ĐẠI HỌC BÁCH KHOA HÀ NỘI

KHOA TOÁN - TIN

TỐI ƯU HÓA DANH MỤC ĐẦU TƯ CHỨNG KHOÁN

ĐỒ ÁN I

Chuyên ngành: Hệ Thống Thông Tin Quản Lý

Giảng viên hướng dẫn:	TS. Trần Ngọc Thăng
Sinh viên thực hiện:	Trần Thị Thanh Tâm
Lớp:	HTTTQL 02 K67
Mã số sinh viên:	20227261

HÀ NỘI - 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Mục tiêu và nội dung của đồ án

2. Kết quả đạt được

3. Ý thức làm việc của sinh viên

Hà Nội, ngày 30 tháng 5 năm 2025
Giảng viên hướng dẫn

PHIẾU BÁO CÁO TIẾN ĐỘ ĐỒ ÁN

Danh sách lớp học / Đồ án I / 751329

Thông tin lớp học mã 751329

Kì học: 20242 Mã học phần: MI3380

Tên học phần: Đồ án I Mã lớp: 751329

Đồ án

Giáo viên hướng dẫn: Trần Ngọc Thăng

Tên đồ án: Tối ưu danh mục đầu tư và ứng dụng

Nội dung:

Các mốc kiểm soát chính:

Giáo viên phản biện:

Danh sách đánh giá đồ án

Ngày đánh giá	Lần	Nội dung kế hoạch	Nội dung đã thực hiện	Điểm tích cực	Điểm nội dung	Ghi chú
13/04/2025	1	Tốt	Tốt	10	10	
15/05/2025	2	Tốt	Tốt	10	10	

Lời cảm ơn

Trong quá trình thực hiện đồ án, em đã nhận được sự hướng dẫn và hỗ trợ quý báu từ các thầy cô giáo. Em xin bày tỏ lòng biết ơn chân thành đến TS. Trần Ngọc Thăng - giảng viên khoa Toán - Tin, Trường Đại học Bách khoa Hà Nội - người đã tận tình chỉ bảo, định hướng và giúp đỡ em trong suốt thời gian thực hiện đề tài.

Em cũng xin chân thành cảm ơn các thầy cô trong Khoa Toán - Tin nói riêng và Trường Đại học Bách khoa Hà Nội nói chung, những người đã giảng dạy, truyền đạt cho em những kiến thức quý báu, giúp em có nền tảng lý thuyết vững vàng để hoàn thành đồ án này.

Mặc dù đã cố gắng hoàn thiện, nhưng do giới hạn về thời gian và kinh nghiệm thực tế, đồ án không tránh khỏi những thiếu sót. Em mong nhận được sự góp ý từ các thầy cô để có thể tiếp tục hoàn thiện bản thân và nâng cao năng lực chuyên môn trong tương lai.

Hà Nội, ngày 30 tháng 5 năm 2025
Sinh viên thực hiện

Trần Thị Thanh Tâm

Tóm tắt nội dung đồ án

Đồ án tập trung vào phân tích lý thuyết và xây dựng hệ thống danh mục đầu tư thông qua dữ liệu chứng khoán. Cấu trúc nội dung của đồ án bao gồm các phần sau:

- **Chương 1:** Tổng quan về chứng khoán tài chính với hai loại chứng khoán chính là cổ phiếu và trái phiếu. Chương này cũng trình bày các phương pháp điều chỉnh dữ liệu giá chứng khoán cho các sự kiện như cổ tách và sáp nhập, cũng như các kỹ thuật thu thập và làm sạch dữ liệu lịch sử.
- **Chương 2:** Thảo luận về việc sử dụng lợi nhuận log để tính giá và mô tả sự biến động giá thông qua mô hình phân phối hỗn hợp chuẩn.
- **Chương 3:** Định nghĩa Tỷ số Sharpe như một thước đo đánh giá hiệu quả đầu tư dựa trên lợi suất vượt trội so với rủi ro (độ lệch chuẩn), thường được quy đổi theo năm để so sánh nhất quán giữa các khoản đầu tư.
- **Chương 4:** Chương này trình bày quá trình triển khai và lập trình xây dựng hệ thống tối ưu hóa danh mục đầu tư.
- **Phần kết luận:** Đưa ra kết luận về kết quả của đồ án và hướng phát triển trong tương lai.

Danh mục viết tắt

Bảng 1. Danh mục viết tắt

Chữ viết tắt/Ký hiệu	Ý nghĩa
PBOC	Ngân hàng Nhân dân Trung Quốc
CNY	Đồng Nhân dân tệ
USD	Đồng Đô la Mỹ
FX	Thị trường ngoại hối
SR	Tỉ số Sharpe

Danh sách bảng

Danh sách hình vẽ

Mục lục

Chương 1: Chứng khoán tài chính

Chứng khoán là một công cụ tài chính có thể mua bán, thể hiện quyền sở hữu hoặc quyền đối với tài sản hay thu nhập trong tương lai. Chứng khoán được chia thành hai loại chính:

Cổ phiếu (Equity Securities) hay còn gọi là chứng khoán vốn: thể hiện quyền sở hữu trong một doanh nghiệp. Có hai loại cổ phiếu chính:

- **Cổ phiếu thường (Common Stock)**: Cổ đông có quyền biểu quyết và hưởng cổ tức nhưng không được đảm bảo cổ tức cố định.
- **Cổ phiếu ưu đãi (Preferred Stock)**: Cổ đông được nhận cổ tức cố định trước cổ đông thường nhưng thường không có quyền biểu quyết.

Trái phiếu (Debt Securities) hay chứng khoán nợ: thể hiện khoản vay mà tổ chức phát hành cam kết trả nợ cho nhà đầu tư. Một vài loại trái phiếu phổ biến như:

- **Trái phiếu chính phủ (Government Bond)**: Do chính phủ phát hành, thường có rủi ro thấp.
- **Trái phiếu doanh nghiệp (Corporate Bond)**: Do doanh nghiệp phát hành, có rủi ro cao hơn nhưng lãi suất hấp dẫn hơn.
- **Trái phiếu không lãi suất (Zero-Coupon Bond)**: Không trả lãi định kỳ mà được bán với giá thấp hơn mệnh giá và thanh toán đủ mệnh giá khi đáo hạn.

Ngoài ra, chứng khoán còn bao gồm một số các công cụ tài chính khác như chứng khoán phái sinh (derivatives) và chứng chỉ quỹ (fund certificates).

1.1 Đầu tư vào trái phiếu

Giả sử một khoản trái phiếu có các khoản thanh toán lãi suất (coupon) sau mỗi sáu tháng, lãi suất r thường được cố định trong suốt thời gian còn lại của trái phiếu. Khi một người mua một trái phiếu, người đó đang "mua dài hạn" trái phiếu, người bán sẽ trở thành người "bán khống trái phiếu". Trái phiếu sẽ có khoảng thời gian nhất định để nhận các khoản lãi suất. Ta có công thức tính giá trị trái phiếu:

Giá trị trái phiếu = Giá trị hiện tại của các khoản coupon+Giá trị hiện tại của mệnh giá

Trái phiếu trả lãi hàng năm:

$$BV_{\text{ann}} = \sum_{t=1}^{T} \frac{C}{(1+r)^t} + \frac{P}{(1+r)^T}$$

Trái phiếu trả lãi nửa năm:

$$BV_{\text{semi}} = \sum_{i=1}^{2T} \frac{C}{(1+r/2)^i} + \frac{P}{(1+r/2)^{2T}}$$

Công thức này giúp chiết khấu giá trị trong tương lai của trái phiếu về giá trị hiện tại. Trong đó:

- BV là giá trị hiện tại của trái phiếu.
- C là khoản giá coupon mỗi 6 tháng.
- P là mệnh giá trái phiếu.
- r là lãi suất hàng năm.
- T là tổng số chu kỳ thanh toán.
- $(1+r/2)^T$ dùng để chiết khấu mỗi khoản thanh toán về hiện tại.
- $(1+r/2)^i$ dùng để chiết khấu mệnh giá trái phiếu về hiện tại.

Chúng ta có thể rút ra mối quan hệ giữa lãi suất r và giá trị trái phiếu:

- Khi lãi suất tăng, giá trị trái phiếu giảm.
- Khi lãi suất giảm, giá trị trái phiếu tăng.

Chương trình R sau đây mô phỏng $r(t)$, tức là lãi suất thị trường dao động theo thời gian, với phân phối Gaussian $N(\mu = 0.03, \sigma^2 = (0.0025)^2)$. Trong chương trình này, giá trị trái phiếu được định nghĩa theo Công thức 4.3 và mở rộng để bao gồm cả các khoản thanh toán coupon đã được trả hoặc tích lũy trong quá khứ.

```
P <- 1000
T <- 20
r <- 0.06  # annual rate
C <- 30

```

```r
BV <- function(P, C, r, t, T) {
  # Finds coupon Bond Value at time t mat T
  tmat <- T - t
  acrued <- C * 2 * t  # already paid
  if (tmat != 0) {  # include interim coupons
    i <- seq(1, 2 * tmat)
    acrued + sum(C / (1 + r / 2)^i) + P / (1 + r / 2)^(2 * tmat)
  } else {  # no coupons left
    acrued + P / (1 + r / 2)^(2 * tmat)
  }
}

par(bg = "white")
par(mfrow = c(1,3))

# Simulate rates market for r
rvec <- round(c(r, r + rnorm(T) * .0050), 4)
plot(rvec, type = "l", ylim = c(0, .07), xlab = "Years", ylab = "r", col =
↪  4)
points(rvec, col = 4)

# Simulate PV of Bond at time t
simBV <- function(P, C, rvec, T) {
  BVvec = rep(0, T)
  for (t in 0:T) {
    i = t + 1
    BVvec[i] <- BV(P, C, rvec[i], t, T)
  }
  plot(BVvec, type = "l", col = 4, ylim = c(0, 2500), xlab = "Years", ylab
  ↪  = "Bond Value")
  points(BVvec, col = 4)
  BVvec
}

BV(P, C, r = 0.06, t = 0, T = 20)
BV(P, C, r = 0.06, t = 1/2, T = 20)
BV(P, C, r = 0.06, t = 1, T = 20)
BV(P, C, r = 0.07, t = 1/2, T = 20)
BV(P, C, r = 0.06, t = 20, T = 20)
BV(P, C, r = 0, t = 0, T = 20)

C <- 0
simBV(P, C, rvec, T)
```

```
47 C <- 30
48 simBV(P, C, rvec, T)
```

Dưới đây là kết quả của chương trình:

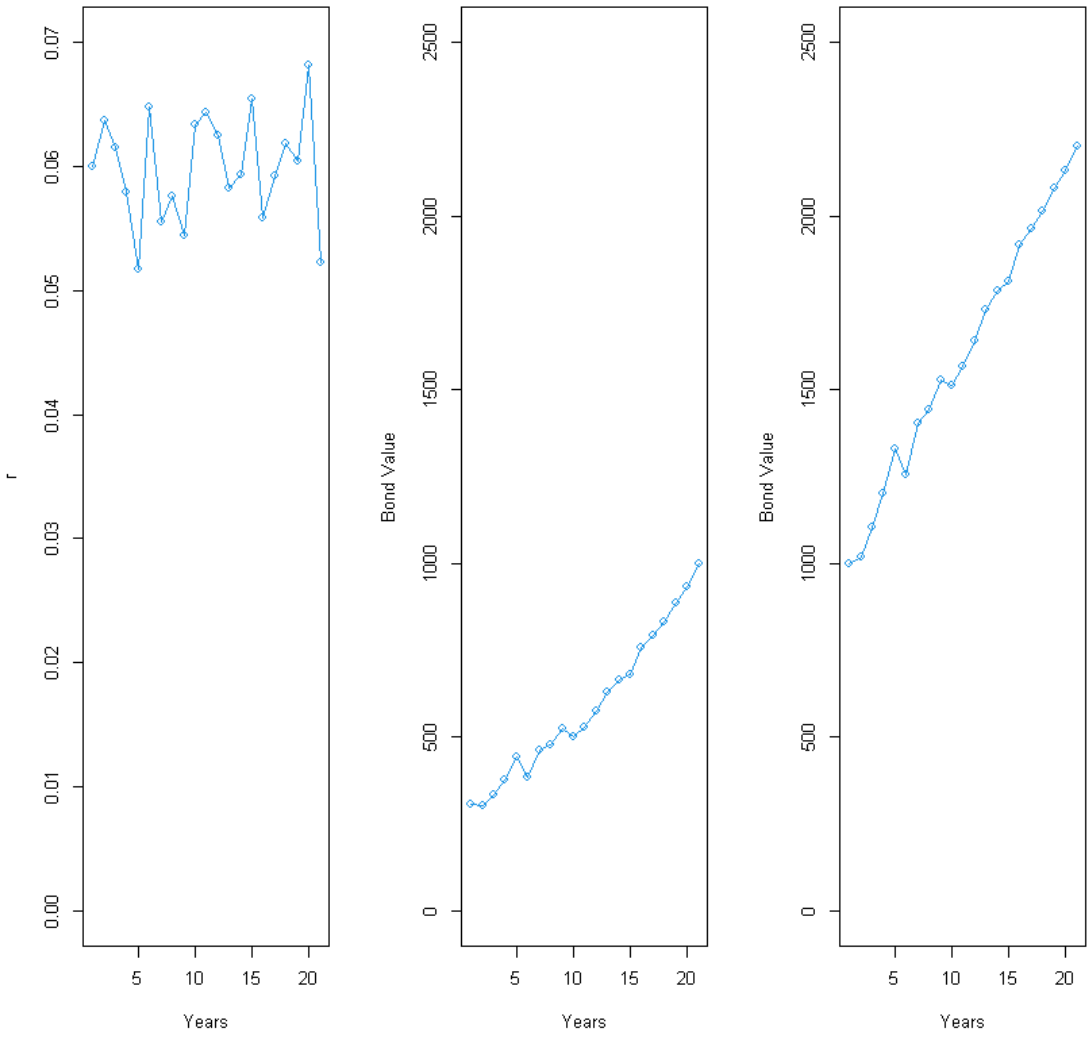
```
BV(P,C,r=.06,t=0,T=20): 999.99 # Giá trị khi chưa đến hạn
BV(P,C,r=.06,t=1/2,T=20): 1030
1060 # Giá trị sau 6 tháng
BV(P,C,r=.06,t=1,T=20): 924.487500654487 # Giá trị sau 1 năm
BV(P,C,r=.07,t=1/2,T=20): 2200 # Giá trị khi lãi suất thay đổi
BV(P,C,r=.06,t=20,T=20): 2200 # Giá trị tại ngày đáo hạn
BV(P,C,r=0,t=0,T=20): 2200 # Trường hợp lãi suất bằng 0
Giá trị trái sau mỗi 6 tháng:
306.556840773806
303.787847579755
336.108607791097
...
1079.25693573346
1132.20114078963
2200
```

1.2 Đầu tư cổ phiếu

Như đã nêu ở trên, cổ phiếu là một loại chứng khoán thể hiện quyền sở hữu một phần doanh nghiệp. Khi mua cổ phiếu, nhà đầu tư trở thành cổ đông của doanh nghiệp, có quyền biểu quyết hoặc nhận cổ tức của doanh nghiệp.

Khi nhà đầu tư mua cổ phiếu của doanh nghiệp, họ ở vị thế mua (long position) và kiếm lợi nhuận khi giá cổ phiếu tăng, thua lỗ khi giá giảm. Ngoài ra, nếu công ty trả cổ tức, nhà đầu tư cũng có thể nhận được khoản thu nhập định kỳ này, tùy thuộc vào chính sách chi trả cổ tức của công ty.

Bán khống (short position) là chiến lược nhà đầu tư vay cổ phiếu của công ty môi giới, bán ra thị trường và chờ giá giảm để mua về trả lại cho công ty môi giới. Lợi nhuận của phương pháp này thu được từ khoảng chênh lệch giữa hai giá. Tuy nhiên, nhà đầu tư phải trả lãi cho công ty môi giới. Ngoài ra, nhà đầu tư còn gặp rủi ro "ép phải mua lại" khi giá đang tăng mạnh để cắt lỗ, khiến giá càng bị đẩy cao hơn nữa. Đây là phương pháp đầu tư kém phổ biến do có rủi ro cao hơn.



Hình 1.1. Biểu đồ lãi suất và giá trị trái phiếu mô tả theo thời gian

1.3 Điều chỉnh cho Cổ Tách

Cổ tách là sự kiện mà công ty chia nhỏ số lượng cổ phiếu, khiến mỗi cổ phiếu có giá thấp hơn nhưng tổng giá trị đầu tư của cổ đông không thay đổi. Ví dụ, nếu tỷ lệ chia là 2:1, nhà đầu tư sẽ có gấp đôi số cổ phiếu nhưng giá trị của mỗi cổ phiếu sẽ giảm một nửa.

Lợi ích của việc chia cổ phiếu bao gồm:

- **Giảm giá cổ phiếu**, giúp cổ phiếu trở nên dễ tiếp cận với nhà đầu tư nhỏ lẻ.
- **Giảm chi phí hoa hồng**, vì hoa hồng thường tính theo số lượng cổ phiếu, không phải giá trị cổ phiếu.

Tuy nhiên, với sự giảm dần của phí hoa hồng trong những năm qua, việc chia cổ phiếu không còn phổ biến như trước. Một số công ty lớn, chẳng hạn như Google,

đã không chia cổ phiếu nữa.

Dữ liệu giá cổ phiếu cần phải được điều chỉnh sau sự kiện chia cổ phiếu để đảm bảo tính chính xác trong phân tích. Đặc biệt, các công cụ như R có thể hỗ trợ tự động phát hiện các sự kiện chia cổ phiếu và điều chỉnh dữ liệu để tránh sự thay đổi đột ngột không mong muốn trong đồ thị giá cổ phiếu. Ta sẽ thực thành điều chỉnh dữ liệu giá trong R:

```
splitAdjust <- function(prices, symbol) {
  len = length(prices)
  origFinalPrice = prices[len]

  for (j in 2:len) {
    split = 0

    # Forward splits (Giá giảm đáng kể, chia tách cổ phiếu)
    if (prices[j-1] >= 1.4 * prices[j]) {
      split = 1.5  # 3-for-2 split
    }
    if (prices[j-1] >= 1.8 * prices[j]) {
      split = 2    # 2-for-1 split
    }
    if (prices[j-1] >= 2.9 * prices[j]) {
      split = 3    # 3-for-1 split
    }
    if (prices[j-1] >= 3.9 * prices[j]) {
      split = 4    # 4-for-1 split
    }
    if (prices[j-1] >= 4.9 * prices[j]) {
      stop(paste(symbol, "detected more than 4:1 split"))
    }

    print(paste("Split adjusting", j, symbol,
    split, prices[j-1], prices[j]))

    # Reverse splits (Giá tăng đáng kể, gộp cổ phiếu)
    if (prices[j-1] <= prices[j] / 1.4) {
      split = -1.5
    }
    if (prices[j-1] <= prices[j] / 1.9 && prices[j-1] >= prices[j] / 2.1)
    ↪ {
      split = -2
    }
```

```
    if (prices[j-1] <= prices[j] / 2.9 && prices[j-1] >= prices[j] / 3.1)
    ↪  {
      split = -3
    }
    if (prices[j-1] <= prices[j] / 5.8 && prices[j-1] >= prices[j] / 6.2)
    ↪  {
      split = -6
    }
    if (prices[j-1] <= prices[j] / 7.7 && prices[j-1] >= prices[j] / 8.3)
    ↪  {
      split = -8
    }
    if (prices[j-1] <= prices[j] / 9.7 && prices[j-1] >= prices[j] / 10.3)
    ↪  {
      split = -10
    }
    if ((split == 0) && (prices[j-1] <= prices[j] / 2.9)) {
      stop(paste(symbol, "detected more than double reverse split"))
    }

    print(
    paste("Reverse split adjusting", j,
    symbol, split, prices[j-1], prices[j]))

    # Điều chỉnh giá theo split
    if (split != 0) {
      for (k in j:len) { # adjust all prices to right from j:len
        if (symbol == "C") {
          prices[k] = prices[k] / 10 # hard coded for Citi
        } else if (split == 1.5) {
          prices[k] = 1.5 * prices[k] # 3 for 2
        } else if (split == 2) {
          prices[k] = 2 * prices[k] # 2 to 1
        } else if (split == 3) {
          prices[k] = 3 * prices[k] # 3 to 1
        } else if (split == 4) {
          prices[k] = 4 * prices[k] # 4 to 1
        } else if (split == -1.5) {
          prices[k] = prices[k] / 1.5 # 2 to 3 reverse
        } else if (split == -2) {
          prices[k] = prices[k] / 2 # 1 to 2 reverse
        } else if (split == -3) {
          prices[k] = prices[k] / 3 # 1 to 3 reverse
```

```
        } else if (split == -6) {
          prices[k] = prices[k] / 6 # 1 to 6 reverse
        } else if (split == -8) {
          prices[k] = prices[k] / 8 # 1 to 8 reverse
        } else if (split == -10) {
          prices[k] = prices[k] / 10 # 1 to 10 reverse
        } else {
          stop('splitAdjust internal error')
        }
      }
    }
  }

  finalPrice = prices[len]
  return(prices * origFinalPrice / finalPrice)
}

# Unit test
p <- c(3.0, 3.0, 2.0, 11.88, 5.9, 1.95, 3.90, 3.90,
       1.5, 0.75, 1.00, 1.2, 1.4, 1.8, 2.1, 1.05, 1.30,
       1.31, 1.32, 0.44, 0.43, 0.11, 0.12, 0.13)
sap <- splitAdjust(p, "SYM")
par(bg = "white")
plot(p, type = 'l', ylim = c(0, 15))
points(sap, col = 4)

```

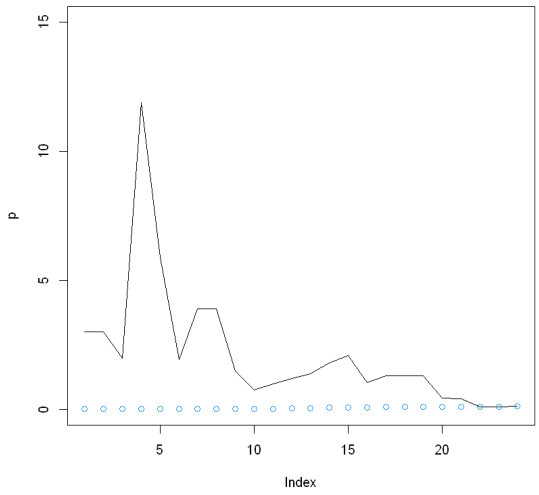
Bảng 1.1. Lịch sử điều chỉnh chia tách cổ phiếu

Loại điều chỉnh	Tỉ lệ cũ	Tỉ lệ mới	Hệ số điều chỉnh	Giá trước	Giá sau
Chia tách 2	0	3	3.000	3.000	1.500
Gộp ngược 2	0	3	3.000	1.500	3.000
Chia tách 3	1.5	3	2.000	3.000	1.500
Gộp ngược 3	1.5	3	2.000	1.500	3.000

Tiếp tục ở trang sau

Tiếp tục Bảng 1.1

Loại điều chỉnh	Tỉ lệ cũ	Tỉ lệ mới	Hệ số điều chỉnh	Giá trước	Giá sau
Chia tách 4	0	3	17.820	3.000	0.168
Gộp ngược 4	−6	3	17.820	0.168	3.000
Chia tách 5	2	2.97	1.475	2.970	2.000
Gộp ngược 5	2	2.97	1.475	2.000	2.970
Chia tách 6	3	2.95	0.975	2.950	3.000
Gộp ngược 6	3	2.95	0.975	3.000	2.950
Chia tách 7	0	2.925	5.850	2.925	0.500
Gộp ngược 7	−2	2.925	5.850	0.500	2.925
Tiếp tục các điều chỉnh khác...					



Hình 1.2. Biểu đồ điều chỉnh giá cổ phiếu

1.4 Điều chỉnh cho các thương vụ sáp nhập

Các thương vụ sáp nhập và mua lại diễn ra liên tục và ảnh hưởng đến tập dữ liệu cổ phiếu. Khi một công ty mua một công ty khác, thông thường một trong hai mã chứng khoán được giữ lại, trong khi mã còn lại sẽ ngừng giao dịch vào ngày sáp nhập hoặc một ngày trước đó.

Khi muốn đánh giá hiệu suất của danh mục đầu tư bằng cách lấy dữ liệu giá, có thể gặp lỗi: hoặc là thiếu tệp hoặc là một hàm truy vấn không thể trả về kết quả chính xác. Giả sử ta có một danh mục đầu tư có 2 sự biến đổi như sau:

- Công ty có mã cổ phiếu TIE đã bị công ty PCP mua lại và cả 2 công ty đều có trong danh mục đầu tư.
- Công ty có mã cổ phiếu CHV trong danh mục đầu tư đã bị công ty PCP ngoài danh mục đàu tư mua lại.

Sau khi hai sự kiện này diễn ra, tập dữ liệu chứa tên hai dữ liệu này sẽ gặp lỗi. Lúc này, ta cần xóa hai hàng liên quan đến hai mã cổ phiếu TIE và CHV rồi tạo một tệp mới

```
adjustForMergers <- function(dir, portFile) {
  # Take in symbols and their weights and emit a
  # rebalanced file summing close to 1.0
  homeuser <- "D:/FinAnalytics"
  setwd(paste0(homeuser, "/FinAnalytics/", dir, "/"))
  df <- read.csv(portFile)

  lab <- df[, 2]  # Tên các cổ phiếu
  w <- df[, 3]    # Trọng số danh mục

  total_weight <- sum(w)

  if (abs(total_weight - 1.0) < 0.002) {
    print('All weights sum to 1.0')
  } else {
    print(paste("Original total weight:", total_weight))

    amtToRealloc <- 1.0 - total_weight
    wInc <- w * (amtToRealloc / total_weight)  # Điều chỉnh tỷ lệ
    new_w <- w + wInc

    print(paste("New total weight after rebalancing:", sum(new_w)))
```

```
23
24      df[, 3] <- new_w  # Cập nhật trọng số mới
25
26      newFile <- paste0("rebal_", portFile)
27      write.csv(df, file = newFile, row.names = FALSE)
28
29      print(paste("Wrote file:", newFile))
30    }
31  }
32
33  # Gọi hàm với các tập dữ liệu khác nhau
34  adjustForMergers('huge', 'resD26QP1Days1258.csv')
35  adjustForMergers('huge', 'resD25Days1258woTIE.csv')
36  adjustForMergers('huge', 'resD24Days1258.csv')
```

[1] "All weights sum to 1.0"

[1] 0.9498

[1] 0.99748

[1] 0.9918

[1] 0.9999328

Điều chỉnh trọng số trong danh mục:

- TIE: Công ty mẹ mới PCP đã có trong danh mục đầu tư. Vì vậy, trọng số của TIE ($w_7 = 0.0491$) sẽ được cộng vào trọng số của PCP ($w_{15} = 0.0269$), tạo thành trọng số mới của PCP là $w_PCP = 0.0760$.
- CVH: Công ty mẹ AET không có trong danh mục đầu tư, vì vậy mã CVH bị loại bỏ hoàn toàn.

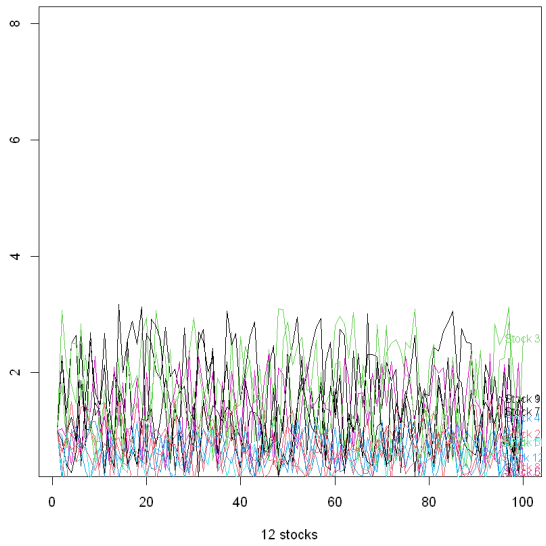
1.5 Vẽ nhiều chuỗi dữ liệu

Khi nhà đầu tư muốn so sánh trực quan lợi nhuận của các cổ phiếu hoặc đánh giá mức độ biến động và rủi ro của chúng, thì có thể sử dụng hàm plotMultSeries() trong R. Hàm này giúp chuẩn hóa tất cả giá cổ phiếu về 1 đơn vị tiền tệ hoặc 1 đơn vị lợi nhuận gộp tại thời điểm bắt đầu chuỗi thời gian. Nhờ đó, các cổ phiếu dù có mức giá ban đầu khác nhau vẫn có thể được so sánh trực quan trên cùng một biểu đồ.

```
1  plotMultSeries <- function(prices, lab, w, D, cc = "days", ret = NA,
2                             ylim = c(0.2, 15), isAlone = TRUE) {
```

```
  if (isAlone) plot.new()

  # Hàm ánh xạ màu sắc theo chỉ số d
  mapToCol <- function(d) {
    if (d %% 8 == 7) 1
    else if (d == 8) 2
    else if (d == 15) 3
    else if (d == 23) 4
    else d
  }

  # Thiết lập bố cục và lề biểu đồ
  par(mar = c(4, 3, 2, 1))
  if (isAlone) par(mfrow = c(1, 1))

  tot <- 0
  len <- nrow(prices)  # Số hàng của prices
  first <- TRUE

  for (d in 1:D) {
    if (!is.na(prices[1, d]) && !is.na(w[d]) && w[d] > 0) {
      print(lab[d])
      tot <- tot + 1

      if (first) {
        first <- FALSE
        plot(prices[, d] / prices[1, d], type = "l",
             col = mapToCol(d), xlab = cc, ylim = ylim,
             ylab = "Normalized Price")
      } else {
        lines(prices[, d] / prices[1, d], type = "l", col = mapToCol(d))
      }

      text(len, prices[len, d] / prices[1, d], labels = lab[d],
           col = mapToCol(d), cex = 0.8)
    }
  }

  print(tot)
  print(paste("Density or non-zero weights (sparsity) is", tot / D))
}

# Unit test:
```

```r
D2 <- 12
w <- rep(1 / D2, D2)

# Giả lập dữ liệu prices và lab
set.seed(123)
prices <- matrix(runif(100 * D2, min = 1, max = 10), ncol = D2)
lab <- paste("Stock", 1:D2)
par(bg = "white")
plotMultSeries(prices, lab, w, D2,
cc = paste(sum(w > 0), "stocks"), ret = "", ylim = c(0.5, 8))

```



Hình 1.3. Biểu đồ giá 12 chứng khoán đầu tiên trong dữ liệu cổ phiếu trên cùng một thang đo với giá khởi điểm là 1 đô la.

1.6 Nhập Dữ Liệu Chứng Khoán

Việc thu thập dữ liệu chứng khoán là bước quan trọng để đánh giá hiệu suất danh mục đầu tư và điều chỉnh chiến lược theo thời gian thực. Do các bộ dữ liệu có sẵn thường bị giới hạn về phạm vi cổ phiếu và khung thời gian, cần xây dựng cơ chế tự động lấy dữ liệu từ các nguồn bên ngoài. Ban đầu, có thể sử dụng hàm get.hist.quote() từ gói tseries để tải dữ liệu lịch sử từ Yahoo! Finance. Ví dụ:

```r
library(tseries)
pv <- get.hist.quote('YHOO',
                     quote="Adj",
                     start="2011-02-09",
                     end="2015-02-09")
```

Tuy nhiên: Yahoo! Finance đã ngừng hỗ trợ API miễn phí từ năm 2017, nên phương pháp này hiện không khả thi. Giải pháp thay thế hiện đại: Chuyển sang các nguồn dữ liệu khác như Alpha Vantage, Tiingo hoặc IEX Cloud thông qua gói quantmod:

```
library(quantmod)
getSymbols("AAPL", src = "av", api.key = "YOUR_API_KEY")  # Alpha Vantage
```

```
library(tidyquant)

stock_data <- tq_get("AAPL", from = "2011-02-09", to = "2015-02-09")
print(stock_data)
```

Bảng 1.2. Dữ liệu giá cổ phiếu AAPL (2011-02-09 đến 2015-02-09)

Ngày	Mở cửa	Cao nhất	Thấp nhất	Đóng cửa	Khối lượng	Điều chỉnh
2011-02-09	12.70	12.80	12.70	12.8082	482 745 200	10.8
2011-02-10	12.80	12.90	12.40	12.7028	928 550 000	10.7
2011-02-11	12.70	12.80	12.60	12.7067	367 572 800	10.7
2011-02-14	12.70	12.80	12.70	12.8010	310 416 400	10.8
2011-02-15	12.80	12.90	12.80	12.9084	284 174 800	10.8
2011-02-16	12.90	13.00	12.90	13.0081	481 157 600	10.9
2011-02-17	12.80	12.90	12.70	12.8030	530 583 200	10.8
2011-02-18	12.80	12.80	12.50	12.8016	816 057 200	10.5
2011-02-22	12.20	12.30	12.10	12.1872	872 555 600	10.2
2011-02-23	12.10	12.30	12.10	12.2671	671 854 400	10.3
995 dòng dữ liệu bổ sung...						

Trong trường hợp cần xây dựng tập dữ liệu mới, hệ thống bắt buộc phải kết nối tới các nguồn dữ liệu chứng khoán trực tuyến. Quá trình này được thực hiện thông qua các API chuyên dụng như get.hist.quote(). Điều này đặc biệt quan trọng khi: Triển khai mô hình phân tích mới, mở rộng phạm vi nghiên cứu sang các mã chứng khoán mới và cập nhật khung thời gian phân tích.

Song song với quá trình thu thập dữ liệu, hệ thống tự động lưu trữ toàn bộ thông tin nhận được vào bộ nhớ đệm cục bộ. Dữ liệu được tổ chức khoa học dưới dạng các file CSV, phân loại rõ ràng theo từng sàn giao dịch và từng mã cổ phiếu riêng biệt. Cấu trúc thư mục được thiết kế tối ưu để đảm bảo khả năng truy xuất nhanh chóng và dễ dàng.

Giải pháp kết hợp giữa thu thập trực tuyến và lưu trữ cục bộ mang lại nhiều lợi ích thiết thực. Về hiệu năng, hệ thống giảm đáng kể thời gian truy vấn nhờ khả năng đọc trực tiếp từ bộ nhớ đệm. Về tính ổn định, các phân tích vẫn có thể tiếp tục ngay cả khi không có kết nối mạng. Đồng thời, giải pháp này còn giúp tiết kiệm đáng kể lượng request gửi tới server nguồn. Mặc dù có nhiều ưu điểm, cơ chế hiện tại vẫn tồn tại một số hạn chế cần lưu ý. Dữ liệu trong bộ nhớ đệm chỉ giới hạn trong phạm vi đã tải về trước đó. Để khắc phục, người dùng cần chủ động xóa cache và thu thập lại dữ liệu khi cần mở rộng phạm vi nghiên cứu. Ngoài ra, hệ thống cần được bổ sung cơ chế tự động kiểm tra và cập nhật dữ liệu định kỳ để đảm bảo tính chính xác.

```r
# Thiết lập thư mục chính
homeuser <- "D:/FinAnalytics"
mainDir <- file.path(homeuser, "FinAnalytics")

# Hàm đọc danh sách mã cổ phiếu
readStockSymbols <- function(filePath) {
  tryCatch({
    symbolData <- read.csv(
      file = filePath,
      header = TRUE,
      sep = "\t",
      fileEncoding = "UTF-8",
      stringsAsFactors = FALSE
    )
    result <- unique(
      na.omit(
        as.character(symbolData[, 1])
      )
    )
    return(result)
  }, error = function(e) {
    message(paste("Lỗi khi đọc file:", filePath))
    return(character(0))
  })
}

# Hàm thiết lập thư mục dữ liệu
setupDataDirectories <- function(projectName,
                                 exchanges = c("NYSE", "NASDAQ")) {
  projectDir <- file.path(mainDir, projectName)
  dir.create(projectDir, showWarnings = FALSE, recursive = TRUE)
```

```
32
33    sapply(exchanges, function(exchange) {
34      exchangeDir <- file.path(projectDir, exchange)
35      dir.create(exchangeDir, showWarnings = FALSE, recursive = TRUE)
36
37      sampleFile <- paste0(exchange, "clean.txt")
38      samplePath <- file.path(exchangeDir, sampleFile)
39
40      if (!file.exists(samplePath)) {
41        file.copy(
42          from = file.path(mainDir, sampleFile),
43          to = exchangeDir
44        )
45      }
46    })
47
48    return(projectDir)
49  }
50
51  # Hàm lấy dữ liệu giá cổ phiếu
52  fetchStockPrices <- function(symbol,
53                               startDate,
54                               endDate,
55                               apiKey,
56                               dataSource = "alpha_vantage") {
57    cacheFile <- paste0("cached_", symbol, ".rds")
58
59    if (file.exists(cacheFile)) {
60      cacheTime <- file.info(cacheFile)$mtime
61      if (as.Date(cacheTime) >= as.Date(endDate)) {
62        return(readRDS(cacheFile))
63      }
64    }
65
66    tryCatch({
67      if (dataSource == "alpha_vantage") {
68        url <- sprintf(
69          paste0(
70            "https://www.alphavantage.co/query?",
71            "function=TIME_SERIES_DAILY_ADJUSTED&",
72            "symbol=%s&outputsize=full&apikey=%s&",
73            "datatype=csv"
74          ),
```

```
        symbol, apiKey
      )
      stockData <- read.csv(url)
      prices <- stockData$adjusted_close
    } else {
      stop("Nguồn dữ liệu chưa được hỗ trợ")
    }

    saveRDS(prices, file = cacheFile)
    return(prices)

  }, error = function(e) {
    message(paste("Lỗi khi lấy dữ liệu cho",
                  symbol, ":", e$message))
    return(NA)
  })
}

# Hàm thu thập dữ liệu thị trường
collectMarketData <- function(projectName,
                              startDate,
                              endDate,
                              apiKey) {
  projectDir <- setupDataDirectories(projectName)

  nyseSymbols <- readStockSymbols(
    file.path(projectDir, "NYSE/NYSEclean.txt")
  )
  nasdaqSymbols <- readStockSymbols(
    file.path(projectDir, "NASDAQ/NASDAQclean.txt")
  )
  allSymbols <- c(nyseSymbols, nasdaqSymbols)

  dateSeq <- seq(
    from = as.Date(startDate),
    to = as.Date(endDate),
    by = "day"
  )
  priceMatrix <- matrix(
    data = NA,
    nrow = length(dateSeq),
    ncol = length(allSymbols),
    dimnames = list(
```

```
      as.character(dateSeq),
      allSymbols
    )
  )

  for (i in seq_along(allSymbols)) {
    symbol <- allSymbols[i]
    cat("Đang xử lý:", symbol,
        paste0("(", i, "/", length(allSymbols), ")"), "\n")

    prices <- fetchStockPrices(
      symbol = symbol,
      startDate = startDate,
      endDate = endDate,
      apiKey = apiKey
    )

    if (!all(is.na(prices))) {
      priceMatrix[, i] <- prices[1:nrow(priceMatrix)]
    }

    if (i %% 5 == 0) Sys.sleep(60)
  }

  validCols <- which(colSums(!is.na(priceMatrix)) > 0)
  priceMatrix <- priceMatrix[, validCols, drop = FALSE]

  saveRDS(priceMatrix,
          file = file.path(projectDir, "price_data.rds"))
  write.csv(priceMatrix,
            file = file.path(projectDir, "price_data.csv"))

  return(priceMatrix)
}

# Thiết lập API key từ biến môi trường
alphaVantageKey <- Sys.getenv("ALPHA_VANTAGE_API_KEY")

# Thu thập dữ liệu thị trường
priceData <- collectMarketData(
  projectName = "MarketAnalysis2023",
  startDate = "2020-01-01",
  endDate = "2023-12-31",
```

```
  apiKey = alphaVantageKey
)

# Kiểm tra kết quả
summary(priceData)
head(priceData[, 1:5])
hi
```

1.7 Làm sạch dữ liệu chứng khoán

Trong quá trình thu thập dữ liệu, hệ thống gặp phải một số mã chứng khoán không thể lấy được giá trị. Để xử lý vấn đề này, có thể triển khai cơ chế làm sạch dữ liệu tự động. Các mã lỗi được phân loại và lưu vào 3 file riêng biệt: badsyms.txt cho các mã không có giá, badsharpes.txt cho các mã không tính được chỉ số Sharpe, và badcors.txt cho các mã có vấn đề về hiệp phương sai.

Hàm elimSyms() đóng vai trò quan trọng trong quá trình này. Hàm sẽ đọc các file lỗi, sau đó tự động loại bỏ các mã có vấn đề khỏi danh sách chứng khoán và ma trận giá. Ví dụ, khi phát hiện mã ACO không có dữ liệu giá trên cả hệ thống và các nguồn tra cứu khác, hệ thống sẽ tự động thêm mã này vào badsyms.txt và loại bỏ khỏi các tính toán tiếp theo.

Quy trình làm sạch dữ liệu mang lại nhiều lợi ích thiết thực như: đảm bảo tính toàn vẹn của dữ liệu phân tích, giúp tiết kiệm thời gian tính toán bằng cách loại bỏ sớm các mã không khả dụng và nâng cao độ tin cậy của các chỉ số tài chính được tính toán từ hệ thống. Đây là bước quan trọng không thể thiếu trong quy trình xử lý dữ liệu chứng khoán.

1.8 Thu thập giá chứng khoán lịch sử

Việc mở rộng thành hàm getHistPrices() kết hợp với khả năng vẽ đồ thị của R sẽ lặp qua tất cả các mã chứng khoán có trọng số trong vector lab để thu thập giá lịch sử:

```
library(tseries)

getHistPrices <- function(lab, w, len, start="2013-11-29",
                          end="2014-11-28", startBck1="2013-11-28",
                          startFwd1="2013-11-27", cached=NA) {
  # Gather recent prices for all lab symbols
  D <- length(lab)
  recentPrices <- matrix(NA, nrow=len, ncol=D)
```

```
9
10  for (d in 1:D) {
11    if (w[d] > 0.0) {
12      print(paste("Fetching:", lab[d]))
13
14      # Use cached data if available
15      if (!is.na(cached) && !is.na(match(lab[d], cached))) {
16        x <- read.csv(paste("cached", lab[d], ".csv", sep=""))[,1]
17        recentPrices[, d] <- x
18      } else {
19        tryCatch({
20          x <- get.hist.quote(
21          instrument = lab[d],
22          quote="AdjClose",
23          start=start,
24          end=end,
25          retclass="zoo")
26
27          if (length(x) != len) {
28            x <- get.hist.quote(
29            instrument = lab[d],
30            quote="AdjClose",
31            start=startBck1, end=end,
32            retclass="zoo")
33          }
34          if (length(x) != len) {
35            x <- get.hist.quote(
36            instrument = lab[d],
37            quote="AdjClose",
38            start=startFwd1, end=end,
39            retclass="zoo")
40          }
41
42          # Kiểm tra số lượng dữ liệu nhận được
43          if (length(x) == len) {
44            recentPrices[, d] <- coredata(x)
45          } else {
46            warning(paste("Partial data for:", lab[d]))
47            recentPrices[1:length(x), d] <- coredata(x)
48          }
49        }, warning = function(w) {
50          print(paste("Warning for:", lab[d], " - ", w))
51        }, error = function(e) {
```

```
52            print(paste("Error for:", lab[d], " - ", e))
53          })
54        }
55      }
56    }
57
58    return(recentPrices)
59  }
60
61  # Unit test: One good ticker (PCLN), one potentially bad (UA)
62  getHistPrices(c('PCLN', 'UA'), c(.5, .5), 252)
63
```

Chương 2: Phân tích dữ liệu và đo lường rủi ro

2.1 Tính giá dựa trên hàm lợi nhuận log

Như đã nói ở chương trước, việc biểu diễn lợi nhuận dưới dạng hàm log là vô cùng quan trọng, cho phép chuyển qua lại giữa giá mô phỏng và giá thị trường mà không mất nhiều độ chính xác (không thể triệt tiêu sai số do mô phỏng chỉ là xấp xỉ và hàm log return thường không tuân theo phân phối chuẩn). Trong R có một cú pháp rất tiện lợi giúp tính log của vector: Ylogrets = diff(log(Y)) Đồng thời cũng có thể chuyển đổi linh hoạt từ lợi nhuận về giá gốc Yprices = c(Y[1], Y[1]*exp(cumsum(Ylogrets))) Ta thực hành với ví dụ đơn giản:

```
>  # nhập vào chuỗi các giá 30,29,28,28,30,32,31
>  Y = c(30,29,28,28,30,32,31)
>  Ylogrets = diff(log(Y))#tính log của vector giá
>  round(Ylogrets,4)
# in ra kết quả log return[1] -0.0339 -0.0351  0.0000  0.0690
[5]  0.0645 -0.0317
>  Yprices = c(Y[1], Y[1]*exp(cumsum(Ylogrets)))
# in ra kết quả giá ban đầu dựa vào biến đổi từ log return
>  Yprices[1] 30 29 28 28 30 32 31

```

Ta sẽ tạo hàm tính giá từ giá trị log return để phục vụ cho việc tính toán sau này:

```
# Chuỗi giá gốc
Y <- c(1.3, 1.2, 1.3, 1.4, 1.5, 1.4, 1.3, 1.4, 1.5)

# Hàm chuyển log returns thành giá
toPrices <- function(Y1, Ylogrets) {
  Yprices <- c(Y1, Y1 * exp(cumsum(Ylogrets)))
  return(Yprices)
}

# Tính lại giá từ log returns
reconstructed_Y <- toPrices(Y[1], diff(log(Y)))

# Kiểm tra độ chính xác
all_close <- sum(abs(Y - reconstructed_Y) < 1e-8) == length(Y)
```

```
15 print(all_close)
16
```

2.2 Các mô hình phân phối hỗn hợp chuẩn trong sự biến động giá

Trong phân phối chuẩn, **độ nhọn (kurtosis)** luôn cố định bằng 3, điều này phản ánh xác suất xuất hiện các giá trị cực đoan là rất thấp. Tuy nhiên, trong thực tế – đặc biệt là trong dữ liệu tài chính – các biến động lớn lại xảy ra thường xuyên hơn so với dự đoán từ phân phối chuẩn. Do đó, việc sử dụng phân phối chuẩn có thể không phù hợp để mô phỏng hành vi thị trường, vốn đặc trưng bởi những cú sốc lớn và rủi ro cao.

Một giải pháp hiệu quả cho vấn đề này là sử dụng **mô hình hỗn hợp chuẩn (normal mixture model)**, cho phép độ nhọn vượt quá giá trị 3. Mô hình này có khả năng mô tả tốt hơn hiện tượng có phần đuôi dày hơn trong phân phối xác suất – một đặc điểm phổ biến trong dữ liệu tài chính thực tế.

Mục tiêu của mô hình là mô phỏng một biến ngẫu nhiên X không tuân theo đúng phân phối chuẩn, mà là một **hỗn hợp của hai phân phối chuẩn**:

> Để mô phỏng phân phối thị trường cho một biến ngẫu nhiên hỗn hợp chuẩn X, ta sử dụng một tổ hợp của hai phân phối chuẩn: phân phối đầu tiên với biến ngẫu nhiên Y có phương sai nhỏ hơn phân phối thứ hai với biến ngẫu nhiên Z *(Hogg and Craig, 1978; Ruppert, 2011)*. Cả hai đều tuân theo phân phối chuẩn:
>
> $$Y \sim \mathcal{N}(\mu, \sigma_1^2), \quad Z \sim \mathcal{N}(\mu, \sigma_2^2), \quad \text{với } \sigma_1 < \sigma_2.$$
>
> Một biến ngẫu nhiên $U \sim \mathcal{U}(0,1)$ (phân phối đều) được sử dụng để quyết định ngưỡng chọn giữa Y và Z:
>
> $$X = \begin{cases} Y, & \text{nếu } U < 0.9 \\ Z, & \text{nếu } U \geq 0.9 \end{cases}$$

Mô hình này giả định rằng:

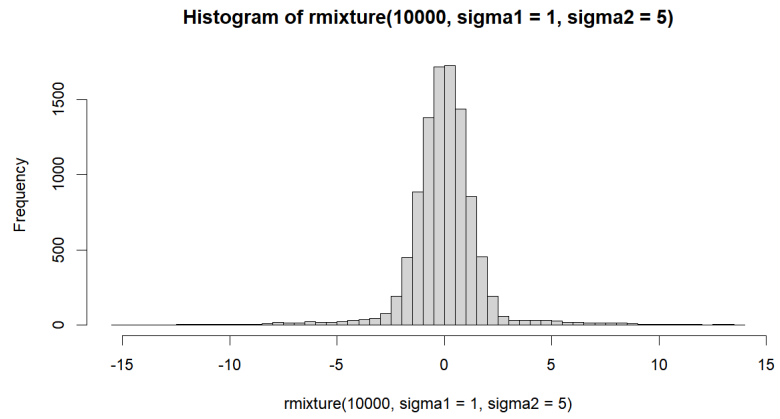
- 90% các trường hợp là "ngày bình thường" – tương ứng với dao động nhỏ, được mô phỏng bởi Y.
- 10% còn lại là "ngày đặc biệt" – tức các ngày thị trường biến động mạnh, được mô phỏng bởi Z.

Cách tiếp cận này giúp ta nắm bắt được cả hành vi thông thường và những hiện tượng cực đoan, từ đó xây dựng mô hình phản ánh sát thực tế hơn. Ta sẽ mô tả nó trong R:

```r
# Hàm sinh các số ngẫu nhiên theo phân phối hỗn hợp chuẩn
rmixture <- function(N, sigma1, sigma2 = 0, thresh = 0.9) {
  # Nếu sigma2 = 0, hàm hoạt động như phân phối chuẩn với độ lệch chuẩn sigma1

  variates <- vector(length = N)
  U <- runif(N)

  for (i in 1:N) {
    # Mặc định sinh giá trị từ phân phối chuẩn với sigma1
    variates[i] <- rnorm(1, mean = 0, sd = sigma1)
  }

  if (sigma2 != 0) {
    for (i in 1:N) {
      # Nếu U[i] >= ngưỡng, thay thế bằng giá trị từ phân phối chuẩn có sigma2
      if (U[i] >= thresh) {
        variates[i] <- rnorm(1, mean = 0, sd = sigma2)
      }
    }
  }

  return(variates)
}

# Vẽ histogram của 10,000 mẫu từ mô hình hỗn hợp chuẩn.
hist(rmixture(10000, sigma1 = 1, sigma2 = 5), breaks = 50)
```

Khi đã có trong tay mô hình phân phối hỗn hợp chuẩn, chúng ta sẽ tiến hành điều chỉnh giả định ban đầu, sao cho log-return không còn tuân theo phân phối chuẩn, mà tuân theo phân phối hỗn hợp chuẩn. Mục tiêu của sự điều chỉnh này là để mô hình có thể mô phỏng các hiện tượng cực đoan xuất hiện ở phần đuôi của phân phối - điều thường xảy ra trên thị trường tài chính nhưng bị đánh giá thấp trong các mô hình truyền thống.

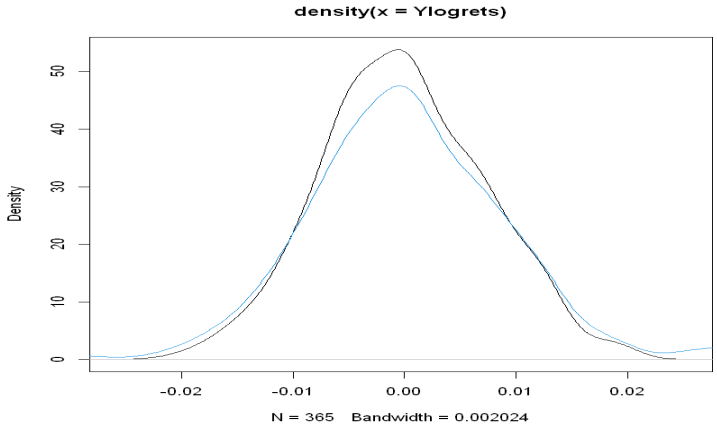
Tiếp theo, chúng ta sẽ xây dựng hàm `simPricePath()`, với chức năng tạo ra đường giá cổ phiếu bắt đầu từ một mức giá ban đầu nhất định.



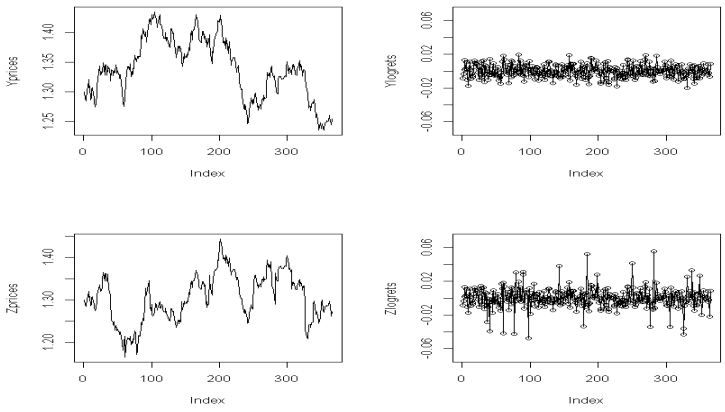
Hình 2.1. Histogram của mô hình phân phối hỗn hợp chuẩn với $\sigma_1 = 1$, $\sigma_2 = 5$

```
1
2  # Hàm tạo đường giá từ phân phối chuẩn hoặc phân phối hỗn hợp chuẩn
3  simPricePath <- function(initPrice, N, seed, sigma1 = 0.05,
4                           sigma2 = 0, thresh = 0.9) {
5    # Thiết lập seed để tái hiện kết quả
6    set.seed(seed)
7
8    # Sinh log-returns từ hàm rmixture (chuẩn hoặc hỗn hợp chuẩn)
9    Xlogrets <- rmixture(N, sigma1, sigma2, thresh = thresh)
10
11   # Chuyển đổi log-returns thành chuỗi giá
12   Xprices <- toPrices(initPrice, Xlogrets)
13
14   # Trả về danh sách gồm chuỗi giá và log-returns
15   list(Xprices, c(Xlogrets))
16 }
17
18 # ======= Kiểm thử hàm với dữ liệu giả =======
19
20 # Khởi tạo tham số
21 seed <- 26                    # Seed cho random generator
22 sigma1 <- 0.007157            # Độ lệch chuẩn nhỏ
23 N <- 365                      # Số ngày mô phỏng
24 par(mfrow = c(2, 2))          # Chia vùng vẽ thành 2x2
25 maxy <- 10 * sigma1           # Cận trên cho trục y của log-returns
26
27 # --- Mô hình chỉ dùng phân phối chuẩn (không hỗn hợp) ---
28 Y <- simPricePath(1.3, N = N, seed = seed, sigma1 = sigma1)
29 Yprices <- Y[[1]]             # Chuỗi giá
30 Ylogrets <- Y[[2]]            # Log-returns
```

```r

# Vẽ đường giá và log-returns
plot(Yprices, type = 'l', main = "Giá - Mô hình chuẩn")
plot(Ylogrets, type = 'l', ylim = c(-maxy, maxy), main = "Log-return
↪  chuẩn")
points(Ylogrets)

# --- Mô hình hỗn hợp chuẩn (trường hợp có biến động mạnh) ---
Z <- simPricePath(1.3, N = N,
                  seed = seed, sigma1 = sigma1,
                  sigma2 = 4 * sigma1)
Zprices <- Z[[1]]
Zlogrets <- Z[[2]]

# Vẽ đường giá và log-returns của mô hình hỗn hợp
plot(Zprices, type = 'l',
     main = "Giá - Mô hình hỗn hợp")
plot(Zlogrets, type = 'l',
     ylim = c(-maxy, maxy),
     main = "Log-return hỗn hợp")
points(Zlogrets)

# So sánh độ lệch chuẩn giữa hai mô hình ---
sd(Ylogrets)   # Độ lệch chuẩn log-return mô hình chuẩn
sd(Zlogrets)   # Độ lệch chuẩn log-return mô hình hỗn hợp

# --- Vẽ đồ thị mật độ phân phối log-return ---
par(mfrow = c(1, 1))  # Trở lại bố cục vẽ 1 biểu đồ
plot(density(Ylogrets), main = "So sánh mật độ phân phối", col = "black")
lines(density(Zlogrets), col = 4)  # Mô hình hỗn hợp vẽ màu xanh
```



Hình 2.2. Biểu đồ mô phỏng giá theo mô hình hỗn hợp chuẩn



Hình 2.3. So sánh đường giá và log-returns của mô hình hỗn hợp và không hỗn hợp

Tại hình 2.2 ta có thể thấy rằng đuôi bên phải dày hơn của Zlogrets phản ánh sự biến động lớn sau bước thời gian thứ 50 trong hình 2.3. Khi so sánh độ nhọn của mô hình không hỗn hợp (non-mixture) và mô hình hỗn hợp (mixture), ta có thể thấy từ kết quả đầu ra rằng độ nhọn (kurtosis) của mô hình hỗn hợp lớn hơn khá nhiều so với mô hình không hỗn hợp. Trong ngôn ngữ R đẻ tính độ nhọn cho Zlogrets và Ylogrets thì ta có thể sử dụng hàm kurtosis() hoặc tính trực tiếp bằng công thức:

$$\text{Kurt}(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right]$$

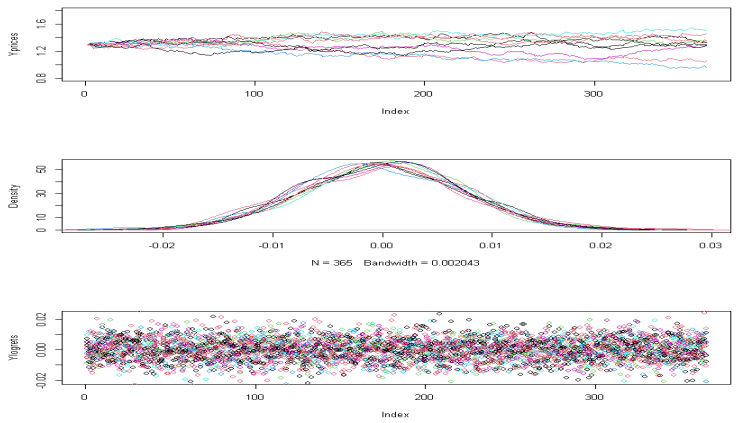
```
1
2  library(moments)
3  # Tính kurtosis thủ công cho log returns từ mô hình chuẩn: Ylogrets
4  KurtYlogrets = length(Ylogrets)^(-1) *
5                 sd(Ylogrets)^(-4) *
6                 sum((Ylogrets - mean(Ylogrets))^4)
7  KurtYlogrets
8  [1] 3.393385
9  # Tính kurtosis bằng hàm có sẵn kurtosis(Ylogrets)
10 [1] 3.412056
11 # Tính kurtosis thủ công cho log returns từ mô hình hỗn hợp: Zlogrets
12 KurtZlogrets = length(Zlogrets)^(-1) *
13                sd(Zlogrets)^(-4) *
14                sum((Zlogrets - mean(Zlogrets))^4)
15 KurtZlogrets
16 [1] 12.09176
17 kurtosis(Zlogrets)
18 [1] 12.15829
```

Dù tính bằng 2 cách nhưng ta thấy rằng kurtosis của phân phối hỗn hợp lớn hơn

khá nhiều so với kurtosis của phân phối chuẩn. Tiếp theo, ta sẽ mô phỏng giá trong 365 ngày của EUR trên USD với giá khởi điểm là 1.3000$ bằng cả mô hình chuẩn và hỗn hợp chuẩn.

```
# Multiple paths
library(moments)
par(mfrow = c(3, 1))
par(bg = "white")
mapToCol <- function(d) {
  if (d == 7) 1
  else if (d == 8) 2
  else if (d == 15) 3
  else if (d == 23) 4
  else d
}

allYlogrets <- matrix(nrow = 10, ncol = N)

for (path in 1:10) {
  Y <- simPricePath(1.3, N, seed = path, sigma1 = 0.007157)
  Yprices <- Y[[1]]
  Ylogrets <- Y[[2]]

  if (path == 1) {
    plot(Yprices, type = 'l', ylim = c(0.8, 1.8))
  } else {
    lines(Yprices, col = mapToCol(path))
  }

  allYlogrets[path, ] <- Ylogrets
}

for (path in 1:10) {
  if (path == 1) {
    plot(density(allYlogrets[path, ]), main = "")
  } else {
    lines(density(allYlogrets[path, ]), col = mapToCol(path))
  }
}

mean_Ylogrets <- mean(Ylogrets)
sd_Ylogrets <- sd(Ylogrets)

```

```
for (path in 1:10) {
  if (path == 1) {
    plot(allYlogrets[path, ], ylab = 'Ylogrets')
  } else {
    points(allYlogrets[path, ], col = mapToCol(path))
  }
}

# In kết quả trung bình và độ lệch chuẩn của log returns
mean_Ylogrets
sd_Ylogrets
```

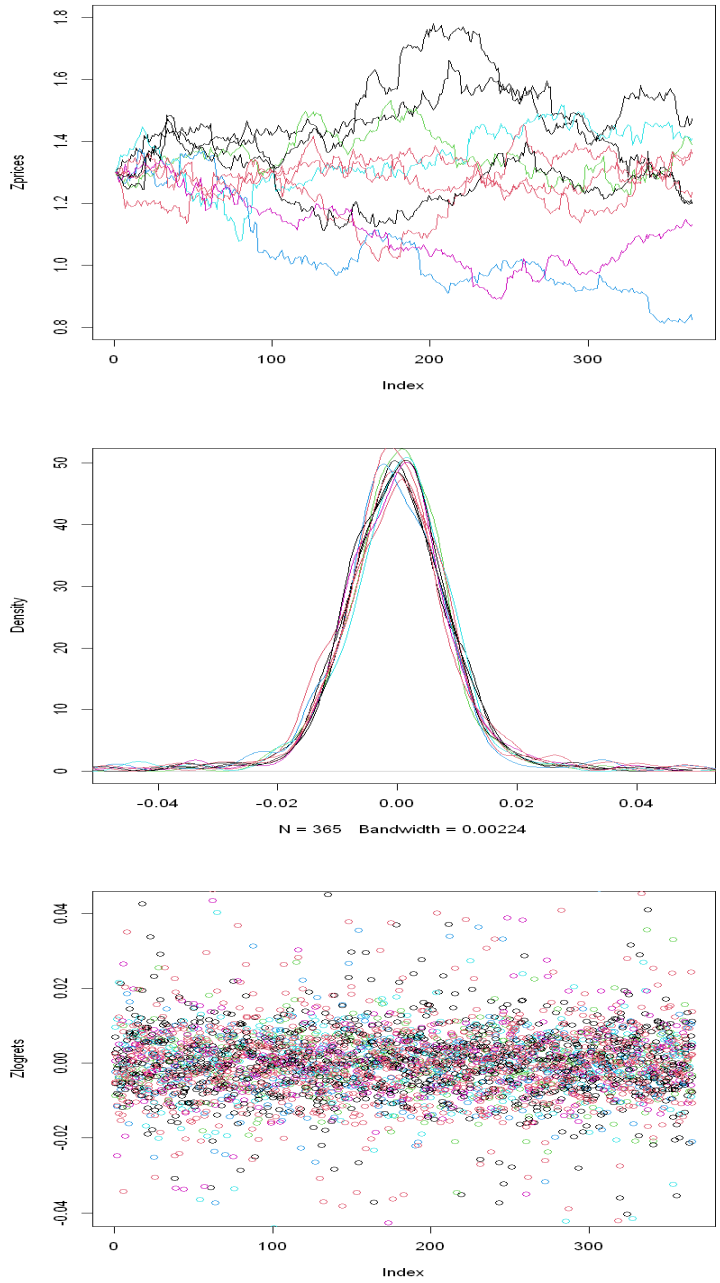


Hình 2.4. Mô phỏng giá trong 365 ngày của EUR trên USD với giá khởi điểm là 1.3000$ bằng mô hình chuẩn.

Với mô hình hỗn hợp chuẩn, giá trị $sigma1 = 0.007157$ vì nó tương ứng với độ biến động thường niên là 13.7$, một giá trị phổ biến. Trong đoạn mã hỗn hợp bên dưới, chúng ta gấp 4 lần sigma1 (tức là $4sigma$) với xác suất 10% (10 percent of the time).

```
# Mixture model simulation
allZlogrets = matrix(nrow=10, ncol=N)
par(bg = "white")
for (path in 1:10) {
  Z <- simPricePath(1.3, N, seed=path, sigma1=.007157, sigma2=4 * .007157)
  Zprices <- Z[[1]]
  Zlogrets <- Z[[2]]

  if (path == 1) {
    plot(Zprices, type='l', ylim=c(.8, 1.8))
  } else {
```

```
    lines(Zprices, col=mapToCol(path))
  }

  allZlogrets[path, ] = Zlogrets
}

for (path in 1:10) {
  if (path == 1) {
    plot(density(allZlogrets[path, ]), main="")
  } else {
    lines(density(allZlogrets[path, ]), col=mapToCol(path))
  }
}

mean(Zlogrets)
sd(Zlogrets)

for (path in 1:10) {
  if (path == 1) {
    plot(allZlogrets[path, ], ylab='Zlogrets')
  } else {
    points(allZlogrets[path, ], col=mapToCol(path))
  }
}
```



Hình 2.5. Mô phỏng giá trong 365 ngày của EUR trên USD với giá khởi điểm là 1.3000$ bằng mô hình hỗn hợp.

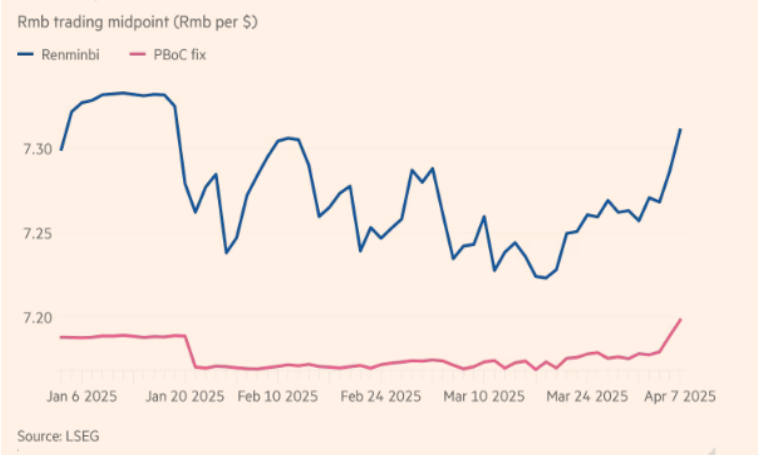
```
> sd(Ylogrets)
[1] 0.007559591
> sd(Zlogrets)
[1] 0.01017884
> sd(Zlogrets)/sd(Ylogrets)
[1] 1.34648
```

Từ kết quả từ đoạn code trên ta có thể thấy được độ biến động của mô hình chuẩn

là 0.007559591 trong khi mô hình hỗn hợp chuẩn là 0.01017884 và biến động của mô hình hỗn hợp chuẩn cao hơn mô hình chuẩn 34,65%

2.3 Biến động tỉ giá đột ngột của đồng Nhân dân tệ tháng 4/2025

Vào ngày 10/4/2025, Ngân hàng Nhân dân Trung Quốc (PBOC) đã hạ tỷ giá tham chiếu xuống mức 7,2092 CNY/USD, đánh dấu ngày thứ 6 liên tiếp điều chỉnh giảm. Đây là mức thấp nhất kể từ tháng 9/2023. Đồng thời, đồng Nhân dân tệ giao dịch trong nước giảm xuống 7,3518 CNY/USD, mức thấp nhất trong gần hai thập kỷ. Nguyên nhân chính là do căng thẳng thương mại, Mỹ tuyên bố áp thuế 125% đối với hàng hóa Trung Quốc, khiến Bắc Kinh phản ứng bằng cách hạ giá đồng nội tệ để hỗ trợ xuất khẩu. Ngoài ra, yếu tố tâm lý thị trường dũng tác động không nhỏ do các nhà đầu tư lo ngại về một cuộc chiến thương mại leo thang, dẫn đến việc bán tháo đồng Nhân dân tệ và tìm kiếm tài sản an toàn như USD. Sự giảm giá nhanh chóng của CNY gây ra biến động lớn trên thị trường ngoại hối, ảnh hưởng đến các doanh nghiệp và nhà đầu tư có liên quan đến đồng tiền này. Ngoài ra, các doanh nghiệp nhập khẩu từ Trung Quốc phải đối mặt với chi phí cao hơn do tỷ giá biến động, ảnh hưởng đến lợi nhuận và kế hoạch tài chính.



Hình 2.6. Diễn biến của đồng Nhân dân tệ và tỷ giá trung tâm do PBOC ấn định

Theo các tính toán dựa trên mô hình phân phối chuẩn, ba độ lệch chuẩn (3σ) đã là một biến động hiếm gặp. Tuy nhiên, **mức tăng của đồng USD so với CNY trong sự kiện này tương đương với hơn 20 lần sigma**, gây ra tổn thất đáng kể cho các nhà đầu tư nắm giữ tài sản định danh bằng CNY hoặc các quỹ đầu tư chênh lệch tỷ giá.

Để mô phỏng sự kiện này, chúng tôi sử dụng một mô hình hỗn hợp ba pha, tương tự như hàm `tmixture()`. Các pha trong mô hình được định nghĩa như sau:

- **Pha đầu tiên**: Trước khi sự kiện xảy ra, tỷ giá được giả định tuân theo phân

phối chuẩn $\mathcal{N}(0, \sigma_1)$ với phương sai nhỏ. Giai đoạn này phản ánh chính sách ổn định tỷ giá của Ngân hàng Trung ương Trung Quốc.

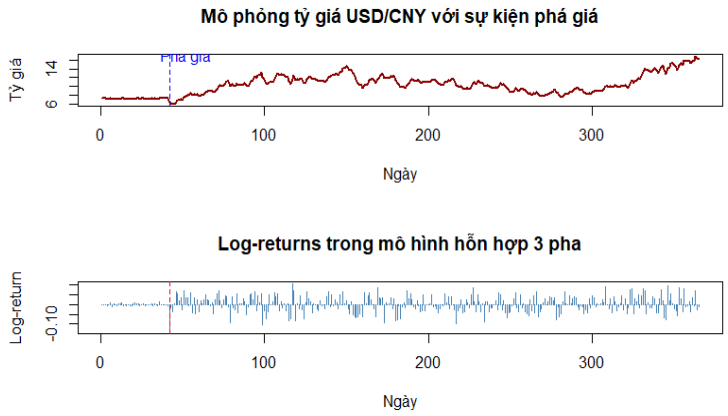
- **Pha sự kiện**: Khi biến nhị phân $B = 1$ (đại diện cho thời điểm phá giá), tỷ giá trải qua một bước nhảy lớn, được mô hình hóa bằng phân phối chuẩn có độ lệch chuẩn $\sigma_2 = 20\sigma_1$.
- **Pha sau sự kiện**: Sau bước nhảy, tỷ giá tiếp tục biến động mạnh hơn bình thường, mô phỏng bằng $\sigma_3 = 8\sigma_1$.

Mô hình có thể được biểu diễn như sau:

$$Y = \begin{cases} Z \sim \mathcal{N}(0, \sigma_1), & \text{khi } B = 0 \\ U \sim \mathcal{N}(0, \sigma_2 = 20\sigma_1), & \text{khi } B = 1 \text{ (tại bước nhảy)} \\ V \sim \mathcal{N}(0, \sigma_3 = 8\sigma_1), & \text{sau sự kiện} \end{cases}$$

```r
set.seed(2025)  # đảm bảo tái lập kết quả

# Tham số mô hình
n <- 365                    # số ngày mô phỏng
sigma1 <- 0.005             # độ lệch chuẩn thường
sigma2 <- 20 * sigma1       # độ lệch chuẩn tại bước nhảy
sigma3 <- 8 * sigma1        # độ lệch chuẩn sau sự kiện

jump_day <- sample(30:60, 1)  # ngày phá giá xảy ra ngẫu nhiên

# Khởi tạo vector log-returns
logrets <- numeric(n)

# Sinh log-returns theo 3 pha
for (i in 1:n) {
  if (i < jump_day) {
    logrets[i] <- rnorm(1, mean = 0, sd = sigma1)
  } else if (i == jump_day) {
    logrets[i] <- rnorm(1, mean = 0, sd = sigma2)
  } else {
    logrets[i] <- rnorm(1, mean = 0, sd = sigma3)
  }
}

# Tạo chuỗi giá từ log-returns
price <- numeric(n)
price[1] <- 7.1452  # giá khởi điểm của USD/CNY
```

```r

for (i in 2:n) {
  price[i] <- price[i-1] * exp(logrets[i])
}

# Vẽ biểu đồ
par(mfrow = c(2, 1))

plot(price, type = "l", col = "darkred", lwd = 2,
     main = "Mô phỏng tỷ giá USD/CNY với sự kiện phá giá",
     ylab = "Tỷ giá", xlab = "Ngày")

abline(v = jump_day, col = "blue", lty = 2)
text(jump_day + 10, max(price), "Phá giá", col = "blue")

plot(logrets, type = "h", col = "steelblue",
     main = "Log-returns trong mô hình hỗn hợp 3 pha",
     ylab = "Log-return", xlab = "Ngày")

abline(v = jump_day, col = "red", lty = 2)

```



Hình 2.7. Mô phỏng tỷ giá USD/CNY và Log-returns với sự kiện phá giá

Khi xem xét rủi ro của việc giao dịch và đầu tư vào thị trường ngoại hối (FX), các chuyển động cực đoan như vậy luôn có khả năng xảy ra. Nếu một người không tin rằng những biến động lớn như vậy có thể xảy ra trong thị trường FX, thì việc chứng kiến sự kiện này sẽ khiến họ thay đổi suy nghĩ.

Việc nghiên cứu các sự kiện như vậy từ góc độ phân tích dữ liệu là rất quan trọng, nhằm xây dựng các mô hình rủi ro có thể tính đến khả năng xảy ra các sự kiện cực đoan.

Chương 3: Tỉ số Sharpe

3.1 Công thức tỉ số Sharpe

Tỉ số Sharpe (Sharpe Ratio) là một chỉ số tài chính dùng để đánh giá hiệu quả đầu tư so với mức độ rủi ro mà nhà đầu tư phải chấp nhận. Tỷ số Sharpe được định nghĩa là:

$$\text{Sharpe Ratio} = \frac{E(R_p) - R_f}{\sigma_p}$$

Trong đó:

- $E(R_P)$: Lợi suất kỳ vọng của danh mục P,
- R_f: Lãi suất phi rủi ro,
- σ_P: Độ lệch chuẩn (biến động) của danh mục P.

Ta có thể xem biến động (volatility) là một thước đo rủi ro. Với mỗi đơn vị rủi ro, tỷ số Sharpe thể hiện phần lợi suất vượt trội (excess return) mà nhà đầu tư nhận được so với mức sinh lời an toàn. Sharpe Ratio càng cao thì đầu tư càng hiệu quả (tức là mỗi đơn vị rủi ro mang lại lợi suất vượt trội hơn).

3.2 Khoảng thời gian và việc quy đổi theo năm (Annualizing)

Khi phân tích hiệu suất đầu tư, việc xác định rõ khoảng thời gian đo lường lợi suất là rất quan trọng. Lợi suất có thể được tính theo ngày, tuần, tháng hoặc năm, tùy theo mục đích phân tích. Tuy nhiên, để so sánh giữa các khoản đầu tư hoặc để đưa ra quyết định đầu tư, các lợi suất này thường được quy đổi về cùng một đơn vị thời gian — phổ biến nhất là lợi suất hàng năm (*annualized return*).

Tương tự, độ lệch chuẩn của lợi suất (còn gọi là độ biến động — *volatility*) cũng cần được quy đổi theo năm để đảm bảo tính nhất quán trong các chỉ số như Tỷ số Sharpe.

Giả sử $R_{\text{ngày}}$ là lợi suất kỳ vọng hàng ngày, thì lợi suất kỳ vọng hàng năm được tính theo công thức:

$$E(R_{\text{năm}}) = E(R_{\text{ngày}}) \times N$$

Trong đó:

- $E(R_{\text{năm}})$: Lợi suất kỳ vọng hàng năm,

- $E(R_{\text{ngày}})$: Lợi suất kỳ vọng hàng ngày,
- N: Số ngày giao dịch trong năm (thường là 252 ngày).

Tương tự, độ lệch chuẩn hàng năm ($\sigma_{\text{năm}}$) được tính như sau:

$$\sigma_{\text{năm}} = \sigma_{\text{ngày}} \times \sqrt{N}$$

Trong đó:

- $\sigma_{\text{năm}}$: Độ lệch chuẩn hàng năm,
- $\sigma_{\text{ngày}}$: Độ lệch chuẩn hàng ngày,
- N: Số ngày giao dịch trong năm.

Lưu ý rằng các đơn vị thời gian trong tính toán cần được thống nhất. Nếu sử dụng lợi suất và độ lệch chuẩn theo năm, thì các chỉ số tài chính liên quan như Tỷ số Sharpe cũng phải sử dụng dữ liệu quy đổi theo năm để đảm bảo tính chính xác.

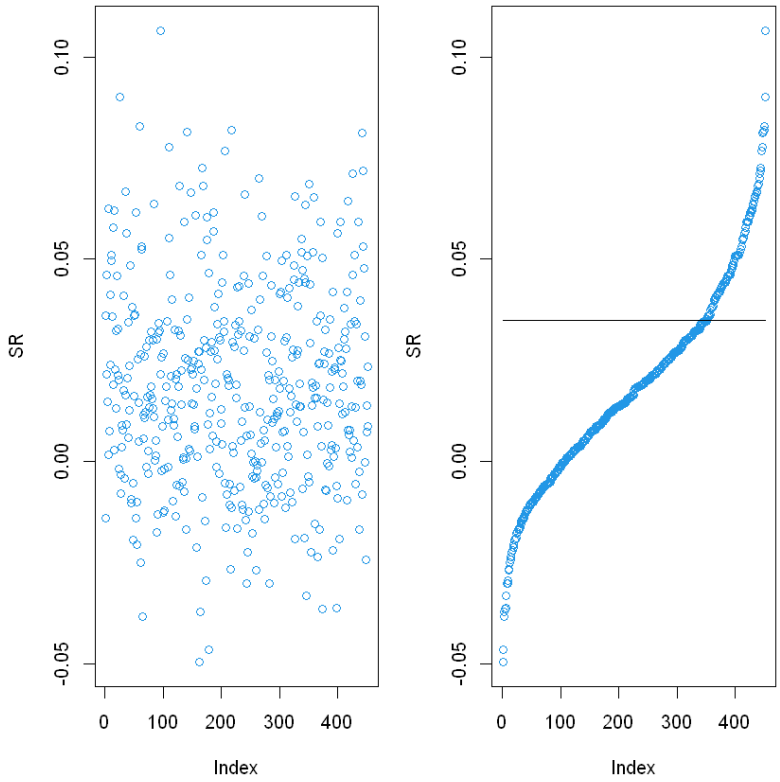
3.3 Xếp hạng các ứng viên đầu tư

Khi đối mặt với nhiều lựa chọn đầu tư, nhà đầu tư cần có một tiêu chí khách quan để đánh giá và so sánh hiệu suất của từng khoản đầu tư. Tỷ số Sharpe cung cấp một phương pháp hiệu quả để thực hiện điều này bằng cách so sánh lợi suất vượt trội so với mức rủi ro (đo bằng độ lệch chuẩn), chúng ta có thể xác định được những tài sản mang lại phần thưởng đầu tư tốt nhất so với rủi ro mà nhà đầu tư phải chịu. Giả sử ta có một tập hợp các tài sản hoặc danh mục đầu tư — mỗi tài sản đều có lợi suất kỳ vọng và rủi ro riêng. Khi đó, tỷ số Sharpe được tính cho từng tài sản sẽ cho thấy "hiệu suất điều chỉnh theo rủi ro" (risk-adjusted performance) của chúng. Trước khi tối ưu danh mục đầu tư, ta sẽ tính tỉ số Sharpe cho từng mã cổ phiếu rồi thực hiện "cắt tỉa" những cổ phiếu dưới mức cố định.

```

library(quantmod)
library(TTR)
# Định nghĩa hàm tính log-returns
findR <- function(prices) {
  returns <- diff(log(prices))  # Tính log-returns
  return(returns)
}
```

```r

# Hàm tính lợi nhuận trung bình, ma trận hiệp phương sai, độ lệch chuẩn
findCovMat <- function(returns) {
  meanv <- colMeans(returns, na.rm = TRUE)  # Lợi nhuận trung bình
  cov_mat <- cov(returns, use="complete.obs")  # Ma trận hiệp phương sai
  diag_cov_mat <- diag(cov_mat)  # Giá trị trên đường chéo
  sdevv <- sqrt(diag_cov_mat)  # Độ lệch chuẩn
  return(list(meanv, cov_mat, diag_cov_mat, sdevv))
}

# Định nghĩa hàm lọc theo Sharpe Ratio
pruneBySharpe <- function(prices, lab, meanv, sdevv, threshSR, mufree=0) {
  par(mar=c(4,4,1,1))
  par(mfrow=c(1,2))
  indepSharpes <- (meanv - mufree) / sdevv
  len <- length(indepSharpes)

  par(bg = "white")
  plot(indepSharpes, ylab="SR", col=4)
  plot(sort(indepSharpes), ylab="SR", col=4)
  lines(1:len, rep(threshSR, len))

  indHighSharpes <- (indepSharpes > threshSR)

  # Xử lý giá trị NA
  indHighSharpes[is.na(indHighSharpes)] <- FALSE

  len <- dim(prices)[1]
  wid <- dim(prices)[2]
  smallerSz <- sum(indHighSharpes)

  newPrices <- matrix(0, nrow=len, ncol=smallerSz)
  newLab <- vector(length=smallerSz)

  e <- 1
  for (d in 1:wid) {
    if (indHighSharpes[d]) {
      print(paste("e", e))
      newPrices[, e] <- prices[, d]
      newLab[e] <- lab[d]
      e <- e + 1
    }
  }
```

```
52
53    print("completed Sharpe pruning")
54    return(list(newPrices, newLab, indepSharpes))
55  }
56
57  # Unit test:
58  library(huge)
59  data(stockdata)
60  D <- length(stockdata$data[1,])
61  p <- stockdata$data[, 1:D]
62  l <- stockdata$info[1:D, 1]
63
64  # Gọi các hàm đã định nghĩa
65  r <- findR(p)
66  res <- findCovMat(r)
67  meanv <- res[[1]]
68  cov_mat <- res[[2]]
69  diag_cov_mat <- res[[3]]
70  sdevv <- res[[4]]
71
72  # Lọc cổ phiếu theo Sharpe Ratio
73  res <- pruneBySharpe(p, l, meanv, sdevv, .035)
74  p <- res[[1]]
75  l <- res[[2]]
76  indepSharpes <- res[[3]]
77
78  # Kiểm tra số lượng cổ phiếu sau khi lọc
79  print(paste('D =', length(l)))
```



Hình 3.1. Biểu đồ Sharpe Ratio từng cổ phiếu và sắp xếp theo thứ tự tăng dần (đường ngang thể hiện ngưỡng lọc)

Sau khi thực thi hàm `pruneBySharpe()`, tập hợp các chứng khoán ứng viên sẽ được tinh gọn đáng kể. Hàm này đánh giá độc lập từng chứng khoán dựa trên tỉ số Sharpe, sau đó loại bỏ các mã có hiệu suất điều chỉnh theo rủi ro thấp hơn ngưỡng đã định (`threshSR`). Kết quả là một bản sao mới của ma trận giá (`prices`) chỉ còn chứa các chứng khoán vượt qua tiêu chí lọc.

Nếu ban đầu, chúng ta sử dụng hàm `findCovMat()` để tính toán ma trận hiệp phương sai từ chuỗi log returns gốc, thì sau bước lọc, một ma trận hiệp phương sai mới — ký hiệu là Σ' — cũng cần được xây dựng lại, dựa trên tập dữ liệu đã được tinh giảm. Việc này được thực hiện thông qua chính hàm `findCovMat()`.

Để đảm bảo độ tin cậy của dữ liệu, một bước kiểm tra bổ sung thông qua một hàm `isnaCheckCovMat()` sẽ được áp dụng để phát hiện và xử lý các giá trị thiếu (NA) có thể phát sinh trong quá trình tính toán.

Trong thực tế, bộ lọc Sharpe Ratio thường loại bỏ một phần không nhỏ các chứng khoán ban đầu, tùy thuộc vào mức ngưỡng mà người phân tích đặt ra. Do đó, việc tái tính toán ma trận hiệp phương sai và kiểm tra chất lượng dữ liệu là bước không thể thiếu nhằm đảm bảo độ chính xác cho các bước tối ưu danh mục tiếp theo.

```
1 isnaCheckCovMat <- function(R) {
2   cor_mat <- cor(R)
```

```
  print("Checking correlation data.")
  isNACor <- FALSE

  for (d in 1:D) {  # Kiểm tra một hàng để phát hiện dữ liệu lỗi
    if (is.na(cor_mat[d, 1])) {
      print(paste("NA for", d, lab[d]))
      cat(lab[d], file = "badsyms.txt", append = TRUE, sep = "\n")
      isNACor <- TRUE
    }
  }

  if (isNACor) stop("NA Cors recorded in badsyms.txt")

  diag_cov_mat <- diag(cov_mat)
  sdevv <- sqrt(diag_cov_mat)

  return(sdevv)
}

sdevv <- isnaCheckCovMat(r)

```

"Checking correlation data."

Một việc vô cùng quan trọng nữa đó là phát hiện những ma trận không khả thi hoặc các cổ phiếu có mối tương quan quá cao (một cổ phiếu có hai mã khác nhau), dẫn đến ma trận hiệp phương sai không thể sử dụng được. Đây là tình trạng khi dữ liệu có thể bị lỗi hoặc không ổn định trong việc tính toán các trọng số danh mục đầu tư.

```

checkDeterminant <- function(prices, R, lab, isSubDir = TRUE) {
  # Kiểm tra và xây dựng dần ma trận tương quan
  # để phát hiện ma trận suy biến
  subdirStr = ifelse(isSubDir, "/NYSE", "")
  D <- dim(R)[2]  # Số lượng biến (cột) trong R

  # Kiểm tra tương quan giữa từng biến với biến thứ 8
  scalar_cov = vector(length = D)  # Tạo vector để lưu tương quan
  for (d in 1:D) {
    scalar_cov[d] = cor(R[, d], R[, 8])
    print(paste(d, round(scalar_cov[d], 6)))
```

```
  }

  # Tìm các cặp biến có giá trị giá giống nhau tại thời điểm thứ 20
  for (d in 1:(D - 1)) {
    if (prices[20, d] == prices[20, d + 1]) {
    # Nếu giá tại thời điểm 20 giống nhau giữa hai biến liên tiếp
      print("adding to badcors.txt")  # In thông báo
      print(lab[d:(d + 1)])  # In nhãn tên hai biến

      # Ghi nhãn biến có vấn đề vào file badcors.txt
      system(paste("echo", lab[d], ">>",
             file.path(homeuser, "FinAnalytics",
                       dir, subdirStr, "badcors.txt")))
    }
  }

  # Kiểm tra định thức của các ma trận con từ 5 biến trở đi
  for (d in 5:D) {
    Rsmall = R[, 1:d]  # Chọn các cột đầu tiên từ 1 đến d
    small_cov_mat = cor(Rsmall)  # Tính ma trận tương quan
    deter = det(small_cov_mat)  # Tính định thức

    print(paste(d, lab[d], deter, dim(Rsmall)[2]))

    if (deter <= 0.0) {  # Nếu định thức không dương (ma trận suy biến)
      # Ghi thông tin biến gây lỗi vào file badcors.txt
      system(paste("echo", lab[d],
             ">>", file.path(homeuser,
                             "FinAnalytics", dir,
                             subdirStr, "badcors.txt")))
      stop(paste(d, lab[d], "det =", deter))
    }
  }
}
# Gọi hàm kiểm tra
checkDeterminant(p, r, l)

```

Kết quả của đoạn code cho ta thấy trong dữ liệu đã cho (trong t.series), không có mã chứng khoán nào bị trùng lặp.

3.4 Gói Quantmod trong R

Gói `quantmod` (viết tắt của *Quantitative Modeling*) là một công cụ mạnh mẽ và phổ biến trong hệ sinh thái R, được thiết kế nhằm hỗ trợ các nhà phân tích tài chính và nhà đầu tư trong việc truy xuất, phân tích và trực quan hóa dữ liệu tài chính, cũng như xây dựng các mô hình định lượng phục vụ cho dự báo, đầu tư và quản trị rủi ro.

Mục tiêu chính của `quantmod` là cung cấp một môi trường thuận tiện để xây dựng các mô hình định lượng tài chính một cách hiệu quả, đồng thời hỗ trợ quy trình nghiên cứu, kiểm định và triển khai các chiến lược đầu tư thông qua dữ liệu thời gian thực hoặc dữ liệu lịch sử.

Tính năng nổi bật

- **Tự động hóa quá trình lấy dữ liệu thị trường tài chính:** `quantmod` hỗ trợ truy xuất dữ liệu giá chứng khoán và dữ liệu kinh tế vĩ mô từ nhiều nguồn uy tín như Yahoo Finance, Google Finance (trước đây), FRED (Federal Reserve Economic Data), Oanda, CSV, RData, MySQL,...
- **Xử lý và phân tích chuỗi thời gian tài chính:** Cho phép phân tích chuỗi thời gian với các hàm tiện ích mạnh như `getSymbols`, `periodReturn`, `lag`, `merge`, `reclass`, v.v.
- **Biểu đồ và trực quan hóa dữ liệu kỹ thuật:** `quantmod` có thể tạo ra các biểu đồ nến (candlestick chart), biểu đồ OHLC, đồng thời hỗ trợ tích hợp các chỉ báo kỹ thuật như MACD, RSI, Bollinger Bands, SMA/EMA, v.v.
- **Tính toán các chỉ báo kỹ thuật phổ biến:** Với sự tích hợp từ gói phụ trợ `TTR`, `quantmod` cho phép người dùng dễ dàng thêm các chỉ báo kỹ thuật vào phân tích.
- **Hỗ trợ xây dựng mô hình định lượng:** Gói này rất phù hợp cho việc phát triển các chiến lược giao dịch, kiểm thử ngược, và mô hình hóa rủi ro/tài sản.

```
1 library(quantmod)
2 symbol <- "GOOG"
3 getSymbols(symbol, src = "yahoo", from = "2020-01-01", to = "2024-01-01")
4 head(GOOG)  # Xem dữ liệu đầu tiên
```

Day	Open	High	Low	Close	Volume	Adjusted
2020-01-02	67.0775	68.4070	67.0775	68.3685	28,132,000	68.04620
2020-01-03	67.3930	68.6250	67.2772	68.0330	23,728,000	67.71227
2020-01-06	67.5000	69.8250	67.5000	69.7105	34,646,000	69.38188
2020-01-07	69.8970	70.1495	69.5190	69.6670	30,054,000	69.33858
2020-01-08	69.6040	70.5790	69.5420	70.2160	30,560,000	69.88500
2020-01-09	71.0285	71.3665	70.5135	70.9915	30,018,000	70.65683

Chúng ta có thể truy vấn dữ liệu tài chính của bất kỳ công ty niêm yết nào bằng cách sử dụng mã cổ phiếu (ví dụ: "GOOG" cho Google) và hàm `getFinancials()`. Lệnh này trả về một đối tượng chứa báo cáo tài chính, trong đó có thể truy xuất báo cáo kết quả kinh doanh hàng năm (`IS, A`) và lấy các chỉ số như EPS pha loãng chuẩn hóa. Từ chuỗi EPS này, ta dễ dàng tính được tốc độ tăng trưởng EPS, một chỉ số lợi nhuận không phụ thuộc vào giá cổ phiếu.

Việc chuẩn bị thư mục `MVO4` với các tệp *cache* dữ liệu giá yêu cầu xác định danh sách các mã cổ phiếu, gán nhãn và thiết lập khoảng thời gian để tải và lưu trữ dữ liệu lịch sử vào thư mục này, giúp quá trình xử lý sau đó nhanh hơn và hiệu quả hơn.

```
library(tseries)
library(quantmod)
readSubDirs <- function(dir) {
  dirs <- list.dirs(dir, recursive = FALSE)  # Lấy danh sách thư mục con
  D1 <- length(dirs)  # Giả sử số lượng thư mục con chính là D1
  D2 <- 0  # Giá trị mặc định, có thể cần thay đổi
  lab <- basename(dirs)  # Lấy tên thư mục làm nhãn
  return(list(D1, D2, lab))
}
createDirs <- function(dir, isSubDir=FALSE) {
  if (!dir.exists(dir)) {
    dir.create(dir, recursive=TRUE)  # Tạo thư mục nếu chưa tồn tại
    message(paste("Thư mục", dir, "đã được tạo."))
  } else {
    message(paste("Thư mục", dir, "đã tồn tại."))
  }
}

acquirePrices <- function(prices, lab, len, D, D1, D2,
                          dir, start, end, isSubDir=TRUE) {
  # Giả lập dữ liệu giá ngẫu nhiên
```

```
  set.seed(123)
  prices <- matrix(rnorm(len * D, mean=100, sd=5), nrow=len, ncol=D)
  return(prices)
}

dir <- 'MVO4'
len <- 1006
isQtrly = FALSE
if(isQtrly) back = 5 else back = 4
if(isQtrly) stmt = 'Q' else stmt = 'A'
res <- readSubDirs(dir)
D1 <- res[[1]]
D2 <- res[[2]]
lab <- res[[3]]
D <- D1 + D2
start <- "2011-02-09"
end <- "2015-02-09"
isPlotInAdjCloses <<- FALSE
isCacheEnabled <<- TRUE
prices <- matrix(rep(NA,len*D),nrow=len,ncol=D)
#Must run acquirePrices if cache files do not yet exist:
library(tseries)
prices <- acquirePrices(prices,lab,len,D,D1,D2,dir,
 start=start,end=end,isSubDir=TRUE)
 dir <- 'QMDM'
createDirs(dir,isSubDir=FALSE)
```

Đoạn mã R thực hiện quy trình mô phỏng dữ liệu giá cổ phiếu với các bước sau:

1. Đọc các thư mục con trong thư mục `MVO4` để lấy danh sách mã cổ phiếu (sử dụng làm nhãn).
2. Thiết lập các tham số:
   - `start`, `end`: mốc thời gian bắt đầu và kết thúc để truy xuất dữ liệu.
   - `len`: độ dài chuỗi dữ liệu (số ngày).
   - `D`: tổng số mã cổ phiếu được xác định từ các thư mục con.
3. Gọi hàm `acquirePrices()` để **giả lập dữ liệu giá** cho D mã cổ phiếu trong khoảng thời gian tương ứng.
4. Tạo thư mục đầu ra có tên `QMDM` nếu nó chưa tồn tại, nhằm lưu trữ kết quả hoặc phục vụ các bước xử lý tiếp theo.

Tiếp theo, ta sẽ đọc các báo cáo kết quả kinh doanh hàng quý hoặc hàng năm đã được chuẩn bị từ trước vào một *data frame*, đồng thời gọi hàm `na.omit()` trong R để loại bỏ các bản ghi có dữ liệu bị thiếu. Khung dữ liệu báo cáo kết quả kinh doanh, được đặt tên là `cleanedISDF`, sẽ được trả về.

Nếu đây là lần đầu tiên đoạn mã được chạy và không tìm thấy tệp dữ liệu, hàm sẽ trả về giá trị `NA` cho hàm gọi. Tương tự, nếu tệp tồn tại nhưng không chứa thông tin báo cáo kết quả kinh doanh cho ít nhất 50% các mã cổ phiếu, hàm cũng sẽ trả về `NA`.

Trong cả hai trường hợp trả về `NA`, hàm gọi sẽ cần sử dụng hàm `getFinancials()` của gói `quantmod` để truy xuất số liệu từ các báo cáo tài chính.

```

readAndCleanISDF <- function(expectedLab,
                             dir='QMDM',
                             stmt='A',
                             expectedD=NULL)
{
  setwd(paste(homeuser, "/FinAnalytics/", dir, "/", sep=""))
  fn <- paste("IncomeStmts", stmt, ".csv", sep="")

  # Kiểm tra nếu tệp tồn tại
  if (file.exists(fn)) {
    ISDF <- read.csv(fn, header = TRUE)
    relevantLab <- intersect(expectedLab, ISDF[,1])

    # Đảm bảo expectedD không bị thiếu
    if (is.null(expectedD)) {
      expectedD <- length(expectedLab)
    }

    # Kiểm tra số lượng mã hợp lệ
    if (length(relevantLab) > 0.50 * expectedD) {
      cleanedISDF <- na.omit(ISDF)
      return(cleanedISDF)
    } else {
      return(NA)  # Thiếu dữ liệu
    }
  } else {
    return(NA)  # Tệp không tồn tại
  }
}
```

31

Bằng cách sử dụng hàm `getFinancials()` từ gói `quantmod`, người dùng có thể truy xuất dữ liệu tăng trưởng hàng năm trong ba năm qua cho một tập hợp lớn các cổ phiếu, từ đó có thể thực hiện so sánh giữa các cổ phiếu. Logic này được triển khai dưới dạng một hàm, trong đó sử dụng một vòng lặp `for` lồng đơn. Bên trong vòng lặp, có bốn câu lệnh `tryCatch` liên tiếp để kiểm tra sự hiện diện của các dữ liệu tài chính cần thiết và xử lý các lỗi nếu có.

```
obtainIncomeStmtFigures <- function(lab, dir = "QMDM", isQtrly = TRUE) {

  D <- length(lab)
  back <- if (isQtrly) 5 else 4
  stmt <- if (isQtrly) "Q" else "A"
  ncol <- 2 + 4 * back

  # Cố gắng đọc dữ liệu đã được lưu cache
  ISDF <- readAndCleanISDF(lab, dir = dir, stmt = "A")
  if (!is.null(dim(ISDF))) return(ISDF)

  print("Không tìm thấy tệp báo cáo kết quả kinh doanh")
  ISDF <- data.frame(matrix(nrow = D, ncol = ncol))

  for (d in 1:D) {
    symbol <- lab[d]
    basedate <- NA
    netinc <- rep(NA, back)
    totrev <- rep(NA, back)
    gsprof <- rep(NA, back)
    dneps  <- rep(NA, back)

    print(symbol)

    # Net Income
    tryCatch({
      Net.Income <- eval(parse(text = paste0(
        symbol, ".f$IS$", stmt, '["Net Income",]'
      )))
      netinc <- if (is.numeric(Net.Income[1])) {
        round(Net.Income, 2)
      } else rep(NA, back)
    }, error = function(e) {
```

```
    print(e)
    netinc <- rep(NA, back)
  })

  # Total Revenue
  tryCatch({
    Total.Revenue <- eval(parse(text = paste0(
      symbol, ".f$IS$", stmt, '["Revenue",]'
    )))
    totrev <- if (is.numeric(Total.Revenue[1])) {
      round(Total.Revenue, 2)
    } else rep(NA, back)
  }, error = function(e) {
    print(e)
    totrev <- rep(NA, back)
  })

  # Gross Profit
  tryCatch({
    Gross.Profit <- eval(parse(text = paste0(
      symbol, ".f$IS$", stmt, '["Gross Profit",]'
    )))
    gsprof <- if (is.numeric(Gross.Profit[1])) {
      round(Gross.Profit, 2)
    } else rep(NA, back)
  }, error = function(e) {
    print(e)
    gsprof <- rep(NA, back)
  })

  # Diluted Normalized EPS
  tryCatch({
    Dil.Norm.EPS <- eval(parse(text = paste0(
      symbol, ".f$IS$", stmt, '["Diluted Normalized EPS",]'
    )))
    if (is.numeric(Dil.Norm.EPS[1])) {
      basedate <- names(Dil.Norm.EPS)[1]
      dneps <- round(Dil.Norm.EPS, 2)
    } else {
      dneps <- rep(NA, back)
    }
  }, error = function(e) {
    print(e)
```

```
77        dneps <- rep(NA, back)
78      })
79
80      # Ghi dữ liệu vào bảng kết quả
81      items <- c(symbol, basedate, netinc, totrev, gsprof, dneps)
82      if (length(items) == ncol) {
83        ISDF[d, ] <- items
84      }
85    }
86
87    return(ISDF)
88  }
89
```

Khi đã có nguồn để thu thập dữ liệu báo cáo kết quả kinh doanh, chúng ta sẽ sử dụng nguồn này để ghi dữ liệu vào một tệp CSV để sử dụng lại sau này. Sau khi tệp được chuẩn bị bởi hàm `obtainIncomeStmtGth()`, tệp có thể được ghi ra và đọc lại sau.

Đoạn mã tiếp theo sẽ tìm tất cả các mã cổ phiếu có thể có trong hai tệp (tệp chứa danh sách mã cổ phiếu và tệp chứa dữ liệu báo cáo kết quả kinh doanh). Sau khi tìm được tất cả các mã cổ phiếu cần truy vấn, đoạn mã này hoạt động như một chương trình **ETL (Extract, Transform, Load)** để ghi khung dữ liệu `ISDF` ra tệp. Tuy nhiên, hàm `getFinancials()` trong gói quantmod hiện tại không còn hoạt động với nguồn dữ liệu từ Google nữa, bởi vì Google Finance đã ngừng cung cấp dữ liệu tài chính từ tháng 3 năm 2018. Chính vì thế, thay vì dùng hàm này thì ta sẽ dùng file csv thủ công hoặc tích hợp các API hiện đại.

3.5 Đo lường Tăng trưởng Báo cáo Kết quả Kinh doanh

Khi tính tỷ suất lợi nhuận ròng ta chỉ cần lấy giá trị mới chia cho giá trị ban đầu. Tuy nhiên, trong trường hợp tỷ suất lợi nhuận gộp dương có thể phát sinh trong trường hợp cả giá trị ban đầu và giá trị cuối cùng đều là số âm. Lấy ví dụ về thu nhập ròng của Porter Bancorp với , mã cổ phiếu PBIB: giả sử tính lợi nhuận 2014

Năm	Thu nhập ròng (triệu USD)
2014-12-31	-11.15
2013-12-31	-1.59
2012-12-31	-32.93
2011-12-31	-107.31

so với 2013 theo công thức này ta sẽ có:

$$\text{netincgth} = \frac{-11.15}{-1.59} = 7.012579$$

Chúng ta thu được con số "tươi sáng" là 7.01 (tức là 701.3%), điều này có thể khiến một chương trình đánh giá rằng cổ phiếu này là một ứng viên đáng đầu tư, trong khi thực tế lại có triển vọng rất bi quan vì toàn bộ thu nhập ròng đều âm. Ta sẽ tạo một hàm mới giúp loại bỏ những trường hợp không hợp lleej và tính toán khi có giá trị thu nhập ròng âm:

```
1
2  # Hàm tính tỷ lệ tăng trưởng thu nhập (gross return)
3  # Đầu vào: a = giá trị ban đầu, b = giá trị mới
4  # Trả về: tỷ lệ tăng trưởng hoặc NA nếu không hợp lệ
5
6  calcGth <- function(a, b) {
7    # Loại bỏ các giá trị không hợp lệ: NA, vô cực, hoặc a quá gần 0
8    if (is.na(a) || is.infinite(a) ||
9        is.na(b) || is.infinite(b) || abs(a) < 0.001) {
10     return(NA)
11   }
12
13   # Trường hợp cả hai giá trị đều âm (cùng thua lỗ)
14   # Trả về kết quả âm để phản ánh rằng vẫn là lỗ, dù có "tăng"
15   if (sign(a) == -1 && sign(b) == -1) {
16     return(-abs(b) / abs(a))
17   }
18
19   # Trường hợp chuyển từ âm sang dương (từ lỗ sang lãi)
20   # Không xác định rõ ràng nên trả về NA
21   if (sign(a) == -1 && sign(b) == +1) {
22     return(NA)
23   }
24
25   # Trường hợp chuyển từ dương sang âm (từ lãi sang lỗ)
26   # Cũng không xác định rõ ràng nên trả về NA
27   if (sign(a) == +1 && sign(b) == -1) {
28     return(NA)
29   }
30
31   # Trường hợp bình thường: cả hai đều không âm
32   # Trả về tỷ lệ tăng trưởng làm tròn đến 2 chữ số thập phân
```

```
  return(round(abs(b) / abs(a), 2) * sign(b))
}

# Kiểm thử hàm với các trường hợp khác nhau
calcGth(1.25, 1.75)
calcGth(-1.25, 1.75)
calcGth(1.25, -1.75)
calcGth(-1.25, -1.75)
calcGth(-1.25, NA)
calcGth(1/0, 1.75)
calcGth(0.0005, 1.75)
```

```
1.4
NA
NA
-1.4
NA
NA
NA
```

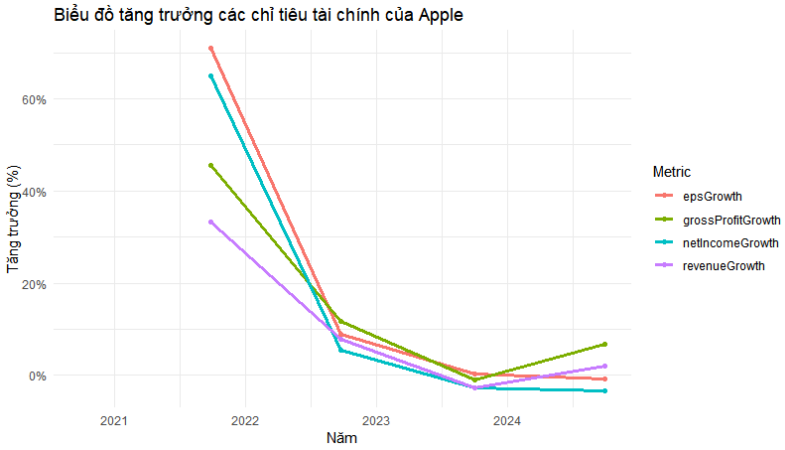
Việc trực quan hóa dữ liệu báo cáo thu nhập thông qua các biểu đồ giúp dễ dàng phát hiện các sai lệch trong quá trình tính toán, đồng thời hỗ trợ phân tích xu hướng tài chính của doanh nghiệp qua từng năm. Phiên bản mã hoàn chỉnh được sử dụng đã xử lý đầy đủ các trường hợp đặc biệt như phép chia cho 0, giá trị thiếu (`NA`) và tăng trưởng âm, đảm bảo kết quả tính toán được chính xác và toàn diện.

Khi áp dụng hàm `plotIncomeStmts()` lên một tập con gồm mười bản ghi liên tiếp từ khung dữ liệu báo cáo thu nhập (`ISDF`), người dùng có thể theo dõi được sự biến động qua các năm của bốn chỉ tiêu tài chính quan trọng, bao gồm:

- Tăng trưởng thu nhập ròng (*net income growth*);
- Tăng trưởng tổng doanh thu (*total revenue growth*);
- Tăng trưởng lợi nhuận gộp (*gross profit growth*);
- Tăng trưởng thu nhập ròng pha loãng trên mỗi cổ phiếu (*diluted net earnings per share growth*).

```
# Đọc dữ liệu
df <- read.csv("C:/Users/Dell/Desktop/đồ án 1/APPL incomestatement.csv")

```

```r
# Chuyển cột ngày sang định dạng Date và sắp xếp theo thời gian
df$date <- as.Date(df$date, format = "%m/%d/%Y")
df <- df[order(df$date), ]

# Hàm tính tăng trưởng phần trăm
growth_rate <- function(x) c(NA, diff(x) / head(x, -1))

# Tính tăng trưởng
df$netIncomeGrowth <- growth_rate(df$netIncome)
df$revenueGrowth <- growth_rate(df$revenue)
df$grossProfitGrowth <- growth_rate(df$grossProfit)
df$epsGrowth <- growth_rate(df$epsdiluted)

# Vẽ biểu đồ
library(ggplot2)
library(tidyr)
library(dplyr)

# Tạo dataframe dạng dài để tiện vẽ
df_long <- df %>%
  select(date,
         netIncomeGrowth,
         revenueGrowth,
         grossProfitGrowth,
         epsGrowth) %>%
  pivot_longer(-date, names_to = "Metric", values_to = "Growth")

# Vẽ biểu đồ tăng trưởng
ggplot(df_long, aes(x = date, y = Growth, color = Metric)) +
  geom_line(size = 1.2) +
  geom_point() +
  labs(title = "Biểu đồ tăng trưởng các chỉ tiêu tài chính của Apple",
       x = "Năm", y = "Tăng trưởng (%)") +
  theme_minimal() +
  scale_y_continuous(labels = scales::percent)
```



Hình 3.2. Biểu đồ tăng trưởng của Apple từ 2021 đến 2024

3.6 Tỉ lệ Sharpe cho báo cáo tăng trưởng của doanh nghiệp

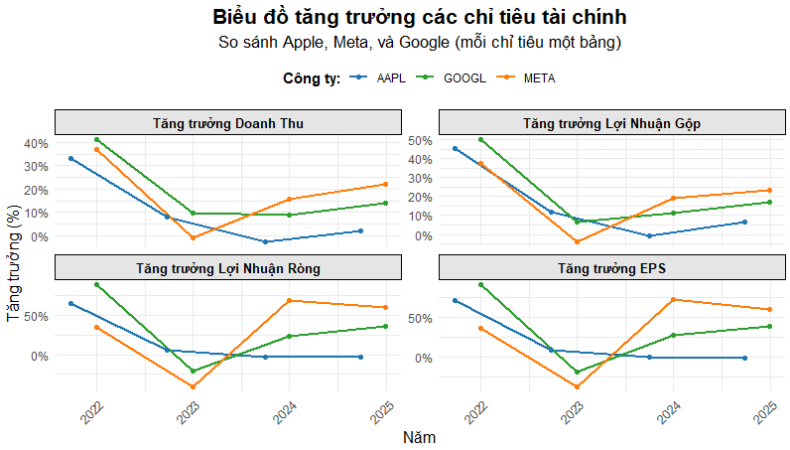
Trong tỉ lệ Sharpe thì chỉ số đầu tiên chúng ta quan tâm đó là lợi nhuận và ở đây là lợi nhuận gộp. Chúng ta sẽ quan sát biểu đồ tăng trưởng của Apple, Meta và Google.

```r
# 0. Nạp các thư viện cần thiết
library(ggplot2)
library(tidyr)
library(dplyr)
library(scales) # Để sử dụng scales::percent

# 1. Định nghĩa đường dẫn đến các file
file_aapl <- "C:/Users/Dell/Desktop/đồ án 1/APPL incomestatement.csv"
file_meta <- "C:/Users/Dell/Desktop/đồ án 1/META_bao_cao_tang_truong
↪  META.csv"
file_googl <- "C:/Users/Dell/Desktop/đồ án
↪  1/GOOGL_bao_cao_tang_truong.csv"

# 2. Hàm tính tăng trưởng phần trăm
growth_rate <- function(x) {
  x_numeric <- as.numeric(x)
  if (length(x_numeric) <= 1 || all(is.na(x_numeric))) {
    return(rep(NA, length(x_numeric)))
  }
  denom <- head(x_numeric, -1)
  denom[denom == 0 | is.na(denom)] <- NA
  c(NA, diff(x_numeric) / denom)
}

```

```r
# 3. Hàm xử lý dữ liệu cho một công ty
process_company_data <- function(file_path, company_name,
                                 date_format = "%m/%d/%Y") {
  df <- read.csv(file_path, stringsAsFactors = FALSE,
                 na.strings = c("", "NA", "N/A"))

  required_cols <- c("date", "netIncome", "revenue", "grossProfit",
                     "epsdiluted")
  if (!all(required_cols %in% names(df))) {
    missing_cols <- setdiff(required_cols, names(df))
    stop(paste("File", file_path, "thiếu một hoặc nhiều cột:",
               paste(missing_cols, collapse = ", ")))
  }

  df$date <- as.Date(df$date, format = date_format)
  df <- df[order(df$date), ]

  clean_numeric <- function(col_data) {
    as.numeric(gsub("[\\s,]", "", col_data))
  }

  df$netIncome <- clean_numeric(df$netIncome)
  df$revenue <- clean_numeric(df$revenue)
  df$grossProfit <- clean_numeric(df$grossProfit)
  df$epsdiluted <- clean_numeric(df$epsdiluted)

  df$netIncomeGrowth <- growth_rate(df$netIncome)
  df$revenueGrowth <- growth_rate(df$revenue)
  df$grossProfitGrowth <- growth_rate(df$grossProfit)
  df$epsGrowth <- growth_rate(df$epsdiluted)

  df$Company <- company_name

  df_long <- df %>%
    select(date, Company, netIncomeGrowth, revenueGrowth,
           grossProfitGrowth, epsGrowth) %>%
    pivot_longer(cols = c(netIncomeGrowth, revenueGrowth,
                          grossProfitGrowth, epsGrowth),
                 names_to = "Metric",
                 values_to = "Growth")

  return(df_long)
}
```

```r

# 4. Xử lý dữ liệu cho từng công ty
df_aapl_long <- process_company_data(file_aapl, "AAPL",
                                     date_format = "%m/%d/%Y")
df_meta_long <- process_company_data(file_meta, "META",
                                     date_format = "%m/%d/%Y")
df_googl_long <- process_company_data(file_googl, "GOOGL",
                                      date_format = "%m/%d/%Y")

# 5. Kết hợp dữ liệu từ tất cả các công ty
all_companies_long <- bind_rows(df_aapl_long, df_meta_long, df_googl_long)

# 6. Vẽ biểu đồ tăng trưởng
all_companies_long_filtered <- all_companies_long %>%
  filter(!is.na(Growth) & is.finite(Growth))

all_companies_long_filtered$Metric <- factor(
  all_companies_long_filtered$Metric,
  levels = c("revenueGrowth", "grossProfitGrowth",
             "netIncomeGrowth", "epsGrowth"),
  labels = c("Tăng trưởng Doanh Thu", "Tăng trưởng Lợi Nhuận Gộp",
             "Tăng trưởng Lợi Nhuận Ròng", "Tăng trưởng EPS")
)

# Định nghĩa màu sắc tùy chỉnh cho các công ty
custom_colors <- c(
  "AAPL" = "#1f77b4",  # Xanh dương
  "META" = "#ff7f0e",  # Cam
  "GOOGL" = "#2ca02c"  # Xanh lá
)

# Vẽ biểu đồ: Mỗi chỉ tiêu một panel, mỗi panel có 3 công ty
plot_metrics_faceted <- ggplot(all_companies_long_filtered,
                               aes(x = date, y = Growth, color = Company,
                                   group = Company)) +
  geom_line(linewidth = 1) +
  geom_point(size = 1.5) +
  scale_color_manual(values = custom_colors) +
  labs(title = "Biểu đồ tăng trưởng các chỉ tiêu tài chính",
       subtitle = "So sánh Apple, Meta, và Google",
       x = "Năm",
       y = "Tăng trưởng (%)",
       color = "Công ty:") +
```

```
  theme_minimal(base_size = 11) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5, size = 13),
    legend.position = "top",
    legend.title = element_text(face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 9),
    axis.title = element_text(size = 12),
    strip.background = element_rect(fill = "grey90", color = "black",
                                    linewidth = 0.5),
    strip.text = element_text(face = "bold", size = 10, color = "black")
  ) +
  facet_wrap(~ Metric, scales = "free_y", ncol = 2)

print(plot_metrics_faceted)
```



Hình 3.3. Biểu đồ tăng trưởng các chỉ tiêu tài chính từ 2021 đến 2024 của Apple, Google, Meta

Một điều đáng chú ý là các doanh nghiệp không thực hiện báo cáo tài chính vào cùng một thời điểm. Ví dụ, Meta và Google thường công bố báo cáo tài chính vào ngày 31/12 hàng năm, trong khi Apple lại chọn ngày 30/9 làm mốc báo cáo. Mặc dù ngày 31/12 là ngày cơ sở phổ biến nhất, vẫn có những mốc thời gian khác được sử dụng, tùy theo chu kỳ kế toán và chính sách báo cáo tài chính riêng của từng công ty.

Tiếp theo, khi phân tích báo cáo thu nhập, chúng ta cần chuẩn hóa các số liệu về cùng một cơ sở, bởi vì quy mô của một công ty trên thị trường chứng khoán được điều chỉnh theo số lượng cổ phiếu đang lưu hành. Ví dụ, nếu Apple có các chỉ tiêu tài chính cao gấp 10 lần một doanh nghiệp nhỏ hơn, thì khả năng cao số lượng cổ

phiếu của Apple cũng cao gấp 10 lần, điều này khiến cho việc so sánh trực tiếp là không hợp lý.

Vì vậy, chúng ta cần một "thước đo" phù hợp để so sánh giữa các công ty – và chỉ tiêu đó chính là lợi nhuận gộp (gross returns), bởi nó có tính tương đồng với việc đo lường hiệu suất cổ phiếu.

Trong phân tích tăng trưởng thu nhập của doanh nghiệp, các chỉ số như doanh thu, lợi nhuận,... sẽ được chuẩn hóa về một mốc cơ sở (ví dụ là 1.0), tương tự như cách ta tính lợi nhuận trong cổ phiếu. Sau đó, các chỉ số này có thể được xử lý tiếp bằng các phương pháp tài chính như log-return, tỷ lệ Sharpe, v.v.

Ví dụ: Nếu một công ty có doanh thu tăng từ 100 triệu USD lên 120 triệu USD trong một năm, thì lợi nhuận gộp sẽ là 1.2. Tương tự, nếu giá cổ phiếu tăng từ 100 USD lên 120 USD, ta cũng có lợi nhuận gộp là 1.2.

Từ đó ta sẽ tính toán lợi nhuận trung bình và độ lệch chuẩn để phục vụ cho việc tính tỉ lệ Sharpe. Ví dụ: một tài sản có mức lợi nhuận gộp trong các năm liên tiếp lần lượt là 1.20, 0.95, 1.43, 1.00, 0.87.

Trung bình lợi nhuận gộp:

$$\bar{x} = \frac{1.20 + 0.95 + 1.43 + 1.00 + 0.87}{5} = \frac{5.45}{5} = 1.09$$

Độ lệch chuẩn:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} = \sqrt{\frac{0.2038}{4}} = \sqrt{0.05095} \approx 0.2258$$

Giả sử lãi suất phi rủi ro bằng 0 ta tính được tỉ lệ Sharpe:

$$\text{Sharpe Ratio} = \frac{\bar{x}}{s} = \frac{1.09}{0.2258} \approx 4.83$$

Tương tự như cách tính toán đã trình bày ở trên, chúng ta cũng có thể tính Tỷ lệ Sharpe cho các chỉ số tài chính khác. Thông thường, bốn chỉ số được quan tâm bao gồm: Tăng trưởng Lợi nhuận Ròng, Tăng trưởng Tổng Doanh thu, Tăng trưởng Lợi nhuận Gộp và Tăng trưởng Lợi nhuận trên Cổ phiếu (EPS). Việc xem xét đồng thời các yếu tố này sẽ giúp chúng ta chọn ra những công ty có hiệu suất tài chính tốt nhất trong từng hạng mục.

Để thực hiện điều đó, ta sẽ truyền vào các véc-tơ bốn chiều — đại diện cho Tỷ lệ Sharpe của bốn chỉ số nêu trên — sau đó tiến hành sắp xếp và lựa chọn các cổ phiếu đáp ứng được một ngưỡng tối thiểu (threshold) đã định. Cách làm này tương

tự như phần lý thuyết về Tỷ lệ Sharpe đã trình bày trước đó.

Lưu ý rằng: ngưỡng tối thiểu này cần được áp dụng đồng đều cho cả bốn yếu tố, và các yếu tố phải được coi trọng như nhau. Bởi lẽ, nếu một công ty chỉ có Tỷ lệ Sharpe cao ở một yếu tố duy nhất thì vẫn là chưa đủ.

Ví dụ: Một công ty có tăng trưởng lợi nhuận gộp (Gross Profit Growth) đều đặn → Tỷ lệ Sharpe cho chỉ số đó sẽ cao. Tuy nhiên, nếu Lợi nhuận Ròng (Net Income) hoặc EPS lại biến động mạnh hoặc tăng trưởng âm thì điều đó cho thấy công ty có thể quản lý chi phí hoặc thuế chưa hiệu quả, hoặc có những rủi ro tài chính tiềm ẩn, gây ảnh hưởng tiêu cực đến nhà đầu tư.

Chương 4: Tối ưu hóa Trung bình-Phương sai Markowitz.

4.1 Tối ưu hóa danh mục đầu tư gồm hai tài sản rủi ro.

Danh mục đầu tư (Portfolio) là một tập hợp các tài sản tài chính, chẳng hạn như cổ phiếu, trái phiếu, hoặc các tài sản khác, được nắm giữ bởi một nhà đầu tư nhằm đạt được mục tiêu tài chính nhất định. Các tài sản này được phân bổ với tỷ lệ khác nhau, gọi là trọng số, tùy thuộc vào mục tiêu của nhà đầu tư, chẳng hạn như tối đa hóa lợi nhuận hoặc giảm thiểu rủi ro. Gỉa sử ta có hai tài sản với hai lợi nhuận X và Y tương ứng với trọng số a và b. Tính toán phương sai của danh mục ta sẽ có công thức:

$$\text{Var}(aX+bY) = E(aX+bY)^2 - E^2(aX+bY) = a^2\text{Var}(X) + b^2\text{Var}(Y) + 2ab\text{Cov}(X,Y).$$

Trường hợp đặc biệt của hai tài sản rủi ro trong một danh mục đầu tư, trong đó $\text{Var}(X) > 0$ và $\text{Var}(Y) > 0$, là một ví dụ đặc biệt tốt để xem tối ưu hóa hoạt động. Là một nhà đầu tư, chúng ta luôn quan tâm đến việc giảm thiểu rủi ro, điều này tương ứng với việc giảm thiểu phương sai.

Nếu chúng ta giả định rằng trọng số đầu tư đầu tiên $a = w_d$ đại diện cho phần nợ (đại diện cho tỷ lệ vốn được đầu tư vào các tài sản thu nhập cố định) của một danh mục đầu tư và trọng số thứ hai $b = w_e$ đại diện cho phần cổ phiếu của một danh mục đầu tư, vốn sẽ nắm giữ cổ phiếu, thì hai trọng số này phải tạo thành toàn bộ danh mục đầu tư, vì vậy:

$$w_d + w_e = 1.$$

Thực tế, không nhất thiết phải để a hay b đại diện cho bất kỳ loại đầu tư cụ thể nào, chúng ta chỉ chọn nợ và cổ phiếu làm ví dụ. Thay thế $a = w_d$ và $b = w_e = 1 - w_d$ và sử dụng một số phép tính vi phân để tìm giá trị tối thiểu, chúng ta có thể xác định công thức cho tỷ lệ nợ tối thiểu. Công thức mới cho phương sai của danh mục đầu tưσ_P^2 xuất hiện dưới đây:

$$\sigma_P^2 = w_d^2\sigma_d^2 + (1-w_d)^2\sigma_e^2 + 2w_d(1-w_d)\sigma_{de}$$

$$= w_d^2\sigma_d^2 + \sigma_e^2 - 2w_d\sigma_e^2 + w_d^2\sigma_e^2 + 2w_d\sigma_{de} - 2w_d^2\sigma_{de}.$$

Bây giờ, chúng ta lấy đạo hàm theo w_d vì chúng ta quan tâm đến trọng số tốt nhất cho phần nợ. Chúng ta cũng sẽ tìm được trọng số tốt nhất cho phần cổ phiếu

vì chỉ có hai phần trong trường hợp này.

$$\frac{\partial \sigma_P^2}{\partial w_d} = 2w_d\sigma_d^2 - 2\sigma_e^2 + 2w_d\sigma_e^2 + 2\sigma_{de} - 4w_d\sigma_{de} = 0$$

$$w_d(2\sigma_d^2 + 2\sigma_e^2 - 4\sigma_{de}) = 2\sigma_e^2 - 2\sigma_{de}$$

$$w_d = \frac{2\sigma_e^2 - 2\sigma_{de}}{2\sigma_d^2 + 2\sigma_e^2 - 4\sigma_{de}}$$

Tỉ lệ nợ của danh mục đầu tư có phương sai tối thiểu là:

$$w_d = \frac{\sigma_e^2 - \sigma_{de}}{\sigma_d^2 + \sigma_e^2 - 2\sigma_{de}} = \frac{\sigma_e^2 - \sigma_d\sigma_e\rho}{\sigma_d^2 + \sigma_e^2 - 2\sigma_{de}}$$

Và tỉ lệ vốn chủ sở hữu của danh mục đầu tư có phương sai tối thiểu là:

$$w_e = 1 - w_d.$$

Ta sẽ thực hiện với ví dụ:

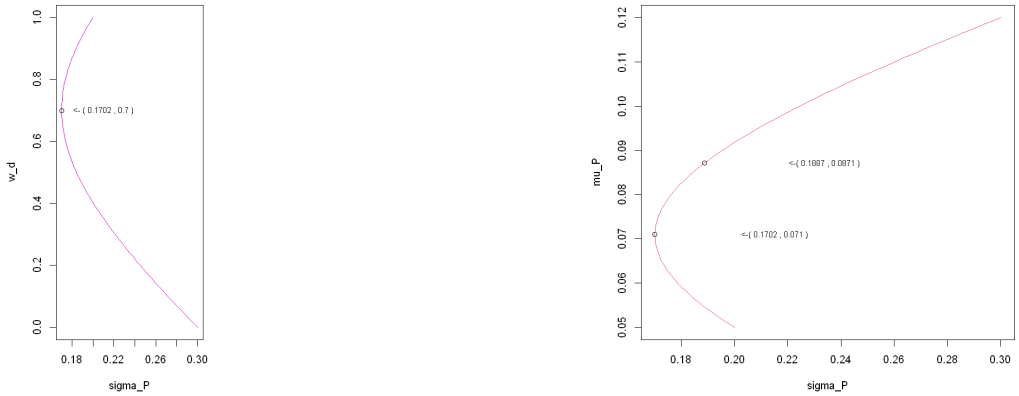
```

mu_d = .05
mu_e = .12
sigma_e = .30
sigma_d = .20
sigma_de = .003
w_d = seq(0,1,.01) #tạo ra một chuỗi số bắt đầu từ 0 đến 1 với bước nhảy là 0.01
mu_P = vector(length=length(w_d))
sigma_P = vector(length=length(w_d))
sr_P = vector(length=length(w_d))
```

Dưới đây là vòng lặp chính:

```

for (u in 1:length(w_d)) {
  mu_P[u] = mu_d * w_d[u] + mu_e * (1 - w_d[u])
  sigma_P[u] = sqrt(w_d[u]^2 * sigma_d^2 +
                    (1 - w_d[u])^2 * sigma_e^2 +
                    2 * w_d[u] * (1 - w_d[u]) * sigma_de)
  sr_P[u] = mu_P[u] / sigma_P[u]
```

```
}
par(bg = "white")
par(mfrow = c(1,2))
plot(sigma_P, w_d, type = "l", ylab = "w_d", col = 6)

ind_min_var_P = which(sigma_P == min(sigma_P))
w_d[ind_min_var_P]

points(sigma_P[ind_min_var_P], w_d[ind_min_var_P])
text(sigma_P[ind_min_var_P] + 0.04, w_d[ind_min_var_P],
     paste("<- (", round(sigma_P[ind_min_var_P], 4), ",",
           w_d[ind_min_var_P], ")"), cex = 0.75)
#Bây giờ vẽ đồ thị
sigma_P dưới dạng hàm của mu_P
par(bg = "white")
plot(sigma_P,mu_P,type="l",ylab="mu_P",col=2)
mu_P[ind_min_var_P]
points(sigma_P[ind_min_var_P],mu_P[ind_min_var_P])
text(sigma_P[ind_min_var_P]+.045,mu_P[ind_min_var_P],
 paste("<-(",round(sigma_P[ind_min_var_P],4),",",
 mu_P[ind_min_var_P],")"),cex=.75)
ind_opt_P = sr_P == max(sr_P)
mu_P[ind_opt_P]
points(sigma_P[ind_opt_P],mu_P[ind_opt_P])
text(sigma_P[ind_opt_P]+.045,mu_P[ind_opt_P],
 paste("<-(",round(sigma_P[ind_opt_P],4),
 ",",mu_P[ind_opt_P],")"),cex=.75)
```

0.7



Hình 4.1. Giá trị của w_d tạo ra danh mục đầu tư có phương sai tối thiểu (bên trái). Giá trị của μ_p tạo ra danh mục đầu tư có phương sai tối thiểu xuất hiện dưới giá trị của μ_p tạo ra danh mục đầu tư có tỷ lệ Sharpe cao nhất (tangency portfolio) (bên phải).

- **Biên giới hiệu quả** là toàn bộ đường cong trong biểu đồ bên phải của Hình 4.1. Dọc theo đường cong này, ta có thể thấy phương sai tối ưu cho mỗi mức lợi suất nhất định.
- **Danh mục đầu tư có phương sai tối thiểu** có lợi suất và phương sai thấp nhất trong hai điểm được chọn trong biểu đồ bên phải của Hình 4.1.
- **Danh mục đầu tư tangency**, với tỷ lệ Sharpe cao nhất, là điểm có lợi suất và phương sai cao hơn trong hai điểm được chọn trong biểu đồ bên phải của Hình 4.1.

4.2 Quy hoạch bậc 2

Công thức tối ưu hóa danh mục đầu tư có thể được nhận dạng và chuyển thành bài toán Quy hoạch bậc hai. Quy hoạch bậc hai (Quadratic Programming - QP) là một loại bài toán tối ưu hóa trong đó hàm mục tiêu có dạng hàm bậc hai, tức là chứa các biến số bậc hai (hoặc các sản phẩm của các biến số). Để qiair bài toán quy hoạch bậc hai trong R ta có hàm `solve.QP()` trong gói `quadprog`.

Cách sử dụng `solve.QP()`

```
solve.QP(Dmat, dvec, Amat, bvec, meq = 0, factorized = FALSE)
```

Các tham số:

- `Dmat`: Ma trận xuất hiện trong hàm bậc hai cần tối thiểu hóa.
- `dvec`: Vector xuất hiện trong hàm bậc hai cần tối thiểu hóa.

- `Amat`: Ma trận xác định các ràng buộc mà theo đó ta muốn tối thiểu hóa hàm bậc hai.
- `bvec`: Vector chứa các giá trị b_0 trong các ràng buộc (mặc định là 0).
- `meq`: Số lượng ràng buộc đầu tiên được xử lý như các ràng buộc đẳng thức, các ràng buộc còn lại là bất đẳng thức (mặc định là 0).
- `factorized`: Cờ logic: nếu đặt là `TRUE`, thì thay vì truyền trực tiếp ma trận D, ta truyền vào ma trận R^{-1} (với $D = R^T R$).

Cụ thể, bài toán Quy hoạch Bậc hai (QP) được mô tả dưới dạng công thức:

$$\arg\min_{\mathbf{b}} \left(\frac{1}{2} \mathbf{b}^T D \mathbf{b} - \mathbf{d}^T \mathbf{b} \right)$$

với ràng buộc:

$$A^T \mathbf{b} \geq \mathbf{b}_0$$

Trong đó:

- **b** là vector biến cần tìm,
- D là ma trận hệ số bậc hai ($p \times p$),
- **d** là vector hệ số tuyến tính ($p \times 1$),
- A là ma trận chứa các hệ số ràng buộc,
- $\mathbf{b}_0$ là vector hằng số ràng buộc.

Để minh họa cho cơ chế giải bài toán QP, ta xét một ví dụ đơn giản:

$$\arg\min_{\mathbf{b}} \left(x_1^2 + 2x_2^2 + 4x_3^2 - x_1 - x_2 + 5x_3 \right)$$

với các ràng buộc:

$$x_1 + x_3 \leq 1, \quad x_1 \geq 5, \quad x_2 \leq 0.$$

Chuyển bài toán về dạng ma trận, ta có:

- Ma trận hệ số bậc hai:

$$D = 2\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

- Vector hệ số tuyến tính:

$$\mathbf{d} = \begin{bmatrix} 1 \\ 1 \\ -5 \end{bmatrix}$$

- Vector nghiệm:

$$\mathbf{b} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- Ma trận ràng buộc và vector hằng số:

$$A^T = \begin{bmatrix} -1 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}, \quad \mathbf{b}_0 = \begin{bmatrix} -1 \\ 5 \\ 0 \end{bmatrix}$$

Hàm `solve.QP()` trong ngôn ngữ R được sử dụng để giải bài toán trên và trả về nghiệm tối ưu cho **b**. Việc áp dụng phương pháp QP trong tài chính giúp tìm ra danh mục đầu tư tối ưu dưới các ràng buộc cụ thể.

```
1
2  library(quadprog)
3  library(tseries)
4  P = 2*diag(c(1,2,4))
5  d = c(1,1,-5)
6  At = matrix(0,nrow=3,ncol=3)
7  At[1,] = c(-1,0,-1)
8  At[2,] = c(1,0,0)
9  At[3,] = c(0,-1,0)
10 b0 = c(-1,5,0)
11 P
```

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 8 \end{bmatrix}$$

```
1  At
```

$$\begin{bmatrix} -1 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

```
1 b0
```

-1.5.0

```
1 xHat = solve.QP(P, d, t(At), b0)$solution
2 xHat
```

5.0.-4

Ta thấy rằng $b = [x_1, x_2, x_3]^T = [5, 0, -4]^T$ thỏa mãn các ràng buộc $x_1 + x_3 \leq 1$, $x_1 \geq 5$ và $x_2 \leq 0$, do đó các ràng buộc của bài toán đã được thỏa mãn.

Tham số thứ ba trong hàm `solve.QP()` có thể gây nhầm lẫn. Công thức trên xác định các ràng buộc dưới dạng A^T, nhưng tham số thứ ba của bộ giải lại được chỉ định lại là ma trận A . Vì vậy, trong mã R trên, một biến ma trận đại diện cho A^T được tạo ra, gọi là `At`, và sau đó biến này sẽ được chuyển vị trước khi đưa vào bộ giải. Điều này rất quan trọng và có thể gây khó khăn khi áp dụng.

4.3 Tối ưu hóa danh mục đầu tư bằng phương pháp Markowitz sử dụng Lập trình bậc hai (Quadratic Programming)

Việc áp dụng Lý thuyết Danh mục Đầu tư giúp cung cấp cơ sở lý thuyết vững chắc cho việc so sánh và kết hợp các chứng khoán cổ phiếu trong dài hạn. Chiến lược của chúng ta tập trung vào việc đầu tư dài hạn vào cổ phiếu (long-only), đồng thời tận dụng tối đa các nguồn dữ liệu công khai sẵn có, cụ thể là giá đóng cửa hàng ngày. Lý thuyết Danh mục Đầu tư của Markowitz và Sharpe (Sharpe, Alexander, và Bailey, 1999) không chỉ giúp tối ưu hóa danh mục mà còn hỗ trợ phân loại các khoản đầu tư trong danh mục P một cách hiệu quả.

Giả sử có một ma trận chuỗi thời gian lợi nhuận log $\mathbf{R}$ có kích thước $N \times p$, trong đó $\mathbf{R} = (R_1, ..., R_p)$ là lợi nhuận log của p cổ phiếu qua N thời điểm. Mỗi phần tử R_{ij} được tính theo công thức:

$$R_{ij} = \ln\left(\frac{S_{ij}}{S_{(i-1)j}}\right)$$

với S_{ij} là giá cổ phiếu j tại thời điểm i. Trường hợp dữ liệu là tỷ giá hối đoái, lợi

nhuận log có thể đơn giản hóa là $R_{ij} = S_{ij}$ nếu giả định các giá trị này phân phối chuẩn.

Kỳ vọng lợi nhuận được tính là:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} R_{ij}, \quad \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}$$

Ma trận hiệp phương sai được ký hiệu là $\boldsymbol{\Sigma} = \text{cov}(\mathbf{R})$, có kích thước $p \times p$. Đây là các thông tin cần thiết để thực hiện tối ưu hóa danh mục theo phương pháp Markowitz.

Lợi nhuận kỳ vọng của danh mục được xác định bởi:

$$R_P = \mathbf{Rw} \quad \Rightarrow \quad \mathbb{E}[R_P] = \mu_P = \boldsymbol{\mu}^\top \mathbf{w}$$

Phương sai danh mục được tính theo:

$$\sigma_P^2 = \mathbf{w}^\top \boldsymbol{\Sigma} \mathbf{w}$$

Tỷ lệ Sharpe, một chỉ số đánh giá hiệu quả đầu tư có điều chỉnh theo rủi ro, được xác định như sau:

$$\text{Sharpe Ratio} = \frac{\mathbb{E}[R_P] - \mu_f}{\sigma_P}$$

trong đó μ_f là lãi suất phi rủi ro. Bài toán tối ưu hoá danh mục đầu tư là tìm vector trọng số $\mathbf{w}$ sao cho phương sai danh mục nhỏ nhất, với một mức lợi nhuận kỳ vọng cố định.

Để sử dụng hàm `solve.QP()` trong ngôn ngữ R, ma trận hiệp phương sai $\boldsymbol{\Sigma}$ cần là ma trận bán xác định dương (PSD). Để đảm bảo điều kiện này, có thể loại bỏ các tài sản có lợi nhuận thấp hoặc tương quan cao với nhau.

4.3.1 Mô hình tối ưu hóa QP

Mục tiêu:

$$\min_{\mathbf{w}} \quad \mathbf{w}^\top \boldsymbol{\Sigma} \mathbf{w}$$

Ràng buộc đẳng thức

$$\mathbf{A}_{\text{eq}}\mathbf{w} = \mathbf{b}_{\text{eq}}$$

Trong đó:

$$\mathbf{A}_{\text{eq}} = \begin{bmatrix} \boldsymbol{\mu}^{\mathsf{T}} \\ \mathbf{e}^{\mathsf{T}} \end{bmatrix}, \quad \mathbf{b}_{\text{eq}} = \begin{bmatrix} \mu_P \\ 1 \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Ràng buộc bất đẳng thức (Không bán khống)

$$\mathbf{A}_{\text{neq}}\mathbf{w} \geq \mathbf{b}_{\text{neq}}$$

Trong đó:

$$\mathbf{A}_{\text{neq}} = \text{diag}(p), \quad \mathbf{b}_{\text{neq}} = \mathbf{0}_p$$

Với $\text{diag}(p)$ là ma trận chéo đơn vị kích thước $p \times p$, và $\mathbf{0}_p$ là vector không kích thước p.

Ma trận tổng hợp các ràng buộc

$$\mathbf{A}_{\text{mat}} = \left[\mathbf{A}_{\text{eq}} \,|\, \mathbf{A}_{\text{neq}}\right], \quad \mathbf{b}_{\text{vec}} = \left[\mathbf{b}_{\text{eq}}^{\top} \,|\, \mathbf{b}_{\text{neq}}^{\top}\right]^{\top}$$

4.4 Ràng buộc, Hình phạt và Phương pháp Lasso

Một cách tổng quát, trong các bài toán tối ưu, ta thường xét một miền xác định trong không gian p-chiều và một hàm mục tiêu f, hàm này ánh xạ các điểm trong không gian đó đến một giá trị thực. Do đó, đồ thị của f sẽ là một bề mặt trong không gian $(p+1)$-chiều. Ví dụ, khi $p = 2$, tức là ta có hai biến đầu vào, thì f sẽ tạo thành một bề mặt trong không gian ba chiều.

Bài toán tối ưu tương ứng là tìm điểm $\mathbf{x} \in \mathbb{R}^p$ sao cho giá trị của hàm $f(\mathbf{x})$ đạt cực tiểu hoặc cực đại. Trong khuôn khổ lý thuyết Markowitz và Sharpe về tối ưu hóa danh mục đầu tư, mục tiêu điển hình là đạt được một mức lợi nhuận kỳ vọng nhất định, sau đó tìm danh mục đầu tư với phương sai lợi nhuận (hoặc độ lệch chuẩn) nhỏ nhất tương ứng với mức lợi nhuận đó. Do đó, bài toán trở thành một bài toán

cực tiểu hóa phương sai danh mục đầu tư dưới các ràng buộc thích hợp. Như vậy, coi phương sai là f(x) thì bài toán của ta sẽ là bài toán tìm cực tiểu của f(x) với các điều kiện ràng buộc (là đẳng thức hoặc bất đẳng thức). Để giải bài toán này, ta sẽ đưa các ràng buộc về dạng thỏa mãn điều kiện Karush–Kuhn–Tucker (KKT). Cụ thể:

Tối thiểu hóa $f(x)$ sao cho $g_i(x) \leq 0$, $h_j(x) = 0$ với $i = 1, \ldots, l$ và $j = 1, \ldots, m$.

Tức là, ta sẽ có l bất phương trình và m phương trình. Phương trình đầy đủ cho điều kiện KKT được viết là:

$$\begin{aligned} x^* &= argmin_x f(x) = argmin_x L(x, \lambda, \mu) \\ &= argmin_x \left[f(x) + \sum_{i=1}^{l} \lambda_i g_i(x) + \sum_{j=1}^{m} \mu_j h_j(x) \right] \end{aligned}$$

Trong đó, $L(x, \lambda, \mu)$ là hàm Lagrangian, phụ thuộc vào các nhân tử Lagrange λ và μ. Để giải bài toán này, ta sẽ đạo hàm theo từng biến và giải hệ phương trình có p+l+m ẩn:

$$\nabla f(x) + \sum_{i=1}^{l} \lambda_i \nabla g_i(x) + \sum_{j=1}^{m} \mu_j \nabla h_j(x) = 0 \tag{4.1}$$

Ví dụ: Ta xét bài toán:

$$\begin{aligned} \text{Minimize} \quad & f(x, y) = x^2 + y^2 \\ \text{với điều kiện} \quad & x + y = 1 \end{aligned}$$

Ta xây dựng hàm mới gọi là hàm Lagrangian:

$$L(x, y, \lambda) = f(x, y) + \lambda \cdot g(x, y)$$

Ở đây:

$$f(x, y) = x^2 + y^2$$

$$g(x, y) = x + y - 1 = 0 \quad \text{(ràng buộc)}$$

λ là nhân tử Lagrange (số phụ thêm, không biết trước).

Vậy:

$$L(x, y, \lambda) = x^2 + y^2 + \lambda(x + y - 1)$$

Giải bằng cách đạo hàm: Lấy đạo hàm riêng theo từng biến:

$$\begin{cases} \frac{\partial L}{\partial x} = 2x + \lambda = 0 \\ \frac{\partial L}{\partial y} = 2y + \lambda = 0 \\ \frac{\partial L}{\partial \lambda} = x + y - 1 = 0 \end{cases}$$

Giải hệ này, ta tìm được x, y thỏa mãn điều kiện ràng buộc và tối thiểu hoá $f(x, y)$.

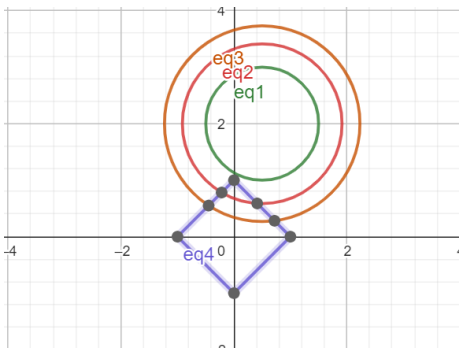
Trong thống kê, có thể thấy rằng một nghiệm sẽ có lợi hơn nếu nó có ít tham số hơn và các giá trị của tham số nhỏ hơn, đây cũng chính là ý tưởng của shrinkage property (tính co rút). Shrinkage là khuynh hướng đưa các thành phần không cần thiết của nghiệm về đúng bằng 0. Ví dụ, trong một mô hình hồi quy, ta sẽ tìm các để mô hình không bị overfit với tập dữ liệu đang có để có thể sử dụng tốt mô hình với tập dữ liệu trong tương lai. Mô hình mới này có thể có độ lệch lớn hơn mô hình overfit ban đầu nhưng lại có tính tổng quát hơn.

Để thực hiện việc co rút này ta có thể thêm một ràng buộc phạt có tác dụng là một giới hạn trên có tác dụng siết chặt (tightening) hoặc "thắt dây thừng" (lassoing) các kết quả. phương pháp này được gọi là LASSO (Least Absolute Selection and Shrinkage Operator). Trong không gian véc tơ p chiều thì chuẩn ℓ_1 là tổng giá trị tuyệt đối của từng thành phần ($\sum_{i=1}^{p} |x_i|$), và chuẩn ℓ_2 là căn bậc hai của tổng bình phương.

Ví dụ: Ta xét bài toán:

$$\begin{aligned} \text{Minimize} \quad & f(x, y) = (x - 0.5)^2 + (y - 2)^2 \\ \text{với điều kiện} \quad & |x| + |y| \leq 1 \end{aligned}$$

Ta có đồ thị:

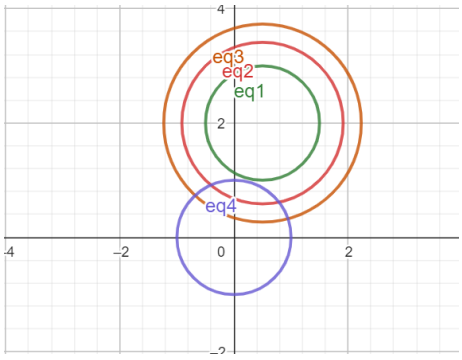


Hình 4.2. Các đường đồng mức của hàm mục tiêu $f(x, y)$ và miền ràng buộc L1 (hình thoi) trong bài toán LASSO.

Như vậy, ta có thể thấy nghiệm tối ưu của bài toán là đỉnh hình thoi có tọa độ

(0;1) thỏa mãn tính co rút.

Tương tự như trên, nếu điều kiện được đổi thành $\sqrt{x^2+y^2} \leq 1$ ta se có miền ràng buộc là hình tròn:



Hình 4.3. Đường đồng mức của hàm mục tiêu $f(x,y)$ và miền ràng buộc L1 (hình tròn màu tím) trong bài toán LASSO.

Lúc này nghiệm tối ưu không còn là điểm có tọa độ (0;1) nữa mà sẽ là một điểm liên quan đến sự kết hợp của x và y khi đó ta sẽ mất đi tính chất co rút (shrinkage).

Ta có thể thấy ℓ_1 cung cấp tính co rút tốt hơn nên được lựa chọn trong Lasso. Thay vì phương pháp hình học như ở trên, ta cũng có thể giải bằng phương pháp KKT với các ràng buộc thỏa mãn:

$$\begin{aligned} \text{Min} \quad & f(x,y) = (x-0.5)^2 + (y-2)^2 \\ \text{với điều kiện} \quad & g_1(x,y) = x + y - 1 \leq 0 \\ & g_2(x,y) = x - y - 1 \leq 0 \\ & g_3(x,y) = -x + y - 1 \leq 0 \\ & g_4(x,y) = -x - y - 1 \leq 0 \end{aligned}$$

Ta xây dựng hàm mới gọi là hàm Lagrangian:

$$L(x,y,\lambda_i) = f(x,y) + \lambda_i \cdot g_i(x,y)$$

Xem xét hình vẽ ở trên ta sẽ thấy rằng g_1 và g_3 sẽ là những đường gần tâm hơn nên ta loại bỏ 2 đường còn lại để đơn giản bài toán.

Đạo hàm theo từng biến, ta thu được hệ phương trình:

$$\begin{aligned}
&2x - 1 + \lambda_1 - \lambda_3 = 0 \\
&2y - 4 + \lambda_1 + \lambda_3 = 0 \\
&x + y = 1 \\
&-x + y = 1
\end{aligned}$$

Viết dưới dạng ma trận:

$$\begin{bmatrix} 2 & 0 & 1 & -1 \\ 0 & 2 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \lambda_1 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 1 \\ 1 \end{bmatrix}$$

$$\Rightarrow (x, y, \lambda_1, \lambda_3) = \left(0, 1, \frac{3}{2}, \frac{1}{2}\right)$$

Hàm Lagrangian tại nghiệm là:

$$L\left((x, y), (\lambda_1, \lambda_3)\right) = \left\{ \left(x - \frac{1}{2}\right)^2 + (y-2)^2 + \lambda_1(x + y - 1) + \lambda_3(-x + y - 1) \right\}$$

Thay các giá trị λ_1 và λ_3 vào biểu thức $\lambda_1(x + y - 1) + \lambda_3(-x + y - 1)$ ta được $x + 2y - 2$. Biểu diễn này là một mặt phẳng trong không gian (x, y, z). Tại $z = 0$, mặt phẳng $x + 2y - 2 = 0$ này sẽ cắt miền ràng buộc tại $x = 0, y = 1$.

Bên cạnh lợi nhuận, các chỉ số rủi ro như phương sai và hiệp phương sai so với các chứng khoán còn lại cũng rất quan trọng. Những chứng khoán có hiệp phương sai thấp hoặc âm với phần còn lại được ưu tiên, thậm chí hơn cả những chứng khoán có tỷ lệ Sharpe cao hơn, bởi vì chúng đóng góp vào khả năng đa dạng hóa danh mục. Một hướng phát triển tự nhiên của thuật toán Markowitz là mở rộng nó cho nhiều chứng khoán hơn và do đó tăng số chiều. Điều này cho phép khai phá tập lớn các chứng khoán để tìm ra những giá trị tốt.

Kết luận

Kết luận đồ án

Đồ án đã trình bày các bước cần thiết trong quy trình xây dựng danh mục đầu tư định lượng.

1. **Xử lý dữ liệu chứng khoán:** Đồ án đã chỉ ra cách thức thu thập, làm sạch và điều chỉnh dữ liệu giá chứng khoán cho các sự kiện doanh nghiệp như cổ tách và sáp nhập, đây là bước nền tảng quan trọng cho mọi phân tích tài chính.
2. **Phân tích rủi ro và biến động giá:** Đồ án đã mô tả việc sử dụng lợi nhuận log và đặc biệt là mô hình phân phối hỗn hợp chuẩn để nắm bắt đặc điểm "đuôi dày" trong dữ liệu tài chính, phản ánh khả năng xảy ra các biến động giá cực đoan như ví dụ về tỷ giá Nhân dân tệ. Điều này nhấn mạnh sự cần thiết của các mô hình rủi ro thực tế hơn so với phân phối chuẩn truyền thống.
3. **Lọc ứng viên đầu tư:** Phương pháp sử dụng Tỷ lệ Sharpe như một tiêu chí độc lập để xếp hạng và loại bỏ các chứng khoán có hiệu suất điều chỉnh theo rủi ro thấp đã được triển khai. Đồ án cũng mở rộng việc áp dụng Tỷ lệ Sharpe để đánh giá sự tăng trưởng của các chỉ tiêu trong báo cáo kết quả kinh doanh của doanh nghiệp, cho thấy khả năng tích hợp dữ liệu cơ bản vào quy trình lọc.
4. **Thực hiện tối ưu hóa danh mục:** ài toán tối ưu hóa Trung bình-Phương sai đã được công thức hóa thành bài toán Quy hoạch bậc hai (QP) và được chứng minh có thể giải quyết bằng hàm solve.QP() trong R.

Hướng phát triển đồ án trong tương lai

Trong tương lai, nghiên cứu của đồ án này có thể được mở rộng và phát triển với những hướng đi tiềm năng sau:

1. **Mở rộng quy mô và phức tạp hóa mô hình tối ưu:** Mở rộng thuật toán Markowitz để xử lý một số lượng lớn hơn các chứng khoán ("tăng số chiều"). Điều này có thể đòi hỏi việc tích hợp các kỹ thuật xử lý dữ liệu lớn và các phương pháp tối ưu hóa hiệu quả hơn.
2. **Kết hợp các kỹ thuật chọn lọc tài sản nâng cao:** Tích hợp các phương pháp hình phạt như Lasso (sử dụng chuẩn $\ell 1$) vào bài toán tối ưu. Lasso có

khả năng thúc đẩy tính "co rút" (shrinkage), giúp loại bỏ các tài sản không cần thiết và tạo ra danh mục với ít thành phần hơn, đặc biệt hữu ích khi làm việc với nhiều tài sản.

Tài liệu tham khảo

[1] Hogg, R. V., & Craig, A. T. (1978). *Introduction to mathematical statistics* (4th ed.). Macmillan.

[2] Ruppert, D. (2011). *Statistics and data analysis for financial engineering*. Springer.

[3] Sharpe, W. F., Alexander, G. J., & Bailey, J. V. (1999). *Investments* (6th ed.). Prentice Hall.

[4] Bennett, M. J., & Hugen, D. L. (2016). *Financial Analytics with R: Building a Laptop Laboratory for Data Science*. Cambridge University Press.

[5] Wickham, H., & Grolemund, G. (2017). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media.

[6] Bennett, M. J., & Hugen, D. L. (2016). *Financial Analytics with R: Building a Laptop Laboratory for Data Science*. Cambridge University Press.