Solutions to Exercises

Portfolio Optimization: Theory and Application
Chapter 10 – Portfolios with Alternative Risk Measures

Daniel P. Palomar (2025). *Portfolio Optimization: Theory and Application.* Cambridge University Press.

portfoliooptimizationbook.com

Exercise 10.1: Computing alternative measures of risk

Generate 10 000 loss_samples following a normal distribution, plot the histogram, and compute the following measures:

- mean
- variance and standard deviation
- semi-variance and semi-deviation
- tail measures (VaR, CVaR, and EVaR) based on raw data
- tail measures (VaR, CVaR, and EVaR) based on a Gaussian approximation.

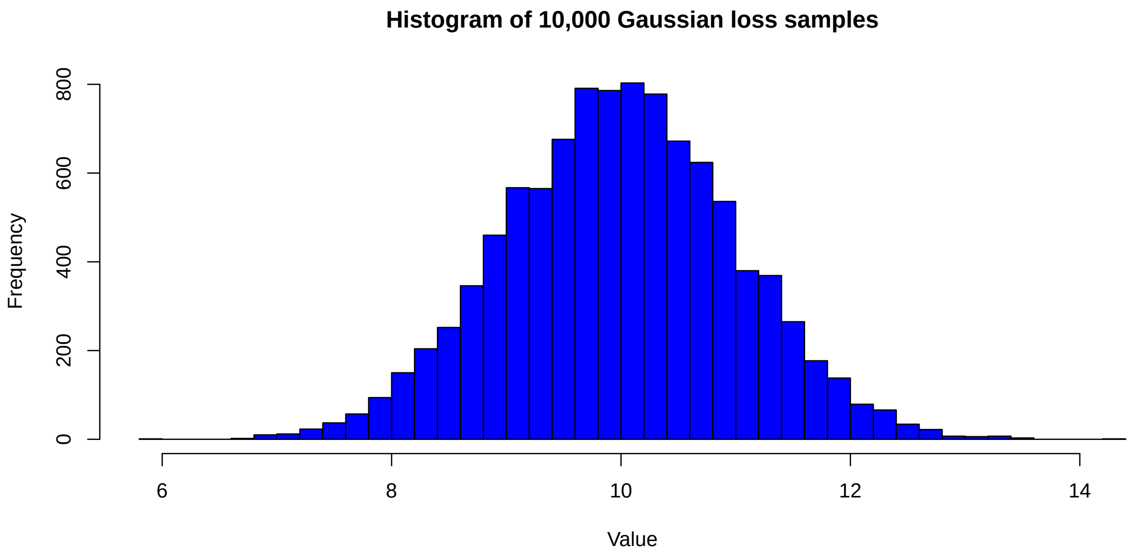
Solution

First, we generate the data:

```
set.seed(42)

n <- 10000
loss_samples <- rnorm(n, mean = 10)
```

Then, plot the histogram:

```
hist(loss_samples, breaks = 40, col = "blue", main = "Histogram of 10,000 Gaussian loss samples",
     xlab = "Value", ylab = "Frequency")
```



Finally, compute the risk measures:

```r
# Basic statistics
mean_val <- mean(loss_samples)
var_val <- var(loss_samples)
sd_val <- sd(loss_samples)

# Semi-variance and semi-deviation (below the mean)
semi_loss_samples <- loss_samples[loss_samples < mean_val]
semi_var <- sum((semi_loss_samples - mean_val)^2)/n
semi_sd <- sqrt(semi_var)

# Tail measures (empirical, alpha = 0.95)
alpha <- 0.95
VaR_emp <- quantile(loss_samples, alpha)
CVaR_emp <- mean(loss_samples[loss_samples >= VaR_emp])

# EVaR (Entropic Value at Risk):
EVaR_empirical <- function(loss_samples, alpha = 0.95) {
  evar_z <- function(z) {
    if (z <= 0) return(Inf)
    exp_term <- mean(exp(z * loss_samples))
    (1/z) * log(exp_term / (1 - alpha))
  }
  opt <- optimize(evar_z, interval = c(1e-8, 10))
  list(EVaR = opt$objective, z_opt = opt$minimum)
}

EVaR_emp <- EVaR_empirical(loss_samples, alpha)$EVaR


# Tail measures (Gaussian approximation)
mu_hat <- mean_val
sigma_hat <- sd_val

# VaR_alpha = mu + sigma * Phi^(-1)(alpha)
VaR_gauss <- mu_hat + sigma_hat * qnorm(alpha)

# CVaR_alpha = mu + sigma * phi(Phi^(-1)(alpha)) / (1 - alpha)
CVaR_gauss <- mu_hat + sigma_hat * dnorm(qnorm(alpha)) / (1 - alpha)

# EVaR_alpha
EVaR_gauss <- mu_hat + sigma_hat * sqrt(-2 * log(1-alpha))
```

```r
# Print results
cat("Mean:", mean_val, "\n")
cat("Variance:", var_val, "\n")
cat("Standard deviation:", sd_val, "\n")
cat("Semi-variance:", semi_var, "\n")
cat("Semi-deviation:", semi_sd, "\n\n")

cat("Empirical VaR (5%):", VaR_emp, "\n")
cat("Empirical CVaR (5%):", CVaR_emp, "\n")
cat("Empirical EVaR (5%):", EVaR_emp, "\n\n")

cat("Gaussian VaR (5%):", VaR_gauss, "\n")
cat("Gaussian CVaR (5%):", CVaR_gauss, "\n")
cat("Gaussian EVaR (5%):", EVaR_gauss, "\n")
```

```
Mean: 9.988691
Variance: 1.012307
Standard deviation: 1.006135
Semi-variance: 0.5063872
Semi-deviation: 0.7116089

Empirical VaR (5%): 11.64691
Empirical CVaR (5%): 12.06624
Empirical EVaR (5%): 12.4911

Gaussian VaR (5%): 11.64364
Gaussian CVaR (5%): 12.06406
Gaussian EVaR (5%): 12.45145
```

Exercise 10.2: CVaR in variational convex form

Consider the following expression for the CVaR:

$$\mathrm{CVaR}_\alpha = \mathbb{E}\left[\xi \mid \xi \geq \mathrm{VaR}_\alpha\right].$$

Show that it can be rewritten in a convex variational form as:

$$\mathrm{CVaR}_\alpha = \inf_\tau \left\{ \tau + \frac{1}{1-\alpha} \mathbb{E}\left[(\xi - \tau)^+\right] \right\},$$

where the optimal τ precisely equals VaR_α.

Solution

Define the auxiliary function

$$F_\alpha(\boldsymbol{w}, \tau) = \tau + \frac{1}{1-\alpha}\mathbb{E}\left[(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r} - \tau)^+\right],$$

where we have defined the loss of portfolio $\boldsymbol{w}$ as $\xi = -\boldsymbol{w}^\mathsf{T}\boldsymbol{r}$ for concreteness.
The minimizer of $F_\alpha(\boldsymbol{w}, \tau)$ with respect to τ satisfies: $0 \in \partial_\tau F_\alpha(\boldsymbol{w}, \tau^\star)$, where ∂ denotes the subdifferential, which is a set containing all the subgradients (if the function is differentiable at that point, then the subdifferential will contain a single element equal to the partial derivative).
We choose the following subgradient s_τ:

$$\begin{aligned} 0 = s_\tau F_\alpha(\boldsymbol{w}, \tau^\star) &= 1 - \frac{1}{1-\alpha}\int \mathbf{1}_{\{-\boldsymbol{w}^\mathsf{T}\boldsymbol{r} > \tau^\star\}} p(\boldsymbol{r}) d\boldsymbol{r} \\ &= 1 - \frac{1}{1-\alpha}\Pr\left(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r} > \tau^\star\right), \end{aligned}$$

where $\mathbf{1}_{\{.\}}$ is the indicator function. Solving the above equation, we have

$$\Pr\left(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r} > \tau^\star\right) = 1 - \alpha \Longrightarrow \tau^\star = \mathrm{VaR}_\alpha(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r}).$$

Now that we have characterized the VaR, we can proceed to analyze the CVaR. First, note that

$$\min_\tau F_\alpha(\boldsymbol{w}, \tau) = F_\alpha(\boldsymbol{w}, \tau^\star) = \tau^\star + \frac{1}{1-\alpha}\mathbb{E}\left[(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r} - \tau^\star)^+\right].$$

Finally, recall that

$$\begin{aligned} \mathrm{CVaR}_\alpha(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r}) &= \mathrm{E}\left[-\boldsymbol{w}^\mathsf{T}\boldsymbol{r} \middle| -\boldsymbol{w}^\mathsf{T}\boldsymbol{r} > \mathrm{VaR}_\alpha(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r})\right] \\ &= \frac{1}{1-\alpha}\int_{-\boldsymbol{w}^\mathsf{T}\boldsymbol{r} > \mathrm{VaR}_\alpha(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r})} \left(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r}\right) p(\boldsymbol{r}) d\boldsymbol{r} \\ &= \frac{1}{1-\alpha}\int \left[-\boldsymbol{w}^\mathsf{T}\boldsymbol{r} - \mathrm{VaR}_\alpha(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r})\right]^+ p(\boldsymbol{r}) d\boldsymbol{r} + \mathrm{VaR}_\alpha(-\boldsymbol{w}^\mathsf{T}\boldsymbol{r}). \end{aligned}$$

Exercise 10.3: Sanity check for variational computation of CVaR

Generate 10 000 samples of the random variable ξ following a normal distribution and compute the CVaR as

$$\mathrm{CVaR}_\alpha = \mathbb{E}\left[\xi \mid \xi \geq \mathrm{VaR}_\alpha\right].$$

Verify numerically that the variational expression for the CVaR gives the same result:

$$\mathrm{CVaR}_\alpha = \inf_\tau\left\{\tau + \frac{1}{1-\alpha}\mathbb{E}\left[(\xi - \tau)^+\right]\right\}.$$

Solution

First, we generate the data:

```
set.seed(42)

n <- 10000
alpha <- 0.95
loss_samples <- rnorm(n, mean = 10)
```

```
# Direct empirical calculation of CVaR
VaR_alpha <- quantile(loss_samples, alpha)
CVaR_alpha <- mean(loss_samples[loss_samples >= VaR_emp])

# Variational calculation of CVaR
variational_CVaR <- function(tau) {
  excess <- pmax(loss_samples - tau, 0)
  tau + mean(excess) / (1 - alpha)
}

opt <- optimize(variational_CVaR, interval = range(loss_samples))
optimal_tau <- opt$minimum
CVaR_var_expr <- opt$objective
```

```
# Print results
cat("VaR_alpha:", VaR_alpha, "\n")
cat("Optimal tau:", optimal_tau, "\n")
cat("CVaR_alpha (conditional expectation):", CVaR_alpha, "\n")
cat("CVaR_alpha (variational expression):", CVaR_var_expr, "\n")
```

```
VaR_alpha: 11.64691
Optimal tau: 11.64763
CVaR_alpha (conditional expectation): 12.06624
CVaR_alpha (variational expression): 12.06624
```

Exercise 10.4: CVaR vs. downside risk

Consider the following two measures of risk in terms of the loss random variable ξ:

- downside risk in the form of lower partial moment (LPM) with $\alpha = 1$:

$$\mathrm{LPM}_1 = \mathbb{E}\left[(\xi - \xi_0)^+\right];$$

- CVaR:

$$\text{CVaR}_\alpha = \mathbb{E}\left[\xi \mid \xi \geq \text{VaR}_\alpha\right].$$

Rewrite LPM_1 in the form of CVaR_α and the other way around. Hint: use $\xi_0 = \text{VaR}_\alpha$.

Solution

First, note that LPM_1 can be expressed as

$$\begin{aligned}\text{LPM}_1 &= \mathbb{E}\left[(\xi - \xi_0)^+\right] \\ &= \mathbb{E}\left[(\xi - \xi_0) \times I\{\xi \geq \xi_0\}\right].\end{aligned}$$

Then, notice that the CVaR can be rewritten as

$$\begin{aligned}\text{CVaR}_\alpha &= \mathbb{E}\left[\xi \mid \xi \geq \text{VaR}_\alpha\right] \\ &= \frac{1}{1-\alpha}\mathbb{E}\left[\xi \times I\{\xi \geq \text{VaR}_\alpha\}\right] \\ &= \frac{1}{1-\alpha}\mathbb{E}\left[(\xi - \text{VaR}_\alpha) \times I\{\xi \geq \text{VaR}_\alpha\}\right] \\ &\quad + \frac{1}{1-\alpha}\mathbb{E}\left[\text{VaR}_\alpha \times I\{\xi \geq \text{VaR}_\alpha\}\right] \\ &= \frac{1}{1-\alpha}\mathbb{E}\left[(\xi - \text{VaR}_\alpha)^+\right] \\ &\quad + \frac{1}{1-\alpha}\text{VaR}_\alpha \times \mathbb{E}\left[I\{\xi \geq \text{VaR}_\alpha\}\right] \\ &= \frac{1}{1-\alpha}\mathbb{E}\left[(\xi - \text{VaR}_\alpha)^+\right] + \text{VaR}_\alpha.\end{aligned}$$

Thus, if we take $\xi_0 = \text{VaR}_\alpha$, we can make the following interesting connection:

$$\text{CVaR}_\alpha = \frac{1}{1-\alpha}\text{LPM}_1 + \text{VaR}_\alpha.$$

In words, under that choice, minimizing the CVaR is equivalent to minimizing the LPM with $\alpha = 1$.

Exercise 10.5: Log-sum-exp function as exponential cone

Show that the following convex constraint involving the perspective operator on the log-sum-exp function,

$$s \geq t \log\left(e^{x_1/t} + e^{x_2/t}\right),$$

for $t > 0$, can be rewritten in terms of the exponential cone $\mathcal{K}_{\text{exp}}$ as

$$\begin{aligned} t &\geq u_1 + u_2, \\ (x_i - s, t, u_i) &\in \mathcal{K}_{\text{exp}}, \qquad i = 1, 2, \end{aligned}$$

where

$$\mathcal{K}_{\text{exp}} \triangleq \left\{(a, b, c) \mid c \geq b\, e^{a/b}, b > 0\right\} \cup \left\{(a, b, c) \mid a \leq 0, b = 0, c \geq 0\right\}.$$

Solution

Starting with the constraint involving the log-sum-exp function and the perspective operator, we have:

$$\begin{aligned} s &\geq t \log\left(e^{x_1/t} + e^{x_2/t}\right) \\ e^{s/t} &\geq e^{x_1/t} + e^{x_2/t} \\ 1 &\geq e^{(x_1 - s)/t} + e^{(x_2 - s)/t} \\ t &\geq te^{(x_1 - s)/t} + te^{(x_2 - s)/t} \end{aligned}$$

Then, we introduce the dummy variables u_1 and u_2:

$$\begin{aligned} t &\geq u_1 + u_2 \\ u_i &\geq te^{(x_i - s)/t}, \qquad i = 1, 2. \end{aligned}$$

Finally, we can recognize the constraint $u_i \geq te^{(x_i - s)/t}$ as the exponential cone:

$$\begin{aligned} t &\geq u_1 + u_2 \\ (x_i - s, t, u_i) &\in \mathcal{K}_{\text{exp}}, \qquad i = 1, 2. \end{aligned}$$

Exercise 10.6: Drawdown and path-dependency

a. Generate 10 000 samples of returns following a normal distribution.
b. Compute and plot the cumulative returns, and plot the drawdown.
c. Randomly reorder the original returns and plot again.
d. Repeat a few times to observe the path-dependency property of the drawdown.

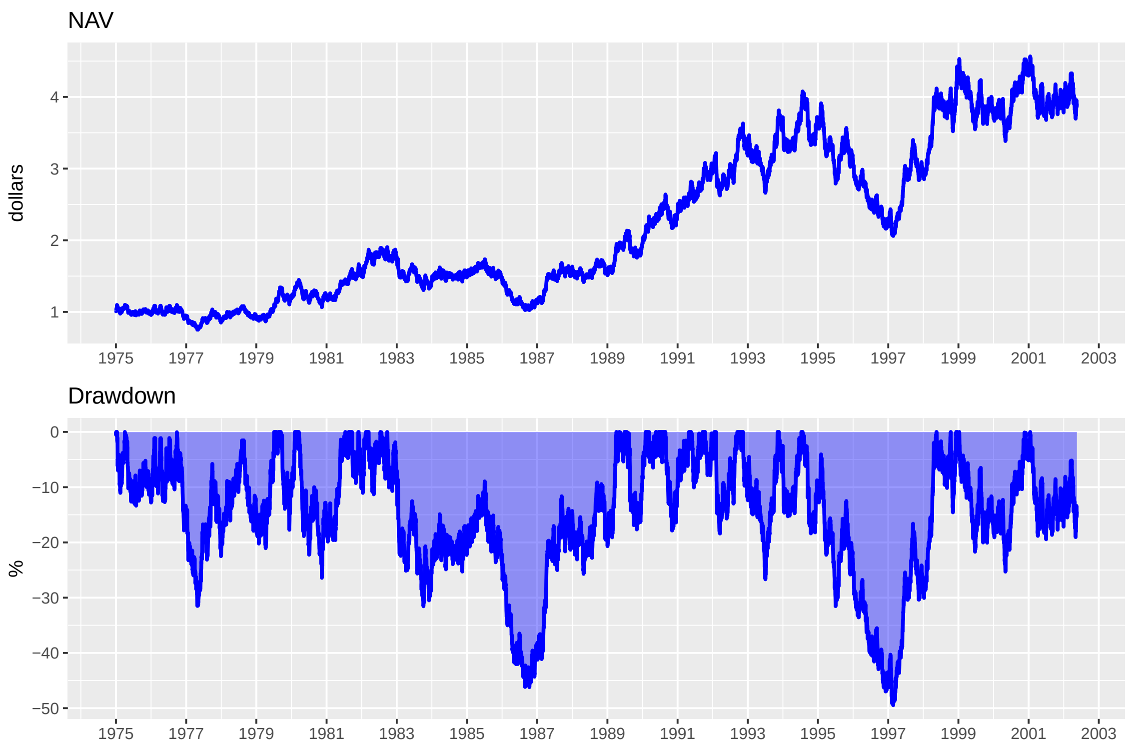
Solution

a. First, we generate Gaussian returns (using reasonable values for the mean and standard deviation as measured from real data):

```r
library(xts)

n <- 10000
set.seed(42)
return_samples <- rnorm(n, mean = 0.0003, sd = 0.01)
return_samples <- xts(return_samples,
                      order.by = seq(as.Date("1975-01-01"), length=n, by="days"))
```

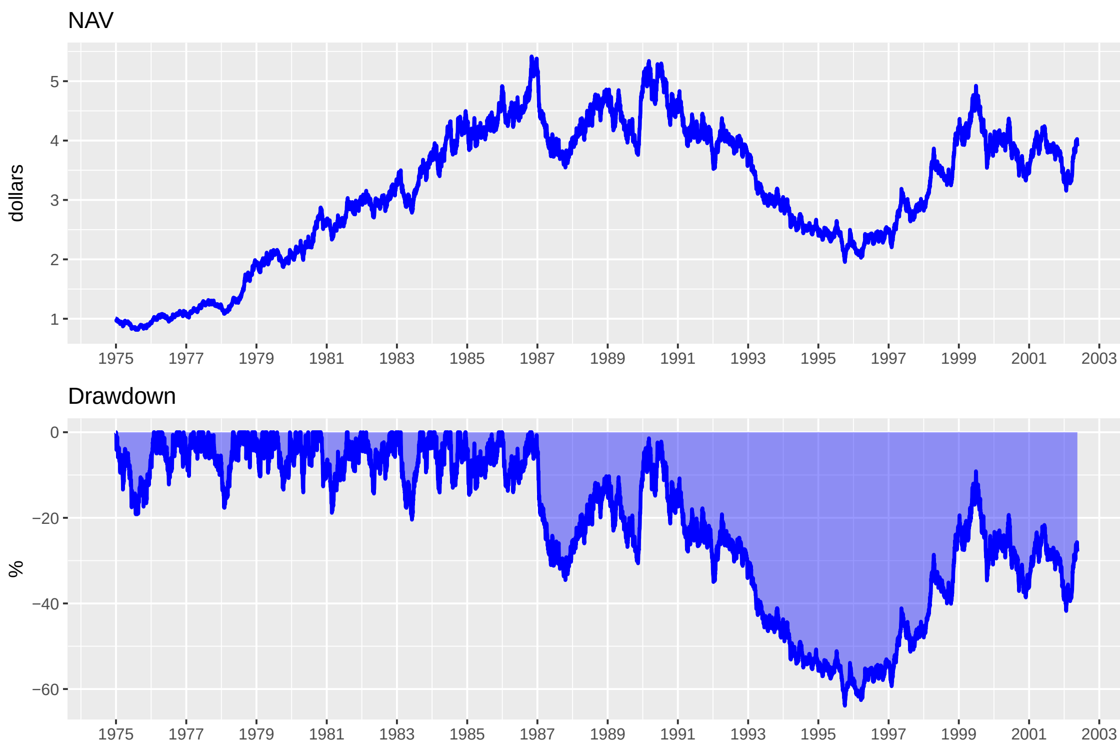
b. Compute and plot the cumulative returns and drawdown:

```r
library(ggplot2)
library(patchwork)
library(PerformanceAnalytics)

# Compute cumulative returns and drawdown
cumreturns <- cumprod(1 + return_samples)
drawdown <- Drawdowns(return_samples)

# Plots
p1 <- fortify(cumreturns, melt = TRUE) |>
  ggplot(aes(x = Index, y = Value)) +
  geom_line(linewidth = 1, color = "blue") +
  scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  labs(title = "NAV", x = NULL, y = "dollars")

p2 <- drawdown |>
  magrittr::multiply_by(100) |>
  fortify(melt = TRUE) |>
  ggplot(aes(x = Index, y = Value, color = Series, fill = Series, alpha = Series)) +
  geom_area(position = "identity", show.legend = FALSE, linewidth = 1, alpha = 0.4, na.rm = TRUE) +
  scale_color_manual(values = "blue") +
  scale_fill_manual(values = "blue") +
  scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  labs(title = "Drawdown", x = NULL, y = "%")

p1 / p2
```



c. Randomly reorder the original returns and plot again:

```r
shuffled_returns <- xts(sample(coredata(return_samples)),
                        order.by = index(return_samples))

# Compute cumulative returns and drawdown
cumreturns <- cumprod(1 + shuffled_returns)
drawdown <- Drawdowns(shuffled_returns)

# Plots
p1 <- fortify(cumreturns, melt = TRUE) |>
  ggplot(aes(x = Index, y = Value)) +
  geom_line(linewidth = 1, color = "blue") +
  scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  labs(title = "NAV", x = NULL, y = "dollars")

p2 <- drawdown |>
  magrittr::multiply_by(100) |>
  fortify(melt = TRUE) |>
  ggplot(aes(x = Index, y = Value, color = Series, fill = Series, alpha = Series)) +
  geom_area(position = "identity", show.legend = FALSE, linewidth = 1, alpha = 0.4, na.rm = TRUE) +
  scale_color_manual(values = "blue") +
  scale_fill_manual(values = "blue") +
  scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  labs(title = "Drawdown", x = NULL, y = "%")

p1 / p2
```



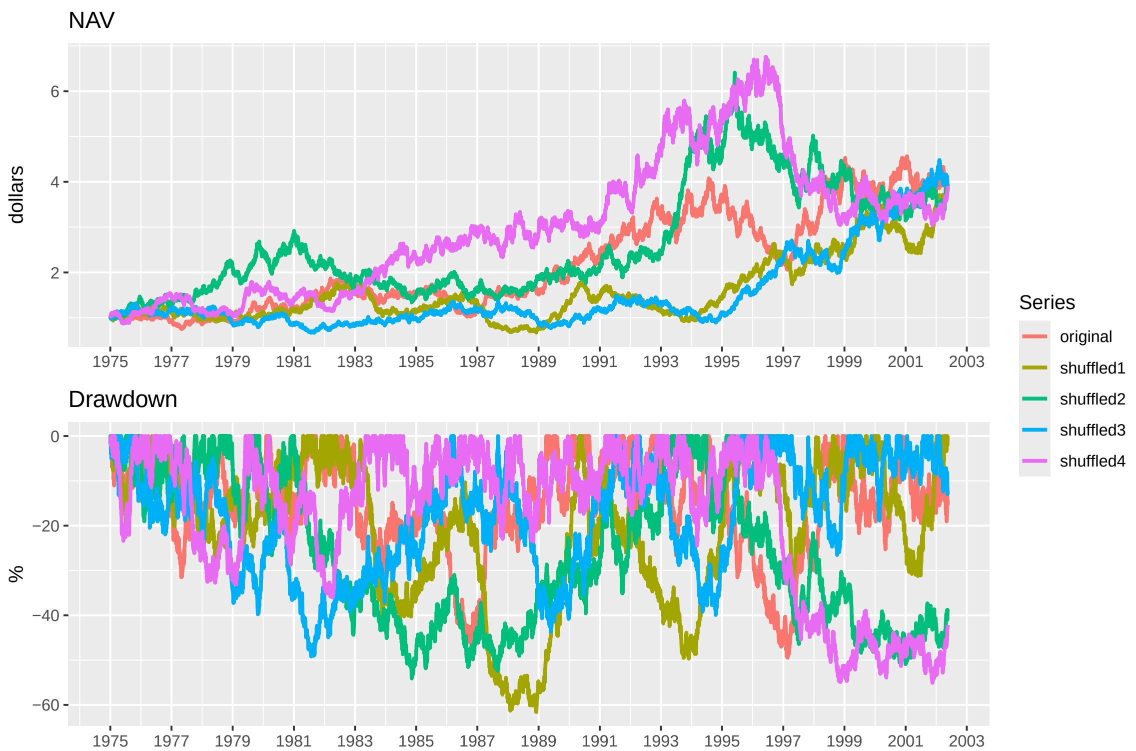
d. Repeat a few times to observe the path-dependency property of the drawdown:

```
# Number of shuffled columns to add
num_shuffles <- 4

# Add shuffled columns one by one
colnames(return_samples) <- "original"
for (i in 1:num_shuffles) {
  new_col <- xts(sample(coredata(return_samples[, "original"])),
                 order.by = index(return_samples))
  colnames(new_col) <- paste0("shuffled", i)
  return_samples <- merge(return_samples, new_col)
}

# Print the first few rows to verify
head(return_samples)
```

```
                original    shuffled1     shuffled2    shuffled3    shuffled4
1975-01-01  0.0140095845 -0.017356211 -0.0008209290  0.007966751  0.002872188
1975-01-02 -0.0053469817  0.006398790 -0.0042010535  0.004958857  0.006349859
1975-01-03  0.0039312841  0.017643203  0.0079261293  0.005162001  0.008825920
1975-01-04  0.0066286260  0.004347926  0.0210442591 -0.001731927  0.019284352
```

```
1975-01-05  0.0043426832  0.003387267  0.0089855462 -0.013221120  0.012063428
1975-01-06 -0.0007612452 -0.006461370 -0.0007582761  0.003697871 -0.018452176
```

```r
# Compute cumulative returns and drawdown
cumreturns <- cumprod(1 + return_samples)
drawdown <- Drawdowns(return_samples)

# Plots
p1 <- fortify(cumreturns, melt = TRUE) |>
  ggplot(aes(x = Index, y = Value, color = Series)) +
  geom_line(linewidth = 1) +
  scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  labs(title = "NAV", x = NULL, y = "dollars")

p2 <- drawdown |>
  magrittr::multiply_by(100) |>
  fortify(melt = TRUE) |>
  ggplot(aes(x = Index, y = Value, color = Series)) +
  geom_line(linewidth = 1) +
  scale_x_date(date_breaks = "2 year", date_labels = "%Y") +
  labs(title = "Drawdown", x = NULL, y = "%")

p1 / p2 + plot_layout(guides = "collect")
```



Indeed, each reshuffled version has a totally different cumulative return and drawdown.

Exercise 10.7: Semi-variance portfolios

a. Download market data corresponding to N assets (e.g., stocks or cryptocurrencies) during a period with T observations, $\boldsymbol{r}_1, \dots, \boldsymbol{r}_T \in \mathbb{R}^N$.
b. Solve the minimization of the semi-variance in a nonparametric way (reformulate it as a quadratic program):

$$\begin{array}{ll} \underset{\boldsymbol{w}}{\text{minimize}} & \frac{1}{T}\sum_{t=1}^{T}\left((\tau - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t)^+\right)^2 \\ \text{subject to} & \boldsymbol{w} \geq \boldsymbol{0}, \quad \boldsymbol{1}^\mathsf{T}\boldsymbol{w} = 1. \end{array}$$

c. Solve the parametric approximation based on the quadratic program:

$$\begin{array}{ll} \underset{\boldsymbol{w}}{\text{minimize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{M}\boldsymbol{w} \\ \text{subject to} & \boldsymbol{w} \geq \boldsymbol{0}, \quad \boldsymbol{1}^\mathsf{T}\boldsymbol{w} = 1, \end{array}$$

where

$$\boldsymbol{M} = \mathbb{E}\left[(\tau\mathbf{1} - \boldsymbol{r})^{+}\left((\tau\mathbf{1} - \boldsymbol{r})^{+}\right)^{\mathsf{T}}\right].$$

d. Comment on the goodness of the approximation.

Solution

a. Market data corresponding to N stocks:

```
library(xts)
library(pob)       # Market data used in the book

# use data from package pob
data(SP500_2015to2020)
N <- 20
set.seed(42)
prices <- SP500_2015to2020$stocks[, sample(ncol(SP500_2015to2020$stocks), N)]
X <- diff(log(prices))[-1]
T <- nrow(X)
```

b. Solve the minimization of the semi-variance in a nonparametric way (reformulate it as a quadratic program):

```
library(CVXR)

tau <- 0

# Direct formulation
w <- Variable(N)
prob <- Problem(Minimize(mean(pos(tau - as.matrix(X) %*% w)^2)),
                constraints = list(w >= 0, sum(w) == 1))
result <- solve(prob)
w_direct <- as.vector(result$getValue(w))
optval_direct <- mean(pmax(0, tau - as.matrix(X) %*% w_direct)^2)  #also: result$value
print(optval_direct)
```

```
[1] 5.07386e-05
```

Now, we do the same but reformulating the problem as a quadratic program:

$$\begin{array}{ll} \underset{\boldsymbol{w},\{s_t\}}{\text{minimize}} & \frac{1}{T}\sum_{t=1}^{T} s_t^2 \\ \text{subject to} & 0 \leq s_t \geq \tau - \boldsymbol{w}^{\mathsf{T}}\boldsymbol{r}_t, \qquad t = 1, \dots, T, \\ & \boldsymbol{w} \geq \mathbf{0}, \quad \mathbf{1}^{\mathsf{T}}\boldsymbol{w} = 1. \end{array}$$

```r
# Reformulating it as a QP:
w <- Variable(N)
s <- Variable(T)
prob <- Problem(Minimize(mean(s^2)),
                constraints = list(
                  s >= 0, s >= tau - as.matrix(X) %*% w,
                  w >= 0, sum(w) == 1
                  )
                )
result <- solve(prob)
w_qp <- as.vector(result$getValue(w))
optval_qp <- mean(pmax(0, tau - as.matrix(X) %*% w_qp)^2)  #also: result$value
print(optval_qp)
```

```
[1] 5.07386e-05
```

```r
# Sanity check
w_direct - w_qp
```

```
 [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

c. Solve the parametric approximation based on the quadratic program:

```r
# Compute approximated matrix M
Xpos <- pmax(tau - X, 0)
M <- cov(Xpos)

# Solve QP:
w <- Variable(N)
prob <- Problem(Minimize(quad_form(w, M)),  # or: w %*% M %*% w
                constraints = list(w >= 0, sum(w) == 1))
result <- solve(prob)
w_approx <- as.vector(result$getValue(w))
optval_approx <- mean(pmax(0, tau - as.matrix(X) %*% w_approx)^2)
print(optval_approx)
```

```
[1] 5.083168e-05
```

```r
# Sanity check
max(abs(w_direct - w_approx))
```

```
[1] 0.01912554
```

d. Comment on the goodness of the approximation:

We can see that the approximation only achieves an objective value of vs. the optimal value of .

Exercise 10.8: CVaR portfolios

a. Download market data corresponding to N assets (e.g., stocks or cryptocurrencies) during a period with T observations, $\boldsymbol{r}_1, \dots, \boldsymbol{r}_T \in \mathbb{R}^N$.
b. Solve the minimum CVaR portfolio as the following linear program for different values of the parameter α:

$$\begin{array}{ll} \underset{\boldsymbol{w},\tau,\boldsymbol{u}}{\text{minimize}} & \tau + \frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T} u_t \\ \text{subject to} & 0 \leq u_t \geq -\boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t - \tau, \qquad t = 1, \dots, T, \\ & \boldsymbol{w} \geq \boldsymbol{0}, \quad \boldsymbol{1}^\mathsf{T}\boldsymbol{w} = 1. \end{array}$$

c. Observe how many observations are actually used ($u_t > 0$) for the different values of α.
d. Add some small perturbation or noise to the sequence of returns $\boldsymbol{r}_1, \dots, \boldsymbol{r}_T$ and repeat the experiment to observe the sensitivity of the solutions to data perturbation.

Solution

a. Market data corresponding to N stocks:

```
library(xts)
library(pob)       # Market data used in the book

# use data from package pob
data(SP500_2015to2020)
N <- 20
set.seed(42)
prices <- SP500_2015to2020$stocks[, sample(ncol(SP500_2015to2020$stocks), N)]
X <- diff(log(prices))[-1]
T <- nrow(X)
```

b. Solve the minimum CVaR portfolio as a linear program:

```r
library(CVXR)

alpha <- 0.95
w <- Variable(N)
tau <- Variable(1)
u <- Variable(T)
prob <- Problem(Minimize( tau + mean(u)/(1-alpha) ),
                constraints = list(u >= 0, u >= -as.matrix(X) %*% w - tau,
                                   w >= 0, sum(w) == 1))
result <- solve(prob)
u_alpha_0_95 <- as.vector(result$getValue(u))
w_alpha_0_95 <- as.vector(result$getValue(w))


alpha <- 0.99
w <- Variable(N)
tau <- Variable(1)
u <- Variable(T)
prob <- Problem(Minimize( tau + mean(u)/(1-alpha) ),
                constraints = list(u >= 0, u >= -as.matrix(X) %*% w - tau,
                                   w >= 0, sum(w) == 1))
result <- solve(prob)
u_alpha_0_99 <- as.vector(result$getValue(u))
w_alpha_0_99 <- as.vector(result$getValue(w))
```

c. Observe how many observations are actually used ($u_t > 0$) for different values of α:

```r
num_tail_samples_alpha_0_95 <- sum(u_alpha_0_95 > 1e-5 * max(u_alpha_0_95))
num_tail_samples_alpha_0_99 <- sum(u_alpha_0_99 > 1e-5 * max(u_alpha_0_99))

# Print results
cat("Number of samples in the tail for alpha=0.95: ", num_tail_samples_alpha_0_95, "\n")
cat("Number of samples in the tail for alpha=0.99: ", num_tail_samples_alpha_0_99, "\n")
```

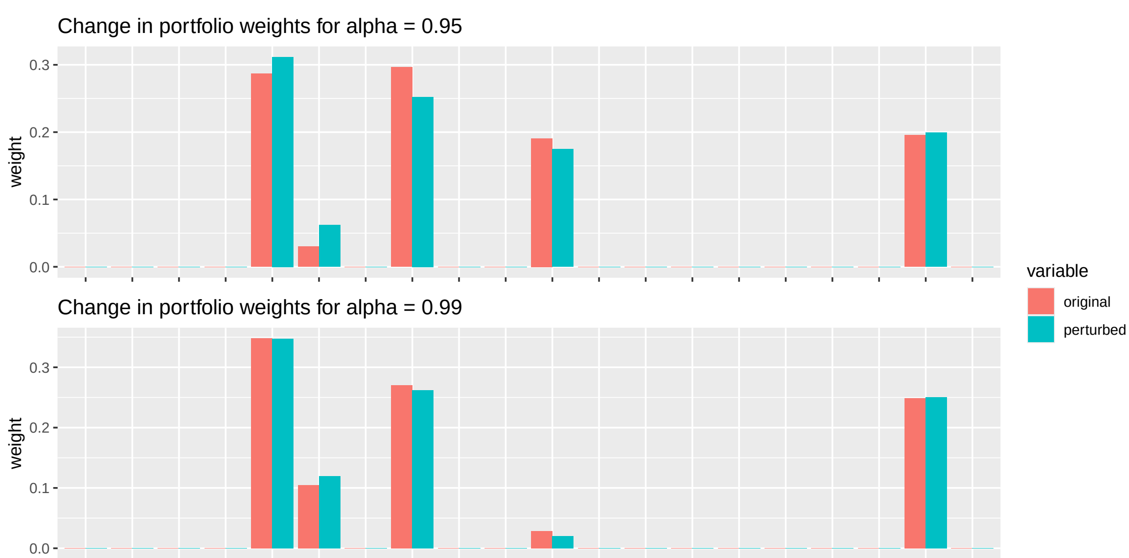
```
Number of samples in the tail for alpha=0.95:  76
Number of samples in the tail for alpha=0.99:  11
```

We can observe that for larger values of α, the number of observations falling in the tail is smaller, which are precisely the samples used in the CVaR computation.

d. Add some small perturbation or noise to the sequence of returns to observe the sensitivity of the solutions to data perturbation:

```r
# # Add overall noise
# noise_level <- 0.10 * sd(X)
# X_noisy <- X + matrix(rnorm(T*N, mean = 0, sd = noise_level),
#                       nrow = T, ncol = N)


# Add noise column by column
cols_sd <- apply(X, 2, sd)
noise_level <- 0.20 * cols_sd
noise_matrix <- matrix(0, nrow = T, ncol = N)
for (j in 1:N)
  noise_matrix[, j] <- rnorm(T, mean = 0, sd = noise_level[j])
X_noisy <- X + noise_matrix


# Solve the perturbed problems
alpha <- 0.95
w <- Variable(N)
tau <- Variable(1)
u <- Variable(T)
prob <- Problem(Minimize( tau + mean(u)/(1-alpha) ),
                constraints = list(u >= 0, u >= -as.matrix(X_noisy) %*% w - tau,
                                   w >= 0, sum(w) == 1))
result <- solve(prob, solver = "GLPK")
w_alpha_0_95_perturbed <- as.vector(result$getValue(w))


alpha <- 0.99
w <- Variable(N)
tau <- Variable(1)
u <- Variable(T)
prob <- Problem(Minimize( tau + mean(u)/(1-alpha) ),
                constraints = list(u >= 0, u >= -as.matrix(X_noisy) %*% w - tau,
                                   w >= 0, sum(w) == 1))
result <- solve(prob, solver = "GLPK")
w_alpha_0_99_perturbed <- as.vector(result$getValue(w))
```

```r
library(ggplot2)
library(reshape2)  # for melt function

# barplots
p_0_95 <- df_w <- data.frame(
  "stocks"    = names(X),
  "original"  = w_alpha_0_95,
  "perturbed" = w_alpha_0_95_perturbed
  ) |>
  melt(id.vars = "stocks") |>
  ggplot(aes(x = stocks, y = value, fill = variable)) +
  geom_bar(stat="identity", position = "dodge") +
  scale_x_discrete(labels = NULL) +
  labs(title = "Change in portfolio weights for alpha = 0.95", x = NULL, y = "weight")

p_0_99 <- df_w <- data.frame(
  "stocks"    = names(X),
  "original"  = w_alpha_0_99,
  "perturbed" = w_alpha_0_99_perturbed
  ) |>
  melt(id.vars = "stocks") |>
  ggplot(aes(x = stocks, y = value, fill = variable)) +
  geom_bar(stat="identity", position = "dodge") +
  scale_x_discrete(labels = NULL) +
  labs(title = "Change in portfolio weights for alpha = 0.99", x = NULL, y = "weight")

p_0_95 / p_0_99 + plot_layout(guides = "collect")
```



In fact, the portfolios do not seem to be too sensitive to the perturbations after all.

Exercise 10.9: Mean–Max-DD formulation as an LP

The mean–Max-DD formulation replaces the usual variance term $\boldsymbol{w}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{w}$ by the Max-DD as a measure of risk, defined as

$$\text{Max-DD}(\boldsymbol{w}) = \max_{1\le t\le T} D_t(\boldsymbol{w}),$$

where $D_t(\boldsymbol{w})$ is the drawdown at time t. This leads to the problem formulation

$$\begin{array}{ll}\underset{\boldsymbol{w}}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda \displaystyle\max_{1\le t\le T}\left\{\max_{1\le\tau\le t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}\right\}\\ \text{subject to} & \boldsymbol{w}\in\mathcal{W}.\end{array}$$

Show that it can be rewritten as the following problem ($u_0 \triangleq -\infty$):

$$\begin{array}{ll}\underset{\boldsymbol{w},\boldsymbol{u},s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s\\ \text{subject to} & \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} \le u_t \le s + \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}, \quad t = 1,\dots,T,\\ & u_{t-1}\le u_t,\\ & \boldsymbol{w}\in\mathcal{W},\end{array}$$

which is a linear program (assuming $\mathcal{W}$ only contains linear constraints).

Solution

Writing the problem in epigraph form leads to

$$\begin{array}{ll}\underset{\boldsymbol{w},s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s\\ \text{subject to} & s \ge \displaystyle\max_{1\le t\le T}\left\{\max_{1\le\tau\le t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}\right\}\\ & \boldsymbol{w}\in\mathcal{W}.\end{array}$$

The outer max operator can be easily removed by converting the constraint into T constraints as

$$\begin{array}{ll}\underset{\boldsymbol{w},s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s\\ \text{subject to} & s \ge \displaystyle\max_{1\le\tau\le t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}, \qquad t = 1,\dots,T\\ & \boldsymbol{w}\in\mathcal{W}.\end{array}$$

The remaining max operator can also be easily removed as

$$
\begin{array}{ll}
\underset{\boldsymbol{w},\boldsymbol{u},s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s\\
\text{subject to} & s \geq u_t - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}, \quad t = 1,\dots,T\\
& u_t \geq \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}}, \qquad t = 1,\dots,T,\; \tau = 1,\dots,t\\
& \boldsymbol{w} \in \mathcal{W}.
\end{array}
$$

Observe that there is a number of $(T+1)T/2$ constraints of the form $u_t \geq \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}}$. These can be reduced by recursion, including instead the constraints $u_t \geq u_{t-1}$:

$$
\begin{array}{lll}
\underset{\boldsymbol{w},\boldsymbol{u},s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s & \\
\text{subject to} & \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} \leq u_t \leq s + \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}, & t = 1,\dots,T\\
& u_{t-1} \leq u_t, & t = 1,\dots,T\\
& \boldsymbol{w} \in \mathcal{W}. &
\end{array}
$$

Exercise 10.10: Mean–Ave-DD formulation as an LP

The mean–Ave-DD formulation replaces the usual variance term $\boldsymbol{w}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{w}$ by the Ave-DD as a measure of risk, defined as

$$
\text{Ave-DD} = \frac{1}{T}\sum_{1\leq t\leq T} D_t(\boldsymbol{w}),
$$

where $D_t(\boldsymbol{w})$ is the drawdown at time t. This leads to the problem formulation

$$
\begin{array}{ll}
\underset{\boldsymbol{w}}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\,\dfrac{1}{T}\displaystyle\sum_{t=1}^{T}\left(\max_{1\leq\tau\leq t}\, \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}\right)\\
\text{subject to} & \boldsymbol{w} \in \mathcal{W}.
\end{array}
$$

Show that it can be rewritten as the following problem ($u_0 \triangleq -\infty$):

$$
\begin{array}{ll}
\underset{\boldsymbol{w},\boldsymbol{u},s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s\\
\text{subject to} & \frac{1}{T}\sum_{t=1}^{T} u_t \leq \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} + s,\\
& \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} \leq u_t, \qquad t = 1,\dots,T,\\
& u_{t-1} \leq u_t,\\
& \boldsymbol{w} \in \mathcal{W},
\end{array}
$$

which is a linear program (assuming $\mathcal{W}$ only contains linear constraints).

Solution

Writing the problem in epigraph form leads to

$$\begin{array}{ll} \underset{\boldsymbol{w},s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s \\ \text{subject to} & s \geq \dfrac{1}{T}\displaystyle\sum_{t=1}^{T}\left\{\max_{1\leq\tau\leq t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}\right\} \\ & \boldsymbol{w} \in \mathcal{W}, \end{array}$$

and the max operator can be removed as

$$\begin{array}{ll} \underset{\boldsymbol{w},\boldsymbol{u},s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s \\ \text{subject to} & s \geq \frac{1}{T}\sum_{t=1}^{T}\left\{u_t - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}\right\} \\ & u_t \geq \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}}, \qquad t = 1,\dots,T,\; \tau = 1,\dots,t \\ & \boldsymbol{w} \in \mathcal{W}. \end{array}$$

Observe that there is a number of $(T+1)T/2$ constraints of the form $u_t \geq \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}}$. These can be reduced by recursion, including instead the constraints $u_t \geq u_{t-1}$ ($u_0 \triangleq -\infty$):

$$\begin{array}{ll} \underset{\boldsymbol{w},\boldsymbol{u},s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s \\ \text{subject to} & \frac{1}{T}\sum_{t=1}^{T} u_t \leq \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} + s \\ & \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} \leq u_t, \qquad t = 1,\dots,T \\ & u_{t-1} \leq u_t \\ & \boldsymbol{w} \in \mathcal{W}. \end{array}$$

Exercise 10.11: Mean–CVaR-DD formulation as an LP

The mean–CVaR-DD formulation replaces the usual variance term $\boldsymbol{w}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{w}$ by the CVaR-DD as a measure of risk, expressed in a variational form as

$$\text{CVaR-DD}(\boldsymbol{w}) = \inf_{\tau}\left\{\tau + \frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T}(D_t(\boldsymbol{w}) - \tau)^+\right\},$$

where $D_t(\boldsymbol{w})$ is the drawdown at time t. This leads to the problem formulation

$$\begin{array}{ll} \underset{\boldsymbol{w},\tau}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\left(\tau + \dfrac{1}{1-\alpha}\dfrac{1}{T}\displaystyle\sum_{t=1}^{T}\left(\max_{1\leq\tau'\leq t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_{\tau'}^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} - \tau\right)^+\right) \\ \text{subject to} & \boldsymbol{w} \in \mathcal{W}. \end{array}$$

Show that it can be rewritten as the following problem ($u_0 \triangleq -\infty$):

$$
\begin{array}{ll}
\underset{\boldsymbol{w},\tau,s,\boldsymbol{z},\boldsymbol{u}}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s \\
\text{subject to} & s \geq \tau + \frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T} z_t, \\
& 0 \leq z_t \geq u_t - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} - \tau, \qquad t = 1,\ldots,T, \\
& \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} \leq u_t, \\
& u_{t-1} \leq u_t, \\
& \boldsymbol{w} \in \mathcal{W},
\end{array}
$$

which is a linear program (assuming $\mathcal{W}$ only contains linear constraints).

Solution

Writing the problem in epigraph form leads to

$$
\begin{array}{ll}
\underset{\boldsymbol{w},\tau,s}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s \\
\text{subject to} & s \geq \tau + \dfrac{1}{1-\alpha}\dfrac{1}{T}\displaystyle\sum_{t=1}^{T}\left(\max_{1\leq\tau'\leq t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_{\tau'}^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} - \tau\right)^{+} \\
& \boldsymbol{w} \in \mathcal{W}.
\end{array}
$$

The $(\cdot)^+$ operator can be removed as

$$
\begin{array}{ll}
\underset{\boldsymbol{w},\tau,s,\boldsymbol{z}}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s \\
\text{subject to} & s \geq \tau + \frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T} z_t \\
& 0 \leq z_t \geq \max\limits_{1\leq\tau'\leq t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_{\tau'}^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} - \tau \\
& \boldsymbol{w} \in \mathcal{W}.
\end{array}
$$

Finally, the max operator can be removed as

$$
\begin{array}{ll}
\underset{\boldsymbol{w},\tau,s,\boldsymbol{z},\boldsymbol{u}}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s \\
\text{subject to} & s \geq \tau + \frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T} z_t \\
& 0 \leq z_t \geq u_t - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} - \tau \\
& u_t \geq \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_{\tau'}^{\text{cum}}, \qquad t = 1,\ldots,T,\ \tau' = 1,\ldots,t \\
& \boldsymbol{w} \in \mathcal{W}.
\end{array}
$$

Observe that there is a number of $(T+1)T/2$ constraints of the form $u_t \geq \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_{\tau'}^{\text{cum}}$. This can be reduced by including instead the constraint $u_t \geq u_{t-1}$ ($u_0 \triangleq -\infty$):

$$
\begin{array}{ll}
\underset{\boldsymbol{w},\tau,s,\boldsymbol{z},\boldsymbol{u}}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s \\
\text{subject to} & s \geq \tau + \frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T} z_t \\
& 0 \leq z_t \geq u_t - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} - \tau, \qquad t = 1,\ldots,T \\
& \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} \leq u_t \\
& u_{t-1} \leq u_t \\
& \boldsymbol{w} \in \mathcal{W}.
\end{array}
$$

Exercise 10.12: Mean–EVaR-DD formulation as a convex problem

The mean–EVaR-DD formulation replaces the usual variance term $\boldsymbol{w}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{w}$ by the EVaR-DD as a measure of risk, defined as

$$\text{EVaR-DD}(\boldsymbol{w}) = \inf_{z>0}\left\{z^{-1}\log\left(\frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T}\exp(zD_t(\boldsymbol{w}))\right)\right\},$$

where $D_t(\boldsymbol{w})$ is the drawdown at time t defined as

$$D_t(\boldsymbol{w}) = \max_{1\leq\tau\leq t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}^{\text{cum}}_\tau - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}^{\text{cum}}_t.$$

a. Write down the mean–EVaR-DD portfolio formulation in convex form.
b. Further rewrite the problem in terms of the exponential cone.

Solution

a. The mean–EVaR-DD portfolio formulation is

$$\begin{array}{ll}\underset{\boldsymbol{w},\tau}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\,\text{EVaR-DD}(\boldsymbol{w})\\ \text{subject to} & \boldsymbol{w}\in\mathcal{W},\end{array}$$

which can be written (using $t = z^{-1}$) as

$$\begin{array}{ll}\underset{\boldsymbol{w},t>0}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\left(t\log\left(\frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T}\exp(D_t(\boldsymbol{w})/t)\right)\right)\\ \text{subject to} & \boldsymbol{w}\in\mathcal{W}.\end{array}$$

and, more explicitly using $D_t(\boldsymbol{w}) = \max_{1\leq\tau\leq t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}^{\text{cum}}_\tau - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}^{\text{cum}}_t$, as

$$\begin{array}{ll}\underset{\boldsymbol{w},z_t,t>0}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\left(t\log\left(\frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T}\exp(z_t/t)\right)\right)\\ \text{subject to} & z_t \geq \max_{1\leq\tau\leq t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}^{\text{cum}}_\tau - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}^{\text{cum}}_t\\ & \boldsymbol{w}\in\mathcal{W},\end{array}$$

which is a convex problem (assuming that the set $\mathcal{W}$ is convex) because the log-sum-exp function is convex and the perspective $tf(\boldsymbol{x}/t)$ of a function $f(\boldsymbol{x})$ preserves convexity.

b. We can further rewrite the problem in terms of the exponential cone. First rewrite the problem in epigraph form:

$$\begin{array}{ll}\underset{\boldsymbol{w},s,z_t,t>0}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\,s\\ \text{subject to} & s \geq t\log\left(\sum_{t=1}^{T}\exp(z_t/t)\right) - t\log\left((1-\alpha)T\right)\\ & z_t \geq \max_{1\leq\tau\leq t} \boldsymbol{w}^\mathsf{T}\boldsymbol{r}^{\text{cum}}_\tau - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}^{\text{cum}}_t\\ & \boldsymbol{w}\in\mathcal{W}\end{array}$$

or

$$\begin{array}{ll}\underset{\boldsymbol{w},s,z_t,t>0}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s\\ \text{subject to} & s + t\,\log\left((1-\alpha)T\right) \geq t\,\log\left(\sum_{t=1}^T \exp(z_t/t)\right)\\ & z_t \geq \underset{1\leq\tau\leq t}{\max}\; \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}}\\ & \boldsymbol{w}\in\mathcal{W}.\end{array}$$

At this point, we can recognize the constraint of the form

$$\tilde{s} \geq t\,\log\left(e^{z_1/t} + e^{z_2/t} + \dots\right),$$

which can be rewritten (see Exercise 10.5) as

$$\begin{array}{c} t \geq u_1 + u_2 + \dots,\\ (z_i - \tilde{s}, t, u_i) \in \mathcal{K}_{\text{exp}}, \qquad i = 1, 2, \dots \end{array}$$

Finally, putting everything together leads to

$$\begin{array}{lll}\underset{\boldsymbol{w},s,u_t,z_t,t>0}{\text{maximize}} & \boldsymbol{w}^\mathsf{T}\boldsymbol{\mu} - \lambda\, s & \\ \text{subject to} & t \geq u_1 + u_2 + \cdots + u_T & \\ & (z_i - s - t\,\log\left((1-\alpha)T\right), t, u_i) \in \mathcal{K}_{\text{exp}}, & i = 1, 2, \dots, T\\ & z_t \geq \underset{1\leq\tau\leq t}{\max}\; \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_\tau^{\text{cum}} - \boldsymbol{w}^\mathsf{T}\boldsymbol{r}_t^{\text{cum}} & \\ & \boldsymbol{w}\in\mathcal{W}. & \end{array}$$