

CODE - INDUSTRY MOMENTUM

Table of Contents

//Whole Period J=6 & K=6	2
//Covid-19 – Disruption Period.....	6
//Covid-19 – Recession period	11
//Covid-19 – Post-Recession Period	16
//The Financial Crisis – Recession Period.....	21
//The Financial Crisis – Post-Recession Period	26
//The Dot-Com Bubble – Recession Period	31
//The Dot-Com Bubble– Post-Recession Period	36

//Whole Period J=6 & K=6

```
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 SHRCDD shrcd  
rename EXCHCD exchcd  
rename SICCD siccd  
rename PRC prc  
rename RET ret  
rename SHROUT shrout
```

```
//Translate time variable to month and make it readable  
gen mofd = mofd(date)  
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
```

//Sort by industry

```
//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
```

```
//Name of industry  
gen sic2 = "Mining" if sic == 1  
replace sic2 = "Food" if sic == 2  
replace sic2 = "Apparel" if sic == 3
```

```

replace sic2 = "Paper" if sic == 4
replace sic2 = "Chemical" if sic == 5
replace sic2 = "Petroleum" if sic == 6
replace sic2 = "Construction" if sic == 7
replace sic2 = "Prim. Metals" if sic == 8
replace sic2 = "Fab. Metals" if sic == 9
replace sic2 = "Machinery" if sic == 10
replace sic2 = "Electrical Eq." if sic == 11
replace sic2 = "Transport Eq." if sic == 12
replace sic2 = "Manufacturing" if sic == 13
replace sic2 = "Transportation" if sic == 14
replace sic2 = "Other Transport." if sic == 15
replace sic2 = "Air transportation" if sic == 16
replace sic2 = "Utilities" if sic == 17
replace sic2 = "Dept. Stores" if sic == 18
replace sic2 = "Retail" if sic == 19
replace sic2 = "Finance" if sic == 20
replace sic2 = "Technology" if sic == 21
replace sic2 = "Hotels & Social Ser." if sic == 22
replace sic2 = "Health & Membership" if sic == 23
replace sic2 = "Other" if sic == 24

```

//Create formation period and holding period per industry group

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
sort mofd

//Holding period: Make returns for 6 months holding period

```

gen fret = ret
sort permno mofd
by permno: replace fret = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret)

```

//Average monthly return per industry group

```

bysort mofd sic: egen preret6_mean= mean(preret6_skip)
bysort mofd sic: egen fret6_mean= mean(fret)

```

duplicates drop mofd sic, force

//Rank industry return based on past 6-month return

```

bysort mofd: egen rank = rank(preret6_mean) if preret6_mean<.

```

//Generate portfolios based on formation period returns

```

bysort mofd: gen losers= fret6_mean if rank==1 | rank==2 | rank==3
bysort mofd: gen p2= fret6_mean if rank==4 | rank==5 | rank==6
bysort mofd: gen p3= fret6_mean if rank==7 | rank==8 | rank==9
bysort mofd: gen p4= fret6_mean if rank==10 | rank==11 | rank==12
bysort mofd: gen p5= fret6_mean if rank==13 | rank==14 | rank==15
bysort mofd: gen p6= fret6_mean if rank==16 | rank==17 | rank==18
bysort mofd: gen p7= fret6_mean if rank == 19 | rank==20 | rank==21
bysort mofd: gen winners= fret6_mean if rank==22 | rank==23 | rank==24

```

//Generate mean returns for every month

```

bysort mofd: egen meanlosers=mean(losers)

```

```
bysort mofd: egen meanp2=mean(p2)
bysort mofd: egen meanp3=mean(p3)
bysort mofd: egen meanp4=mean(p4)
bysort mofd: egen meanp5=mean(p5)
bysort mofd: egen meanp6=mean(p6)
bysort mofd: egen meanp7=mean(p7)
bysort mofd: egen meanwinners=mean(winners)
```

// Create momentum return

```
gen mom=meanwinners-meanlosers
duplicates drop mofd, force
```

//Clean data

```
drop ret
drop fret
drop losers
drop p2
drop p3
drop p4
drop p5
drop p6
drop p7
drop winners
drop preret6_skip
drop rank
```

//Perform tests

```
//Merge with market returns
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-
months individual mom\famafrench.dta"
```

```
drop _merge
```

//Average monthly raw return

```
gen losers=meanlosers/6
gen p2=meanp2/6
gen p3=meanp3/6
gen p4=meanp4/6
gen p5=meanp5/6
gen p6=meanp6/6
gen p7=meanp7/6
gen winners=meanwinners/6
replace mom=mom/6
```

//Excess returns

```
gen losers_excessret = losers-rf
gen p2_excessret = p2-rf
gen p3_excessret = p3-rf
gen p4_excessret = p4-rf
gen p5_excessret = p5-rf
gen p6_excessret = p6-rf
gen p7_excessret = p7-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf
```

//Results from tests

```
// Table of descriptive statistics: mean raw monthly return, and t-statistics
//mean return is expressed as monthly return in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = mean t-stat
local row = 1
foreach var of varlist `varlist' {
    qui summarize `var'
    qui reg `var'
        mat A[`row',1] = _b[_cons]*100
    mat A[`row',2] = _b[_cons] / _se[_cons]
    local ++row
}
matlist A, border(rows) rowtitle(Portfolio) format(%8.2f) twidth(10)
```

//Table of overall descriptive statistic

```
//Variables based on excess return
local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret
p7_excessret winners_excessret mom_excessret
local n : word count `varlist'
matrix define A = J(`n',5,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = "FF3" "FF3 t-stat" "FF5" "FF5 t-stat" "Sharpe ratio"
local row = 1
foreach var of varlist `varlist' {
    qui reg `var' mktf smb hml
    mat A[`row',1] = _b[_cons]*100
    mat A[`row',2] = _b[_cons] / _se[_cons]
    qui reg `var' mktf smb hml rmw cma
    mat A[`row',3] = _b[_cons]*100
    mat A[`row',4] = _b[_cons] / _se[_cons]
    qui summarize `var'
        mat A[`row',5] = r(mean)/r(sd)
    local ++row
}
matlist A, border(rows) format(%8.2f)
```

//Covid-19 – Disruption 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
```

```
//Translate time variable to month and make it readable
//Keep only dates needed
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
```

//Sort by industry

```
//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
```

```
//Name of industry
gen sic2 = "Mining" if sic == 1
replace sic2 = "Food" if sic == 2
replace sic2 = "Apparel" if sic == 3
replace sic2 = "Paper" if sic == 4
replace sic2 = "Chemical" if sic == 5
replace sic2 = "Petroleum" if sic == 6
replace sic2 = "Construction" if sic == 7
replace sic2 = "Prim. Metals" if sic == 8
replace sic2 = "Fab. Metals" if sic == 9
replace sic2 = "Machinery" if sic == 10
replace sic2 = "Electrical Eq." if sic == 11
replace sic2 = "Transport Eq." if sic == 12
replace sic2 = "Manufacturing" if sic == 13
replace sic2 = "Transportation" if sic == 14
replace sic2 = "Other Transport." if sic == 15
replace sic2 = "Air transportation" if sic == 16
replace sic2 = "Utilities" if sic == 17
replace sic2 = "Dept. Stores" if sic == 18
replace sic2 = "Retail" if sic == 19
replace sic2 = "Finance" if sic == 20
replace sic2 = "Technology" if sic == 21
replace sic2 = "Hotels & Social Ser." if sic == 22
replace sic2 = "Health & Membership" if sic == 23
replace sic2 = "Other" if sic == 24
```

//Create formation period and holding period per industry group

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
sort mofd

//Holding period: Make 6-month returns for holding period

```
gen fret = ret
sort permno mofd
by permno: replace fret = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret)
```

//Average 6-month past return per industry group

```
tostring mofd, generate(time)
bysort mofd sic: egen preret6_mean= mean(preret6_skip) if inrange(time, "720", "731")
bysort mofd sic: egen fret6_mean= mean(fret)
```

//Average monthly return per industry group to use in table

```
bysort sic: egen meanret_industry = mean(ret) if inrange(time, "720", "731")
replace meanret_industry = meanret_industry*100
```

```
duplicates drop mofd sic, force
```

//Rank industry return based on past return: *the command creates the variable port which contains 7 quantiles which contains 3 industry returns per portfolio*

```
bysort mofd: egen rank = rank(preret6_mean) if preret6_mean<.
```

//Generate portfolios based on formation period returns

```
bysort mofd: gen losers= fret6_mean if rank==1 | rank==2 | rank==3  
bysort mofd: gen p2= fret6_mean if rank==4 | rank==5 | rank==6  
bysort mofd: gen p3= fret6_mean if rank==7 | rank==8 | rank==9  
bysort mofd: gen p4= fret6_mean if rank==10 | rank==11 | rank==12  
bysort mofd: gen p5= fret6_mean if rank==13 | rank==14 | rank==15  
bysort mofd: gen p6= fret6_mean if rank==16 | rank==17 | rank==18  
bysort mofd: gen p7= fret6_mean if rank == 19 | rank==20 | rank==21  
bysort mofd: gen winners= fret6_mean if rank==22 | rank==23 | rank==24
```

//Table: Frequency of winners and losers

```
bysort sic: gen lossess=1 if rank == 1 | rank == 2 | rank==3  
bysort sic: egen losses=count(lossess)  
bysort sic: gen winss=1 if rank == 22 | rank == 23 | rank==24  
bysort sic: egen wins=count(winss)  
drop lossess  
drop winss  
tabstat wins losses meanret_industry, by(sic) stat(mean) format(%8.2f)
```

//Calculate mean returns for every month

```
bysort mofd: egen meanlosers=mean(losers)  
bysort mofd: egen meanp2=mean(p2)  
bysort mofd: egen meanp3=mean(p3)  
bysort mofd: egen meanp4=mean(p4)  
bysort mofd: egen meanp5=mean(p5)  
bysort mofd: egen meanp6=mean(p6)  
bysort mofd: egen meanp7=mean(p7)  
bysort mofd: egen meanwinners=mean(winners)
```

// Create momentum return, winners-losers

```
gen mom=meanwinners-meanlosers  
duplicates drop mofd, force
```

//Clean data

```
drop ret  
drop fret  
drop losers  
drop p2  
drop p3  
drop p4  
drop p5  
drop p6  
drop p7  
drop winners
```



```
drop preret6_skip
drop rank
```

//Perform tests

```
//Merge with market returns
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-
months individual mom\famafrench.dta"
drop _merge
```

```
//Average monthly raw return
gen losers=meanlosers/6
gen p2=meanp2/6
gen p3=meanp3/6
gen p4=meanp4/6
gen p5=meanp5/6
gen p6=meanp6/6
gen p7=meanp7/6
gen winners=meanwinners/6
replace mom=mom/6
```

```
//Excess returns
gen losers_excessret = losers-rf
gen p2_excessret = p2-rf
gen p3_excessret = p3-rf
gen p4_excessret = p4-rf
gen p5_excessret = p5-rf
gen p6_excessret = p6-rf
gen p7_excessret = p7-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf
```

// Results from tests

```
// Table of descriptive statistics: mean return, and t-statistics
//mean return is expressed as monthly return , in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = mean t-stat
local row = 1
foreach var of varlist `varlist' {
    qui summarize `var'
    qui reg `var'
        mat A[`row',1] = _b[_cons]*100
        mat A[`row',2] = _b[_cons] / _se[_cons]
    local ++row
}
matlist A, border(rows) rowtitle(Portfolio) format(%8.2f) twidth(10)
```

```

//Table of overall descriptive statistic
//Variables based on excess return
local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret
p7_excessret winners_excessret mom_excessret
local n : word count `varlist'
matrix define A = J(`n',5,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = "FF3" "FF3 t-stat" "FF5" "FF5 t-stat" "Sharpe ratio"
local row = 1
foreach var of varlist `varlist' {
    qui reg `var' mktf smb hml
    mat A[`row',1] = _b[_cons]*100
    mat A[`row',2] = _b[_cons] / _se[_cons]
    qui reg `var' mktf smb hml rmw cma
    mat A[`row',3] = _b[_cons]*100
    mat A[`row',4] = _b[_cons] / _se[_cons]
    qui summarize `var'
    mat A[`row',5] = r(mean)/r(sd)
    local ++row
}
matlist A, border(rows) rowtitle(Return statistic) format(%8.2f)

```

//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 SHRCDD shrcd

rename EXCHCD exchcd

rename SICCD siccd

rename PRC prc

rename RET ret

rename SHROUT shrout

//Translate time variable to month and make it readable

//Keep only dates needed

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

//Sort by industry

//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
```

```
//Name of industry
gen sic2 = "Mining" if sic == 1
replace sic2 = "Food" if sic == 2
replace sic2 = "Apparel" if sic == 3
replace sic2 = "Paper" if sic == 4
replace sic2 = "Chemical" if sic == 5
replace sic2 = "Petroleum" if sic == 6
replace sic2 = "Construction" if sic == 7
replace sic2 = "Prim. Metals" if sic == 8
replace sic2 = "Fab. Metals" if sic == 9
replace sic2 = "Machinery" if sic == 10
replace sic2 = "Electrical Eq." if sic == 11
replace sic2 = "Transport Eq." if sic == 12
replace sic2 = "Manufacturing" if sic == 13
replace sic2 = "Transportation" if sic == 14
replace sic2 = "Other Transport." if sic == 15
replace sic2 = "Air transportation" if sic == 16
replace sic2 = "Utilities" if sic == 17
replace sic2 = "Dept. Stores" if sic == 18
replace sic2 = "Retail" if sic == 19
replace sic2 = "Finance" if sic == 20
replace sic2 = "Technology" if sic == 21
replace sic2 = "Hotels & Social Ser." if sic == 22
replace sic2 = "Health & Membership" if sic == 23
replace sic2 = "Other" if sic == 24
```

//Create formation period and holding period per industry group

```
//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
sort mofd
```

//Holding period: Make 6-month returns for holding period

```
gen fret = ret
sort permno mofd
by permno: replace fret = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret)
```

//Average 6-month past return per industry group

```
tostring mofd, generate(time)
bysort mofd sic: egen preret6_mean= mean(preret6_skip) if inrange(time, "721", "723")
bysort mofd sic: egen fret6_mean= mean(fret)
```

//Average monthly return per industry group to use in table

```
bysort sic: egen meanret_industry = mean(ret) if inrange(time, "721", "723")
replace meanret_industry = meanret_industry*100
```

```
duplicates drop mofd sic, force
```

//Rank industry return based on past return: *the command creates the variable port which contains 7 quantiles which contains 3 industry returns per portfolio*

```
bysort mofd: egen rank = rank(preret6_mean) if preret6_mean<.
```

//Generate portfolios based on formation period returns

```
bysort mofd: gen losers= fret6_mean if rank==1 | rank==2 | rank==3  
bysort mofd: gen p2= fret6_mean if rank==4 | rank==5 | rank==6  
bysort mofd: gen p3= fret6_mean if rank==7 | rank==8 | rank==9  
bysort mofd: gen p4= fret6_mean if rank==10 | rank==11 | rank==12  
bysort mofd: gen p5= fret6_mean if rank==13 | rank==14 | rank==15  
bysort mofd: gen p6= fret6_mean if rank==16 | rank==17 | rank==18  
bysort mofd: gen p7= fret6_mean if rank == 19 | rank==20 | rank==21  
bysort mofd: gen winners= fret6_mean if rank==22 | rank==23 | rank==24
```

//Table: Frequency of winners and losers

```
bysort sic: gen lossess=1 if rank == 1 | rank == 2 | rank==3  
bysort sic: egen losses=count(lossess)  
bysort sic: gen winss=1 if rank == 22 | rank == 23 | rank==24  
bysort sic: egen wins=count(winss)  
drop lossess  
drop winss  
tabstat wins losses meanret_industry, by(sic) stat(mean) format(%8.2f)
```

//Calculate mean returns for every month

```
bysort mofd: egen meanlosers=mean(losers)  
bysort mofd: egen meanp2=mean(p2)  
bysort mofd: egen meanp3=mean(p3)  
bysort mofd: egen meanp4=mean(p4)  
bysort mofd: egen meanp5=mean(p5)  
bysort mofd: egen meanp6=mean(p6)  
bysort mofd: egen meanp7=mean(p7)  
bysort mofd: egen meanwinners=mean(winners)
```

// Create momentum return, winners-losers

```
gen mom=meanwinners-meanlosers  
duplicates drop mofd, force
```

//Clean data

```
drop ret  
drop fret  
drop losers  
drop p2  
drop p3  
drop p4  
drop p5  
drop p6  
drop p7  
drop winners
```

```
drop preret6_skip
drop rank
```

//Perform tests

```
//Merge with market returns
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-
months individual mom\famafrench.dta"
drop _merge
```

```
//Average monthly raw return
gen losers=meanlosers/6
gen p2=meanp2/6
gen p3=meanp3/6
gen p4=meanp4/6
gen p5=meanp5/6
gen p6=meanp6/6
gen p7=meanp7/6
gen winners=meanwinners/6
replace mom=mom/6
```

```
//Excess returns
gen losers_excessret = losers-rf
gen p2_excessret = p2-rf
gen p3_excessret = p3-rf
gen p4_excessret = p4-rf
gen p5_excessret = p5-rf
gen p6_excessret = p6-rf
gen p7_excessret = p7-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf
```

// Results from tests

```
// Table of descriptive statistics: mean return, and t-statistics
//mean return is expressed as monthly return , in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = mean t-stat
local row = 1
foreach var of varlist `varlist' {
    qui summarize `var'
    qui reg `var'
        mat A[`row',1] = _b[_cons]*100
        mat A[`row',2] = _b[_cons] / _se[_cons]
    local ++row
}
matlist A, border(rows) rowtitle(Portfolio) format(%8.2f) twidth(10)
```

```

//Table of overall descriptive statistic
//Variables based on excess return
local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret
p7_excessret winners_excessret mom_excessret
local n : word count `varlist'
matrix define A = J(`n',5,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = "FF3" "FF3 t-stat" "FF5" "FF5 t-stat" "Sharpe ratio"
local row = 1
foreach var of varlist `varlist' {
    qui reg `var' mktf smb hml
    mat A[`row',1] = _b[_cons]*100
    mat A[`row',2] = _b[_cons] / _se[_cons]
    qui reg `var' mktf smb hml rmw cma
    mat A[`row',3] = _b[_cons]*100
    mat A[`row',4] = _b[_cons] / _se[_cons]
    qui summarize `var'
    mat A[`row',5] = r(mean)/r(sd)
    local ++row
}
matlist A, border(rows) rowtitle(Return statistic) format(%8.2f)

```

//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 SHRCDD shrcd

rename EXCHCD exchcd

rename SICCD siccd

rename PRC prc

rename RET ret

rename SHROUT shrout

//Translate time variable to month and make it readable

//Keep only dates needed

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

//Sort by industry

//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
```

```
//Name of industry
gen sic2 = "Mining" if sic == 1
replace sic2 = "Food" if sic == 2
replace sic2 = "Apparel" if sic == 3
replace sic2 = "Paper" if sic == 4
replace sic2 = "Chemical" if sic == 5
replace sic2 = "Petroleum" if sic == 6
replace sic2 = "Construction" if sic == 7
replace sic2 = "Prim. Metals" if sic == 8
replace sic2 = "Fab. Metals" if sic == 9
replace sic2 = "Machinery" if sic == 10
replace sic2 = "Electrical Eq." if sic == 11
replace sic2 = "Transport Eq." if sic == 12
replace sic2 = "Manufacturing" if sic == 13
replace sic2 = "Transportation" if sic == 14
replace sic2 = "Other Transport." if sic == 15
replace sic2 = "Air transportation" if sic == 16
replace sic2 = "Utilities" if sic == 17
replace sic2 = "Dept. Stores" if sic == 18
replace sic2 = "Retail" if sic == 19
replace sic2 = "Finance" if sic == 20
replace sic2 = "Technology" if sic == 21
replace sic2 = "Hotels & Social Ser." if sic == 22
replace sic2 = "Health & Membership" if sic == 23
replace sic2 = "Other" if sic == 24
```

//Create formation period and holding period per industry group

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
sort mofd

//Holding period: Make 6-month returns for holding period

```
gen fret = ret
sort permno mofd
by permno: replace fret = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret)
```

//Average 6-month past return per industry group

```
tostring mofd, generate(time)
bysort mofd sic: egen preret6_mean= mean(preret6_skip) if inrange(time, "724", "735")
bysort mofd sic: egen fret6_mean= mean(fret)
```

//Average monthly return per industry group to use in table

```
bysort sic: egen meanret_industry = mean(ret) if inrange(time, "724", "735")
replace meanret_industry = meanret_industry*100
```

```
duplicates drop mofd sic, force
```

//Rank industry return based on past return: *the command creates the variable port which contains 7 quantiles which contains 3 industry returns per portfolio*

```
bysort mofd: egen rank = rank(preret6_mean) if preret6_mean<.
```

//Generate portfolios based on formation period returns

```
bysort mofd: gen losers= fret6_mean if rank==1 | rank==2 | rank==3
```

```
bysort mofd: gen p2= fret6_mean if rank==4 | rank==5 | rank==6
```

```
bysort mofd: gen p3= fret6_mean if rank==7 | rank==8 | rank==9
```

```
bysort mofd: gen p4= fret6_mean if rank==10 | rank==11 | rank==12
```

```
bysort mofd: gen p5= fret6_mean if rank==13 | rank==14 | rank==15
```

```
bysort mofd: gen p6= fret6_mean if rank==16 | rank==17 | rank==18
```

```
bysort mofd: gen p7= fret6_mean if rank == 19 | rank==20 | rank==21
```

```
bysort mofd: gen winners= fret6_mean if rank==22 | rank==23 | rank==24
```

//Table: Frequency of winners and losers

```
bysort sic: gen lossess=1 if rank == 1 | rank == 2 | rank==3
```

```
bysort sic: egen losses=count(lossess)
```

```
bysort sic: gen winss=1 if rank == 22 | rank == 23 | rank==24
```

```
bysort sic: egen wins=count(winss)
```

```
drop lossess
```

```
drop winss
```

```
tabstat wins losses meanret_industry, by(sic) stat(mean) format(%8.2f)
```

//Calculate mean returns for every month

```
bysort mofd: egen meanlosers=mean(losers)
```

```
bysort mofd: egen meanp2=mean(p2)
```

```
bysort mofd: egen meanp3=mean(p3)
```

```
bysort mofd: egen meanp4=mean(p4)
```

```
bysort mofd: egen meanp5=mean(p5)
```

```
bysort mofd: egen meanp6=mean(p6)
```

```
bysort mofd: egen meanp7=mean(p7)
```

```
bysort mofd: egen meanwinners=mean(winners)
```

// Create momentum return, winners-losers

```
gen mom=meanwinners-meanlosers
```

```
duplicates drop mofd, force
```

//Clean data

```
drop ret
```

```
drop fret
```

```
drop losers
```

```
drop p2
```

```
drop p3
```

```
drop p4
```

```
drop p5
```

```
drop p6
```

```
drop p7
```

```
drop winners
```

```
drop preret6_skip
drop rank
```

//Perform tests

```
//Merge with market returns
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-
months individual mom\famafrench.dta"
drop _merge
```

```
//Average monthly raw return
gen losers=meanlosers/6
gen p2=meanp2/6
gen p3=meanp3/6
gen p4=meanp4/6
gen p5=meanp5/6
gen p6=meanp6/6
gen p7=meanp7/6
gen winners=meanwinners/6
replace mom=mom/6
```

```
//Excess returns
gen losers_excessret = losers-rf
gen p2_excessret = p2-rf
gen p3_excessret = p3-rf
gen p4_excessret = p4-rf
gen p5_excessret = p5-rf
gen p6_excessret = p6-rf
gen p7_excessret = p7-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf
```

// Results from tests

```
// Table of descriptive statistics: mean return, and t-statistics
//mean return is expressed as monthly return , in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = mean t-stat
local row = 1
foreach var of varlist `varlist' {
    qui summarize `var'
    qui reg `var'
        mat A[`row',1] = _b[_cons]*100
        mat A[`row',2] = _b[_cons] / _se[_cons]
    local ++row
}
matlist A, border(rows) rowtitle(Portfolio) format(%8.2f) twidth(10)
```

```

//Table of overall descriptive statistic
//Variables based on excess return
local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret
p7_excessret winners_excessret mom_excessret
local n : word count `varlist'
matrix define A = J(`n',5,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = "FF3" "FF3 t-stat" "FF5" "FF5 t-stat" "Sharpe ratio"
local row = 1
foreach var of varlist `varlist' {
    qui reg `var' mktf smb hml
    mat A[`row',1] = _b[_cons]*100
    mat A[`row',2] = _b[_cons] / _se[_cons]
    qui reg `var' mktf smb hml rmw cma
    mat A[`row',3] = _b[_cons]*100
    mat A[`row',4] = _b[_cons] / _se[_cons]
    qui summarize `var'
    mat A[`row',5] = r(mean)/r(sd)
    local ++row
}
matlist A, border(rows) rowtitle(Return statistic) format(%8.2f)

```

//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 SHRCDD shrcd

rename EXCHCD exchcd

rename SICCD siccd

rename PRC prc

rename RET ret

rename SHROUT shrout

//Translate time variable to month and make it readable

//Keep only dates needed

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

//Sort by industry

//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
```

```
//Name of industry
gen sic2 = "Mining" if sic == 1
replace sic2 = "Food" if sic == 2
replace sic2 = "Apparel" if sic == 3
replace sic2 = "Paper" if sic == 4
replace sic2 = "Chemical" if sic == 5
replace sic2 = "Petroleum" if sic == 6
replace sic2 = "Construction" if sic == 7
replace sic2 = "Prim. Metals" if sic == 8
replace sic2 = "Fab. Metals" if sic == 9
replace sic2 = "Machinery" if sic == 10
replace sic2 = "Electrical Eq." if sic == 11
replace sic2 = "Transport Eq." if sic == 12
replace sic2 = "Manufacturing" if sic == 13
replace sic2 = "Transportation" if sic == 14
replace sic2 = "Other Transport." if sic == 15
replace sic2 = "Air transportation" if sic == 16
replace sic2 = "Utilities" if sic == 17
replace sic2 = "Dept. Stores" if sic == 18
replace sic2 = "Retail" if sic == 19
replace sic2 = "Finance" if sic == 20
replace sic2 = "Technology" if sic == 21
replace sic2 = "Hotels & Social Ser." if sic == 22
replace sic2 = "Health & Membership" if sic == 23
replace sic2 = "Other" if sic == 24
```

//Create formation period and holding period per industry group

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
sort mofd

//Holding period: Make 6-month returns for holding period

```
gen fret = ret
sort permno mofd
by permno: replace fret = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret)
```

//Average 6-month past return per industry group

```
tostring mofd, generate(time)
bysort mofd sic: egen preret6_mean= mean(preret6_skip) if inrange(time, "575", "593")
bysort mofd sic: egen fret6_mean= mean(fret)
```

//Average monthly return per industry group to use in table

```
bysort sic: egen meanret_industry = mean(ret) if inrange(time, "575", "593")
replace meanret_industry = meanret_industry*100
```

```
duplicates drop mofd sic, force
```

//Rank industry return based on past return: *the command creates the variable port which contains 7 quantiles which contains 3 industry returns per portfolio*

```
bysort mofd: egen rank = rank(preret6_mean) if preret6_mean<.
```

//Generate portfolios based on formation period returns

```
bysort mofd: gen losers= fret6_mean if rank==1 | rank==2 | rank==3  
bysort mofd: gen p2= fret6_mean if rank==4 | rank==5 | rank==6  
bysort mofd: gen p3= fret6_mean if rank==7 | rank==8 | rank==9  
bysort mofd: gen p4= fret6_mean if rank==10 | rank==11 | rank==12  
bysort mofd: gen p5= fret6_mean if rank==13 | rank==14 | rank==15  
bysort mofd: gen p6= fret6_mean if rank==16 | rank==17 | rank==18  
bysort mofd: gen p7= fret6_mean if rank == 19 | rank==20 | rank==21  
bysort mofd: gen winners= fret6_mean if rank==22 | rank==23 | rank==24
```

//Table: Frequency of winners and losers

```
bysort sic: gen lossess=1 if rank == 1 | rank == 2 | rank==3  
bysort sic: egen losses=count(lossess)  
bysort sic: gen winss=1 if rank == 22 | rank == 23 | rank==24  
bysort sic: egen wins=count(winss)  
drop lossess  
drop winss  
tabstat wins losses meanret_industry, by(sic) stat(mean) format(%8.2f)
```

//Calculate mean returns for every month

```
bysort mofd: egen meanlosers=mean(losers)  
bysort mofd: egen meanp2=mean(p2)  
bysort mofd: egen meanp3=mean(p3)  
bysort mofd: egen meanp4=mean(p4)  
bysort mofd: egen meanp5=mean(p5)  
bysort mofd: egen meanp6=mean(p6)  
bysort mofd: egen meanp7=mean(p7)  
bysort mofd: egen meanwinners=mean(winners)
```

// Create momentum return, winners-losers

```
gen mom=meanwinners-meanlosers  
duplicates drop mofd, force
```

//Clean data

```
drop ret  
drop fret  
drop losers  
drop p2  
drop p3  
drop p4  
drop p5  
drop p6  
drop p7  
drop winners
```

```
drop preret6_skip
drop rank
```

//Perform tests

```
//Merge with market returns
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-
months individual mom\famafrench.dta"
drop _merge
```

```
//Average monthly raw return
gen losers=meanlosers/6
gen p2=meanp2/6
gen p3=meanp3/6
gen p4=meanp4/6
gen p5=meanp5/6
gen p6=meanp6/6
gen p7=meanp7/6
gen winners=meanwinners/6
replace mom=mom/6
```

```
//Excess returns
gen losers_excessret = losers-rf
gen p2_excessret = p2-rf
gen p3_excessret = p3-rf
gen p4_excessret = p4-rf
gen p5_excessret = p5-rf
gen p6_excessret = p6-rf
gen p7_excessret = p7-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf
```

// Results from tests

```
// Table of descriptive statistics: mean return, and t-statistics
//mean return is expressed as monthly return , in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = mean t-stat
local row = 1
foreach var of varlist `varlist' {
    qui summarize `var'
    qui reg `var'
        mat A[`row',1] = _b[_cons]*100
        mat A[`row',2] = _b[_cons] / _se[_cons]
    local ++row
}
matlist A, border(rows) rowtitle(Portfolio) format(%8.2f) twidth(10)
```



```

//Table of overall descriptive statistic
//Variables based on excess return
local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret
p7_excessret winners_excessret mom_excessret
local n : word count `varlist'
matrix define A = J(`n',5,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = "FF3" "FF3 t-stat" "FF5" "FF5 t-stat" "Sharpe ratio"
local row = 1
foreach var of varlist `varlist' {
    qui reg `var' mktf smb hml
    mat A[`row',1] = _b[_cons]*100
    mat A[`row',2] = _b[_cons] / _se[_cons]
    qui reg `var' mktf smb hml rmw cma
    mat A[`row',3] = _b[_cons]*100
    mat A[`row',4] = _b[_cons] / _se[_cons]
    qui summarize `var'
    mat A[`row',5] = r(mean)/r(sd)
    local ++row
}
matlist A, border(rows) rowtitle(Return statistic) format(%8.2f)

```

//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 SHRCDD shrccd

rename EXCHCD exchcd

rename SICCD siccd

rename PRC prc

rename RET ret

rename SHROUT shrout

//Translate time variable to month and make it readable

//Keep only dates needed

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 shrccd == 10 | shrccd == 11

//Remove share price below 5

drop if prc < 5

//Sort by industry

//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
```

```
//Name of industry
gen sic2 = "Mining" if sic == 1
replace sic2 = "Food" if sic == 2
replace sic2 = "Apparel" if sic == 3
replace sic2 = "Paper" if sic == 4
replace sic2 = "Chemical" if sic == 5
replace sic2 = "Petroleum" if sic == 6
replace sic2 = "Construction" if sic == 7
replace sic2 = "Prim. Metals" if sic == 8
replace sic2 = "Fab. Metals" if sic == 9
replace sic2 = "Machinery" if sic == 10
replace sic2 = "Electrical Eq." if sic == 11
replace sic2 = "Transport Eq." if sic == 12
replace sic2 = "Manufacturing" if sic == 13
replace sic2 = "Transportation" if sic == 14
replace sic2 = "Other Transport." if sic == 15
replace sic2 = "Air transportation" if sic == 16
replace sic2 = "Utilities" if sic == 17
replace sic2 = "Dept. Stores" if sic == 18
replace sic2 = "Retail" if sic == 19
replace sic2 = "Finance" if sic == 20
replace sic2 = "Technology" if sic == 21
replace sic2 = "Hotels & Social Ser." if sic == 22
replace sic2 = "Health & Membership" if sic == 23
replace sic2 = "Other" if sic == 24
```

//Create formation period and holding period per industry group

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
sort mofd

//Holding period: Make 6-month returns for holding period

```
gen fret = ret
sort permno mofd
by permno: replace fret = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret)
```

//Average 6-month past return per industry group

```
tostring mofd, generate(time)
bysort mofd sic: egen preret6_mean= mean(preret6_skip) if inrange(time, "594", "605")
bysort mofd sic: egen fret6_mean= mean(fret)
```

//Average monthly return per industry group to use in table

```
bysort sic: egen meanret_industry = mean(ret) if inrange(time, "594", "605")
replace meanret_industry = meanret_industry*100
```

```
duplicates drop mofd sic, force
```

//Rank industry return based on past return: *the command creates the variable port which contains 7 quantiles which contains 3 industry returns per portfolio*

```
bysort mofd: egen rank = rank(preret6_mean) if preret6_mean<.
```

//Generate portfolios based on formation period returns

```
bysort mofd: gen losers= fret6_mean if rank==1 | rank==2 | rank==3  
bysort mofd: gen p2= fret6_mean if rank==4 | rank==5 | rank==6  
bysort mofd: gen p3= fret6_mean if rank==7 | rank==8 | rank==9  
bysort mofd: gen p4= fret6_mean if rank==10 | rank==11 | rank==12  
bysort mofd: gen p5= fret6_mean if rank==13 | rank==14 | rank==15  
bysort mofd: gen p6= fret6_mean if rank==16 | rank==17 | rank==18  
bysort mofd: gen p7= fret6_mean if rank == 19 | rank==20 | rank==21  
bysort mofd: gen winners= fret6_mean if rank==22 | rank==23 | rank==24
```

//Table: Frequency of winners and losers

```
bysort sic: gen lossess=1 if rank == 1 | rank == 2 | rank==3  
bysort sic: egen losses=count(lossess)  
bysort sic: gen winss=1 if rank == 22 | rank == 23 | rank==24  
bysort sic: egen wins=count(winss)  
drop lossess  
drop winss  
tabstat wins losses meanret_industry, by(sic) stat(mean) format(%8.2f)
```

//Calculate mean returns for every month

```
bysort mofd: egen meanlosers=mean(losers)  
bysort mofd: egen meanp2=mean(p2)  
bysort mofd: egen meanp3=mean(p3)  
bysort mofd: egen meanp4=mean(p4)  
bysort mofd: egen meanp5=mean(p5)  
bysort mofd: egen meanp6=mean(p6)  
bysort mofd: egen meanp7=mean(p7)  
bysort mofd: egen meanwinners=mean(winners)
```

// Create momentum return, winners-losers

```
gen mom=meanwinners-meanlosers  
duplicates drop mofd, force
```

//Clean data

```
drop ret  
drop fret  
drop losers  
drop p2  
drop p3  
drop p4  
drop p5  
drop p6  
drop p7  
drop winners
```

```
drop preret6_skip
drop rank
```

//Perform tests

```
//Merge with market returns
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-
months individual mom\famafrench.dta"
drop _merge
```

```
//Average monthly raw return
gen losers=meanlosers/6
gen p2=meanp2/6
gen p3=meanp3/6
gen p4=meanp4/6
gen p5=meanp5/6
gen p6=meanp6/6
gen p7=meanp7/6
gen winners=meanwinners/6
replace mom=mom/6
```

```
//Excess returns
gen losers_excessret = losers-rf
gen p2_excessret = p2-rf
gen p3_excessret = p3-rf
gen p4_excessret = p4-rf
gen p5_excessret = p5-rf
gen p6_excessret = p6-rf
gen p7_excessret = p7-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf
```

// Results from tests

```
// Table of descriptive statistics: mean return, and t-statistics
//mean return is expressed as monthly return , in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = mean t-stat
local row = 1
foreach var of varlist `varlist' {
    qui summarize `var'
    qui reg `var'
        mat A[`row',1] = _b[_cons]*100
        mat A[`row',2] = _b[_cons] / _se[_cons]
    local ++row
}
matlist A, border(rows) rowtitle(Portfolio) format(%8.2f) twidth(10)
```

```

//Table of overall descriptive statistic
//Variables based on excess return
local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret
p7_excessret winners_excessret mom_excessret
local n : word count `varlist'
matrix define A = J(`n',5,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = "FF3" "FF3 t-stat" "FF5" "FF5 t-stat" "Sharpe ratio"
local row = 1
foreach var of varlist `varlist' {
    qui reg `var' mktf smb hml
    mat A[`row',1] = _b[_cons]*100
    mat A[`row',2] = _b[_cons] / _se[_cons]
    qui reg `var' mktf smb hml rmw cma
    mat A[`row',3] = _b[_cons]*100
    mat A[`row',4] = _b[_cons] / _se[_cons]
    qui summarize `var'
    mat A[`row',5] = r(mean)/r(sd)
    local ++row
}
matlist A, border(rows) rowtitle(Return statistic) format(%8.2f)

```

//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
```

```
//Translate time variable to month and make it readable
//Keep only dates needed
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
```

//Sort by industry

```
//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
```

```
//Name of industry
gen sic2 = "Mining" if sic == 1
replace sic2 = "Food" if sic == 2
replace sic2 = "Apparel" if sic == 3
replace sic2 = "Paper" if sic == 4
replace sic2 = "Chemical" if sic == 5
replace sic2 = "Petroleum" if sic == 6
replace sic2 = "Construction" if sic == 7
replace sic2 = "Prim. Metals" if sic == 8
replace sic2 = "Fab. Metals" if sic == 9
replace sic2 = "Machinery" if sic == 10
replace sic2 = "Electrical Eq." if sic == 11
replace sic2 = "Transport Eq." if sic == 12
replace sic2 = "Manufacturing" if sic == 13
replace sic2 = "Transportation" if sic == 14
replace sic2 = "Other Transport." if sic == 15
replace sic2 = "Air transportation" if sic == 16
replace sic2 = "Utilities" if sic == 17
replace sic2 = "Dept. Stores" if sic == 18
replace sic2 = "Retail" if sic == 19
replace sic2 = "Finance" if sic == 20
replace sic2 = "Technology" if sic == 21
replace sic2 = "Hotels & Social Ser." if sic == 22
replace sic2 = "Health & Membership" if sic == 23
replace sic2 = "Other" if sic == 24
```

//Create formation period and holding period per industry group

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
sort mofd

//Holding period: Make 6-month returns for holding period

```
gen fret = ret
sort permno mofd
by permno: replace fret = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret)
```

//Average 6-month past return per industry group

```
tostring mofd, generate(time)
bysort mofd sic: egen preret6_mean= mean(preret6_skip) if inrange(time, "494", "502")
bysort mofd sic: egen fret6_mean= mean(fret)
```

//Average monthly return per industry group to use in table

```
bysort sic: egen meanret_industry = mean(ret) if inrange(time, "494", "502")
replace meanret_industry = meanret_industry*100
```

```
duplicates drop mofd sic, force
```


//Rank industry return based on past return: *the command creates the variable port which contains 7 quantiles which contains 3 industry returns per portfolio*

```
bysort mofd: egen rank = rank(preret6_mean) if preret6_mean<.
```

//Generate portfolios based on formation period returns

```
bysort mofd: gen losers= fret6_mean if rank==1 | rank==2 | rank==3
```

```
bysort mofd: gen p2= fret6_mean if rank==4 | rank==5 | rank==6
```

```
bysort mofd: gen p3= fret6_mean if rank==7 | rank==8 | rank==9
```

```
bysort mofd: gen p4= fret6_mean if rank==10 | rank==11 | rank==12
```

```
bysort mofd: gen p5= fret6_mean if rank==13 | rank==14 | rank==15
```

```
bysort mofd: gen p6= fret6_mean if rank==16 | rank==17 | rank==18
```

```
bysort mofd: gen p7= fret6_mean if rank == 19 | rank==20 | rank==21
```

```
bysort mofd: gen winners= fret6_mean if rank==22 | rank==23 | rank==24
```

//Table: Frequency of winners and losers

```
bysort sic: gen lossess=1 if rank == 1 | rank == 2 | rank==3
```

```
bysort sic: egen losses=count(lossess)
```

```
bysort sic: gen winss=1 if rank == 22 | rank == 23 | rank==24
```

```
bysort sic: egen wins=count(winss)
```

```
drop lossess
```

```
drop winss
```

```
tabstat wins losses meanret_industry, by(sic) stat(mean) format(%8.2f)
```

//Calculate mean returns for every month

```
bysort mofd: egen meanlosers=mean(losers)
```

```
bysort mofd: egen meanp2=mean(p2)
```

```
bysort mofd: egen meanp3=mean(p3)
```

```
bysort mofd: egen meanp4=mean(p4)
```

```
bysort mofd: egen meanp5=mean(p5)
```

```
bysort mofd: egen meanp6=mean(p6)
```

```
bysort mofd: egen meanp7=mean(p7)
```

```
bysort mofd: egen meanwinners=mean(winners)
```

// Create momentum return, winners-losers

```
gen mom=meanwinners-meanlosers
```

```
duplicates drop mofd, force
```

//Clean data

```
drop ret
```

```
drop fret
```

```
drop losers
```

```
drop p2
```

```
drop p3
```

```
drop p4
```

```
drop p5
```

```
drop p6
```

```
drop p7
```

```
drop winners
```

```
drop preret6_skip
drop rank
```

//Perform tests

```
//Merge with market returns
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-
months individual mom\famafrench.dta"
drop _merge
```

```
//Average monthly raw return
gen losers=meanlosers/6
gen p2=meanp2/6
gen p3=meanp3/6
gen p4=meanp4/6
gen p5=meanp5/6
gen p6=meanp6/6
gen p7=meanp7/6
gen winners=meanwinners/6
replace mom=mom/6
```

```
//Excess returns
gen losers_excessret = losers-rf
gen p2_excessret = p2-rf
gen p3_excessret = p3-rf
gen p4_excessret = p4-rf
gen p5_excessret = p5-rf
gen p6_excessret = p6-rf
gen p7_excessret = p7-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf
```

// Results from tests

```
// Table of descriptive statistics: mean return, and t-statistics
//mean return is expressed as monthly return , in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = mean t-stat
local row = 1
foreach var of varlist `varlist' {
    qui summarize `var'
    qui reg `var'
        mat A[`row',1] = _b[_cons]*100
        mat A[`row',2] = _b[_cons] / _se[_cons]
    local ++row
}
matlist A, border(rows) rowtitle(Portfolio) format(%8.2f) twidth(10)
```

```

//Table of overall descriptive statistic
//Variables based on excess return
local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret
p7_excessret winners_excessret mom_excessret
local n : word count `varlist'
matrix define A = J(`n',5,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = "FF3" "FF3 t-stat" "FF5" "FF5 t-stat" "Sharpe ratio"
local row = 1
foreach var of varlist `varlist' {
    qui reg `var' mktf smb hml
    mat A[`row',1] = _b[_cons]*100
    mat A[`row',2] = _b[_cons] / _se[_cons]
    qui reg `var' mktf smb hml rmw cma
    mat A[`row',3] = _b[_cons]*100
    mat A[`row',4] = _b[_cons] / _se[_cons]
    qui summarize `var'
    mat A[`row',5] = r(mean)/r(sd)
    local ++row
}
matlist A, border(rows) rowtitle(Return statistic) format(%8.2f)

```

//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 SHRCDD shrccd

rename EXCHCD exchcd

rename SICCD siccd

rename PRC prc

rename RET ret

rename SHROUT shrout

//Translate time variable to month and make it readable

//Keep only dates needed

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 shrccd == 10 | shrccd == 11

//Remove share price below 5

drop if prc < 5

//Sort by industry

//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
```

```
//Name of industry
gen sic2 = "Mining" if sic == 1
replace sic2 = "Food" if sic == 2
replace sic2 = "Apparel" if sic == 3
replace sic2 = "Paper" if sic == 4
replace sic2 = "Chemical" if sic == 5
replace sic2 = "Petroleum" if sic == 6
replace sic2 = "Construction" if sic == 7
replace sic2 = "Prim. Metals" if sic == 8
replace sic2 = "Fab. Metals" if sic == 9
replace sic2 = "Machinery" if sic == 10
replace sic2 = "Electrical Eq." if sic == 11
replace sic2 = "Transport Eq." if sic == 12
replace sic2 = "Manufacturing" if sic == 13
replace sic2 = "Transportation" if sic == 14
replace sic2 = "Other Transport." if sic == 15
replace sic2 = "Air transportation" if sic == 16
replace sic2 = "Utilities" if sic == 17
replace sic2 = "Dept. Stores" if sic == 18
replace sic2 = "Retail" if sic == 19
replace sic2 = "Finance" if sic == 20
replace sic2 = "Technology" if sic == 21
replace sic2 = "Hotels & Social Ser." if sic == 22
replace sic2 = "Health & Membership" if sic == 23
replace sic2 = "Other" if sic == 24
```

//Create formation period and holding period per industry group

//Formation period: Generate past 6 months return, and skip 1 month – J=6
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
sort mofd

//Holding period: Make 6-month returns for holding period

```
gen fret = ret
sort permno mofd
by permno: replace fret = (ret + f.ret + f2.ret + f3.ret + f4.ret + f5.ret)
```

//Average 6-month past return per industry group

```
tostring mofd, generate(time)
bysort mofd sic: egen preret6_mean= mean(preret6_skip) if inrange(time, "503", "514")
bysort mofd sic: egen fret6_mean= mean(fret)
```

//Average monthly return per industry group to use in table

```
bysort sic: egen meanret_industry = mean(ret) if inrange(time, "503", "514")
replace meanret_industry = meanret_industry*100
```

```
duplicates drop mofd sic, force
```

//Rank industry return based on past return: *the command creates the variable port which contains 7 quantiles which contains 3 industry returns per portfolio*

```
bysort mofd: egen rank = rank(preret6_mean) if preret6_mean<.
```

//Generate portfolios based on formation period returns

```
bysort mofd: gen losers= fret6_mean if rank==1 | rank==2 | rank==3  
bysort mofd: gen p2= fret6_mean if rank==4 | rank==5 | rank==6  
bysort mofd: gen p3= fret6_mean if rank==7 | rank==8 | rank==9  
bysort mofd: gen p4= fret6_mean if rank==10 | rank==11 | rank==12  
bysort mofd: gen p5= fret6_mean if rank==13 | rank==14 | rank==15  
bysort mofd: gen p6= fret6_mean if rank==16 | rank==17 | rank==18  
bysort mofd: gen p7= fret6_mean if rank == 19 | rank==20 | rank==21  
bysort mofd: gen winners= fret6_mean if rank==22 | rank==23 | rank==24
```

//Table: Frequency of winners and losers

```
bysort sic: gen lossess=1 if rank == 1 | rank == 2 | rank==3  
bysort sic: egen losses=count(lossess)  
bysort sic: gen winss=1 if rank == 22 | rank == 23 | rank==24  
bysort sic: egen wins=count(winss)  
drop lossess  
drop winss  
tabstat wins losses meanret_industry, by(sic) stat(mean) format(%8.2f)
```

//Calculate mean returns for every month

```
bysort mofd: egen meanlosers=mean(losers)  
bysort mofd: egen meanp2=mean(p2)  
bysort mofd: egen meanp3=mean(p3)  
bysort mofd: egen meanp4=mean(p4)  
bysort mofd: egen meanp5=mean(p5)  
bysort mofd: egen meanp6=mean(p6)  
bysort mofd: egen meanp7=mean(p7)  
bysort mofd: egen meanwinners=mean(winners)
```

// Create momentum return, winners-losers

```
gen mom=meanwinners-meanlosers  
duplicates drop mofd, force
```

//Clean data

```
drop ret  
drop fret  
drop losers  
drop p2  
drop p3  
drop p4  
drop p5  
drop p6  
drop p7  
drop winners
```

```
drop preret6_skip
drop rank
```

//Perform tests

```
//Merge with market returns
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-
months individual mom\famafrench.dta"
drop _merge
```

```
//Average monthly raw return
gen losers=meanlosers/6
gen p2=meanp2/6
gen p3=meanp3/6
gen p4=meanp4/6
gen p5=meanp5/6
gen p6=meanp6/6
gen p7=meanp7/6
gen winners=meanwinners/6
replace mom=mom/6
```

```
//Excess returns
gen losers_excessret = losers-rf
gen p2_excessret = p2-rf
gen p3_excessret = p3-rf
gen p4_excessret = p4-rf
gen p5_excessret = p5-rf
gen p6_excessret = p6-rf
gen p7_excessret = p7-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf
```

// Results from tests

```
// Table of descriptive statistics: mean return, and t-statistics
//mean return is expressed as monthly return , in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = mean t-stat
local row = 1
foreach var of varlist `varlist' {
    qui summarize `var'
    qui reg `var'
        mat A[`row',1] = _b[_cons]*100
        mat A[`row',2] = _b[_cons] / _se[_cons]
    local ++row
}
matlist A, border(rows) rowtitle(Portfolio) format(%8.2f) twidth(10)
```

```

//Table of overall descriptive statistic
//Variables based on excess return
local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret
p7_excessret winners_excessret mom_excessret
local n : word count `varlist'
matrix define A = J(`n',5,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P8-P1
mat colnames A = "FF3" "FF3 t-stat" "FF5" "FF5 t-stat" "Sharpe ratio"
local row = 1
foreach var of varlist `varlist' {
    qui reg `var' mktf smb hml
    mat A[`row',1] = _b[_cons]*100
    mat A[`row',2] = _b[_cons] / _se[_cons]
    qui reg `var' mktf smb hml rmw cma
    mat A[`row',3] = _b[_cons]*100
    mat A[`row',4] = _b[_cons] / _se[_cons]
    qui summarize `var'
    mat A[`row',5] = r(mean)/r(sd)
    local ++row
}
matlist A, border(rows) rowtitle(Return statistic) format(%8.2f)

```