

## 0) Dữ liệu & chuẩn bị

```
rm(list = ls())  
  
data("EuStockMarkets")    # có sẵn trong R  
P <- EuStockMarkets      # giá (ts)  
head(P)
```

## 1) Tỷ suất sinh lời: tính chất thống kê (mean, sd, skew, kurt, JB)

### 1.1 Tính return (simple & log)

```
# Log return (khuyên nghị trong finance)  
R <- diff(log(P))           # (T-1) x N  
R <- as.matrix(R)  
head(R)  
  
# Simple return (nếu muốn)  
R_simple <- P[-1,]/P[-nrow(P),] - 1  
R_simple <- as.matrix(R_simple)
```

### 1.2 Thống kê mô tả cơ bản

```
mu <- colMeans(R)  
sdv <- apply(R, 2, sd)  
varv <- apply(R, 2, var)  
  
summary_stats <- cbind(mean = mu, sd = sdv, var = varv,  
                        min = apply(R, 2, min),  
                        q05 = apply(R, 2, quantile, probs=0.05),  
                        median = apply(R, 2, median),  
                        q95 = apply(R, 2, quantile, probs=0.95),  
                        max = apply(R, 2, max))  
round(summary_stats, 6)
```

### 1.3 Skewness, Kurtosis

```
# Skewness và excess kurtosis: công thức moment  
skew <- apply(R, 2, function(x) {  
  m <- mean(x); s <- sd(x)  
  mean((x - m)^3) / (s^3)  
})  
  
ex_kurt <- apply(R, 2, function(x) {  
  m <- mean(x); s <- sd(x)  
  mean((x - m)^4) / (s^4) - 3  
})  
  
round(cbind(skew = skew, ex_kurt = ex_kurt), 4)
```

## 1.4 Jarque–Bera (tự tính theo công thức)

```
n <- nrow(R)
JB <- (n/6) * (skew^2 + (ex_kurt^2)/4)
pval <- 1 - pchisq(JB, df = 2)

round(cbind(JB = JB, p_value = pval), 6)
```

## 1.5 Minh họa đồ thị (Histogram & QQ)

```
x <- R[, "DAX"]
hist(x, breaks = 50, main = "DAX log-returns", xlab = "log return")
qqnorm(x); qqline(x)
```

## 2) E(R), sigma, variance, covariance, correlation

```
E_R <- colMeans(R)
Sigma <- cov(R)      # covariance matrix
Corr  <- cor(R)      # correlation matrix

E_R
Sigma
Corr

# Ví dụ cặp DAX-FTSE
cov(R[, "DAX"], R[, "FTSE"])
cor(R[, "DAX"], R[, "FTSE"])
```

## 3) Lý thuyết danh mục

### 3.1) Chuẩn bị dữ liệu & return

```
# Chọn 2 tài sản
R2 <- R[, c("DAX", "FTSE")]
head(R2)
```

### 3.2) Mean, variance, covariance

```
mu <- colMeans(R2)
Sigma <- cov(R2)

mu
Sigma

mu_DAX <- mu["DAX"]
mu_FTSE <- mu["FTSE"]
```

```
var_DAX <- Sigma["DAX", "DAX"]
var_FTSE <- Sigma["FTSE", "FTSE"]
cov_DF <- Sigma["DAX", "FTSE"]
```

### 3.3) Tạo danh mục với tỷ trọng thay đổi

Giả sử:

- $w$  = tỷ trọng DAX
- $1 - w$  = tỷ trọng FTSE
- $w \in [0,1]$

```
w <- seq(0, 1, by = 0.01)

# Expected return danh mục
ER_p <- w * mu_DAX + (1 - w) * mu_FTSE

# Variance danh mục (công thức Markowitz 2 tài sản)
Var_p <- w^2 * var_DAX +
  (1 - w)^2 * var_FTSE +
  2 * w * (1 - w) * cov_DF

SD_p <- sqrt(Var_p)
```

### 3.4) Vẽ đường kết hợp (Mean–SD)

```
plot(SD_p, ER_p, type = "l", lwd = 2,
  xlab = "Portfolio Risk (SD)",
  ylab = "Expected Return",
  main = "Two-Asset Portfolio: DAX & FTSE")

# Điểm 2 tài sản riêng lẻ
points(sqrt(var_DAX), mu_DAX, pch = 16, col = "blue")
points(sqrt(var_FTSE), mu_FTSE, pch = 16, col = "red")

text(sqrt(var_DAX), mu_DAX, "DAX", pos = 4)
text(sqrt(var_FTSE), mu_FTSE, "FTSE", pos = 4)
```

### 3.5) Điểm danh mục phương sai tối thiểu (Minimum Variance Portfolio)

Công thức với 2 tài sản:

$$w_{DAX}^* = \frac{\sigma_{FTSE}^2 - \text{Cov}_{DF}}{\sigma_{DAX}^2 + \sigma_{FTSE}^2 - 2\text{Cov}_{DF}}$$

```
w_mvp <- (var_FTSE - cov_DF) / (var_DAX + var_FTSE - 2*cov_DF)
w_mvp
```

Tính mean & sd của MVP:

```
ER_mvp <- w_mvp * mu_DAX + (1 - w_mvp) * mu_FTSE
SD_mvp <- sqrt(
  w_mvp^2 * var_DAX +
  (1 - w_mvp)^2 * var_FTSE +
  2 * w_mvp * (1 - w_mvp) * cov_DF
)
points(SD_mvp, ER_mvp, pch = 17, col = "darkgreen", cex = 1.3)
text(SD_mvp, ER_mvp, "MVP", pos = 4)
```

## Mở rộng cho nhiều tài sản:

**Giả sử trọng số w và tính E(Rp), Var(Rp), SD(Rp)**

```
w <- c(0.25, 0.25, 0.25, 0.25) # equal weight
mu <- colMeans(R)
S <- cov(R)

ERp <- sum(w * mu)
Varp <- as.numeric(t(w) %*% S %*% w)
SDp <- sqrt(Varp)

ERp; SDp; Varp
```

**Minh họa “đa dạng hóa”: so sánh risk của từng tài sản vs danh mục**

```
sd_assets <- sqrt(diag(S))
sd_assets
SDp
```

**Efficient frontier “minh họa” bằng random weights**

```
set.seed(1)
N <- 3000
W <- matrix(runif(N*4), ncol=4)
W <- W / rowSums(W)

ER <- W %*% mu
SD <- apply(W, 1, function(wi) sqrt(as.numeric(t(wi) %*% S %*% wi)))

plot(SD, ER, pch=16, cex=0.5,
      xlab="Risk (SD)", ylab="Expected return",
      main="Random portfolios (EuStockMarkets)")
```

## 4) NPV & IRR

Giả sử dự án: CF = -100,30,40,50,20, discount rate 10%.

## 4.1 NPV

```
cf <- c(-100, 30, 40, 50, 20)
r  <- 0.10
t  <- 0:(length(cf)-1)

NPV <- sum(cf / (1+r)^t)
NPV
```

## 4.2 IRR (dùng uniroot trực tiếp)

```
# IRR là nghiệm của NPV(r)=0
f <- function(rate) sum(cf / (1+rate)^t)

IRR <- uniroot(f, interval = c(-0.99, 2))$root
IRR
```

# 5) Thời giá của tiền (TVM) – PV, FV, annuity

```
PV0 <- 100
r <- 0.08
n <- 5

FV <- PV0 * (1+r)^n
FV

# PV của 1 khoản FV
PV_back <- FV / (1+r)^n
PV_back

# PV annuity (ordinary): C mỗi năm trong n năm
C <- 10
PV_ann <- C * (1 - (1+r)^(-n)) / r
PV_ann

# FV annuity
FV_ann <- C * ((1+r)^n - 1) / r
FV_ann
```

# 6) Trái phiếu: giá, YTM, duration

Giả sử: Face=100, coupon 8%/năm, maturity 5 năm, trả coupon semiannual (freq=2).

## 6.1 Price từ YTM

```
face <- 100
c_rate <- 0.08
```

```

y <- 0.07
maturity <- 5
freq <- 2

n <- maturity * freq
cpn <- face * c_rate / freq
k <- y / freq

times <- 1:n
PV_cpn <- sum(cpn / (1+k)^times)
PV_face <- face / (1+k)^n
Price <- PV_cpn + PV_face
Price

```

## 6.2 YTM từ Price (uniroot)

```

P_obs <- Price
g <- function(y_guess){
  k <- y_guess/freq
  PV_cpn <- sum(cpn / (1+k)^times)
  PV_face <- face / (1+k)^n
  (PV_cpn + PV_face) - P_obs
}
YTM <- uniroot(g, interval = c(0, 1))$root
YTM

```

## 6.3 Duration (Macaulay & Modified)

```

# Cash flows
CF <- rep(cpn, n)
CF[n] <- CF[n] + face

disc <- (1+k)^times
PV <- CF / disc
P <- sum(PV)

D_mac <- sum((times/freq) * PV) / P
D_mod <- D_mac / (1 + k)

P
D_mac
D_mod

```

# 7) Định giá cổ phiếu bằng DDM

## 7.1 Gordon Growth Model

```

D0 <- 2
g <- 0.05
k <- 0.10

D1 <- D0 * (1+g)

```

```
P0 <- D1 / (k - g)
P0
```

## 7.2 Two-stage DDM (tính thăng)

Giả sử tăng trưởng cao  $g_1=12\%$  trong 5 năm, sau đó ổn định  $g_2=4\%$ ,  $k=10\%$ ,  $D_0=2$ .

```
D0 <- 2
g1 <- 0.12
n <- 5
g2 <- 0.04
k <- 0.10

# Dividend years 1..n
D <- numeric(n)
D[1] <- D0*(1+g1)
if(n>1){
  for(i in 2:n) D[i] <- D[i-1]*(1+g1)
}
PV_div <- sum(D / (1+k)^(1:n))

# Terminal value at year n:  $P_n = D_{n+1} / (k - g_2)$ 
Dn1 <- D[n]*(1+g2)
Pn <- Dn1/(k-g2)
PV_term <- Pn/(1+k)^n

P0_2stage <- PV_div + PV_term
P0_2stage
```

## 8) CAPM, SML

```
Rm <- rowMeans(R) # market return
```

---

### 8.3 Risk-free rate

Giả sử:

- lãi suất phi rủi ro **3%/năm**
- dữ liệu là **daily**,  $\sim 250$  ngày/năm

```
rf_annual <- 0.03
rf_daily <- rf_annual / 250
```

---

### 8.4 Excess returns

```
Ri_excess <- R - rf_daily
```

```
Rm_excess <- Rm - rf_daily
```

---

## 8.5 Ước lượng beta (cách công thức)

$$\beta_i = \frac{\text{Cov}(R_i, R_m)}{\text{Var}(R_m)}$$

```
beta <- apply(R, 2, function(x) cov(x, Rm) / var(Rm))  
beta
```

---

## 8.6 Ước lượng CAPM bằng hồi quy (cách chuẩn)

$$R_i - R_f = \alpha_i + \beta_i(R_m - R_f) + \varepsilon_i$$

Ví dụ với DAX:

```
capm_DAX <- lm(Ri_excess[, "DAX"] ~ Rm_excess)  
summary(capm_DAX)
```

Giải thích cho sinh viên:

- (Intercept) = **alpha**
  - Rm\_excess = **beta**
  - R-squared = mức độ CAPM giải thích return
- 

## 8.7 Ước lượng beta cho tất cả tài sản

### Ước lượng cho từng tài sản

```
Ri_DAX <- Ri_excess[, 1]  
  
model_DAX <- lm(Ri_DAX ~ Rm_excess)  
  
alpha_DAX <- coef(model_DAX)[1]  
beta_DAX <- coef(model_DAX)[2]
```

### Ước lượng cho toàn bộ

```
beta_lm <- apply(Ri_excess, 2, function(x) {  
  coef(lm(x ~ Rm_excess))[2]  
})
```

```

alpha_lm <- apply(Ri_excess, 2, function(x){
  coef(lm(x ~ Rm_excess))[1]
})

round(cbind(alpha = alpha_lm, beta = beta_lm), 4)

```

---

## 8.8 Expected return theo CAPM

$$E(R_i) = R_f + \beta_i(E(R_m) - R_f)$$

```

ERm <- mean(as.numeric(Rm)) * 250      # annualize market return
ERP <- ERm - as.numeric(rf_annual)      # market risk premium

ER_capm <- as.numeric(rf_annual) + as.numeric(beta_lm) * as.numeric(ERP)

ER_capm

```

#So sánh với **return thực tế**:

```

ER_real <- colMeans(R) * 250
round(cbind(Real = ER_real, CAPM = ER_capm), 4)

```

## 8.9 Vẽ SML

```

# =====
# Chuẩn bị dữ liệu
# =====
assets <- colnames(R)                      # DAX, SMI, CAC, FTSE
beta_all <- as.numeric(beta_lm)
ER_real_all <- as.numeric(ER_real)

# =====
# SML line
# =====
x_line <- seq(
  min(beta_all) - 0.1,
  max(beta_all) + 0.1,
  length.out = 200
)
y_line <- rf_annual + x_line * ERP

# =====
# Plot (set trục theo dữ liệu thực)
# =====
plot(
  x_line, y_line,
  type = "l", lwd = 2,

```

```

xlim = range(beta_all) + c(-0.1, 0.1),
ylim = range(ER_real_all) + c(-0.02, 0.02),
xlab = expression(beta),
ylab = "Expected return (annual)",
main = "Security Market Line (SML) - European Indices"
)

# =====
# 4 điểm tròn: return thực tế
# =====
points(
  beta_all,
  ER_real_all,
  pch = 16, cex = 1.4
)

# =====
# Ghi tên từng chỉ số
# =====
text(
  beta_all,
  ER_real_all,
  labels = assets,
  pos = 4,
  cex = 0.9
)

# =====
# Legend
# =====
legend(
  "topleft",
  legend = c("SML", "Realized return"),
  lty = c(1, NA),
  pch = c(NA, 16),
  bty = "n"
)

```

## 8.10 Kiểm định $\alpha = 0$ (CAPM có đúng không?)

summary(capm\_DAX)\$coefficients

- Nếu  $\alpha$  không có ý nghĩa thống kê  $\rightarrow$  CAPM không bị bác bỏ
- Nếu  $\alpha$  khác 0  $\rightarrow$  CAPM không giải thích hết return

