Financial Econometrics

- Week 3, Lecture 2 -

Estimation and Analysis of Multivariate GARCH Models

VU Econometrics and Data Science

2024-2025

Paolo Gorgi

VU Vrije Universiteit Amsterdam

Today's class

1. Estimation of multivariate GARCH
    - Maximum likelihood estimation
    - Estimation with covariance targeting
    - Estimation of CCC equation by equation
2. Financial analysis of multivariate GARCH
    - VaR of a portfolio
    - Dynamic portfolio optimization
    - Out-of-sample portfolio evaluation

Estimation of multivariate GARCH

ML estimation of multivariate GARCH (i)

Note: Multivariate GARCH models can be estimated by maximum likelihood

The ML estimator is naturally given by

$$\hat{\theta}_T = \arg\max_{\theta \in \Theta} L(y_1, ..., y_T, \theta).$$

The log-likelihood function is given by

$$L(y_1, ..., y_T, \theta) = -\frac{1}{2} \sum_{t=1}^{T} \left(\log |\boldsymbol{\Sigma}_t| + \boldsymbol{y}_t^\top \boldsymbol{\Sigma}_t^{-1} \boldsymbol{y}_t \right).$$

The time-varying conditional covariance matrix $\boldsymbol{\Sigma}_t$ is filtered using the updating equation!

ML estimation of multivariate GARCH (ii)

Example: for a bivariate DVECH(1,1) we can use the following updating equations

$$\sigma^2_{1,t} = \omega_{11} + \beta_{11}\sigma^2_{1,t-1} + \alpha_{11}y^2_{1,t-1},$$
$$\sigma^2_{2,t} = \omega_{22} + \beta_{22}\sigma^2_{2,t-1} + \alpha_{22}y^2_{2,t-1},$$
$$\sigma_{21,t} = \omega_{21} + \beta_{21}\sigma^2_{21,t-1} + \alpha_{21}y_{1,t-1}y_{2,t-1},$$

where the updating equation can be initialized by setting $\mathbf{\Sigma}_1$ equal to the sample covariance matrix!

In large samples: the ML estimator is normally distributed

$$\sqrt{T}\Big(\hat{\theta}_T - \theta_0\Big) \overset{app}{\sim} N\Big(0\ ,\ \mathcal{I}(\theta_0)^{-1}\ \Big) \quad \text{as } T \to \infty.$$

The Fisher information $\mathcal{I}(\theta_0)$ can be estimated as discussed for the univariate GARCH.

Estimating a sDVECH(1,1) with R (i)

Example: bivariate scalar DVECH(1,1)

Note: Maximizing the log likelihood with R is the same as for the univariate case!

Question: How do we write the log-likelihood function?

R files: `estimation_sDVECH.R` and `llik_fun_sDVECH.R`.

Input for `llik_fun_sDVECH()`: observed time series `x` and parameter vector `par`;

Output from `llik_fun_sDVECH()`: average log-likelihood value `llik`.

Estimating a sDVECH(1,1) with R (ii)

First line: defines name of function and inputs

```
llik_fun_sDVECH <- function(par,x){
```

Next: define sample size and parameter values with appropriate link functions

```
  w11 <- exp(par[1])
  w12 <- par[2]
  w22 <- exp(par[3])
  a <- exp(par[4])/(1+exp(par[4]))
  b <- exp(par[5])/(1+exp(par[5]))

  d <- dim(x)
  n <- d[1]
```

Estimating a sDVECH(1,1) with R (iii)

Initialize VECHt and llik value

```
VECHt <- matrix(0,nrow=n,ncol=3)
llik <- 0
C <- cov(x)
VECHt[1,] <- c(C[1,1],C[1,2],C[2,2])
```

Run a for loop

```
for(t in 2:n){
  VECHt[t,1] <- w11+b*VECHt[t-1,1]+a*x[t-1,1]^2
  VECHt[t,3] <- w22+b*VECHt[t-1,3]+a*x[t-1,2]^2
  VECHt[t,2] <- w12+b*VECHt[t-1,2]+a*x[t-1,1]*x[t-1,2]

  SIGMAt <- cbind(c(VECHt[t,1],VECHt[t,2]),c(VECHt[t,2],VECHt[t,3]))
  llik <- llik-0.5*(log(det(SIGMAt))+x[t,]%*%solve(SIGMAt)%*%t(t(x[t,])))/n
}
```

Estimating a sDVECH(1,1) with R (iv)

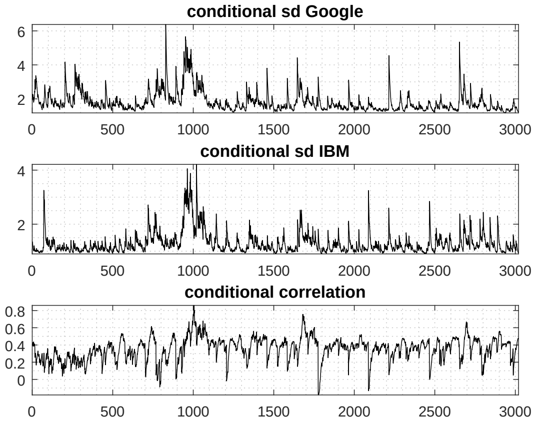

Figure: Conditional variances and correlations estimated from IBM and Google log-returns.

Estimation with covariance targeting (i)

Important: VECH can be estimated by **covariance targeting**!

Idea:

- Estimate the unconditional covariance $\mathbf{\Sigma}$ using sample variance.
- Plug-in the estimated covariance $\hat{\mathbf{\Sigma}}$ into the likelihood function.
- Optimize the likelihood over the remaining parameters.

Advantage: **covariance targeting** reduces the number of parameters in the likelihood optimization.

- Numerical optimization methods become time-consuming for large parameter vectors;
- Optimization over large parameter vectors often lead to numerical problems.

Estimation with covariance targeting (ii)

Example: sDVECH(1,1)

1. The updating equation of sDVECH(1,1) is

$$\boldsymbol{\Sigma}_t = \boldsymbol{W} + \alpha_1 \boldsymbol{y}_{t-1}\boldsymbol{y}_{t-1}^\top + \beta_1 \boldsymbol{\Sigma}_{t-1}.$$

2. The unconditional covariance $\boldsymbol{\Sigma} = \mathbb{V}ar(\boldsymbol{y}_t) = \mathbb{E}(\boldsymbol{y}_t\boldsymbol{y}_t^\top)$ is

$$\boldsymbol{\Sigma} = (1 - \alpha_1 - \beta_1)^{-1} \boldsymbol{W}, \quad \text{therefore} \quad \boldsymbol{W} = (1 - \alpha_1 - \beta_1)\boldsymbol{\Sigma}.$$

3. $\boldsymbol{\Sigma}$ is estimated by sample variance $\hat{\boldsymbol{\Sigma}} = T^{-1}\sum_{t=1}^{T} \boldsymbol{y}_t\boldsymbol{y}_t^\top$ and plugged-in into the updating equation:

$$\boldsymbol{\Sigma}_t = (1 - \alpha_1 - \beta_1)\hat{\boldsymbol{\Sigma}} + \alpha_1 \boldsymbol{y}_{t-1}\boldsymbol{y}_{t-1}^\top + \beta_1 \boldsymbol{\Sigma}_{t-1}.$$

4. Use this new updating equation to get the likelihood. This way the likelihood only depends on two parameters: α_1 and β_1.

Covariance targeting sDVECH(1,1) with R (i)

R files: likelihood function with covariance targeting in `llik_CT_sDVECH.R` and code to optimize it in `CT_estimation_sDVECH.R`.

Write likelihood: define input and output of `llik_CT_sDVECH()`

```
llik_CT_sDVECH <- function(par,x){
```

Define: parameters α_1 and β_1

```
  a <- exp(par[1])/(1+exp(par[1]))
  b <- exp(par[2])/(1+exp(par[2]))
  d <- dim(x)
  n <- d[1]
  VECHt <- matrix(0,nrow=n,ncol=3)
  llik <- 0
```

Covariance targeting sDVECH(1,1) with R (ii)

Compute the sample covariance matrix

```
C <- cov(x)
VECHt[1,] <- c(C[1,1],C[1,2],C[2,2])
```

Run *for loop* and plug-in sample variance in updating equation

```
for(t in 2:n){
   VECHt[t,1] <- C[1,1]*(1-a-b)+b*VECHt[t-1,1]+a*x[t-1,1]^2
   VECHt[t,3] <- C[2,2]*(1-a-b)+b*VECHt[t-1,3]+a*x[t-1,2]^2
   VECHt[t,2] <- C[1,2]*(1-a-b)+b*VECHt[t-1,2]+a*x[t-1,1]*x[t-1,2]
   SIGMAt <- cbind(c(VECHt[t,1],VECHt[t,2]),c(VECHt[t,2],VECHt[t,3]))
   llik <- llik-0.5*(log(det(SIGMAt))+x[t,]%*%solve(SIGMAt)%*%t(t(x[t,])))/n
 }
```

Important: the likelihood function `llik_CT_sDVECH()` can be optimized with respect to α_1 and β_1 only.

Estimation of CCC equation by equation (i)

Important: CCC model can be estimated **equation-by-equation**!

Recall: the CCC is

$$\boldsymbol{\Sigma}_t = \boldsymbol{D}_t \boldsymbol{R} \boldsymbol{D}_t,$$

$$\text{with} \quad \boldsymbol{D}_t = \text{diag}(\sigma_{1,t}, \ldots, \sigma_{n,t}), \quad \text{and} \quad \sigma_{i,t}^2 = \omega_i + \beta_i \sigma_{i,t-1}^2 + \alpha_i y_{i,t}^2.$$

Idea:

- We can estimate n univariate GARCH models separately by ML.
- The constant correlation matrix $\boldsymbol{R}$ is then estimated using the standardized residuals obtained from the univariate GARCHs.

Estimation of CCC equation by equation (ii)

Note: the steps to estimate a **CCC model equation-by-equation** are the following:

1. Estimate a univariate GARCH model for each series $\{y_{it}\}_{t=1}^{T}$, $i = 1, \ldots, n$.
2. Obtain the standardized errors from each of these series $\hat{\varepsilon}_{it} = (y_{it} - \hat{\mu}_i)/\hat{\sigma}_{it}$, $i = 1, \ldots, n$.
3. Estimate the correlation matrix from the residuals $\hat{\boldsymbol{R}} = T^{-1} \sum_{t=1}^{T} \hat{\boldsymbol{\varepsilon}}_t \hat{\boldsymbol{\varepsilon}}_t^\top$, where $\hat{\boldsymbol{\varepsilon}}_t = (\hat{\varepsilon}_{1t}, \ldots, \hat{\varepsilon}_{nt})^\top$.

Advantage: no need to optimize the likelihood function over the full parameter vector: **fast and more reliable estimation**.

Estimating a CCC with R (i)

MATLAB file: `estimation_CCC.R`

Define initial parameter values to estimate univaraite GARCH

```
alpha_ini <- 0.2
beta_ini <- 0.6
omega_ini <- var(x[,1])*(1-alpha_ini-beta_ini)
par_ini <- c(log(omega_ini),log(alpha_ini/(1-alpha_ini)),log(beta_ini/(
```

Obtain parameter estimates of univariate GARCH:

```
est1 <- optim(par=par_ini,fn=function(par)-llik_fun_GARCH(par,x[,1]))
est2 <- optim(par=par_ini,fn=function(par)-llik_fun_GARCH(par,x[,2]))
```

Estimating a CCC with R (ii)

Obtain the filtered variances from univariate GARCH:

```
n <- length(x[,1])
s1 <- rep(0,n)
s2 <- rep(0,n)

s1[1] <- var(x[,1])
s2[1] <- var(x[,2])

for(t in 2:n){
  s1[t] <- omega_hat1 + alpha_hat1*x[t-1,1]^2 + beta_hat1*s1[t-1]
  s2[t] <- omega_hat2 + alpha_hat2*x[t-1,2]^2 + beta_hat2*s2[t-1]
}
```

Estimating a CCC with R (iii)

Obtain residuals:

```
e1 <- x[,1]/sqrt(s1)
e2 <- x[,2]/sqrt(s2)
```

Calculate: the correlation between the residuals of the first and second series:

```
r <- cor(e1,e2)
```

Important: It is easy to extend this method to more than 2 series.

Financial analysis of multivariate GARCH

Financial analysis of multivariate GARCH

Question: why is all we have learned so far about multivariate GARCH models useful?

1. Assess risk of multiple assets;
2. Risk metrics for financial investment (VaR of large portfolios);
3. Portfolio optimization: decide on which assets to invest.

Portfolio of financial assets (i)

Consider a portfolio of n assets.

Let $y_{i,t}$ denotes the return of asset i at time t.

$\boldsymbol{y}_t = (y_{1,t}, \dots, y_{n,t})^\top$ is the vector of returns at time t.

$k_{it} \in [0, 1]$ is the fraction our portfolio invested in asset i at time t. k_{it} is also called the *weight* of asset i.

$\boldsymbol{k}_t = (k_{1t}, \dots, k_{nt})^\top$ is the vector of weights.

Note: the weights sum to 1, i.e. $\sum_{i=1}^{n} k_{it} = 1$.

Portfolio's return: the return of our portfolio at time t is

$$y_{p,t} = \sum_{i=1}^{n} k_{i,t} y_{i,t} = \boldsymbol{k}_t^\top \boldsymbol{y}_t.$$

Portfolio of financial assets (ii)

Multivariate GARCH: describes conditional distribution of vector of returns

$$\boldsymbol{y}_t | Y^{t-1} \sim N_n(\mathbf{0}, \boldsymbol{\Sigma}_t).$$

Hence: portfolio's conditional distribution is given by

$$y_{p,t} | Y^{t-1} \sim N(0, \sigma^2_{p,t}),$$

where:

$$\begin{aligned}
\sigma^2_{p,t} &= \mathbb{V}ar\left(y_{p,t} | Y^{t-1}\right) \\
&= \mathbb{V}ar\left(\boldsymbol{k}_t^\top \boldsymbol{y}_t | Y^{t-1}\right) \\
&= \boldsymbol{k}_t^\top \mathbb{V}ar\left(\boldsymbol{y}_t | Y^{t-1}\right) \boldsymbol{k}_t \\
&= \boldsymbol{k}_t^\top \boldsymbol{\Sigma}_t \boldsymbol{k}_t.
\end{aligned}$$

Portfolio of financial assets: VaR

Simple example ($n = 2$): two-asset portfolio

Conditional variance $\sigma^2_{p,t}$ is given by

$$\begin{aligned}\sigma^2_{p,t} &= \boldsymbol{k}_t^\top \boldsymbol{\Sigma}_t \boldsymbol{k}_t \\ &= \begin{bmatrix} k_{1,t} & k_{2,t} \end{bmatrix} \begin{bmatrix} \sigma^2_{1,t} & \sigma_{12,t} \\ \sigma_{12,t} & \sigma^2_{2,t} \end{bmatrix} \begin{bmatrix} k_{1,t} \\ k_{2,t} \end{bmatrix} \\ &= k^2_{1,t}\sigma^2_{1,t} + k^2_{2,t}\sigma^2_{2,t} + 2k_{1,t}k_{2,t}\sigma_{12,t}.\end{aligned}$$

Finally: we obtain that the conditional α-VaR of the portfolio at time t

$$\alpha\text{-VaR}_t = z_\alpha \sigma_{p,t},$$

where: z_α is the quantile of level α of a standard normal.

VaR of a portfolio: conditional mean extension (i)

Consider a time-varying conditional mean: $\boldsymbol{\mu}_t = (\mu_{1t}, \ldots, \mu_{nt})^\top$.

Conditional distribution of returns: $\boldsymbol{y}_t | Y^{t-1} \sim N_n(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$.

- Use Vector AutoRegression (VAR) for conditional mean $\boldsymbol{\mu}_t$.
- Use multivariate GARCH for conditional variance $\boldsymbol{\Sigma}_t$.
- ARMA-GARCH, ARMA-EGARCH, ARFIMA-NGARCH, etc.

Portfolio's conditional distribution: $y_{p,t} | Y^{t-1} \sim N(\mu_{p,t}, \sigma^2_{p,t})$

where $\mu_{p,t} = \mathbb{E}\left(y_{p,t} | Y^{t-1}\right) = \mathbb{E}\left(\boldsymbol{k}_t^\top \boldsymbol{y}_t | Y^{t-1}\right) = \boldsymbol{k}_t^\top \mathbb{E}\left(\boldsymbol{y}_t | Y^{t-1}\right) = \boldsymbol{k}_t^\top \boldsymbol{\mu}_t$,

and $\sigma^2_{p,t} = \boldsymbol{k}_t^\top \mathbb{V}ar\left(\boldsymbol{y}_t | Y^{t-1}\right) \boldsymbol{k}_t = \boldsymbol{k}_t^\top \boldsymbol{\Sigma}_t \boldsymbol{k}_t$.

VaR of a portfolio: conditional mean extension (ii)

Extension of simple example ($n = 2$): two assets

$$\mu_{p,t} = \boldsymbol{k}_t^\top \boldsymbol{\mu}_t = k_{1,t}\mu_{1,t} + k_{2,t}\mu_{2,t}$$

$$\sigma_{p,t}^2 = k_{1,t}^2\sigma_{1,t}^2 + k_{2,t}^2\sigma_{2,t}^2 + 2k_{1,t}k_{2,t}\sigma_{12,t}$$

Hence: the portfolio's conditional α-VaR at time t is given by

$$\alpha\text{-VaR}_t = \mu_{p,t} + z_\alpha \sigma_{p,t}$$

Note: conditional VaR sequence depends on:

1. Time-varying conditional means $\mu_{1,t}$, $\mu_{2,t}$;
2. Time-varying conditional variances $\sigma_{1,t}^2$, $\sigma_{2,t}^2$;
3. Time-varying conditional covariance $\sigma_{12,t}$;
4. Weights $k_{1,t}$ and $k_{2,t}$ (weight adjustment is crucial!).

Dynamic portfolio optimization (i)

Portfolio optimization: selection of portfolio composition or weights k_{it} that optimize the performance of the portfolio according to some measure of interest!

Example (minimize variance):

- Weights k_{1t} and $k_{2t} = 1 - k_{1t}$: two-asset portfolio;
- Measure of performance: minimize variance (risk);
- Assumption: uncorrelated returns $\sigma_{12t} = 0$.

Question: Find the weights k_{1t} and $k_{2t} = 1 - k_{1t}$ that minimizes the variance of the portfolio σ^2_{pt}.

Dynamic portfolio optimization (ii)

Answer (example):

Given that $\sigma_{12t} = 0$, we obtain that the variance of the portfolio is

$$\begin{aligned}\sigma_{pt}^2 = \mathbb{V}\mathrm{ar}(y_{pt}|Y^{t-1}) &= k_{1t}^2\sigma_{1t}^2 + k_{2t}^2\sigma_{2t}^2 + 2k_{1t}k_{2t}\sigma_{12t} \\ &= k_{1t}^2\sigma_{1t}^2 + k_{2t}^2\sigma_{2t}^2 \\ &= k_{1t}^2\sigma_{1t}^2 + (1 - k_{1t})^2\sigma_{2t}^2\end{aligned}$$

Find k_{1t} that minimizes variance: $\arg\min_{k_{1t}} \sigma_{pt}^2$

Set derivative to zero: $2k_{1t}\sigma_{1t}^2 - 2(1 - k_{1t})\sigma_{2t}^2 = 0$

Optimal weights: $k_{1t} = \sigma_{2t}^2/(\sigma_{1t}^2 + \sigma_{2t}^2)$ and $k_{2t} = \sigma_{1t}^2/(\sigma_{1t}^2 + \sigma_{2t}^2)$.

For instance: If $\sigma_{2t}^2 = 1$ and $\sigma_{2t}^2 = 2$, then $k_{1t} = 2/3$ and $k_{2t} = 1/3$.

Dynamic portfolio optimization: Sharpe ratio

Important performance measure: Sharpe ratio

$$S_{p,t} = \frac{\mu_{p,t}}{\sigma_{p,t}}$$

Optimize $\boldsymbol{k}_t$: maximize the Sharpe ration

1. Maximize portfolio returns $\mu_{p,t}$;
2. Minimize financial volatility (risk) $\boldsymbol{\Sigma}_t$;

$$\max_{\boldsymbol{k}_t} \frac{\boldsymbol{k}_t^\top \boldsymbol{\mu}_t}{\sqrt{\boldsymbol{k}_t^\top \boldsymbol{\Sigma}_t \boldsymbol{k}_t}}, \quad \text{s.t. } \sum_{i=1}^{n} k_{i,t} = 1, \; k_{it} \geq 0$$

Simple case: Sharpe ratio can be maximized analytically.

In practice: maximize Sharpe ratio using R.

Simple Example: Sharpe ratio

Example: maximize portfolio's Sharpe ratio with two assets:

$$\max_{k_{1t}} \frac{k_{1t}\mu_{1t} + (1-k_{1t})\mu_{2t}}{\sqrt{k_{1t}^2\sigma_{1t}^2 + (1-k_{1t})^2\sigma_{2t}^2 + 2k_{1t}(1-k_{1t})\sigma_{12t}}}, \quad \text{s.t. } 0 \leq k_{1t} \leq 1$$

Note: a closed form solution is known for the optimal weights when the constraint $k_{1t}, k_{2t} \geq 0$ is not imposed!

> this makes sense if you are allowed to hold a *short position* on a stock (negative weights! contrary of *long*)

$$k_{1t} = \frac{\mu_{1t}\sigma_{2t}^2 - \mu_{2t}\sigma_{12t}}{\mu_{1t}\sigma_{2t}^2 + \mu_{2t}\sigma_{1t}^2 - (\mu_{1t}+\mu_{2t})\sigma_{12t}},$$
$$k_{2t} = 1 - k_{1t}$$

Dynamic portfolio optimization with R (i)

Objective: optimize portfolio with R (maximize Sharpe Ratio)

R file: `portfolio_CCC.R`

Note:

- We consider a bivariate CCC model for the conditional covariance matrix $\boldsymbol{\Sigma}_t$;
- The conditional mean $\boldsymbol{\mu}_t$ is assumed to be constant $\boldsymbol{\mu}_t = \boldsymbol{\mu}$.

Means: `mu1` and `mu2` contain μ_1 and μ_2, respectively.

Vectors that store conditional variances: `s1`, `s2` and `s12` contain σ^2_{1t}, σ^2_{2t} and σ_{12t} respectively for $t = 1, \ldots, T$

Dynamic portfolio optimization with R (ii)

Portfolio optimization steps:

1. **Define** matrix `kt` to store the optimal portfolio weights over time.

2. **Run** *for loop* to obtain the portfolio weights at time each time t from $t = 1, \ldots, T$.

   - **Use** the function `max_SR_portfolio()` (`max_SR_portfolio.R`).
   - This function requires the expected returns $\boldsymbol{\mu}_t$ and their covariance $\boldsymbol{\Sigma}_t$ as input.
   - It returns the optimal weights that maximize the Sharpe Ratio under the constraint that all weights are non-negative.

Dynamic portfolio optimization with R (iii)

For loop:

```
kt = matrix(0,nrow=n,ncol=2)

mu1 <- mean(x[,1])
mu2 <- mean(x[,2])
mut <- cbind(mu1,mu2)

for(t in 1:n){
  SIGMAt <- cbind(c(s1[t],s12[t]),c(s12[t],s2[t]))
  kt[t,] <- max_SR_portfolio(mut,SIGMAt)
}
```

Dynamic portfolio optimization with R (iv)

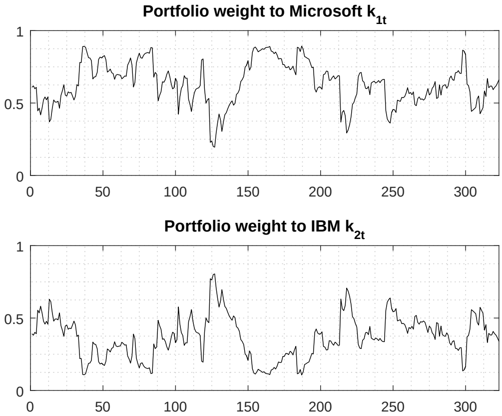

Figure: Optimal portfolio weights obtained using Microsoft and IBM monthly log-returns.

Out-of-sample portfolio evaluation (i)

In practice: The optimal weights depends on the multivariate GARCH model we use. A portfolio strategy based on the CCC model gives different weights that a strategy based on the DVECH.

Question: how can we decide which portfolio strategy is best?

Answer: we can consider a sub-sample of the observed data and see how different strategies perform in this sub-sample.

- Split dataset $\{\boldsymbol{y}_t\}_{t=1}^{T}$ into two sub-samples: *in-sample* dataset $\{\boldsymbol{y}_t\}_{t=1}^{T_1}$ and *out-of-sample* dataset $\{\boldsymbol{y}_t\}_{t=T_1+1}^{T}$.
- Use *in-sample* dataset to estimate the models and *out-of-sample* dataset to evaluate the performance of the portfolio strategies.

Out-of-sample portfolio evaluation (ii)

Out-of-sample portfolio evaluation:

1. Estimate a multivariate GARCH model using the in-sample dataset, $t = 1, \ldots, T_1$.
2. For the out-of-sample dataset, obtain an estimate of $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ using the GARCH model estimated in-sample.
3. For the out-of-sample dataset, obtain the log-returns of the optimal portfolio $\{y_{p,t}\}_{t=T_1+1}^{T}$.
4. Estimate of the sharpe ratio of the portfolio as

$$\hat{S}_p = \frac{\bar{y}_p}{\hat{\sigma}_p}, \quad \text{where} \quad \bar{y}_p = \frac{1}{T - T_1} \sum_{t=T_1+1}^{T} y_{p,t}, \quad \hat{\sigma}_p^2 = \frac{1}{T - T_1} \sum_{t=T_1+1}^{T} (y_{p,t} - \bar{y}_p)^2.$$

5. Choose strategy with largest *out-of-sample* Sharpe ratio.