

...

Mô hình Định giá Tài sản Tài chính 1

Nhóm 6:

1. Nghiêm Gia Phương
2. Trần Đức Hiền
3. Lê Tuấn Minh
4. Nguyễn Ngọc Tùng
5. Sầm Thị Mai Chi

Đề bài:

- Thu thập dữ liệu về giá đóng cửa các phiên giao dịch của 3 nhóm cổ phiếu theo ngành trên sàn giao dịch chứng khoán TP HCM cùng với các dữ liệu cần thiết trong tính toán (lượng giao dịch,...). Thực hiện các yêu cầu sau:
 1. Tính các chuỗi lợi suất tương ứng.
 2. Tính chỉ số của từng ngành theo 3 phương pháp tương ứng: price-weighted average, market-value-weighted index, equally weighted indexes.
 3. Trình bày về cách tính chỉ số của một số thị trường trong nước và thế giới.



” Lấy dữ liệu 3 ngành:



Bất động sản



Công nghiệp



Tài chính

• Với mỗi ngành lấy dữ liệu giá đóng cửa của 15 mã cổ phiếu trong khoảng thời gian từ 23/10/2023 – 23/09/2024 229 quan sát

• Nguồn dữ liệu: vietstock.com

Tính Chuỗi lợi suất và vẽ đồ thị

Công thức tính chuỗi lợi suất:

Lợi suất tài sản trong khoảng thời gian $(t - 1, t)$: Công thức số học: $r_t = \frac{S_t - S_{t-1} + D_t}{S_{t-1}}$

Nếu $|r_t|$ khá nhỏ: $r_t \approx \ln\left(\frac{S_t}{S_{t-1}}\right)$

Chứng minh: Từ công thức số học ta có: $S_t = S_{t-1}(1 + r_t)$

suy ra: $\frac{S_t}{S_{t-1}} = 1 + r_t \rightarrow \ln\left(\frac{S_t}{S_{t-1}}\right) = \ln(1 + r_t)$

Khi $r_t \rightarrow 0$, áp dụng tiêu chuẩn L' Hopital ta có: $\lim_{r_t \rightarrow 0} \frac{\ln(1+r_t)}{r_t} = \lim_{r_t \rightarrow 0} \frac{\frac{1}{1+r_t}}{1} = 1$

”

Sử dụng 2 file data gồm:

Market_cap

Closing_price

Sử dụng Rstudio:



Bước 1: Đọc

Load các file data

15 mã đầu: Ngành Bất Động Sản
(Real Estate)

15 mã tiếp theo: Ngành Công
Nghiệp (Industry)

15 mã cuối: Ngành Tài Chính
(Finance)

A tibble: 5 × 46

Date <S3: POSIXct>	VHM <dbl>	VIC <dbl>	BCM <dbl>	VRE <dbl>	KDH <dbl>
2023-10-23	44.5	43.40	63.6	26.00	30.15
2023-10-24	44.6	43.45	63.4	26.60	30.60
2023-10-25	44.9	44.70	62.6	26.45	30.25
2023-10-26	41.8	41.60	59.6	24.60	29.80
2023-10-27	39.2	41.60	59.0	22.90	29.60

5 rows | 1-10 of 46 columns

closing_price

A tibble: 5 × 46

Ngày <S3: POSIXct>	VHM <dbl>	VIC <dbl>	BCM <dbl>	VRE <dbl>	KDH <dbl>
2023-10-23	193769	165525	65826	59080	24099
2023-10-24	194205	165716	65619	60444	24459
2023-10-25	195511	170483	64791	60103	24179
2023-10-26	182013	158660	61686	55899	23819
2023-10-27	170691	158660	61065	52036	23660

5 rows | 1-10 of 46 columns

market_cap

#Tính chuỗi lợi suất của từng cổ phiếu



Tỷ suất lợi nhuận logarit từ dòng 2 đến dòng 229 cho tất cả các cột từ cột 2 đến cột 46 trong bảng closing_price được tính bằng câu lệnh:

Description: df [228 x 45]

VHM <dbl>	VIC <dbl>	BCM <dbl>	VRE <dbl>	KDH <dbl>
0.002244670	0.001151411	-0.003149609	0.022814678	0.014815086
0.006703936	0.028362650	-0.012698583	-0.005655057	-0.011503824
-0.071541455	-0.071873334	-0.049109704	-0.072509715	-0.014987791
-0.064219593	0.000000000	-0.010118130	-0.071609532	-0.006734032
0.000000000	-0.002406740	0.008438869	0.008695707	-0.034367644
-0.005115101	-0.024391453	-0.015241616	-0.039740329	0.000000000
-0.014202950	-0.002472189	-0.010291686	0.022272636	0.000000000
0.038269976	0.025658097	0.013698844	0.023940208	0.047790664
0.013673302	0.003612286	-0.005115101	0.048278000	-0.027028672
0.012270093	0.011947573	0.015267472	-0.016529302	0.018660423

1-10 of 228 rows | 1-5 of 45 columns

Previous 1 2 3 4 5 6 ... 23 Next

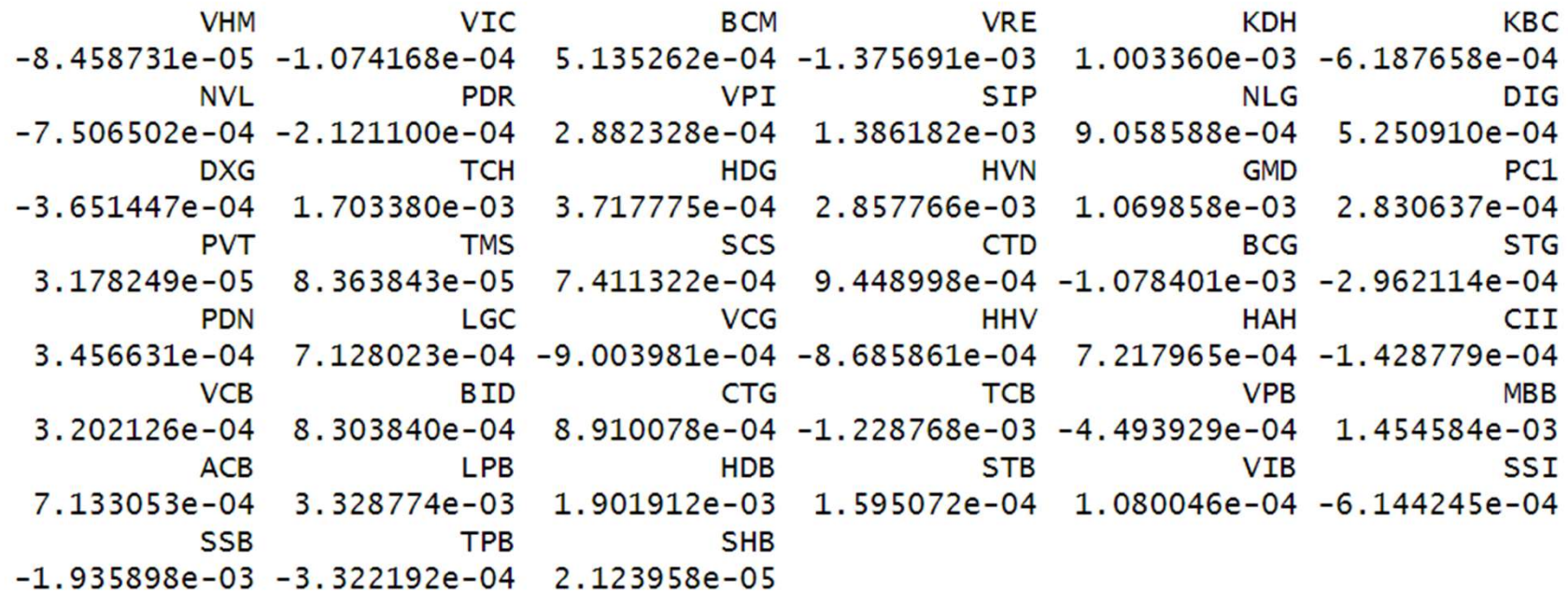
#Tính chuỗi lợi suất của từng cổ phiếu

```
rate_of_return = log(closing_price[2:229, 2:46]/closing_price[1:228, 2:46])
rate_of_return
```

Xác định lợi nhuận kỳ vọng (expected return)

#Tính lợi suất kì vọng

```
expected_r = colMeans(rate_of_return)
```



```
VHM      VIC      BCM      VRE      KDH      KBC
-8.458731e-05 -1.074168e-04  5.135262e-04 -1.375691e-03  1.003360e-03 -6.187658e-04
 NVL      PDR      VPI      SIP      NLG      DIG
-7.506502e-04 -2.121100e-04  2.882328e-04  1.386182e-03  9.058588e-04  5.250910e-04
 DXG      TCH      HDG      HVN      GMD      PC1
-3.651447e-04  1.703380e-03  3.717775e-04  2.857766e-03  1.069858e-03  2.830637e-04
 PVT      TMS      SCS      CTD      BCG      STG
 3.178249e-05  8.363843e-05  7.411322e-04  9.448998e-04 -1.078401e-03 -2.962114e-04
 PDN      LGC      VCG      HHV      HAH      CII
 3.456631e-04  7.128023e-04 -9.003981e-04 -8.685861e-04  7.217965e-04 -1.428779e-04
 VCB      BID      CTG      TCB      VPB      MBB
 3.202126e-04  8.303840e-04  8.910078e-04 -1.228768e-03 -4.493929e-04  1.454584e-03
 ACB      LPB      HDB      STB      VIB      SSI
 7.133053e-04  3.328774e-03  1.901912e-03  1.595072e-04  1.080046e-04 -6.144245e-04
 SSB      TPB      SHB
-1.935898e-03 -3.322192e-04  2.123958e-05
```


Tính phương sai và độ lệch chuẩn của tỷ suất lợi nhuận

#Tính phương sai, độ lệch chuẩn của lợi suất

```
var_r = apply(rate_of_return,2,var)
sigma_r = sqrt(var_r)
stock = cbind(expected_r, sigma_r)
stock
```

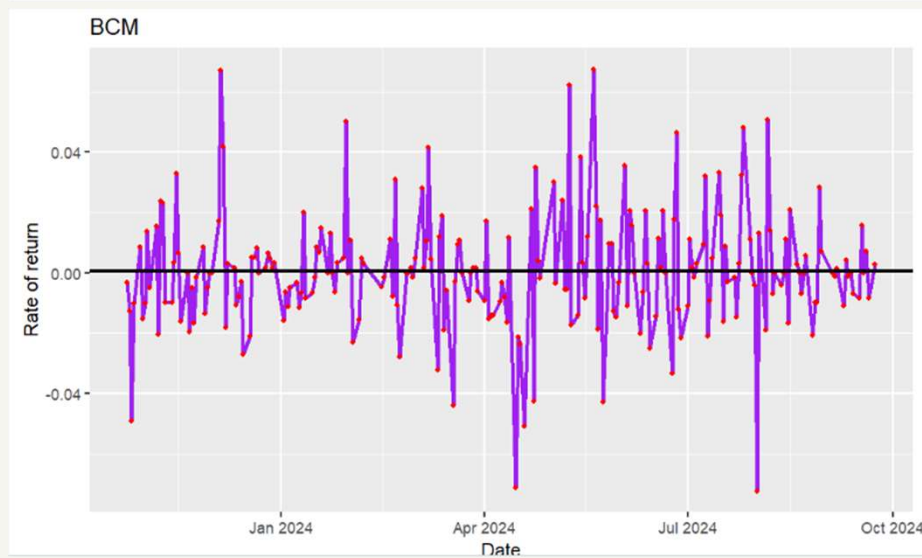
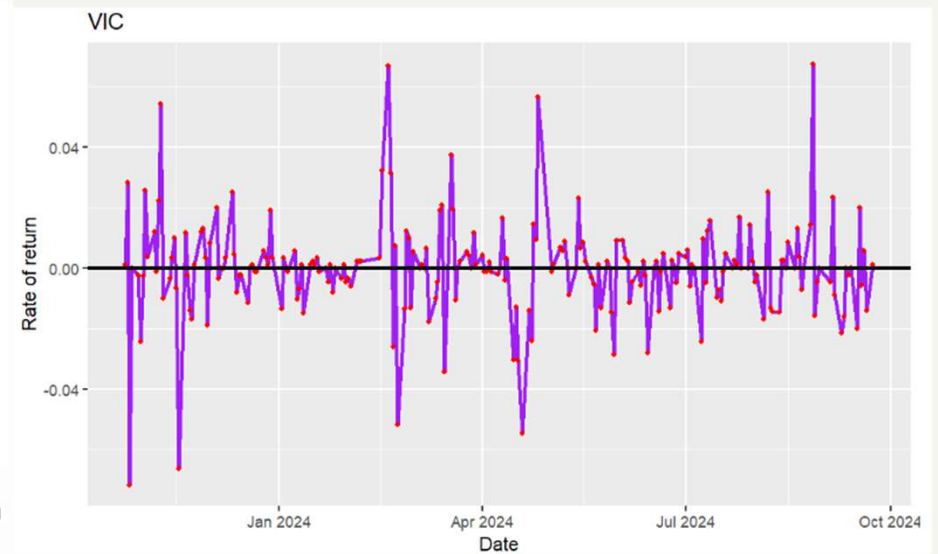
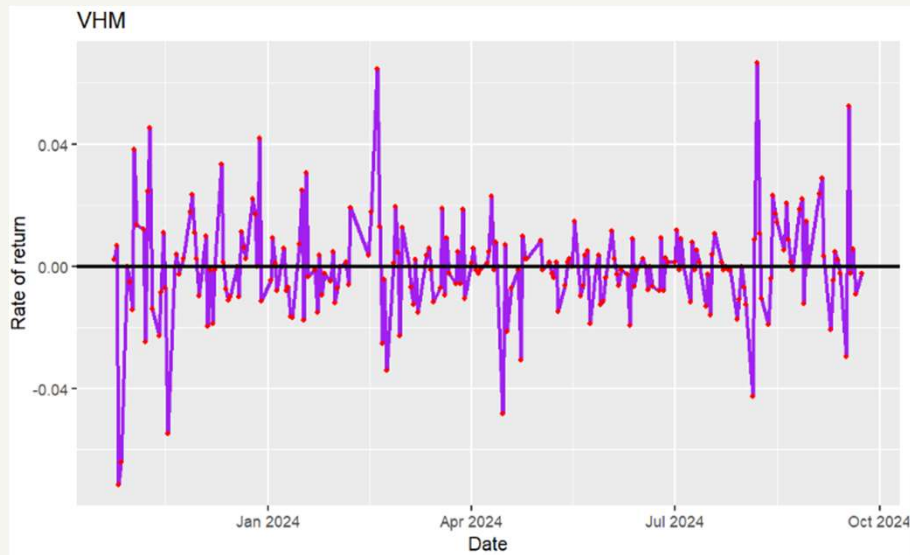
	expected_r	sigma_r
VHM	-8.458731e-05	0.01653443
VIC	-1.074168e-04	0.01623369
BCM	5.135262e-04	0.01961208
VRE	-1.375691e-03	0.02269743
KDH	1.003360e-03	0.01605161
KBC	-6.187658e-04	0.02183845
NVL	-7.506502e-04	0.02403828
PDR	-2.121100e-04	0.02560764
VPI	2.882328e-04	0.01670601
SIP	1.386182e-03	0.02386214
NLG	9.058588e-04	0.02204265
DIG	5.250910e-04	0.02470477
DXG	-3.651447e-04	0.02470506
TCH	1.703380e-03	0.02674909
HDG	3.717775e-04	0.02574565
HVN	2.857766e-03	0.03041338
GMD	1.069858e-03	0.01648671
PC1	2.830637e-04	0.02375688
PVT	3.178249e-05	0.02193910
TMS	8.363843e-05	0.02353842
SCS	7.411322e-04	0.01619748
CTD	9.448998e-04	0.02382197
BCG	-1.078401e-03	0.02358704
STG	-2.962114e-04	0.03213013
PDN	3.456631e-04	0.02807094

LGC	7.128023e-04	0.02842401
VCG	-9.003981e-04	0.02130568
HHV	-8.685861e-04	0.02084259
HAH	7.217965e-04	0.02298861
CII	-1.428779e-04	0.02135107
VCB	3.202126e-04	0.01116748
BID	8.303840e-04	0.01747732
CTG	8.910078e-04	0.01819390
TCB	-1.228768e-03	0.04777412
VPB	-4.493929e-04	0.01608643
MBB	1.454584e-03	0.01571909
ACB	7.133053e-04	0.01718131
LPB	3.328774e-03	0.01956121
HDB	1.901912e-03	0.01565594
STB	1.595072e-04	0.01682043
VIB	1.080046e-04	0.01765119
SSI	-6.144245e-04	0.02528941
SSB	-1.935898e-03	0.01749790
TPB	-3.322192e-04	0.01973557
SHB	2.123958e-05	0.01455315

...

Vẽ đồ thị lợi suất với câu lệnh

```
library(ggplot2)
#Vẽ đồ thị lợi suất
ror_df=data.frame(closing_price[-1,1],rate_of_return)
for (i in 1:45) {
  rate_of_return_plot <- ggplot(ror_df, aes(x = as.Date(ror_df$Date), y
= ror_df[,i+1])) +
    geom_line(color = "purple", size = 1) +
    geom_point(color = "red", size = 1)+
    geom_hline(yintercept = expected_r[i],color = "black", size=1)+
    labs(title=colnames(ror_df)[i+1],x="Date",y="Rate of return")
  print(rate_of_return_plot)
}
```



Các đồ thị lợi suất khác

1. Phương pháp 1

Phương pháp: Price-weighted Average

Ý tưởng: Đầu tư vào danh mục gồm tất cả các mã cổ phiếu, mỗi mã nắm giữ một số lượng cổ phiếu bằng nhau (tỷ trọng đầu tư vào mỗi mã tỷ lệ thuận với giá cổ phiếu của mã đó ở thời điểm ban đầu)

$$AP_t = \frac{\sum_{i=1}^K S_t^i}{d_t}$$

Nếu ngày thứ t có cổ phiếu chia tách, để triệt tiêu ảnh hưởng của việc chia tách cổ phiếu, d mới phải là nghiệm của phương trình:

$$\frac{\sum_{i=1}^K S_t^i \times Split^i}{d_{t-1}} = \frac{\sum_{i=1}^K S_t^i}{d_t}$$

$$\text{Hay } d_t = \frac{\sum_{i=1}^K S_t^i \times d_{t-1}}{\sum_{i=1}^K S_t^i \times Split^i}$$

Nếu ngày thứ t , không có cổ phiếu nào chia tách thì $d_t = d_{t-1}$

$$\text{Tính chỉ số: } I_t = I_0 \times \frac{AP_t}{AP_0}$$

Ưu điểm: Không cần tái cân bằng.

Nhược điểm: Khi có cổ phiếu chia tách, cần điều chỉnh ước số d .

1

Tính toán chỉ số ngành dựa trên phương pháp trọng số giá

#PHƯƠNG PHÁP 1: PRICE-WEIGHTED AVERAGE

```
closing_price_data = closing_price[, -1]
#Tính tổng các giá cổ phiếu của từng ngành theo ngày
price_sum = data.frame(matrix(0, nrow = 229,
ncol = 3))
for (i in 0:2) {
  price_sum[,i+1] = rowSums(closing_price_
data[, (15*i+1):(15*(i+1))])
}
head(price_sum, 5)
```

##		X1	X2	X3
##	1	489.45	584.47	407.75
##	2	496.65	582.33	414.05
##	3	495.45	585.13	411.65
##	4	470.80	557.96	398.70
##	5	473.50	570.50	404.80

Cập nhật tỷ lệ
chia tách cổ
phiếu (split
ratio) cho các
cổ phiếu
trong dữ liệu
giá đóng cửa.

```
#Tính ma trận split
#Tạo ma trận split ban đầu với tất cả phần tử = 1
split = data.frame(matrix(1, nrow = 229, ncol = 45))
#Viết hàm để đưa các hệ số tách (split ratio) vào trong ma trận split
func = function(date, stock, split_ratio){
  i = which(closing_price$Date == as.POSIXct(date, tz = "UTC"))
  j = which(names(closing_price) == stock)
  split[i, j-1] <- split_ratio
}
func("2024-07-25", "VPI", 1.2)
func("2024-07-10", "SIP", 1.15)
func("2024-06-24", "HDG", 1.1)
func("2024-07-19", "HHV", 1.05)
func("2023-11-22", "PC1", 1.15)
func("2024-04-11", "PVT", 1.1)
func("2024-09-23", "TMS", 1.07)
func("2024-08-01", "BCG", 1.1)
func("2024-06-14", "VCG", 1.12)
func("2024-06-21", "HAH", 1.15)
func("2023-11-28", "BID", 1.1269)
func("2023-11-30", "CTG", 1.117415)
func("2024-06-20", "TCB", 2)
func("2024-05-31", "ACB", 1.15)
func("2024-08-22", "VIB", 1.17)
colnames(split) = colnames(closing_price_data)
```

#Tính ma trận d: Ma trận chứa các ước số d
#Tạo ma trận d ban đầu với tất cả phần tử = 15
vì số mã cổ phiếu của 1 ngành là 15

```
d = data.frame(matrix(15, nrow = 229, ncol = 3)
)
for(j in 0:2){
  for(i in 2:229){
    if(any(split[i, (15*j+1):(15*(j+1))]>1)){
      d[i, j+1] = price_sum[i, j+1]*d[i-1, j+1]
    }else{d[i, j+1] = d[i-1, j+1]}
  }
}
```

```
colnames(d) = c("Real Estate",
"Industry", "Finance")
head(d, 30)
```

Description: df [30 × 3]

	Real Estate <dbl>	Industry <dbl>	Finance <dbl>
21	15	15.0000	15.00000
22	15	15.0000	15.00000
23	15	14.9023	15.00000
24	15	14.9023	15.00000
25	15	14.9023	15.00000
26	15	14.9023	15.00000
27	15	14.9023	14.81958
28	15	14.9023	14.81958
29	15	14.9023	14.70588
30	15	14.9023	14.70588

21-30 of 30 rows

Previous 1 2 3 Next

#Tính AP (Average Price)

```
AP = price_sum/d
```

*#Tạo ma trận AP_0 là vector AP tại t
hời điểm ban đầu được lặp lại 229 lần*


```
AP_0 = as.data.frame(matrix(rep(as.n  
umeric(AP[1, ]), times = 229), nrow  
= 229, byrow = TRUE))
```

#Tính chỉ số

```
I_1 = 100*AP/AP_0
```

```
colnames(I_1) = c("Real Estate", "In  
dustry", "Finance")
```

```
I_1
```



##	Real Estate	Industry	Finance
## 1	100.00000	100.00000	100.00000
## 2	101.47104	99.63386	101.54506
## 3	101.22587	100.11292	100.95647
## 4	96.18960	95.46427	97.78050
## 5	96.74124	97.60980	99.27652
## 6	95.05567	95.22302	97.14286
## 7	92.86955	93.93296	96.89761
## 8	93.51313	95.64905	97.90313
## 9	97.44611	98.77496	100.85837
## 10	97.78323	100.27717	100.38013
## 11	98.52896	102.35256	101.94972
## 12	97.10900	101.42009	100.84611
## 13	101.26673	104.12339	103.92397
## 14	104.22924	103.44586	102.94298
## 15	103.92277	103.35004	101.53280

” 2. Phương pháp 2

Phương pháp: Market-value-weighted Index

Ý tưởng: Đầu tư vào một danh mục gồm tất cả các mã cổ phiếu, tỷ trọng đầu tư vào mỗi mã tỉ lệ thuận với vốn hóa thị trường của mã đó tại thời điểm ban đầu.

$$VA_t = \sum_{i=1}^K S_t^i \times k_t^i$$

Tính chỉ số $I_t = \frac{VA_t}{VA_0} \times I_0$

Ưu điểm: Phản ánh đúng tỷ trọng giá trị của chứng khoán trong thị trường. Không cần phải điều chỉnh khi có chia tách cổ phiếu. Không cần tái cân bằng.

Nhược điểm: Xuất hiện thiên lệch về vốn hóa lớn (large-cap bias) khi cổ phiếu có giá trị vốn hóa thị trường lớn hơn có tác động lớn hơn.



2. MARKET-VALUE-WEIGHTED INDEX

```
#PHƯƠNG PHÁP 2: MARKET-VALUE-WEIGHTED  
INDEX  
#Tính tổng vốn hóa thị trường của từng  
ngành theo ngày  
market_cap_data = market_cap[, -1]  
total_market_cap = data.frame(matrix(0  
, nrow = 229, ncol = 3))  
for (i in 0:2) {  
  total_market_cap[, i+1] = rowSums(mar  
ket_cap_data[, (15*i+1):(15*(i+1))])  
}  
colnames(total_market_cap) = c("Real E  
state", "Industry", "Finance")  
head(total_market_cap, 5)
```

##	Real Estate	Industry	Finance
## 1	647211	125724	1607925
## 2	653066	126664	1631757
## 3	657173	126635	1647356
## 4	616315	120265	1603134
## 5	604255	122146	1623122

■ #Tính chỉ số

```
I_2 = data.frame(matrix(0, nrow = 229, ncol = 3))
```

#Tạo ma trận total_market_cap_0 là vốn hóa thị trường của từng ngành tại thời điểm ban đầu, được lặp lại 229 lần

```
total_market_cap_0 = as.data.frame(matrix(rep(as.numeric(total_market_cap[1, ]), times = 229), nrow = 229, byrow = TRUE))
```

```
I_2 = 100*total_market_cap/total_market_cap_0
```

```
colnames(I_2) = c("Real Estate", "Industry", "Finance")
```

```
I_2
```

Description: df [229 x 3]

Real Estate <dbl>	Industry <dbl>	Finance <dbl>
100.00000	100.00000	100.00000
100.90465	100.74767	101.48216
101.53922	100.72460	102.45229
95.22629	95.65795	99.70204
93.36291	97.15408	100.94513
92.66962	94.34476	99.43797
90.80377	91.32385	99.47243
90.81613	93.75855	100.11947
94.17315	97.32748	103.34120
95.06869	98.26843	102.85666

1-10 of 229 rows

Previous 2 3



3. Phương pháp 3

Ưu điểm: Không bị ảnh hưởng bởi việc chia tách cổ phiếu và giảm sự lệ thuộc vào các cổ phiếu lớn và phản ánh rõ hơn hiệu suất tổng thể.

Nhược điểm: Trọng số không phản ánh đúng tỷ trọng giá trị của chứng khoán trong thị trường mục tiêu. Cần tái cân bằng thường xuyên để duy trì trọng số bằng nhau.

Phương pháp: Equally-weighted index

Ý tưởng: Đầu tư vào danh mục gồm tất cả các mã cổ phiếu, với trọng số mỗi mã bằng nhau. Nếu giá cổ phiếu thay đổi, cần tái cân bằng (rebalance) để đảm bảo các trọng số bằng nhau

Công thức:

$$PI_t = \frac{1}{K} \sum_{i=1}^N \frac{S_t^i}{S_{t-1}^i}$$

Tính chỉ số:

$$I_t = I_{t-1} \times PI_t$$

(Với $I_0 = 100$)

3. EQUALLY-WEIGHTED INDEX

#PHƯƠNG PHÁP 3: EQUALLY-WEIGHTED INDEX

#Tính PI

```
PI = data.frame(matrix(0, nrow = 228, ncol = 3))
for (i in 0:2) {
  PI[, i+1] = rowMeans(closing_price_data[2:229, (15*
i+1):(15*(i+1))]/closing_price_data[1:228, (15*i+1)
:(15*(i+1))])
}
colnames(PI) = c("Real Estate", "Industry",
"Finance")
head(PI, 5)
```

##		Real Estate	Industry	Finance
##	1	1.0201750	1.0073115	1.0147254
##	2	0.9973221	1.0017716	0.9947275
##	3	0.9474032	0.9473570	0.9641291
##	4	1.0108996	1.0153583	1.0185695
##	5	0.9783248	0.9678841	0.9746469

● ● ●

#Tính chỉ số

```
I_3 = data.frame(matrix(0, nrow =  
228, ncol = 3))
```

```
I_3 = rbind(c(rep(100, 3)), I_3)
```

#Các phần tử ở dòng 1 bằng 100

```
for(j in 1:3){  
  for (i in 2:229){  
    I_3[i, j] = I_3[i - 1, j]*PI[i  
- 1, j]  
  }  
}
```

```
colnames(I_3) = c("Real Estate",  
"Industry", "Finance")
```

```
I_3
```

Description: df [229 × 3]

Real Estate <dbl>	Industry <dbl>	Finance <dbl>
100.00000	100.00000	100.00000
102.01750	100.73115	101.47254
101.74431	100.90960	100.93753
96.39288	95.59742	97.31681
97.44353	97.06564	99.12393
95.33142	93.94829	96.61083
92.74470	91.10373	96.36425
93.77565	93.74369	97.42581
98.40948	97.59248	100.61156
98.92610	98.86527	99.97372

1-10 of 229 rows

Previous 2 3

#TỔNG HỢP CÁC CHỈ SỐ

```
combined_df=data.frame(closing_price[,1],I_1,I_2,I_3)
colnames(combined_df)=c("Date","Real Estate 1","Industry 1","Finance 1","Real Estate 2","Industry 2","Finance 2","Real Estate 3","Industry 3","Finance 3")
head(combined_df, 5)
```

```
##      Date Real Estate 1 Industry 1 Finance 1 Real Estate 2 Industry 2
## 1 2023-10-23      100.00000    100.00000  100.00000      100.00000    100.00000
## 2 2023-10-24      101.47104     99.63386  101.54506      100.90465    100.74767
## 3 2023-10-25      101.22587    100.11292  100.95647      101.53922    100.72460
## 4 2023-10-26       96.18960     95.46427   97.78050       95.22629     95.65795
## 5 2023-10-27       96.74124     97.60980   99.27652       93.36291     97.15408
##      Finance 2 Real Estate 3 Industry 3 Finance 3
## 1 100.00000      100.00000    100.00000  100.00000
## 2 101.48216      102.01750    100.73115  101.47254
## 3 102.45229      101.74431    100.90960  100.93753
## 4  99.70204       96.39288     95.59742   97.31681
## 5 100.94513       97.44353     97.06564   99.12393
```

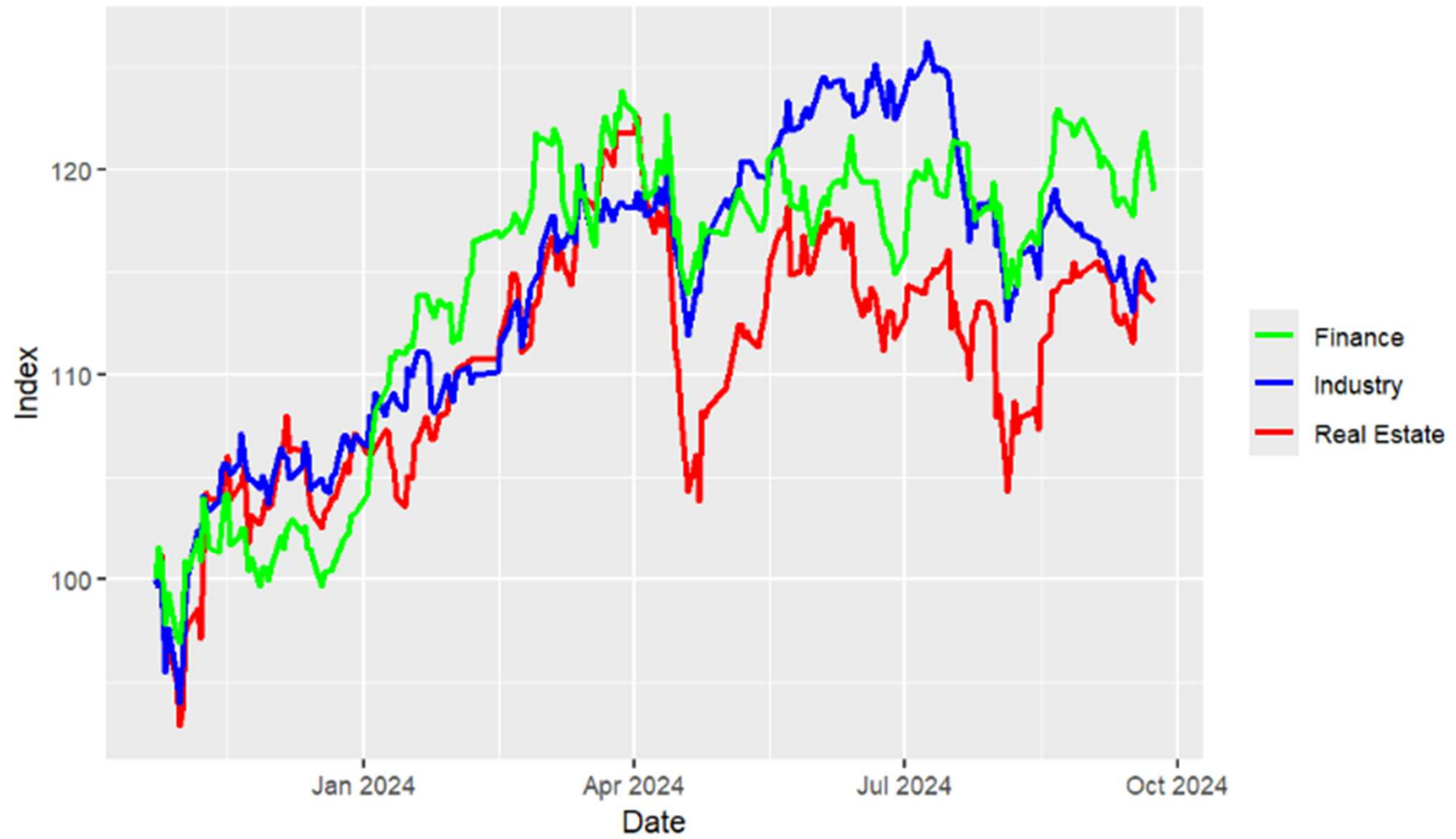

#Vẽ đồ thị các ngành theo từng chỉ số

```
index_name=c("Price Weighting","Value Weighting","Equal Weighting")

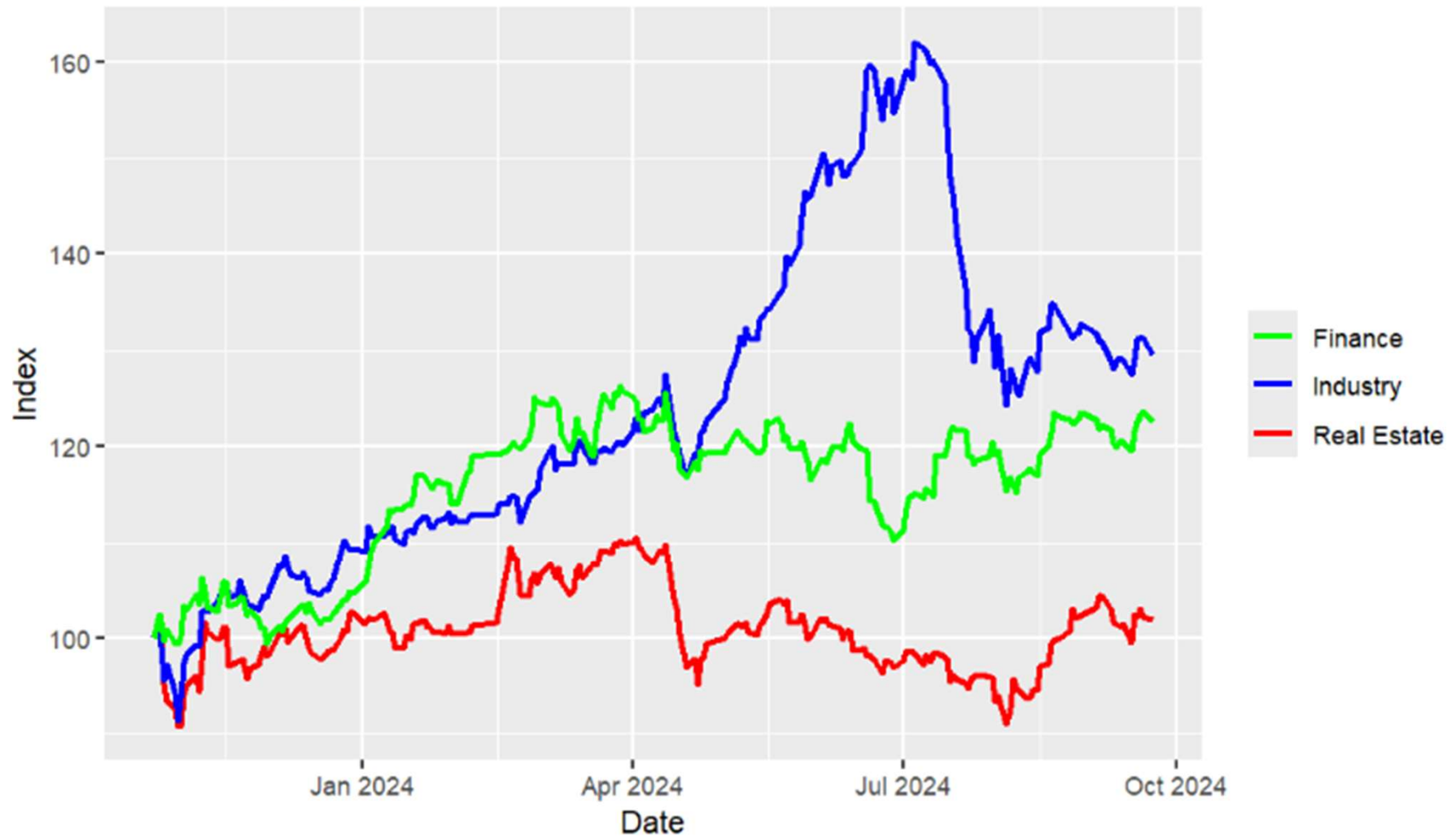
for(j in 1:3){
  index_plot <- ggplot(data = combined_df, aes(x = as.Date(combined_df[,
1])))+
    geom_line(aes(y = combined_df[, 3*j-1], color = "Real Estate"), size = 1)
  +
    geom_line(aes(y = combined_df[, 3*j], color = "Industry"), size = 1) +
    geom_line(aes(y = combined_df[, 3*j+1], color = "Finance"), size = 1) +
    labs(title = index_name[j], x = "Date", y = "Index") +
    scale_color_manual(values=c("Real Estate" = "red","Industry" =
"blue","Finance" = "green")) +
    theme(legend.title = element_blank())

  print(index_plot)
}
```

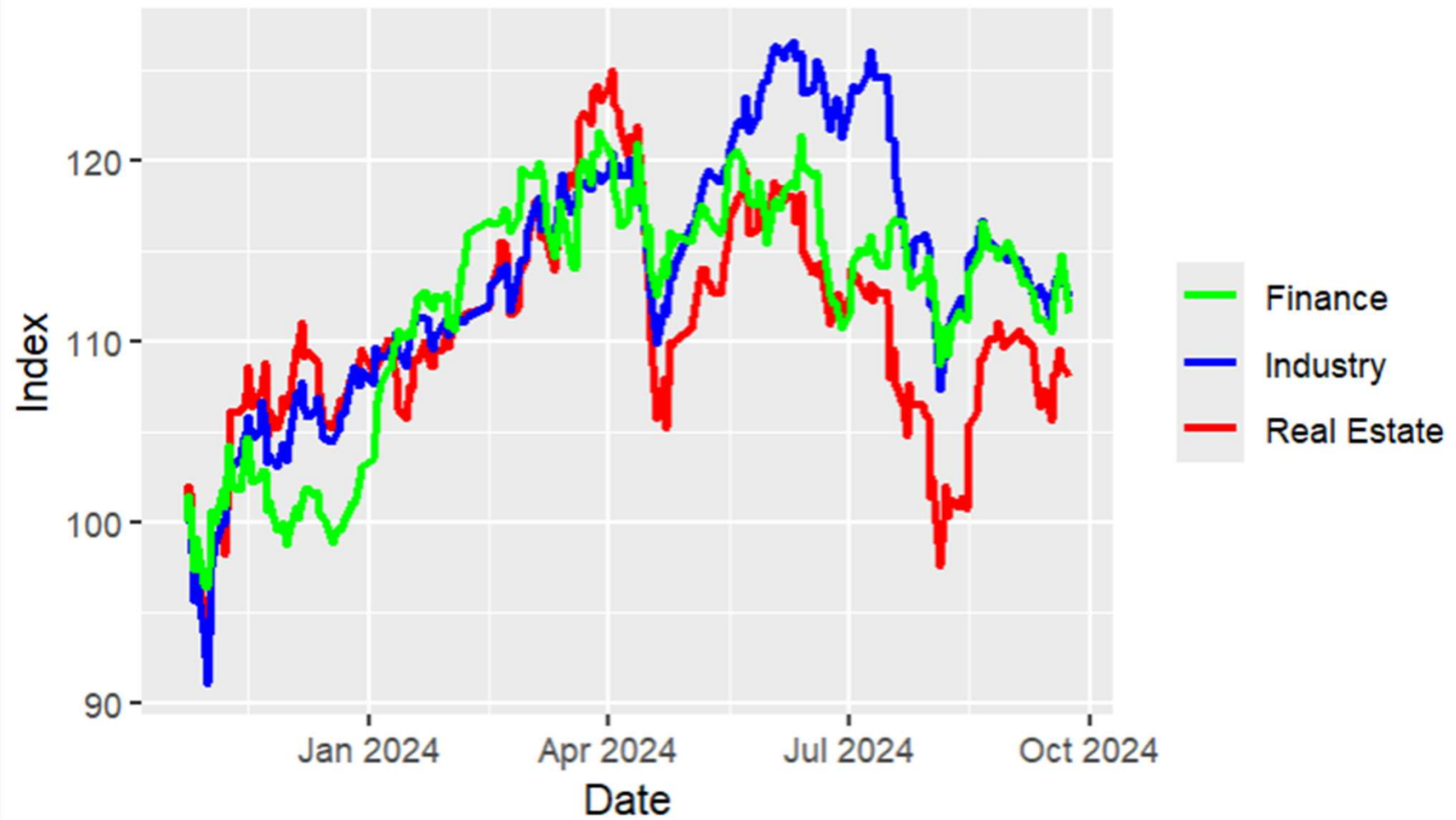

Price Weighting



Value Weighting



Equal Weighting



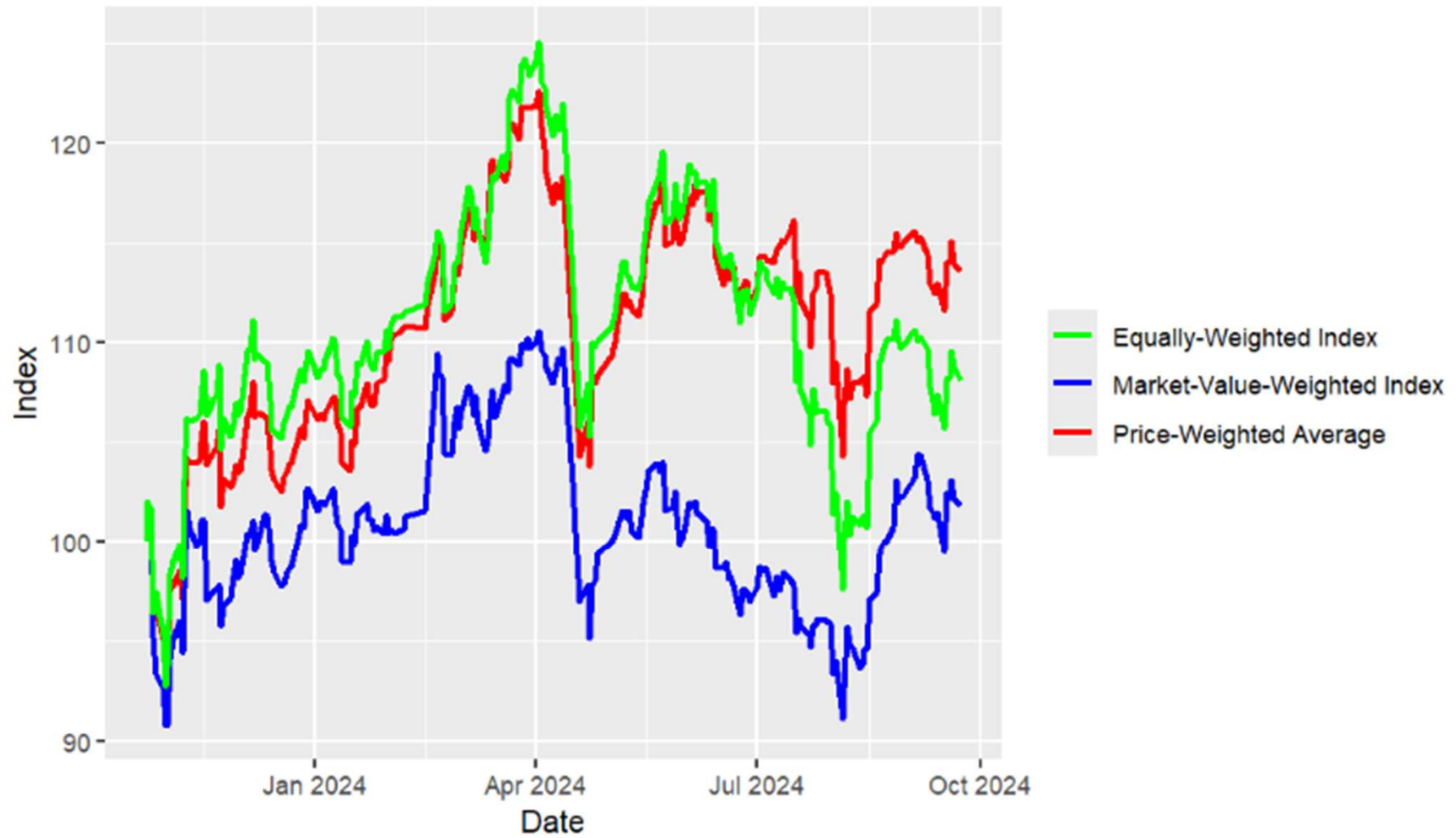
Vẽ đồ thị các chỉ số theo từng ngành

```
nganh=c("Real Estate","Industry","Finance")

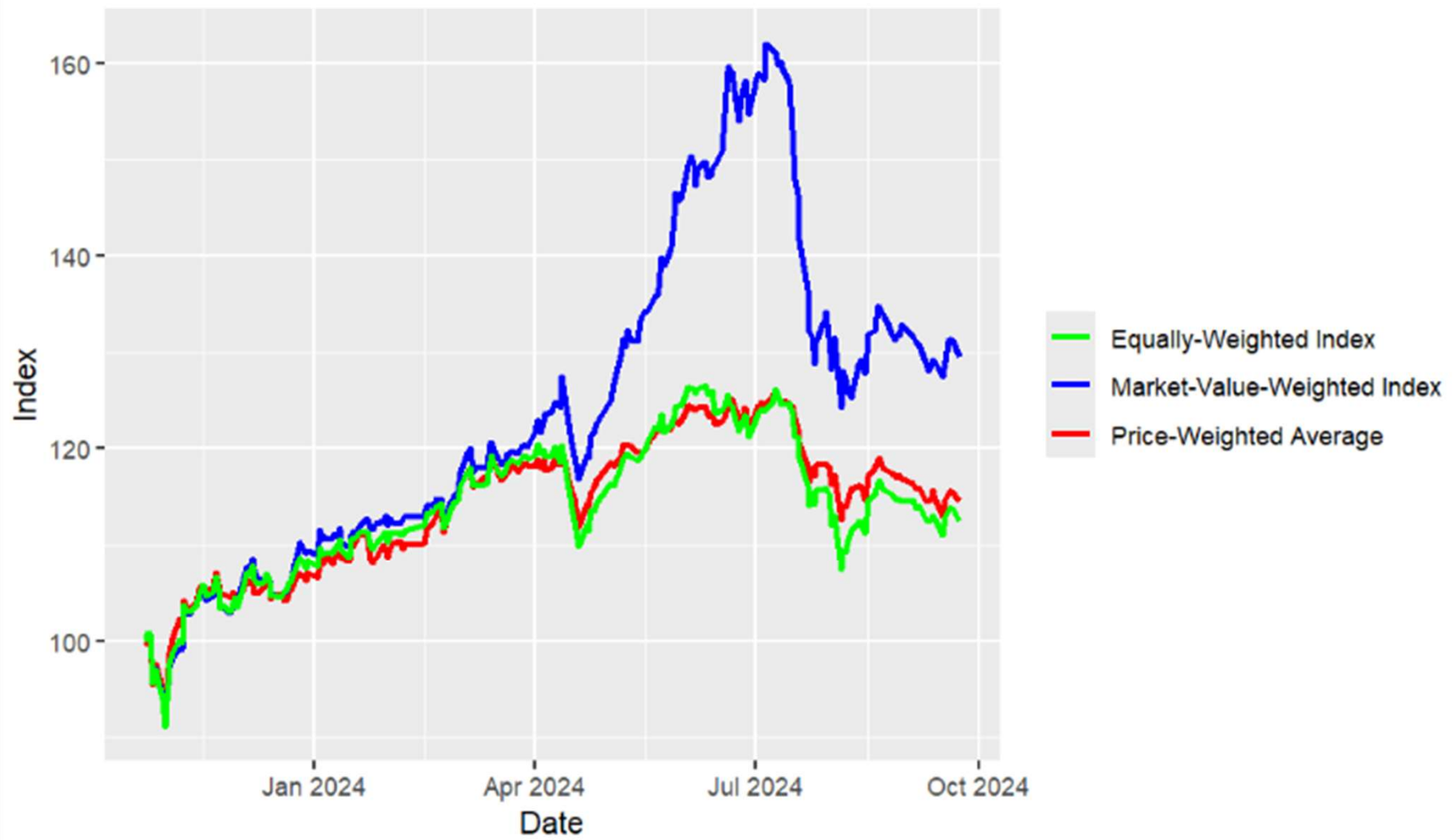
for(j in 1:3){
  nganh_plot <- ggplot(data = combined_df, aes(x = as.Date(combined_df[, 1]))) +
    geom_line(aes(y = combined_df[, j+1], color = "Price-Weighted Average"), size = 1)
+
    geom_line(aes(y = combined_df[, j+4], color = "Market-Value-Weighted Index"), size
= 1) +
    geom_line(aes(y = combined_df[, j+7], color = "Equally-Weighted Index"), size = 1)
+
    labs(title = nganh[j], x = "Date", y = "Index") +
    scale_color_manual(values=c("Price-Weighted Average" = "red","Market-Value-Weighte
d Index" = "blue","Equally-Weighted Index" = "green")) +
    theme(legend.title = element_blank())

  print(nganh_plot)
}
```

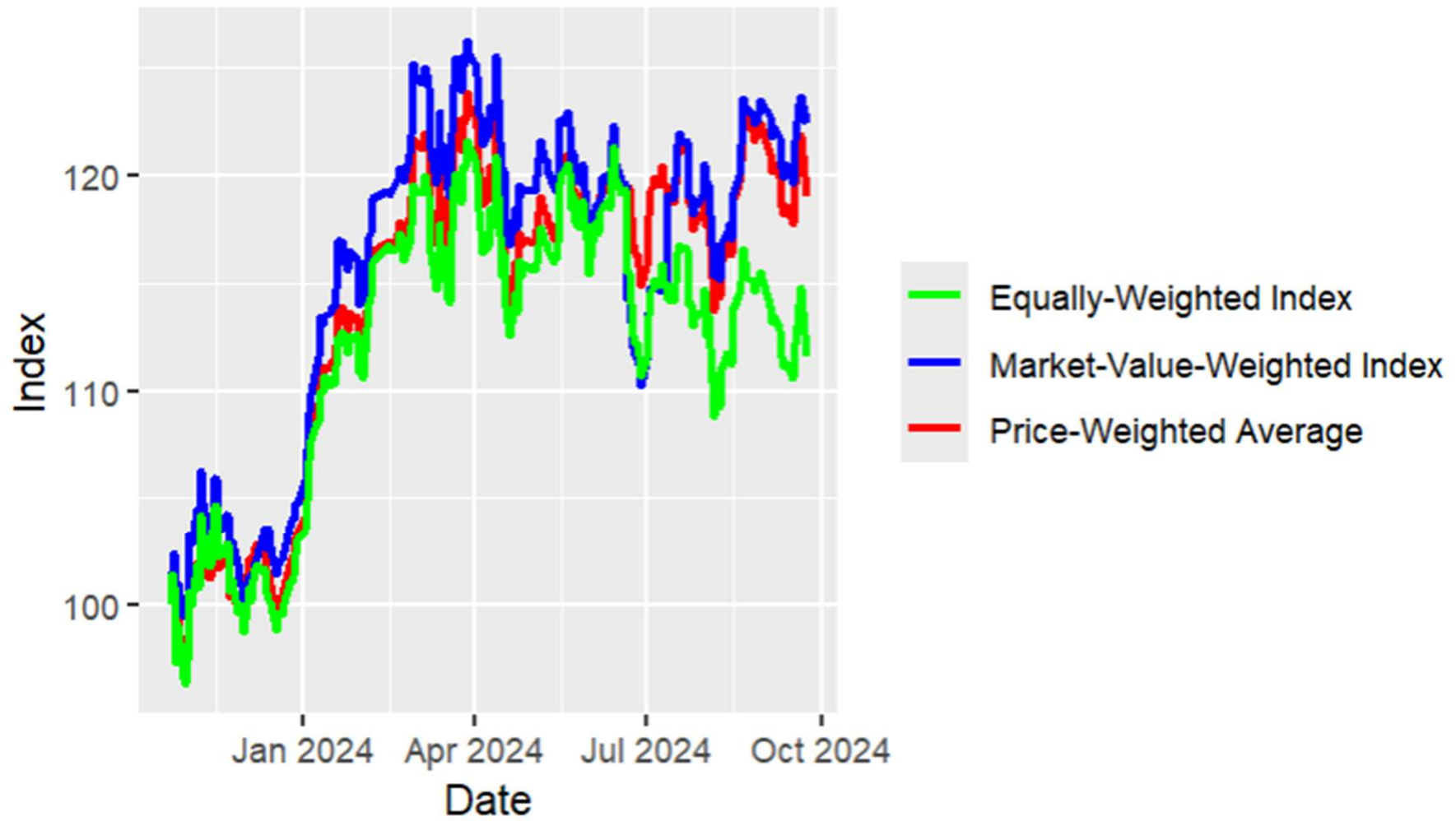
Real Estate



Industry



Finance



Một số cách tính chỉ số giá thị trường trong nước



“VN-Index = (Tổng giá trị thị trường của các cổ phiếu niêm yết hiện tại / Tổng giá trị của các cổ phiếu niêm yết cơ sở) x 100”

Đổi 1 điểm VN-Index:

“1 điểm Vn-index = Tổng giá trị cổ phiếu của Vn-index/100”



“HNX-Index = (Giá trị vốn hoá thị trường hiện tại) / (Giá trị vốn hoá thị trường cơ sở) x 100”

Một số cách tính chỉ số giá thị trường nước ngoài



$$DJIA = \sum P_i / n$$

Trong đó:

P_i : giá cổ phiếu trong danh sách

n : số cổ phiếu được xét (30)



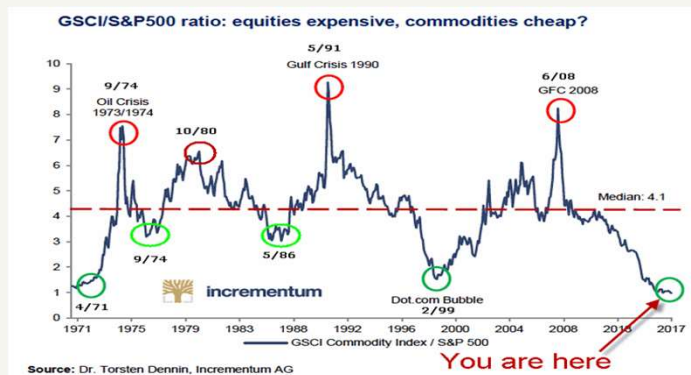
Trọng số = (Vốn hóa thị trường của công ty thành phần)/(Tổng vốn hóa thị trường của 500 công ty trong nhóm S&P 500)



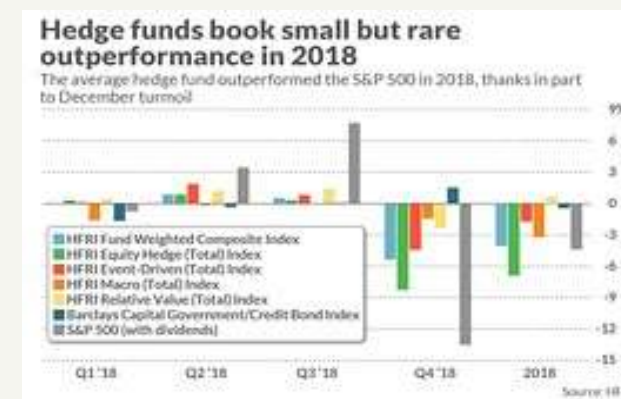
FTSE 100 = tổng giá trị thị trường của 100 công ty bao gồm nó+ giá trị của chỉ số



*Chỉ số Nikkei 225 = (Giá trị vốn hóa của 225 công ty đang niêm yết trên thị trường chứng khoán Nhật Bản / Giá trị vốn hóa thị trường) * Chỉ số toàn thị trường.*



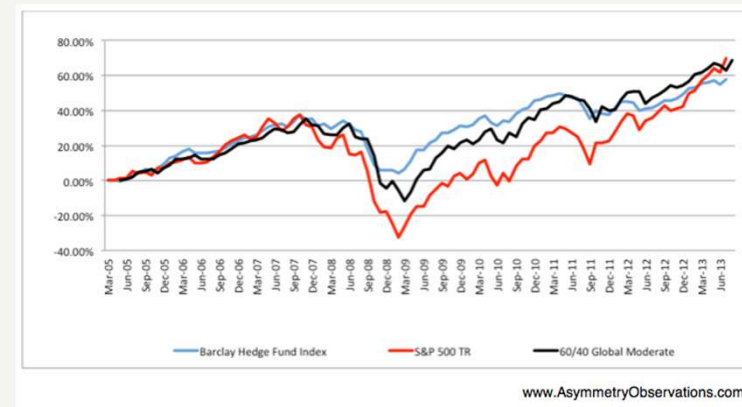
*S&P GSCI = Tổng (trọng số của hàng hóa i * (giá hàng hóa i tại thời điểm t / giá hàng hóa i tại thời điểm cơ sở))*



Chỉ số HFRI 500 = (Tỷ suất sinh lợi của quỹ 1 + Tỷ suất sinh lợi của quỹ 2 + ... + Tỷ suất sinh lợi của quỹ 500) / 500



*CRB Index = Tổng của (trọng số của hàng hóa iii * (giá của hàng hóa iii tại thời điểm ttt / giá của hàng hóa iii tại thời điểm cơ sở))*



Chỉ số Barclay Hedge Fund = Tổng (tỷ suất sinh lợi của mỗi quỹ) / Số lượng quỹ

Ứng dụng của chỉ số thị trường

1 Đơn giản hóa quy trình nghiên cứu

2 ETF chỉ số và quỹ tương hỗ giúp dễ dàng đa dạng hóa

- Quỹ tương hỗ
- Exchange-traded fund (ETF)

3 Cho phép các nhà đầu tư tiếp cận với thị trường hàng hóa

Mã ETF	Ngày thành lập	Công ty quản lý quỹ	Mô phỏng chỉ số
E1VFN30	06/10/2014	Công ty Cổ phần quản lý Quỹ đầu tư Việt Nam- VFM	VN30
FUEMAV30	08/12/2020	Mirae Asset	VN30
FUESSV30	12/08/2020	Chứng khoán SSI	VN30
FUESSV50	10/12/2014	Chứng khoán SSI	VNX50
FUEVNI00	16/06/2020	Công ty quản lý quỹ VinaCapital	VNI00
FUESSVFL	24/02/2020	Chứng khoán SSI	VNFIN LEAD
FUEVFNVD	22/04/2020	Công ty cổ phần quản lý Quỹ đầu tư Việt Nam-VFM	VN Diamond

...

**Thank you
for your
attention!**

