

Portfolios based on crowding

```
setwd("C:\\Users\\user\\Desktop")
data<-read.csv("stockdata1.csv")
data[,6]<-data$法人持股./data$週轉率._月 #新增新變數actratio
colnames(data)[6]<-"Actratio"
data$年月<-as.Date(data$年月)
library(lubridate)

## Warning: 套件 'lubridate' 是用 R 版本 4.1.3 來建造的

##
## 載入套件：'lubridate'

## 下列物件被遮斷自 'package:base':
##
##      date, intersect, setdiff, union

library(tidyverse)

## Warning: 套件 'tidyverse' 是用 R 版本 4.1.3 來建造的

## Warning: 套件 'ggplot2' 是用 R 版本 4.1.3 來建造的

## Warning: 套件 'tibble' 是用 R 版本 4.1.3 來建造的

## Warning: 套件 'tidyr' 是用 R 版本 4.1.3 來建造的

## Warning: 套件 'readr' 是用 R 版本 4.1.3 來建造的

## Warning: 套件 'purrr' 是用 R 版本 4.1.3 來建造的

## Warning: 套件 'dplyr' 是用 R 版本 4.1.3 來建造的

## Warning: 套件 'stringr' 是用 R 版本 4.1.3 來建造的

## Warning: 套件 'forcats' 是用 R 版本 4.1.3 來建造的

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.0      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.1      v tibble    3.2.1
## v purrr      1.0.1      v tidyr     1.3.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
NumberOfStock<- data%>%group_by(年月)%>%summarise(n=n())
```

```
NumberOfStock      #檢測是否所有股票的日期數量皆相同
```

```
## # A tibble: 40 x 2
```

```
##   年月      n
```

```
##   <date>  <int>
```

```
## 1 2012-03-30    789
```

```
## 2 2012-06-29    789
```

```
## 3 2012-09-28    789
```

```
## 4 2012-12-28    789
```

```
## 5 2013-03-29    789
```

```
## 6 2013-06-28    789
```

```
## 7 2013-09-30    789
```

```
## 8 2013-12-31    789
```

```
## 9 2014-03-31    789
```

```
## 10 2014-06-30    789
```

```
## # ... with 30 more rows
```

```
weightReturn<- matrix(nrow=40,ncol=789) #新增一個加權報酬的矩陣
```

```
return<- matrix(nrow=40,ncol=789)      #原始報酬的矩陣
```

```
i<-1
```

```
j<-789
```

```
k<-1
```

```
#總共有 789 支股票 4 季*10 年=40 期
```

```
while(k<41){
```

```
  return[k,]<-data[i:j,5] #將報酬矩陣計算為 789 支股票*40 期
```

```
  i=i+789
```

```
  j=j+789
```

```
  k=k+1
```

```
}
```

```
return= return/100
```

```
weightReturn[1,]<-return[1,] #加權報酬第一期為原始報酬
```

```
x<-2
```

```
while(x<41){
```

```
  for(j in c(1:789)){
```

```
    weightReturn[x,j]<- (1+return[(x-1),j])*return[x,j] #將加權報酬定義為  $W_{i,t} = (1+R_{i,t-1}) * R_{i,t}$ 
```

```
  }
```

```
  x=x+1
```

```
}
```

```
actratio<- matrix(nrow=40,ncol=789) #Actratio 的矩陣
```

```
i<-1
```

```
j<-789
```

```
k<-1
```

```

while(k<41){
  actratio[k,]<-data[i:j,6]      #將資料移入 Actratio 的矩陣 總共有789 支股票
  i=i+789
  j=j+789
  k=k+1
}
exhibit<-data.frame(cbind(weightReturn[40,],actratio[40,])) #抽出 2021/1
2/30 的加權報酬以及擁擠度
colnames(exhibit)<-c("adjustReturn","Actratio")
library(dplyr)
exhibit<-exhibit[order(exhibit$Actratio),] #將資料依據擁擠度大小排序
i<-79
j<-2
result<-matrix(nrow = 10,ncol = 3) #result 用以儲存10 組(依據擁擠度大小分
組)的平均報酬
                                     #報酬的標準差 以及 擁擠度的平均
group1<-exhibit[1:78,]              #因總共有789 支股票 令第一組為78 支 其
餘為79 支
result[1,1]<-mean(group1[,2])
result[1,2]<-mean(group1[,1])
result[1,3]<-sd(group1[,1])
while (i<790) {
  result[j,2]<-mean(exhibit[i:(i+78),1]) #將報酬平均放入 result
  result[j,1]<-mean(exhibit[i:(i+78),2]) #將擁擠度平均丟入
  result[j,3]<-sd(exhibit[i:(i+78),1])   #將報酬標準差丟入
  i=i+79
  j=j+1
}
table<- t(result)                   #將表格轉置變為橫向
colnames(table)<-c(1:10)
rownames(table)<-c("Actratio","RET(Mean in%","RET(STD dev.)") #建立 ta
ble
table<-round(table,2)
knitr::kable(as.data.frame(table))

```

	1	2	3	4	5	6	7	8	9	10
Actratio	0.19	0.70	1.39	2.82	5.07	9.25	14.47	23.85	44.41	162.23
RET(Mean in%)	0.14	0.09	0.06	0.06	0.04	0.03	0.04	0.02	0.02	0.02
RET(STD dev.)	0.17	0.11	0.11	0.14	0.06	0.06	0.05	0.03	0.03	0.03