Time-Varying Global Minimum-Variance Portfolio for Three Oil Majors

One-factor Stochastic Volatility Model + Per-Period GMV Allocation

Hedibert F. Lopes

May 2026

Contents

1 Overview

This document fits a **one-factor stochastic volatility model** to the daily standardised returns of three large integrated oil companies and then uses the resulting *time-varying* posterior conditional covariance matrix $\mathbf{\Sigma}_t$ to compute the **global minimum-variance (GMV) portfolio weights at each date** t. The strategy combines two standard ingredients:

1. **Factor stochastic volatility (FSV)** of Aguilar & West (2000), implemented in the R package `factorstochvol` (Kastner, Fr"uhwirth-Schnatter & Lopes, 2017), which delivers a Bayesian posterior over a parsimonious time-varying covariance trajectory.

2. **The closed-form unconstrained GMV formula** $\boldsymbol{w}_t = \boldsymbol{\Sigma}_t^{-1}\mathbf{1}/(\mathbf{1}^\top\boldsymbol{\Sigma}_t^{-1}\mathbf{1})$, evaluated at each t on the posterior-mean covariance.

The end product is a *time series of optimal weights* and an associated time series of portfolio standard deviations. Because the assets are all in the same sector, the portfolio standard deviation is naturally bounded above by the smallest individual standard deviation — a diversification effect we can read directly off the plots.

2 Data: ExxonMobil, Chevron, Shell

We pull adjusted closing prices for the three tickers from Yahoo Finance over ~ 26 years.

```
# install.packages(c("quantmod","stochvol","factorstochvol","MTS"))
library(quantmod)
library(stochvol)
library(factorstochvol)
library(MTS)
```

```
tickers <- c("XOM",        "CVX",     "SHEL")
names   <- c("ExxonMobil", "Chevron", "Shell")

getSymbols(tickers, from = "2000-05-15", to = "2026-05-14",
           warnings = FALSE, auto.assign = TRUE)
```

```
## [1] "XOM"  "CVX"  "SHEL"
```

```
price     <- as.matrix(cbind(XOM[,6], CVX[,6], SHEL[,6]))   # adjusted close
n         <- nrow(price)
p         <- ncol(price)
price.std <- scale(price)
colnames(price) <- colnames(price.std) <- names
```

```
par(mfrow = c(1, 2))
ts.plot(price,     col = 1:p, ylab = "Price")
ts.plot(price.std, col = 1:p, ylab = "Standardised price")
legend("topleft", legend = names, col = 1:p, bty = "n", lty = 1, lwd = 3)
```

We work with **simple daily percentage returns** $r_{it} = 100\,(P_{it}/P_{i,t-1} - 1)$, then standardise each series so each column has unit unconditional sample variance. The standardisation helps the FSV sampler mix and keeps the loadings on a comparable scale. The conditional covariance $\boldsymbol{\Sigma}_t$ that we extract — and the GMV weights it implies — are therefore on the **standardised** return scale. Re-scaling to raw returns is straightforward but not done here.

```
return <- 100 * (price[2:n, ] / price[1:(n-1), ] - 1)
return <- return[stats::complete.cases(return), ]
n      <- nrow(return)
ret.std <- scale(return)
colnames(ret.std) <- names
```

```
par(mfrow = c(2, 3))
for (i in 1:p)
```

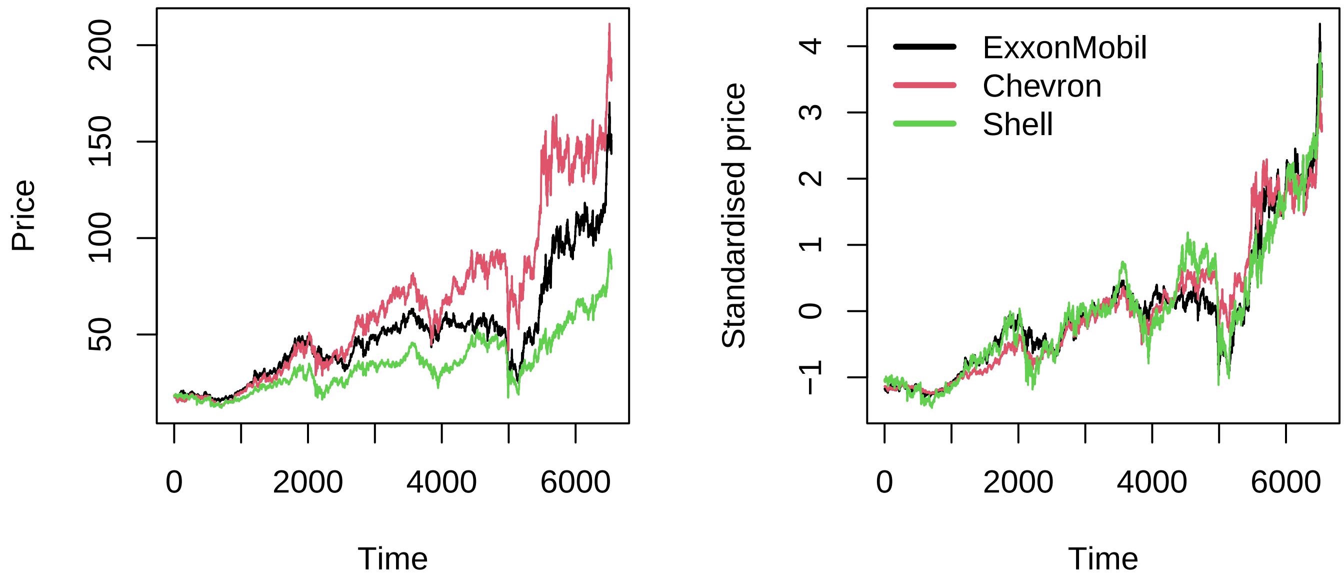

Figure 1: Levels (left) and standardised levels (right).

```
  ts.plot(return[, i], ylab = "Returns (%)", main = names[i])
for (i in 1:p)
  ts.plot(ret.std[, i], ylab = "Std. returns", main = names[i],
          ylim = range(ret.std))
```

3 The factor stochastic volatility model

For the standardised return vector $\boldsymbol{y}_t \in \mathbb{R}^p$ we assume the **single-factor** representation

$$\boldsymbol{y}_t \;=\; \boldsymbol{\beta}\, f_t \;+\; \boldsymbol{\varepsilon}_t, \qquad t = 1, \ldots, n,$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ are the factor loadings, f_t is the (scalar) latent factor, and $\boldsymbol{\varepsilon}_t \in \mathbb{R}^p$ is idiosyncratic noise. Both the factor and the idiosyncratic shocks have **stochastic volatility**:

$$f_t = e^{h_t/2}\, \eta_t, \qquad \varepsilon_{it} = e^{\ell_{it}/2}\, \xi_{it}, \qquad \eta_t,\ \xi_{it} \overset{iid}{\sim} N(0,1),$$

with independent AR(1) log-volatility processes,

$$h_t = \mu_h + \phi_h(h_{t-1} - \mu_h) + \sigma_h\, \omega_t^h, \qquad \ell_{it} = \mu_{\ell,i} + \phi_{\ell,i}(\ell_{i,t-1} - \mu_{\ell,i}) + \sigma_{\ell,i}\, \omega_{it}^{\ell}.$$

Conditional on the loadings and on the volatility states, the **conditional covariance** is

$$\boxed{\boldsymbol{\Sigma}_t \;=\; e^{h_t}\, \boldsymbol{\beta}\boldsymbol{\beta}^\top \;+\; \operatorname{diag}(e^{\ell_{1t}}, \ldots, e^{\ell_{pt}}).}$$

The first term is the *common* component (the same factor drives all three series, modulated by its current variance e^{h_t}); the second term is the *asset-specific* idiosyncratic variance.

A flexible Gaussian prior on $\boldsymbol{\beta}$, log-Normal-Wishart-type priors on the AR(1) parameters, and the Kim–Shephard–Chib mixture-of-normals approximation make this model fully amenable to a Gibbs sampler. The package `factorstochvol` implements the version with the *ASIS* (ancillarity-sufficiency interweaving) re-parameterisation, which dramatically improves mixing for the latent log-volatilities.

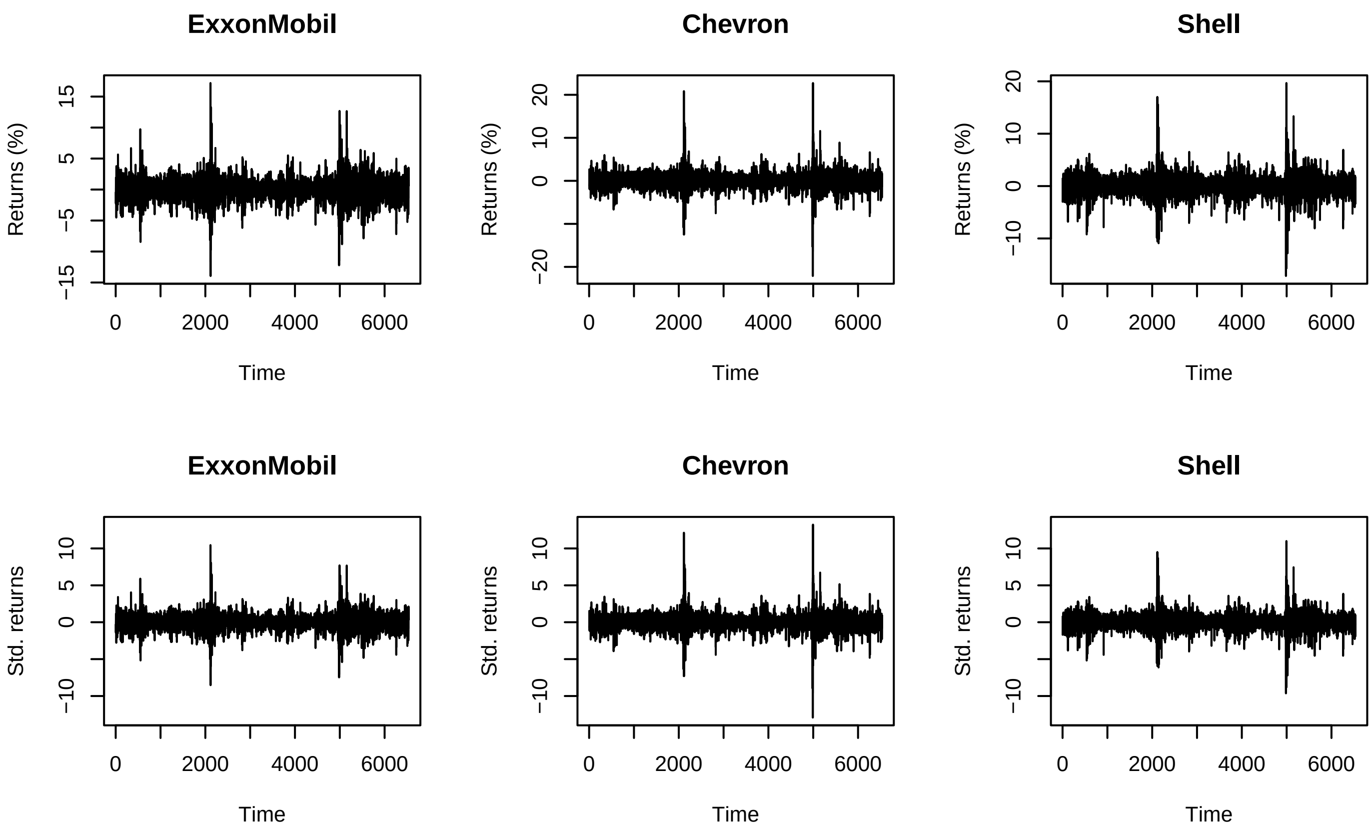

Figure 2: Daily percentage returns (top row) and standardised returns (bottom row) for the three oil majors.

4 Posterior inference

```
fsv <- fsvsample(ret.std, factors = 1, runningstore = 6)
```

`runningstore = 6` is the highest level: it asks the sampler to accumulate, on the fly, posterior summaries (mean and quantiles) of the factor and idiosyncratic log-variances, the latent factor, the conditional covariance $\boldsymbol{\Sigma}_t$, and the conditional correlation $\boldsymbol{R}_t$ at every date t. We will use the posterior mean of $\boldsymbol{\Sigma}_t$ for the portfolio computation.

4.1 Factor loadings

```
par(mfrow = c(1, 1))
plot(density(fsv$facload[1, 1, ]),
     xlim = range(fsv$facload), ylim = c(0, 13),
     main = "Factor loadings", xlab = "", lwd = 3)
for (i in 2:p)
  lines(density(fsv$facload[i, 1, ]), col = i, lwd = 3)
legend("topright", legend = names, col = 1:p, bty = "n", lty = 1, lwd = 3)
```

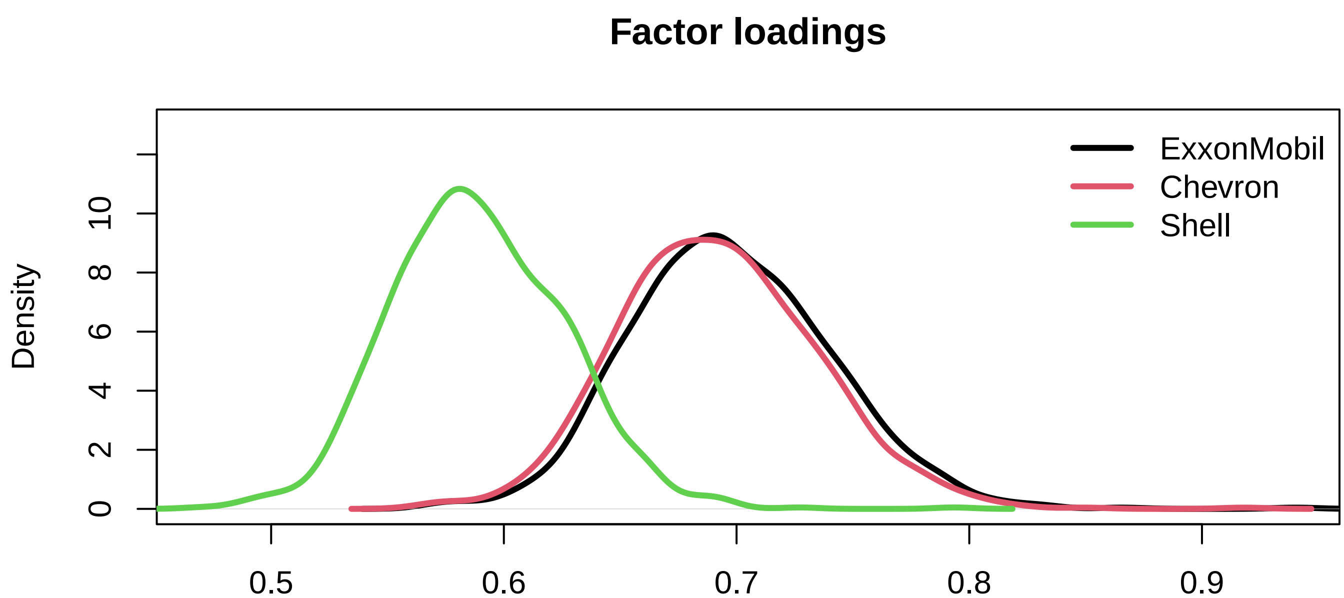

Figure 3: Posterior densities of the three factor loadings β_i. A common factor that loads positively on all three series is what one would expect from oil majors.

4.2 Time-varying volatilities

The package stores symmetric $p \times p$ matrices in lower-triangular column-major order, so for $p = 3$ the six columns of `runningstore$cov` correspond to $(\sigma_{11}, \sigma_{21}, \sigma_{31}, \sigma_{22}, \sigma_{32}, \sigma_{33})$. We pull out the variance terms (positions 1, 4, 6) and the covariance terms (positions 2, 3, 5).

```
diag_idx <- c(1, 4, 6)            # variances
off_idx  <- c(2, 3, 5)            # covariances
```

```
sds   <- matrix(0, n, p)
for (i in 1:p)
  sds[, i] <- sqrt(fsv$runningstore$cov[, diag_idx[i], 1])

Sigma <- array(0, c(p, p, n))
for (i in 1:p)
  Sigma[i, i, ] <- sds[, i]^2

l <- 0
for (i in 2:p)
  for (j in 1:(i-1)) {
    l <- l + 1
    Sigma[i, j, ] <- fsv$runningstore$cov[, off_idx[l], 1]
    Sigma[j, i, ] <- Sigma[i, j, ]
  }
```

```
par(mfrow = c(1, 1))
ts.plot(sds, col = 1:p, ylab = "Conditional standard deviation")
legend("topright", legend = names, col = 1:p, bty = "n", lty = 1, lwd = 3)
```

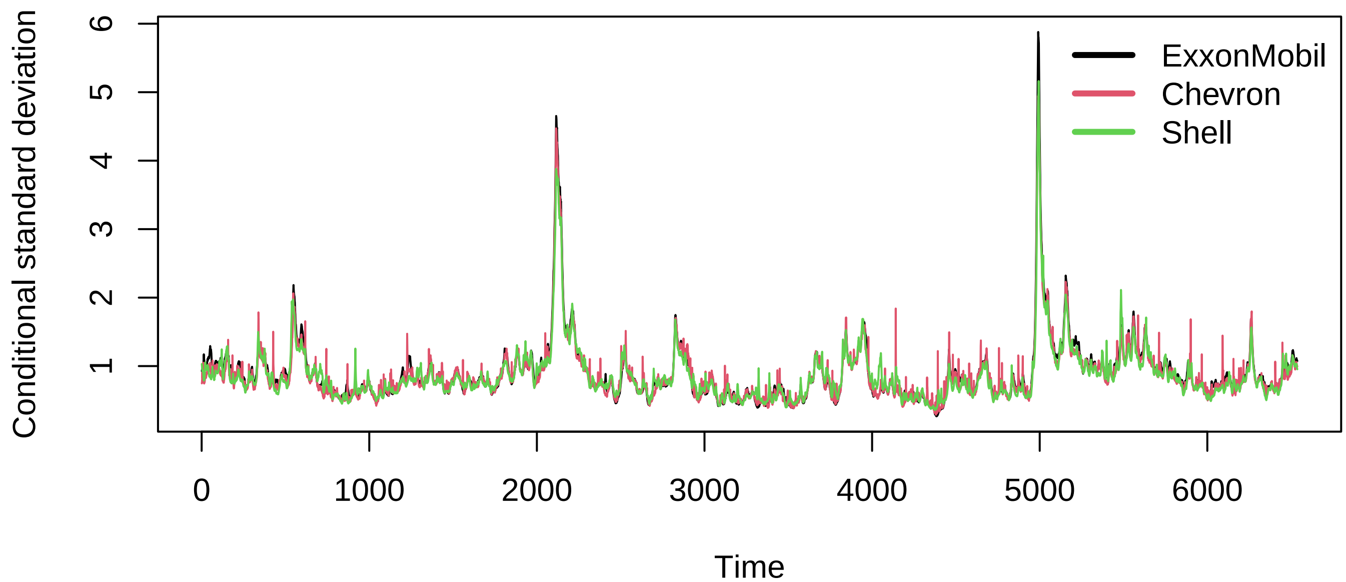

Figure 4: Posterior-mean conditional standard deviations over the full sample.

```
ts.plot(sds[(n-499):n, ], col = 1:p, ylab = "Conditional standard deviation")
legend("topright", legend = names, col = 1:p, bty = "n", lty = 1, lwd = 3)
title("Most recent 500 days")
```

4.3 Time-varying correlations

The conditional correlations $\rho_{ij,t}$ summarise how tightly the three series move together at each date. With a single common factor and persistent idiosyncratic log-variances, $\rho_{ij,t}$ tracks roughly with the *relative* size of the common-component contribution $e^{h_t}\beta_i\beta_j$ versus the diagonal pieces.

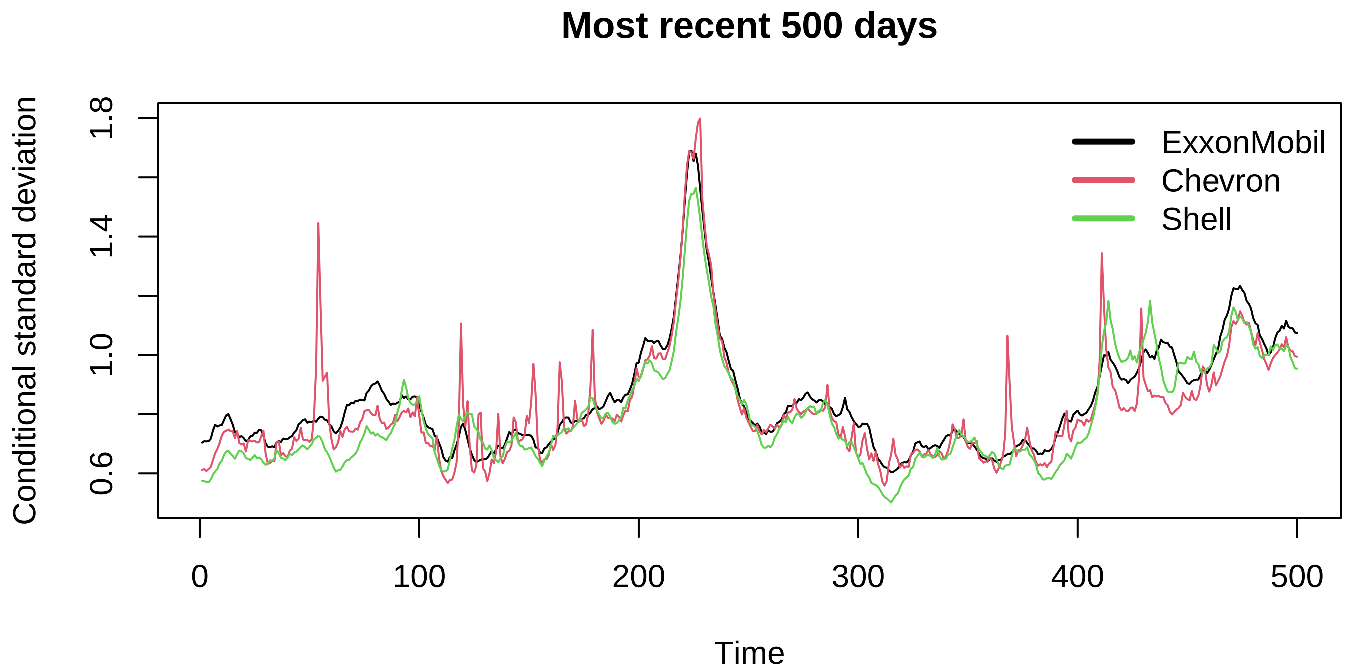

Figure 5: The last 500 trading days, May 2024 to May 2026.

```
par(mfrow = c(2, 2))
ts.plot(fsv$runningstore$cor[1:750,          , 1], col = 1:3, ylim = c(0,1)); title("Era 1")
ts.plot(fsv$runningstore$cor[751:1500,       , 1], col = 1:3, ylim = c(0,1)); title("Era 2")
ts.plot(fsv$runningstore$cor[1501:2250,      , 1], col = 1:3, ylim = c(0,1)); title("Era 3")
ts.plot(fsv$runningstore$cor[2251:n,         , 1], col = 1:3, ylim = c(0,1)); title("Era 4")
```

```
par(mfrow = c(1, 1))
ts.plot(fsv$runningstore$cor[(n-499):n, , 1], col = 1:3, ylim = c(0, 1))
abline(h = c(0.2, 0.5, 0.8), lty = 2)
title("Most recent 500 days")
legend("bottomleft",
       legend = c("CVX/XOM", "SHEL/XOM", "SHEL/CVX"),
       col = 1:3, bty = "n", lty = 1, lwd = 3)
```

4.4 A diagnostic: returns versus the posterior-mean factor

A useful sanity check: the (sample) correlation between each asset's standardised returns and the posterior-mean trajectory of the latent factor should be sizeable and positive — consistent with the loading posteriors above.

```
round(cor(cbind(ret.std, factor = fsv$runningstore$fac[1, , 1])), 3)
```

```
##            ExxonMobil Chevron Shell  factor
## ExxonMobil      1.000   0.842 0.752   0.923
## Chevron         0.842   1.000 0.760   0.959
## Shell           0.752   0.760 1.000   0.834
## factor          0.923   0.959 0.834   1.000
```

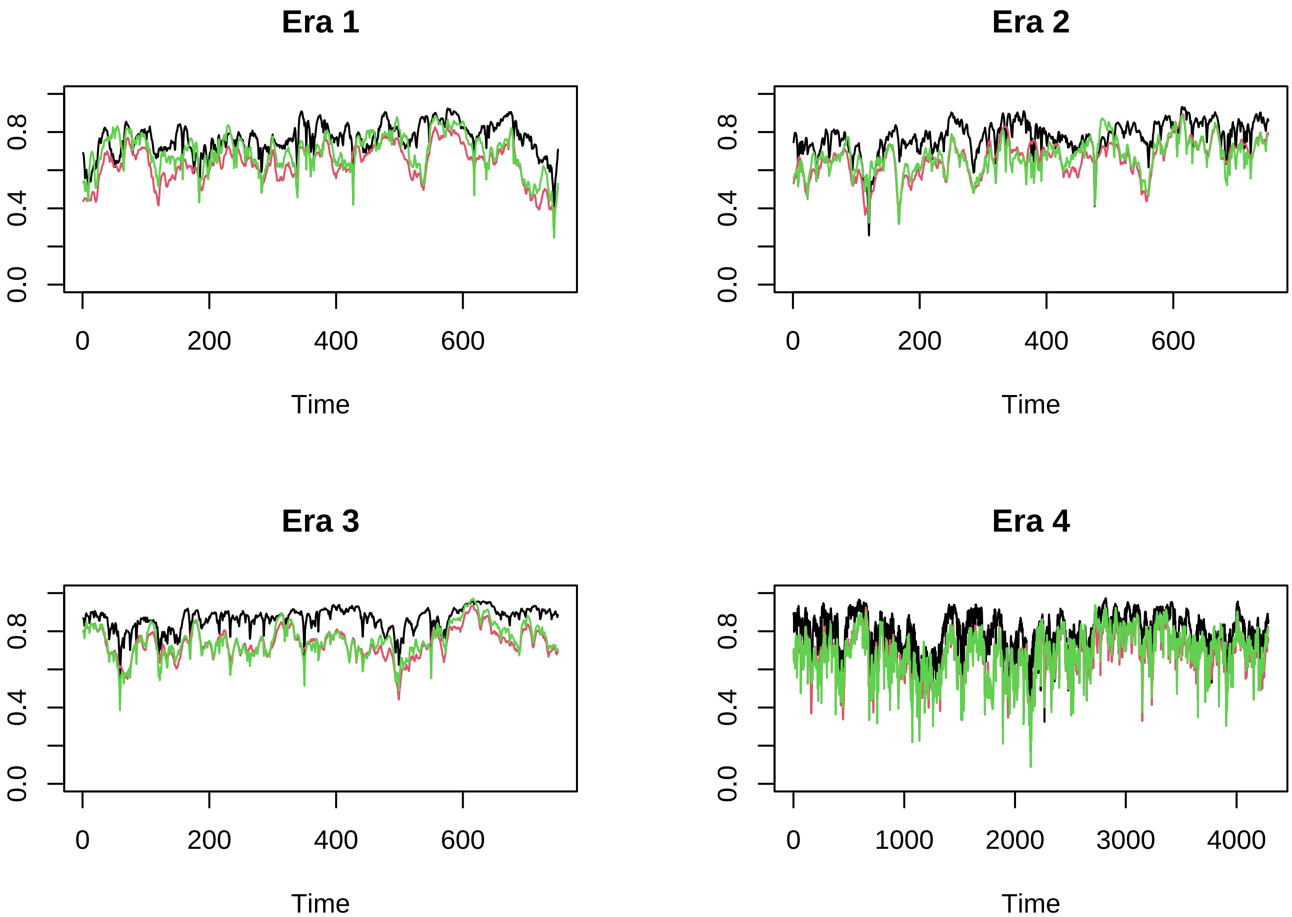

Figure 6: Pairwise conditional correlations split into four equal eras.

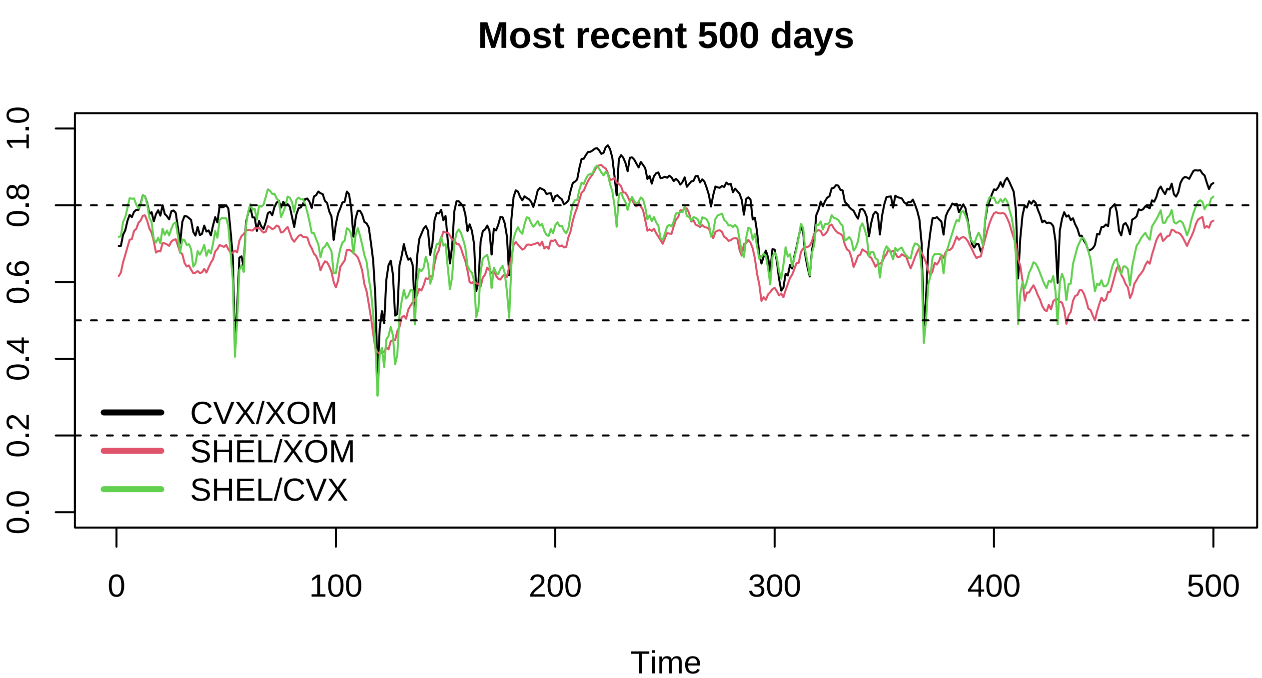

Figure 7: The last 500 trading days. Reference lines at 0.2, 0.5, 0.8.

5 Time-varying global minimum-variance portfolio

For each date t we compute the **unconstrained** (short sales allowed) global minimum-variance portfolio

$$\boldsymbol{w}_t \;=\; \arg\min_{\boldsymbol{w}} \; \boldsymbol{w}^\top \boldsymbol{\Sigma}_t \boldsymbol{w} \quad \text{s.t.} \quad \boldsymbol{w}^\top \mathbf{1} = 1,$$

which has the closed-form solution

$$\boldsymbol{w}_t \;=\; \frac{\boldsymbol{\Sigma}_t^{-1}\mathbf{1}}{\mathbf{1}^\top \boldsymbol{\Sigma}_t^{-1}\mathbf{1}}, \qquad \sigma_t^{\text{GMV}} \;=\; \sqrt{\frac{1}{\mathbf{1}^\top \boldsymbol{\Sigma}_t^{-1}\mathbf{1}}}.$$

Implementation is one line per date:

```
ones      <- matrix(1, p, 1)
w.gmv     <- matrix(0, n, p)
sig2.gmv  <- numeric(n)

for (t in 1:n) {
  iSigma      <- solve(Sigma[, , t])
  den         <- as.numeric(t(ones) %*% iSigma %*% ones)
  sig2.gmv[t] <- 1 / den
  w.gmv[t, ]  <- iSigma %*% ones / den
}
sig.gmv <- sqrt(sig2.gmv)
colnames(w.gmv) <- names
```

5.0.0.1 Plug-in vs. fully Bayesian. Because we feed the per-t solver the *posterior-mean* covariance, $\boldsymbol{w}_t$ is a **plug-in** time-varying portfolio. A fully Bayesian treatment would propagate posterior uncertainty in $\boldsymbol{\Sigma}_t$ through the QP/closed-form for every retained MCMC draw, yielding a posterior distribution of weights at each t (cf. the companion R Markdown on posterior portfolios). The plug-in version used here is substantially cheaper and is the right starting point.

6 Results

6.1 Weights over time

```
par(mfrow = c(2, 2))
c1 <- adjustcolor("steelblue", alpha.f = 0.45)
c2 <- adjustcolor("firebrick", alpha.f = 0.45)
c3 <- adjustcolor("darkgreen", alpha.f = 0.45)

ts.plot(w.gmv[, 1], ylim = c(-1.1, 1.1), col = c1,
        ylab = "Weight", main = names[1])
abline(h = 0, lty = 2); abline(h = 1, lty = 2)

ts.plot(w.gmv[, 2], ylim = c(-1.1, 1.1), col = c2,
        ylab = "Weight", main = names[2])
abline(h = 0, lty = 2); abline(h = 1, lty = 2)
```

```r
ts.plot(w.gmv[, 1] + w.gmv[, 2], ylim = c(-1.1, 1.1), col = c3,
        ylab = "Weight", main = paste0(names[1], " + ", names[2]))
abline(h = 0, lty = 2); abline(h = 1, lty = 2)

brks <- seq(min(w.gmv), max(w.gmv), length.out = 40)
ymax <- max(sapply(1:p, function(i) max(hist(w.gmv[, i], brks, plot = FALSE)$density)))
hist(w.gmv[, 1], breaks = brks, freq = FALSE, col = c1, border = "white",
     xlab = "Weight", main = "", ylim = c(0, ymax))
hist(w.gmv[, 2], breaks = brks, freq = FALSE, col = c2, border = "white", add = TRUE)
hist(w.gmv[, 3], breaks = brks, freq = FALSE, col = c3, border = "white", add = TRUE)
legend("topright", names, fill = c(c1, c2, c3), bty = "n")
```

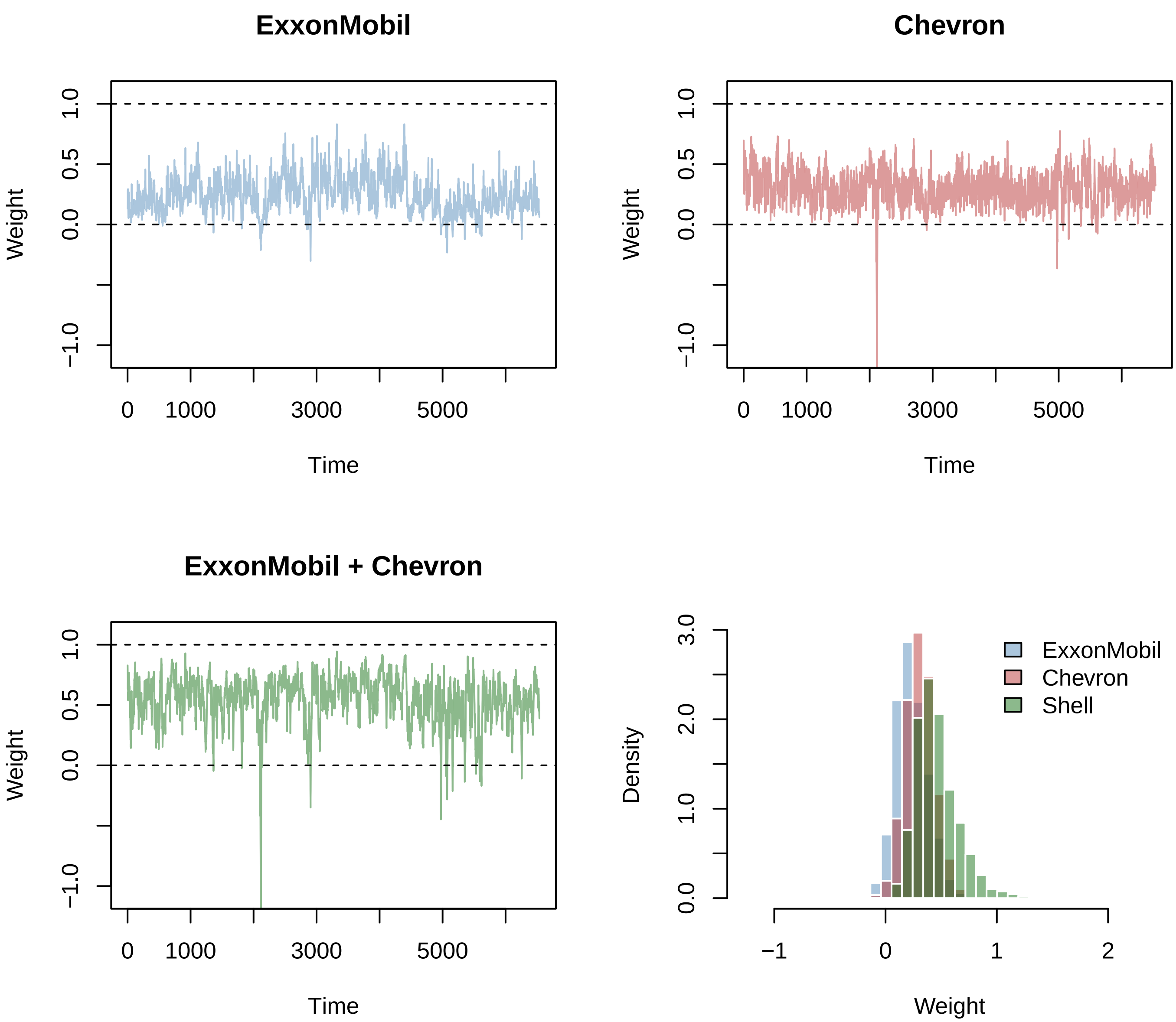

Figure 8: GMV weights through time. Top-left: ExxonMobil. Top-right: Chevron. Bottom-left: ExxonMobil + Chevron (i.e. $1 - w_{\text{Shell}}$). Bottom-right: overlay of the marginal posterior histograms.

Table 1: Standard deviations (standardised-return scale): 2.5%, 50%, 97.5% quantiles across all dates.

	2.5%	50%	97.5%
ExxonMobil	0.433	0.786	1.910
Chevron	0.444	0.775	1.813
Shell	0.460	0.777	1.814
GMV portfolio	0.381	0.696	1.706

The third panel plots $w_{1t} + w_{2t}$, which by the budget constraint equals $1 - w_{3t}$ — a one-glance summary of how much of the portfolio is in the two American majors versus Shell at each date. The overlay histogram makes the marginal time-distribution of each weight directly comparable.

6.2 Risk reduction: GMV portfolio versus individual assets

By construction, the GMV portfolio's standard deviation σ_t^{GMV} cannot exceed the smallest individual σ_{it}. The scatter plots below place each σ_{it} on the horizontal axis and the corresponding σ_t^{GMV} on the vertical axis. All points should lie *on or below* the 45° line — which they do, often with substantial slack. The vertical gap is the diversification benefit at date t.

```
par(mfrow = c(1, 3))
for (i in 1:p) {
  plot(sds[, i], sig.gmv, col = grey(0.7), pch = 16,
       ylab = "GMV portfolio std. dev.",
       xlab = paste0(names[i], " std. dev."))
  abline(0, 1, lwd = 2)
}
```

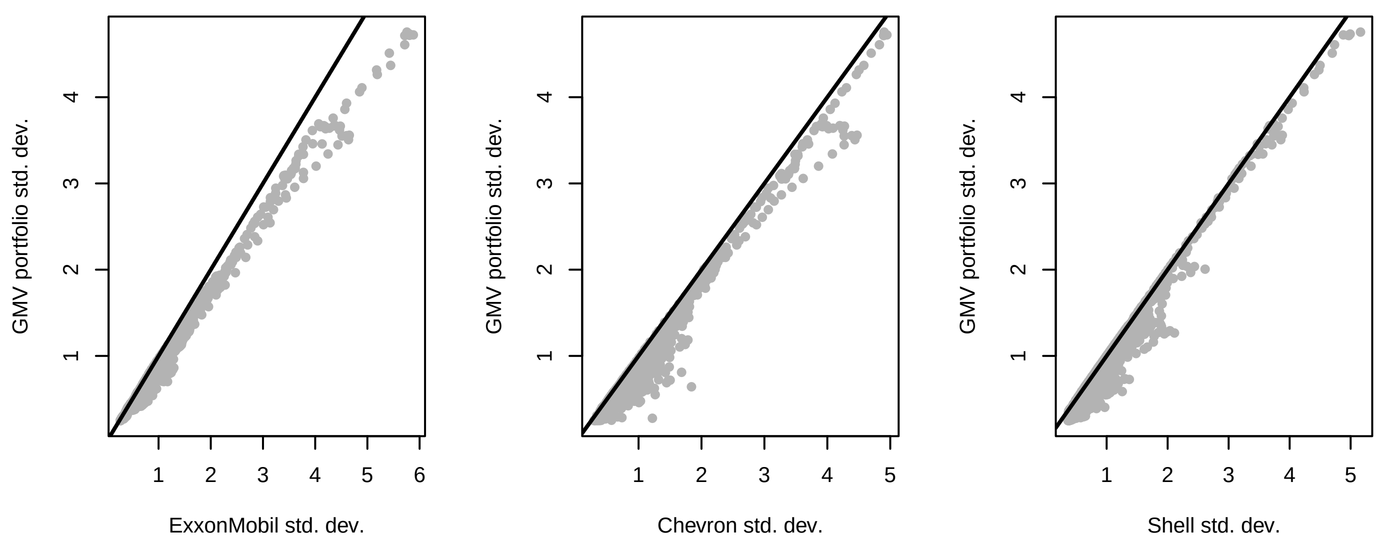

Figure 9: GMV portfolio standard deviation against each asset's individual standard deviation. The 45° line is the upper bound; vertical gaps are the diversification benefit at that date.

7 Discussion

7.0.0.1 Why one factor is enough. For three oil majors that all load on the same crude-oil cycle, refining margins, and risk-on / risk-off flows, a single common factor explains a very large share of the co-movement. The loadings posteriors confirm this: all three are positive, of similar magnitude, and well separated from zero. Adding a second factor (set `factors = 2` in `fsvsample`) would reveal whether there is a residual idiosyncratic European/American or upstream/downstream split, but it is not needed for the covariance-driven portfolio decision considered here.

7.0.0.2 Why the GMV weights move so much. The unconstrained GMV portfolio is famously sensitive to small changes in $\mathbf{\Sigma}_t^{-1}$ when the assets are highly correlated, as oil majors are. The time-varying nature of $\mathbf{\Sigma}_t$ then translates directly into weight variation, occasionally producing short positions in one of the three assets. The long-only version (Problems A and B in the companion document) tames these swings at the cost of a slightly higher portfolio standard deviation.

7.0.0.3 Plug-in vs. fully Bayesian, revisited. Replacing the posterior mean $\widehat{\mathbf{\Sigma}}_t$ by individual MCMC draws $\mathbf{\Sigma}_t^{(m)}$ would let us produce, at each t, a *posterior distribution* over weights instead of a single trajectory. That is the right object if one wants to quantify how confident the model is about the current allocation. The framework is a direct extension of what is done above and uses the same closed-form formula — just looped over draws as well as over t.

7.0.0.4 Standardised vs. raw returns. We fit the FSV model to standardised returns. The resulting $\mathbf{\Sigma}_t$ and the GMV weights are therefore on the standardised scale. Re-scaling to raw returns is a multiplication by the diagonal sample-SD matrix; the *qualitative* pattern (which asset gets more weight when, the time-variation of the diversification benefit) is the same on both scales when individual volatilities are similar, as is the case for these three majors.

8 Reproducibility

```
## R version 4.5.1 (2025-06-13)
## Platform: aarch64-apple-darwin20
## Running under: macOS Tahoe 26.3.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRlapack.dylib;
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Sao_Paulo
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
```

```
## other attached packages:
## [1] MTS_1.2.1            factorstochvol_1.1.2 stochvol_3.2.9
## [4] quantmod_0.4.28      TTR_0.24.4           xts_0.14.1
## [7] zoo_1.8-14
##
## loaded via a namespace (and not attached):
##  [1] cli_3.6.5           knitr_1.50          rlang_1.1.6
##  [4] xfun_0.56           jsonlite_2.0.0      fGarch_4033.92
##  [7] corrplot_0.95       timeSeries_4041.111 htmltools_0.5.8.1
## [10] tinytex_0.57        rmarkdown_2.30      grid_4.5.1
## [13] evaluate_1.0.5      fastmap_1.2.0       yaml_2.3.10
## [16] mvtnorm_1.3-3       cvar_0.5            compiler_4.5.1
## [19] codetools_0.2-20    GIGrvg_0.8          coda_0.19-4.1
## [22] timeDate_4051.111   Rcpp_1.1.1          rstudioapi_0.18.0
## [25] lattice_0.22-7      digest_0.6.37       Rdpack_2.6.6
## [28] curl_7.0.0          parallel_4.5.1      gbutils_0.5
## [31] rbibutils_2.4.1     Matrix_1.7-3        tools_4.5.1
## [34] fBasics_4041.97     spatial_7.3-18
```