Portfolio Theory A1

MV Backtesting and Out-of-Sample Performance

Christopher Eason (ESNCHR001)

2025-09-11

Table of contents

Part I: Introduction to Strategy Backtesting

Question 1: Asymptotic distribution of the estimated annualized Sharpe Ratio

Show that the distribution of the estimated annualised Sharpe Ratio (SR) converges asymptotically as $y \to \infty$ to:

$$\hat{SR} \underset{y \to \infty}{\overset{a}{\sim}} N\left(\mathrm{SR}, \frac{1 + \frac{\mathrm{SR}^2}{2q}}{y}\right)$$

Definitions and Notation

- Let q be the number of return observations per year (e.g. q = 12 for monthly)
- Let y be the number of years of data.
- Let T be the total number of observations such that $T = qy$
- Let R_f be the risk-free rate
- Let R_t denote the one-period simple return of a portfolio or fund between the times $t-1$ and t. Assume $R_t \sim N(\mu, \sigma^2)$.
- Let $\mu = E(R_t) - R_f$ be the mean of the excess returns and $\sigma^2 = Cov(R_t)$ be the variance of the excess returns.
- Let SR be the annualised Sharpe Ratio that is defined as

$$SR = \frac{\mu}{\sigma}\sqrt{q} \tag{1}$$

- Since μ and σ are the population movements of the distribution of R_t however they are unobservable and must be estimated using historical data. So given a sample of historical returns $(R_1, R_2, \ldots, R_T)$, we let $\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} R_t$ and $\hat{\sigma}^2 = \frac{1}{T}\sum_{t=1}^{T}(R_t - \hat{\mu})^2$ be our estimates (Lo, 2002).
- Let $\hat{SR}$ be the annualised estimate of the Sharpe Ratio that is defined as

$$\hat{SR} = \frac{\hat{\mu}}{\hat{\sigma}}\sqrt{q} \tag{2}$$

Central Limit Theorem

In order to derive the distribution of the estimated Sharpe ratio, we begin by assuming that the portfolio returns R_t are independently and identically distributed (IID). Practically, this means that the distribution of returns at one period is the same as at any other period and that returns are not correlated across time.

Under the IID assumption, and given that $R_t \sim N(\mu, \sigma^2)$, the sample mean $\widehat{\mu}$ and sample variance $\widehat{\sigma}^2$ of returns are sums of IID random variables. The Normality assumption is what allows us to use the properties of sums of independent Normal random variables and the χ^2 distribution to derive the variances of these estimates.

For the sample mean:$\widehat{\mu} = \frac{1}{T}\sum_{t=1}^{T} R_t$ the variance of a sum of T independent random variables is $T\sigma^2$, and dividing by T^2 (because of the $1/T$ factor in the mean) gives

$$\mathrm{Var}(\widehat{\mu}) = \frac{\sigma^2}{T} \tag{3}$$

For the sample variance: $\widehat{\sigma}^2 = \frac{1}{T}\sum_{t=1}^{T}(R_t - \widehat{\mu})^2$ the CLT and the properties of the χ^2 distribution imply that

$$T\frac{\widehat{\sigma}^2}{\sigma^2} \sim \chi^2_{T-1}.$$

The variance of a χ^2 with $T - 1 \approx T$ degrees of freedom is $2T$, so rescaling back to $\widehat{\sigma}^2$ gives

$$\mathrm{Var}(\widehat{\sigma}^2) \approx \frac{2\sigma^4}{T} \tag{4}$$

Thus, taking Equation 3 and Equation 4 and considering the total number of observations $T = qy$, by the Central Limit Theorem, the distributions of $\widehat{\mu}$ and $\widehat{\sigma}^2$ converge asymptotically to Normal distributions. At this first stage, the CLT applies to sums of IID random variables, allowing us to get a joint asymptotic distribution for $\widehat{\mu}$ and $\widehat{\sigma}^2$ scaled by T.

$$\sqrt{T}(\widehat{\mu} - \mu) \underset{T\to\infty}{\overset{a}{\sim}} N(0, \sigma^2), \quad \sqrt{T}(\widehat{\sigma}^2 - \sigma^2) \underset{T\to\infty}{\overset{a}{\sim}} N(0, 2\sigma^4). \tag{5}$$

These asymptotic distributions allow us to approximate the estimation error of $\widehat{\mu}$ and $\widehat{\sigma}^2$, and note that as T increases, both variances shrink toward zero. This reflects the intuitive fact that the larger the dataset (i.e., the more periods per year q and/or the more years y), the smaller the uncertainty in our estimates.

Asymptotic Joint Distribution

We can take Equation 5 and for an asymptotic joint distribution of $\widehat{\mu}$ and $\widehat{\sigma}^2$.

$$\sqrt{\mathrm{T}}\begin{bmatrix}\widehat{\mu} - \mu \\ \widehat{\sigma}^2 - \sigma^2\end{bmatrix} \underset{T\to\infty}{\overset{a}{\sim}} \mathcal{N}\left(\begin{bmatrix}0 \\ 0\end{bmatrix}, \begin{bmatrix}\sigma^2 & 0 \\ 0 & 2\sigma^4\end{bmatrix}\right) \tag{6}$$

Delta Method

- Let $\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{\mu} \\ \hat{\sigma}^2 \end{bmatrix}$ be a column vector
- Let $\boldsymbol{\theta} = \begin{bmatrix} \mu \\ \sigma^2 \end{bmatrix}$ be a column vector
- Let $\mathbf{V}_{\boldsymbol{\theta}} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 2\sigma^4 \end{bmatrix}$ be a matrix of joint covariance-variance matrix
- Let $g(\mu, \sigma^2) = SR$ be a function that takes μ and σ as parameters, and uses Equation 1. This means that $g(\hat{\mu}, \hat{\sigma^2}) = \hat{SR}$ be a function that takes $\hat{\mu}$ and $\hat{\sigma}$ as parameters, and uses Equation 2

We apply the delta method to propagate the uncertainty from the estimators μ and σ^2 through the nonlinear function $g(\mu,\ \sigma^2) = SR$. This allows us to derive the asymptotic distribution of the Sharpe ratio estimator $\hat{SR}$ using the gradient of g and the covariance matrix of μ and σ^2 (Lo, 2002).

First, we can re-write Equation 6 as

$$\sqrt{T}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \underset{T\to\infty}{\overset{a}{\sim}} \mathrm{N}(0, \mathbf{V}_{\boldsymbol{\theta}})$$

Employing the delta method:

$$\sqrt{T}\left(g(\hat{\boldsymbol{\theta}}) - g(\boldsymbol{\theta})\right) \underset{T\to\infty}{\overset{a}{\sim}} \mathrm{N}\left(0, \left(\frac{\partial g}{\partial \boldsymbol{\theta}}\right)' \mathbf{V}_{\boldsymbol{\theta}} \frac{\partial g}{\partial \boldsymbol{\theta}}\right) \tag{7}$$

Looking at just the variance term we can compute the gradient:

$$\frac{\partial g}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial g}{\partial \mu} \\ \frac{\partial g}{\partial \sigma^2} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mu} \frac{\mu}{\sigma} \sqrt{q} \\ \frac{\partial}{\partial \sigma^2} \frac{\mu}{\sigma} \sqrt{q} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \mu} \frac{\mu}{\sigma} \sqrt{q} \\ \frac{\partial}{\partial \sigma^2} \frac{\mu}{\sqrt{\sigma^2}} \sqrt{q} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{q}}{\sigma} \\ -2\mu(\sigma^2)^{-\frac{3}{2}} \sqrt{q} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{q}}{\sigma} \\ -\frac{\mu}{2\sigma^3} \sqrt{q} \end{bmatrix}.$$

Taking this partial derivative and calculating the variance term for Equation 7.

$$\left(\frac{\partial g}{\partial \boldsymbol{\theta}}\right)' \mathbf{V}_{\theta} \frac{\partial g}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\sqrt{q}}{\sigma} & -\frac{\mu}{2\sigma^3}\sqrt{q} \end{bmatrix} \begin{bmatrix} \sigma^2 & 0 \\ 0 & 2\sigma^4 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{q}}{\sigma} \\ -\frac{\mu}{2\sigma^3}\sqrt{q} \end{bmatrix} = \begin{bmatrix} \sqrt{q}\sigma & -\mu\sigma\sqrt{q} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{q}}{\sigma} \\ -\frac{\mu}{2\sigma^3}\sqrt{q} \end{bmatrix}$$

$$\left(\frac{\partial g}{\partial \boldsymbol{\theta}}\right)' \mathbf{V}_{\theta} \frac{\partial g}{\partial \boldsymbol{\theta}} = q + \frac{q\mu^2}{2\sigma^2} = q + \frac{SR^2}{2} = q\left(1 + \frac{SR^2}{2q}\right)$$

Using Equation 1 we see that $SR = \frac{\mu}{\sigma}\sqrt{q}$ so $SR^2 = \frac{\mu^2}{\sigma^2}q$. We can substitute this back into Equation 7 and find the distribution for $\hat{SR}$, remember that $g(\hat{\boldsymbol{\theta}}) = \hat{SR}$ and $g(\boldsymbol{\theta}) = SR$

$$\sqrt{T}(\hat{SR} - SR) \underset{T\to\infty}{\overset{a}{\sim}} \mathrm{N}\left(0, q\left(1 + \frac{SR^2}{2q}\right)\right)$$

$$\hat{SR} \underset{T\to\infty}{\overset{a}{\sim}} \mathrm{N}\left(SR, \frac{q\left(1 + \frac{SR^2}{2q}\right)}{T}\right)$$

Annualisation

Since $T = yq$, we express the asymptotic variance per year by switching the limiting argument from $T \to \infty$ to $y \to \infty$ to reflect the annualized Sharpe ratio. Writing the variance in terms of years makes it explicit that the uncertainty in the estimate decreases as the number of years of data grows, which is the meaningful timescale for investors.

$$\hat{SR} \underset{y\to\infty}{\overset{a}{\sim}} \mathrm{N}\left(SR, \frac{q\left(1 + \frac{SR^2}{2q}\right)}{qy}\right)$$

$$\boxed{\hat{SR} \underset{y\to\infty}{\overset{a}{\sim}} \mathrm{N}\left(SR, \frac{1 + \frac{SR^2}{2q}}{y}\right)} \tag{8}$$

The final asymptotic variance $\frac{1+\frac{SR^2}{2q}}{y}$ shows two effects: (1) the variance shrinks with more years of data, and (2) higher Sharpe ratios increase estimation error slightly due to their dependence on both μ and σ^2.

Question 2: Question 2: Expected Maximum of a Sample of IID Normal Variables

Motivate and justify the following approximation for large N:

Theorem 1.1. Given a sample of N IID Normal random variables X_n, $n = 1, 2, \ldots, N$, where Z is the CDF of the standard normal distribution, the expected maximum of the sample is:

$$E[\max_N] := E[\max\{X_n\}].$$

The expected maximum can be approximated as:

$$E[\max_N] \approx (1-\gamma)Z^{-1}\left(1-\frac{1}{N}\right) + \gamma Z^{-1}\left(1-\frac{1}{N}e^{-1}\right)$$

for some constant γ

To approximate the expected maximum of N i.i.d. Normal random variables, we proceed in three steps.

Step 1: Show the Normal is von Mises. Using Example 3.3.29 (Embrechts et al., 1997), we first verify that the standard Normal distribution is a von Mises function with auxiliary function $a(x)$.

Step 2: Connect to the Gumbel MDA. By Proposition 3.3.25 (Embrechts et al., 1997), any von Mises function belongs to the maximum domain of attraction of the Gumbel distribution (MDA(Λ))Moreover, Proposition 3.3.28 (Embrechts et al., 1997) shows that if two distributions are tail equivalent, they share the same MDA and norming constants. Together, these results guarantee that the maxima of a Normal sample, once properly normalized, converge in distribution to the Gumbel law, which is the Gumbel case of the Fisher–Tippett–Gnedenko theorem.

Step 3: Convergence of moments.We apply Resnick's Proposition (iii) on moment convergence, we obtain that the expectation of the normalised maximum converges to the Euler–Mascheroni constant γ (Resnick, 1987). Together, these results yield the approximation $\mathbb{E}(x) \approx \alpha + \gamma\beta$ with α, β being norming constants derived from the Normal distribution.

Definition of the von Mises Function

Let F be a cumulative distribution function (CDF) with right endpoint x_F is the largest possible value that the random variable X can take (if it exists) or $+\infty$ if X is unbounded.

$$x_F = sup\{x \in \mathbb{R} : F(x) < 1\} \in (-\infty, \infty]$$

We denote the survival function by

$$\bar{F}(x) = 1 - F(x)$$

We say F is a von Mises Function if there exists a scalar $z < x_F$ and functions $a(x),\ c(x)$ satisfying the following conditions:

- $a : (z,\ x_f) \to (0,\ \infty))$ is a is a positive, absolutely continuous function (called the auxiliary function). It is a positive function that controls the rate of decay of the tail of F.
- $c : (z,\ x_F) \to (0,\ \infty)$ is a positive function such that $\underset{x \to x_f}{lim}\, c(x) = c > 0$ i.e. This means that as x gets arbitrarily close to x_F from below, the function $c(x)$ approaches a finite positive constant c. It serves as a normalizing factor to make the representation exact.

Then for all $x \in (z, x_F)$ the survival function admits the representation

$$\bar{F}(x) = c(x)exp\left(-\int_z^x \frac{1}{a(t)}dt\right) \quad z < x < x_F \tag{9}$$

Showing that the Normal Distribution is a von Mises Function

Let $X \sim N(0,\ 1)$ with the cumulative distribution function $Z(x)$ and the probability density function $\phi(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$. Denote the survival function (tail) by

$$\bar{Z}(x) = 1 - Z(x)$$

We are going to check the von Mises conditions to show that $\bar{F}(x) = \bar{Z}(x)$.

- The standard normal is unbounded above, so $x_F = +\infty$. We can choose arbitrary $z = 0$ where $z < x_F$.
- We can define the auxilliary function as

  $$a(x) = \frac{Z\bar{(x)}}{\phi(x)}$$

  where $\phi(x) > 0$ and $\bar{Z}(x) > 0$ for all x, so $a(x) > 0$.
- Defining the normalising factor c(x):
  - Using the von Mises representation, it suggests we can find

    $$c(x) = \bar{Z}(x)exp\left(-\int_z^x \frac{1}{a(t)}dt\right)$$

  - To obtain the asymptotic form of the standard normal tail we apply L'Hôpital's rule, this will give us Mills' Ratio. Firstly we check that the limits of the numerator and denominator of our proposed ratio

    $$\lim_{x\to\infty} \frac{\bar{Z}(x) \to 0}{\frac{\phi(x)}{x} \to 0}$$

Secondly, we find the first derivatives for the numerator and the denominator

$$\lim_{x\to\infty} \frac{\bar{Z}(x)}{\frac{\phi(x)}{x}} = \lim_{x\to\infty} \frac{\frac{d}{dx}(\bar{Z}(x))}{\frac{d}{dx}\frac{\phi(x)}{x}} = \lim_{x\to\infty} \frac{\frac{d}{dx}(1-Z(x))}{\frac{\phi'(x)x-\phi(x)}{x^2}} = \lim_{x\to\infty} \frac{-Z'(x)}{\frac{-x\phi(x)\cdot x-\phi(x)}{x^2}} =$$

$$= \lim_{x\to\infty} \frac{-\phi(x)}{-\phi(x)\left(1+\frac{1}{x^2}\right)} = \lim_{x\to\infty} \frac{-\phi(x)}{-\phi(x)} = 1$$

Therefore, we can use the following rates of change form instead of the raw values (i.e. Mills' Ratio)

$$\bar{Z}(x) \sim \frac{\phi(x)}{x} \quad x \to \infty \tag{10}$$

– We can apply Mills' Ratio $\bar{Z}(x) \sim \frac{\phi(x)}{x}$ as $x \to \infty$ to our auxiliary function becomes

$$a(x) = \frac{\bar{Z(x)}}{\phi(x)} = \frac{\frac{\bar{\phi(x)}}{x}}{\phi(x)} \sim \frac{1}{x} \tag{11}$$

– Therefore solving for c(x)

$$c(x) \sim \overline{Z}(x) \exp\Big(\int_z^x \frac{1}{a(t)} dt\Big) \sim \overline{Z}(x) \exp\Big(\int_z^x \frac{1}{\frac{1}{x}} dt\Big) \sim \frac{\varphi(x)}{x} \cdot \exp\Big(\frac{x^2}{2}\Big).$$

$$c(x) \sim \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \frac{1}{x} \exp\left(\frac{x^2}{2}\right) \sim \frac{1}{x\sqrt{2\pi}}$$

Note that $c(x) > 0$ can go to zero as $x \to \infty$ for unbounded distributions like the normal, however as long as it varies slower than the exponential decay i.e. the exponetial term on the right-hand side of Equation 9.

Therefore, with $a(x) \sim \frac{1}{x}$ and $c(x) \sim \frac{1}{x\sqrt{2\pi}}$, using Equation 9 we have

$$\bar{Z}(x) \sim c(x) \exp\left(-\int_z^x \frac{1}{a(t)} dt\right) \sim \frac{1}{x\sqrt{2\pi}} \exp\Big(-\frac{x^2}{2}\Big)$$

which is exactly the standard tail approximation (Mills' ratio Equation 10 i.e. $\bar{Z}(x) \sim \frac{\phi(x)}{x} \sim \frac{\frac{1}{x\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)}{x}$) for the standard normal distribution and we further show that $c(x) \sim \frac{1}{x\sqrt{2\pi}}$ decays much slower than the exponential decay i.e. $\exp\left(-\frac{x^2}{2}\right)$ as $x \to \infty$. Since the standard normal distribution satisfies all the von Mises conditions we conclude that the standard normal distribution is a von Mises function.

Proposition 3.3.25: von Mises Functions and the Max-Domain of Attraction (MDA)

Suppose the distribution function F is a von Mises function with auxiliary function $a(x)$. Then F belongs to the max-domain of attraction (MDA) of the Gumbel distribution.

- Max-domain of attraction (MDA): This means that if $X_1,\ X_2,\ ...\ X_N$, where $n = 1, 2 \ldots N$, are IID random variables with distribution F, then the properly normalized maximum
$$M_N := max\{X_1,\ X_2,\ ...\ X_N\} \underset{N\to\infty}{\overset{a}{\sim}} G \tag{12}$$
Where G is the Gumbel distribution i.e. $F \in$ MDA(Λ)
- A possible choice of norming constants for continous and strictly increasing distribution function like the normal distribution is:
$$d_N := F^{-1}\left(1 - \frac{1}{N}\right) \quad c_N := a(d_N) \tag{13}$$
where $a(x)$ is the auxiliary function of F. Then the normalized maximum converges to
$$\frac{M_N - d_N}{c_N} \underset{N\to\infty}{\overset{d}{\to}} G \tag{14}$$
where G is the standard Gumbel distribution with cdf $G(x) = exp(-e^{-x})$

Linking the von Mises–MDA Proposition to the Standard Normal Distribution

We now apply Proposition 3.3.25 to the standard normal distribution $X \sim N(0, 1)$. Since we have already shown from Equation 11 that $Z(x)$ is a von Mises function with auxiliary function $a(x) \sim \frac{1}{x}$ as $x \to \infty$, the standard normal belongs to the max-domain of attraction (MDA) of the Gumbel distribution i.e. $Z(x) \in$ MDA(Λ) Equation 12.

Proposition 3.3.28: Closure Property of MDA under Tail Equivalence

Let F and H be two distribution functions with the same right endpoint $x_F = x_H$. Suppose F belongs to the max-domain of attraction (MDA) $F \in$ MDA(Λ) with norming constants $(c_N > 0, d_N \in \mathbb{R})$.

Then G also belongs to the same MDA with the same norming constants (c_N, d_N), i.e.

$$\lim_{n\to\infty} F^n(c_N x + d_N) = \Lambda(x) \quad x \in \mathbb{R}$$

then

$$\lim_{N\to\infty} H^N(c_N x + d_N) = \Lambda(x + b) \quad x \in \mathbb{R}$$

if and only if F and H are tail equivalent with

$$\lim_{x \to x_F} \frac{\bar{F}(x)}{\bar{H}(x)} = e^b \tag{15}$$

for some finite constant $b \in \mathbb{R}$

Applying the closure proposition

Now that we have estabilished the standard normal belongs to the max-domain of attraction of the Gumbel Distribution, we need to compute the norming constants d_N and c_N to properly normalise the maximum M_N i.e. Equation 14 establishing the Gumbel limiting distribution for the standard normal maximum.

First we acknowledge the usage of the L'Hôpital's rule to give us Equation 10 showing that $\bar{Z}(x) \sim \frac{\phi(x)}{x}$ as $x \to \infty$. We apply Proposition 3.3.28 to justify replacing the standard normal tail $\bar{Z}(x) = 1 - Z(x)$, a more complicated function, with the simplier Mills' ratio $\bar{H}(x) = \frac{\pi(x)}{x}$, which is tail equivalent. This simplifies the calculation of the norming constants d_N and c_N for the maximum.

So taking $\bar{Z}(x) \sim \frac{\phi(x)}{x}$ as $x \to \infty$ and denoting $\bar{H}(x) := \frac{\phi(x)}{x}$, we can see that $\bar{Z}$ and $\bar{H}$ are tail equivalent with

$$\lim_{x \to \infty} \frac{\bar{Z}(x)}{\bar{H}(x)} = \lim_{x \to \infty} \frac{\frac{\phi(x)}{x}}{\frac{\phi(x)}{x}} = e^0 = 1$$

where $b = 0$, therefore showing Equation 15 and allowing us to use the norming constants computed for $\bar{H}$ directly for $\bar{Z}$ i.e.

$$\bar{Z}(x) \sim \bar{H}(x) \tag{16}$$

where $x \to \infty$. This simplifies the calculation for d_N and c_N for the standard normal maximum and ensures that the asymptotic Gumbel approximation holds.

Calculation d_N normalising constant

From Equation 13, we choose $d_N = H^{-1}\left(1 - \frac{1}{N}\right)$. Intuitively, d_n represents the level such that the probability of exceeding it is $\frac{1}{N}$, i.e., the level of the expected maximum in a sample of size N. Equivalently, we can express this using the survival function $\bar{H}(x) := 1 - H(x)$ which gives us gives the probability of exceeding a value x:

$$\bar{H}(d_N) = 1 - H(d_N) = \frac{1}{N}$$

Taking the negative logarithm of both sides, we obtain

$$-\ln \bar{H}(d_N) = -\ln\Big(\frac{1}{N}\Big) = \ln N$$

This step shows explicitly how the survival function transforms the original inverse CDF condition into a logarithmic equation, which is convenient for solving d_N asymptotically.

For the tail of the normal distribution, $\bar{H}(x) = \frac{\phi(x)}{x} = \frac{1}{\sqrt{2\pi}x}e^{-\frac{x^2}{2}}$, so taking the logarithm and using its properties gives

$$-\ln \bar{H}(d_N) = \frac{1}{\sqrt{2\pi}d_N}e^{-\frac{d_N^2}{2}} = -\left[-\frac{d_N^2}{2} - \ln(\sqrt{2\pi}d_N\right]$$

Finally, we can split the log to obtain

$$-\ln \bar{H}(d_N) = \frac{1}{2}d_N^2 + \ln\,(d_N) + \frac{1}{2}\ln 2\pi$$

Now we can solve for d_N, but since this equation is non-linear, we'll find the asymptotic solution for large N using a Taylor expansion. The leading-order term

$$\frac{1}{2}d_N^2 \approx \ln N \quad \Rightarrow \quad d_N \sim \sqrt{2\ln N}$$

Including the next-order correction from $\ln(d_N) + \frac{1}{2}\ln(2\pi)$, we expand the equation and solve for d_N asymptotically:

$$d_N \approx \sqrt{2\ln N} - \frac{\ln(\sqrt{2\ln N}) + \frac{1}{2}\ln(2\pi)}{\sqrt{2\ln N}}$$

Simplifying the logarithms yields the refined expansion:

$$d_N \approx \sqrt{2\ln N} - \frac{\ln(\ln N) + \ln(4\pi)}{2\sqrt{2\ln N}} + O\left((\ln N)^{-1/2}\right)$$

where the $O\left((\ln N)^{-1/2}\right)$ term represents higher-order terms in the asymptotic expansion, i.e., terms that are smaller than the retained correction and vanish relative to the main terms as $N \to \infty$.

Since these higher-order contributions are negligible for large N, we often drop them, leaving the practical approximation:

$$d_N \approx \sqrt{2\ln N} - \frac{\ln(\ln N) + \ln(4\pi)}{2\sqrt{2\ln N}} \tag{17}$$

This shows explicitly that the leading-order term $\sqrt{2\ln N}$ dominates, the next-order logarithmic correction refines the approximation, and all remaining terms fall away asymptotically.

Calculation c_N normalising constant

From Equation 13, we choose $c_N = a(d_N)$. Since $a(x) \sim \frac{1}{x}$ as $x \to \infty$ from Equation 11, we can use the leading order term for $d_N \sim \sqrt{2\ln N}$ we therefore get

$$c_N = a(d_N) \sim \frac{1}{d_N} \sim \frac{1}{\sqrt{2\ln N}} = (2\ln\ N)^{-\frac{1}{2}} \tag{18}$$

Proposition (iii): Convergence of Normalized Moments in the Gumbel MDA

Let $X_1, X_2 \ldots X_N$ be i.i.d random variables with common distribution function H with a right endpoint (x_F). Let

$$M_N := max\{X_1,\ X_2,\ \ldots X_N\} \underset{N\to\infty}{\overset{a}{\sim}}$$

denote the sample maximum. Define the norming constants

$$d_N = H^{-1}\left(1 - \frac{1}{N}\right), \quad c_N = a(d_N),$$

where $a(X)$ is the auxiliary function of H. If for some integer $(k > 0)$

$$\int_{-\infty}^{x_H} |x|^k H(dx) < \infty,$$

then

$$\underset{n\to\infty}{lim}\, \mathbb{E}\left[\left(\frac{M_N - d_N}{c_N}\right)^k\right] = \int_{-\infty}^{\infty} x^k \Lambda(dx) = (-1)^k \Gamma^{(k)}(1), \tag{19}$$

where $\Gamma^{(k)}(1) = \gamma \approx 0.5772$ is the k-th derivative of the Gamma function evaluated at $x = 1$ which is Euler–Mascheroni constant.

Application to the Standard Normal Maximum

We can use Equation 14 and Equation 19 to show the approximation to the normalized maximum of the standard normal

$$\frac{M_N - d_N}{c_N} \underset{n\to\infty}{\overset{d}{\to}} G$$

Using Proposition (iii) on momements Equation 19 to get the Gumbel limit where k=1

$$\lim_{N\to\infty} \mathbb{E}\left[\left(\frac{M_N - d_N}{c_N}\right)^1\right] \approx \int_{-\infty}^{\infty} x^1 \Lambda(dx) \approx \Gamma^{(1)}(1) \approx \gamma$$

Solving for the $\mathbb{E}[M_N]$ we get the following

$$\lim_{N\to\infty} \mathbb{E}\left[\left(\frac{M_N - d_N}{c_N}\right)^1\right] \approx \gamma \quad \Rightarrow \mathbb{E}[M_N] \approx d_N + c_N\gamma \approx (1-\gamma)d_N + \gamma(d_N + c_N)$$

Remembering that $d_N = H^{-1}\left(1 - \frac{1}{N}\right) = Z^{-1}\left(1 - \frac{1}{N}\right)$ from Equation 13 and using the result of the closure property Equation 16. Also from Equation 13 $c_N = a(d_N) = \frac{1}{Z^{-1}\left(1-\frac{1}{N}\right)}$

$$\mathbb{E}[M_N] \approx (1-\gamma)Z^{-1}\left(1 - \frac{1}{N}\right) + \gamma\left(Z^{-1}\left(1 - \frac{1}{N}\right) + \frac{1}{Z^{-1}\left(1 - \frac{1}{N}\right)}\right)$$

Equivalently,

$$\boxed{\mathbb{E}[M_N] \approx (1-\gamma)Z^{-1}\big(1 - \tfrac{1}{N}\big) + \gamma\, Z^{-1}\big(1 - \tfrac{1}{N}e^{-1}\big)} \tag{20}$$

The left-hand term is equivalent to the result Equation 17 i.e. $Z^{-1}\left(1 - \frac{1}{N}\right) = d_N$, this tells us the size of the maximum. The right-hand term is equivalent to Equation 17 + Equation 18 i.e. $Z^{-1}\left(1 - \frac{1}{N}e^{-1}\right) \sim d_N + c_N$, this gives the appropriate scaling of the maximum so that when we normalize M_N it converges in distribution to a standard Gumbel. Calculating d_N and c_N is crucial because it grounds the general limit theorem in the specific case of the Normal distribution and makes the result practically useful. These constants let us explicitly approximate $\mathbb{E}[M_N]$, turning a purely theoretical Gumbel convergence (as guaranteed by the Fisher–Tippett–Gnedenko theorem) into a concrete formula that quantifies both the location of the maximum and the scale of its fluctuations

Question 3: Minimum Backtest Length to Avoid Overfitting

Derive and discuss the minimum back test length T_{min} :
Theorem 1.2. The minimum back test length T_{min} needed to avoid selecting a strategy with an in-sample Sharpe Ratio as the average $\mathbb{E}[max_N]$ among N independent strategies with an out-of-sample Sharpe Ratio of zero is:

$$T_{min} < \frac{2\ln(N)}{\mathbb{E}[\mathbb{E}[max_N]]^2}$$

This shows that the minimum backtest length grows as the analyst tries more independent configurations of the model so as to keep the Sharpe Ratio at a given level. The key point here: the analyst that does not report the number of simulations used to select a particular strategy configuration makes it very dificult to assess the overall risk of strategy overfitting

Null Hypothesis

The goal of Theorem 1.2 is to determine the minimum backtest length required to avoid selecting a strategy that only appears profitable due to chance. In practice, researchers test many candidate strategies and naturally choose the one with the highest in-sample Sharpe ratio. If none of the strategies has genuine skill, this maximum Sharpe ratio arises purely from noise.

To capture this scenario, we impose the null hypothesis:

$$H_0 : SR = 0$$

Relating the Maximum Sharpe Ratio to the Maximum of Normal Random Variables

From Question 1 result Equation 8 we know that the annualised Sharpe ratio estimator satisfies:

$$\hat{SR} \underset{y\to\infty}{\overset{a}{\sim}} \mathrm{N}\left(SR, \frac{1+\frac{SR^2}{2q}}{y}\right)$$

Under $H_0 : SR = 0$, this simplifies to

$$\hat{SR} \underset{y\to\infty}{\overset{a}{\sim}} \mathrm{N}\left(0, \frac{1+\frac{0}{2q}}{y}\right) \underset{y\to\infty}{\overset{a}{\sim}} \mathrm{N}\left(0, \frac{1}{y}\right)$$

Thus any nonzero in-sample Sharpe ratio is entirely due to sampling variability. We can then rescale $\hat{SR}$ by $\sqrt{y}$:

$$Z := \sqrt{y}\hat{SR} \sim N(0,1) \tag{21}$$

Thus, each estimated Sharpe ratio under the null can be represented as a standard normal variable scaled by $\frac{1}{\sqrt{y}}$.

Now suppose we evaluate N independent strategies under the null, each estimated Sharpe ratio has the form

$$\hat{SR}_i = \frac{Z_i}{\sqrt{y}} \quad Z - i \sim N(0,1)$$

so the in-sample maximum Sharpe ratio across strategies the N strategies can be written as

$$\underset{1 \leq i \leq N}{max} \hat{SR} = \frac{1}{\sqrt{y}} \underset{1 \leq i \leq N}{max} Z_i$$

This expression makes explicit how the problem of evaluating the maximum estimated Sharpe ratio reduces to the classical problem of the maximum of N standard normal random variables, already solved in Question 2. Therefore, the expected maximum Sharpe ratio under the null can be written to highlight the two layers of expectation:

$$\mathbb{E}\Bigg[\underbrace{\max_i \hat{SR}_i}_{\substack{\text{outer expectation over} \\ \text{Sharpe samples}}} \Bigg] = \frac{1}{\sqrt{y}} \mathbb{E}\Bigg[\underbrace{\max_i Z_i}_{\substack{\text{inner expectation from extreme-value} \\ \text{distribution of standard normals } Z_i}} \Bigg] = \frac{1}{\sqrt{y}} \mathbb{E}[\mathbb{E}[M_N]] \tag{22}$$

where $M_N := \underset{1 \leq i \leq N}{Z}_i$ and $Z_i \sim N(0,1)$. Note the following:

- The inner expectation $\mathbb{E}[M_N]$ comes from the extreme-value approximation for the maximum of N IID standard normals, using the normalising constants d_N at Equation 17 and c_N at Equation 18 together with the approximation for $\mathbb{E}[M_N]$ at Equation 20.
- The outer expectation accounts for the sampling variability of $\hat{SR}$ itselt, which under the null scales like $\frac{1}{\sqrt{y}}$

This bridge between the two proofs is crucial, because it lets us transfer the extreme-value asymptotics from Question 2 into the analysis of the expected in-sample Sharpe ratio across N strategies.

Deriving the Minimum Backtest Length T_{min}

The next step is to use the relationship between the expected maximum Sharpe ratio and the sample length to determine the minimum backtest length needed to avoid selecting a purely lucky strategy. We can rearrange Equation 22 to solve for the number of years of data y which gives

$$y = \frac{\mathbb{E}\big[\mathbb{E}[M_N]\big]}{(\mathbb{E}[\underset{i}{max}\hat{SR_i}])^2}$$

To obtain a conservative bound for the minimum backtest length T_{min} we make 3 key substitutions:

1. Substitute y with T_{min} because we are solving for the minimum number of years of data required such that the expected maximum Sharpe ratio is not artificially large under the null hypothesis due to random luck.

2. Substitute the numerator $\mathbb{E}\big[\mathbb{E}[M_N]\big]$ with the upper bound $\sqrt{2\ln N}$ because using our final result from Question 2 Equation 14, we see that

$$\mathbb{E}[M_N] \approx (1-\gamma)Z^{-1}\big(1-\tfrac{1}{N}\big) + \gamma\, Z^{-1}\big(1-\tfrac{1}{N}e^{-1}\big)$$
$$\mathbb{E}[M_N] \approx (1-\gamma)d_N + \gamma(d_N + c_N) \approx d_N + \gamma c_N$$

   Both the d_N and $d_N + c_N$ have the same dominant, leading terms of $\sqrt{2\text{ln}N}$ but with slightly different correction terms, which vanish asymptotically compared to the dominant term. Putting this together

$$Z^{-1}\big(1-\tfrac{1}{N}\big) \sim Z^{-1}\left(1-\tfrac{1}{N}e^{-1}\right) \sim \sqrt{2\text{ln}N}$$

   When substituting $\sqrt{2\ln N}$ into the numerator of the T_{min} formula, we want an effective asymptotic ceiling that the expected maximum cannot grow faster than.

$$\mathbb{E}[M_N] \lesssim \sqrt{2\ln N} \Rightarrow \mathbb{E}[\mathbb{E}[M_N]] \lesssim \sqrt{2\ln N}$$

   The $\lesssim$ turns into a strict $<$ in practice because using the ceiling slightly overestimates the true expectation, meaning the actual required backtest length is strictly less than the bound computed. This turns the equality into an inequality of:

$$T_{min} < \frac{2\ln N}{(\mathbb{E}[\max_i \hat{SR_i}])^2}.$$

3. Substitute the denominator $\mathbb{E}[\underset{i}{max}\hat{SR_i}]$ with $\mathbb{E}\big[\mathbb{E}[M_N]\big]$ because under the null H_0 : $SR = 0$, the observed in-sample Sharpe ratio that drives selection bias is precisely given by the scaled expected maximum across the N strategies shown by Equation 22 i.e.

$$T_{min} < \frac{2\ln N}{(\frac{1}{\sqrt{y}}\,\mathbb{E}\big[\mathbb{E}[M_N]\big])^2} = \frac{2\ln N}{(\mathbb{E}\big[\mathbb{E}[M_N]\big])^2}y$$

   Here, y acts purely as a scaling factor. Since we are primarily interested in a conservative ceiling for the expected maximum, we can drop y to simplify the inequality. This leads directly to the final, conservative bound:

$$\boxed{T_{min} < \frac{2\ln N}{(\mathbb{E}[\mathbb{E}[M_N]])^2}}$$

This final form clearly highlights that T_{min} is bounded above by the square of the asymptotic ceiling of the expected maximum, ensuring a safe estimate for the minimum backtest length.

Discussion: Implications of the Minimum Backtest Length

Overfitting Risk Increases with Strategy Exploration The bound on T_{min} directly highlights that as the number of independent strategies N grows, the required backtest length increases logarithmically. Practically, this means that testing more configurations or tuning more parameters without extending the data history proportionally significantly raises the risk of selecting a strategy that appears profitable purely by chance. Analysts must be transparent about how many variations they tested to allow realistic assessment of overfitting risk.

Setting Realistic Expectations for Backtest Performance Theorem 1.2 quantifies the minimum amount of data needed to meaningfully evaluate a strategy's Sharpe ratio. Even if an in-sample Sharpe ratio looks impressive, this result reminds practitioners that without sufficient historical data, apparent "skill" could simply reflect noise. This provides a concrete metric to calibrate confidence in backtest results and avoid over-interpreting short-term performance.

Guiding Practical Backtest Design and Reporting. The inequality $T_{min} \lesssim \frac{2\ln N}{(\mathbb{E}[\mathbb{E}[M_N]])^2}$ offers a practical tool for designing backtests: it informs how much data is required relative to the number of strategies being evaluated. From a reporting standpoint, it encourages analysts to disclose not only the best in-sample performance but also the number of trials and the backtest length, helping decision-makers gauge the robustness of the strategy before committing capital.

Part II: Backtest Performance of the Tangency Portfolio

Data Pre-processing

1. Data Import

   The dataset, stored in the Excel file `PT-DATA-ALBI-JIBAR-JSEIND-Daily-1994-2017.xlsx`, was loaded into MATLAB. Each sheet in the file was imported as a timetable using readtimetable, and stored in a cell array for structured handling. This step preserved the time-series format of the financial data, which is essential for later portfolio return calculations.

2. Filtering and Preprocessing

   A whitelist of relevant tickers was defined, including the short-term risk-free proxy (RATESTEFI), the ALBI index, government bonds, and major JSE equity indices. Within the equity and bond sheets, only Total Return Index (TRI) columns were retained, while irrelevant fields such as SOURCE68779 and Var2 were dropped. Rows containing initial NaNs were excluded, and string-based numerical values were converted to numeric format to ensure compatibility.

3. Merging Data

   The datasets were merged into a single timetable using the synchronize function. Variable names were harmonised to standard finance terminology, for example renaming RATESTEFI to STEFI, RATEJ2Y4 to JIBAR, and J203 to ALSI.

4. Resampling

   The raw dataset was provided at daily frequency. It was resampled to monthly frequency using the Financial Toolbox function convert2monthly.

5. Handling Missing Data

   - Initial Missing Data Visualisation: The first plot provides a baseline view of the dataset's missing values.
   - Fill Missing Data with Zero-Order Hold: A forward-fill approach (Last Observation Carried Forward) is applied to replace missing entries with the most recent valid observation.
   - Remove Rows Using Best Proxy Asset: To further clean the dataset, rows containing missing values in the asset with the fewest NaNs (used as a proxy) are removed.
   - Trim Leading Rows with NaNs and Final: The dataset is trimmed to remove initial rows containing missing values at the start of the time series.

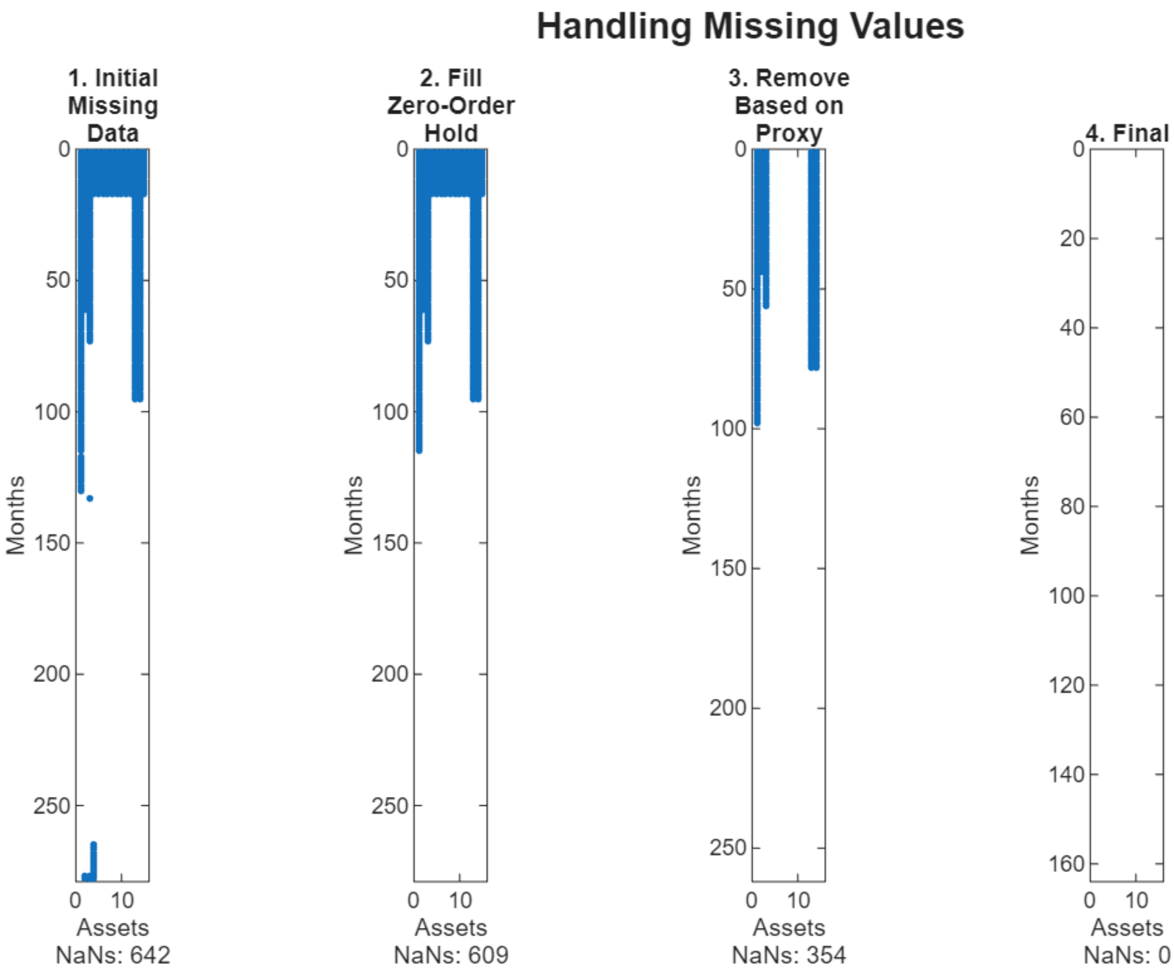

Figure 1: Missing Data Cleaing

Table 1: Summary of Dataset Information

Rows	Cols	Starting Month	Ending Month	Tickers
163	14	31-Aug-2003	28-Feb-2017	ALBI, JIBAR, STEFI, J510, J520, J530, J540, J550, J560, J580, J590, J331, J330, ALSI

Assumptions

Geometric Returns: We will be using geometric (continuous) returns as they correctly capture compounding and provide an accurate measure of portfolio growth, consistent with a mutual fund–style approach. Arithmetic returns ignore compounding, can misrepresent long-term performance, and are better suited to short-term hedge fund reporting. Continuous returns are also time-additive, making them more practical for multi-period analysis.

Choice of Risk-Free Rate STEFI: We use the Short-Term Fixed Interest (STEFI) index as the risk-free rate, since it reflects returns on tradable money market instruments and is therefore practical for portfolio analysis. By contrast, the Johannesburg Interbank Agreed Rate (JIBAR) measures interbank funding costs and cannot be directly traded, making it less suitable as an investable risk-free proxy. JIBAR will, however, be tested as an additional benchmark, given that its theoretically closer to a true risk-free rate.

Tickers we Excluded: We exclude cash-like instruments (STEFI, JIBAR) and broad market indices (ALFI, J330, J331) from the efficient frontier and portfolio optimization. Cash instruments have negligible risk and return, which makes their inclusion trivial, while market indices represent aggregates of the underlying assets and would introduce redundancy, double-counting, and distortions in the efficient frontier. The focus is therefore on investable individual bonds to accurately reflect diversification opportunities.

Monthly JIBAR Conversion: The JIBAR rate is quoted as a 3-month yield, which in reality should remain fixed for each 3-month period. In the dataset, some consecutive months show different 3-month rates due to reporting or interpolation. For consistency with the monthly portfolio returns, each monthly 3-month JIBAR value was converted to a 1-month equivalent using the standard compounding formula:

$$r_{\text{monthly}} = (1 - r_{3\text{m}})^{\frac{1}{3}} - 1$$

This approach preserves the month-to-month variation in the dataset, even though some differences do not reflect actual changes in the 3-month fixing.

Experiment 1 : In-Sample and Out-Of-Sample Sharpe Ratios

Initial Set-up

The dataset is split into a training set comprising the first 70% of observations and a test set comprising the remaining 30%. The analysis uses continuous returns for the selected asset tickers, excluding benchmark and risk-free series (STEFI, JIBAR, ALSI, J330, J331). The Short-Term Fixed Interest (STEFI) index is used as the risk-free rate for portfolio calculations.

Table 2: Training and Test Sample Sizes for 0.7 Split with Corresponding Time Periods

Data Set	Number of Months	Time Period
Training	114	31-Aug-2003 to 31-Jan-2013
Test	49	28-Feb-2013 to 28-Feb-2017

Training To Find Optimal Weights

To obtain the training portfolio weights, we focus on just calculating the weights for the Tangency Portfolio by maximising the sharpe ratio, which enforces long-only positions and fully invested portfolios while optimizing for target returns. In arriving at this step, the preliminary stages ensure that key considerations are accounted for: Step 1 explores a fully invested portfolio with varying risk aversion to understand the shape of the efficient frontier and the trade-off between risk and return; Step 2 excludes cash (STEFI, JIBAR) and market indices

(ALFI, J330, J331) to focus on truly investable assets and recalculates their statistical properties; Step 3 introduces non-negativity constraints to enforce realistic long-only positions. Progressing through these steps ensures that the final training weights are practical, fully invested, long-only, and optimized for achievable returns.

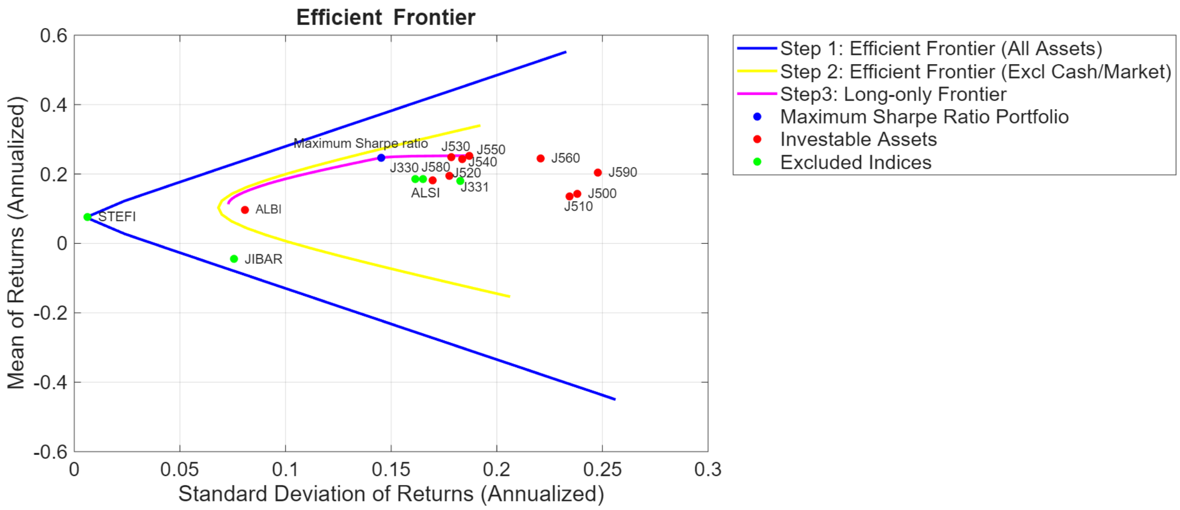

Figure 2: Efficient Frontier steps

STEFI exhibits the lowest variance, making it a practical proxy for the risk-free rate, while JIBAR occasionally shows negative returns due to market fluctuations. The investable assets cluster in similar volatility and mean return ranges, reflecting comparable risk-return profiles. The downward slope of the Step 1 and Step 2 frontiers occurs because short-selling is allowed, letting high-risk assets reduce overall variance. The all-assets frontier peaks at STEFI and is nearly linear because cash dominates low-risk allocations, whereas excluding cash and market indices shifts the peak to ALBI and produces a rounder curve due to more balanced trade-offs among risky assets. In Step 3, imposing long-only constraints prevents negative weights, flattening the downward slope and slightly lowering the frontier relative to Step 2; the left side aligns with low-risk ALBI, while the right side is dominated by higher-return assets J530, J540, and J550, with notable contributions from J520 and J560.

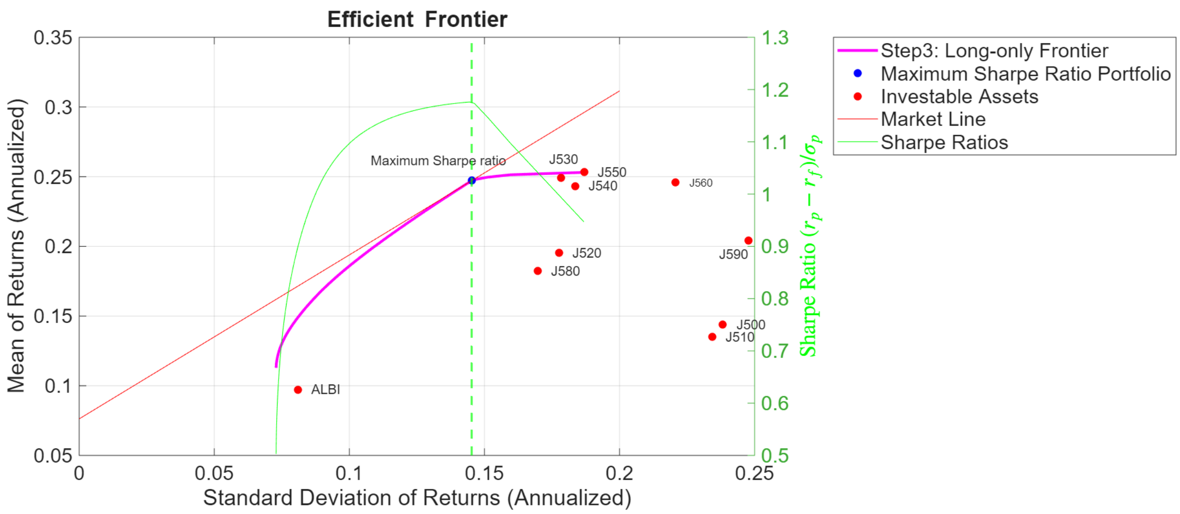

Figure 3: Efficient Frontier with Market Line and Sharpe Ratio's

The graph highlights how portfolio choices interact with risk and return. The maximum Sharpe ratio portfolio aligns with the peak of the Sharpe ratio curve, confirming it as the tangency portfolio and showing the most efficient trade-off between risk and return. This maximum portfolio lies closer to the investeable assets J530, J550 and J540, while ALBI, along with J510 and J590, are among the farthest away, possibly because their risk-return profiles differ significantly from the optimal combination, making them less influential in the tangency portfolio. Since STEFI represents short-term, low-risk instruments and serves as the risk-free rate, it anchors the Sharpe ratio, pulling the tangency portfolio toward assets with higher returns and moderate risk rather than toward low-risk assets like ALBI.

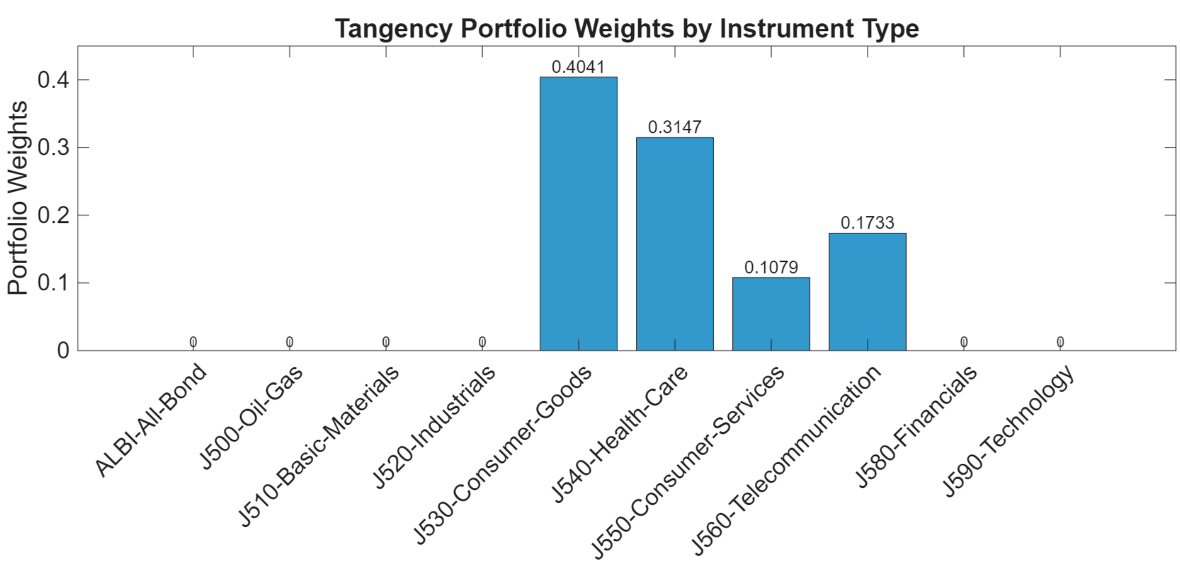

Figure 4: Optimal Tangency Portfolio weights from training

Table 3: Portfolio Weights (Final Allocation)

ALBI	J500	J510	J520	J530	J540	J550	J560	J580	J590
0	0	0	0	0.4041	0.3147	0.1079	0.1733	0	0

The optimal Sharpe ratio weights for the 0.7 split (2003–2013) concentrate entirely in J530 (Consumer Goods), J540 (Health Care), J550 (Consumer Services), and J560 (Telecommunications), reinforcing that the tangency portfolio lies closest to J530–J550–J540 and highlighting these sectors' strong risk-return balance. The exclusion of ALBI aligns with its limited contribution once the risk-free rate (STEFI) anchors the Sharpe ratio. J560's inclusion, despite higher volatility, complements the other sectors through diversification. These allocations make sense in the context of 2003–2013, which included the 2008–2009 global financial crisis and the subsequent recovery in South Africa. Consumer-oriented and defensive sectors like J530–J550 offered more stable returns, while Health Care and Telecommunications captured growth during the recovery, explaining their prominence in the optimized weights.

Testing Constant-Mix (CM) and Buy-Hold (BH) Strategies

The constant mix strategy keeps the portfolio weights fixed by continuously rebalancing back to the original allocation. At each step, asset returns are averaged and multiplied by the chosen weights to calculate expected return, while risk is measured from the weighted covariance of returns. This mimics an investor who regularly adjusts holdings to stay aligned with target weights. Its advantage is controlled diversification and risk, while its drawback is higher trading costs and potentially missing gains from strong asset trends.

The buy-and-hold strategy starts by investing a notional amount (set to 1) across assets according to the initial weights. Each period, the value of each asset grows with its return, and no rebalancing is done—the portfolio simply tracks the changing values of the assets over time. The portfolio value is the sum of all asset values, and the return series is calculated from changes in this total value. Intuitively, this mimics an investor who sets weights once and holds them. Its main advantage is simplicity and low trading costs, while its drawback is that the portfolio can drift toward a few outperforming assets, increasing concentration risk.

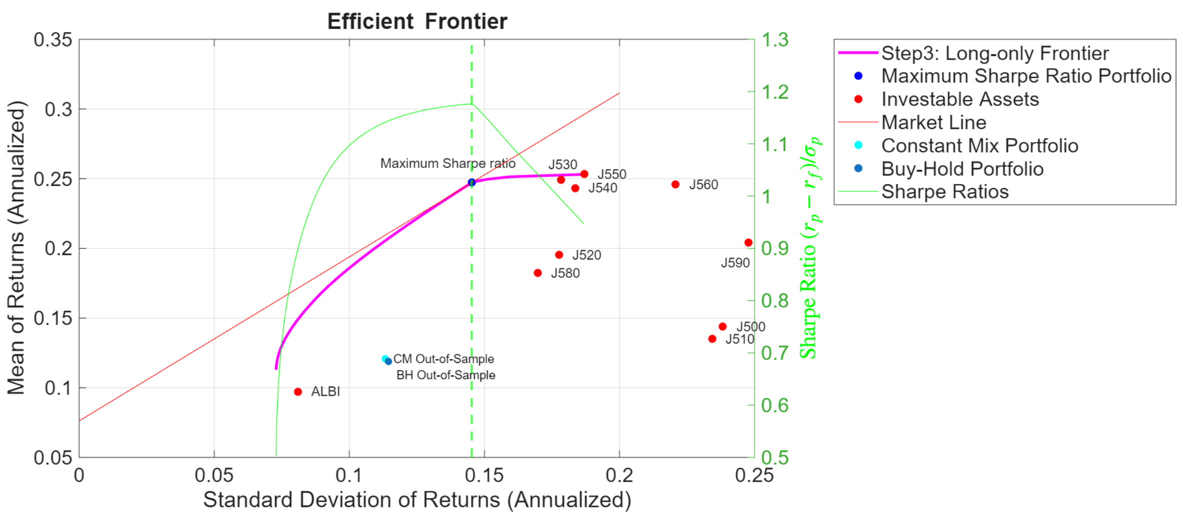

Figure 5: Efficient Frontier with Market Line and Sharpe Ratio's

Table 4: Portfolio Performance Metrics

Portfolio	Mean	Variance	Sharpe Ratio
In-Sample SR	0.0206	0.0018	1.1765
Out-of-Sample CM	0.0100	0.0011	0.5486
Out-of-Sample BH	0.0099	0.0011	0.5274

Out-of-sample, the CM and BH strategies deliver very similar returns and risk, reflecting that both start from the same in-sample maximum Sharpe ratio weights and the market exhibited limited trends during the test period, so neither rebalancing (CM) nor drift (BH) substantially altered performance. Their position between the investable assets and ALBI shows a balanced outcome: returns are higher than the low-risk bond index but lower than the top-performing assets, while variance is reduced through diversification yet remains above ALBI's level. The closer proximity to ALBI compared with the tangency portfolio or full efficient frontier reflects the conservative, long-only allocations embedded in the initial weights, which naturally anchor the portfolios toward lower-risk, bond-like assets rather than the high-return sectors (J530–J550–J540) emphasized in-sample. This pattern suggests that out-of-sample market conditions did not favor the high-return sectors as strongly as in-sample, highlighting both the stability of CM and BH strategies and the sensitivity of optimized portfolios to shifts in market behavior.

Additional Statistical

This additional test evaluates the performance of portfolios across different training-to-test splits. For each split, the data is divided into a training set and a test set, and the maximum Sharpe ratio (tangency) portfolio is computed on the training set. The resulting weights are then applied to the test set to calculate out-of-sample performance for both constant mix (CM) and buy-and-hold (BH) strategies.

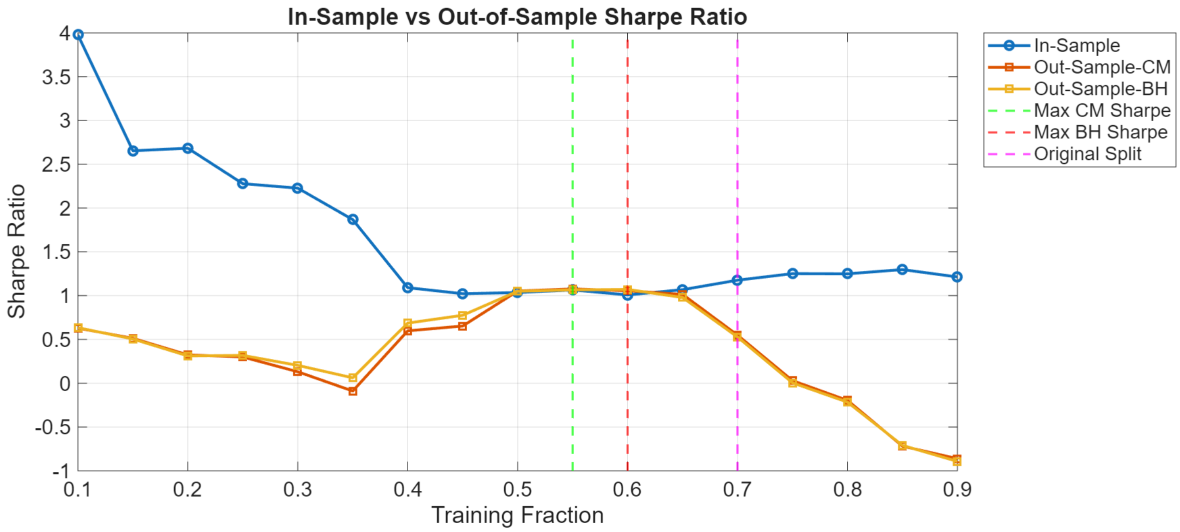

Figure 6: Sharpe Ratio performance of In-Sample and Both Constant Mix and Buy-Hold Strategies with their maximums

Table 5: Sharpe Ratios with Descriptive Training Splits

Training Split Description	In-Sample Sharpe	Out CM Sharpe	Out BH Sharpe
CM Max Sharpe Ratio Split 0.55	1.065	1.076	1.065
BH Max Sharpe Ratio Split 0.60	1.008	1.056	1.071
Original In-Sample Split 0.7	1.177	0.549	0.527

The most significant difference between CM and BH occurs for training splits between 0.25 and 0.5, where the smaller training sets make rebalancing (CM) slightly more effective than passive drift (BH). For all larger training splits, CM and BH produce nearly identical Sharpe ratios, reflecting that with sufficient data, rebalancing has little impact on performance. The only time out-of-sample Sharpe slightly exceeds the in-sample Sharpe is for splits around 0.55–0.6, likely because moderate training periods produce weights that generalize well to the test set. However, at a larger training fraction (0.7), out-of-sample Sharpe drops sharply for both strategies despite a high in-sample Sharpe, indicating overfitting. The largest out-of sample Sharpe Ratio occurs using Constant Mix (SR = 1.076) strategy with a 0.55 split, identifying the best set of weights to use going forward.

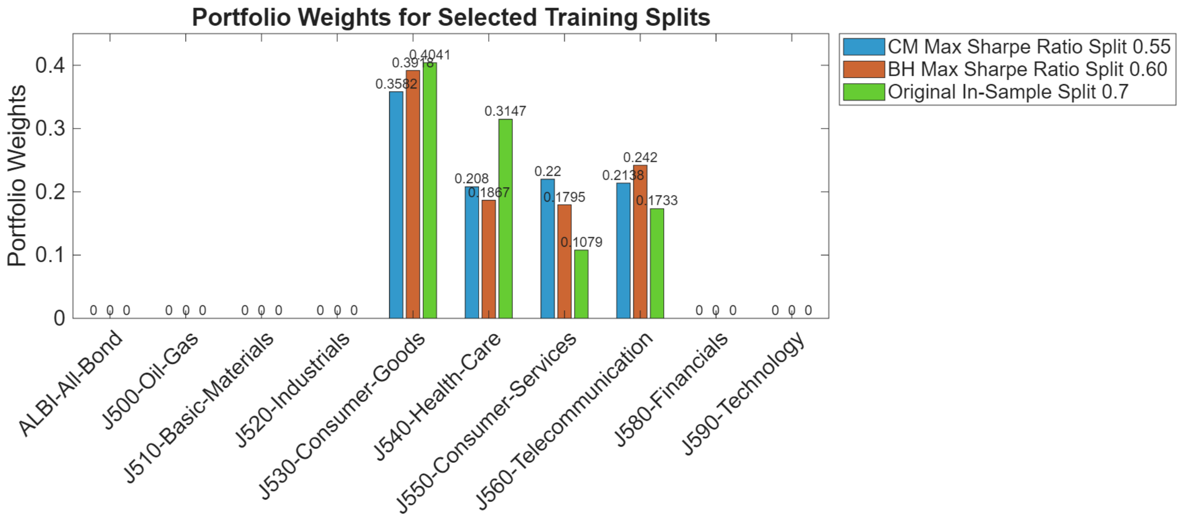

Figure 7: Portfolio weights of the highlighted splits

Table 6: Portfolio Weights for Different Training Fractions

TrainFraction	ALBI	J500	J510	J520	J530	J540	J550	J560	J580	J590
0.55	0	0	0	0	0.3582	0.2080	0.2200	0.2138	0	0
0.60	0	0	0	0	0.3918	0.1867	0.1795	0.2420	0	0
0.70	0	0	0	0	0.4041	0.3147	0.1080	0.1733	0	0

The maximum out of sample Sharpe ratio weights for the 0.55 (CM) and 0.6 (BH) splits focus on the same four sectors as the original 0.7 split, i.e. Consumer Goods (J530), Health Care (J540), Consumer Services (J550), and Telecommunications (J560), but at different levels. The 0.55 split emphasizes Consumer Services, reflecting its resilience during the 2003–2011 period, which included the global financial crisis, while the 0.6 split favors Consumer Goods and Telecommunications, capturing post-crisis recovery trends. Both splits avoid ALBI and other cyclical sectors, and the differences in weighting helped it match stable, risk-adjusted sectors during turbulent periods and explains why the 0.55 split achieved the highest out-of-sample Sharpe.

Experiment 2

Initial Set-up

We analyze the investable assets excluding cash (STEFI, JIBAR) and certain indices (ALSI, J330, J331), using continuous returns and STEFI as the risk-free rate. A rolling window of 30% of the dataset is applied to evaluate portfolio performance over time.

Table 7: Training and Test Sample Sizes with Corresponding Time Periods

Data Set	Number of Months	Time Period
Training	48	31-Aug-2003 to 31-Jul-2007
Test	115	31-Aug-2007 to 28-Feb-2017

Testing Rolling Window, Constant Mix and Buy-Hold within a timeseries framework

The rolling window strategy updates portfolio weights at each step using only the most recent subset of data. At each window, the maximum Sharpe ratio portfolio is recalculated from the training returns, and out-of-sample performance is evaluated on the next observation. This mimics an investor who continually adapts allocations to recent market conditions. Its advantage is responsiveness to changing trends, while its drawback is higher sensitivity to short-term noise, which can reduce stability in risk-adjusted returns.

In Experiment 2, at each step after the initial window, CM uses the fixed weights from the first window, BH holds the same weights without rebalancing, and RW updates weights based on the current window's data. In-sample Sharpe ratios are calculated from the training window, while out-of-sample performance is measured using the following month's returns. This setup allows us to compare how well each strategy responds to changing market conditions as a timeseries.

Figure 8: Portfolio weights of the highlighted splits

Table 8: Starting Portfolio Weights For Each Strategy

ALBI	J500	J510	J520	J530	J540	J550	J560	J580	J590
0	0	0.2391	0	0.2649	0	0	0.2567	0.2394	0

The initial portfolio concentrates on Basic Materials (J510), Consumer Goods (J530), Telecommunications (J560), and Financials (J580), with no exposure to bonds (ALBI) or more defensive sectors. This allocation makes sense in the pre-2008 context: from 2003 to 2007, South Africa benefited from the global commodity boom, fueling Basic Materials; rising incomes boosted consumer demand, supporting Consumer Goods; rapid telecom expansion drove Telecommunications and easy credit conditions supported strong financial sector growth. In this environment of optimism and high growth, riskier, growth-oriented assets offered the most attractive balance of risk and return, while safer assets were deprioritized.

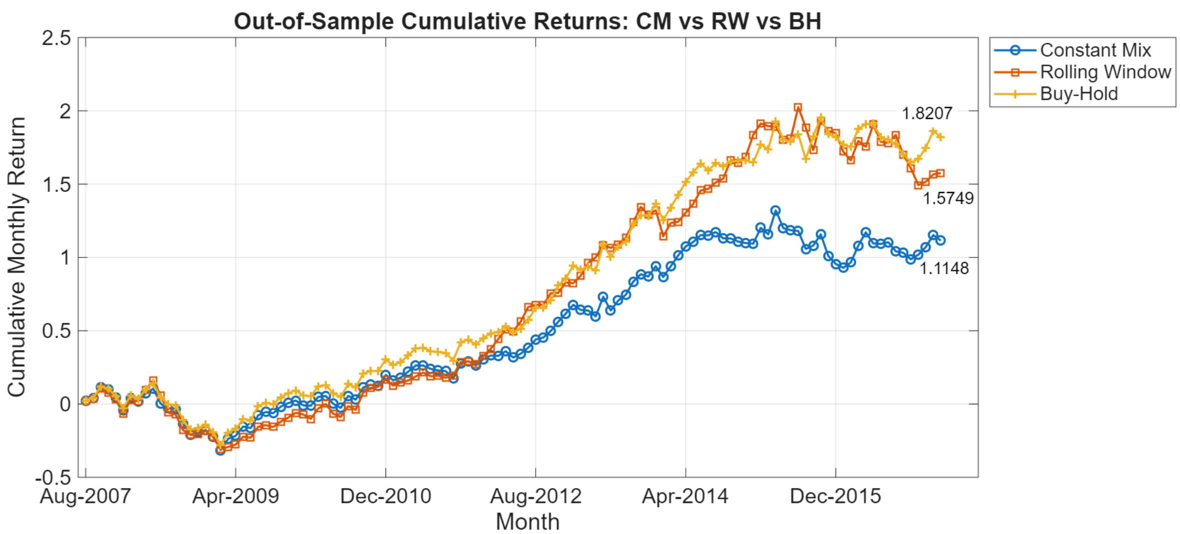

Figure 9: Cumulative monthly returns for Constant Mix, Rolling Window, Buy-hold Strategies

2008–2009 Downturn: All three strategies began with allocations tilted toward Basic Materials, Consumer Goods, Telecommunications, and Financials sectors that had thrived pre-crisis during the commodity boom, credit expansion, and rising consumer demand. However, when the 2008 global financial crisis hit, these same cyclical, growth-oriented sectors were among the hardest hit. The similarity in poor performance across CM, BH, and RW during this time highlights that no rebalancing or adaptive strategy could fully shield portfolios from widespread market shocks.

2010–2014 Recovery and Outperformance: From early 2010, the portfolios recovered in line with the global and South African rebound, as commodity prices stabilized, consumer demand returned, and the financial sector regained footing. Here, BH and RW outperformed CM because their allocations retained or adapted exposure to the same growth sectors in

Consumer Goods, Health Care, and Telecommunications which powered the recovery. RW's rolling updates allowed it to capture shifts in momentum, sometimes surpassing BH, while BH benefitted from simply holding onto sectors that bounced back strongly. By contrast, CM's rebalancing constrained its ability to fully exploit these surges, leading to lower cumulative growth compared to the adaptive and passive strategies.

Plateau Periods (2014–2015 onwards): By 2014, CM plateaued at a cumulative return of 1.11, its conservative rebalancing stabilizing performance but limiting upside as markets slowed. BH and RW plateaued slightly later, around early 2015, ending with 1.82 and 1.57, respectively, reflecting continued gains from high-return sectors before momentum faded. South Africa's sluggish growth, electricity supply issues, and lower global commodity demand contributed to the slower incremental returns during this period. RW's adaptability helped capture short-term trends, keeping it near peak performance at times, while BH benefitted from holding strong sectors consistently.

Overall, these outcomes highlight the strengths and weaknesses of each strategy: CM provided stability but capped returns, explaining its lower final cumulative return, BH leveraged sector rebounds effectively, leading to the highest cumulative gain and RW adapted to trends achieving solid performance but slightly below BH.

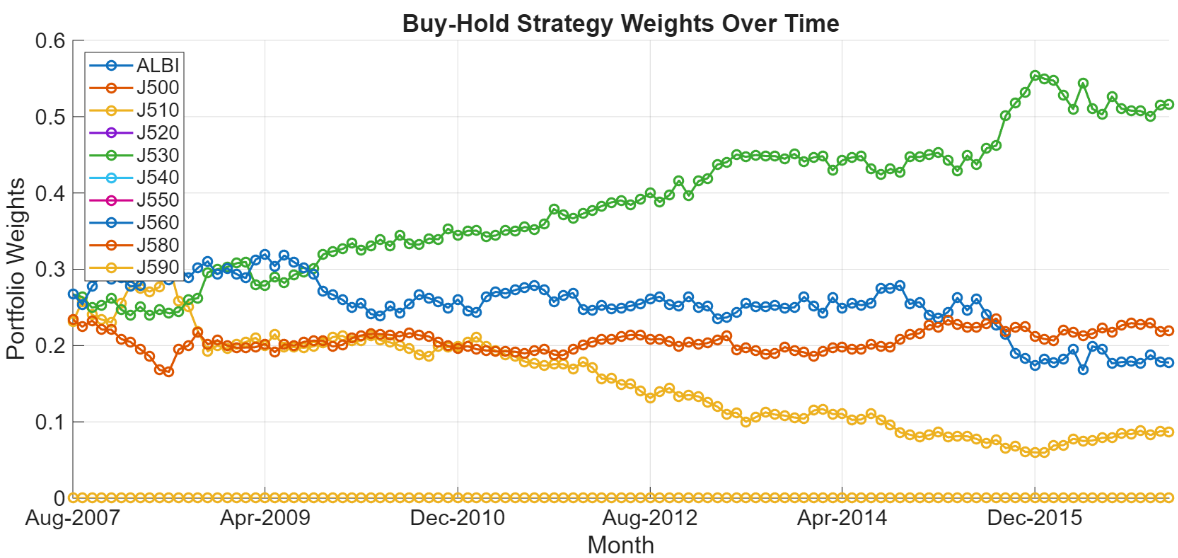

Figure 10: Weights for Buy Hold over the time series

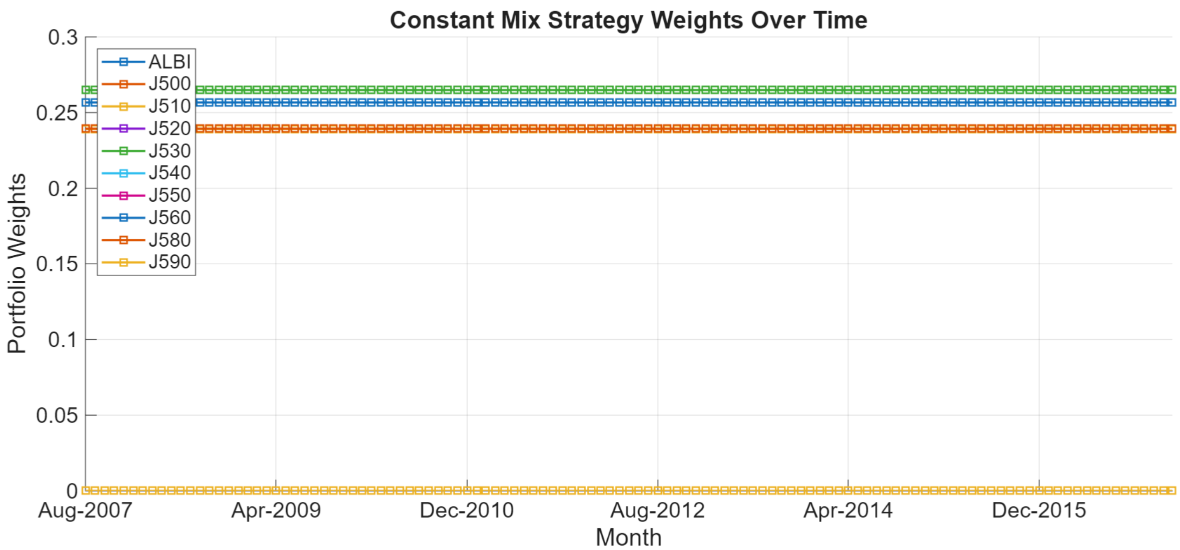

Figure 11: Weights for Constant Mix over the time series

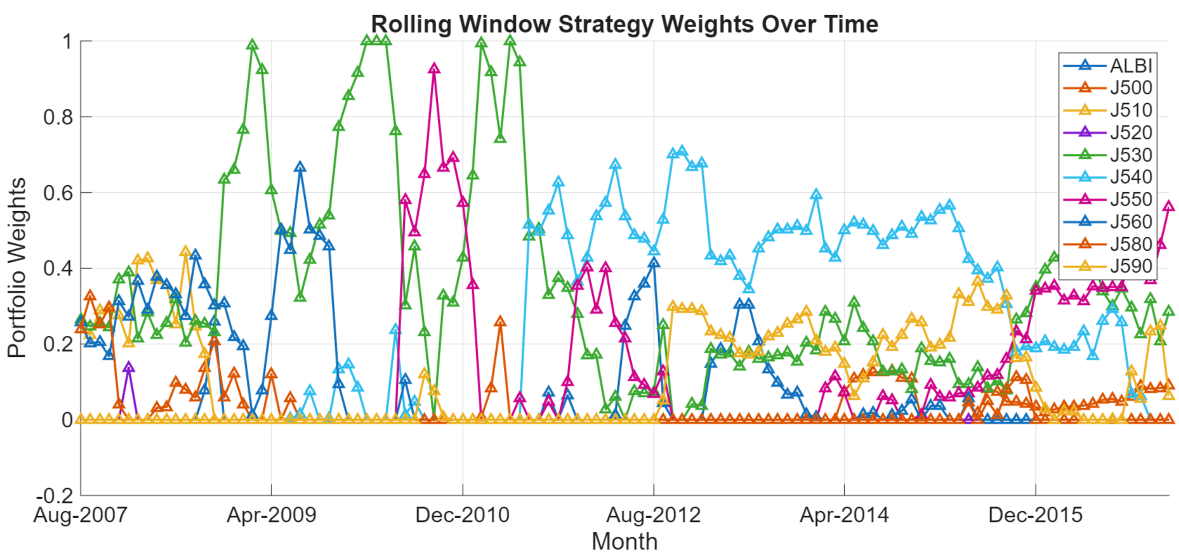

Table 9: Initial and Ending Portfolio Weights

Asset	ALBI	J500	J510	J520	J530	J540	J550	J560	J580	J590
Initial	0	0.0000	0.2312	0	0.2671	0	0.000	0.2676	0.2341	0.0000
End_CM	0	0.0000	0.2391	0	0.2648	0	0.000	0.2567	0.2394	0.0000
End_BH	0	0.0000	0.0867	0	0.5158	0	0.000	0.1777	0.2198	0.0000
End_RW	0	0.0901	0.0000	0	0.2852	0	0.562	0.0000	0.0000	0.0627

The weight patterns reveal key differences between strategies. CM remains close to initial weights, keeping diversification but limiting upside. BH shifts toward high-performing sectors like J530 while reducing exposure to other assets, particularly J510, whose weight drops by a third, allowing it to capture rebounds without rebalancing. RW changes dynamically: 2007–2008 saw low allocations across assets due to the financial crisis, from 2007–2008, allocations were low across assets due to the financial crisis; 2009–2011 focused on 1–3 assets, with J530 reaching 100% during the recovery; post-2011 J540 dominated, then dropped to zero after 2015 as market trends plateaued and structural headwinds reduced its appeal.

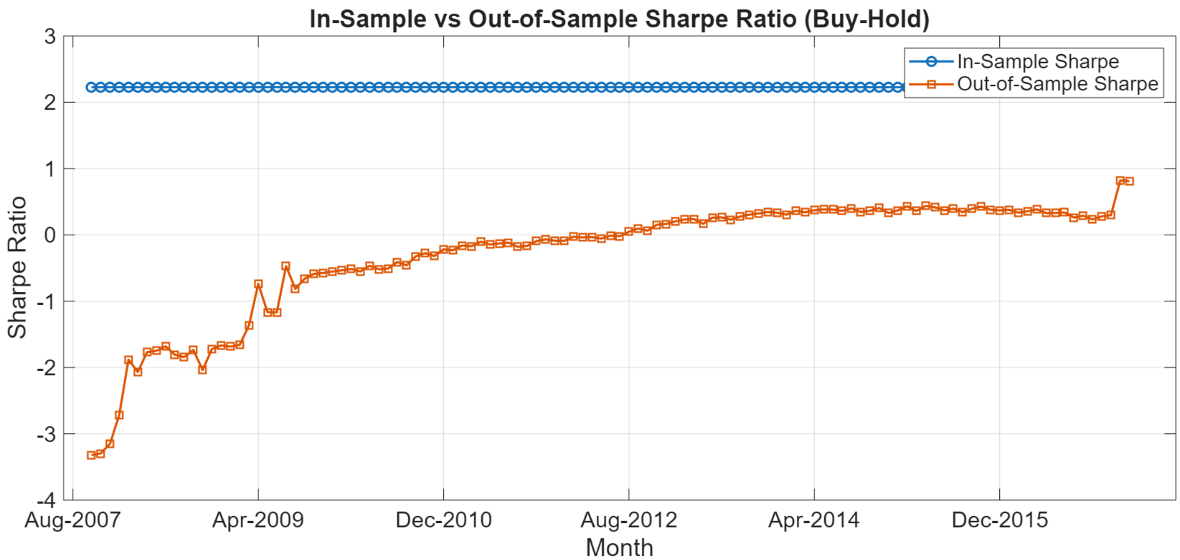

Figure 12: In-Sample Sharpe Ratio vs Out-Sample Sharpe Ratio

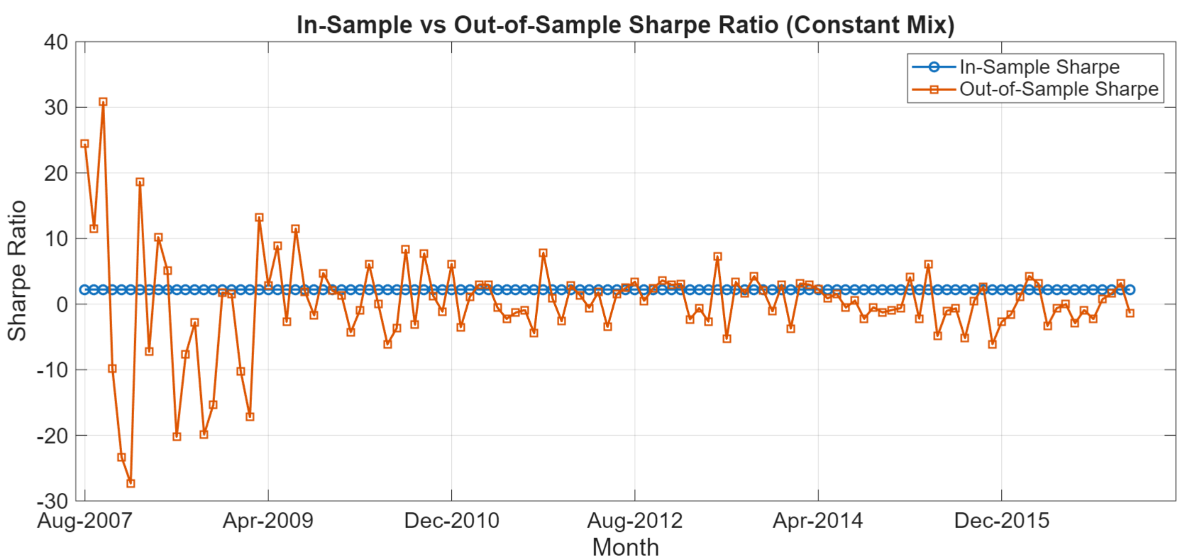

Figure 13: In-Sample Sharpe Ratio vs Out-Sample Sharpe Ratio

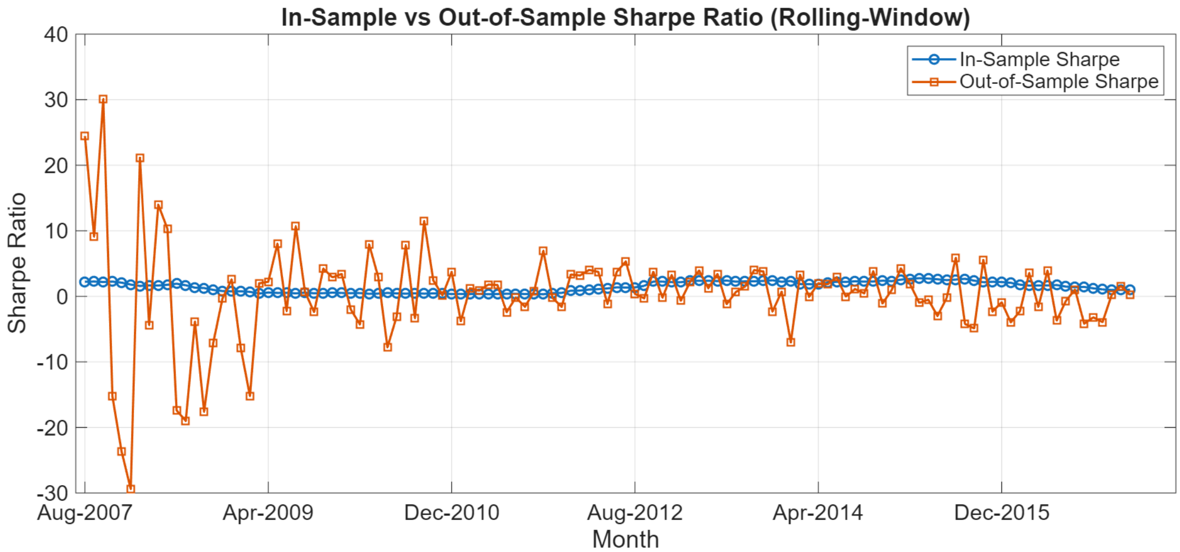

Figure 14: In-Sample Sharpe Ratio vs Out-Sample Sharpe Ratio

These patterns show strategy sensitivity to market conditions. Buy-and-hold's Sharpe stays negative until after 2012, reflecting poor risk-adjusted returns during the slow recovery. CM and RW oscillate sharply in 2007–2008 due to rebalancing and adaptive weights reacting to shocks, but post-2011 their Sharpe ratios narrow as markets stabilize, showing more consistent risk-adjusted performance once trends persist.

Table 10: Summary Statistics for the Different Portfolio Strategies

Metric	ConstantMix	RollingWindow	BuyHold
Mean Return	0.0074	0.0092	0.0100
Variance	0.0012	0.0013	0.0017
Sharpe	0.1724	0.3631	0.3452
Min Return	-0.1146	-0.1152	-0.1091
Max Return	0.1129	0.1225	0.1191
Min Sharpe	-27.3320	-29.3910	-9.9628
Max Sharpe	30.8090	30.1060	9.2620
Cumulative Return	1.1148	1.5749	1.8207

The table highlights the trade-offs between the three strategies. Buy-and-hold delivered the highest cumulative return (1.82) and mean return, reflecting the benefit of remaining invested in strong sectors like Consumer Goods, Health Care, and Telecommunications, though with higher variance (0.00169) exposing it to market swings. Rolling window achieved a slightly lower cumulative return (1.57) but the highest Sharpe ratio (0.36), as its adaptive weighting captured short-term trends while moderating exposure to weaker sectors. Constant mix had the lowest cumulative return (1.11) and Sharpe (0.17) but also the lowest variance (0.00119), demonstrating how fixed weights stabilize performance but constrain upside potential.

Additional Statistical Test

This test examines how different training window sizes affect out-of-sample performance for CM, RW, and BH strategies. For each window fraction, optimal weights are computed on the training data and applied to the next period to calculate returns, risk, and Sharpe ratios.

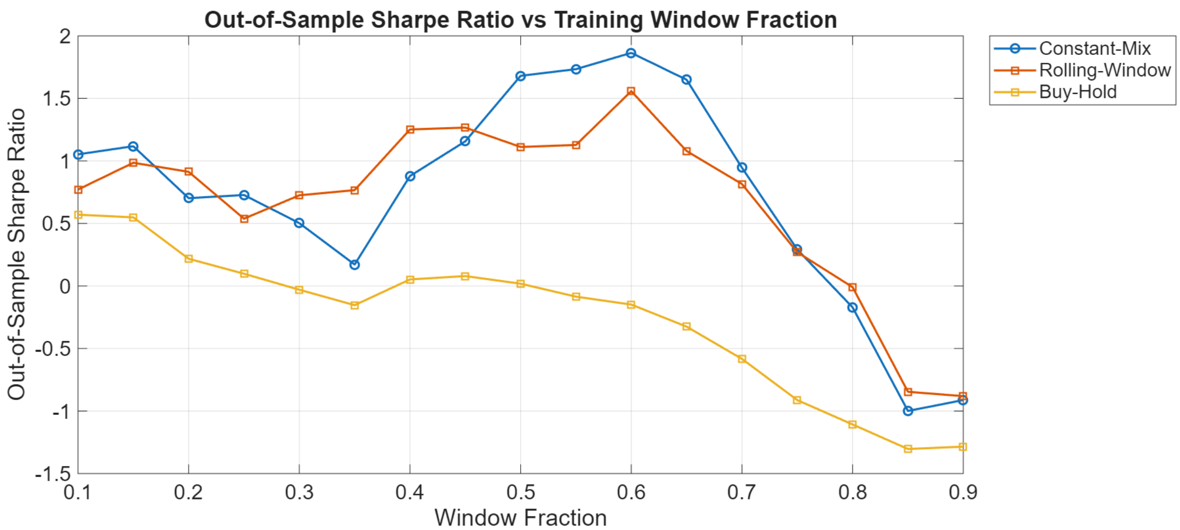

Figure 15: Out of sample sharpe ratios for each strategy over different window periods

These observations reinforce key insights about the strategies rather than revealing fundamentally new behavior. The fact that buy-and-hold consistently has the lowest out-of-sample Sharpe despite high cumulative returns highlights its vulnerability to risk: it captures the upside of strong sectors but offers no mitigation against volatility, confirming earlier conclusions about its high-return, high-risk profile. Meanwhile, the peak Sharpe for constant mix at a 0.6 window fraction emphasizes the benefit of fixed weights combined with a moderately sized training window, which balances responsiveness to historical trends with portfolio stability, underscoring CM's strength in delivering superior risk-adjusted performance under certain conditions.

Appendix

Data Pre-Processing

```matlab
% 0. Clear environment and remove all plots
clc % clears the command window
close all %removes all figures
clear % clears the workspace

% 1. Load the data
% 1.1 Loading the excel file into matlab
fileName = "C:\Users\User\OneDrive - University of Cape Town\Notes Honours 2025\Portfolio Theory\Portfolio_Theory_A1\Data\PT-DATA-ALBI-JIBAR-JSEIND-Daily-1994-2017.xlsx";
excelSheetNames = sheetnames(fileName);
% 1.2 Preallocate cell array before looping
data{numel(excelSheetNames)} = [];
% 1.3 Load the dataset by sheet
for sheet = 1:numel(excelSheetNames)
    data{sheet} = readtimetable(fileName, 'Sheet', excelSheetNames{sheet}, 'VariableNamingRule', 'preserve');
end

% 2. Filter and Clean Data: Keep Only Relevant Tickers and Columns
% 2.1 Define a list of all the assets we want to include in the universe
% - Includes RATESTEFI, ALBI, key J-bonds, variable J5x tickers, and major JSE indices
variableTickers = string("J5" + (10 : 10 : 90));
entities = {'RATESTEFI', 'RATEJ2Y4', 'ALBI', 'J203', 'J500', 'J330', 'J331', variableTickers{:}};
% 2.2 Loop over each sheet in the dataset
for i = 1:numel(data)
    % 2.2.1 Keep only TRI (Total Return Index) columns for sheets 3 and 4
    if i == 3
        allVarTable = data{i};
        TRITable = allVarTable(:, (3:3:27)); % select every 3rd column (TRI)
        TRITable = TRITable(4:end,:); % remove the first 3 rows (NaNs)
        % Convert all columns to numeric if they are cells/strings
        % (i.e. J500 needs converting)
        for col = 1:width(TRITable)
            colName = TRITable.Properties.VariableNames{col};
            % Check if the column is cell or string
            if iscell(TRITable.(colName)) || isstring(TRITable.(colName))
                TRITable.(colName) = str2double(TRITable.(colName)); % convert to numeric
            end
        end
        data{i} = TRITable;
    elseif i == 4
```

```
        allVarTable = data{i};
        allVarTable = removevars(allVarTable, ["SOURCE68779","Var2"]); % remove unwanted columns
        TRITable = allVarTable(:, (4:4:19)); % select every 4th column (TRI)
        TRITable = TRITable(4:end,:); % remove the first 3 rows (NaNs)
        data{i} = TRITable;
    else
        % 2.2.2 For other sheets, just skip the first 3 rows
        data{i} = data{i}(4:end, :);
    end
    % 2.3 Match column names to entities (whitelist)
    opts = data{i}.Properties; % get table properties
    variableMatch = zeros(size(opts.VariableNames)); % preallocate array for matching
    % 2.3.1 Perform string comparison for column matching
    if i == 2
        % Sheet 2: compare first 8 characters to avoid unwanted matches
        for k = 1:numel(entities)
            variableMatch(strncmp(opts.VariableNames, entities{k}, 8)) = k;
        end
    else
        % Other sheets: compare first 4 characters
        for k = 1:numel(entities)
            variableMatch(strncmp(opts.VariableNames, entities{k}, 4)) = k;
        end
    end
    % 2.3.2 Identify and remove columns not in the whitelist
    idx = find(variableMatch == 0); % find unwanted tickers
    tickersToBeRemoved = opts.VariableNames(idx); % get their names
    data{i} = removevars(data{i}, tickersToBeRemoved); % remove them from the table
    % 2.4 Clean up remaining column names
    % - Remove any text after ':' to simplify variable names
    hasColon = contains(data{i}.Properties.VariableNames, ':');
    data{i}.Properties.VariableNames(hasColon) = extractBefore(data{i}.Properties.VariableNames(hasColon), ':');
end

% 3. Clean and convert into a single timetable
allDataTable = synchronize(data{1},data{2},data{3},data{4});
%Rename variables
allDataTable = renamevars(allDataTable,{'RATESTEFI','RATEJ2Y4','J203'},{'STEFI','JIBAR','ALSI'});

% 4. Down-sample (by decimation)
% 4.1 Decimate the daily data to monthly data
allDataTable = convert2monthly(allDataTable); % Home -> Add-ons -> Financial Toolbox

```

```matlab
% 5. Visualising and Handling Missing Data
% Helper function to get number of NaNs
countNaNs = @(T) sum(isnan(table2array(T)),'all');
% 5.1 Initial Missing Data Visualisation
figure;
subplot(1,6,1);
spy(isnan(table2array(allDataTable)));
xlabel(["Assets","NaNs: " + countNaNs(allDataTable)]);
ylabel("Months")
title({"1. Initial","Missing","Data"})
% 5.2 Fill Missing Data with Zero-Order Hold (Last Observation Carried Forward)
allDataTable = fillmissing(allDataTable,'previous');
subplot(1,6,2);
spy(isnan(table2array(allDataTable)));
xlabel(["Assets","NaNs: " + countNaNs(allDataTable)]);
ylabel("Months")
title({"2. Fill","Zero-Order","Hold"})
% 5.3 Remove Rows Using the Asset With The Least NaN's (best proxy asset)
[minNans,idx] = min(sum(isnan(allDataTable{:,:}),1));
rmmissingProxy = allDataTable.Properties.VariableNames{idx}; % Find the column with the FEWEST missing values
allDataTable = rmmissing(allDataTable,"DataVariables",rmmissingProxy); % Remove rows where the proxy has NaNs
subplot(1,6,3);
spy(isnan(table2array(allDataTable)));
xlabel(["Assets","NaNs: " + countNaNs(allDataTable)]);
ylabel("Months")
title({"3. Remove","Based on","Proxy"})
% 5.5 Remove Leading Rows with Missing Data
% Detect rows at the start of the dataset that contain NaNs and trim them
idx = isnan(allDataTable{:,:});
allDataTable = allDataTable(max(find(idx,max(max(cumsum(idx)))))+1:end,:);
subplot(1,6,4);
spy(isnan(table2array(allDataTable)));
xlabel(["Assets","NaNs: " + countNaNs(allDataTable)]);
ylabel("Months")
title("4. Final")
sgtitle('Handling Missing Values', 'FontSize', 14, 'FontWeight', 'bold');

% 6. Visualise the data and the returns on a single plot
% 6.1 Compute simple returns (Rt = (Pt - Pt-1)/Pt-1)
allDataReturnsTable = tick2ret(allDataTable,'Method','Simple');
% 6.2 Plot of the time series
figure;
plot(allDataTable.Time,allDataTable{:,:})
ylabel("Returns")
```

```
title("TRI for sectors")
legend(allDataTable.Properties.VariableNames,Location="westoutside")
% 6.3 Plot of returns
figure;
plot(allDataReturnsTable.Time,allDataReturnsTable{:,:})
ylabel("Returns")
title("Monthly Sampled Simple Returns")
legend(allDataTable.Properties.VariableNames,Location="westoutside")
% 6.4 Handle outlier (J330 and J331)
outlierIdx = allDataReturnsTable.("J330") < -0.5;
outlierRows = find(outlierIdx);
disp(table( ...
    outlierRows, ...
    allDataReturnsTable.Time(outlierRows), ...
    allDataReturnsTable.("J330")(outlierRows), ...
    allDataReturnsTable.("J331")(outlierRows), ...
    'VariableNames', {'RowIndex', 'Date', 'J330_Return', 'J331_Return' }));  % Display the outliers
allDataReturnsTable.("J330")(outlierIdx) = NaN; % set the outliers to NaN
allDataReturnsTable.("J331")(outlierIdx) = NaN; % set the outliers to NaN
% Check J500
outlierIdx2 = allDataReturnsTable.("J500") > 0.5;
outlierRows2 = find(outlierIdx2);
disp(table( ...
    outlierRows2, ...
    allDataReturnsTable.Time(outlierRows2), ...
    allDataReturnsTable.("J500")(outlierRows), ...
    'VariableNames', {'RowIndex', 'Date', 'J500_Return'}));
allDataReturnsTable.("J500")(outlierIdx2) = NaN;
% 6.5 Plot of returns without the outliers
figure;
plot(allDataReturnsTable.Time,allDataReturnsTable{:,:})
ylabel("Returns")
title("Monthly Sampled Simple Returns")
legend(allDataTable.Properties.VariableNames,Location="westoutside")

% 7. Manage Missing Data
% 7.1 Check if there is still missing data or cells of zero
any(isnan(allDataReturnsTable{:,:}))
nZeros = sum(allDataTable{:,:} == 0,'all');
disp(['Total zero values: ', num2str(nZeros)]);
% 7.2 Compute the Arithmetic means correcting for missing data (NAN)
portfolioMean = mean(allDataReturnsTable{:,:},'omitnan');
portfolioStdDev = std(allDataReturnsTable{:,:},1,'omitnan');
portfolioVariance = var(allDataReturnsTable{:,:},'omitnan');
% 7.3 Fill those missing values with the previous results
allDataTable{:,:}(allDataTable{:,:} == 0) = NaN; % Replace zeros with NaN
allDataTable = fillmissing(allDataTable,'previous'); % Replace NaNs with the previous observation
```

```

% 8. Discounting the 3 month jibar yield
threeMonthJibar = allDataTable{:, 'JIBAR'};
% Convert 3-month JIBAR to monthly equivalent
monthlyJibar = (1 + threeMonthJibar).^(1/3) - 1;
% Update the table with the monthly JIBAR values
allDataTable{:, 'JIBAR'} = monthlyJibar;
mean(monthlyJibar)
```

Experiment 1 (training)

```
% 1. Split into two sets of data
% 1.1 Number of observations
nObs = size(allDataTable,1);
% 1.2 Choose a split point (e.g. 70% training, 30% test)
splitPoint = floor(0.7 * nObs);
% 1.3 Also keep the dates if you need them
train = allDataTable(1:splitPoint,:);
test = allDataTable(splitPoint+1:end,:);
disp(['Training rows: ', num2str(size(train,1))]);
disp(['Test rows: ', num2str(size(test,1))]);
returnType = "continuous";
tickers = setdiff(train.Properties.VariableNames,{'STEFI','JIBAR','ALSI','J330','J331'}); % Tickers to include
riskFreeTicker = "STEFI";

% 2.1 Portfolio Set Up
q = Portfolio('AssetList',train.Properties.VariableNames);
% 2.2 Simple Arithmetic Returns
trainReturnsTable = tick2ret(train,'Method',returnType); % Change this if needed
q = estimateAssetMoments(q,trainReturnsTable{:,:});
% 2.3 Visualise the Risk-Return Relationship
clf;
portfolioexamples_plot('Historic Risk and Return', ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});

% 3. Fully invested portfolio with varying risk aversion
function [PortWts, ret, rsk] = efficientFrontier1(q, lambda)
    % 3.1 Fully Invested
    q.AEquality = ones(1,length(q.AssetMean));
    q.bEquality = 1;
    PortWts = NaN(length(lambda),length(q.AssetMean));

    % 3.2 Find the optimal portfolio weights
    options = optimoptions('quadprog','Display','off');
    for i = 1:length(lambda)
        f = - lambda(i) * q.AssetMean; % This moves the solution up along the efficient frontier
        H = q.AssetCovar; %the covariance matrix
```

```
        PortWts(i,:) = quadprog(H, f, [], [], q.AEquality, q.bEquality, [], [], [], options) ;
    end
    % 3.3 Compute Portfolio Risk and Return
    ret = estimatePortReturn(q, PortWts');
    rsk = estimatePortRisk(q,PortWts');
    %ret = PortWts * q.AssetMean';
    %rsk = sqrt(diag(PortWts * q.AssetCovar * PortWts'));
end

% 3.4 Efficient Frontier
% Create the range of risk aversion parameters
lambda = linspace(-0.25,0.25,45);
[Wts,ret1,rsk1] = efficientFrontier1(q,lambda);
% 3.5 Plot the Curve
clf;
portfolioexamples_plot('Historic Risk and Return', ...
    {'line', rsk1, ret1}, ...
    {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});

% 4. Exclude the Cash and Market Indices
function [PortWts, ret, rsk] = efficientFrontier2(trainReturnsTable, lambda, includingTickers)
    % 4.1 Statistics
    Returns = trainReturnsTable{:,includingTickers};
    mu = mean(Returns); % Use Mean With Excluded Indices
    Sigma = cov(Returns); % Use Covar With Excluded Indices

    % 4.2 Fully Invested
    Aeq = ones(1,length(mu));
    beq = 1;
    % initialise the weights
    PortWts = NaN(length(lambda),length(mu));

    % 4.3 Find the optimal portfolio weights
    optionsQP = optimset('quadprog')';
    optionsQP= optimset(optionsQP,'Display','off');
    for i = 1:length(lambda)
        f = - lambda(i) * mu; % This moves the solution up along the efficient frontier
        H = Sigma; %the covariance matrix
        [PortWts(i,:),fVal,exitFlag(i)] = quadprog(H,f,[],[],Aeq,beq,[],[],[],optionsQP);
    end

    % 4.4 compute risk and return
    ret = PortWts * mu';
    rsk = sqrt(diag(PortWts * Sigma * PortWts'));
end

% 4.5 Efficient frontier
```

```
% Tickers we are excluding
lambda = linspace(-0.25,0.25,45);
[Wts, ret2,rsk2] = efficientFrontier2(trainReturnsTable,lambda,tickers
    );

% 4.6 Plot
plotIdx = ismember(q.AssetList, tickers);
clf;
portfolioexamples_plot('Historic Risk and Return', ...
    {'line', rsk2, ret2}, ...
    {'scatter', sqrt(diag(q.AssetCovar(plotIdx,plotIdx))), q.AssetMean(
        plotIdx), q.AssetList(plotIdx), '.r'});
% {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'
    });

% 5. Exclude the Cash and Market Indices
function [PortWts,ret, rsk] = efficientFrontier4(trainReturnsTable,
    exceedingValue, includingTickers)
    % 5.1 Statistics
    Returns = trainReturnsTable{:,includingTickers};
    mu = mean(Returns); % Use Mean With Excluded Indices
    Sigma = cov(Returns); % Use Covar With Excluded Indices

    % 5.2 Create the range of risk aversion parameters
    retTargetAll = min(mu):((max(mu)-min(mu))/90):max(mu);
    retTarget = retTargetAll(retTargetAll > exceedingValue);

    % 5.3 Equality constraint (fully invested)
    Aeq = ones(1,length(mu));
    beq = 1;
    % initialise the weights
    PortWts = NaN(length(retTarget),length(mu));

    % 5.4 No Short-selling (upper and lower bounds)
    ub = ones(length(mu),1);
    lb = zeros(length(mu),1);

    % 5.5 Equality constraint (return target)
    Aeq = [Aeq; mu];
    beq = [beq; 0];

    % 5.6 Find the optimal portfolio weights
    optionsQP = optimset('quadprog')';
    optionsQP= optimset(optionsQP,'Display','off');
    for i = 1:length(retTarget)
        beq(2) = retTarget(i); % This moves the solution up along the
            efficient frontier
        H = Sigma; %the covariance matrix
        [PortWts(i,:),fVal,exitFlag(i)] = quadprog(H,zeros(size(mu))'
            ,[],[],Aeq,beq,lb,ub,[],optionsQP);
    end

```

```
    % 5.7 compute risk and return
    ret = PortWts * mu';
    rsk = sqrt(diag(PortWts * Sigma * PortWts'));
end

% 6. Exclude the Cash and Market Indices
function [PortWts,ret, rsk, ERFR] = maxSharpeRatio(trainReturnsTable, includingTickers, riskFreeTicker)
    % 6.1 Statistics
    Returns = trainReturnsTable{:,includingTickers};
    mu = mean(Returns); % Use Mean With Excluded Indices
    Sigma = cov(Returns); % Use Covar With Excluded Indices

    % 6.2 Risk free rate
    RFR = trainReturnsTable{:,riskFreeTicker};
    % average risk free rate (when to use geometric average)
    ERFR = mean(RFR);

    % 6.3 Equality constraint (fully invested)
    Aeq = ones(1,length(mu));
    beq = 1;
    % initialise the weights for Equally weighted portfolio
    Wts0 = ones(size(mu))/length(mu);

    % 6.4 No Short-selling (upper and lower bounds)
    ub = ones(length(mu),1);
    lb = zeros(length(mu),1);

    % 6.5 objective function to maximise the SR
    fn0 = @(x) (-(x*mu' - ERFR)/sqrt(x*Sigma*x'));

    % 6.6 Use SQP to solve for the tangency portfolio
    options = optimoptions(@fmincon,'Algorithm','sqp','OptimalityTolerance',1e-8,'Display','off');
    PortWts = fmincon(fn0,Wts0,[],[],Aeq,beq,lb,ub,[],options); % Maximum Sharpe Ratio Portfolio Weights

    % 6.7 compute risk and return
    ret = PortWts*mu';
    rsk = sqrt(PortWts*Sigma*PortWts');
end

% 6.8 Tickers we are excluding
[Wts_train,retSR, rskSR,ERFR_Train] = maxSharpeRatio(trainReturnsTable, tickers, riskFreeTicker);

% 6.9.1 Plot Combined
plotIdx = ismember(q.AssetList, tickers);
excludedIdx = ~plotIdx;
clf;
portfolioexamples_plot('Efficient Frontier (QP)', ...
    {'line', rsk1(12:end-12), ret1(12:end-12),'','b'}, ...
```

```
    {'line', rsk2(10:end-10), ret2(10:end-10),'','y'}, ...
    {'line', rsk4, ret4,'','m'}, ...
    {'scatter', rskSR,retSR,{'Maximum Sharpe ratio'},'b'}, ...
    {'scatter', sqrt(diag(q.AssetCovar(plotIdx,plotIdx))), q.AssetMean(plotIdx), q.AssetList(plotIdx), 'r'}, ... % Investable tickers
    {'scatter', sqrt(diag(q.AssetCovar(excludedIdx,excludedIdx))), q.AssetMean(excludedIdx), q.AssetList(excludedIdx), 'g'}); % Excluded indices in green
% {'scatter', sqrt(diag(q.AssetCovar)), q.AssetMean, q.AssetList, '.r'});
% Add manual legend outside the plot
hLegend = legend({'Step 1: Efficient Frontier (All Assets)','Step 2: Efficient Frontier (Excl Cash/Market)','Step3: Long-only Frontier',...
    'Maximum Sharpe Ratio Portfolio','Investable Assets','Excluded Indices'}, ...
    'Location','northeastoutside'); % Puts legend outside on the right
set(hLegend,'FontSize',15);
% 6.9.2 Isolated Plot
plotIdx = ismember(q.AssetList, tickers);
clf;
portfolioexamples_plot('Efficient Frontier (QP)', ...
    {'line', rsk4, ret4,'','m'}, ...
    {'scatter',rskSR,retSR,{'Maximum Sharpe ratio'},'b'}, ...
    {'scatter', sqrt(diag(q.AssetCovar(plotIdx,plotIdx))), q.AssetMean(plotIdx), q.AssetList(plotIdx), 'r'});

% 7. Market Security Line
hold on
% 7.1 SML through the tangency portfolio
SML = @(x) (12*ERFR_Train + sqrt(12)*(retSR-ERFR_Train)/rskSR*x);
x = linspace(0,0.20,20);
% 7.2 Plot market line
plot(x,SML(x),'r')
hold off

% 8.1 Plot the Sharpe Ratio against risk levels
yyaxis right
% 8.2 plot the Sharpe Ratio against risk level
plot(sqrt(12)*rsk4,sqrt(12)*(ret4-ERFR_Train)./rsk4,'g', 'DisplayName', 'Sharpe Ratio')
xline(rskSR*sqrt(12), '--g', 'LineWidth', 1.5, ...
    'Label', '', 'LabelVerticalAlignment','bottom', 'LabelOrientation', 'horizontal', 'DisplayName', '');
legend( {'Step3: Long-only Frontier','Maximum Sharpe Ratio Portfolio','Investable Assets','Market Line','Sharpe Ratios'}, ...
    'Location', 'northeastoutside');
set(legend,'FontSize',15);
ylabel('Sharpe Ratio ${ ( r_p -r_{f} ) / \sigma_p }$','interpreter','latex','Color','g')
hold off
```

Experiment 1 (test)

```
% 9. Constant Mix Strategy on Test Dataset
% 9.1 Compute Constant Mix Out of Sample Portfolio statistics
function [ret, rsk] = constantMix(testReturnsTable, Wts, excludingTickers)
    testReturnsTable = testReturnsTable{:,excludingTickers};
    if size(testReturnsTable,1) == 1
        % If one row, 'mean' calculates the mean for the row,
        % therefore skip it
        ret = testReturnsTable * Wts';
    else
        % If Multiple rows, 'mean' calculates the mean for the column
        % therefore use it
        ret = mean(testReturnsTable) * Wts';
    end
    rsk = sqrt(Wts*cov(testReturnsTable)*Wts');
end
function [Sharpe] = sharpeRatio(ret, rsk, testReturnsTable, riskFreeTicker)
    ERFR = mean(testReturnsTable{:,riskFreeTicker});
    Sharpe = sqrt(12)*(ret-ERFR)./rsk;
end

% 9.2 Extract test returns (excluding cash and market indices)
testReturnsTable = tick2ret(test,'Method',returnType); % Change this if needed
[cmMean, cmSd] = constantMix(testReturnsTable,Wts_train, tickers);
[cmSharpe] = sharpeRatio(cmMean, cmSd,testReturnsTable, riskFreeTicker);

% 9.3 Plot
yyaxis left
hold on
scatter(sqrt(12) * cmSd, 12 * cmMean, 30, 'c', 'filled','DisplayName', 'Constant Mix Portfolio');
text(sqrt(12) * cmSd+0.003, 12 * cmMean, 'CM Out-of-Sample', ...
     'FontSize', 9, 'Color', 'c', 'FontWeight', 'bold');
hold off

% 10.1 Buy hold strategy
function [ret, rsk] = buyHold(testReturnsTable, Wts, includingTickers)
    % Exclude unwanted tickers
    R = testReturnsTable{:, includingTickers};
    [nPeriods, nAssets] = size(R);
    % Start with a notional portfolio value of 1 and allocate it based on initial weights
    assetValues = Wts;
    portValues = zeros(nPeriods + 1, 1);
    portValues(1) = sum(assetValues); % Initial portfolio value is the sum of initial asset values
    for t = 1:nPeriods
        % Grow each asset by its return for the current period
```

```
        assetValues = assetValues .* (1+R(t, :)); % use exp(R(t,:)) if continuous returns
        % The portfolio value is the sum of the new asset values
        portValues(t+1) = sum(assetValues);
        portRetSeries(t) = (portValues(t+1) - portValues(t)) / portValues(t);
    end
    % Calculate the mean and standard deviation of returns
    ret = mean(portRetSeries);
    rsk = std(portRetSeries);
end
function [Sharpe] = sharpeRatioM2Y(ret, rsk, testReturnsTable, riskFreeTicker)
    % Takes in Monthly ret and rsk and outputs an annualised Year sharpe ratio
    ERFR = mean(testReturnsTable{:,riskFreeTicker});
    Sharpe = sqrt(12)*(ret-ERFR)./rsk;
end

% 10.2 Extract test returns (excluding cash and market indices)
testReturnsTable = tick2ret(test,'Method',returnType); % Change this if needed
[bhMean, bhSd] = buyHold(testReturnsTable,Wts_train, tickers);
[bhSharpe] = sharpeRatioM2Y(bhMean, bhSd,testReturnsTable, riskFreeTicker);

% 10.3 Plot
yyaxis left
hold on
scatter(sqrt(12) * bhSd, bhMean*12, 30, 'b', 'filled','DisplayName', 'Buy-Hold Portfolio');
text(sqrt(12) * bhSd+0.003, bhMean*12-0.01, 'BH Out-of-Sample', ...
     'FontSize', 9, 'Color', 'b', 'FontWeight', 'bold');
hold off

% 11.1 In-sample statistics
inMean = retSR;
inVar = rskSR^2;
inSharpe = sharpeRatio(retSR, rskSR, trainReturnsTable, riskFreeTicker);

% 11.2 Comparison table
ResultsComparison = table( ...
    [inMean; cmMean; bhMean], ...
    [inVar; cmSd^2; bhSd^2], ...
    [inSharpe; cmSharpe; bhSharpe], ...
    'VariableNames', {'Mean','Variance','SharpeRatio'}, ...
    'RowNames', {'In-Sample SR','Out-of-Sample CM','Out-of-Sample BH'});
disp(ResultsComparison);

% 11.3 Portfolio weights table
```

```
tickersWithDesc = {...
    'ALBI-All-Bond','J500-Oil-Gas','J510-Basic-Materials','J520-Industrials','J530-Consumer-Goods','J540-Health-Care', ...
    'J550-Consumer-Services','J560-Telecommunication','J580-Financials','J590-Technology'};

% 11.3.1 Display table
WeightTable = array2table(round(Wts_train,4), 'VariableNames', tickersWithDesc);
disp(WeightTable);

% 11.3.2 Bar plot of portfolio weights
figure;
bar(Wts_train, 'FaceColor',[0.2 0.6 0.8]); % blueish bars
xticks(1:length(tickersWithDesc));
xticklabels(tickersWithDesc);
xtickangle(45); % tilt labels for readability
ylabel('Portfolio Weights');
title('Tangency Portfolio Weights by Instrument Type');
% Add values on top of each bar
xPos = 1:length(Wts_train);
yPos = Wts_train;
labels = string(round(Wts_train,4)); % round to 4 decimals (adjust if needed)
text(xPos, yPos, labels, 'HorizontalAlignment','center', ...
     'VerticalAlignment','bottom', 'FontSize',10);
grid on;
```

Experiment 1 (Additional Statistical Sophistication)

```
% 1. Parameters
splits = 0.1:0.05:0.9; % training fraction
nObs = size(allDataTable,1);

% 2.1 Preallocate results
results = table('Size',[length(splits),10], ...
                'VariableTypes',{'double','double','double','double','double','double','double','double','double','double'}, ...
                'VariableNames',{'TrainFraction','InReturn','InVar','InSharpe','OutCMReturn','OutCMVar','OutCMSharpe','OutBHReturn','OutBHVar','OutBHSharpe'});
% 2.2 Preallocate weight table
tickersWithDesc = {...
    'TrainFraction','ALBI-All-Bond','J500-Oil-Gas','J510-Basic-Materials','J520-Industrials','J530-Consumer-Goods','J540-Health-Care', ...
    'J550-Consumer-Services','J560-Telecommunication','J580-Financials','J590-Technology'};
weightsTable = array2table(NaN(length(splits), length(tickers)+1), ...
                           'VariableNames', tickersWithDesc);

```

```
% 3. Loop over training/test splits
for i = 1:length(splits)
    trainFrac = splits(i);
    splitPoint = floor(trainFrac * nObs);

    % 3.1 Split data
    train = allDataTable(1:splitPoint,:);
    test = allDataTable(splitPoint+1:end,:);

    % 3.2 Compute returns
    trainReturnsTable = tick2ret(train,'Method',returnType);
    testReturnsTable = tick2ret(test,'Method',returnType);

    % 3.3 Max Sharpe Tangency Portfolio on training set
    [Wts_train,retSR,rskSR,ERFR_Train] = maxSharpeRatio(
       trainReturnsTable, tickers, riskFreeTicker);

    % 3.4 Store training weights
    weightsTable{i,:} = [trainFrac, Wts_train];

    % 3.5 Compute in-sample Sharpe, return, variance
    inRet = retSR;
    inVar = rskSR^2;
    inSharpe = sharpeRatio(retSR, rskSR, trainReturnsTable,
       riskFreeTicker);

    % 3.6 Implement CM on test set
    [cmMean, cmSd] = constantMix(testReturnsTable,Wts_train,tickers);
    outCMRet = cmMean;
    outCMVar = cmSd^2;
    outCMSharpe = sharpeRatio(cmMean, cmSd, testReturnsTable,
       riskFreeTicker);

    % 3.6 Implement CM on test set
    [bhMean, bhSd] = buyHold(testReturnsTable,Wts_train,tickers);
    outBHRet = bhMean;
    outBHVar = bhSd^2;
    outBHSharpe = sharpeRatio(bhMean, bhSd, testReturnsTable,
       riskFreeTicker);

    % 3.7 Store results
    results.TrainFraction(i) = trainFrac;
    results.InReturn(i) = inRet;
    results.InVar(i) = inVar;
    results.InSharpe(i) = inSharpe;
    results.OutCMReturn(i) = outCMRet;
    results.OutCMVar(i) = outCMVar;
    results.OutCMSharpe(i) = outCMSharpe;
    results.OutBHReturn(i) = outBHRet;
    results.OutBHVar(i) = outBHVar;
    results.OutBHSharpe(i) = outBHSharpe;
end
```

```
% 4. Display results table
disp(results);

% 5. Display weights table
disp(weightsTable);

% 6. Plot Sharpe Ratio
figure;
plot(results.TrainFraction, results.InSharpe, '-o', 'LineWidth',2, '
    DisplayName','In-Sample');
hold on;
plot(results.TrainFraction, results.OutCMSharpe, '-s', 'LineWidth',2, '
    DisplayName','Out-of-Sample');
plot(results.TrainFraction, results.OutBHSharpe, '-s', 'LineWidth',2, '
    DisplayName','Out-of-Sample');
% Find indices of maximum Sharpe ratios
[~, idxCM] = max(results.OutCMSharpe);
[~, idxBH] = max(results.OutBHSharpe);
% Add vertical lines at maximum Sharpe ratio points
xline(results.TrainFraction(idxCM), '--g', 'LineWidth',1.5, '
    DisplayName','Max CM Sharpe');
xline(results.TrainFraction(idxBH), '--r', 'LineWidth',1.5, '
    DisplayName','Max BH Sharpe');
xline(0.7, '--m', 'LineWidth',1.5, 'DisplayName','Original Split');
xlabel('Training Fraction');
ylabel('Sharpe Ratio');
title('In-Sample vs Out-of-Sample Sharpe Ratio');
legend({'In-Sample','Out-Sample-CM','Out-Sample-BH','Max CM Sharpe','
    Max BH Sharpe','Original Split'},'Location','northeastoutside');
grid on;
hold off;
% Select the three training splits
selectedSplits = weightsTable(ismember(weightsTable.TrainFraction,
    [0.7, 0.55, 0.6]), :);
disp(selectedSplits)
% Extract weights as a matrix (rows = splits, columns = assets)
weightsMatrix = selectedSplits{:, 2:end}; % remove TrainFraction column
% Assets and number of splits
assets = selectedSplits.Properties.VariableNames(2:end);
numAssets = length(assets);
numSplits = height(selectedSplits);
% Create grouped bar chart
figure;
b = bar(weightsMatrix', 'grouped'); % transpose so assets on x-axis
% Customize colors (optional)
colors = [0.2 0.6 0.8; 0.8 0.4 0.2; 0.4 0.8 0.2]; % one color per split
for i = 1:numSplits
    b(i).FaceColor = colors(i,:);
end
% Set x-axis labels
xticks(1:numAssets);
```

```
xticklabels(assets);
xtickangle(45);
ylabel('Portfolio Weights');
xlabel('Assets');
title('Portfolio Weights for Selected Training Splits');
legend(strcat('Split ', string(selectedSplits.TrainFraction)), '
    Location', 'northeastoutside');
hold on;
% Add values on top of each bar
for i = 1:numSplits
    x = b(i).XEndPoints; % x positions of bars
    y = b(i).YEndPoints; % heights of bars
    labels = string(round(weightsMatrix(i,:),4));
    text(x, y, labels, 'HorizontalAlignment','center', '
        VerticalAlignment','bottom', 'FontSize',10);
end
hold off;

% Show sharpe ratios of maximum
% Define target training fractions
targetSplits = [0.7, 0.55, 0.6];
% Find the rows corresponding to those splits
rows = ismember(results.TrainFraction, targetSplits);
% Create table
SharpeTable = table(...
    results.TrainFraction(rows), ...
    results.InSharpe(rows), ...
    results.OutCMSharpe(rows), ...
    results.OutBHSharpe(rows), ...
    'VariableNames', {'TrainFraction', 'InSampleSharpe', 'OutCMSharpe',
         'OutBHSharpe'});
% Display the table
disp(SharpeTable);

% 7. Plot Mean Return vs Training Fraction
figure;
plot(results.TrainFraction, results.InReturn, '-o', 'LineWidth',2, '
    DisplayName','In-Sample');
hold on;
plot(results.TrainFraction, results.OutCMReturn, '-s', 'LineWidth',2, '
    DisplayName','Out-of-Sample');
plot(results.TrainFraction, results.OutBHReturn, '-s', 'LineWidth',2, '
    DisplayName','Out-of-Sample');
xlabel('Training Fraction');
ylabel('Mean Return');
title('In-Sample vs Out-of-Sample Mean Return');
legend('Location','best');
grid on;
hold off;

% 8. Plot Standard Deviation vs Training Fraction
figure;
```

```
plot(results.TrainFraction, sqrt(results.InVar), '-o', 'LineWidth',2, 'DisplayName','In-Sample');
hold on;
plot(results.TrainFraction, sqrt(results.OutCMVar), '-s', 'LineWidth',2, 'DisplayName','Out-of-Sample');
plot(results.TrainFraction, sqrt(results.OutBHVar), '-s', 'LineWidth',2, 'DisplayName','Out-of-Sample');
xlabel('Training Fraction');
ylabel('Standard Deviation (Risk)');
title('In-Sample vs Out-of-Sample Risk');
legend('Location','best');
grid on;
hold off;
```

Experiment 2

```
tickers = setdiff(allDataTable.Properties.VariableNames, ...
    {'STEFI','JIBAR','ALSI','J330','J331'}); % Excluded Tickers
returnType = 'continuous';
riskFreeRate = 'STEFI';

% 1. Parameters
windowFrac = 0.3; % training fraction of total data
nObs = size(allDataTable,1); % total months
windowSize = floor(windowFrac * nObs); % training window size in months

% 2. Preallocate result time series
nSteps = nObs - windowSize;
CMresults = table('Size',[nSteps,7], ...
    'VariableTypes',{'double','double','double','double','double','double','double'}, ...
    'VariableNames',{'CM_InReturn','CM_InVar','CM_InSharpe','CM_OutReturn','CM_OutVar','CM_OutSharpe','CM_SharpeDiff'});
RWresults = table('Size',[nSteps,7], ...
    'VariableTypes',{'double','double','double','double','double','double','double'}, ...
    'VariableNames',{'RW_InReturn','RW_InVar','RW_InSharpe','RW_OutReturn','RW_OutVar','RW_OutSharpe','RW_SharpeDiff'});
BHresults = table('Size',[nSteps,7], ...
    'VariableTypes',{'double','double','double','double','double','double','double'}, ...
    'VariableNames',{'BH_InReturn','BH_InVar','BH_InSharpe','BH_OutReturn','BH_OutVar','BH_OutSharpe','BH_SharpeDiff'});

% Preallocate tables for weights and test month
CM_WeightsTable = table('Size',[nSteps, length(tickers)+1], ...
                        'VariableTypes',[ "datetime", repmat("double",1, length(tickers)) ], ...
                        'VariableNames', [{'TestMonth'}, tickers]);
RW_WeightsTable = table('Size',[nSteps, length(tickers)+1], ...
                        'VariableTypes',[ "datetime", repmat("double",1, length(tickers)) ], ...
```

```
                         'VariableNames', [{'TestMonth'}, tickers]);
BH_WeightsTable = table('Size',[nSteps, length(tickers)+1], ...
                         'VariableTypes',[ "datetime", repmat("double",1,
                             length(tickers)) ], ...
                         'VariableNames', [{'TestMonth'}, tickers]);


% 3. CM weights (fixed from first window)
train_CM = allDataTable(1:windowSize,:);
trainReturns_CMBH = tick2ret(train_CM,'Method',returnType);
[Wts_CMBH,retSR_CMBH,rskSR_CMBH,~] = maxSharpeRatio(trainReturns_CMBH,
    tickers, riskFreeRate);

% 4. Store in-sample Sharpe for CM
inSharpeCMBH = sharpeRatio(retSR_CMBH, rskSR_CMBH, trainReturns_CMBH,
    riskFreeRate);

% Buy hold set up
assetValues = Wts_CMBH;
portValues = zeros(nSteps + 1, 1);
portValues(1) = sum(assetValues);
portRetSeries = zeros(nSteps, 1);

% Preallocate vectors to store one-month test returns for each strategy
CM_portRetSeries = zeros(nSteps,1); % Constant Mix
RW_portRetSeries = zeros(nSteps,1); % Rolling Window

% 5. Loop for both RW and CM time-series
for t = 1:nSteps
    % 5.1 Rolling Window Training
    train_RW = allDataTable(t:(t+windowSize-1),:);
    trainReturns_RW = tick2ret(train_RW,'Method',returnType);

    [Wts_RW,retSR_RW,rskSR_RW,~] = maxSharpeRatio(trainReturns_RW,
        tickers, riskFreeRate);
    inSharpeRW = sharpeRatio(retSR_RW, rskSR_RW, trainReturns_RW,
        riskFreeRate);

    % 5.2 Test month (same for CM and RW)
    testRow = [train_RW(end,:); allDataTable(t+windowSize,:)]; % one
        month forward
    testReturns = tick2ret(testRow,'Method',returnType);

    % 5.3 CM fixed weights (balancing)
    [cmMean, ~] = constantMix(testReturns, Wts_CMBH, tickers);
    CM_portRetSeries(t) = cmMean;
    cmSd = std(CM_portRetSeries);
    CMresults.CM_InReturn(t) = retSR_CMBH;
    CMresults.CM_InVar(t) = rskSR_CMBH^2;
    CMresults.CM_InSharpe(t) = inSharpeCMBH;
    CMresults.CM_OutReturn(t) = cmMean;
    CMresults.CM_OutVar(t) = cmSd^2;
```

```
    CMresults.CM_OutSharpe(t) = sharpeRatio(cmMean, cmSd, testReturns, riskFreeTicker);
    CMresults.CM_SharpeDiff(t) =abs(sharpeRatio(cmMean, cmSd, testReturns, riskFreeTicker) -inSharpeCMBH);
    CM_WeightsTable.TestMonth(t) = allDataTable.Time(t+windowSize);
    CM_WeightsTable{t, 2:end} = Wts_CMBH;

    % BH fixed weights (No rebalancing)
    % Use exp(returns) if continuous
    assetValues = assetValues .* exp(testReturns{:, tickers});
    portValues(t+1) = sum(assetValues);
    portRetSeries(t) = (portValues(t+1) - portValues(t)) / portValues(t);
    BHresults.BH_InReturn(t) = retSR_CMBH;
    BHresults.BH_InVar(t) = rskSR_CMBH^2;
    BHresults.BH_InSharpe(t) = inSharpeCMBH;
    BHresults.BH_OutReturn(t) = mean(portRetSeries);
    BHresults.BH_OutVar(t) = std(portRetSeries)^2;
    BHresults.BH_OutSharpe(t) = sharpeRatioM2Y(mean(portRetSeries), std(portRetSeries), testReturns, riskFreeTicker);
    BHresults.BH_SharpeDiff(t) =abs(sharpeRatioM2Y(mean(portRetSeries), std(portRetSeries), testReturns, riskFreeTicker) -inSharpeCMBH);
    BH_WeightsTable.TestMonth(t) = allDataTable.Time(t+windowSize);
    BH_WeightsTable{t, 2:end} = assetValues./sum(assetValues);

    % 5.4 Rolling Window updated weights
    [rwMean, ~] = constantMix(testReturns, Wts_RW, tickers);
    RW_portRetSeries(t) = rwMean;
    rwSd = std(RW_portRetSeries);
    RWresults.RW_InReturn(t) = retSR_RW;
    RWresults.RW_InVar(t) = rskSR_RW^2;
    RWresults.RW_InSharpe(t) = inSharpeRW;
    RWresults.RW_OutReturn(t) = rwMean;
    RWresults.RW_OutVar(t) = rwSd^2;
    RWresults.RW_OutSharpe(t) = sharpeRatio(rwMean, rwSd, testReturns, riskFreeTicker);
    RWresults.RW_SharpeDiff(t) = abs(sharpeRatio(rwMean, rwSd, testReturns, riskFreeTicker) -inSharpeRW);
    RW_WeightsTable.TestMonth(t) = allDataTable.Time(t+windowSize);
    RW_WeightsTable{t, 2:end} = Wts_RW;
end

% Initial Weights for CM and BH
tickersWithDesc = {...
    'ALBI-All-Bond','J500-Oil-Gas','J510-Basic-Materials','J520-Industrials','J530-Consumer-Goods','J540-Health-Care', ...
    'J550-Consumer-Services','J560-Telecommunication','J580-Financials','J590-Technology'};
figure;
bar(Wts_CMBH, 'FaceColor',[0.2 0.6 0.8]); % blueish bars
xticks(1:length(tickersWithDesc));
```

```
xticklabels(tickersWithDesc);
xtickangle(45); % tilt labels for readability
ylabel('Portfolio Weights');
title('Starting Portfolio Weights For Each Strategy');
% Add values on top of each bar
xPos = 1:length(Wts_CMBH);
yPos = Wts_CMBH;
labels = string(round(Wts_CMBH,4)); % round to 4 decimals
text(xPos, yPos, labels, 'HorizontalAlignment','center', ...
     'VerticalAlignment','bottom', 'FontSize',10);
grid on;
disp(Wts_CMBH)


% 6. Time-Series of Strategy Performance (Out-of-Sample)
% 6.1 Compute cumulative returns
cumCM = cumprod(1 + CMresults.CM_OutReturn) - 1;
cumRW = cumprod(1 + RWresults.RW_OutReturn) - 1;
cumBH = cumprod(1 + portRetSeries) - 1;
% 6.2 Plot
% Assume your allDataTable has a datetime column called 'Date'
dates = allDataTable.Time(windowSize+1:end); % test period dates
nSteps = length(dates);
% Indices for tick labels (every 20th month)
tickStep = 20;
tickIdx = 1:tickStep:nSteps;
% Plot cumulative returns
% 6.2 Plot with final cumulative returns
figure;
hCM = plot(1:nSteps, cumCM, '-o', 'LineWidth',1.5);
hold on;
hRW = plot(1:nSteps, cumRW, '-s', 'LineWidth',1.5);
hBH = plot(1:nSteps, cumBH, '-+', 'LineWidth',1.5);
% Set x-axis labels as dates for every 20th month
xticks(tickIdx);
xticklabels(datestr(dates(tickIdx), 'mmm-yyyy')); % format as "Aug
    -2003"
xlabel('Month');
ylabel('Cumulative Monthly Return');
title('Out-of-Sample Cumulative Returns: CM vs RW vs BH');
% Annotate final cumulative return at end of each line
text(nSteps, cumCM(end), sprintf('%.4f', cumCM(end)), '
    VerticalAlignment','bottom','HorizontalAlignment','right', 'Color',
    'k');
text(nSteps, cumRW(end), sprintf('%.4f', cumRW(end)), '
    VerticalAlignment','bottom','HorizontalAlignment','right', 'Color',
    'k');
text(nSteps, cumBH(end), sprintf('%.4f', cumBH(end)), '
    VerticalAlignment','bottom','HorizontalAlignment','right', 'Color',
    'k');
legend('Constant Mix','Rolling Window', 'Buy-Hold','Location', '
    northeastoutside');
```

```matlab
grid on;
hold off;


% 7. Compare In-Sample Sharpe Ratios vs Out-of-Sample CM Sharpe
figure;
plot(1:nSteps, CMresults.CM_InSharpe, '-o','LineWidth',1.5);
hold on;
plot(1:nSteps, CMresults.CM_OutSharpe, '-s','LineWidth',1.5);
xticks(tickIdx);
xticklabels(datestr(dates(tickIdx), 'mmm-yyyy')); % format as "Aug
    -2003"
xlabel('Month');
ylabel('Sharpe Ratio');
title('In-Sample vs Out-of-Sample Sharpe Ratio (Constant Mix)');
legend('In-Sample Sharpe','Out-of-Sample Sharpe');
grid on;
hold off;

% 8. Compare In-Sample vs Out-of-Sample Sharpe for Rolling-Window
figure;
plot(1:nSteps, RWresults.RW_InSharpe, '-o', 'LineWidth', 1.5);
hold on;
plot(1:nSteps, RWresults.RW_OutSharpe, '-s', 'LineWidth', 1.5);
xticks(tickIdx);
xticklabels(datestr(dates(tickIdx), 'mmm-yyyy')); % format as "Aug
    -2003"
xlabel('Month');
ylabel('Sharpe Ratio');
title('In-Sample vs Out-of-Sample Sharpe Ratio (Rolling-Window)');
legend('In-Sample Sharpe', 'Out-of-Sample Sharpe');
grid on;
hold off;
% 8. Compare In-Sample vs Out-of-Sample Sharpe for Buy-hold
figure;
plot(3:nSteps, BHresults.BH_InSharpe(3:end), '-o', 'LineWidth', 1.5);
hold on;
plot(3:nSteps, BHresults.BH_OutSharpe(3:end), '-s', 'LineWidth', 1.5);
xticks(tickIdx);
xticklabels(datestr(dates(tickIdx), 'mmm-yyyy')); % format as "Aug
    -2003"
xlabel('Month');
ylabel('Sharpe Ratio');
title('In-Sample vs Out-of-Sample Sharpe Ratio (Buy-Hold)');
legend('In-Sample Sharpe', 'Out-of-Sample Sharpe');
grid on;
hold off;

% 9. Table of Portfolio Statistics (two columns: CM and RW)
stats = {'MeanReturn'; 'Variance'; 'Sharpe'; 'MinReturn'; 'MaxReturn';
    'MinSharpe'; 'MaxSharpe';'Cumulative Return'};
% 9.1 CM statistics vector
```

```
CMstats = [ ...
    mean(CMresults.CM_OutReturn); ...
    mean(CMresults.CM_OutVar); ...
    mean(CMresults.CM_OutSharpe); ...
    min(CMresults.CM_OutReturn); ...
    max(CMresults.CM_OutReturn); ...
    min(CMresults.CM_OutSharpe); ...
    max(CMresults.CM_OutSharpe); ...
    cumCM(end)...
];
% 9.2 RW statistics vector
RWstats = [ ...
    mean(RWresults.RW_OutReturn); ...
    mean(RWresults.RW_OutVar); ...
    mean(RWresults.RW_OutSharpe); ...
    min(RWresults.RW_OutReturn); ...
    max(RWresults.RW_OutReturn); ...
    min(RWresults.RW_OutSharpe); ...
    max(RWresults.RW_OutSharpe); ...
    cumRW(end)...
];
% 9.2 RW statistics vector
BHstats = [ ...
    mean(BHresults.BH_OutReturn); ...
    mean(BHresults.BH_OutVar); ...
    mean(BHresults.BH_OutSharpe(3:end)); ...
    min(BHresults.BH_OutReturn); ...
    max(BHresults.BH_OutReturn); ...
    min(BHresults.BH_OutSharpe(3:end)); ...
    max(BHresults.BH_OutSharpe(3:end)); ...
    cumBH(end)...
];

% 9.3 Combine into a table
SummaryTable = table(CMstats, RWstats, BHstats,'RowNames', stats, 'VariableNames', {'ConstantMix','RollingWindow','BuyHold'});
% 9.4 Display
disp(SummaryTable);

% Define the tick step for x-axis spacing
tickStep = 20;

%% Buy-Hold Weights Plot
figure;
hold on;
for i = 2:width(BH_WeightsTable) % skip first column (TestMonth)
    plot(BH_WeightsTable.TestMonth, BH_WeightsTable{:, i}, '-o', 'LineWidth', 1.5, 'DisplayName', BH_WeightsTable.Properties.VariableNames{i});
end
xticks(BH_WeightsTable.TestMonth(1:tickStep:end));
```

```
xticklabels(datestr(BH_WeightsTable.TestMonth(1:tickStep:end), 'mmm-
    yyyy'));
xlabel('Month');
ylabel('Portfolio Weights');
title('Buy-Hold Strategy Weights Over Time');
legend('Location','northwest');
grid on;
hold off;

%% Constant Mix Weights Plot
figure;
hold on;
for i = 2:width(CM_WeightsTable)
    plot(CM_WeightsTable.TestMonth, CM_WeightsTable{:, i}, '-s', '
        LineWidth', 1.5, 'DisplayName', CM_WeightsTable.Properties.
        VariableNames{i});
end
xticks(CM_WeightsTable.TestMonth(1:tickStep:end));
xticklabels(datestr(CM_WeightsTable.TestMonth(1:tickStep:end), 'mmm-
    yyyy'));
xlabel('Month');
ylabel('Portfolio Weights');
title('Constant Mix Strategy Weights Over Time');
legend('Location','northwest');
grid on;
hold off;

%% Rolling Window Weights Plot
figure;
hold on;
for i = 2:width(RW_WeightsTable)
    plot(RW_WeightsTable.TestMonth, RW_WeightsTable{:, i}, '-^', '
        LineWidth', 1.5, 'DisplayName', RW_WeightsTable.Properties.
        VariableNames{i});
end
xticks(RW_WeightsTable.TestMonth(1:tickStep:end));
xticklabels(datestr(RW_WeightsTable.TestMonth(1:tickStep:end), 'mmm-
    yyyy'));
xlabel('Month');
ylabel('Portfolio Weights');
title('Rolling Window Strategy Weights Over Time');
legend('Location','northeast');
grid on;
hold off;

% Get asset names (skip TestMonth column)
assets = BH_WeightsTable.Properties.VariableNames(2:end);
% Extract initial weights (first row of BH table, could use any)
initialWeights = BH_WeightsTable{1, 2:end};
% Extract ending weights (last row of each table)
endingBH = BH_WeightsTable{end, 2:end};
endingCM = CM_WeightsTable{end, 2:end};
```

```
endingRW = RW_WeightsTable{end, 2:end};
% Create summary table
WeightsSummary = array2table([initialWeights; endingCM; endingBH; endingRW], ...
    'VariableNames', assets, ...
    'RowNames', {'Initial', 'End_CM', 'End_BH', 'End_RW'});
% Display the table
disp(WeightsSummary);
```

Experiment 2 (Additional Statistical Sophistication)

```
% 1. Testing Window Fractions
windowFracs = 0.1:0.05:0.9;
nFracs = length(windowFracs);

% 2. Preallocate summary table
SummaryWindow = table('Size',[nFracs, 13], ...
    'VariableTypes', repmat("double",1,13), ...
    'VariableNames', {'WindowFrac', 'CM_MeanReturn','CM_Std','CM_Sharpe','CM_Cum','RW_MeanReturn','RW_Std','RW_Sharpe','RW_Cum','BH_MeanReturn','BH_Std','BH_Sharpe','BH_Cum'});

% 3. Loop
for w = 1:nFracs
    windowFrac = windowFracs(w);
    nObs = size(allDataTable,1);
    windowSize = floor(windowFrac * nObs);
    nSteps = nObs - windowSize;

    CMresults = table('Size',[nSteps,3], ...
        'VariableTypes',{'double','double','double'}, ...
        'VariableNames',{'CM_OutReturn','CM_OutVar','CM_OutSharpe'});
    RWresults = table('Size',[nSteps,3], ...
        'VariableTypes',{'double','double','double'}, ...
        'VariableNames',{'RW_OutReturn','RW_OutVar','RW_OutSharpe'});
    BHresults = table('Size',[nSteps,3], ...
        'VariableTypes',{'double','double','double'}, ...
        'VariableNames',{'BH_OutReturn','BH_OutVar','BH_OutSharpe'});

    % 3. CM weights (fixed from first window)
    train_CM = allDataTable(1:windowSize,:);
    trainReturns_CMBH = tick2ret(train_CM,'Method',returnType);
    [Wts_CMBH,retSR_CMBH,rskSR_CMBH,~] = maxSharpeRatio(trainReturns_CMBH, tickers, riskFreeRate);

    % 4. Store in-sample Sharpe for CM
    inSharpeCMBH = sharpeRatio(retSR_CMBH, rskSR_CMBH, trainReturns_CMBH, riskFreeRate);

    % Buy hold set up
    assetValues = Wts_CMBH;
    portValues = zeros(nSteps + 1, 1);
```

```
    portValues(1) = sum(assetValues);
    portRetSeries = zeros(nSteps, 1);

    % Preallocate vectors to store one-month test returns for each strategy
    CM_portRetSeries = zeros(nSteps,1); % Constant Mix
    RW_portRetSeries = zeros(nSteps,1); % Rolling Window

    % 5. Loop for both RW and CM time-series
    for t = 1:nSteps
        % 5.1 Rolling Window Training
        train_RW = allDataTable(t:(t+windowSize-1),:);
        trainReturns_RW = tick2ret(train_RW,'Method',returnType);

        [Wts_RW,retSR_RW,rskSR_RW,~] = maxSharpeRatio(trainReturns_RW, tickers, riskFreeRate);
        inSharpeRW = sharpeRatio(retSR_RW, rskSR_RW, trainReturns_RW, riskFreeRate);

        % 5.2 Test month (same for CM and RW)
        testRow = [train_RW(end,:); allDataTable(t+windowSize,:)]; % one month forward
        testReturns = tick2ret(testRow,'Method',returnType);

        % 5.3 CM fixed weights (balancing)
        [cmMean, ~] = constantMix(testReturns, Wts_CMBH, tickers);
        CM_portRetSeries(t) = cmMean;
        cmSd = std(CM_portRetSeries);
        CMresults.CM_OutReturn(t) = cmMean;
        CMresults.CM_OutVar(t) = cmSd^2;
        CMresults.CM_OutSharpe(t) = sharpeRatio(cmMean, cmSd, testReturns, riskFreeTicker);

        % BH fixed weights (No rebalancing)
        % Use exp(returns) if continuous
        assetValues = assetValues .* (1+testReturns{:,tickers});
        portValues(t+1) = sum(assetValues);
        portRetSeries(t) = (portValues(t+1) - portValues(t)) / portValues(t);
        BHresults.BH_OutReturn(t) = mean(portRetSeries);
        BHresults.BH_OutVar(t) = std(portRetSeries)^2;
        BHresults.BH_OutSharpe(t) = sharpeRatioM2Y(mean(portRetSeries), std(portRetSeries), testReturns, riskFreeTicker);

        % 5.4 Rolling Window updated weights
        [rwMean, ~] = constantMix(testReturns, Wts_RW, tickers);
        RW_portRetSeries(t) = rwMean;
        rwSd = std(RW_portRetSeries);
        RWresults.RW_OutReturn(t) = rwMean;
        RWresults.RW_OutVar(t) = rwSd^2;
        RWresults.RW_OutSharpe(t) = sharpeRatio(rwMean, rwSd, testReturns, riskFreeTicker);
```

```
    end
    % Compute cumulative returns
    cumCM = cumprod(1 + CMresults.CM_OutReturn) - 1;
    cumRW = cumprod(1 + RWresults.RW_OutReturn) - 1;
    cumBH = cumprod(1 + BHresults.BH_OutReturn) - 1;

    % 3.3.5 Store summary metrics in the windowFrac table
    SummaryWindow.WindowFrac(w) = windowFrac;
    SummaryWindow.CM_MeanReturn(w) = mean(CMresults.CM_OutReturn);
    SummaryWindow.CM_Std(w) = sqrt(mean(CMresults.CM_OutVar));
    SummaryWindow.CM_Sharpe(w) = mean(CMresults.CM_OutSharpe);
    SummaryWindow.CM_Cum(w) = cumCM(end);
    SummaryWindow.RW_MeanReturn(w) = mean(RWresults.RW_OutReturn);
    SummaryWindow.RW_Std(w) = sqrt(mean(RWresults.RW_OutVar));
    SummaryWindow.RW_Sharpe(w) = mean(RWresults.RW_OutSharpe);
    SummaryWindow.RW_Cum(w) = cumRW(end);
    SummaryWindow.BH_MeanReturn(w) = mean(BHresults.BH_OutReturn);
    SummaryWindow.BH_Std(w) = sqrt(mean(BHresults.BH_OutVar));
    SummaryWindow.BH_Sharpe(w) = mean(BHresults.BH_OutSharpe(3:end));
    SummaryWindow.BH_Cum(w) = cumBH(end);
end

% 4. Display results
disp(SummaryWindow);

% 5. Plot Sharpe Ratio vs Window Fraction
figure;
plot(SummaryWindow.WindowFrac, SummaryWindow.CM_Sharpe, '-o', '
    LineWidth', 1.5);
hold on;
plot(SummaryWindow.WindowFrac, SummaryWindow.RW_Sharpe, '-s', '
    LineWidth', 1.5);
plot(SummaryWindow.WindowFrac, SummaryWindow.BH_Sharpe, '-s', '
    LineWidth', 1.5);
xlabel('Window Fraction');
ylabel('Out-of-Sample Sharpe Ratio');
title('Out-of-Sample Sharpe Ratio vs Training Window Fraction');
legend('Constant-Mix', 'Rolling-Window','Buy-Hold', 'Location','
    northeastoutside');
grid on;
hold off;

% 6. Cumulative Plot
% Extract the relevant data
windowFrac = SummaryWindow.WindowFrac;
cumCM = SummaryWindow.CM_Cum;
cumRW = SummaryWindow.RW_Cum;
cumBH = SummaryWindow.BH_Cum;
% Create the plot
figure;
plot(windowFrac, cumCM, '-o', 'LineWidth',1.5, 'DisplayName','Constant
    Mix');
```

```
hold on;
plot(windowFrac, cumRW, '-s', 'LineWidth',1.5, 'DisplayName','Rolling Window');
plot(windowFrac, cumBH, '-^', 'LineWidth',1.5, 'DisplayName','Buy-Hold');
xlabel('Training Fraction (WindowFrac)');
ylabel('Cumulative Return');
title('Cumulative Return vs Training Window Fraction');
legend('Location','northeastoutside');
grid on;
hold off;
```

References

Embrechts, P., Klüppelberg, C., & Mikosch, T. (1997). *Modelling extremal events: For insurance and finance* (Vol. 33). Springer-Verlag. https://doi.org/10.1007/978-3-642-33483-2

Lo, A. W. (2002). The statistics of sharpe ratios. *Financial Analysts Journal*, *58*(4), 36–52. https://doi.org/10.2469/faj.v58.n4.2453

Resnick, S. I. (1987). *Extreme values, regular variation and point processes*. Springer. https://doi.org/10.1007/978-0-387-75953-1