Empirical Finance: Tutorial - Week 5

Tomás Cordeiro da Silva

2024-02-05

Contents

Setup

Working Directory

```
# --- confirm working directory
getwd()
```

```
## [1] "/Users/rodologie/Documents/Imperial/Term2/Empirical Finance/TA/TA5/PS4_Solutions"
```

```
# --- list existing files
list.files()
```

```
## [1] "data"                  "functions"             "tutorial_week5.html"
## [4] "tutorial_week5.pdf"    "tutorial_week5.Rmd"    "tutorial_week5.Rproj"
```

Housekeeping

```
# --- removes all objects from the the current environment
rm(list = ls())
```

Packages

```
# --- defining the packages required
packages <- c(
  "quantmod",   # download and handling of financial time-series data
  "tidyverse",  # data manipulation, transformation, and visualization
  "lubridate",  # date and time manipulation
  "patchwork",  # composition and layout of multiple ggplot2 graphs
  "latex2exp",  # LaTeX-style mathematical expressions in plots
  "readxl",     # import of Excel files (.xls, .xlsx)
  "forecast",   # time-series forecasting and model evaluation
  "tseries",    # time-series analysis, including unit root tests (adf.test),
  "knitr",
  "MASS"        # access to `ginv()`
)

# --- from the ones needed extract the ones not installed
to_install <- packages[!packages %in% installed.packages()[, "Package"]]

# --- install the packages
if (length(to_install) > 0) {
  install.packages(to_install)
}

# --- load the packages in our session
invisible(lapply(packages, library, character.only = TRUE))
```

Functions

```
source("functions/log_excess_return.R")
```

Exercise 1

Data

```
# --- return indices
equity_index   <- read.csv("data/equity_index.csv")

# --- dividend yields for the three stocks
dividend_yield <- read.csv("data/dividend_yield.csv")

# --- riskless asset
riskless_rate  <- read.csv("data/riskless_rate.csv")
```

```
# --- extracting the dates vector
date <- as.Date(equity_index$date, format = "%d/%m/%Y")
```

```
# --- converting the data frames to matrices
equity_index   <- as.matrix(equity_index[,2:ncol(equity_index)])
dividend_yield <- as.matrix(dividend_yield[,2:ncol(dividend_yield)])
riskless_rate  <- as.matrix(riskless_rate[,2:ncol(riskless_rate)])
```

OOS Economic Evalution

Maximum Expected Return

Consider an investor allocating wealth between a riskless asset and 3 risky stocks, with a target volatility of 10% per annum. The investor wishes to maximize the expected returns of its portfolio in each period t. This problem can be defined as

$$\max_{w_t} \quad \hat{r}_{p,t+1} = w_t' \hat{r}_{t+1} + (1 - w_t' l) r_{f,t} \quad s.t. \quad \sigma_p^* = \frac{10}{12 \times 100}$$

with the following solution

$$w_t^* = \frac{\sigma_p^*}{\sqrt{C_t}} \hat{\Sigma}_{t+1}^{-1} (\hat{r}_{t+1} - r_{f,t})$$
$$C_t = (\hat{r}_{t+1} - r_{f,t})' \hat{\Sigma}_{t+1}^{-1} (\hat{r}_{t+1} - r_{f,t})$$

To solve this problem we must estimate $\hat{r}_{t+1}$ — the vector of total forecast returns at time t — and $\hat{\Sigma}_{t+1}^{-1}$ — the matrix of variance-covariance between the residuals of the risky assets at time t — which take the following form

$$\hat{r}_{t+1} = \begin{bmatrix} \hat{y}_{1,t+1} + r_{f,t} \\ \hat{y}_{2,t+1} + r_{f,t} \\ \hat{y}_{3,t+1} + r_{f,t} \end{bmatrix} \qquad \hat{\Sigma}_{t+1|t} = \begin{bmatrix} \hat{\sigma}_{1,t}^2 & \hat{\sigma}_{12,t} & \hat{\sigma}_{13,t} \\ \hat{\sigma}_{21,t} & \hat{\sigma}_{2,t}^2 & \hat{\sigma}_{23,t} \\ \hat{\sigma}_{31,t} & \hat{\sigma}_{32,t} & \hat{\sigma}_{3,t}^2 \end{bmatrix}$$

In turn to obtain this, we must then calculate its components $\hat{y}_{i,t+1}$ — the excess return forecast — and $e_{i,t}$ — the residuals of the model used to forecast excess returns — with $i \in \{1, 2, 3\}$

Once this has been computed, we can plug it in the corresponding vectors to obtain w_t^* — the vector of weights at time t — which depicts how the investor should optimally divide its wealth between the two asset types and how to split it within the risky type among the 3 stocks

$$w_t^* = \begin{bmatrix} w_{1,t}^* \\ w_{2,t}^* \\ w_{3,t}^* \end{bmatrix}$$

Finally, with w_t^* we can obtain the **realized portfolio return**, which represents the actual returns by following the maximum return strategy. We simply need to plug the optimal weighting vector in the portfolio expression

$$r_{p,t+1} = w_t^{*\prime} r_{t+1} + (1 - w_t^{*\prime} l) r_{f,t}$$

With this in mind, here's what we need to do:

A) Use a predictive model to forecast excess returns and collect the model residuals for the 3 risky assets

- A.1. Compute log excess returns
- A.2. Loop an one-step ahead regression over the three risky assets
  - A.2.1. Within the loop collect the residuals
  - A.2.2. Within the loop compute the forecast

B) Compute the vector of portfolio weights

- B.1. Use the residuals collected in step A.2.1. to calculate the sample variance-covariance matrix
- B.2. Use the forecasts computed in step A.2.2. to calculate the total returns vector
- B.3. Plug B.1. and B.2. in the weight formula to obtain the optimal weight vector

C. Compute the portfolio returns

- C.1. Construct the vector of total returns
- C.2. Plug C.1. and B.3. in the realized portfolio return formula

D) Expand the window and repeat until the end of the sample

Disclaimer

In the empirical analysis, data are organized with time indexing rows and assets indexing columns. Whenever required by the portfolio optimization, row vectors are transposed into column vectors. This is a notational convention and does not affect the underlying results. That is why you might observe a slight change of notation.

A.1. Compute log excess returns

We first convert annual rates to monthly.

$$\underbrace{\begin{bmatrix} TBILL_1 \\ \dots \\ TBILL_T \end{bmatrix}}_{\text{riskless rate vector}} \longrightarrow \frac{1}{12 \times 100} \begin{bmatrix} TBILL_1 \\ \dots \\ TBILL_T \end{bmatrix} = \text{rf}$$

$$\underbrace{\begin{bmatrix} \text{yield}_{1,1} & \text{yield}_{1,2} & \text{yield}_{1,3} \\ \dots & \dots & \dots \\ \text{yield}_{T,1} & \text{yield}_{T,2} & \text{yield}_{T,3} \end{bmatrix}}_{\text{dividend yields matrix}} \longrightarrow \frac{1}{12 \times 100} \begin{bmatrix} \text{yield}_{1,1} & \text{yield}_{1,2} & \text{yield}_{1,3} \\ \dots & \dots & \dots \\ \text{yield}_{T,1} & \text{yield}_{T,2} & \text{yield}_{T,3} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \end{bmatrix} = \text{xx}$$

Let L be the lag operator (i.e. returns the previous). Then, we apply the following operations to obtain excess returns.

$$\underbrace{\begin{bmatrix} \text{stock}_{1,1} & \text{stock}_{1,2} & \text{stock}_{3,1} \\ \dots & \dots & \dots \\ \text{stock}_{T,1} & \text{stock}_{T,2} & \text{stock}_{T,3} \end{bmatrix}}_{\text{equity indexes matrix}} = \begin{bmatrix} \text{stock}_1 & \text{stock}_2 & \text{stock}_3 \end{bmatrix}_{T \times 3}$$

$$\begin{aligned}
&\longrightarrow \begin{bmatrix} log\left(\frac{\text{stock}_1}{L(\text{stock}_1)}\right) & log\left(\frac{\text{stock}_2}{L(\text{stock}_2)}\right) & log\left(\frac{\text{stock}_3}{L(\text{stock}_3)}\right) \end{bmatrix}_{T-1 \times 3} \\
&\longrightarrow \begin{bmatrix} log\left(\frac{\text{stock}_1}{L(\text{stock}_1)}\right) - \frac{L(r_f)}{1200} & log\left(\frac{\text{stock}_2}{L(\text{stock}_2)}\right) - \frac{L(r_f)}{1200} & log\left(\frac{\text{stock}_3}{L(\text{stock}_3)}\right) - \frac{L(r_f)}{1200} \end{bmatrix}_{T-1 \times 3} \\
&\longrightarrow \begin{bmatrix} NA & NA & NA \\ log\left(\frac{\text{stock}_1}{L(\text{stock}_1)}\right) - \frac{L(r_f)}{1200} & log\left(\frac{\text{stock}_2}{L(\text{stock}_2)}\right) - \frac{L(r_f)}{1200} & log\left(\frac{\text{stock}_3}{L(\text{stock}_3)}\right) - \frac{L(r_f)}{1200} \end{bmatrix}_{T \times 3} \\
&= \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \end{bmatrix}_{T \times 3} = \text{ex}
\end{aligned}$$

Note that we also lag the r_f vector as the returns on the riskless asset are predetermined - they are known one period advance.

But why excess returns?

We aim to forecast excess returns rather than total returns because the investment decision concerns the reward for bearing risk. The risk-free rate is known at the time of the decision and can always be earned without uncertainty, so it does not need to be forecast. What matters for allocating wealth to the risky asset is the expected return above this safe alternative. Forecasting excess returns isolates this risk premium, and the total expected return on the risky asset is then obtained by adding back the risk-free rate.

A.2. Loop an one-step ahead regression over the three risky assets

Let $y_{t,i}$ be the log excess return at time t of stock i — or algebraically the entry (t,i) of matrix ex — and $x_{t-1,i}$ be the monthly dividend yield at time $t-1$ of stock i — or algebraically the entry $(t-1,i)$ of matrix xx

$$\begin{aligned} y_{t,i} = \alpha + \beta x_{t-1,i} + \epsilon_{t,i} \quad & \text{with } t = 2, \dots, 120 \\ & \text{with } t-1 = 1, \dots, 119 \end{aligned}$$

Using OLS

$$\hat{\beta} = \frac{Cov(y_{t,i}, x_{t-1,i})}{Var(x_{t-1,i})} \quad \text{and} \quad \hat{\alpha} = \bar{y}_i - \hat{\beta}\bar{x}_i$$

or in matrix form

$$\underbrace{\begin{bmatrix} y_{2,i} \\ y_{3,i} \\ \dots \\ y_{120,i} \end{bmatrix}}_{\mathbf{y}_i} = \underbrace{\begin{bmatrix} 1 & x_{1,i} \\ 1 & x_{2,i} \\ \dots & \dots \\ 1 & x_{119,i} \end{bmatrix}}_{\mathbf{x}_i} \underbrace{\begin{bmatrix} \alpha \\ \beta \end{bmatrix}}_{\beta} + \underbrace{\begin{bmatrix} \epsilon_{2,i} \\ \epsilon_{3,i} \\ \dots \\ \epsilon_{120,i} \end{bmatrix}}_{\epsilon_i}$$

and let b be the vectors of OLS coefficients

$$b = \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = (X'X)^{-1}(X'Y)$$

A.2.1. Within the loop collect the residuals

$$\hat{\epsilon}_{t,i} = y_{t,i} - \hat{\alpha} - \hat{\beta} x_{t-1,i}$$

or in matrix form

$$\underbrace{\begin{bmatrix} \hat{\epsilon}_{2,i} \\ \hat{\epsilon}_{3,i} \\ \dots \\ \hat{\epsilon}_{120,i} \end{bmatrix}}_{\hat{\epsilon}_i} = \underbrace{\begin{bmatrix} y_{2,i} \\ y_{3,i} \\ \dots \\ y_{120,i} \end{bmatrix}}_{\mathbf{y}_i} - \underbrace{\begin{bmatrix} 1 & x_{1,i} \\ 1 & x_{2,i} \\ \dots & \dots \\ 1 & x_{119,i} \end{bmatrix}}_{\mathbf{x}_i} \underbrace{\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix}}_{b}$$

There will be a column vector of residuals for each stock i (that is why I have been indexing this vectors with a i), which will store in the following matrix

$$\hat{\epsilon} = \begin{bmatrix} \hat{\epsilon}_{2,1} & \hat{\epsilon}_{2,2} & \hat{\epsilon}_{2,3} \\ \hat{\epsilon}_{3,1} & \hat{\epsilon}_{3,2} & \hat{\epsilon}_{3,3} \\ \dots & \dots & \dots \\ \hat{\epsilon}_{120,1} & \hat{\epsilon}_{120,2} & \hat{\epsilon}_{120,3} \end{bmatrix} = \begin{bmatrix} \hat{\epsilon}_1 & \hat{\epsilon}_2 & \hat{\epsilon}_3 \end{bmatrix}$$

A.2.2. Within the loop compute the forecast

$$\hat{y}_{t+1,i} = \hat{\alpha} + \hat{\beta} x_{t,i}$$

There will be a forecast for each stock i, which we will store in the following vector

$$\hat{f} = \begin{bmatrix} \hat{y}_{t+1,1} \\ \hat{y}_{t+1,2} \\ \hat{y}_{t+1,3} \end{bmatrix}$$

B.1. Use the residuals collected in step A.2.1. to calculate the sample variance-covariance matrix

Note that here we are already out of the loop and have the residuals of each model stored in $\hat{\epsilon}$. Now, we just need to compute the variance and covariance among these

$$\hat{\Sigma}_{t+1|t} = \begin{bmatrix} \mathbb{Var}(\hat{\epsilon_1}) & \mathbb{Cov}(\hat{\epsilon_1}, \hat{\epsilon_2}) & \mathbb{Cov}(\hat{\epsilon_1}, \hat{\epsilon_3}) \\ \mathbb{Cov}(\hat{\epsilon_2}, \hat{\epsilon_1}) & \mathbb{Var}(\hat{\epsilon_2}) & \mathbb{Cov}(\hat{\epsilon_2}, \hat{\epsilon_3}) \\ \mathbb{Cov}(\hat{\epsilon_3}, \hat{\epsilon_1}) & \mathbb{Cov}(\hat{\epsilon_3}, \hat{\epsilon_2}) & \mathbb{Var}(\hat{\epsilon_3}) \end{bmatrix} = \begin{bmatrix} \hat{\sigma}^2_{1,t} & \hat{\sigma}_{12,t} & \hat{\sigma}_{13,t} \\ \hat{\sigma}_{21,t} & \hat{\sigma}^2_{2,t} & \hat{\sigma}_{23,t} \\ \hat{\sigma}_{31,t} & \hat{\sigma}_{32,t} & \hat{\sigma}^2_{3,t} \end{bmatrix}$$

B.2. Use the forecasts computed in step A.2.2. to calculate the total returns vector

$$\hat{r}_{t+1} = \hat{f} + \mathrm{rf}_{t,1} = \begin{bmatrix} \hat{y}_{t+1,1} + r_{f,t} \\ \hat{y}_{t+1,2} + r_{f,t} \\ \hat{y}_{t+1,3} + r_{f,t} \end{bmatrix}$$

B.3. Plug B.1. and B.2. in the weight formula to obtain the optimal weight vector

$$\mathrm{wvec} = w_t^* = \frac{\sigma_p^*}{\sqrt{C_t}} \hat{\Sigma}_{t+1}^{-1} (\hat{r}_{t+1} - r_{f,t}) \quad \text{,with} \quad C_t = (\hat{r}_{t+1} - r_{f,t})' \hat{\Sigma}_{t+1}^{-1} (\hat{r}_{t+1} - r_{f,t})$$

C.1. Construct the vector of total returns

$$rr = \begin{bmatrix} \mathrm{ex}_{t+1,1} & \mathrm{ex}_{t+1,2} & \mathrm{ex}_{t+1,3} \end{bmatrix} + \mathrm{rf}_{t,1}$$

and transpose, for alignment with the with the loop structure

$$rr' = \begin{bmatrix} \mathrm{ex}_{t+1,1} \\ \mathrm{ex}_{t+1,2} \\ \mathrm{ex}_{t+1,3} \end{bmatrix} + \mathrm{rf}_{t,1}$$

C.2. Plug C.1. and B.3. in the realized portfolio return formula

$$\begin{aligned} r_{p,t+1} &= w_t^{*\prime} r_{t+1} + (1 - w_t^{*\prime} l) r_{f,t} \\ \mathrm{rp} &= \mathrm{wvec}' rr' + (1 - \mathrm{wvec}' l) r_{f,t} \end{aligned}$$

Model

```
# --- computing log returns (keeping the first observation missing) and bringing annualized rates to mo
ex <- log_excess_return(equity_index, riskless_rate/1200)
rf <- riskless_rate/1200
xx <- dividend_yield/1200

# --- parameters
window <- 120 # window size
volstr <- 10  # target volatility in percentage per annum
delta  <- 5   # risk aversion parameter
```

```r
# --- dimensions
T <- dim(xx)[1]
N <- dim(xx)[2]

# --- initialize storage objects
mod_ww <- matrix(NA, nrow = T, ncol = N) # weight matrix
mod_rp <- matrix(NA, nrow = T, ncol = 1) # portfolio total return matrix
mod_ep <- matrix(NA, nrow = T, ncol = 1) # portfolio excess return matrix
mod_mm <- matrix(NA, nrow = T, ncol = 1) # performance measure

for (t in window:(T-1)){

  # --- storage objects
  fhat <- matrix(0,N,1)      # store excess returns forecasts
  ehat <- matrix(0,t-1,N)    # store residuals

  for (i in 1:N) {

    # -- define variables
    y <- ex[2:t,i]
    x <- cbind(1, xx[1:(t-1),i])

    # -- OLS
    b <- solve(t(x) %*% x) %*% (t(x) %*% y)

    # --- residuals
    ehat[,i] <- y - x %*% b

    # -- next period forecast
    fhat[i,] <- b[1] + b[2]*xx[t,i]

  }

  # --- construct the variance-covariance matrix
  Shat <- cov(ehat)

  # --- construct the returns vector
  rhat <- fhat + rf[t-1]

  # --- construct portfolio weights
  sigstr <- volstr/(100*sqrt(12))
  den    <- t(rhat - rf[t-1]) %*% ginv(Shat) %*% (rhat - rf[t-1])
  num    <- sigstr * ginv(Shat) %*% (rhat - rf[t-1])
  wvec   <- matrix(num / sqrt(as.numeric(den)), ncol = 1)

  # --- construct portfolio returns
  rr <- ex[t+1, , drop = FALSE] + rf[t]                # row vector of asset returns
  rr <- t(rr)                                          # conversion to column vector
  l  <- matrix(1,N,1)                                  # column vector of ones
  rp <- t(wvec) %*% rr + (1 - t(wvec) %*% l) %*% rf[t] # portfolio return
  ep <- rp - rf[t]                                     # portfolio excess return
  mm <- ((1 + rp) / (1 + rf[t]))^(1-delta)             # term inside the expected value in the CER
  # --- store the results
  mod_ww[t+1,] <- as.vector(t(wvec))
  mod_rp[t+1,] <- rp
  mod_ep[t+1,] <- ep
  mod_mm[t+1,] <- mm
}
```

Benchmark

```
# --- initialize storage objects
ben_ww <- matrix(NA, nrow = T, ncol = N) # weight matrix
ben_rp <- matrix(NA, nrow = T, ncol = 1) # portfolio total return matrix
ben_ep <- matrix(NA, nrow = T, ncol = 1) # portfolio excess return matrix
ben_mm <- matrix(NA, nrow = T, ncol = 1) # performance measure

for (t in window:(T-1)){

  # --- storage objects
  fhat <- matrix(0,N,1)      # store excess returns forecasts
  ehat <- matrix(0,t-1,N) # store residuals

  for (i in 1:N) {

    # -- define variables
    y <- ex[2:t,i]

    # -- OLS
    b <- mean(y)

    # --- residuals
    ehat[,i] <- y - b

    # -- next period forecast
    fhat[i,] <- b

  }

  # --- construct the variance-covariance matrix
  Shat <- cov(ehat)

  # --- construct the returns vector
  rhat <- fhat + rf[t-1]

  # --- construct portfolio weights
  sigstr <- volstr/(100*sqrt(12))
  den    <- t(rhat - rf[t-1]) %*% ginv(Shat) %*% (rhat - rf[t-1])
  num    <- sigstr * ginv(Shat) %*% (rhat - rf[t-1])
  wvec   <- matrix(num / sqrt(as.numeric(den)), ncol = 1)

  # --- construct portfolio returns
  rr  <- ex[t+1, , drop = FALSE] + rf[t]                # row vector of asset returns
  rr  <- t(rr)                                          # conversion to column vector
  l   <- matrix(1,N,1)                                  # column vector of ones
```

```r
  rp  <- t(wvec) %*% rr + (1 - t(wvec) %*% 1) %*% rf[t] # portfolio return
  ep  <- rp - rf[t]                                     # portfolio excess return
  mm <- ((1 + rp) / (1 + rf[t]))^(1-delta)              # certainty equivalent return

  # --- store the results
  ben_ww[t+1,] <- as.vector(wvec)
  ben_rp[t+1,] <- rp
  ben_ep[t+1,] <- ep
  ben_mm[t+1,] <- mm
}
```

Summary

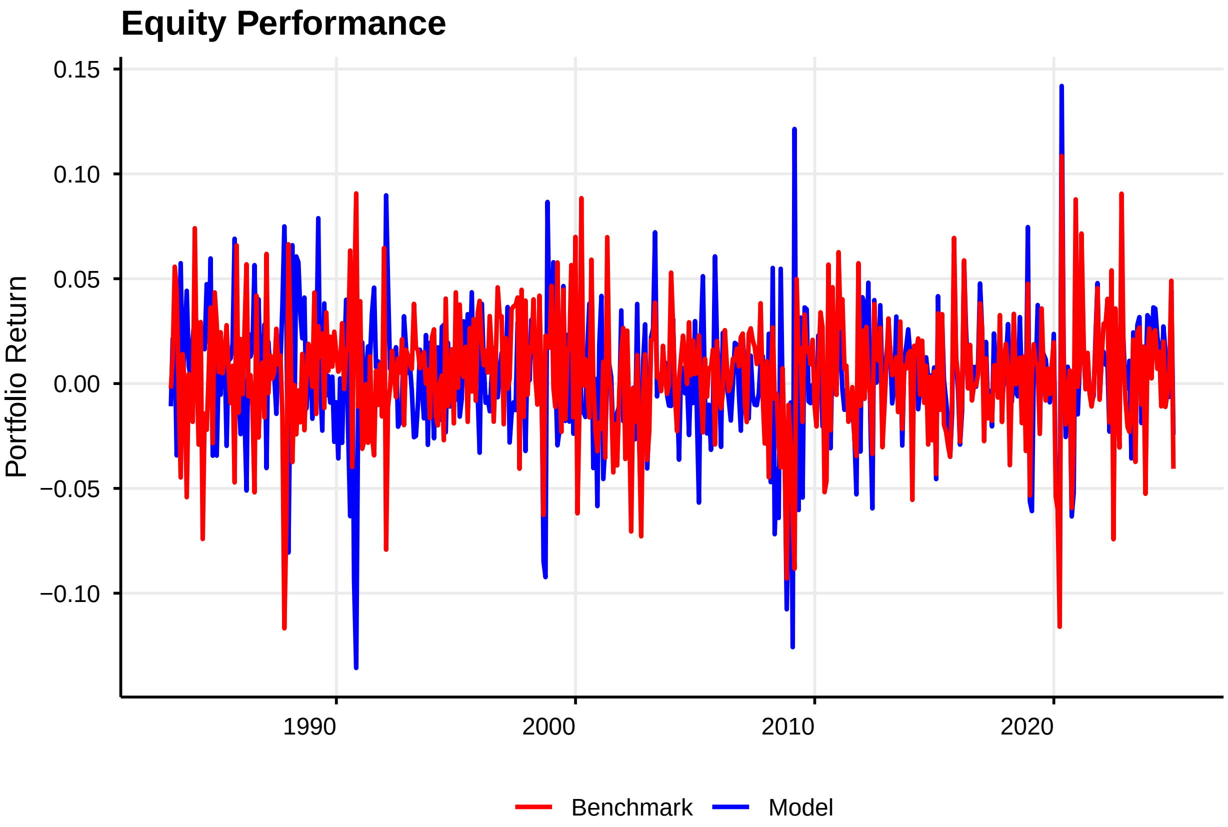
```r
mod_mean <- mean(mod_ep, na.rm = TRUE)*12*100
mod_std  <- sd(mod_ep, na.rm = TRUE)*sqrt(12)*100
mod_SR   <- mod_mean/mod_std
mod_CER  <- (1 / (1-delta)) * log(mean(mod_mm, na.rm =  TRUE))

ben_mean <- mean(ben_ep, na.rm = TRUE)*12*100
ben_std  <- sd(ben_ep, na.rm = TRUE)*sqrt(12)*100
ben_SR   <- ben_mean/ben_std
ben_CER  <- (1 / (1-delta)) * log(mean(ben_mm, na.rm = TRUE))

P        <- (mod_CER - ben_CER)*12*100*100

summary  <- data.frame(
  Metric = c(
    "mean",
    "sd",
    "SR",
    "CER"
  ),
  Model = c(
    mod_mean,
    mod_std,
    mod_SR,
    mod_CER

  ),
  Benchmark = c(
    ben_mean,
    ben_std,
    ben_SR,
    ben_CER
  )
)

latex_labels <- c(
  mean = "$\\mathbb{E}(r_{p,t+1} - r_{f,t})$",
  sd   = "$\\sqrt{\\mathbb{Var}(r_{p,t+1} - r_{f,t})}$",
  SR   = "$SR$",
  CER  = "$CER$"
)
```

```r
summary$Metric <- latex_labels[summary$Metric]

kable(
  summary,
  caption = "Economic Evaluation (Expanding Window)",
  digits = 4
)
```

Table 1: Economic Evaluation (Expanding Window)

Metric	Model	Benchmark
$\mathbb{E}(r_{p,t+1} - r_{f,t})$	0.2244	1.4710
$\sqrt{\mathbb{V}\Game\diagdown(r_{p,t+1} - r_{f,t})}$	10.5314	10.2157
SR	0.0213	0.1440
CER	-0.0022	-0.0010

```r
returns <- data.frame(
  date,
  Model = mod_rp,
  Benchmark = ben_rp
  ) %>%
  filter(!is.na(Model))

ggplot(returns, aes(x = date)) +
  geom_line(aes(y = Model, color = "Model"), linewidth = 0.8) +
  geom_line(aes(y = Benchmark, color = "Benchmark"), linewidth = 0.8) +
  labs(
    x = NULL,
    y = "Portfolio Return",
    title = "Equity Performance",
    color = NULL
  ) +
  scale_color_manual(
    values = c("Model" = "blue", "Benchmark" = "red")
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(color = "black", hjust = 1),
    axis.text.y = element_text(color = "black"),
    plot.title  = element_text(face = "bold"),
    axis.line   = element_line(color = "black"),
    axis.ticks  = element_line(color = "black"),
    panel.grid.minor.x = element_blank(),
    panel.grid.minor.y = element_blank(),
    legend.position = "bottom"
  )
```



Exercise 2

OOS Economic Evalution

Model

```
# --- computing log returns (keeping the first observation missing) and bringing annualized rates to mo
ex <- log_excess_return(equity_index, riskless_rate/1200)
rf <- riskless_rate/1200
xx <- dividend_yield/1200

# --- parameters
window <- 120 # window size
volstr <- 10  # target volatility in percentage per annum
delta  <- 5   # risk aversion parameter

# --- dimensions
T <- dim(xx)[1]
N <- dim(xx)[2]

# --- initialize storage objects
mod_ww <- matrix(NA, nrow = T, ncol = N) # weight matrix
mod_rp <- matrix(NA, nrow = T, ncol = 1) # portfolio total return matrix
mod_ep <- matrix(NA, nrow = T, ncol = 1) # portfolio excess return matrix
mod_mm <- matrix(NA, nrow = T, ncol = 1) # performance measure

j <- 0
```

```
for (t in window:(T-1)){

  # --- storage objects
  fhat <- matrix(0,N,1)      # store excess returns forecasts
  ehat <- matrix(0,t-1-j,N)    # store residuals

  for (i in 1:N) {

    # -- define variables
    y <- ex[(2+j):t,i]
    x <- cbind(1, xx[(1+j):(t-1),i])

    # -- OLS
    b <- solve(t(x) %*% x) %*% (t(x) %*% y)

    # --- residuals
    ehat[,i] <- y - x %*% b

    # -- next period forecast
    fhat[i,] <- b[1] + b[2]*xx[t,i]

  }

  # --- construct the variance-covariance matrix
  Shat <- cov(ehat)

  # --- construct the returns vector
  rhat <- fhat + rf[t-1]

  # --- construct portfolio weights
  sigstr <- volstr/(100*sqrt(12))
  den    <- t(rhat - rf[t-1]) %*% ginv(Shat) %*% (rhat - rf[t-1])
  num    <- sigstr * ginv(Shat) %*% (rhat - rf[t-1])
  wvec   <- matrix(num / sqrt(as.numeric(den)), ncol = 1)

  # --- construct portfolio returns
  rr <- ex[t+1, , drop = FALSE] + rf[t]                # row vector of asset returns
  rr <- t(rr)                                          # conversion to column vector
  l  <- matrix(1,N,1)                                  # column vector of ones
  rp <- t(wvec) %*% rr + (1 - t(wvec) %*% l) %*% rf[t] # portfolio return
  ep <- rp - rf[t]                                     # portfolio excess return
  mm <- ((1 + rp) / (1 + rf[t]))^(1-delta)             # term inside the expected value in the CER

  # --- store the results
  mod_ww[t+1,] <- as.vector(t(wvec))
  mod_rp[t+1,] <- rp
  mod_ep[t+1,] <- ep
  mod_mm[t+1,] <- mm

  j <- j + 1
}
```

Benchmark

```
# --- initialize storage objects
ben_ww <- matrix(NA, nrow = T, ncol = N) # weight matrix
ben_rp <- matrix(NA, nrow = T, ncol = 1) # portfolio total return matrix
ben_ep <- matrix(NA, nrow = T, ncol = 1) # portfolio excess return matrix
ben_mm <- matrix(NA, nrow = T, ncol = 1) # performance measure

j <- 0

for (t in window:(T-1)){

  # --- storage objects
  fhat <- matrix(0,N,1)      # store excess returns forecasts
  ehat <- matrix(0,t-1-j,N) # store residuals

  for (i in 1:N) {

    # -- define variables
    y <- ex[(2+j):t,i]

    # -- OLS
    b <- mean(y)

    # --- residuals
    ehat[,i] <- y - b

    # -- next period forecast
    fhat[i,] <- b

  }

  # --- construct the variance-covariance matrix
  Shat <- cov(ehat)

  # --- construct the returns vector
  rhat <- fhat + rf[t-1]

  # --- construct portfolio weights
  sigstr <- volstr/(100*sqrt(12))
  den    <- t(rhat - rf[t-1]) %*% ginv(Shat) %*% (rhat - rf[t-1])
  num    <- sigstr * ginv(Shat) %*% (rhat - rf[t-1])
  wvec   <- matrix(num / sqrt(as.numeric(den)), ncol = 1)

  # --- construct portfolio returns
  rr  <- ex[t+1, , drop = FALSE] + rf[t]                # row vector of asset returns
  rr  <- t(rr)                                          # conversion to column vector
  l   <- matrix(1,N,1)                                  # column vector of ones
  rp  <- t(wvec) %*% rr + (1 - t(wvec) %*% l) %*% rf[t] # portfolio return
  ep  <- rp - rf[t]                                     # portfolio excess return
  mm <- ((1 + rp) / (1 + rf[t]))^(1-delta)              # certainty equivalent return

  # --- store the results
  ben_ww[t+1,] <- as.vector(wvec)
  ben_rp[t+1,] <- rp
  ben_ep[t+1,] <- ep
  ben_mm[t+1,] <- mm

  j <- j + 1
}
```

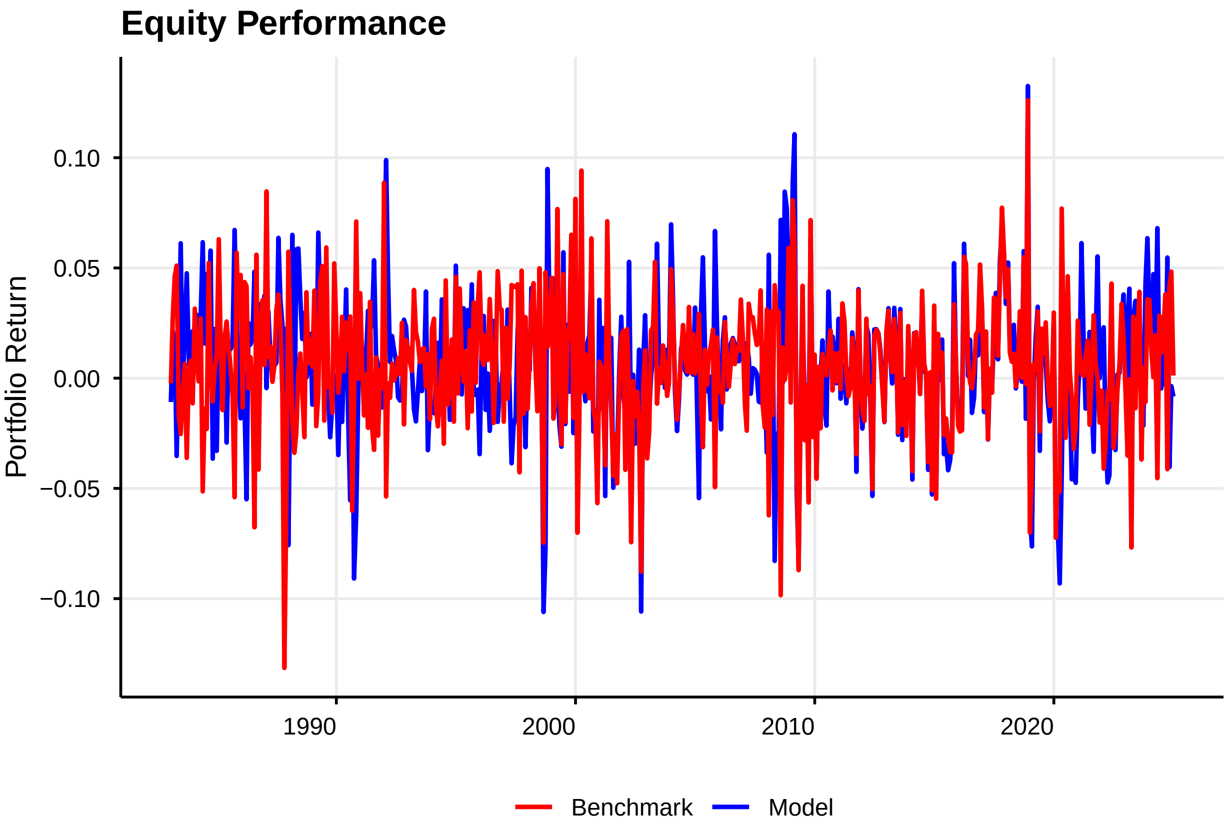
Summary

```
mod_mean <- mean(mod_ep, na.rm = TRUE)*12*100
mod_std  <- sd(mod_ep, na.rm = TRUE)*sqrt(12)*100
mod_SR   <- mod_mean/mod_std
mod_CER  <- (1 / (1-delta)) * log(mean(mod_mm, na.rm =  TRUE))

ben_mean <- mean(ben_ep, na.rm = TRUE)*12*100
ben_std  <- sd(ben_ep, na.rm = TRUE)*sqrt(12)*100
ben_SR   <- ben_mean/ben_std
ben_CER  <- (1 / (1-delta)) * log(mean(ben_mm, na.rm = TRUE))

P        <- (mod_CER - ben_CER)*12*100*100

summary  <- data.frame(
  Metric = c(
    "mean",
    "sd",
    "SR",
    "CER"
  ),
  Model = c(
    mod_mean,
    mod_std,
    mod_SR,
    mod_CER

  ),
  Benchmark = c(
    ben_mean,
    ben_std,
    ben_SR,
    ben_CER
  )
)

latex_labels <- c(
  mean = "$\\mathbb{E}(r_{p,t+1} - r_{f,t})$",
  sd   = "$\\sqrt{\\mathbb{Var}(r_{p,t+1} - r_{f,t})}$",
  SR   = "$SR$",
  CER  = "$CER$"
)

summary$Metric <- latex_labels[summary$Metric]

kable(
  summary,
  caption = "Economic Evaluation (Rolling Window)",
  digits = 4
)
```

Table 2: Economic Evaluation (Rolling Window)

Metric	Model	Benchmark
$\mathbb{E}(r_{p,t+1} - r_{f,t})$	3.5651	3.0553
$\sqrt{\mathbb{V}\eth\diagdown(r_{p,t+1} - r_{f,t})}$	10.9436	10.6940
SR	0.3258	0.2857
CER	0.0005	0.0001

```
returns <- data.frame(
  date,
  Model = mod_rp,
  Benchmark = ben_rp
  ) %>%
  filter(!is.na(Model))

ggplot(returns, aes(x = date)) +
  geom_line(aes(y = Model, color = "Model"), linewidth = 0.8) +
  geom_line(aes(y = Benchmark, color = "Benchmark"), linewidth = 0.8) +
  labs(
    x = NULL,
    y = "Portfolio Return",
    title = "Equity Performance",
    color = NULL
  ) +
  scale_color_manual(
    values = c("Model" = "blue", "Benchmark" = "red")
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(color = "black", hjust = 1),
    axis.text.y = element_text(color = "black"),
    plot.title  = element_text(face = "bold"),
    axis.line   = element_line(color = "black"),
    axis.ticks  = element_line(color = "black"),
    panel.grid.minor.x = element_blank(),
    panel.grid.minor.y = element_blank(),
    legend.position = "bottom"
  )
```


Equity Performance
Portfolio Return
0.10
0.05
0.00
−0.05
−0.10
1990
2000
2010
2020
Benchmark
Model