Quantitative Methods

Project and Research replication *(1)*

Ludovic Vigneron

University of Lille (IAE)

September 6th, 2022

1 / 57

Afternoon program

1. Outlines of the student project
2. Jegadeesh & Titman (1993) Momentum Portfolio (*replication 1*)

Outlines of the student project

Find a question

Get the data on WRDS database

Try to elaborate statistics that could answer the question

Some advices

- Make it simple
- Plot your data
- Focus on descriptive statistics
- Try to use statistical tests and (naive) regression analysis
- Do not hesitate to e-mail technical questions (I will try to find answers)

The elements that you have to send me

- A at least a one page pdf that explains your question and the data you need to answer it (reference with previous academic works will be appreciated but is not mandatory)
- Somes captures of your request on WRDS data base
- .csv file of your data base
- R script of your traitments and analysis
- A at least a two pages pdf presenting your analysis (tables, graphs) and explaining the answer to your question.

The deadline

October the 30th

Q & A

Let's ask your question about the report

Jegadeesh & Titman (1993) Momentum Portfolio

(replication 1)

A little of theory

The program is designed to create momentum portfolios based on past 3 to 12 month returns

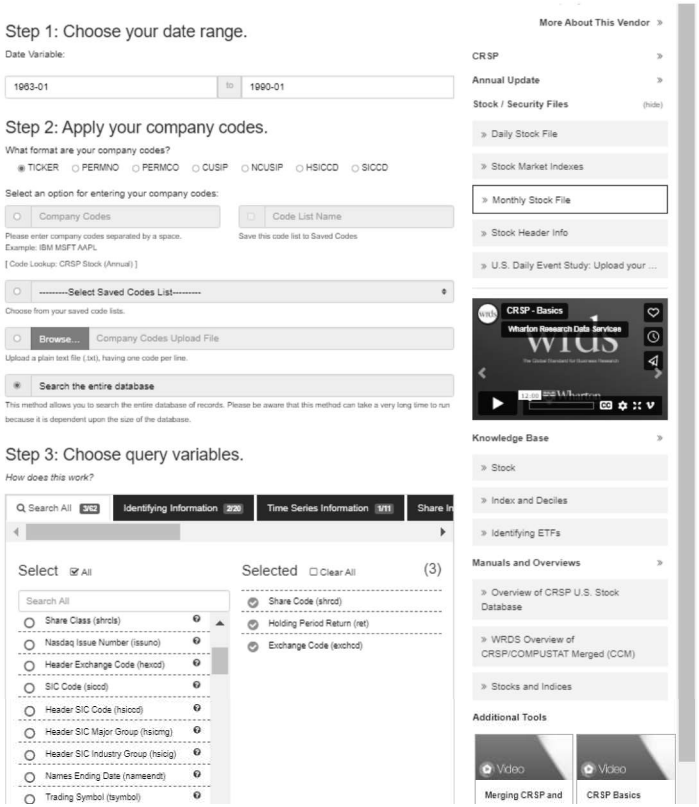
Install the package

```
library(tidyverse)
library(scales)
library(lubridate)
library(zoo)
```

Get the data

Let take all CRPS monthly data between '01/01/1963' and '12/31/1989' (permno, date, ret, shrcd, exchcd).

Get the data


Step 1: Choose your date range.
Date Variable:
1963-01
to
1990-01
Step 2: Apply your company codes.
What format are your company codes?
TICKER PERMNO PERMCO CUSIP NCUSIP HSICCD SICCD
Select an option for entering your company codes:
Company Codes
Code List Name
Please enter company codes separated by a space.
Example: IBM MSFT AAPL
[Code Lookup: CRSP Stock (Annual)]
Save this code list to Saved Codes
---------Select Saved Codes List---------
Choose from your saved code lists.
Browse...
Company Codes Upload File
Upload a plain text file (.txt), having one code per line.
Search the entire database
This method allows you to search the entire database of records. Please be aware that this method can take a very long time to run because it is dependent upon the size of the database.
Step 3: Choose query variables.
How does this work?
Search All 5/62
Identifying Information 2/20
Time Series Information 1/11
Select All
Search All
Share Class (shrcls)
Nasdaq Issue Number (issuno)
Header Exchange Code (hexcd)
SIC Code (sicd)
Header SIC Code (hsicd)
Header SIC Major Group (hsicmg)
Header SIC Industry Group (hsicig)
Names Ending Date (nameendt)
Trading Symbol (tsymbol)
Selected Clear All (3)
Share Code (shrcd)
Holding Period Return (ret)
Exchange Code (exchcd)
More About This Vendor
CRSP
Annual Update
Stock / Security Files (hide)
Daily Stock File
Stock Market Indexes
Monthly Stock File
Stock Header Info
U.S. Daily Event Study: Upload your ...
CRSP - Basics
Wharton Research Data Services
Knowledge Base
Stock
Index and Deciles
Identifying ETFs
Manuals and Overviews
Overview of CRSP U.S. Stock Database
WRDS Overview of CRSP/COMPUSTAT Merged (CCM)
Stocks and Indices
Additional Tools
Video
Merging CRSP and
Video
CRSP Basics

Import the data in R

```
crps_m <- read_csv("c7fl9v3ksaszhhr2.csv")
```

```
## Rows: 1500991 Columns: 5
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): RET
## dbl (4): PERMNO, date, SHRCD, EXCHCD
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Limit the data to the usefull ones

We only consider stocks listed on the NYSE or the AMEX (exchcd between -2 and 2).

We also only consider ordinary stocks (shrcd 10 or 11).

```
crps_m <- crps_m %>%
  filter(EXCHCD%in%c(-2,-1,0,1,2)&SHRCD%in%c(10,11))
```

Limit the data to the usefull ones

We have been to wide in our extraction widow. Let's reduce the data base.

```
crps_m <- crps_m %>% mutate(date=ymd(date)) %>%
  filter(date<"1990-01-31")
```

We also have no more use for the variables SHRCD and EXCHCD. So, we can drop them.

```
crps_m <- crps_m %>% select(-SHRCD,-EXCHCD)
```

Format the data

In order to make them easier to manipulate, we transform some variables (PERMNO and RET) in numeric ones.

```
crps_m <- crps_m %>% mutate(PERMNO=as.integer(PERMNO),
                            RET=na.fill(as.numeric(RET),fill=0))
```

```
## Warning in na.fill(as.numeric(RET), fill = 0): NAs introduits lors de la
## conversion automatique
```

Format the data

```
head(crps_m,5)
```

```
## # A tibble: 5 × 3
##   PERMNO date           RET
##    <int> <date>       <dbl>
## 1  10006 1963-01-31  0.0470
## 2  10006 1963-02-28  0.0387
## 3  10006 1963-03-29 -0.00901
## 4  10006 1963-04-30  0.0848
## 5  10006 1963-05-31  0.0916
```

Calculate the formation period returns

Once done, we can establish the log returns (they are easer to cumulate, we can just add them).

```
crps_m <- crps_m %>% mutate(log_RET=log(1+RET))
```

Calculate the formation period returns

```
tail(crps_m,5)
```

```
## # A tibble: 5 × 4
##   PERMNO date          RET log_RET
##    <int> <date>      <dbl>   <dbl>
## 1  93201 1989-08-31 0.188   0.172
## 2  93201 1989-09-29 0.0351  0.0345
## 3  93201 1989-10-31 0.0508  0.0496
## 4  93201 1989-11-30 0.210   0.190
## 5  93201 1989-12-29 0.04    0.0392
```

Calculate the formation period returns

Let's take a **holding period of 6 month** and compute the corresponding rolling cumulative returns for each stocks.

```
j=6
```

We use the **rollsum()** function from the **zoo package** to compute the rolling sum of returns.

```
umd <- crps_m %>% arrange(PERMNO,date) %>%
  group_by(PERMNO) %>%
  mutate(sumlogret=rollsum(log_RET, j, align = "right", fill = NA)) %>%
  ungroup()
```

Calculate the formation period returns

```
tail(umd,5)
```

```
## # A tibble: 5 × 5
##   PERMNO date          RET log_RET sumlogret
##    <int> <date>      <dbl>   <dbl>     <dbl>
## 1  93201 1989-08-31 0.188   0.172     0.305
## 2  93201 1989-09-29 0.0351  0.0345    0.271
## 3  93201 1989-10-31 0.0508  0.0496   -0.0317
## 4  93201 1989-11-30 0.210   0.190     0.190
## 5  93201 1989-12-29 0.04    0.0392    0.368
```

Calculate the formation period returns

Our sum of log returns is done, so we can go back to simple returns.

```
umd <- umd %>% mutate(cumret=exp(sumlogret)-1)
```

Calculate the formation period returns

We don't need anymore the variable RET and log_RED. Let's drop them.

```
umd <- umd %>% select(-RET,-log_RET)
```

Calculate the formation period returns

```
tail(umd,5)
```

```
## # A tibble: 5 × 4
##   PERMNO date       sumlogret  cumret
##    <int> <date>         <dbl>   <dbl>
## 1  93201 1989-08-31    0.305   0.357
## 2  93201 1989-09-29    0.271   0.311
## 3  93201 1989-10-31   -0.0317 -0.0312
## 4  93201 1989-11-30    0.190   0.210
## 5  93201 1989-12-29    0.368   0.444
```

Form portfolios

We need to compose portflio using stocks ranked according their returns.

We chose to compose ten portfolios. They are named following the rank of the stocks composing them.

The portfolio with rank 1 stocks is the worst ones (the loser portfolio).

The portfolio with rank 10 stocks is the worst ones (the winner portfolio).

Form portfolios

Let's do a little cleaning before.

```
umd<-umd %>% drop_na()
```

Let's create the variable rk*p* (ranked portfolio) using the **cut()** function and the **quantile()** one.

```
umd <- umd %>% group_by(date) %>%
  mutate(rk_p_=cut(sumlogret,
                   breaks=c(quantile(sumlogret,
                                     probs = seq(0, 1, by = 1/10),na.rm = T)),
                   include.lowest=TRUE,
                   labels=c('1','2','3','4','5','6','7','8','9','10')))
```

Form portfolios

```
tail(umd,5)
```

```
## # A tibble: 5 × 5
## # Groups:   date [5]
##   PERMNO date       sumlogret  cumret rk_p_
##    <int> <date>         <dbl>   <dbl> <fct>
## 1  93201 1989-08-31    0.305   0.357  9
## 2  93201 1989-09-29    0.271   0.311  9
## 3  93201 1989-10-31   -0.0317 -0.0312 4
## 4  93201 1989-11-30    0.190   0.210  9
## 5  93201 1989-12-29    0.368   0.444  10
```

Form portfolios

Wat are the average return of those portfolios ?

```
umd %>% group_by(rk_p_) %>%
  summarise(moy_re_=mean(cumret))
```

```
## # A tibble: 10 × 2
##    rk_p_ moy_re_
##    <fct>   <dbl>
##  1 1     -0.308
##  2 2     -0.161
##  3 3     -0.0880
##  4 4     -0.0325
##  5 5      0.0186
##  6 6      0.0707
##  7 7      0.128
##  8 8      0.199
##  9 9      0.305
## 10 10     0.643
```

Calculate the holding period return

Now we have our portfolios, so we need to compute their returns for the holding period (6 month).

So, we need to mark this holding period for each one of them. Let's build a date frame (umd_r) including the date of the begining of the holding period (hdate1) and the date of its end (hdate2).

For these tasks, we use the functions **ceiling_date()**, **floor_date()** and **month()** from the lubridate package.

Calculate the holding period return

```
umd_r<-umd %>% mutate(form_date=date,
        medate=ceiling_date(date,"month"),
        hdate1=floor_date(medate,unit='day'),
        hdate2=ceiling_date(medate,unit='month')+months(5)-1) %>%
  ungroup() %>%
  select(PERMNO,form_date,medate,rk_p_,hdate1,hdate2)
```

Calculate the holding period return

```
tail(umd_r,5)
```

```
## # A tibble: 5 × 6
##   PERMNO form_date  medate     rk_p_ hdate1     hdate2
##    <int> <date>     <date>     <fct> <date>     <date>
## 1  93201 1989-08-31 1989-09-01 9     1989-09-01 1990-02-28
## 2  93201 1989-09-29 1989-10-01 9     1989-10-01 1990-03-31
## 3  93201 1989-10-31 1989-11-01 4     1989-11-01 1990-04-30
## 4  93201 1989-11-30 1989-12-01 9     1989-12-01 1990-05-31
## 5  93201 1989-12-29 1990-01-01 10    1990-01-01 1990-06-30
```

Calculate the holding period return

Let's merge this data frame with the returns data.

```
port<-inner_join(umd_r,crps_m,by='PERMNO')
```

Calculate the holding period return

Limit the data to the ones we need. Make sure that *date* is higher than *hdate1* and smaller than *hdate2*.

Drop log returns (we don't need them).

```
port<-port %>% filter(hdate1<=date&date<=hdate2) %>%
  select(-log_RET)
```

Calculate the holding period return

Let see what we have for one stock and given portfolio formation date.

```
port %>% filter(PERMNO==93172&form_date=='1986-03-31')
```

```
## # A tibble: 6 × 8
##   PERMNO form_date  medate     rk_p_ hdate1     hdate2     date           RET
##    <int> <date>     <date>     <fct> <date>     <date>     <date>       <dbl>
## 1  93172 1986-03-31 1986-04-01 2     1986-04-01 1986-09-30 1986-04-30 -0.162
## 2  93172 1986-03-31 1986-04-01 2     1986-04-01 1986-09-30 1986-05-30 -0.0323
## 3  93172 1986-03-31 1986-04-01 2     1986-04-01 1986-09-30 1986-06-30 -0.267
## 4  93172 1986-03-31 1986-04-01 2     1986-04-01 1986-09-30 1986-07-31  0
## 5  93172 1986-03-31 1986-04-01 2     1986-04-01 1986-09-30 1986-08-29 -0.773
## 6  93172 1986-03-31 1986-04-01 2     1986-04-01 1986-09-30 1986-09-30 -0.4
```

Calculate Portfolio return

From our combined data base, we can compute the equaly weighted returns form our different portfolio each date.

```
umd_port<- port %>% group_by(date,rk_p_,form_date) %>%
  summarise(ret_port=mean(RET))
```

```
## `summarise()` has grouped output by 'date', 'rk_p_'. You can override using the
## `.groups` argument.
```

Calculate Portfolio return

```
head(umd_port,10)
```

```
## # A tibble: 10 × 4
## # Groups:   date, rk_p_ [10]
##    date       rk_p_ form_date   ret_port
##    <date>     <fct> <date>         <dbl>
##  1 1963-07-31 1     1963-06-28  0.00282
##  2 1963-07-31 2     1963-06-28 -0.00792
##  3 1963-07-31 3     1963-06-28 -0.00981
##  4 1963-07-31 4     1963-06-28 -0.00502
##  5 1963-07-31 5     1963-06-28 -0.00277
##  6 1963-07-31 6     1963-06-28 -0.00143
##  7 1963-07-31 7     1963-06-28 -0.0140
##  8 1963-07-31 8     1963-06-28 -0.0162
##  9 1963-07-31 9     1963-06-28 -0.00509
## 10 1963-07-31 10    1963-06-28 -0.000961
```

Calculate the holding period return

Skip the first 2 years of the sample as Jegadeesh & Titman have done. Let's create variables indicating years using the function **year()** from lubridate.

```
umd_port_<-umd_port %>% mutate(start_yr=year(date)+2,
                               yr=year(date))
```

Calculate the holding period return

Before skiping

```
min(year(umd_port$date))
```

```
## [1] 1963
```

```
min(year(umd_port$date))+2
```

```
## [1] 1965
```

Calculate the holding period return

Let's do the cut and drop the variables that are no more usefull.

```
umd_port_1<-umd_port_ %>%
  filter(yr>=1965) %>%
  select(-start_yr,-yr)
```

Calculate the holding period return

```
head(umd_port_1,6)
```

```
## # A tibble: 6 × 4
## # Groups:   date, rk_p_ [1]
##   date       rk_p_ form_date  ret_port
##   <date>     <fct> <date>        <dbl>
## 1 1965-01-29 1     1964-07-31    0.125
## 2 1965-01-29 1     1964-08-31    0.130
## 3 1965-01-29 1     1964-09-30    0.114
## 4 1965-01-29 1     1964-10-30    0.125
## 5 1965-01-29 1     1964-11-30    0.123
## 6 1965-01-29 1     1964-12-31    0.134
```

Calculate the holding period return

We can now compute the expected returns (and the related standard deviation) for the reduce sample on each date and each portfolio.

```
ewretdf<-umd_port_1 %>% group_by(date,rk_p_) %>%
  summarise(ewret=mean(ret_port),ewstd=sd(ret_port)) %>%
  ungroup() %>%
  arrange(rk_p_,date)
```

```
## `summarise()` has grouped output by 'date'. You can override using the
## `.groups` argument.
```

Calculate the holding period return

```
head(ewretdf,5)
```

```
## # A tibble: 5 × 4
##   date       rk_p_   ewret   ewstd
##   <date>     <fct>   <dbl>   <dbl>
## 1 1965-01-29 1      0.125  0.00660
## 2 1965-02-26 1      0.0234 0.00583
## 3 1965-03-31 1      0.0297 0.00475
## 4 1965-04-30 1      0.0306 0.00817
## 5 1965-05-28 1     -0.0181 0.00232
```

Calculate the holding period return

Now, we can compute the average return of our equaly weighted portfolio (and their standard deviation)

```
ewretdf_<- ewretdf %>% group_by(rk_p_) %>%
  summarise(count=n(),moy=mean(ewret),std=sd(ewret))
ewretdf_
```

```
## # A tibble: 10 × 4
##    rk_p_ count     moy    std
##    <fct> <int>   <dbl>  <dbl>
##  1 1       300 0.00821 0.0852
##  2 2       300 0.0112  0.0688
##  3 3       300 0.0128  0.0625
##  4 4       300 0.0122  0.0589
##  5 5       300 0.0126  0.0569
##  6 6       300 0.0134  0.0558
##  7 7       300 0.0136  0.0555
##  8 8       300 0.0144  0.0574
##  9 9       300 0.0151  0.0600
## 10 10      300 0.0166  0.0683
```

Calculate the long short portfolio returns

We can now create the long short portfolio (buy the losers and sell the winners simultaneously).

Calculate the long short portfolio returns

First, we have to transpose our data frame using **pivot_wider()** and rename our portfolio 1 loser and our portfolio 10 winner.

```
ewret_t<-ewretdf %>% select(-ewstd) %>%
  pivot_wider(names_from = rk_p_,names_prefix = "port_",
              values_from = ewret) %>%
  rename(loser=port_1,winner=port_10) %>%
  mutate(long_short=winner-loser)
```

Calculate the long short portfolio returns

```
head(ewret_t)
```

```
## # A tibble: 6 × 12
##   date         loser  port_2   port_3   port_4   port_5   port_6   port_7
##   <date>       <dbl>   <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 1965-01-29  0.125   0.0869  0.0711   0.0660   0.0633   0.0588   0.0578
## 2 1965-02-26  0.0234  0.0359  0.0394   0.0295   0.0311   0.0295   0.0311
## 3 1965-03-31  0.0297  0.0130  0.0120   0.0158   0.0112   0.00390  0.00740
## 4 1965-04-30  0.0306  0.0428  0.0412   0.0333   0.0377   0.0419   0.0399
## 5 1965-05-28 -0.0181 -0.0127 -0.00880 -0.00872 -0.00782 -0.00713 -0.00411
## 6 1965-06-30 -0.0968 -0.0738 -0.0707  -0.0661  -0.0701  -0.0774  -0.0774
## # … with 4 more variables: port_8 <dbl>, port_9 <dbl>, winner <dbl>,
## #   long_short <dbl>
```

Calculate the long short portfolio returns

Do the returns of the momentum strategy (the long-short) statistically different from 0? What about the winner and the loser portfolio?

```
moy<-c(winner=mean(ewret_t$winner),
       loser=mean(ewret_t$loser),
       long_short=mean(ewret_t$long_short))
moy
```

```
##      winner       loser  long_short
## 0.016563827 0.008212044 0.008351783
```

Calculate the long short portfolio returns

```
t<-rbind(t.test(ewret_t$winner)$stat,
t.test(ewret_t$loser)$stat,
t.test(ewret_t$long_short)$stat)
p<-round(rbind(t.test(ewret_t$winner)$p.value,
t.test(ewret_t$loser)$p.value,
t.test(ewret_t$long_short)$p.value),digits=3)
data.frame(moy,t,p)
```

```
##                    moy        t     p
## winner     0.016563827 4.198367 0.000
## loser      0.008212044 1.670019 0.096
## long_short 0.008351783 2.769061 0.006
```

Calculate the long short portfolio returns

With our portfolio/strategy returns computed for each date, we can obtain the cumulated return for each date with the function **cumprod()**.

```
ewret_t<-ewret_t %>% mutate(
  cumret_winners=cumprod(1+winner)-1,
  cumret_losers=cumprod(1+loser)-1,
  cumret_long_short=cumprod(1+long_short)-1)
```

Visualisation

Let' do some graphs to have a better idea of our portfolios' performance.

Cumulative returns of the long short portfolio strategy

```
ggplot(data=ewret_t,aes(x=date,y=cumret_long_short))+
  geom_line()+
  geom_text(aes(x=as.Date("1988-01-01"),y=5.5,label='long short'))+
  labs(y='Cumulated returns')+
  scale_y_continuous(breaks = seq(0,7,1),
                     labels = label_percent())+
  scale_x_date(date_breaks = "2 year", date_labels = "%Y")+
  coord_cartesian(expand=FALSE,xlim=c(as.Date("1965-01-01"),
                                      as.Date("1989-12-31")))+
  theme_minimal()+
  theme(axis.title.x = element_blank())
```

Cumulative returns of the winners and losers portfolio strategy

Cumulative returns of the winners and losers portfolio strategy

```
ggplot(data=ewret_t,aes(x=date))+
  geom_line(aes(y=cumret_winners),color='blue')+
  geom_line(aes(y=cumret_losers),color='red',linetype=2)+
  geom_text(aes(x=as.Date("1988-01-01"),y=70,label='Winners'),
         color='blue')+
  geom_text(aes(x=as.Date("1988-01-01"),y=10,label='Losers'),
         color='red')+
  labs(y='Cumulated returns')+
  scale_y_continuous(breaks = seq(0,70,10),
                     labels = label_percent())+
  scale_x_date(date_breaks = "2 year", date_labels = "%Y")+
  coord_cartesian(expand=FALSE,xlim=c(as.Date("1965-01-01"),
                                      as.Date("1989-12-31")))+
  theme_minimal()+
  theme(axis.title.x = element_blank())
```

Cumulative returns of the winners and losers portfolio strategy