

# CODE – INDIVIDUAL STOCK MOMENTUM

## Table of Contents

//Whole Period J=6 & K=6 .....	2
//Covid-19 – Disruption .....	7
//Covid-19 – Recession Period .....	11
//Covid-19 – Post-Recession Period .....	15
//The Financial Crisis – Recession Period.....	19
//The Financial Crisis – Post-Recession Period .....	23
//The Dot-Com Bubble – Recession Period .....	27
//The Dot-Com Bubble – Post-Recession Period .....	31

## //Whole Period J=6 & K=6

### //Set up the import options and import the data//

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

```
//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
```

```
//Data from January 1965 – March 2022
```

```
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
```

### //Momentum portfolio J=6//

```
//Formation period: Calculate last 6 months return, and skip 1 month – J=6
```

```
sort permno mofd
```

```
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
```

```
//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 p_6 = preret6_skip, by(mofd) nq(10)
```

### //Holding period: Generate future 6 months return, K=6

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

```
//Holding period: Generate returns for future months in each of the 10 portfolios
```

```
bysort mofd: gen losers=fret if p_6==1
bysort mofd: gen p2=fret if p_6==2
bysort mofd: gen p3=fret if p_6==3
bysort mofd: gen p4=fret if p_6==4
bysort mofd: gen p5=fret if p_6==5
bysort mofd: gen p6=fret if p_6==6
bysort mofd: gen p7=fret if p_6==7
bysort mofd: gen p8=fret if p_6==8
bysort mofd: gen p9=fret if p_6==9
bysort mofd: gen winners=fret if p_6==10
```

```
//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
```

```
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 meanp8=mean(p8)
bysort mofd: egen meanp9=mean(p9)
bysort mofd: egen meanwinners=mean(winners)
```

```
//Create momentum return, winners-losers
```

```
gen momentum=meanwinners-meanlosers
```

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

```
drop ret
drop fret
drop losers
drop p2
drop p3
drop p4
drop p5
drop p6
drop p7
drop p8
drop p9
drop winners
drop preret6_skip
drop p_6
```

```
//save file save "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\wholeperiod.dta"
```

### **//Perform tests//**

```
//Make data from Ken French website to match individual stock momentum J=6  
import excel "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\ff5.xlsx", sheet("F-F_Research_Data_5_Factors_2x3") firstrow
```

```
//Set up the import options and import the data//
```

```
//small letters on all variables
```

```
rename MktRF mktrf
```

```
rename SMB smb
```

```
rename HML hml
```

```
rename RF rf
```

```
rename RMW rmw
```

```
rename CMA cma
```

```
//Translate time variable to month and make it readable
```

```
gen mofd = tm(1965m1)+_n-1
```

```
format %tm mofd
```

```
tsset mofd
```

```
//Destring variables
```

```
destring mktrf, replace
```

```
destring smb, replace
```

```
destring hml, replace
```

```
destring rf, replace
```

```
destring rmw, replace
```

```
destring cma, replace
```

```
//Divide variables by 100 to get decimals, to match the other dataset
```

```
replace mktrf = mktrf/100
```

```
replace smb = smb/100
```

```
replace hml = hml/100
```

```
replace rf = rf/100
```

```
replace rmw = rmw/100
```

```
replace cma = cma/100
```

```
//save "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-months individual mom\famafrench.dta"
```

### **//Merge Fama-French data with Individual stock momentum**

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

```
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis oppgave\6-
months individual mom\fama french.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 p8=meanp8/6
gen p9=meanp9/6
gen winners=meanwinners/6
gen mom=momentum/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 p8_excessret = p8-rf
gen p9_excessret = p9-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 (not ), in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 p8 p9 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P10-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) title("Individual Stock Momentum") tindent(14) left(10)
format(%8.2f) twidth(10)
```

```

//Table of overall descriptive statistic
//Variables based on excess return, and t-stat
local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret
p7_excessret p8_excessret p9_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 P9 P10 P10-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) title("Individual Stock Momentum")
tindent(50)format(%8.2f)

```

## //Covid-19 – Disruption

### //Set up the import options and import the data//

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

```
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
```

### //Formation period: Calculate last 6 months return, and skip the current month – J=6

```
sort permno mofd  
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
```

```
//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 p_6 = preret6_skip, by(mofd) nq(10)
```

### //Holding period: Generate future 6 months return, K=6

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

```
//Holding period: Generate returns for future months in each of the 10 portfolios  
bysort mofd: gen losers=fret if p_6==1  
bysort mofd: gen p2=fret if p_6==2  
bysort mofd: gen p3=fret if p_6==3
```

```
bysort mofd: gen p4=fret if p_6==4
bysort mofd: gen p5=fret if p_6==5
bysort mofd: gen p6=fret if p_6==6
bysort mofd: gen p7=fret if p_6==7
bysort mofd: gen p8=fret if p_6==8
bysort mofd: gen p9=fret if p_6==9
bysort mofd: gen winners=fret if p_6==10
```

```
//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
```

```
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 meanp8=mean(p8)
bysort mofd: egen meanp9=mean(p9)
bysort mofd: egen meanwinners=mean(winners)
```

```
// Create momentum return, winners-losers
```

```
gen momentum=meanwinners-meanlosers
```

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

```
//Drop variables
```

```
drop ret
drop fret
drop losers
drop p2
drop p3
drop p4
drop p5
drop p6
drop p7
drop p8
drop p9
drop winners
drop preret6_skip
drop p_6
```

```
//Merge Fama-French data with Individual stock momentum
```

```
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 p8=meanp8/6
gen p9=meanp9/6
gen winners=meanwinners/6
gen mom=momentum/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 p8_excessret = p8-rf
gen p9_excessret = p9-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf

```

### // Results from tests

```

//Variables based on excess return, and t-stat based on regression with excess return and market risk premium

```

```

// Table of descriptive statistics: mean excess return, and t-statistics

```

```

//mean return is expressed as monthly return in percentage form

```

```

local varlist losers p2 p3 p4 p5 p6 p7 p8 p9 winners mom

```

```

local n : word count `varlist'

```

```

matrix define A = J(`n',2,.)

```

```

mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P10-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, and t-stat based on regression with excess return and market risk premium

```

```

local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret

```

```

p7_excessret p8_excessret p9_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 P9 P10 P10-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

### //Set up the import options and import the data//

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

```
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
```

### //Formation period: Calculate last 6 months return, and skip the current month – J=6

```
sort permno mofd  
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
```

```
//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 p_6 = preret6_skip, by(mofd) nq(10)
```

### //Holding period: Generate future 6 months return, K=6

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

```
//Holding period: Generate returns for future months in each of the 10 portfolios
```

```
bysort mofd: gen losers=fret if p_6==1  
bysort mofd: gen p2=fret if p_6==2
```

```
bysort mofd: gen p3=fret if p_6==3
bysort mofd: gen p4=fret if p_6==4
bysort mofd: gen p5=fret if p_6==5
bysort mofd: gen p6=fret if p_6==6
bysort mofd: gen p7=fret if p_6==7
bysort mofd: gen p8=fret if p_6==8
bysort mofd: gen p9=fret if p_6==9
bysort mofd: gen winners=fret if p_6==10
```

```
//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
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 meanp8=mean(p8)
bysort mofd: egen meanp9=mean(p9)
bysort mofd: egen meanwinners=mean(winners)
```

```
//Create momentum return, winners-losers
gen momentum=meanwinners-meanlosers
```

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

```
//Drop variables
drop ret
drop fret
drop losers
drop p2
drop p3
drop p4
drop p5
drop p6
drop p7
drop p8
drop p9
drop winners
drop preret6_skip
drop p_6
```

```
//Merge Fama-French data with Individual stock momentum
```

```
merge m:1 mofd using "C:\Users\Katar\OneDrive\Dokumenter\Master\Master thesis opgave\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 p8=meanp8/6
gen p9=meanp9/6
gen winners=meanwinners/6
gen mom=momentum/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 p8_excessret = p8-rf
gen p9_excessret = p9-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf

```

### **//Results from tests**

```

//Variables based on excess return, and t-stat based on regression with excess return and market risk premium

```

```

// Table of descriptive statistics: mean excess return, and t-statistics

```

```

//mean return is expressed as monthly return in percentage form

```

```

local varlist losers p2 p3 p4 p5 p6 p7 p8 p9 winners mom

```

```

local n : word count `varlist'

```

```

matrix define A = J(`n',2,.)

```

```

mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P10-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, and t-stat based on regression with excess return and market risk premium

```

```

local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret

```

```

p7_excessret p8_excessret p9_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 P9 P10 P10-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

### //Set up the import options and import the data//

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

```
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
```

### //Formation period: Calculate last 6 months return, and skip the current month – J=6

```
sort permno mofd  
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
```

```
//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 p_6 = preret6_skip, by(mofd) nq(10)
```

### //Holding period: Generate future 6 months return, K=6

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

```
//Holding period: Generate returns for future months in each of the 10 portfolios  
bysort mofd: gen losers=fret if p_6==1  
bysort mofd: gen p2=fret if p_6==2  
bysort mofd: gen p3=fret if p_6==3
```

```
bysort mofd: gen p4=fret if p_6==4
bysort mofd: gen p5=fret if p_6==5
bysort mofd: gen p6=fret if p_6==6
bysort mofd: gen p7=fret if p_6==7
bysort mofd: gen p8=fret if p_6==8
bysort mofd: gen p9=fret if p_6==9
bysort mofd: gen winners=fret if p_6==10
```

```
//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
```

```
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 meanp8=mean(p8)
bysort mofd: egen meanp9=mean(p9)
bysort mofd: egen meanwinners=mean(winners)
```

```
//Create momentum return, winners-losers
```

```
gen momentum=meanwinners-meanlosers
```

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

```
//Drop variables
```

```
drop ret
drop fret
drop losers
drop p2
drop p3
drop p4
drop p5
drop p6
drop p7
drop p8
drop p9
drop winners
drop preret6_skip
drop p_6
```

```
//Merge Fama-French data with Individual stock momentum
```

```
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 p8=meanp8/6
gen p9=meanp9/6
gen winners=meanwinners/6
gen mom=momentum/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 p8_excessret = p8-rf
gen p9_excessret = p9-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf

```

### **// Results from tests**

```

//Variables based on excess return, and t-stat based on regression with excess return and market risk premium

```

```

// Table of descriptive statistics: mean excess return, and t-statistics

```

```

//mean return is expressed as monthly return in percentage form

```

```

local varlist losers p2 p3 p4 p5 p6 p7 p8 p9 winners mom

```

```

local n : word count `varlist'

```

```

matrix define A = J(`n',2,.)

```

```

mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P10-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, and t-stat based on regression with excess return and market risk premium

```

```

local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret

```

```

p7_excessret p8_excessret p9_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 P9 P10 P10-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

### //Set up the import options and import the data//

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

```
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
```

### //Formation period: Calculate last 6 months return, and skip the current month – J=6

```
sort permno mofd  
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
```

```
//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 p_6 = preret6_skip, by(mofd) nq(10)
```

### //Holding period: Generate future 6 months return, K=6

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

```
//Holding period: Generate returns for future months in each of the 10 portfolios  
bysort mofd: gen losers=fret if p_6==1  
bysort mofd: gen p2=fret if p_6==2  
bysort mofd: gen p3=fret if p_6==3
```

```
bysort mofd: gen p4=fret if p_6==4
bysort mofd: gen p5=fret if p_6==5
bysort mofd: gen p6=fret if p_6==6
bysort mofd: gen p7=fret if p_6==7
bysort mofd: gen p8=fret if p_6==8
bysort mofd: gen p9=fret if p_6==9
bysort mofd: gen winners=fret if p_6==10
```

```
//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
```

```
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 meanp8=mean(p8)
bysort mofd: egen meanp9=mean(p9)
bysort mofd: egen meanwinners=mean(winners)
```

```
//Create momentum return, winners-losers
```

```
gen momentum=meanwinners-meanlosers
```

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

```
//Drop variables
```

```
drop ret
drop fret
drop losers
drop p2
drop p3
drop p4
drop p5
drop p6
drop p7
drop p8
drop p9
drop winners
drop preret6_skip
drop p_6
```

```
//Merge Fama-French data with Individual stock momentum
```

```
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 p8=meanp8/6
gen p9=meanp9/6
gen winners=meanwinners/6
gen mom=momentum/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 p8_excessret = p8-rf
gen p9_excessret = p9-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf

```

### **// Results from tests**

```

//Variables based on excess return, and t-stat based on regression with excess return and market risk premium

```

```

// Table of descriptive statistics: mean excess return, and t-statistics

```

```

//mean return is expressed as monthly return in percentage form

```

```

local varlist losers p2 p3 p4 p5 p6 p7 p8 p9 winners mom

```

```

local n : word count `varlist'

```

```

matrix define A = J(`n',2,.)

```

```

mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P10-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, and t-stat based on regression with excess return and market risk premium

```

```

local varlist losers_excessret p2_excessret p3_excessret p4_excessret p5_excessret p6_excessret

```

```

p7_excessret p8_excessret p9_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 P9 P10 P10-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

### //Set up the Import Options and import the data//

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

```
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
```

### //Formation period: Calculate last 6 months return, and skip the current month – J=6

```
sort permno mofd  
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
```

```
//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 p_6 = preret6_skip, by(mofd) nq(10)
```

### //Holding period: Generate future months return, K=6

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

```

//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen losers=fret if p_6==1
bysort mofd: gen p2=fret if p_6==2
bysort mofd: gen p3=fret if p_6==3
bysort mofd: gen p4=fret if p_6==4
bysort mofd: gen p5=fret if p_6==5
bysort mofd: gen p6=fret if p_6==6
bysort mofd: gen p7=fret if p_6==7
bysort mofd: gen p8=fret if p_6==8
bysort mofd: gen p9=fret if p_6==9
bysort mofd: gen winners=fret if p_6==10

//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
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 meanp8=mean(p8)
bysort mofd: egen meanp9=mean(p9)
bysort mofd: egen meanwinners=mean(winners)

//Create momentum return, winners-losers
gen momentum=meanwinners-meanlosers

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

//Drop variables
drop ret
drop fret
drop losers
drop p2
drop p3
drop p4
drop p5
drop p6
drop p7
drop p8
drop p9
drop winners
drop preret6_skip
drop p_6

//Merge Fama-French data with Individual stock momentum
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 p8=meanp8/6
gen p9=meanp9/6
gen winners=meanwinners/6
gen mom=momentum/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 p8_excessret = p8-rf
gen p9_excessret = p9-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf

```

### **// Results from tests**

```

//Variables based on excess return, and t-stat based on regression with excess return and market risk
premium
// Table of descriptive statistics: mean excess return, and t-statistics
//mean return is expressed as monthly return in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 p8 p9 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rownames A = P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P10-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, and t-stat based on regression with excess return and market risk premium

local varlist losers\_excessret p2\_excessret p3\_excessret p4\_excessret p5\_excessret p6\_excessret  
p7\_excessret p8\_excessret p9\_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 P9 P10 P10-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

### //Set up the Import Options and import the data//

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

```
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.  
//Recession from March 2001 – November 2001.  
//Run data from September 2000 until April 2002. Include 6 months before and, 5 months after  
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
```

### //Formation period: Calculate last 6 months return, and skip the current month – J=6

```
sort permno mofd  
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
```

```
//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 p_6 = preret6_skip, by(mofd) nq(10)
```

### //Holding period: Generate future 6 months return, K=6

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

```

//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen losers=fret if p_6==1
bysort mofd: gen p2=fret if p_6==2
bysort mofd: gen p3=fret if p_6==3
bysort mofd: gen p4=fret if p_6==4
bysort mofd: gen p5=fret if p_6==5
bysort mofd: gen p6=fret if p_6==6
bysort mofd: gen p7=fret if p_6==7
bysort mofd: gen p8=fret if p_6==8
bysort mofd: gen p9=fret if p_6==9
bysort mofd: gen winners=fret if p_6==10

//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
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 meanp8=mean(p8)
bysort mofd: egen meanp9=mean(p9)
bysort mofd: egen meanwinners=mean(winners)

//Create momentum return, winners-losers
gen momentum=meanwinners-meanlosers

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

//Drop variables
drop ret
drop fret
drop losers
drop p2
drop p3
drop p4
drop p5
drop p6
drop p7
drop p8
drop p9
drop winners
drop preret6_skip
drop p_6

//Merge Fama-French data with Individual stock momentum
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 p8=meanp8/6
gen p9=meanp9/6
gen winners=meanwinners/6
gen mom=momentum/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 p8_excessret = p8-rf
gen p9_excessret = p9-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf

```

### **//Results from tests**

```

//Variables based on excess return, and t-stat based on regression with excess return and market risk premium
// Table of descriptive statistics: mean excess return, and t-statistics
//mean return is expressed as monthly return in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 p8 p9 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rnames A = P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P10-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, and t-stat based on regression with excess return and market risk premium

local varlist losers\_excessret p2\_excessret p3\_excessret p4\_excessret p5\_excessret p6\_excessret  
p7\_excessret p8\_excessret p9\_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 P9 P10 P10-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

### //Set up the Import Options and import the data//

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

```
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.  
//Recession from December 2001 until December 2002.  
//Run data from June 2001 until May 2003. Include 6 months before and, 5 months after  
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
```

### //Formation period: Calculate last 6 months return, and skip the current month – J=6

```
sort permno mofd  
by permno: gen preret6_skip = (1+l2.ret)*(1+l3.ret)*(1+l4.ret)*(1+l5.ret)*(1+l6.ret)
```

```
//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 p_6 = preret6_skip, by(mofd) nq(10)
```

### //Holding period: Generate future 6 months return, K=6

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

```

//Holding period: Generate returns for future months in each of the 10 portfolios
bysort mofd: gen losers=fret if p_6==1
bysort mofd: gen p2=fret if p_6==2
bysort mofd: gen p3=fret if p_6==3
bysort mofd: gen p4=fret if p_6==4
bysort mofd: gen p5=fret if p_6==5
bysort mofd: gen p6=fret if p_6==6
bysort mofd: gen p7=fret if p_6==7
bysort mofd: gen p8=fret if p_6==8
bysort mofd: gen p9=fret if p_6==9
bysort mofd: gen winners=fret if p_6==10

//Summarize and sort mean returns on each of portfolios; get mean winners and mean losers
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 meanp8=mean(p8)
bysort mofd: egen meanp9=mean(p9)
bysort mofd: egen meanwinners=mean(winners)

//Create momentum return, winners-losers
gen momentum=meanwinners-meanlosers

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

//Drop variables
drop ret
drop fret
drop losers
drop p2
drop p3
drop p4
drop p5
drop p6
drop p7
drop p8
drop p9
drop winners
drop preret6_skip
drop p_6

//Merge Fama-French data with Individual stock momentum
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 p8=meanp8/6
gen p9=meanp9/6
gen winners=meanwinners/6
gen mom=momentum/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 p8_excessret = p8-rf
gen p9_excessret = p9-rf
gen winners_excessret = winners-rf
gen mom_excessret = mom-rf

```

### // Results from tests

```

//Variables based on excess return, and t-stat based on regression with excess return and market risk premium
// Table of descriptive statistics: mean excess return, and t-statistics
//mean return is expressed as monthly return in percentage form
local varlist losers p2 p3 p4 p5 p6 p7 p8 p9 winners mom
local n : word count `varlist'
matrix define A = J(`n',2,.)
mat rnames A = P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P10-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, and t-stat based on regression with excess return and market risk premium

local varlist losers\_excessret p2\_excessret p3\_excessret p4\_excessret p5\_excessret p6\_excessret  
p7\_excessret p8\_excessret p9\_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 P9 P10 P10-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)