)
(60/69=20) (73=21) (70/72=22) (74/79=22) (80/86=23) (87/99=24)


//Only keep industries we want to analyze
keep if sic == 1 | sic == 2 | sic == 3 | sic == 4 | sic == 5 | sic == 6 | sic == 7| sic == 8 | sic == 9 | sic ==
10 | sic == 11 | sic == 12 | sic == 13 | sic == 14 | sic == 15 | sic == 16 | sic == 17 | sic == 18 | sic == 19
| sic == 20 | sic == 21 | sic == 22 | sic == 23 | sic == 24
```

//Formation period: Generate past 6 months return, and skip 1 month – J=6

```
by permno: gen preret1 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==1
by permno: gen preret2 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==2
by permno: gen preret3 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==3
by permno: gen preret4 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==4
by permno: gen preret5 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==5
by permno: gen preret6 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==6
by permno: gen preret7 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==7
by permno: gen preret8 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==8
by permno: gen preret9 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==9
by permno: gen preret10 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==10
by permno: gen preret11 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==11
by permno: gen preret12 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==12
by permno: gen preret13 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==13
by permno: gen preret14 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==14
by permno: gen preret15 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==15
by permno: gen preret16 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==16
by permno: gen preret17 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==17
by permno: gen preret18 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==18
by permno: gen preret19 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==19
by permno: gen preret20 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==20
by permno: gen preret21 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==21
by permno: gen preret22 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==22
by permno: gen preret23 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==23
by permno: gen preret24 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==24

sort mofd
```

//Holding period: Make returns for 6 months, expressed returns in percentage points

```
sort permno mofd
by permno: gen fret1 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==1
by permno: gen fret2 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==2
by permno: gen fret3 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==3
by permno: gen fret4 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==4
by permno: gen fret5 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==5
by permno: gen fret6 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==6
by permno: gen fret7 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==7
by permno: gen fret8 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==8
by permno: gen fret9 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==9
by permno: gen fret10 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==10
```

```
by permno: gen fret11 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==11
by permno: gen fret12 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==12
by permno: gen fret13 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==13
by permno: gen fret14 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==14
by permno: gen fret15 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==15
by permno: gen fret16 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==16
by permno: gen fret17 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==17
by permno: gen fret18 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==18
by permno: gen fret19 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==19
by permno: gen fret20 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==20
by permno: gen fret21 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==21
by permno: gen fret22 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==22
by permno: gen fret23 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==23
by permno: gen fret24 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==24

//Rank stocks based on past 6-month return: the command creates the variable p_6 which ranks each
stock based on their past 6 months return every month, into 10 quantiles//
astile port1 = preret1, by(mofd) nq(10), if sic==1
astile port2 = preret2, by(mofd) nq(10), if sic==2
astile port3 = preret3, by(mofd) nq(10), if sic==3
astile port4 = preret4, by(mofd) nq(10), if sic==4
astile port5 = preret5, by(mofd) nq(10), if sic==5
astile port6 = preret6, by(mofd) nq(10), if sic==6
astile port7 = preret7, by(mofd) nq(10), if sic==7
astile port8 = preret8, by(mofd) nq(10), if sic==8
astile port9 = preret9, by(mofd) nq(10), if sic==9
astile port10 = preret10, by(mofd) nq(10), if sic==10
astile port11 = preret11, by(mofd) nq(10), if sic==11
astile port12 = preret12, by(mofd) nq(10), if sic==12
astile port13 = preret13, by(mofd) nq(10), if sic==13
astile port14 = preret14, by(mofd) nq(10), if sic==14
astile port15 = preret15, by(mofd) nq(10), if sic==15
astile port16 = preret16, by(mofd) nq(10), if sic==16
astile port17 = preret17, by(mofd) nq(10), if sic==17
astile port18 = preret18, by(mofd) nq(10), if sic==18
astile port19 = preret19, by(mofd) nq(10), if sic==19
astile port20 = preret20, by(mofd) nq(10), if sic==20
astile port21 = preret21, by(mofd) nq(10), if sic==21
astile port22 = preret22, by(mofd) nq(10), if sic==22
astile port23 = preret23, by(mofd) nq(10), if sic==23
astile port24 = preret24, by(mofd) nq(10), if sic==24


//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen p1_1=fret1 if port1==1
bysort mofd: gen p10_1=fret1 if port1==10
bysort mofd: gen p1_2=fret2 if port2==1
bysort mofd: gen p10_2=fret2 if port2==10
bysort mofd: gen p1_3=fret3 if port3==1
bysort mofd: gen p10_3=fret3 if port3==10
bysort mofd: gen p1_4=fret4 if port4==1
bysort mofd: gen p10_4=fret4 if port4==10
bysort mofd: gen p1_5=fret5 if port5==1
```

```
bysort mofd: gen p10_5=fret5 if port5==10
bysort mofd: gen p1_6=fret6 if port6==1
bysort mofd: gen p10_6=fret6 if port6==10
bysort mofd: gen p1_7=fret7 if port7==1
bysort mofd: gen p10_7=fret7 if port7==10
bysort mofd: gen p1_8=fret8 if port8==1
bysort mofd: gen p10_8=fret8 if port8==10
bysort mofd: gen p1_9=fret9 if port9==1
bysort mofd: gen p10_9=fret9 if port9==10
bysort mofd: gen p1_10=fret10 if port10==1
bysort mofd: gen p10_10=fret10 if port10==10
bysort mofd: gen p1_11=fret11 if port11==1
bysort mofd: gen p10_11=fret11 if port11==10
bysort mofd: gen p1_12=fret12 if port12==1
bysort mofd: gen p10_12=fret12 if port12==10
bysort mofd: gen p1_13=fret13 if port13==1
bysort mofd: gen p10_13=fret13 if port13==10
bysort mofd: gen p1_14=fret14 if port14==1
bysort mofd: gen p10_14=fret14 if port14==10
bysort mofd: gen p1_15=fret15 if port15==1
bysort mofd: gen p10_15=fret15 if port15==10
bysort mofd: gen p1_16=fret16 if port16==1
bysort mofd: gen p10_16=fret16 if port16==10
bysort mofd: gen p1_17=fret17 if port17==1
bysort mofd: gen p10_17=fret17 if port17==10
bysort mofd: gen p1_18=fret18 if port18==1
bysort mofd: gen p10_18=fret18 if port18==10
bysort mofd: gen p1_19=fret19 if port19==1
bysort mofd: gen p10_19=fret19 if port19==10
bysort mofd: gen p1_20=fret20 if port20==1
bysort mofd: gen p10_20=fret20 if port20==10
bysort mofd: gen p1_21=fret21 if port21==1
bysort mofd: gen p10_21=fret21 if port21==10
bysort mofd: gen p1_22=fret22 if port22==1
bysort mofd: gen p10_22=fret22 if port22==10
bysort mofd: gen p1_23=fret23 if port23==1
bysort mofd: gen p10_23=fret23 if port23==10
bysort mofd: gen p1_24=fret24 if port24==1
bysort mofd: gen p10_24=fret24 if port24==10


//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
bysort mofd: egen meanp1_1=mean(p1_1)
bysort mofd: egen meanp10_1=mean(p10_1)
bysort mofd: egen meanp1_2=mean(p1_2)
bysort mofd: egen meanp10_2=mean(p10_2)
bysort mofd: egen meanp1_3=mean(p1_3)
bysort mofd: egen meanp10_3=mean(p10_3)
bysort mofd: egen meanp1_4=mean(p1_4)
bysort mofd: egen meanp10_4=mean(p10_4)
bysort mofd: egen meanp1_5=mean(p1_5)
bysort mofd: egen meanp10_5=mean(p10_5)
bysort mofd: egen meanp1_6=mean(p1_6)
```

```
bysort mofd: egen meanp10_6=mean(p10_6)
bysort mofd: egen meanp1_7=mean(p1_7)
bysort mofd: egen meanp10_7=mean(p10_7)
bysort mofd: egen meanp1_8=mean(p1_8)
bysort mofd: egen meanp10_8=mean(p10_8)
bysort mofd: egen meanp1_9=mean(p1_9)
bysort mofd: egen meanp10_9=mean(p10_9)
bysort mofd: egen meanp1_10=mean(p1_10)
bysort mofd: egen meanp10_10=mean(p10_10)
bysort mofd: egen meanp1_11=mean(p1_11)
bysort mofd: egen meanp10_11=mean(p10_11)
bysort mofd: egen meanp1_12=mean(p1_12)
bysort mofd: egen meanp10_12=mean(p10_12)
bysort mofd: egen meanp1_13=mean(p1_13)
bysort mofd: egen meanp10_13=mean(p10_13)
bysort mofd: egen meanp1_14=mean(p1_14)
bysort mofd: egen meanp10_14=mean(p10_14)
bysort mofd: egen meanp1_15=mean(p1_15)
bysort mofd: egen meanp10_15=mean(p10_15)
bysort mofd: egen meanp1_16=mean(p1_16)
bysort mofd: egen meanp10_16=mean(p10_16)
bysort mofd: egen meanp1_17=mean(p1_17)
bysort mofd: egen meanp10_17=mean(p10_17)
bysort mofd: egen meanp1_18=mean(p1_18)
bysort mofd: egen meanp10_18=mean(p10_18)
bysort mofd: egen meanp1_19=mean(p1_19)
bysort mofd: egen meanp10_19=mean(p10_19)
bysort mofd: egen meanp1_20=mean(p1_20)
bysort mofd: egen meanp10_20=mean(p10_20)
bysort mofd: egen meanp1_21=mean(p1_21)
bysort mofd: egen meanp10_21=mean(p10_21)
bysort mofd: egen meanp1_22=mean(p1_22)
bysort mofd: egen meanp10_22=mean(p10_22)
bysort mofd: egen meanp1_23=mean(p1_23)
bysort mofd: egen meanp10_23=mean(p10_23)
bysort mofd: egen meanp1_24=mean(p1_24)
bysort mofd: egen meanp10_24=mean(p10_24)


// Create momentum return, winners-losers and check for significance
gen mom1=meanp10_1-meanp1_1
gen mom2=meanp10_2-meanp1_2
gen mom3=meanp10_3-meanp1_3
gen mom4=meanp10_4-meanp1_4
gen mom5=meanp10_5-meanp1_5
gen mom6=meanp10_6-meanp1_6
gen mom7=meanp10_7-meanp1_7
gen mom8=meanp10_8-meanp1_8
gen mom9=meanp10_9-meanp1_9
gen mom10=meanp10_10-meanp1_10
gen mom11=meanp10_11-meanp1_11
gen mom12=meanp10_12-meanp1_12
```

```
gen mom13=meanp10_13-meanp1_13
gen mom14=meanp10_14-meanp1_14
gen mom15=meanp10_15-meanp1_15
gen mom16=meanp10_16-meanp1_16
gen mom17=meanp10_17-meanp1_17
gen mom18=meanp10_18-meanp1_18
gen mom19=meanp10_19-meanp1_19
gen mom20=meanp10_20-meanp1_20
gen mom21=meanp10_21-meanp1_21
gen mom22=meanp10_22-meanp1_22
gen mom23=meanp10_23-meanp1_23
gen mom24=meanp10_24-meanp1_24


//drop duplicates, one average return for each month
duplicates drop mofd, force
```

// Table of descriptive statistics

```
//mean return is expressed as monthly return ,in percentage form
local varlist mom1 mom2 mom3 mom4 mom5 mom6 mom7 mom8 mom9 mom10 mom11 mom12
mom13 mom15 mom16 mom17 mom18 mom19 mom20 mom21 mom22 mom23 mom24
local n : word count `varlist'
matrix define A = J(`n',3,.)
mat rownames A = `varlist'
mat colnames A = Mean SD T
local row = 1
foreach var of varlist `varlist' {
   qui summarize `var'
   mat A[`row',1] = r(mean)/6*100
   mat A[`row',2] = r(sd)
   qui reg `var'
   mat A[`row',3] = _b[_cons] / _se[_cons]
   local ++row
}
mat list A, format(%9.2f)
```

//The Financial Crisis – Post-Recession Period

use "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\Only data.dta"

//Set up the Import Options and import the data//
//small letters on all variables
rename PERMNO permno
rename SHRCD shrcd
rename EXCHCD exchcd
rename SICCD siccd
rename PRC prc
rename RET ret
rename SHROUT shrout

//Create date variable mofd, and make data readable.
gen mofd = mofd(date)
sort date
keep if inrange(mofd, 588, 610)
format mofd %tm
xtset permno mofd, monthly
unique permno

//Data cleaning//
//Remove top 1% return and bottom 1% return
winsor2 ret, cuts(1 99) trim by(mofd)
replace ret=ret_tr
drop ret_tr

//keep exchange code if it is in NYSE, NASDAQ, NYSE American
keep if exchcd == 1 | exchcd == 2 | exchcd == 3

//keep share code if they are ordinary common securities
keep if shrcd == 10 | shrcd == 11

//Remove share price below 5
drop if prc < 5

//Create new variable for two-digit SIC codes
gen sic = substr(string(siccd),1,2)

//Convert sic to numeric variable
destring sic, replace

```
//recode sic to numeric industry
recode sic (10/14=1) (20=2) (22/23=3) (26=4) (28=5) (29=6) (32=7) (33=8) (34=9) (35=10) (36=11)
(37=12) (38/39=13) (40=14) (41/44=15) (46/47=15) (45=16) (49=17) (53=18) (50/52=19) (54/59=19)
(60/69=20) (73=21) (70/72=22) (74/79=22) (80/86=23) (87/99=24)


//Only keep industries we want to analyze
keep if sic == 1 | sic == 2 | sic == 3 | sic == 4 | sic == 5 | sic == 6 | sic == 7| sic == 8 | sic == 9 | sic ==
10 | sic == 11 | sic == 12 | sic == 13 | sic == 14 | sic == 15 | sic == 16 | sic == 17 | sic == 18 | sic == 19
| sic == 20 | sic == 21 | sic == 22 | sic == 23 | sic == 24
```

//Formation period: Generate past 6 months return, and skip 1 month – J=6

```
by permno: gen preret1 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==1
by permno: gen preret2 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==2
by permno: gen preret3 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==3
by permno: gen preret4 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==4
by permno: gen preret5 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==5
by permno: gen preret6 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==6
by permno: gen preret7 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==7
by permno: gen preret8 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==8
by permno: gen preret9 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==9
by permno: gen preret10 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==10
by permno: gen preret11 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==11
by permno: gen preret12 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==12
by permno: gen preret13 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==13
by permno: gen preret14 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==14
by permno: gen preret15 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==15
by permno: gen preret16 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==16
by permno: gen preret17 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==17
by permno: gen preret18 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==18
by permno: gen preret19 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==19
by permno: gen preret20 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==20
by permno: gen preret21 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==21
by permno: gen preret22 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==22
by permno: gen preret23 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==23
by permno: gen preret24 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==24

sort mofd
```

//Holding period: Make returns for 6 months, expressed returns in percentage points

```
sort permno mofd
by permno: gen fret1 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==1
by permno: gen fret2 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==2
by permno: gen fret3 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==3
by permno: gen fret4 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==4
by permno: gen fret5 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==5
by permno: gen fret6 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==6
by permno: gen fret7 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==7
by permno: gen fret8 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==8
by permno: gen fret9 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==9
by permno: gen fret10 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==10
by permno: gen fret11 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==11
```

```
by permno: gen fret12 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==12
by permno: gen fret13 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==13
by permno: gen fret14 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==14
by permno: gen fret15 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==15
by permno: gen fret16 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==16
by permno: gen fret17 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==17
by permno: gen fret18 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==18
by permno: gen fret19 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==19
by permno: gen fret20 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==20
by permno: gen fret21 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==21
by permno: gen fret22 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==22
by permno: gen fret23 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==23
by permno: gen fret24 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==24

//Rank stocks based on past 6-month return: the command creates the variable p_6 which ranks each
stock based on their past 6 months return every month, into 10 quantiles//
astile port1 = preret1, by(mofd) nq(10), if sic==1
astile port2 = preret2, by(mofd) nq(10), if sic==2
astile port3 = preret3, by(mofd) nq(10), if sic==3
astile port4 = preret4, by(mofd) nq(10), if sic==4
astile port5 = preret5, by(mofd) nq(10), if sic==5
astile port6 = preret6, by(mofd) nq(10), if sic==6
astile port7 = preret7, by(mofd) nq(10), if sic==7
astile port8 = preret8, by(mofd) nq(10), if sic==8
astile port9 = preret9, by(mofd) nq(10), if sic==9
astile port10 = preret10, by(mofd) nq(10), if sic==10
astile port11 = preret11, by(mofd) nq(10), if sic==11
astile port12 = preret12, by(mofd) nq(10), if sic==12
astile port13 = preret13, by(mofd) nq(10), if sic==13
astile port14 = preret14, by(mofd) nq(10), if sic==14
astile port15 = preret15, by(mofd) nq(10), if sic==15
astile port16 = preret16, by(mofd) nq(10), if sic==16
astile port17 = preret17, by(mofd) nq(10), if sic==17
astile port18 = preret18, by(mofd) nq(10), if sic==18
astile port19 = preret19, by(mofd) nq(10), if sic==19
astile port20 = preret20, by(mofd) nq(10), if sic==20
astile port21 = preret21, by(mofd) nq(10), if sic==21
astile port22 = preret22, by(mofd) nq(10), if sic==22
astile port23 = preret23, by(mofd) nq(10), if sic==23
astile port24 = preret24, by(mofd) nq(10), if sic==24


//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen p1_1=fret1 if port1==1
bysort mofd: gen p10_1=fret1 if port1==10
bysort mofd: gen p1_2=fret2 if port2==1
bysort mofd: gen p10_2=fret2 if port2==10
bysort mofd: gen p1_3=fret3 if port3==1
bysort mofd: gen p10_3=fret3 if port3==10
bysort mofd: gen p1_4=fret4 if port4==1
bysort mofd: gen p10_4=fret4 if port4==10
bysort mofd: gen p1_5=fret5 if port5==1
bysort mofd: gen p10_5=fret5 if port5==10
```

```
bysort mofd: gen p1_6=fret6 if port6==1
bysort mofd: gen p10_6=fret6 if port6==10
bysort mofd: gen p1_7=fret7 if port7==1
bysort mofd: gen p10_7=fret7 if port7==10
bysort mofd: gen p1_8=fret8 if port8==1
bysort mofd: gen p10_8=fret8 if port8==10
bysort mofd: gen p1_9=fret9 if port9==1
bysort mofd: gen p10_9=fret9 if port9==10
bysort mofd: gen p1_10=fret10 if port10==1
bysort mofd: gen p10_10=fret10 if port10==10
bysort mofd: gen p1_11=fret11 if port11==1
bysort mofd: gen p10_11=fret11 if port11==10
bysort mofd: gen p1_12=fret12 if port12==1
bysort mofd: gen p10_12=fret12 if port12==10
bysort mofd: gen p1_13=fret13 if port13==1
bysort mofd: gen p10_13=fret13 if port13==10
bysort mofd: gen p1_14=fret14 if port14==1
bysort mofd: gen p10_14=fret14 if port14==10
bysort mofd: gen p1_15=fret15 if port15==1
bysort mofd: gen p10_15=fret15 if port15==10
bysort mofd: gen p1_16=fret16 if port16==1
bysort mofd: gen p10_16=fret16 if port16==10
bysort mofd: gen p1_17=fret17 if port17==1
bysort mofd: gen p10_17=fret17 if port17==10
bysort mofd: gen p1_18=fret18 if port18==1
bysort mofd: gen p10_18=fret18 if port18==10
bysort mofd: gen p1_19=fret19 if port19==1
bysort mofd: gen p10_19=fret19 if port19==10
bysort mofd: gen p1_20=fret20 if port20==1
bysort mofd: gen p10_20=fret20 if port20==10
bysort mofd: gen p1_21=fret21 if port21==1
bysort mofd: gen p10_21=fret21 if port21==10
bysort mofd: gen p1_22=fret22 if port22==1
bysort mofd: gen p10_22=fret22 if port22==10
bysort mofd: gen p1_23=fret23 if port23==1
bysort mofd: gen p10_23=fret23 if port23==10
bysort mofd: gen p1_24=fret24 if port24==1
bysort mofd: gen p10_24=fret24 if port24==10


//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
bysort mofd: egen meanp1_1=mean(p1_1)
bysort mofd: egen meanp10_1=mean(p10_1)
bysort mofd: egen meanp1_2=mean(p1_2)
bysort mofd: egen meanp10_2=mean(p10_2)
bysort mofd: egen meanp1_3=mean(p1_3)
bysort mofd: egen meanp10_3=mean(p10_3)
bysort mofd: egen meanp1_4=mean(p1_4)
bysort mofd: egen meanp10_4=mean(p10_4)
bysort mofd: egen meanp1_5=mean(p1_5)
bysort mofd: egen meanp10_5=mean(p10_5)
bysort mofd: egen meanp1_6=mean(p1_6)
bysort mofd: egen meanp10_6=mean(p10_6)
```

```
bysort mofd: egen meanp1_7=mean(p1_7)
bysort mofd: egen meanp10_7=mean(p10_7)
bysort mofd: egen meanp1_8=mean(p1_8)
bysort mofd: egen meanp10_8=mean(p10_8)
bysort mofd: egen meanp1_9=mean(p1_9)
bysort mofd: egen meanp10_9=mean(p10_9)
bysort mofd: egen meanp1_10=mean(p1_10)
bysort mofd: egen meanp10_10=mean(p10_10)
bysort mofd: egen meanp1_11=mean(p1_11)
bysort mofd: egen meanp10_11=mean(p10_11)
bysort mofd: egen meanp1_12=mean(p1_12)
bysort mofd: egen meanp10_12=mean(p10_12)
bysort mofd: egen meanp1_13=mean(p1_13)
bysort mofd: egen meanp10_13=mean(p10_13)
bysort mofd: egen meanp1_14=mean(p1_14)
bysort mofd: egen meanp10_14=mean(p10_14)
bysort mofd: egen meanp1_15=mean(p1_15)
bysort mofd: egen meanp10_15=mean(p10_15)
bysort mofd: egen meanp1_16=mean(p1_16)
bysort mofd: egen meanp10_16=mean(p10_16)
bysort mofd: egen meanp1_17=mean(p1_17)
bysort mofd: egen meanp10_17=mean(p10_17)
bysort mofd: egen meanp1_18=mean(p1_18)
bysort mofd: egen meanp10_18=mean(p10_18)
bysort mofd: egen meanp1_19=mean(p1_19)
bysort mofd: egen meanp10_19=mean(p10_19)
bysort mofd: egen meanp1_20=mean(p1_20)
bysort mofd: egen meanp10_20=mean(p10_20)
bysort mofd: egen meanp1_21=mean(p1_21)
bysort mofd: egen meanp10_21=mean(p10_21)
bysort mofd: egen meanp1_22=mean(p1_22)
bysort mofd: egen meanp10_22=mean(p10_22)
bysort mofd: egen meanp1_23=mean(p1_23)
bysort mofd: egen meanp10_23=mean(p10_23)
bysort mofd: egen meanp1_24=mean(p1_24)
bysort mofd: egen meanp10_24=mean(p10_24)


// Create momentum return, winners-losers
gen mom1=meanp10_1-meanp1_1
gen mom2=meanp10_2-meanp1_2
gen mom3=meanp10_3-meanp1_3
gen mom4=meanp10_4-meanp1_4
gen mom5=meanp10_5-meanp1_5
gen mom6=meanp10_6-meanp1_6
gen mom7=meanp10_7-meanp1_7
gen mom8=meanp10_8-meanp1_8
gen mom9=meanp10_9-meanp1_9
gen mom10=meanp10_10-meanp1_10
gen mom11=meanp10_11-meanp1_11
gen mom12=meanp10_12-meanp1_12
gen mom13=meanp10_13-meanp1_13
gen mom14=meanp10_14-meanp1_14
```

```
gen mom15=meanp10_15-meanp1_15
gen mom16=meanp10_16-meanp1_16
gen mom17=meanp10_17-meanp1_17
gen mom18=meanp10_18-meanp1_18
gen mom19=meanp10_19-meanp1_19
gen mom20=meanp10_20-meanp1_20
gen mom21=meanp10_21-meanp1_21
gen mom22=meanp10_22-meanp1_22
gen mom23=meanp10_23-meanp1_23
gen mom24=meanp10_24-meanp1_24


//drop duplicates, one average return for each month
duplicates drop mofd, force
```

// Table of descriptive statistics

```
//mean return is expressed as monthly return ,in percentage form
local varlist mom1 mom2 mom3 mom4 mom5 mom6 mom8 mom9 mom10 mom11 mom12 mom13
mom15 mom16 mom17 mom18 mom19 mom20 mom21 mom22 mom23 mom24
local n : word count `varlist'
matrix define A = J(`n',3,.)
mat rownames A = `varlist'
mat colnames A = Mean SD T
local row = 1
foreach var of varlist `varlist' {
   qui summarize `var'
   mat A[`row',1] = r(mean)/6*100
   mat A[`row',2] = r(sd)
   qui reg `var'
   mat A[`row',3] = _b[_cons] / _se[_cons]
   local ++row
}
mat list A, format(%9.2f)
```

//The Dot-Com Bubble – Recession Period

use "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\Only data.dta"

//Set up the Import Options and import the data//
//small letters on all variables
rename PERMNO permno
rename SHRCD shrcd
rename EXCHCD exchcd
rename SICCD siccd
rename PRC prc
rename RET ret
rename SHROUT shrout

//Create date variable mofd, and make data readable.
gen mofd = mofd(date)
sort date
keep if inrange(mofd, 488, 507)
format mofd %tm
xtset permno mofd, monthly
unique permno

//Data cleaning//
//Remove top 1% return and bottom 1% return
winsor2 ret, cuts(1 99) trim by(mofd)
replace ret=ret_tr
drop ret_tr

//keep exchange code if it is in NYSE, NASDAQ, NYSE American
keep if exchcd == 1 | exchcd == 2 | exchcd == 3

//keep share code if they are ordinary common securities
keep if shrcd == 10 | shrcd == 11

//Remove share price below 5
drop if prc < 5

//Create new variable for two-digit SIC codes
gen sic = substr(string(siccd),1,2)

//Convert sic to numeric variable
destring sic, replace

```
//recode sic to numeric industry
recode sic (10/14=1) (20=2) (22/23=3) (26=4) (28=5) (29=6) (32=7) (33=8) (34=9) (35=10) (36=11)
(37=12) (38/39=13) (40=14) (41/44=15) (46/47=15) (45=16) (49=17) (53=18) (50/52=19) (54/59=19)
(60/69=20) (73=21) (70/72=22) (74/79=22) (80/86=23) (87/99=24)


//Only keep industries we want to analyze
keep if sic == 1 | sic == 2 | sic == 3 | sic == 4 | sic == 5 | sic == 6 | sic == 7| sic == 8 | sic == 9 | sic ==
10 | sic == 11 | sic == 12 | sic == 13 | sic == 14 | sic == 15 | sic == 16 | sic == 17 | sic == 18 | sic == 19
| sic == 20 | sic == 21 | sic == 22 | sic == 23 | sic == 24
```

//Formation period: Generate past 6 months return, and skip 1 month – J=6

```
by permno: gen preret1 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==1
by permno: gen preret2 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==2
by permno: gen preret3 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==3
by permno: gen preret4 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==4
by permno: gen preret5 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==5
by permno: gen preret6 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==6
by permno: gen preret7 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==7
by permno: gen preret8 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==8
by permno: gen preret9 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==9
by permno: gen preret10 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==10
by permno: gen preret11 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==11
by permno: gen preret12 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==12
by permno: gen preret13 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==13
by permno: gen preret14 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==14
by permno: gen preret15 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==15
by permno: gen preret16 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==16
by permno: gen preret17 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==17
by permno: gen preret18 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==18
by permno: gen preret19 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==19
by permno: gen preret20 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==20
by permno: gen preret21 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==21
by permno: gen preret22 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==22
by permno: gen preret23 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==23
by permno: gen preret24 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==24

sort mofd
```

//Holding period: Make returns for 6 months, expressed returns in percentage points

```
sort permno mofd
by permno: gen fret1 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==1
by permno: gen fret2 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==2
by permno: gen fret3 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==3
by permno: gen fret4 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==4
by permno: gen fret5 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==5
by permno: gen fret6 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==6
by permno: gen fret7 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==7
by permno: gen fret8 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==8
by permno: gen fret9 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==9
by permno: gen fret10 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==10
```

```
by permno: gen fret11 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==11
by permno: gen fret12 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==12
by permno: gen fret13 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==13
by permno: gen fret14 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==14
by permno: gen fret15 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==15
by permno: gen fret16 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==16
by permno: gen fret17 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==17
by permno: gen fret18 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==18
by permno: gen fret19 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==19
by permno: gen fret20 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==20
by permno: gen fret21 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==21
by permno: gen fret22 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==22
by permno: gen fret23 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==23
by permno: gen fret24 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==24

//Rank stocks based on past 6-month return: the command creates the variable p_6 which ranks each
stock based on their past 6 months return every month, into 10 quantiles//
astile port1 = preret1, by(mofd) nq(10), if sic==1
astile port2 = preret2, by(mofd) nq(10), if sic==2
astile port3 = preret3, by(mofd) nq(10), if sic==3
astile port4 = preret4, by(mofd) nq(10), if sic==4
astile port5 = preret5, by(mofd) nq(10), if sic==5
astile port6 = preret6, by(mofd) nq(10), if sic==6
astile port7 = preret7, by(mofd) nq(10), if sic==7
astile port8 = preret8, by(mofd) nq(10), if sic==8
astile port9 = preret9, by(mofd) nq(10), if sic==9
astile port10 = preret10, by(mofd) nq(10), if sic==10
astile port11 = preret11, by(mofd) nq(10), if sic==11
astile port12 = preret12, by(mofd) nq(10), if sic==12
astile port13 = preret13, by(mofd) nq(10), if sic==13
astile port14 = preret14, by(mofd) nq(10), if sic==14
astile port15 = preret15, by(mofd) nq(10), if sic==15
astile port16 = preret16, by(mofd) nq(10), if sic==16
astile port17 = preret17, by(mofd) nq(10), if sic==17
astile port18 = preret18, by(mofd) nq(10), if sic==18
astile port19 = preret19, by(mofd) nq(10), if sic==19
astile port20 = preret20, by(mofd) nq(10), if sic==20
astile port21 = preret21, by(mofd) nq(10), if sic==21
astile port22 = preret22, by(mofd) nq(10), if sic==22
astile port23 = preret23, by(mofd) nq(10), if sic==23
astile port24 = preret24, by(mofd) nq(10), if sic==24


//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen p1_1=fret1 if port1==1
bysort mofd: gen p10_1=fret1 if port1==10
bysort mofd: gen p1_2=fret2 if port2==1
bysort mofd: gen p10_2=fret2 if port2==10
bysort mofd: gen p1_3=fret3 if port3==1
bysort mofd: gen p10_3=fret3 if port3==10
bysort mofd: gen p1_4=fret4 if port4==1
bysort mofd: gen p10_4=fret4 if port4==10
bysort mofd: gen p1_5=fret5 if port5==1
```

```
bysort mofd: gen p10_5=fret5 if port5==10
bysort mofd: gen p1_6=fret6 if port6==1
bysort mofd: gen p10_6=fret6 if port6==10
bysort mofd: gen p1_7=fret7 if port7==1
bysort mofd: gen p10_7=fret7 if port7==10
bysort mofd: gen p1_8=fret8 if port8==1
bysort mofd: gen p10_8=fret8 if port8==10
bysort mofd: gen p1_9=fret9 if port9==1
bysort mofd: gen p10_9=fret9 if port9==10
bysort mofd: gen p1_10=fret10 if port10==1
bysort mofd: gen p10_10=fret10 if port10==10
bysort mofd: gen p1_11=fret11 if port11==1
bysort mofd: gen p10_11=fret11 if port11==10
bysort mofd: gen p1_12=fret12 if port12==1
bysort mofd: gen p10_12=fret12 if port12==10
bysort mofd: gen p1_13=fret13 if port13==1
bysort mofd: gen p10_13=fret13 if port13==10
bysort mofd: gen p1_14=fret14 if port14==1
bysort mofd: gen p10_14=fret14 if port14==10
bysort mofd: gen p1_15=fret15 if port15==1
bysort mofd: gen p10_15=fret15 if port15==10
bysort mofd: gen p1_16=fret16 if port16==1
bysort mofd: gen p10_16=fret16 if port16==10
bysort mofd: gen p1_17=fret17 if port17==1
bysort mofd: gen p10_17=fret17 if port17==10
bysort mofd: gen p1_18=fret18 if port18==1
bysort mofd: gen p10_18=fret18 if port18==10
bysort mofd: gen p1_19=fret19 if port19==1
bysort mofd: gen p10_19=fret19 if port19==10
bysort mofd: gen p1_20=fret20 if port20==1
bysort mofd: gen p10_20=fret20 if port20==10
bysort mofd: gen p1_21=fret21 if port21==1
bysort mofd: gen p10_21=fret21 if port21==10
bysort mofd: gen p1_22=fret22 if port22==1
bysort mofd: gen p10_22=fret22 if port22==10
bysort mofd: gen p1_23=fret23 if port23==1
bysort mofd: gen p10_23=fret23 if port23==10
bysort mofd: gen p1_24=fret24 if port24==1
bysort mofd: gen p10_24=fret24 if port24==10


//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
bysort mofd: egen meanp1_1=mean(p1_1)
bysort mofd: egen meanp10_1=mean(p10_1)
bysort mofd: egen meanp1_2=mean(p1_2)
bysort mofd: egen meanp10_2=mean(p10_2)
bysort mofd: egen meanp1_3=mean(p1_3)
bysort mofd: egen meanp10_3=mean(p10_3)
bysort mofd: egen meanp1_4=mean(p1_4)
bysort mofd: egen meanp10_4=mean(p10_4)
bysort mofd: egen meanp1_5=mean(p1_5)
bysort mofd: egen meanp10_5=mean(p10_5)
bysort mofd: egen meanp1_6=mean(p1_6)
```

bysort mofd: egen meanp10_6=mean(p10_6)
bysort mofd: egen meanp1_7=mean(p1_7)
bysort mofd: egen meanp10_7=mean(p10_7)
bysort mofd: egen meanp1_8=mean(p1_8)
bysort mofd: egen meanp10_8=mean(p10_8)
bysort mofd: egen meanp1_9=mean(p1_9)
bysort mofd: egen meanp10_9=mean(p10_9)
bysort mofd: egen meanp1_10=mean(p1_10)
bysort mofd: egen meanp10_10=mean(p10_10)
bysort mofd: egen meanp1_11=mean(p1_11)
bysort mofd: egen meanp10_11=mean(p10_11)
bysort mofd: egen meanp1_12=mean(p1_12)
bysort mofd: egen meanp10_12=mean(p10_12)
bysort mofd: egen meanp1_13=mean(p1_13)
bysort mofd: egen meanp10_13=mean(p10_13)
bysort mofd: egen meanp1_14=mean(p1_14)
bysort mofd: egen meanp10_14=mean(p10_14)
bysort mofd: egen meanp1_15=mean(p1_15)
bysort mofd: egen meanp10_15=mean(p10_15)
bysort mofd: egen meanp1_16=mean(p1_16)
bysort mofd: egen meanp10_16=mean(p10_16)
bysort mofd: egen meanp1_17=mean(p1_17)
bysort mofd: egen meanp10_17=mean(p10_17)
bysort mofd: egen meanp1_18=mean(p1_18)
bysort mofd: egen meanp10_18=mean(p10_18)
bysort mofd: egen meanp1_19=mean(p1_19)
bysort mofd: egen meanp10_19=mean(p10_19)
bysort mofd: egen meanp1_20=mean(p1_20)
bysort mofd: egen meanp10_20=mean(p10_20)
bysort mofd: egen meanp1_21=mean(p1_21)
bysort mofd: egen meanp10_21=mean(p10_21)
bysort mofd: egen meanp1_22=mean(p1_22)
bysort mofd: egen meanp10_22=mean(p10_22)
bysort mofd: egen meanp1_23=mean(p1_23)
bysort mofd: egen meanp10_23=mean(p10_23)
bysort mofd: egen meanp1_24=mean(p1_24)
bysort mofd: egen meanp10_24=mean(p10_24)

// Create momentum return
gen mom1=meanp10_1-meanp1_1
gen mom2=meanp10_2-meanp1_2
gen mom3=meanp10_3-meanp1_3
gen mom4=meanp10_4-meanp1_4
gen mom5=meanp10_5-meanp1_5
gen mom6=meanp10_6-meanp1_6
gen mom7=meanp10_7-meanp1_7
gen mom8=meanp10_8-meanp1_8
gen mom9=meanp10_9-meanp1_9
gen mom10=meanp10_10-meanp1_10
gen mom11=meanp10_11-meanp1_11
gen mom12=meanp10_12-meanp1_12
gen mom13=meanp10_13-meanp1_13

```
gen mom14=meanp10_14-meanp1_14
gen mom15=meanp10_15-meanp1_15
gen mom16=meanp10_16-meanp1_16
gen mom17=meanp10_17-meanp1_17
gen mom18=meanp10_18-meanp1_18
gen mom19=meanp10_19-meanp1_19
gen mom20=meanp10_20-meanp1_20
gen mom21=meanp10_21-meanp1_21
gen mom22=meanp10_22-meanp1_22
gen mom23=meanp10_23-meanp1_23
gen mom24=meanp10_24-meanp1_24


//drop duplicates, one average return for each month
duplicates drop mofd, force


// Table of descriptive statistics
//mean return is expressed as monthly return ,in percentage form
local varlist mom1 mom2 mom3 mom4 mom5 mom6 mom7 mom8 mom9 mom10 mom11 mom12
mom13 mom14 mom15 mom16 mom17 mom18 mom19 mom20 mom21 mom22 mom23 mom24
local n : word count `varlist'
matrix define A = J(`n',3,.)
mat rownames A = `varlist'
mat colnames A = Mean SD T
local row = 1
foreach var of varlist `varlist' {
   qui summarize `var'
   mat A[`row',1] = r(mean)/6*100
   mat A[`row',2] = r(sd)
   qui reg `var'
   mat A[`row',3] = _b[_cons] / _se[_cons]
   local ++row
}
mat list A, format(%9.2f)
```

//The Dot-Com Bubble – Post-Recession Period

use "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\Only data.dta"

//Set up the Import Options and import the data//
//small letters on all variables
rename PERMNO permno
rename SHRCD shrcd
rename EXCHCD exchcd
rename SICCD siccd
rename PRC prc
rename RET ret
rename SHROUT shrout

//Create date variable mofd, and make data readable.
gen mofd = mofd(date)
sort date
keep if inrange(mofd, 497, 519)
format mofd %tm
xtset permno mofd, monthly
unique permno

//Data cleaning//
//Remove top 1% return and bottom 1% return
winsor2 ret, cuts(1 99) trim by(mofd)
replace ret=ret_tr
drop ret_tr

//keep exchange code if it is in NYSE, NASDAQ, NYSE American
keep if exchcd == 1 | exchcd == 2 | exchcd == 3

//keep share code if they are ordinary common securities
keep if shrcd == 10 | shrcd == 11

//Remove share price below 5
drop if prc < 5

//Create new variable for two-digit SIC codes
gen sic = substr(string(siccd),1,2)

//Convert sic to numeric variable
destring sic, replace

//recode sic to numeric industry
recode sic (10/14=1) (20=2) (22/23=3) (26=4) (28=5) (29=6) (32=7) (33=8) (34=9) (35=10) (36=11) (37=12) (38/39=13) (40=14) (41/44=15) (46/47=15) (45=16) (49=17) (53=18) (50/52=19) (54/59=19) (60/69=20) (73=21) (70/72=22) (74/79=22) (80/86=23) (87/99=24)

//Only keep industries we want to analyze
keep if sic == 1 | sic == 2 | sic == 3 | sic == 4 | sic == 5 | sic == 6 | sic == 7| sic == 8 | sic == 9 | sic == 10 | sic == 11 | sic == 12 | sic == 13 | sic == 14 | sic == 15 | sic == 16 | sic == 17 | sic == 18 | sic == 19 | sic == 20 | sic == 21 | sic == 22 | sic == 23 | sic == 24

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret1 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==1
by permno: gen preret2 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==2
by permno: gen preret3 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==3
by permno: gen preret4 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==4
by permno: gen preret5 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==5
by permno: gen preret6 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==6
by permno: gen preret7 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==7
by permno: gen preret8 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==8
by permno: gen preret9 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==9
by permno: gen preret10 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==10
by permno: gen preret11 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==11
by permno: gen preret12 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==12
by permno: gen preret13 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==13
by permno: gen preret14 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==14
by permno: gen preret15 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==15
by permno: gen preret16 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==16
by permno: gen preret17 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==17
by permno: gen preret18 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==18
by permno: gen preret19 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==19
by permno: gen preret20 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==20
by permno: gen preret21 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==21
by permno: gen preret22 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==22
by permno: gen preret23 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==23
by permno: gen preret24 = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret) if sic==24

sort mofd

//Holding period: Make returns for 6 months, expressed returns in percentage points
sort permno mofd
by permno: gen fret1 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==1
by permno: gen fret2 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==2
by permno: gen fret3 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==3
by permno: gen fret4 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==4
by permno: gen fret5 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==5
by permno: gen fret6 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==6
by permno: gen fret7 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==7
by permno: gen fret8 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==8
by permno: gen fret9 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==9
by permno: gen fret10 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==10
by permno: gen fret11 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==11

```
by permno: gen fret12 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==12
by permno: gen fret13 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==13
by permno: gen fret14 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==14
by permno: gen fret15 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==15
by permno: gen fret16 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==16
by permno: gen fret17 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==17
by permno: gen fret18 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==18
by permno: gen fret19 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==19
by permno: gen fret20 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==20
by permno: gen fret21 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==21
by permno: gen fret22 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==22
by permno: gen fret23 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==23
by permno: gen fret24 = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret) if sic==24
```

```
//Rank stocks based on past 6-month return: the command creates the variable p_6 which ranks each
stock based on their past 6 months return every month, into 10 quantiles//
astile port1 = preret1, by(mofd) nq(10), if sic==1
astile port2 = preret2, by(mofd) nq(10), if sic==2
astile port3 = preret3, by(mofd) nq(10), if sic==3
astile port4 = preret4, by(mofd) nq(10), if sic==4
astile port5 = preret5, by(mofd) nq(10), if sic==5
astile port6 = preret6, by(mofd) nq(10), if sic==6
astile port7 = preret7, by(mofd) nq(10), if sic==7
astile port8 = preret8, by(mofd) nq(10), if sic==8
astile port9 = preret9, by(mofd) nq(10), if sic==9
astile port10 = preret10, by(mofd) nq(10), if sic==10
astile port11 = preret11, by(mofd) nq(10), if sic==11
astile port12 = preret12, by(mofd) nq(10), if sic==12
astile port13 = preret13, by(mofd) nq(10), if sic==13
astile port14 = preret14, by(mofd) nq(10), if sic==14
astile port15 = preret15, by(mofd) nq(10), if sic==15
astile port16 = preret16, by(mofd) nq(10), if sic==16
astile port17 = preret17, by(mofd) nq(10), if sic==17
astile port18 = preret18, by(mofd) nq(10), if sic==18
astile port19 = preret19, by(mofd) nq(10), if sic==19
astile port20 = preret20, by(mofd) nq(10), if sic==20
astile port21 = preret21, by(mofd) nq(10), if sic==21
astile port22 = preret22, by(mofd) nq(10), if sic==22
astile port23 = preret23, by(mofd) nq(10), if sic==23
astile port24 = preret24, by(mofd) nq(10), if sic==24
```

```
//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen p1_1=fret1 if port1==1
bysort mofd: gen p10_1=fret1 if port1==10
bysort mofd: gen p1_2=fret2 if port2==1
bysort mofd: gen p10_2=fret2 if port2==10
bysort mofd: gen p1_3=fret3 if port3==1
bysort mofd: gen p10_3=fret3 if port3==10
bysort mofd: gen p1_4=fret4 if port4==1
bysort mofd: gen p10_4=fret4 if port4==10
bysort mofd: gen p1_5=fret5 if port5==1
```

```
bysort mofd: gen p10_5=fret5 if port5==10
bysort mofd: gen p1_6=fret6 if port6==1
bysort mofd: gen p10_6=fret6 if port6==10
bysort mofd: gen p1_7=fret7 if port7==1
bysort mofd: gen p10_7=fret7 if port7==10
bysort mofd: gen p1_8=fret8 if port8==1
bysort mofd: gen p10_8=fret8 if port8==10
bysort mofd: gen p1_9=fret9 if port9==1
bysort mofd: gen p10_9=fret9 if port9==10
bysort mofd: gen p1_10=fret10 if port10==1
bysort mofd: gen p10_10=fret10 if port10==10
bysort mofd: gen p1_11=fret11 if port11==1
bysort mofd: gen p10_11=fret11 if port11==10
bysort mofd: gen p1_12=fret12 if port12==1
bysort mofd: gen p10_12=fret12 if port12==10
bysort mofd: gen p1_13=fret13 if port13==1
bysort mofd: gen p10_13=fret13 if port13==10
bysort mofd: gen p1_14=fret14 if port14==1
bysort mofd: gen p10_14=fret14 if port14==10
bysort mofd: gen p1_15=fret15 if port15==1
bysort mofd: gen p10_15=fret15 if port15==10
bysort mofd: gen p1_16=fret16 if port16==1
bysort mofd: gen p10_16=fret16 if port16==10
bysort mofd: gen p1_17=fret17 if port17==1
bysort mofd: gen p10_17=fret17 if port17==10
bysort mofd: gen p1_18=fret18 if port18==1
bysort mofd: gen p10_18=fret18 if port18==10
bysort mofd: gen p1_19=fret19 if port19==1
bysort mofd: gen p10_19=fret19 if port19==10
bysort mofd: gen p1_20=fret20 if port20==1
bysort mofd: gen p10_20=fret20 if port20==10
bysort mofd: gen p1_21=fret21 if port21==1
bysort mofd: gen p10_21=fret21 if port21==10
bysort mofd: gen p1_22=fret22 if port22==1
bysort mofd: gen p10_22=fret22 if port22==10
bysort mofd: gen p1_23=fret23 if port23==1
bysort mofd: gen p10_23=fret23 if port23==10
bysort mofd: gen p1_24=fret24 if port24==1
bysort mofd: gen p10_24=fret24 if port24==10


//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
bysort mofd: egen meanp1_1=mean(p1_1)
bysort mofd: egen meanp10_1=mean(p10_1)
bysort mofd: egen meanp1_2=mean(p1_2)
bysort mofd: egen meanp10_2=mean(p10_2)
bysort mofd: egen meanp1_3=mean(p1_3)
bysort mofd: egen meanp10_3=mean(p10_3)
bysort mofd: egen meanp1_4=mean(p1_4)
bysort mofd: egen meanp10_4=mean(p10_4)
bysort mofd: egen meanp1_5=mean(p1_5)
bysort mofd: egen meanp10_5=mean(p10_5)
bysort mofd: egen meanp1_6=mean(p1_6)
```

```
bysort mofd: egen meanp10_6=mean(p10_6)
bysort mofd: egen meanp1_7=mean(p1_7)
bysort mofd: egen meanp10_7=mean(p10_7)
bysort mofd: egen meanp1_8=mean(p1_8)
bysort mofd: egen meanp10_8=mean(p10_8)
bysort mofd: egen meanp1_9=mean(p1_9)
bysort mofd: egen meanp10_9=mean(p10_9)
bysort mofd: egen meanp1_10=mean(p1_10)
bysort mofd: egen meanp10_10=mean(p10_10)
bysort mofd: egen meanp1_11=mean(p1_11)
bysort mofd: egen meanp10_11=mean(p10_11)
bysort mofd: egen meanp1_12=mean(p1_12)
bysort mofd: egen meanp10_12=mean(p10_12)
bysort mofd: egen meanp1_13=mean(p1_13)
bysort mofd: egen meanp10_13=mean(p10_13)
bysort mofd: egen meanp1_14=mean(p1_14)
bysort mofd: egen meanp10_14=mean(p10_14)
bysort mofd: egen meanp1_15=mean(p1_15)
bysort mofd: egen meanp10_15=mean(p10_15)
bysort mofd: egen meanp1_16=mean(p1_16)
bysort mofd: egen meanp10_16=mean(p10_16)
bysort mofd: egen meanp1_17=mean(p1_17)
bysort mofd: egen meanp10_17=mean(p10_17)
bysort mofd: egen meanp1_18=mean(p1_18)
bysort mofd: egen meanp10_18=mean(p10_18)
bysort mofd: egen meanp1_19=mean(p1_19)
bysort mofd: egen meanp10_19=mean(p10_19)
bysort mofd: egen meanp1_20=mean(p1_20)
bysort mofd: egen meanp10_20=mean(p10_20)
bysort mofd: egen meanp1_21=mean(p1_21)
bysort mofd: egen meanp10_21=mean(p10_21)
bysort mofd: egen meanp1_22=mean(p1_22)
bysort mofd: egen meanp10_22=mean(p10_22)
bysort mofd: egen meanp1_23=mean(p1_23)
bysort mofd: egen meanp10_23=mean(p10_23)
bysort mofd: egen meanp1_24=mean(p1_24)
bysort mofd: egen meanp10_24=mean(p10_24)
```

// Create momentum return, winners-losers

```
gen mom1=meanp10_1-meanp1_1
gen mom2=meanp10_2-meanp1_2
gen mom3=meanp10_3-meanp1_3
gen mom4=meanp10_4-meanp1_4
gen mom5=meanp10_5-meanp1_5
gen mom6=meanp10_6-meanp1_6
gen mom7=meanp10_7-meanp1_7
gen mom8=meanp10_8-meanp1_8
gen mom9=meanp10_9-meanp1_9
gen mom10=meanp10_10-meanp1_10
gen mom11=meanp10_11-meanp1_11
gen mom12=meanp10_12-meanp1_12
gen mom13=meanp10_13-meanp1_13
```

```
gen mom14=meanp10_14-meanp1_14
gen mom15=meanp10_15-meanp1_15
gen mom16=meanp10_16-meanp1_16
gen mom17=meanp10_17-meanp1_17
gen mom18=meanp10_18-meanp1_18
gen mom19=meanp10_19-meanp1_19
gen mom20=meanp10_20-meanp1_20
gen mom21=meanp10_21-meanp1_21
gen mom22=meanp10_22-meanp1_22
gen mom23=meanp10_23-meanp1_23
gen mom24=meanp10_24-meanp1_24


//drop duplicates, one average return for each month
duplicates drop mofd, force


// Table of descriptive statistics
//mean return is expressed as monthly return ,in percentage form
local varlist mom1 mom2 mom3 mom4 mom5 mom6 mom7 mom8 mom9 mom10 mom11 mom12
mom13 mom14 mom15 mom16 mom17 mom18 mom19 mom20 mom21 mom22 mom23 mom24
local n : word count `varlist'
matrix define A = J(`n',3,.)
mat rownames A = `varlist'
mat colnames A = Mean SD T
local row = 1
foreach var of varlist `varlist' {
   qui summarize `var'
   mat A[`row',1] = r(mean)/6*100
   mat A[`row',2] = r(sd)
   qui reg `var'
   mat A[`row',3] = _b[_cons] / _se[_cons]
   local ++row
}
mat list A, format(%9.2f)
```