Solutions to selected mandatory assignment

Advanced Empirical Finance

Transaction cost robust portfolio optimization (inspired by Hautsch et al (2019), available in Absalon)

For the empirical analysis, use the file `data_mandatory_assignment_3.RData` (available on Absalon) which can be imported with the command `load()`. Loading the file provides you with two tibbles in your environment: `returns` and `amihud_measures`.

```
load("data_mandatory_assignment_3.RData")

returns <- returns %>%
  select(-date) %>%
  mutate_all(~./100) %>%
  as.matrix() %>%
  na.omit()

N <- ncol(returns)
```

Data description: The object `returns` contains $T = 500$ daily log returns *in percent* for $N = 40$ assets. I extracted the returns from the daily CRSP file and cleaned the data already. The tibble `amihud_measures` contains a measure for illiquidity of the individual assets. For each ticker, the column `illiquidity` provides you with the (log of) full sample Amihud measure. The measure is based on Amihud (2002) and is popular in the literature because it only requires data about returns and volume to compute. For a detailed description of the Amihud measure consult the textbook *Empirical Asset Pricing and the cross-section of stock returns* in Absalon, Chapter 13. All you need to know for this assignment is that larger values in the column `illiquidity` indicate stocks that are costlier to trade than tickers with lower values.[1]

Exercise 1. (30%)

Consider the portfolio choice problem for transaction-cost adjusted certainty equivalent maximization with risk aversion parameter γ

$$\omega_{t+1}^* := \arg \max_{\omega \in \mathbb{R}^N, \iota'\omega = 1} \omega'\mu - \nu_t(\omega, \beta) - \frac{\gamma}{2}\omega'\Sigma\omega$$

where Σ and μ are (estimators of) the variance-covariance matrix of the returns and the vector of expected returns. Assume for now that transaction costs are quadratic in rebalancing **and** proportional to stock illiquidity such that

$$\nu_t(\omega, \beta) := \frac{\beta}{2}(\omega - \omega_{t^+})' B (\omega - \omega_{t^+})$$

where $B = \text{diag}(ill_1, \ldots, ill_N)$ is a diagonal matrix where $ill_1, \ldots, ill_N$ correspond to the Amihud measures provided to you. $\beta \in \mathbb{R}_+$ is a cost parameter and $\omega_{t^+} := \omega_t \circ (1 + r_t)/\sum_{i=1}^N(\omega_{i,t}(1 + r_{i,t}))$ is the weight vector before rebalancing. $\circ$ denotes element-wise multiplication. Note that r_{t+1} are the log returns $\log(P_{t+1}/P_t)$ and the data in `returns` is in percent!

Problem: Derive a closed-form solution for the mean-variance efficient portfolio ω_{t+1}^* based on the transaction cost specification above. Discuss the effect of illiquidity ill_i on the individual portfolio weights relative to an investor that myopically ignores transaction costs in her decision.

[1] The source code used to create the files, including the Amihud measure is available in Absalon.

Solution: Substituting yields

$$\begin{aligned}\omega_{t+1}^* &:= \arg\max_{\omega\in\mathbb{R}^N, \iota'\omega=1} \omega'\mu - \frac{\beta}{2}(\omega-\omega_{t^+})' B (\omega-\omega_{t^+}) - \frac{\gamma}{2}\omega'\Sigma\omega \\ &= \arg\max_{\omega\in\mathbb{R}^N, \iota'\omega=1} \omega'(\mu+\beta B\omega_{t^+}) - \frac{\gamma}{2}\omega'\left(\Sigma+\frac{\beta}{\gamma}B\right)\omega.\end{aligned}$$

The resulting closed form solution for the efficient portfolio has been derived in the lecture slides (Part 3, slide 24) where the parameters μ and Σ are replaced with $\mu+\beta B\omega_{t^+}$ and $\tilde{\Sigma} := \Sigma + \frac{\beta}{\gamma}B$, respectively, such that we get

$$\omega_{t+1}^* = \frac{1}{\gamma}\left(\tilde{\Sigma}^{-1} - \frac{1}{\iota'\tilde{\Sigma}^{-1}\iota}\tilde{\Sigma}^{-1}\iota\iota'\tilde{\Sigma}^{-1}\right)(\mu+\beta B\omega_{t^+}) + \frac{1}{\iota'\tilde{\Sigma}^{-1}\iota}\tilde{\Sigma}^{-1}\iota.$$

Intuitively, illiquid assets (high values of ill_i) are costlier to rebalance. In fact, the optimal weights reflect the costs by implictly reinstalling mean-variance optimization with the adjusted expected return vector $(\mu+\beta B\omega_{t^+})$ which is high for illiquid assets with high current holdings. Thus, illiquid assets are treated as if they yield a more attractive risk-return trade-off if the current exposure ω_{t^+} is already high and vice versa. As shown in the lecture slides (Part 3, slide 31), the optimal readjustment converges towards ω_{t^+} as β or ill_i approaches infinity.

Problem: Write a function that computes the optimal weights ω_{t+1}^* based on the inputs $\mu, \Sigma, \gamma, \beta, ill_1, \ldots, ill_N$ and the current weights ω_{t^+}. You can assume $\gamma = 4$ throughout the entire assignment.

```
optimal_tc_weight_l2 <- function(w_prev,
                                 mu,
                                 Sigma,
                                 beta = rep(0, ncol(Sigma)),
                                 gamma = 4){

  N <- ncol(Sigma)
  if(length(beta) == 1) beta <- rep(beta, N)
  B <- diag(beta)
  iota <- rep(1, N)
  Sigma_proc <- Sigma + 1 / gamma * B
  mu_proc <- mu + B %*% w_prev

  Sigma_inv <- solve(Sigma_proc)

  w_mvp <- Sigma_inv %*% iota
  w_mvp <- w_mvp / sum(w_mvp)
  w_opt <- w_mvp  + 1/gamma * (Sigma_inv - w_mvp %*% t(iota) %*% Sigma_inv) %*% mu_proc
  return(w_opt)
}
```

Exercise 2. (15%)

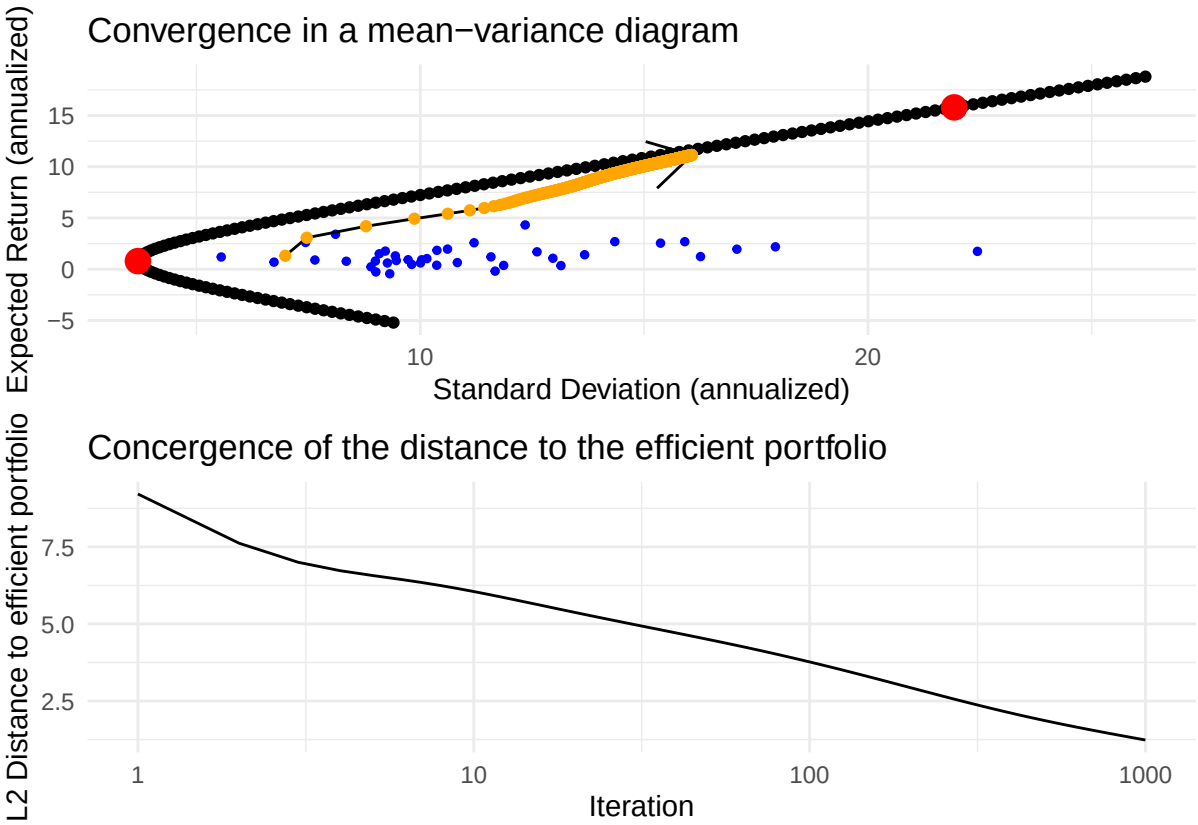
Problem Illustrate the convergence of the portfolio towards the efficient portfolio in the spirit of Proposition 4 in Hautsch et al. (2019): Use the full sample to estimate μ and Σ (you are free regarding the method to estimate μ and Σ but state your estimator and model assumptions in the report) and assume that the initial allocation, ω_0, is the naive portfolio $\frac{1}{N}\iota$. Then, compute iteratively the rebalanced portfolio weights ω_{t+1}^* which takes ω_t^* as input (thus you can ignore the distorting effect of returns for now and assume $\omega_{t^+} = \omega_t$).

Find a meaningful illustration of the weight dynamics towards the efficient portfolio and use the illustration to discuss the effect of illiquidity on the convergence towards the the efficient portfolio.

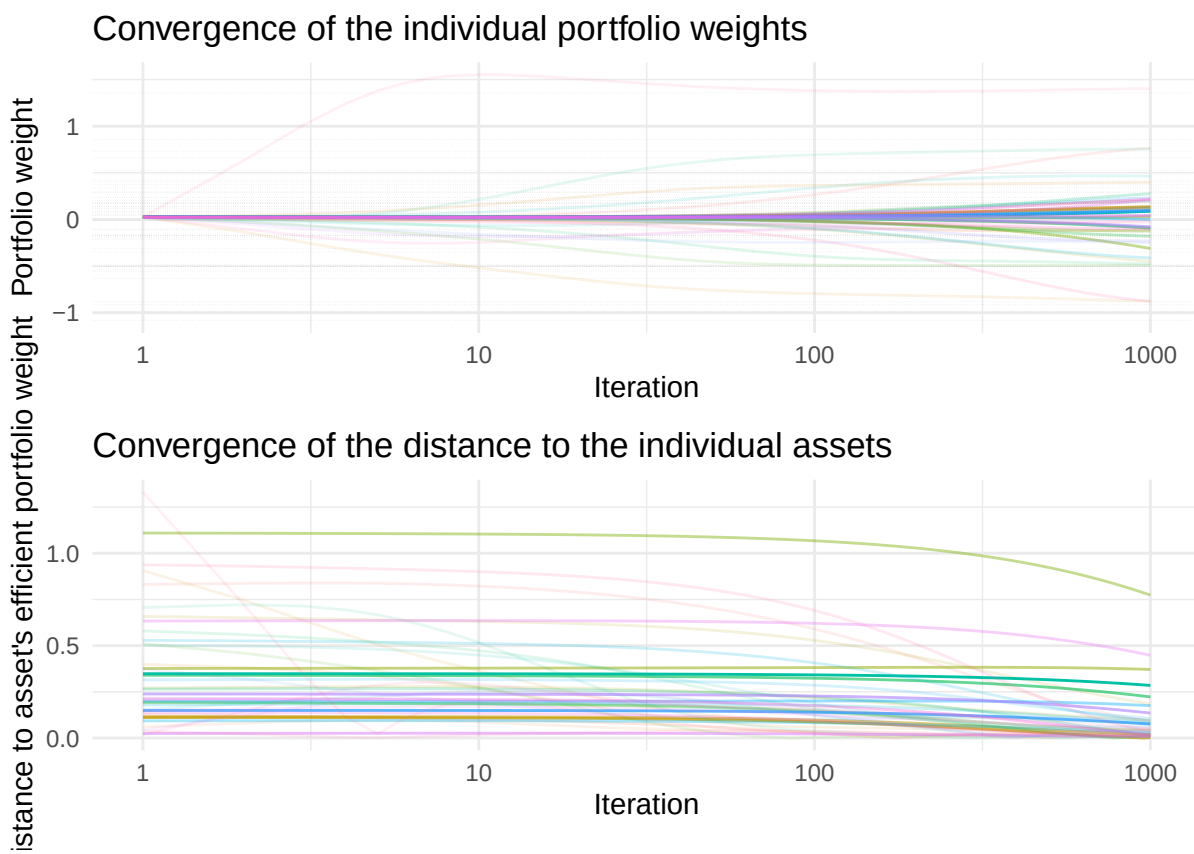
Solution: There are multiple feasible ways to illustrate the main feature of transaction cost penalization. The code below rests on using the sample moments but other approaches such as Ledoit Wolf shrinkage or a factor model are accepted as well.

```
# Mean-variance efficient portfolio
w_opt <- optimal_tc_weight_l2(w_prev = rep(1/N, ncol(returns)),
                              mu = colMeans(returns),
                              Sigma = cov(returns),
                              beta = 0 * amihud_measures$illiquidity,
                              gamma = 4)

weights <- matrix(NA, nc = ncol(returns), nr = 1000)
weights[1, ] <- rep(1/N, ncol(returns))
for(i in 2:nrow(weights)){
  weights[i, ] <- optimal_tc_weight_l2(w_prev = weights[i-1, ],
                                       mu = colMeans(returns),
                                       Sigma = cov(returns),
                                       beta = 2 * amihud_measures$illiquidity,
                                       gamma = 4)
}

colnames(weights) <- colnames(returns)
```

The following two figures do illustrate the convergence towards the efficient portfolio but do **not** allow to investigate asset specific illiquidity.



The following two illustrations illustrate that more illiquid assets (identified by a higher value ill_i converge slower to the efficient portfolio.



Exercise 3. (35%)

Problem: Implement a *function* that takes a value for the transaction cost parameter β as input and returns the out-of-sample Sharpe ratio of a corresponding backtest with a rolling window of length 250 days. Assume the initial allocation ω_0 is the naive portfolio. To be more precise: The function should recompute the optimal portfolios each day by taking the last 250 days of data into account to compute sample estimates $\hat{\mu}$ and $\hat{\Sigma}$.[2] The optimal allocation according to the optimization problem above should take the current allocation ω_{t+} into account and depends on the illiquidity measures. For each day, compute the realized portfolio return $\omega_{t+1}^{*\prime} r_{t+1}$ and subtract the realized transaction costs $\nu_t(\omega_{t+1}^*, \beta)$. The out-of-sample (annualized) Sharpe ratio is then computed as $\sqrt{250}\frac{\hat{\mu}^{pf}}{\hat{\sigma}^{pf}}$ where $\hat{\mu}^{pf}$ and $\hat{\sigma}^{pf}$ are the sample out-of-sample average return and standard deviation, respectively (thus you can assume a risk-free rate of 0 for the computation of the Sharpe ratio).

```
window_length <- 250
periods <- nrow(returns) - window_length # total number of out-of-sample periods

compute_sr <- function(beta = 0,
                       beta_suppose = NA_real_){

beta_suppose <- if_else(is.na(beta_suppose), beta, beta_suppose)
oos_values <- matrix(as.double(NA),
                     nrow = periods,
                     ncol = 1) # A matrix to collect all returns
```

[2] You can consider alternative estimators for $\hat{\mu}$ and $\hat{\Sigma}$. If you do so, explain your approach in the report.

```r
w_prev_1 <- rep(1/N ,N)

for(i in 1:periods){ # Rolling window

  # Extract information
  return_window <- returns[i : (i + window_length - 1),]

  # Sample moments
  Sigma <- cov(return_window)
  mu <- 0 * colMeans(return_window)

  # Optimal TC robust portfolio
  w_1 <- optimal_tc_weight_l2(w_prev = w_prev_1,
                              mu = mu,
                              Sigma = Sigma,
                              beta = beta_suppose * amihud_measures$illiquidity,
                              gamma = 4)
  # Evaluation
  raw_return <- 1 + returns[i + window_length, ] %*% w_1
  turnover <- t(w_1 - w_prev_1) %*% diag(amihud_measures$illiquidity) %*% (w_1 - w_prev_1)
  # Store realized returns
  net_return <- raw_return - beta * turnover
  oos_values[i, 1] <- c(net_return)
  #Computes adjusted weights based on the weights and next period returns
  w_prev_1 <- w_1 * as.vector(1 + returns[i + window_length, ])
  w_prev_1 <- w_prev_1 / sum(as.vector(w_prev_1))

}
sr <- apply(oos_values, 2, function(.) sqrt(250) * mean(. - 1 ) / sd(. - 1)) %>%
  as_tibble()
return(sr)
}
```

Problem: Illustrate the effect of different values of the transaction cost parameter β on the out-of-sample Sharpe ratio. β should take values between 0 to 100. Discuss your results.

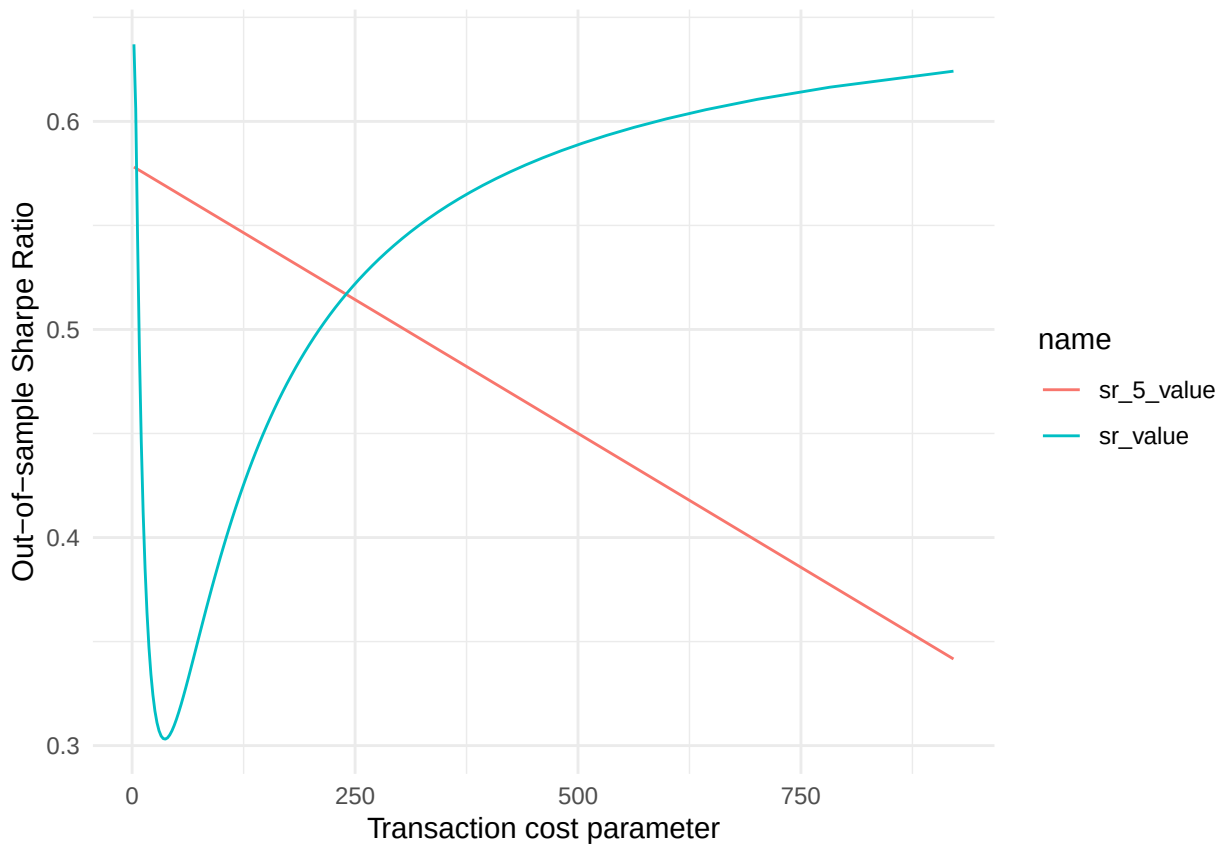
Solution: The figure below illustrate the effect of different transaction cost parameters β on the out-of-sample Sharpe ratio. Similar to Figure 1 in Hautsch et al (2019), a higher penalization of transaction costs can lead to an *increase* in performance due to the implied shrinkage effect which helps to mitigate estimation uncertainty. The figure illustrates at first a decrease in performance (as one would expect if there would be no effect on estimation error but only on the penalization of turnover) and then a sharp increase again.

Problem: Is it always optimal to choose the same β in the optimization problem than the β used in evaluating the portfolio performance? In other words: Can it be optimal to choose theoretically sub-optimal portfolios based on transaction costs considerations that do not reflect the actual incurred costs? Provide a brief discussion.

Solution The red line in the figure below illustrates the out-of-sample performance for a supposed β of 5, irrespective of the actual applied turnover penalization which affects the out-of-sample Sharpe ratio. In

these scenarios, the penalty parameter implies a theoretical sub-optimal portfolio which, however, does not necessarily imply a decrease in out of sample performance. Instead, the figure illustrates that for higher values of β until about 130 the resulting out-of-sample performance would have been better than taking the theoretically optimal value of the penalty factor β. These findings are in line with Hautsch et al (2019).

```
beta_effect <- tibble(beta = 200 * qexp((1:99/100))) %>%
    mutate(sr = map(beta, compute_sr),
           sr_5 = map(beta, compute_sr, beta_suppose = 5)) %>%
  unnest(c(sr, sr_5), names_sep = "_") %>%
  pivot_longer(-beta)

beta_effect %>%
  mutate(value = if_else(value < 0, NA_real_, value)) %>%
  ggplot(aes(x = beta, y = value, color = name)) +
  geom_line() +
  labs(x = "Transaction cost parameter",
       y = "Out-of-sample Sharpe Ratio")
```



Exercise 4. (20%)

Problem: Implement a full-fledged portfolio backtesting strategy with proportional L_1 transaction costs and suppose that $\beta = 50/10000 = 50bp$ as in Equation (12) of Hautsch et al. (2019) such that

$$v(\omega_{t+1}, \omega_{t^+}, \beta) = \beta||\omega_{t+1} - \omega_{t^+}||_1 = \beta \sum_{i=1}^{N} |\omega_{i,t+1} - \omega_{i,t^+}|.$$

Your aim is to generate a strong out-of-sample portfolio performance after adjusting for the proportional L_1 transaction costs. In order to improve the portfolio performance, you are free to compute parameter estimates either based on Ledoit-Wolf linear shrinkage towards the equicorrelation matrix or other approaches such as a rolling window multivariate DCC Garch or a factor model volatility model. Briefly describe your chosen model, the forecasting procedure and discuss the effects on the out-of-sample portfolio performance.

Irrespective of the way you estimate μ_t and Σ_t, the simulation should compare the performance of the following strategies: a. The naive portfolio which rebalances daily to $\omega = \frac{1}{N}\iota$ b. A portfolio which computes the theoretical optimal portfolio weights with optimal ex-ante adjustment for transaction costs in the spirit of Hautsch et al. (2019), e.g. that implements

$$\omega_{t+1}^* := \arg \max_{\omega \in \mathbb{R}^N, \iota'\omega = 1} \omega'\mu - \beta||\omega_{t+1} - \omega_{t+}||_1 - \frac{\gamma}{2}\omega'\Sigma\omega$$

c. A mean-variance efficient portfolio with a no short-selling constraint in the spirit of Jagannathan and Ma (2003)

To evaluate the performance of each strategy, the out-of-sample portfolio returns net of transaction costs should be aggregated into the out-of-sample Sharpe-Ratio and the average Turnover (Equations 44, and 46 in Hautsch et al., 2019).

Discuss the results: Which portfolio strategy performed best after adjusting for transaction costs and what are possible reasons for the differences in performance?
Note for the optimization: For the specific case with L_1 transaction costs, no closed form solution is available anymore. I recommend to use the package `alabama` for nonlinear constrained optimization (consult `?alabama::constrOptim.nl`). —

Solution: See code and results below. Depending on the model specification (mean and variance estimator) the results and the order can change. In the setup below, the naive portfolio outperforms all competitors which hints at substantial estimation error with 40 assets. It is remarkable that the simple mean-variance approach fails entirely and only after imposing additional constraints, e.g. no short-selling or turnover penalization is do we detect any meaningful economic performance. A good discussion should include the estimator of the moments of the return distribution and the resulting 1-step ahead forecast specification. The discussion of the portfolio performance should focus on the trade-off between the flexibility of the model and estimation uncertainty which implies potentially unstable portfolio weights and thus sub-optimal out-of-sample performance.

```
optimal_tc_weight_l1 <- function(w_prev,
                                 mu,
                                 Sigma,
                                 beta = 50 / 10000,
                                 gamma = 4){

fn <- function(w){
  obj <- w%*%as.vector(mu) -
    as.numeric(beta) * sum(abs(w - w_prev)) -
    gamma/2 * t(w) %*% Sigma %*% w
  return(-obj)}

out <- alabama::constrOptim.nl(par = w_prev,
                       fn = fn,
                       heq = function(w) return(sum(w) - 1),
                       control.outer =list(trace = FALSE))

return(as.numeric(out$par))
}
```

```r
optimal_no_short_sale_weight <- function(mu,
                                         Sigma,
                                         gamma = 4){
  N <- ncol(Sigma)
  A <- cbind(1, diag(N))
out <- quadprog::solve.QP(Dmat = gamma * Sigma,
         dvec = mu,
         Amat = A,
         bvec = c(1, rep(0, N)),
         meq = 1)

return(as.numeric(out$solution))
}
```

```r
periods <- nrow(returns) - window_length # total number of out-of-sample periods

oos_values <- matrix(NA,
                     nrow = periods,
                     ncol = 3) # A matrix to collect all returns
colnames(oos_values) <- c("raw_return", "turnover", "net_return")

all_values <- list(oos_values,
                   oos_values,
                   oos_values,
                   oos_values)

w_prev_1 <- w_prev_2 <- w_prev_3 <- w_prev_4 <- rep(1/N ,N)

beta <- 50 / 10000
for(i in 1:periods){ # Rolling window

  # Extract information
  return_window <- returns[i : (i + window_length - 1),]

  # Sample moments
  Sigma <- cov(return_window)
  mu <- colMeans(return_window)

  # Optimal TC robust portfolio
  w_1 <- optimal_tc_weight_l1(w_prev = w_prev_1,
                              mu = mu,
                              Sigma = Sigma,
                              beta = beta)

  # Evaluation
  raw_return <- 1 + returns[i + window_length, ] %*% w_1
  turnover <- sum(abs(w_1 - w_prev_1))
  # Store realized returns
  net_return <- raw_return - beta * turnover
  all_values[[1]][i, ] <- c(raw_return, turnover, net_return)
  #Computes adjusted weights based on the weights and next period returns
  w_prev_1 <- w_1 * as.vector(1 + returns[i + window_length, ])
  w_prev_1 <- w_prev_1 / sum(as.vector(w_prev_1))
```

```r
  # Efficient portfolio
  w_2 <- optimal_tc_weight_l1(w_prev = w_prev_2,
                              mu = mu,
                              Sigma = Sigma,
                              beta = 0)

  # Evaluation
  raw_return <- 1 + returns[i + window_length, ] %*% w_2
  turnover <- sum(abs(w_2 - w_prev_2))
  # Store realized returns
  net_return <- raw_return - beta * turnover
  all_values[[2]][i, ] <- c(raw_return, turnover, net_return)
  #Computes adjusted weights based on the weights and next period returns
  w_prev_2 <- w_2 * as.vector(1 + returns[i + window_length, ])
  w_prev_2 <- w_prev_2 / sum(as.vector(w_prev_2))

  # Naive Portfolio
  w_3 <- rep(1/N, N)

  # Evaluation
  raw_return <- 1 + returns[i + window_length, ] %*% w_3
  turnover <- sum(abs(w_3 - w_prev_3))
  # Store realized returns
  net_return <- raw_return - beta * turnover
  all_values[[3]][i, ] <- c(raw_return, turnover, net_return)
  #Computes adjusted weights based on the weights and next period returns
  w_prev_3 <- w_3 * as.vector(1 + returns[i + window_length, ])
  w_prev_3 <- w_prev_3 / sum(as.vector(w_prev_3))

  # No-short sale portfolio
  w_4 <- optimal_no_short_sale_weight(mu, Sigma)

  # Evaluation
  raw_return <- 1 + returns[i + window_length, ] %*% w_4
  turnover <- sum(abs(w_4 - w_prev_4))
  # Store realized returns
  net_return <- raw_return - beta * turnover
  all_values[[4]][i, ] <- c(raw_return, turnover, net_return)
  #Computes adjusted weights based on the weights and next period returns
  w_prev_4 <- w_4 * as.vector(1 + returns[i + window_length, ])
  w_prev_4 <- w_prev_4 / sum(as.vector(w_prev_4))

}
```

```r
all_values <- lapply(all_values, as_tibble) %>%
  bind_rows(.id = "strategy")

all_values %>%
  group_by(strategy) %>%
  summarise(Mean = 250 * 100 * (mean(net_return - 1)),
            SD = sqrt(250) * 100 * sd(net_return - 1),
            Sharpe = if_else(Mean>0, Mean/SD, NA_real_),
            Turnover = 100 * mean(turnover)) %>%
```

```r
mutate(strategy = case_when(strategy == 1 ~ "MV (TC)",
                            strategy == 2 ~ "MV",
                            strategy == 3 ~ "Naive",
                            strategy == 4 ~ "MV (no-short selling)")) %>%
knitr::kable(digits = 3)
```

strategy	Mean	SD	Sharpe	Turnover
MV (TC)	26.948	40.206	0.670	0.000
MV	-5127.070	4483.310	NA	195.195
Naive	32.166	42.360	0.759	2.039
MV (no-short selling)	14.047	46.755	0.300	15.156