CASE 3 - eLAB Business Case

07 June 2023

Authors:

- Cornelius Kötting
- Jakub Czubaszek
- Tshiamo Kgosidiile
- Ajay Rayudu

Tutor: Hugo Schyns

1. Introduction

Amidst the contemporary financial vista, modest interest yields on savings accounts have driven an upsurge of investors to venture beyond conventional avenues in the quest for enhanced capital growth. This drift has triggered a substantial wave of both individual and institutional investors navigating towards the stock market’s shores. While these turbulent waters offer the tantalizing promise of elevated returns, they inherently harbor greater risks attributed to their volatile disposition. Hence, it becomes critically essential to craft a finely balanced and optimized investment portfolio, a bulwark against these uncertainties, designed to boost potential returns. The conceptual frameworks of Mean-Variance, Capital Asset Pricing Model (CAPM), and Shrinkage Optimization emerge as quintessential instruments in this endeavor. Our main quest in this assignment is to delve deeper into these financial paradigms and their practical utility in the realm of portfolio optimization. We shall unfurl these techniques using R, a powerhouse language for statistical computation and graphical visualization. R’s flexible and robust environment for data manipulation, simulation execution, and crafting intricate visualizations has made it a cherished tool in the financial sphere. Taking center stage in modern portfolio theory is the Mean-Variance portfolio theory, a brainchild of Harry Markowitz. This theory is our guide in delineating the delicate balance between the expected return and the inherent risk (denoted in terms of variance) in investment decisions. The Capital Asset Pricing Model (CAPM), another cornerstone in financial theory, amplifies the Mean-Variance doctrine by injecting the correlation between expected return and systematic risk (beta). This model sheds light on the process of pricing an asset in tandem with its risk concerning the overarching market. Lastly, we encounter Shrinkage Optimization, a relatively newer entrant, which sets its sights on enhancing the estimation of the portfolio covariance matrix. By maneuvering the sample covariance matrix towards a structured estimator – a process quaintly termed "shrinking" – this technique aims to yield more steadfast and precise portfolio optimizations. Upon the conclusion of this assignment, we shall be equipped with a deeper understanding of these methodologies, coupled with hands-on experience in deploying them in R. This acquired knowledge will serve as an invaluable compass in the crafting, optimizing, and management of better portfolios, enabling us to skillfully traverse the tempestuous waters of the stock market.

1.1 Data import

Before any analysis can be conducted, we need to import stock data to later construct and optimize the portfolios. We received a list of 20 stock tickers listed either on NYSE or NASDAQ, as well as the ticker 'SPY, which is an ETF that tracks the S&P500 index, which we will use for market return and volatility. The data was imported in daily format from 31/12/2002 until now. For improved efficiency, the data was downloaded once from Yahoo Finance and stored on our local machines for later usage. The code can be seen below:

```
tickers <- read.csv("/Users/cornelius/Downloads/teamticker.csv")
```

```
team <- "2B"
ticker_pick <- paste0("X", "2B")
```

```r
tickers_used <- c(tickers[ticker_pick])

datafile <- paste0("returns_", ticker_pick, ".csv")

if(!file.exists(datafile)) {
  getSymbols(tickers_used[[1]], from = "2002-12-31")
  returns_stocks <- (do.call(merge, lapply(tickers_used[[1]], function(x) Del
  colnames(returns_stocks) <- tickers_used[[1]]
  returns_stocks <- 100*returns_stocks["2003/"]
  write.csv(as.matrix(returns_stocks), file = datafile)

} else{
  returns_stocks <- read.csv(datafile, header = TRUE)
}
returns_stocks <- returns_stocks[, colSums(is.na(returns_stocks)) == 0]
```

2.Data preparation

After successfully importing the data, we had to prepare it for later analysis and modelling. This includes setting a risk aversion parameter, annualizing the data under the assumption that a year has 252 trading days on average. We also calculated the volatility of the individual stocks.

```r
 # Setting risk aversion parameter
gamma <- 3

# Split the data into two sub-samples: odd and even observations

even <- 2*seq(nrow(returns_stocks)/2)
odd <- even -1
returns_stocks_odd <- returns_stocks[odd,-1]
```

```r
returns_stocks_even <- returns_stocks[even,-1]

# Calculate means and variances
mu_odd <- 252*colMeans(returns_stocks_odd, na.rm = TRUE)
mu_even <- 252*colMeans(returns_stocks_even, na.rm = TRUE)

sigma_odd <- 252*cov(returns_stocks_odd)
sigma_even <- 252*cov(returns_stocks_even)

#Standard deviation
sd_odd <- sqrt(252)*apply(returns_stocks_odd,2,sd)
sd_even <- sqrt(252)*apply(returns_stocks_even,2,sd)
```

3. Portfolio construction

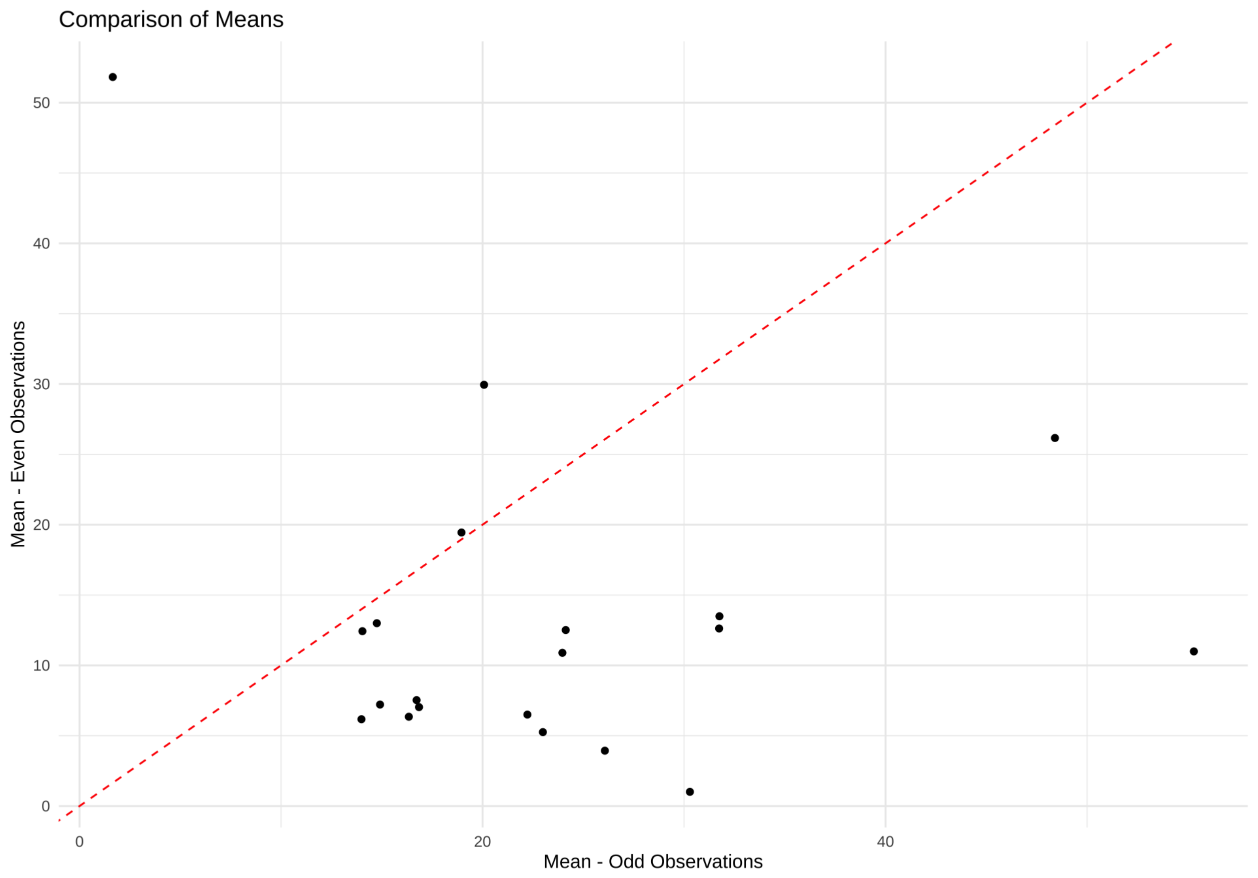
3.1 Visualizations

Before we constructed portfolios, we wanted investigate possible differences between odd and even samples. To accomplish this, we created two graphs, one comparing means of the odd sample against the mean of the even sample. The second graph does the same, but instead of means we compare the standard deviation (volatility).

3.1.1 Odd Means vs. Even Means

```r
 #Creating new dataframe for ggplot graph
ggplot_data_1 <- data.frame(mu_odd, mu_even)
ggplot(ggplot_data_1, aes(x = mu_odd, y = mu_even)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
```

```
labs(x = "Mean - Odd Observations", y = "Mean - Even Observations", title =
theme_minimal()
```

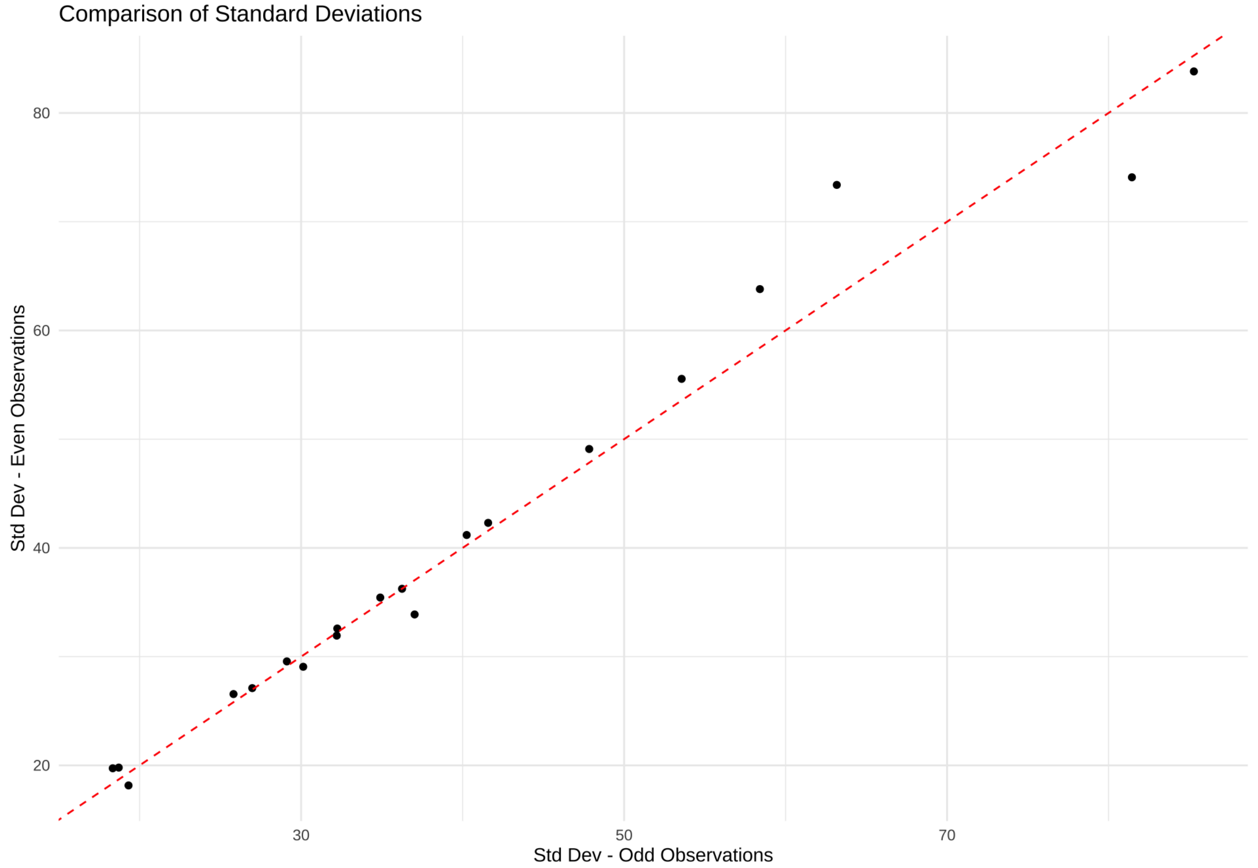


As stated before, the graph above compares means between the two samples. One can observe, that while some observations are close to the red line (y=x), most are scattered in the plot without a distinct pattern, suggesting that there are differences in the means between odd and even samples. Furthermore, the vast majority of observations are located below the red line, suggesting that the mean returns are generally higher in the even sample opposed to the odd sample. We cannot infer a specific reason for this difference, without further analysis that would go beyond the scope of this assignment. A possible explanation could be that since our data spans two decades, that the odd and even samples are capturing different parts of the year and therefore exhibiting seasonal effects.

3.1.2 Odd Standard deviations vs. Even Standard deviations

```
#Creating new dataframe for ggplot graph
ggplot_data_2 <- data.frame(sd_odd, sd_even)
```

```
ggplot(ggplot_data_2, aes(x = sd_odd, y = sd_even)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(x = "Std Dev - Odd Observations", y = "Std Dev - Even Observations", t
  theme_minimal()
```



The second graph above visualizes the relationship between the standard deviations between the two samples. All observations are on the line or very close to it. This suggests that the risk or volatility of the stocks, measured in standard deviation is not significantly different between odd and even observations. If a large difference existed this could point to potential volatility clustering, which is not the case here.

3.2 Portfolio construction

Here we stand, ready to dive into the upcoming exploration of portfolio creation, a task based on a bedrock of previously curated stock data. And the tool at our disposal? None other than the process of quadratic programming. In the realm of mathematical optimization, this method emerges as a key player. It essentially deals with an objective function, quadratic in nature, that is subject to amplification or reduction within certain linear constraints. Why exactly quadratic programming, you ask? It’s all down to its natural alignment with the tenets of Modern Portfolio Theory (MPT), as proposed by Markowitz. It’s a principle where an investor’s goal is to strike a balance—minimize risk for a specific level of expected return, or alternatively, maximize return for a specific degree of risk. In our case, quadratic programming steps in to tame the variance, a marker of risk in our portfolio, and keep it under control, subject to specific conditions. One such condition is where the magical solve.QP() function, a part of the quadprog package, steps in. We employ this function to minimize portfolio variance. The constraints it operates within are that the combined weight of all stocks in the portfolio equals one, and none of the weights are negative—thus, prohibiting short selling. When calling the solve.QP() function, it’s done as follows: solve.QP(Dmat, dvec, Amat, bvec) Here, Dmat is a symmetric matrix, doubling up the covariance matrix of the stock returns. It presents the variances of stock returns diagonally, while showcasing the covariances between each pair of stocks off-diagonally. The dvec is a vector reflecting expected returns. The Amat, on the other hand, is a matrix personifying the constraints of our optimization dilemma. Every row in Amat carries a linear inequality constraint with its values reflecting the coefficients of this inequality. The bvec is a vector demarcating the far side of the linear inequality constraints. The requirements, which state that the portfolio weights’ sum equals one and that no weight is negative, are personified in Amat and bvec components. After working out the weights for each stock in the portfolio, we aim to minimize the risk for a certain level of expected return. Following this, we calculate both the portfolio’s return and its volatility, in the context of both in-sample and out-of-sample scenarios. We further explore the performance of the portfolio, allowing for short sales. The final step of our journey involves an evaluation of the portfolio’s performance. We do this by calculating the Sharpe Ratio, a risk-adjusted return measure. This ratio represents the average return over the risk-free rate per unit of total risk, which, in this case, is volatility. A higher Sharpe Ratio is preferable, signifying a higher return for the same level of risk. The results can be seen in the tables below the code:

```
# Number of stocks
N <- ncol(returns_stocks_odd)

# Vector of ones
iota <- rep(1, N)

# Identity matrix of size N
Imat <- diag(N)
w0 <- 1
# Vector of zeros
nulvec <- rep(0, N)

A = as.matrix(cbind(iota, Imat))
b = as.matrix(c(w0, nulvec))

# Solve the model
noshort_odd = solve.QP(sigma_odd*gamma, mu_odd, Amat=A, bvec=b, meq=1)
w_odd <- noshort_odd$solution
# Solve the model
noshort_even = solve.QP(sigma_even*gamma, mu_even, Amat=A, bvec=b, meq=1)

w_even <- noshort_even$solution

#In-sample

portfolio_volatility_odd <- sqrt(t(w_odd)%*%sigma_odd%*%w_odd)
portfolio_returns_odd <- w_odd%*%mu_odd

portfolio_volatility_even <- sqrt(t(w_even)%*%sigma_even%*%w_even)
portfolio_returns_even <- w_even%*%mu_even

#Out-of sample
```

```
portfolio_volatility_odd_oos <- sqrt(t(w_odd)%*%sigma_even%*%w_odd)
portfolio_returns_odd_oos <- w_odd%*%mu_even

portfolio_volatility_even_oos <- sqrt(t(w_even)%*%sigma_odd%*%w_even)
portfolio_returns_even_oos <- w_even%*%mu_odd

#risk_free_rate = 1
#for no short selling costraints, remove Imat and nulvec from A and B

A_short = as.matrix(cbind(iota))
b_short = as.matrix(c(w0))

# Solve the model without short-sell constraints
short_odd = solve.QP(sigma_odd*gamma, mu_odd, Amat=A_short, bvec=b_short, meq
w_odd_short <- short_odd$solution

short_even = solve.QP(sigma_even*gamma, mu_even, Amat=A_short, bvec=b_short,
w_even_short <- short_even$solution

# In-sample w/ short-selling

portfolio_volatility_odd_short <- sqrt(t(w_odd_short) %*% sigma_odd %*% w_odd
portfolio_returns_odd_short <- w_odd_short %*%mu_odd

portfolio_volatility_even_short <- sqrt(t(w_even_short) %*% sigma_even %*% w_
portfolio_returns_even_short <- w_even_short %*%mu_even

# Out-of-sample w/ short-selling

portfolio_volatility_odd_oos_short <- sqrt(t(w_odd_short) %*% sigma_even %*%
portfolio_returns_odd_oos_short <- w_odd_short %*%mu_even
```

```
portfolio_volatility_even_oos_short <- sqrt(t(w_even_short) %*% sigma_odd %*%
portfolio_returns_even_oos_short <- w_even_short %*%mu_odd



risk_free_rate <- 1

# Sharpe ratios
sharpe_odd <- as.numeric((portfolio_returns_odd - risk_free_rate)) / as.numer
sharpe_even <- as.numeric((portfolio_returns_even - risk_free_rate)) / as.num
sharpe_odd_oos <- as.numeric((portfolio_returns_odd_oos - risk_free_rate)) /
sharpe_even_oos <- as.numeric((portfolio_returns_even_oos - risk_free_rate))

# Sharpe ratios for no short selling constraints portfolios
sharpe_odd_short <- (as.numeric(portfolio_returns_odd_short) - risk_free_rate
sharpe_even_short <- (as.numeric(portfolio_returns_even_short) - risk_free_ra
sharpe_odd_oos_short <- (as.numeric(portfolio_returns_odd_oos_short) - risk_f
sharpe_even_oos_short <- (as.numeric(portfolio_returns_even_oos_short) - risk
```

3.3 Interpreting results

To interpret the results of the mean-variance portfolio constructed above, we will display tables showcasing returns, volatility and Sharpe ratios for even and odd portfolios, in sample and out of sample as well as with short selling constraints and without. Addtionally we will analyze the weight of *SPY* (S&P500) in our portfolios. For increased readability, the code chunk will be omitted.

3.3.1 Summary tables

Returns wo/ short selling

Estimation	Odd	Even

Estimation	Odd	Even
Odd	17.099985	17.725750
Even	9.450403	9.449684

Volatility wo/ short selling

Estimation	Odd	Even
Odd	15.20512	15.36627
Even	15.44244	15.28011

Returns w/ short selling

Estimation	Odd	Even
Odd	16.746290	16.85749
Even	8.655346	9.06258

Volatility w/ short selling

Estimation	Odd	Even
Odd	14.75848	15.08762
Even	15.15502	14.87950

Sharpe ratio wo/ short selling

Estimation	Odd	Even
Odd	1.0588531	0.552986
Even	0.5472192	1.088472

Estimation	Odd	Even

Sharpe ratio w/ short selling

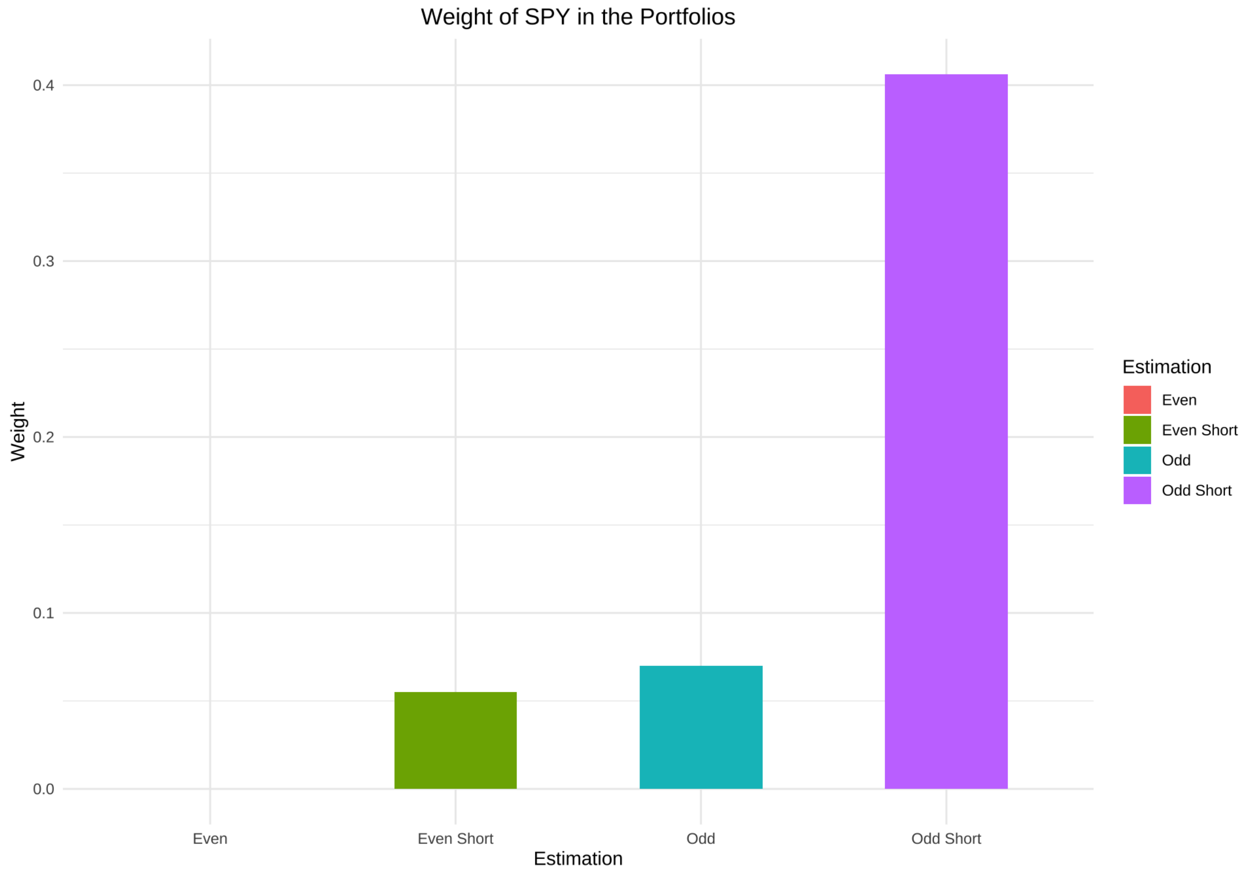
Estimation	Odd	Even
Odd	1.0669320	0.5418581
Even	0.5051359	1.0510271

We can observe a number of interesting differences between the portfolios. Firstly, the returns are slightly higher for all portfolios without short selling when compared to those with short selling. A possible reason for this difference could be that long-only portfolios are able to capture more upside potential of the stocks. Interestingly, the volatility is lower in portfolios that do not have a short selling constraint. This suggests, that these portfolios allow for improved risk management, as it provides more flexibility in adjusting the weights of the stocks within the portfolio, which could help to reduce overall volatility. The Sharpe ratios are higher for the portfolios without short selling, with the exception of the in-sample odd portfolio with short selling. This exception could indicate that shorting can improve risk-adjusted returns, however as this is in-sample and the out of sample Sharpe ratio for this portfolio is higher long-only, this does not seem to be the case here. These ratios insinuate that long-only performance generally provide better risk-adjusted returns compared to portfolios that allow for short-selling. When interpreting and analyzing these results, it is crucial to consider that these metrics were calculated using historical data, which might not predict future performance. Furthermore, these portfolios does not consider transaction costs and taxes, which can significantly impact portfolio performance.

3.3.2 Weight of *SPY*

After analyzing returns, volatility and Sharpe ratios, we now want to analyze the weight of *SPY*, which is an ETF that tracks the S&P500 index in our portfolios. The results can be seen in the graph below:

```
#Weight of SPY
spy_index <- which(tickers_used[[1]] == "SPY")
spy_weight <- data.frame(
  Estimation = c("Odd", "Even", "Odd Short", "Even Short"),
  Weight = c(
    w_odd[spy_index],
    w_even[spy_index],
    w_odd_short[spy_index],
    w_even_short[spy_index]
  )
)

# Display SPY weight as a bar chart for better visualisation
spy_weight %>%
  ggplot(aes(x=Estimation, y=Weight, fill=Estimation)) +
  geom_bar(stat="identity", width=0.5) +
  theme_minimal() +
  labs(x = "Estimation", y = "Weight", title = "Weight of SPY in the Portfoli
  theme(plot.title = element_text(hjust = 0.5))
```



As can be observed above, the proportion of *SPY* differs greatly across portfolios. It ranges from essentially 0 for the long-only even portfolio to 0.4 for the odd portfolio that allows for short selling. These differences can be attributed to a variety of reasons. Firstly, the portfolios are constructed using different subsets (odd & even observations) that could contain different market conditions, resulting in contrasting risk-return profiles for *SPY*. Furthermore, portfolios with a short selling constraints will evaluate risk-return trade-offs differently that those that allow for short selling, as there is greater flexibility when shorting is allowed.

4. CAPM Portfolios

Behold, the advent of section 4, aptly titled 'CAPM Portfolios'. This part signifies the beginning of a new epoch in the construction of our portfolios. As hinted by its name, we will be weaving in the principles of the Capital Asset Pricing Model (CAPM) into our portfolio building strategy this time around. This model, which enjoys widespread adoption, sketches out the intricate ties between expected returns and the risk tied to investing in a given security. It brings into play the concept of systematic risk, symbolized by Beta (β), to project expected returns. The birthing process of these portfolios echoes the methodology we've utilized before, with one significant enhancement—the inclusion of CAPM implied expected returns. We calculate these returns with the formula: risk_free_rate + betas * market premium, where 'betas' stand for the systematic risk associated with each security, and 'market premium' refers to the market's expected return minus the risk-free rate. The inaugural phase of our CAPM portfolio generation involves the computation of the lambda values (market premiums) for both even and odd portfolios. We base these calculations on the first expected return value (mu) of each respective portfolio. Thereafter, we figure out the Beta values for each portfolio by dividing each column of our covariance matrices by their initial elements. After that, we compute our CAPM implied expected returns with the aid of the aforementioned formula. Upon determining our CAPM implied expected returns, we employ the function solve.QP() to streamline our portfolios once again. We pursue a minimization of portfolio variance, bound by the constraints that the total weight of all stocks equals one, and that no weight plunges into the negative zone, thereby prohibiting short selling. The optimized weights for each security in the portfolios are then computed, resulting in the "no short selling" variants of the CAPM portfolios. Post-optimization, we calculate the expected return and volatility for each of these portfolios under their respective training (in-sample) and validation (out-of-sample) scenarios. These statistics provide us with a glimpse into the performance and risk exposure of each portfolio. Next, we loosen the reins on short selling and re-adjust the portfolios. We compute the weights for each security in the portfolios, leading to the "short selling permitted" iterations of the CAPM portfolios. Once again, we determine the expected return and volatility for each of these portfolios under their in-sample and out-of-sample scenarios. Distinct from our preceding portfolios, the CAPM portfolios account for systematic risk, which causes the weight allocations across the securities in the portfolios to diverge. This innovative approach could potentially amp up the risk-adjusted returns of the portfolios, contingent on the actual returns fetched by the stocks. Yet, one must bear in mind that, akin to our prior portfolios, these CAPM portfolios are founded on historical data and overlook factors like transaction costs or taxes. Given that these elements can markedly sway the performance of real-world portfolios, they warrant serious consideration when evaluating these strategies. Moreover, the CAPM model makes a number of underlying assumptions about the market and its participants that are not necessarily realistic, such as that we do not have to pay fees or taxes.

4.1 Portfolio construction

```
# CAPM

lambda_even <- mu_even[1] - risk_free_rate
lambda_odd <- mu_odd[1] - risk_free_rate

betas_odd <- sigma_odd[, 1]/sigma_odd[1, 1]
betas_even <- sigma_even[, 1]/sigma_even[1, 1]

# Compute CAPM implied means
capm_mu_odd <- risk_free_rate + betas_odd * lambda_odd
capm_mu_even <- risk_free_rate + betas_even * lambda_even

# Re-optimize the portfolios using the CAPM implied means
capm_noshort_odd = solve.QP(sigma_odd * gamma, capm_mu_odd, Amat = A, bvec =
capm_w_odd <- capm_noshort_odd$solution

capm_noshort_even = solve.QP(sigma_even * gamma, capm_mu_even, Amat = A, bvec
    meq = 1)
capm_w_even <- capm_noshort_even$solution

# Compute the expected return and standard deviation for the optimized
# portfolios in their training samples
capm_portfolio_returns_odd <- capm_w_odd %*% mu_odd
capm_portfolio_volatility_odd <- sqrt(t(capm_w_odd) %*% sigma_odd %*% capm_w_

capm_portfolio_returns_even <- capm_w_even %*% mu_even
capm_portfolio_volatility_even <- sqrt(t(capm_w_even) %*% sigma_even %*% capm
```

```
# Compute the expected return and standard deviation for the optimized
# portfolios in their validation samples
capm_portfolio_returns_odd_oos <- capm_w_odd %*% mu_even
capm_portfolio_volatility_odd_oos <- sqrt(t(capm_w_odd) %*% sigma_even %*% ca

capm_portfolio_returns_even_oos <- capm_w_even %*% mu_odd
capm_portfolio_volatility_even_oos <- sqrt(t(capm_w_even) %*% sigma_odd %*% c

# With short selling

# Re-optimize the portfolios using the CAPM implied means
capm_short_odd = solve.QP(sigma_odd * gamma, capm_mu_odd, Amat = A_short, bve
    meq = 1)
capm_w_odd_short <- capm_short_odd$solution

capm_short_even = solve.QP(sigma_even * gamma, capm_mu_even, Amat = A_short,
    meq = 1)
capm_w_even_short <- capm_short_even$solution

# Compute the expected return and standard deviation for the optimized
# portfolios in their training samples
capm_portfolio_returns_odd_short <- capm_w_odd_short %*% mu_odd
capm_portfolio_volatility_odd_short <- sqrt(t(capm_w_odd_short) %*% sigma_odd
    capm_w_odd_short)

capm_portfolio_returns_even_short <- capm_w_even_short %*% mu_even
capm_portfolio_volatility_even_short <- sqrt(t(capm_w_even_short) %*% sigma_e
    capm_w_even_short)

# Compute the expected return and standard deviation for the optimized
# portfolios in their validation samples
```

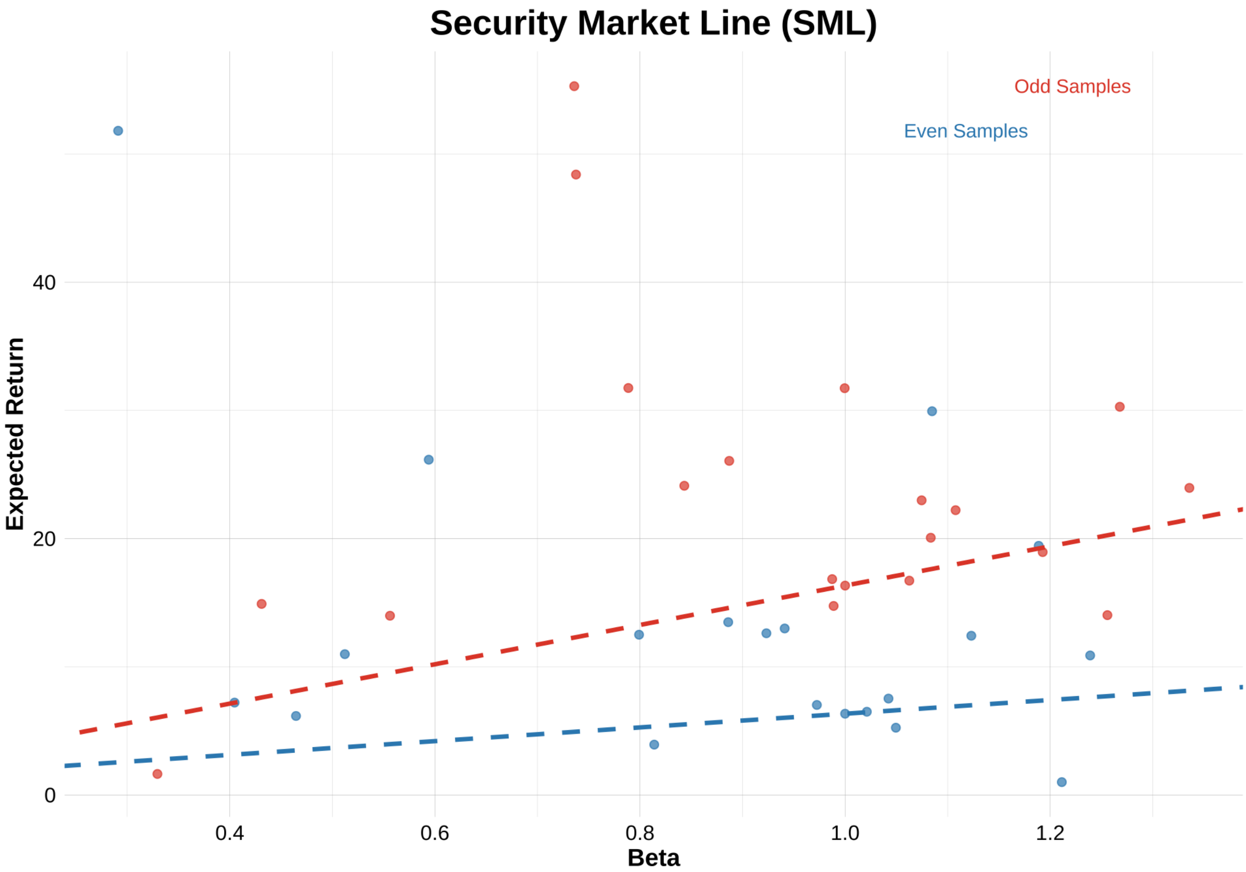
```
capm_portfolio_returns_odd_oos_short <- capm_w_odd_short %*% mu_even
capm_portfolio_volatility_odd_oos_short <- sqrt(t(capm_w_odd_short) %*% sigma
    capm_w_odd_short)

capm_portfolio_returns_even_oos_short <- capm_w_even_short %*% mu_odd
capm_portfolio_volatility_even_oos_short <- sqrt(t(capm_w_even_short) %*% sig
    capm_w_even_short)



# Sharpe ratios
capm_sharpe_odd <- as.numeric((capm_portfolio_returns_odd - risk_free_rate))/
capm_sharpe_even <- as.numeric((capm_portfolio_returns_even - risk_free_rate)
capm_sharpe_odd_oos <- as.numeric((capm_portfolio_returns_odd_oos - risk_free
capm_sharpe_even_oos <- as.numeric((capm_portfolio_returns_even_oos - risk_fr

# Sharpe ratios for no short selling constraints portfolios
capm_sharpe_odd_short <- (as.numeric(capm_portfolio_returns_odd_short) - risk
capm_sharpe_even_short <- (as.numeric(capm_portfolio_returns_even_short) - ri
capm_sharpe_odd_oos_short <- (as.numeric(capm_portfolio_returns_odd_oos_short
    risk_free_rate)/as.numeric(capm_portfolio_volatility_odd_oos_short)
capm_sharpe_even_oos_short <- (as.numeric(capm_portfolio_returns_even_oos_sho
    risk_free_rate)/as.numeric(capm_portfolio_volatility_even_oos_short)
```

4.2 Visualizing CAPM & SML

After birthing and optimizing the CAPM portfolios, in the next subsection, we will visualize these portfolios and the Security Market Line (SML).

```
 # Scatterplot for βi against the sample means mi for odd and even samples
ggplot_data_sml <- data.frame(beta_odd = betas_odd, mean_odd = mu_odd, beta_e
```

```
ggplot() +
  # Add points for odd samples
  geom_point(data = ggplot_data_sml, aes(x = beta_odd, y = mean_odd), color =
  # Add points for even samples
  geom_point(data = ggplot_data_sml, aes(x = beta_even, y = mean_even), color
  # Add SML line for odd samples
  geom_abline(intercept = risk_free_rate, slope = lambda_odd, color = '#E24A3
  # Add SML line for even samples
  geom_abline(intercept = risk_free_rate, slope = lambda_even, color = '#348A
  labs(x = "Beta", y = "Expected Return") +
  theme_minimal() +
  ggtitle("Security Market Line (SML)") +
  theme(
    plot.title = element_text(hjust = 0.5, face="bold", size=20, color="black
    axis.title.x = element_text(face="bold", size=14, color="black"),
    axis.title.y = element_text(face="bold", size=14, color="black"),
    axis.text = element_text(size=12, color="black"),
    legend.position = "bottom",
    legend.title = element_blank(),
    panel.grid.major = element_line(color = "grey", linewidth = 0.1),
    panel.grid.minor = element_line(color = "grey", linewidth = 0.05)
  ) +
  annotate("text", x = max(ggplot_data_sml$beta_odd, na.rm = TRUE), y = max(g
  annotate("text", x = max(ggplot_data_sml$beta_even, na.rm = TRUE), y = max(
```



Each point on the graph above represents an individual stock. The x-coordinate of a point is a measure of the asset's market risk (beta). The y-coordinate represents the asset's expected return. The two SML lines describe the relationship between expected return and beta (systemic risk) for an assets. If a point lies above its respective SML line, this suggests that the assets provides a higher return than expected for the level of risk. Conversely, if a point lies below the line, its expected return is lower than expected of the asset's risk profile. As can be observed in the graph, the assets in the odd sample generally exhibit a higher expected return for their level of risk when compared to the even sample. Furthermore, very few individual stocks are situated below their SML lines, indicating that the portfolios are generally well optimized for their given risk tolerance.

4.3 Interpreting results

To compare the CAPM optimized portfolios, we will construct the same tables showcasing returns, volatility and Sharpe ratios. Once again the code chunk will be omitted for increased visibility.

4.3.1 Summary tables

Returns wo/ short selling

Estimation	Odd	Even
Odd	16.746171	17.682685
Even	9.366497	9.239863

Volatility wo/ short selling

Estimation	Odd	Even
Odd	15.20207	15.36136
Even	15.48812	15.27800

Returns w/ short selling

Estimation	Odd	Even
Odd	16.277822	16.855742
Even	8.630352	8.587221

Volatility w/ short selling

Estimation	Odd	Even
Odd	14.75410	15.07132
Even	15.19281	14.87441

Sharpe ratio wo/ short selling

Estimation	Odd	Even
Odd	1.0357909	0.5393286

Estimation	Odd	Even
Even	0.5401879	1.0860164

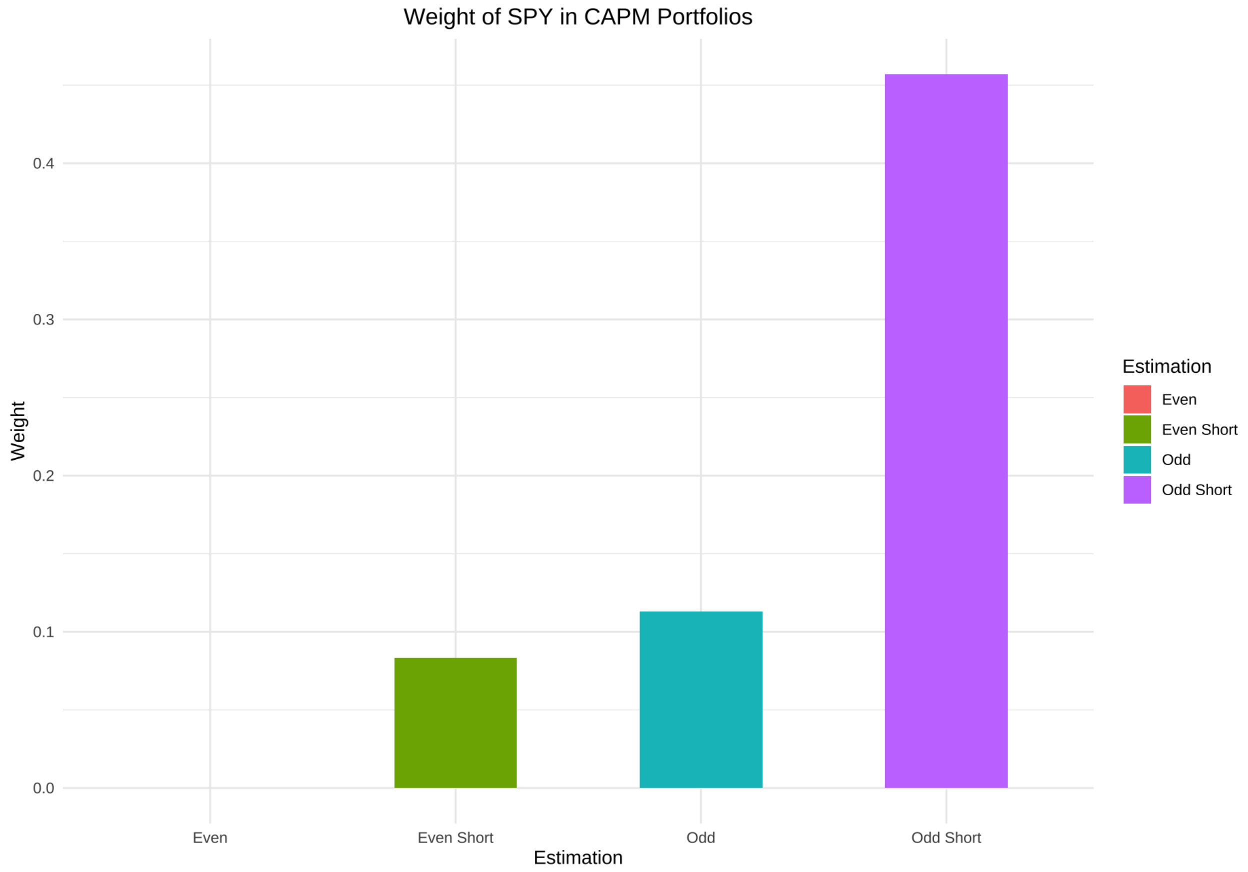
Sharpe ratio w/ short selling

Estimation	Odd	Even
Odd	1.0669320	0.5418581
Even	0.5051359	1.0510271

We can observe the same relationships in the CAPM optimized portfolios when compared to those that were not optimized using CAPM. Returns and volatility are higher for the portfolios that are long-only, while the lower volatility in portfolios with short selling highlights the possibility for greater risk management using short selling. Once again, the Sharpe ratios are higher for portfolios that are long-only with the exception of the in sample odd portfolio with short selling, however these differences are marginal. More interestingly though, returns are lower for both long and long/short portfolios optimized using CAPM compared to those that were not. Volatility is also higher in the CAPM portfolios. The Sharpe ratios are lower in CAPM portfolios without short selling, while they are identical in those with short selling. The observed results point to some criticisms of the Capital Assets Pricing Model. Firstly, the model assumes that return is solely determined by its beta, which is not applicable to real world scenarios. Furthermore, mean-variance optimized portfolios seek to identify the portfolio that maximizes the level of returns for a given risk level. As this approach is consistently outperforming CAPM optimized portfolios, it is reasonable to conclude that it is beneficial to utilize a more nuanced understanding of risk and return rather than relying on beta alone. This does not mean that the CAPM model is without value, as it has been influential in financial theory and practice, however when it comes to constructing actual portfolios there are more effective strategies available.

4.3.2 Weight of SPY in CAPM portfolios

```
#Weight of SPY in CAPM
capm_spy_weight <- data.frame(
  Estimation = c("Odd", "Even", "Odd Short", "Even Short"),
  Weight = c(
    capm_w_odd[spy_index],
    capm_w_even[spy_index],
    capm_w_odd_short[spy_index],
    capm_w_even_short[spy_index]
  )
)


# Display SPY weight as a bar chart for better visualisation
capm_spy_weight %>%
  ggplot(aes(x=Estimation, y=Weight, fill=Estimation)) +
  geom_bar(stat="identity", width=0.5) +
  theme_minimal() +
  labs(x = "Estimation", y = "Weight", title = "Weight of SPY in CAPM Portfol
  theme(plot.title = element_text(hjust = 0.5))
```



As can be seen in the bar graph above, the weight of *SPY* also differs greatly among CAPM portfolios. While the weight is generally a little higher when compared to the mean-variance optimized portfolios, the same tendencies can be observed. The odd portfolio with short selling has the highest weight of *SPY*, while the even portfolio without short selling has a weight of virtually zero. Possible reasons for this are explained under 3.3.2.

5. Shrinkage portfolios

Upon embarking on the journey through Section 5, we're introduced to the intriguing concept of 'Shrinkage Portfolios'. This segment embraces the principle of shrinkage in our portfolio creation process. Shrinkage, an approach favored far and wide, provides a valuable antidote to the problems triggered by data overfitting. Overfitting is a pervasive pitfall in portfolio optimization; an optimal portfolio crafted from historical returns may falter when faced with fresh, unseen data due to an excessive dependence on the nuances of past data. Shrinkage comes to the rescue here, as it essentially 'shrinks' or nudges the estimates of expected returns towards a more cautious estimate, in this scenario, the CAPM implied returns. Shrinkage implementation invites a new variable into the equation—kappa (κ). This value, a number between 0 and 1, is a measure of the shrinkage degree. A kappa of 0 implies no shrinkage; the portfolio is optimized based on historical mean returns. A kappa of 1 signals full shrinkage, meaning that the portfolio is optimized solely based on the CAPM implied returns. Intermediate kappa values represent a mix of the historical mean and the CAPM implied returns, thus infusing an element of adaptability and authority over the shrinkage degree. Our journey into shrinkage portfolio creation starts with specifying a sequence of kappa values. For each kappa, we calculate the shrinkage estimators by weighing the historical means and the CAPM implied returns by (1 - kappa) and kappa, respectively. Subsequently, we reconfigure our portfolios using these shrinkage estimators.Echoing the modus operandi of previous sections, we first assemble the portfolios excluding short selling. This process involves solving the quadratic programming conundrum with the constraints that all weights add up to one, and none of the weights dip into negative territory. Thereafter, we estimate the expected return and volatility for each of these portfolios under their respective out-of-sample scenarios. Next, we allow for short selling and reorganize the portfolios, thus establishing the “short selling allowed” versions of the shrinkage portfolios. Again, we calculate the out-of-sample expected return and volatility for these portfolios. The Sharpe ratios, both for portfolios that include and exclude short selling constraints, are then computed for each kappa value. These ratios gauge the risk-adjusted returns of the portfolios. We consolidate all these findings into a data frame for each kappa value, crafting a panoramic view of the performance of the shrinkage portfolios across varying shrinkage levels. The standout difference between shrinkage portfolios and those crafted through mean-variance or CAPM optimization resides in the adaptability of shrinkage portfolios. They effectively tackle the overfitting issue, a potential inherent shortcoming of the other two methods. This can potentially birth a more resilient and efficient portfolio, delivering superior out-of-sample performance. However, a word of caution: like the other methods, shrinkage portfolios also hinge on historical data and turn a blind eye to transaction costs or taxes, factors that could notably alter the performance of real-world portfolios.

```
# Set a range of kappa values to test
kappa_values <- seq(0, 1, by = 0.1)

# Create a data frame to store the cross-validation results
cross_val_results <- data.frame(kappa = double(), Odd = double(), Even = doub
    Odd_Short = double(), Even_Short = double())

for (kappa in kappa_values) {

    # Compute shrinkage estimators
    mu_odd_shrink <- kappa * capm_mu_odd + (1 - kappa) * mu_odd
    mu_even_shrink <- kappa * capm_mu_even + (1 - kappa) * mu_even

    # Re-optimize the portfolios using the shrinkage estimates
    shrink_noshort_odd = solve.QP(sigma_odd * gamma, mu_odd_shrink, Amat = A,
        meq = 1)
    shrink_w_odd <- shrink_noshort_odd$solution

    shrink_noshort_even = solve.QP(sigma_even * gamma, mu_even_shrink, Amat =
        bvec = b, meq = 1)
    shrink_w_even <- shrink_noshort_even$solution

    # Re-optimize the portfolios using the shrinkage estimates w/ short selli
    shrink_short_odd = solve.QP(sigma_odd * gamma, mu_odd_shrink, Amat = A_sh
        bvec = b_short, meq = 1)
    shrink_w_odd_short <- shrink_short_odd$solution

    shrink_short_even = solve.QP(sigma_even * gamma, mu_even_shrink, Amat = A
        bvec = b_short, meq = 1)
    shrink_w_even_short <- shrink_short_even$solution

    # Compute the out-of-sample Sharpe ratios without shorting
```

```
    sharpe_odd_oos <- (as.numeric(shrink_w_odd %*% mu_even) - risk_free_rate)
      sigma_even %*% shrink_w_odd)
    sharpe_even_oos <- (as.numeric(shrink_w_even %*% mu_odd) - risk_free_rate
      sigma_odd %*% shrink_w_even)

    # Compute the out-of-sample Sharpe ratios with shorting
    sharpe_odd_oos_short <- (as.numeric(shrink_w_odd_short %*% mu_even) - ris
      sigma_even %*% shrink_w_odd_short)
    sharpe_even_oos_short <- (as.numeric(shrink_w_even_short %*% mu_odd) - ri
      sigma_odd %*% shrink_w_even_short)

    # Store the results
    cross_val_results <- rbind(cross_val_results, c(kappa, sharpe_odd_oos, sh
      sharpe_odd_oos_short, sharpe_even_oos_short))
}

# Name the columns of the data frame
names(cross_val_results) <- c("kappa", "Odd", "Even", "Odd_Short", "Even_Shor
```

5.1 Interpreting results

```
 # Print the cross-validation results
cross_val_results %>% kable("html", caption = "Sharpe Ratios by Kappa and por
```

Sharpe Ratios by Kappa and portfolio

kappa	Odd	Even	Odd_Short	Even_Short
0.0	0.5472192	1.088472	0.5051359	1.051027
0.1	0.5465215	1.088241	0.5048634	1.051162

kappa	Odd	Even	Odd_Short	Even_Short
0.2	0.5458206	1.088006	0.5045870	1.051290
0.3	0.5451165	1.087768	0.5043066	1.051410
0.4	0.5444093	1.087528	0.5040222	1.051523
0.5	0.5436991	1.087284	0.5037340	1.051629
0.6	0.5429857	1.087037	0.5034418	1.051727
0.7	0.5422693	1.086786	0.5031458	1.051818
0.8	0.5415498	1.086533	0.5028458	1.051902
0.9	0.5408274	1.086276	0.5025420	1.051979
1.0	0.5401879	1.086016	0.5022343	1.052048

The table above, displays the Sharpe ratios for the shrinkage optimized validation portfolios by kappa. Once again we can observe some interesting results. The shrinkage method can increase the Sharpe ratio and therefore risk-adjusted performance of the even portfolios, while it decreases the Sharpe ratios of the odd portfolios. However, as the goal of shrinkage is to reduce the influence of possible overfitting on historical data, we would need to test the performance of these portfolios on new data that they were not optimized on to get a more robust estimate of the portfolios performance. The increased Sharpe ratios of the even portfolios with and without short selling do indicate however that shrinkage can improve portfolio performance.

5.2 Optimal Kappa

While the performance measured by Sharpe ratio of some portfolios could be increased using shrinkage, this was not consistent, as the odd portfolios did not benefit from shrinkage. Therefore it is not possible to determine one optimal value for Kappa that increases performance accross portfolios. Furthermore, as mentioned before the performance of these shrinkage portfolios should be measured on new data rather than historical data for a more robust estimate.

6. Conclusion

The exploration of portfolio optimization has been a journey through a spectrum of methodologies - from the nuanced art of mean-variance optimization, traversing the landscape of the Capital Asset Pricing Model (CAPM), to the culmination in the realm of shrinkage portfolios. Each technique unfurls unique principles and mechanisms contributing to portfolio construction, and the resultant analysis has birthed a fascinating array of findings. The implementation of mean-variance optimization offered the advantage of pinpointing portfolios that, for a pre-defined risk level, optimize the return, or, for a given expected return, minimize the risk. This strategy yielded dividends in the form of higher Sharpe ratios compared to those realized by portfolios optimized employing the CAPM. The gleaned observations point towards a potentially more productive strategy for portfolio construction, grounded in a sophisticated understanding of risk and return, rather than a solitary reliance on beta. A noteworthy finding pertained to the portfolios allowing short selling, which displayed reduced volatility. This suggests that short selling might serve as an additional tool for managing risk. Curiously though, the portfolios prohibiting short selling, the so-called long-only portfolios, manifested higher returns and Sharpe ratios. This phenomenon could be ascribed to the inherent ability of long-only portfolios to harness the upside potential of the stocks more efficiently. With regards to the Capital Asset Pricing Model, portfolios optimized using this model exhibited lower returns and increased volatility in comparison to the portfolios optimized through mean-variance. Nevertheless, despite the somewhat idealistic assumptions about the market and its participants made by the model, it continues to offer vital theoretical frameworks that deepen our comprehension of financial markets and decision-making. The incorporation of shrinkage into our portfolio optimization procedure introduced an added layer of adaptability. This addition granted us the ability to modulate the reliance on historical returns versus the CAPM implied returns. This adjustment turned out to be advantageous in bolstering the Sharpe ratios of the even portfolios, signaling that shrinkage could potentially enhance portfolio performance. A crucial caveat to bear in mind is that the performance of these portfolios was assessed on historical data, and the real-world deployment of these strategies would necessitate accounting for real-world factors such as transaction costs and taxes. Moreover, these strategies would require validation against new, unseen data to affirm their effectiveness. Nonetheless, the strategies dissected in this assignment have imparted valuable insights into portfolio optimization methodologies and their potential influence on portfolio performance.