Project

2024-11-26

1. Daily returns for industry for 2017-2019.

For this part, we download two types of data for this - the industry level series on Yahoo Finance, and the top 3 companies in the sector, and calculate daily returns for your industry for 2017-2019. . Then we will Justify the ticker choice we made to represent the industries (e.g., the most famous ones, largest ones, ones we are particularly curious about, etc.). And then we will plot the cumulative performance of the industry and the individual assets.

```
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(dplyr)
```

```
##
## ######################### Warning from 'xts' package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                            #
## #                                                                             #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                             #
## ###############################################################################
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

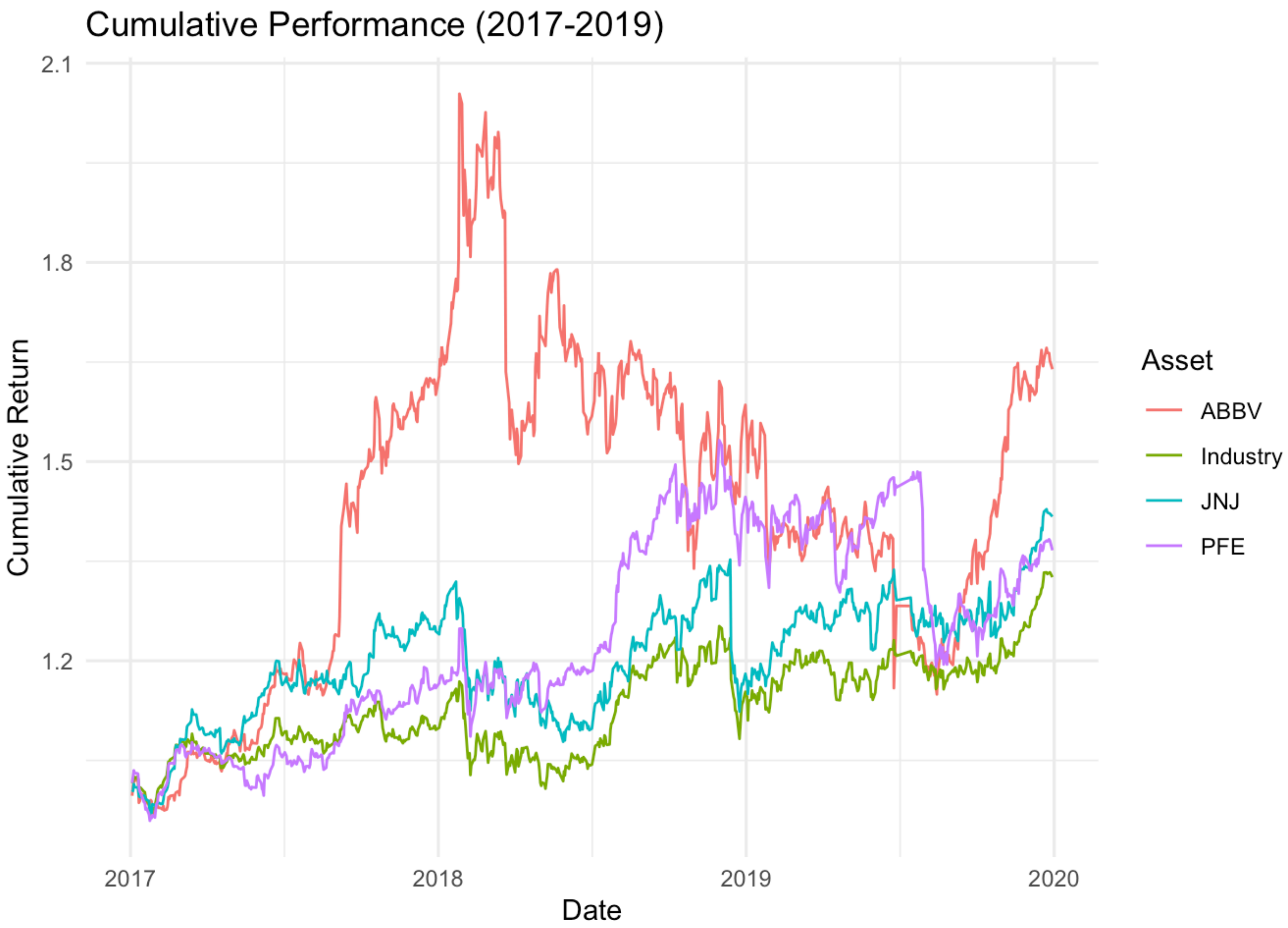
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyr)

# Define tickers and date range
start_date <- "2016-12-30"
end_date <- "2019-12-31"
industry_ticker <- "^SP500-352020"  # Industry index
company_tickers <- c("PFE", "JNJ", "ABBV")  # Selected companies
```

```
# Download industry and company data
industry_data <- getSymbols(industry_ticker, src = "yahoo",
                            from = start_date, to = end_date,
                            auto.assign = FALSE)
```

```
## Warning: ^SP500-352020 contains missing values. Some functions will not work if
## objects contain missing values in the middle of the series. Consider using
## na.omit(), na.approx(), na.fill(), etc to remove or replace them.
```

```r
industry_prices <- na.omit(Ad(industry_data))  # Extract Adjusted Close prices

company_prices<-do.call(merge,
                        lapply(company_tickers,
                        function(ticker) Ad(getSymbols(ticker,
                        src = "yahoo",
                        from = start_date,
                        to = end_date,
                        auto.assign = FALSE))))
colnames(company_prices) <- company_tickers  # Name columns by tickers

# Combine industry and company prices
all_prices <- merge(industry_prices, company_prices)
colnames(all_prices) <- c("Industry", company_tickers)

# Calculate daily returns
daily_returns <- na.omit(ROC(all_prices, type = "discrete"))

# Calculate cumulative returns
cumulative_returns <- cumprod(1 + daily_returns)

# Convert to long format for plotting
cumulative_returns_long <- as.data.frame(cumulative_returns) %>%
  mutate(Date = index(cumulative_returns)) %>%
  pivot_longer(-Date, names_to = "Asset", values_to = "CumulativeReturn")

# Plot cumulative performance
ggplot(cumulative_returns_long, aes(x = Date, y = CumulativeReturn,
                                    color = Asset)) +
  geom_line() +
  labs(title = "Cumulative Performance (2017-2019)",
       x = "Date",
       y = "Cumulative Return",
       color = "Asset") +
  theme_minimal()
```



2.Summary statistics

For this part, we will report summary statistics like mean and sd for daily returns by year and industry i.e., in two Nx2 table with periods along the columns and assets along the rows. N denotes the number of assets (i.e., number of individual companies and the overall series). One table is for reporting mean, the other for sd.

```
# Add a "Year" column to daily returns
daily_returns_with_year <- daily_returns %>%
  as.data.frame() %>%
  mutate(Date = index(daily_returns),
         Year = format(Date, "%Y")) %>%
  select(-Date)  # Remove the Date column as it’s not needed for grouping

# Calculate mean and standard deviation for each year and asset
mean_table <- daily_returns_with_year %>%
  group_by(Year) %>%
  summarise(across(everything(), mean, .names = "Mean_{.col}(%)")) %>%
  pivot_longer(-Year, names_to = "Asset", values_to = "Mean") %>%
  pivot_wider(names_from = Year, values_from = Mean)

sd_table <- daily_returns_with_year %>%
  group_by(Year) %>%
  summarise(across(everything(), sd, .names = "SD_{.col}(%)")) %>%
  pivot_longer(-Year, names_to = "Asset", values_to = "SD") %>%
  pivot_wider(names_from = Year, values_from = SD)

# Convert back to data frames for display
mean_table <- as.data.frame(mean_table)
sd_table <- as.data.frame(sd_table)
mean_table[, sapply(mean_table, is.numeric)] <-
  lapply(mean_table[, sapply(mean_table, is.numeric)],
         function(x) round(x * 100, 3))
sd_table[, sapply(sd_table, is.numeric)] <-
  lapply(sd_table[, sapply(sd_table, is.numeric)],
         function(x) round(x * 100, 3))
# Print the tables
print("Mean Table:")
```

```
## [1] "Mean Table:"
```

```
print(mean_table)
```

```
##              Asset  2017   2018   2019
## 1 Mean_Industry(%) 0.038  0.026  0.061
## 2      Mean_PFE(%) 0.061  0.096 -0.016
## 3      Mean_JNJ(%) 0.090 -0.011  0.081
## 4     Mean_ABBV(%) 0.194  0.021  0.029
```

```
print("Standard Deviation Table:")
```

```
## [1] "Standard Deviation Table:"
```

```
print(sd_table)
```

```
##             Asset  2017  2018  2019
## 1 SD_Industry(%) 0.580 1.103 0.828
## 2      SD_PFE(%) 0.707 1.245 1.194
## 3      SD_JNJ(%) 0.721 1.380 1.013
## 4     SD_ABBV(%) 1.104 2.217 1.730
```

3. Optimal portfolio

For this part, we will use techniques we learnt in portfolio optimization to create an optimal portfolio of the individual companies in your industry. Comment on the riskiness of the assets, the overall series and your constructed portfolio in the Markowitz portfolio optimization world (meaning using standard deviations and Sharpe ratios).

```
# 30-Year Treasury Bond Yield as risk free proxy and convert rate to daily assuming
continuous compounding
rf <- getSymbols("^TYX", src = "yahoo", from = start_date,
                 to = end_date, auto.assign = FALSE)
```

```
## Warning: ^TYX contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

```r
rf <- mean(na.omit(rf$TYX.Adjusted))
rf <- log(1 + rf) / 365

asset.names <- c("PFE", "JNJ", "ABBV")
mu.vec = c(mean(daily_returns[,2]), mean(daily_returns[,3]),
           mean(daily_returns[,4]))
names(mu.vec) = asset.names

# var-cov matrix
# Compute variance-covariance matrix
sigma.mat <- cov(daily_returns[, 2:4], use = "complete.obs")
rownames(sigma.mat) <- asset.names
colnames(sigma.mat) <- asset.names
sd.vec = sqrt(diag(sigma.mat))

num = solve(sigma.mat)%*%(mu.vec-rf/100)
den = as.numeric(t(rep(1,3))%*%solve(sigma.mat)%*%(mu.vec-rf/100))
tan.vec = num/den
mu_tan = as.numeric(crossprod(tan.vec, mu.vec))
sd_tan = sqrt(as.numeric(t(tan.vec)%*%sigma.mat%*%tan.vec))
SR_tan = (mu_tan - rf/100)/sd_tan

mu_industry <- mean(daily_returns[,1])*100
sd_industry <- sd(daily_returns[,1]*100)
SR_industry <- (mu_industry - rf)/sd_industry

mu_PFE <- mean(daily_returns[,2])*100
sd_PFE <- sd(daily_returns[,2]*100)
SR_PFE <- (mu_PFE - rf)/sd_PFE

mu_JNJ <- mean(daily_returns[,3])*100
sd_JNJ <- sd(daily_returns[,3]*100)
SR_JNJ <- (mu_JNJ - rf)/sd_JNJ

mu_ABBV <- mean(daily_returns[,4])*100
sd_ABBV <- sd(daily_returns[,4]*100)
SR_ABBV <- (mu_ABBV - rf)/sd_ABBV

cat("Weight of Pfizer (PFE):", round(tan.vec[1],3))
```

```
## Weight of Pfizer (PFE): 0.229
```

```r
cat("\nWeight of Johnson and Johnson (JNJ):", round(tan.vec[2],3))
```

```
##
## Weight of Johnson and Johnson (JNJ): 0.482
```

```r
cat("\nWeight of AbbVie (ABBV):", round(tan.vec[3],3))
```

```
##
## Weight of AbbVie (ABBV): 0.289
```

```
metrics <- data.frame(
  Asset = c("PFE", "JNJ", "ABBV", "Industry", "Optimal Portfolio"),
  Std_Dev = round(c(sd_PFE, sd_JNJ, sd_ABBV, sd_industry, sd_tan*100),3),
  Sharpe_Ratio = round(c(SR_PFE, SR_JNJ, SR_ABBV, SR_industry, SR_tan),3)
)
print(metrics)
```

```
##               Asset Std_Dev Sharpe_Ratio
## 1               PFE   1.074        0.041
## 2               JNJ   1.072        0.046
## 3              ABBV   1.744        0.045
## 4          Industry   0.863        0.044
## 5 Optimal Portfolio   0.993        0.057
```

The industry has a standard deviation that is the lowest among all securities with moderate Sharpe ratio. This is expected given that it is the most diversified with many assets. Individual assets (PFE, JNJ and ABBV) have the highest standard deviations due to the absence of any diversification benefits. Individual assets have different levels of Sharpe ratios with PFE having the lowest Sharpe ratio and JNJ having the highest one among the three. The optimal portfolio has the highest Sharpe ratio which is also expected due to the goal of maximizing this objective. Although it bears slightly higher risks in terms of standard deviation than the industry series, its risk to reward ratio is the most optimal.

4. Comments on the pf

For this part, we will compare the performance of the pf constructed to the overall series during your sample period, and also over the next 3 years, namely, 2020-2022.

```r
# Calculate daily and cumulative return given optimal portfolio weights
pf_daily_return <- tan.vec[1]*daily_returns[,2] + tan.vec[2]*daily_returns[,3] + ta
n.vec[3]*daily_returns[,4]
names(pf_daily_return) = "Optimal Pf Returns"
pf_cum_returns <- cumprod(1 + pf_daily_return)*100

# Add performance metrics to the data frame.
metrics <- data.frame(
  Asset = c("PFE", "JNJ", "ABBV", "Industry", "Optimal Portfolio"),
  Mean_Return = round(c(mu_PFE, mu_JNJ, mu_ABBV, mu_industry, mu_tan*100),3),
  Cumulative_return = round(c(cumulative_returns[nrow(cumulative_returns),2]*100,
                      cumulative_returns[nrow(cumulative_returns),3]*100,
                      cumulative_returns[nrow(cumulative_returns),4]*100,
                      cumulative_returns[nrow(cumulative_returns),1]*100,
                      pf_cum_returns[nrow(pf_cum_returns)]),3),
  Std_Dev = round(c(sd_PFE, sd_JNJ, sd_ABBV, sd_industry, sd_tan*100),3),
  Sharpe_Ratio = round(c(SR_PFE, SR_JNJ, SR_ABBV, SR_industry, SR_tan),3)
)
rownames(metrics) <- NULL
colnames(metrics) <- c("Asset/Portfolio (2017-2019)", "Mean_daily_R(%)", "Cumulativ
e_R(%)", "Std_Dev", "Sharpe_Ratio")
print(metrics)
```

```
##   Asset/Portfolio (2017-2019) Mean_daily_R(%) Cumulative_R(%) Std_Dev
## 1                         PFE           0.048         136.648   1.074
## 2                         JNJ           0.053         141.718   1.072
## 3                        ABBV           0.082         163.919   1.744
## 4                    Industry           0.042         132.594   0.863
## 5           Optimal Portfolio           0.060         150.577   0.993
##   Sharpe_Ratio
## 1        0.041
## 2        0.046
## 3        0.045
## 4        0.044
## 5        0.057
```

```r
# Download asset and industry prices and calculate daily and cumulative returns
start_date_2020 <- "2019-12-31"
end_date_2022 <- "2022-12-31"
company_prices <- do.call(merge, lapply(company_tickers, function(ticker) Ad(getSym
bols(ticker, src = "yahoo", from = start_date_2020, to = end_date_2022, auto.assign
= FALSE))))
industry_data <- getSymbols(industry_ticker, src = "yahoo", from = start_date_2020,
to = end_date_2022, auto.assign = FALSE)
industry_prices <- na.omit(Ad(industry_data))
all_prices <- merge(industry_prices, company_prices)
colnames(all_prices) <- c("Industry", company_tickers)
daily_returns_2020 <- na.omit(ROC(all_prices, type = "discrete"))
cumulative_returns_2020 <- cumprod(1+daily_returns_2020)*100
# Assuming the same portfolio weights, calculate the daily and cumulative returns f
rom prices in the new period
pf_daily_return_2020 <- tan.vec[1]*daily_returns_2020[,2] + tan.vec[2]*daily_return
```

```
s_2020[,3] + tan.vec[3]*daily_returns_2020[,4]
pf_cum_returns_2020 <- cumprod(1 + pf_daily_return_2020)*100

# Calculate mean daily returns
mu_industry_2020 <- mean(daily_returns_2020[,1]*100)
mu_PFE_2020 <- mean(daily_returns_2020[,2])*100
mu_JNJ_2020 <- mean(daily_returns_2020[,3])*100
mu_ABBV_2020 <- mean(daily_returns_2020[,4])*100
mu_tan_2020 <- mean(pf_daily_return_2020)*100

# Calculate standard deviation of daily returns
sd_PFE_2020 <- sd(daily_returns_2020[,2]*100)
sd_JNJ_2020 <- sd(daily_returns_2020[,3]*100)
sd_ABBV_2020 <- sd(daily_returns_2020[,4]*100)
sd_industry_2020 <- sd(daily_returns_2020[,1]*100)
sd_tan_2020 <- sd(pf_daily_return_2020*100)

# Calculate Sharpe ratio of daily returns
SR_PFE_2020 <- (mu_PFE_2020 - rf)/sd_PFE_2020
SR_JNJ_2020 <- (mu_JNJ_2020 - rf)/sd_JNJ_2020
SR_ABBV_2020 <- (mu_ABBV_2020 - rf)/sd_ABBV_2020
SR_industry_2020 <- (mu_industry_2020 - rf)/sd_industry_2020
SR_tan_2020 <- (mu_tan_2020 - rf)/sd_tan_2020


metrics_2020 <- data.frame(
  Asset = c("PFE", "JNJ", "ABBV", "Industry", "Optimal Portfolio"),
  Mean_Return = round(c(mu_PFE_2020, mu_JNJ_2020, mu_ABBV_2020, mu_industry_2020, m
u_tan_2020),3),
  Cumulative_return = round(c(cumulative_returns_2020[nrow(cumulative_returns),2],
                        cumulative_returns_2020[nrow(cumulative_returns),3],
                        cumulative_returns_2020[nrow(cumulative_returns),4],
                        cumulative_returns_2020[nrow(cumulative_returns),1],
                        pf_cum_returns_2020[nrow(pf_cum_returns_2020)]),3),
  Std_Dev = round(c(sd_PFE_2020,
              sd_JNJ_2020,
              sd_ABBV_2020,
              sd_industry_2020,
              sd_tan_2020),3),
  Sharpe_ratio = round(c(SR_PFE_2020,
                   SR_JNJ_2020,
                   SR_ABBV_2020,
                   SR_industry_2020,
                   SR_tan_2020),3)
)
rownames(metrics_2020) <- NULL
colnames(metrics_2020) <- c("Asset/Portfolio (2020-2022)", "Mean_daily_R(%)", "Cumu
lative_R(%)", "Std_Dev", "Sharpe_Ratio")
print(metrics_2020)
```

```
##   Asset/Portfolio (2020-2022) Mean_daily_R(%) Cumulative_R(%) Std_Dev
## 1                         PFE           0.075         159.427   1.885
## 2                         JNJ           0.045         132.775   1.374
## 3                        ABBV           0.112         213.864   1.675
## 4                    Industry           0.049         137.823   1.306
## 5           Optimal Portfolio           0.071         160.181   1.332
##   Sharpe_Ratio
## 1        0.038
## 2        0.030
## 3        0.065
## 4        0.034
## 5        0.051
```

The optimal portfolio's mean return and cumulative return are much higher than the industry in both periods, although both have increased from the first to the second period.

With respective to risk, the optimal portfolio has a higher standard deviation than the industry in both periods which aligns with our expectation, given the diversification benefits of the industry series with a large pool of stocks. Risk increased significantly from the first period to the second for both the portfolio and the industry.

In terms of Sharpe ratios, we see the same pattern where the industry series has a moderate Sharpe ratio that is lower than the optimal portfolio. Since the portfolio aims to maximize this metric, its Sharpe ratio surpasses all assets and the industry series in both periods. We also see that this measure for both the industry and the optimal portfolio increased from the first to the second period. One thing to note is that Sharpe ratio decreased in the latter period since the portfolio was optimized in the former period not the second. Our portfolio is no longer the tangency portfolio going into 2020. ABBV as a single stock has a higher Sharpe ratio than the portfolio we previously constructed. This implies that portfolio management and optimization might be better when they are dynamic.

If we were to construct a new pf in the second period, the performance may change, which we will observe from the results below.

```
# Add performance metrics to the data frame.
metrics <- data.frame(
  Asset = c("PFE", "JNJ", "ABBV", "Industry", "Optimal Portfolio"),
  Mean_Return = round(c(mu_PFE, mu_JNJ, mu_ABBV, mu_industry, mu_tan*100),3),
  Cumulative_return = round(c(cumulative_returns[nrow(cumulative_returns),2]*100,
                          cumulative_returns[nrow(cumulative_returns),3]*100,
                          cumulative_returns[nrow(cumulative_returns),4]*100,
                          cumulative_returns[nrow(cumulative_returns),1]*100,
                          pf_cum_returns[nrow(pf_cum_returns)]),3),
  Std_Dev = round(c(sd_PFE, sd_JNJ, sd_ABBV, sd_industry, sd_tan*100),3),
  Sharpe_Ratio = round(c(SR_PFE, SR_JNJ, SR_ABBV, SR_industry, SR_tan),3)
)
rownames(metrics) <- NULL
colnames(metrics) <- c("Asset/Portfolio (2017-2019)", "Mean_daily_R(%)",
                       "Cumulative_R(%)", "Std_Dev", "Sharpe_Ratio")
print(metrics)
```

```
##   Asset/Portfolio (2017-2019) Mean_daily_R(%) Cumulative_R(%) Std_Dev
## 1                         PFE           0.048         136.648   1.074
## 2                         JNJ           0.053         141.718   1.072
## 3                        ABBV           0.082         163.919   1.744
## 4                    Industry           0.042         132.594   0.863
## 5           Optimal Portfolio           0.060         150.577   0.993
##   Sharpe_Ratio
## 1        0.041
## 2        0.046
## 3        0.045
## 4        0.044
## 5        0.057
```

```
# Construct tangency pf for 2020 to 2022
# var-cov matrix
# Compute variance-covariance matrix
mu.vec = c(mean(daily_returns_2020[,2]), mean(daily_returns_2020[,3]),
           mean(daily_returns_2020[,4]))
names(mu.vec) = asset.names
sigma.mat <- cov(daily_returns_2020[, 2:4], use = "complete.obs")
rownames(sigma.mat) <- asset.names
colnames(sigma.mat) <- asset.names
sd.vec = sqrt(diag(sigma.mat))

num = solve(sigma.mat)%*%(mu.vec-rf/100)
den = as.numeric(t(rep(1,3))%*%solve(sigma.mat)%*%(mu.vec-rf/100))
tan.vec_2020 = num/den
mu_tan = as.numeric(crossprod(tan.vec_2020, mu.vec))
sd_tan = sqrt(as.numeric(t(tan.vec_2020)%*%sigma.mat%*%tan.vec_2020))
SR_tan = (mu_tan - rf/100)/sd_tan
cat("Weight of Pfizer (PFE):", round(tan.vec_2020[1],3))
```

```
## Weight of Pfizer (PFE): 0.181
```

```
cat("\nWeight of Johnson and Johnson (JNJ):", round(tan.vec_2020[2],3))
```

```
##
## Weight of Johnson and Johnson (JNJ): -0.308
```

```
cat("\nWeight of AbbVie (ABBV):", round(tan.vec_2020[3],3))
```

```
##
## Weight of AbbVie (ABBV): 1.127
```

```r
# Calculate daily and cumulative return for pf
pf_daily_return_2020<-tan.vec_2020[1]*daily_returns_2020[,2]+
  tan.vec_2020[2]*daily_returns_2020[,3] + tan.vec_2020[3]*daily_returns_2020[,4]
pf_cum_returns_2020 <- cumprod(1 + pf_daily_return_2020)*100

# Calculate mean daily returns
mu_tan_2020 <- mean(pf_daily_return_2020)*100

# Calculate standard deviation of daily returns
sd_tan_2020 <- sd(pf_daily_return_2020*100)

# Calculate Sharpe ratio of daily returns
SR_tan_2020 <- (mu_tan_2020 - rf)/sd_tan_2020


metrics_2020 <- data.frame(
  Asset = c("PFE", "JNJ", "ABBV", "Industry", "Optimal Portfolio"),
  Mean_Return = round(c(mu_PFE_2020, mu_JNJ_2020, mu_ABBV_2020,
                        mu_industry_2020, mu_tan_2020),3),
  Cumulative_return = round(c(cumulative_returns_2020[nrow(cumulative_returns),2],
                        cumulative_returns_2020[nrow(cumulative_returns),3],
                        cumulative_returns_2020[nrow(cumulative_returns),4],
                        cumulative_returns_2020[nrow(cumulative_returns),1],
                        pf_cum_returns_2020[nrow(pf_cum_returns_2020)]),3),
  Std_Dev = round(c(sd_PFE_2020,
              sd_JNJ_2020,
              sd_ABBV_2020,
              sd_industry_2020,
              sd_tan_2020),3),
  Sharpe_ratio = round(c(SR_PFE_2020,
                   SR_JNJ_2020,
                   SR_ABBV_2020,
                   SR_industry_2020,
                   SR_tan_2020),3)
)
rownames(metrics_2020) <- NULL
colnames(metrics_2020) <- c("Asset/Portfolio (2020-2022)", "Mean_daily_R(%)",
                            "Cumulative_R(%)", "Std_Dev", "Sharpe_Ratio")
print(metrics_2020)
```

```
##   Asset/Portfolio (2020-2022) Mean_daily_R(%) Cumulative_R(%) Std_Dev
## 1                         PFE           0.075         159.427   1.885
## 2                         JNJ           0.045         132.775   1.374
## 3                        ABBV           0.112         213.864   1.675
## 4                    Industry           0.049         137.823   1.306
## 5           Optimal Portfolio           0.126         227.305   1.856
##   Sharpe_Ratio
## 1        0.038
## 2        0.030
## 3        0.065
## 4        0.034
## 5        0.066
```

All the trends that we observed previously still holds in both periods. And, since we covered the comparison between the portfolio and the industry series, this section will focus on the Covid period and how the newly constructed portfolio performs against the portfolio from 2017-2019 moving into the new period (2020-2022).

If we were to construct a new portfolio that is optimized given statistics from 2020 to 2022, our portfolio performs much better since it is optimized in that specific period. Our cumulative return is almost the double of the industry series due to high weights on ABBV which did very well during the COVID period. Mean return is the same where the portfolio outperforms even more than the 2017 portfolio. The Sharpe ratio is again higher than industry but is also higher than the previously constructed portfolio in the first sample period. We can only be “optimized” if we continue to update the tangency portfolio. Otherwise, our risk-to-reward ratio falls behind as the tangency portfolio is no longer efficient given new statistics of individual assets.

5. CAPM with respect to SP500 during 2017-2019

The CAPM model is derived as

$$R_{i,t} - r_{f,t} = \alpha_i + \beta_i(\mu_{M,t} - r_{f,t}) + \epsilon_{i,t}$$

Here we take a fixed the return of risk free asset, S&P500 as the market, and construct an tangency-portfolio.

```
library(quantmod)
getSymbols("JNJ", src = "yahoo", from = start_date, to = end_date)
```

```
## [1] "JNJ"
```

```
getSymbols("ABBV", src = "yahoo", from = start_date, to = end_date)
```

```
## [1] "ABBV"
```

```
getSymbols("PFE", src = "yahoo", from = start_date, to = end_date)
```

```
## [1] "PFE"
```

```
getSymbols("^GSPC", src="yahoo", from = start_date, to = end_date)
```

```
## [1] "GSPC"
```

```
getSymbols("^SP500-352020", src = "yahoo", from = start_date, to = end_date)
```

```
## [1] "SP500-352020"
```

```r
getSymbols("^TYX", src="yahoo", from = start_date, to = end_date)
```

```
## [1] "TYX"
```

```r
# daily return of assets
daily_PFE <- dailyReturn(PFE$PFE.Adjusted)
daily_JNJ <- dailyReturn(JNJ$JNJ.Adjusted)
daily_ABBV <- dailyReturn(ABBV$ABBV.Adjusted)

daily_rf <- TYX$TYX.Adjusted/100/365

daily_GSPC <- dailyReturn(GSPC$GSPC.Adjusted)

daily_DRG <- dailyReturn(na.approx(`SP500-352020`[,6]))

daily_pf <- as.numeric(tan.vec)[1] * daily_PFE +
  as.numeric(tan.vec)[2] * daily_JNJ +
  as.numeric(tan.vec)[3] * daily_ABBV

# excess return of three companies
excess_return_PFE <- daily_PFE - daily_rf
excess_return_ABBV <- daily_ABBV - daily_rf
excess_return_JNJ <- daily_JNJ - daily_rf
excess_return_pf <- daily_pf - daily_rf

# excess return of industry
excess_return_DRG <- daily_DRG - daily_rf

# excess return of market
market_return <- daily_GSPC - daily_rf

# linear regression for three companies
CAPM_JNJ <- lm(excess_return_JNJ~market_return)
CAPM_ABBV <- lm(excess_return_ABBV~market_return)
CAPM_PFE <- lm(excess_return_PFE~market_return)

# linear regression for pf
CAPM_pf <- lm(excess_return_pf~market_return)

# linear regression for industry
CAPM_DRG <- lm(excess_return_DRG~market_return)

names <- c("Industry", "Pfizer", "AbbVie", "J&J", "Optimal Portfolio")
alpha <- c(round(as.numeric(CAPM_DRG$coefficients[1]), 3),
           round(as.numeric(CAPM_PFE$coefficients[1]), 3),
           round(as.numeric(CAPM_ABBV$coefficients[1]), 3),
           round(as.numeric(CAPM_JNJ$coefficients[1]), 3),
           round(as.numeric(CAPM_pf$coefficients[1]), 3))
beta <- c(round(as.numeric(CAPM_DRG$coefficients[2]), 3),
          round(as.numeric(CAPM_PFE$coefficients[2]), 3),
          round(as.numeric(CAPM_ABBV$coefficients[2]), 3),
          round(as.numeric(CAPM_JNJ$coefficients[2]), 3),
```

```
          round(as.numeric(CAPM_pf$coefficients[2]), 3))
p_value <- c(round(as.numeric(summary(CAPM_DRG)$coefficients[,4][1]), 3),
             round(as.numeric(summary(CAPM_PFE)$coefficients[,4][1]), 3),
             round(as.numeric(summary(CAPM_ABBV)$coefficients[,4][1]), 3),
             round(as.numeric(summary(CAPM_JNJ)$coefficients[,4][1]), 3),
             round(as.numeric(summary(CAPM_pf)$coefficients[,4][1]), 3))

table <- rbind(names, alpha, beta, p_value)
table
```

```
##         [,1]       [,2]     [,3]     [,4]    [,5]
## names   "Industry" "Pfizer" "AbbVie" "J&J"   "Optimal Portfolio"
## alpha   "0"        "0"      "0"      "0"     "0"
## beta    "0.73"     "0.762"  "1.062"  "0.669" "0.804"
## p_value "0.909"    "0.92"   "0.662"  "0.764" "0.646"
```

From the result we can find out that

- CAPM holds for the industry, the three companies and the optimal-weight-portfolio.
- ABBV has the highest expected beta value of 1.062, and J&J has the lowest expected beta value of 0.669.
- The beta for the industry is 0.73.
- The optimal portfolio has a beta of 0.804, meaning that our constructed portfolio increases as the market increases, but is a little bit slower.

6. CAPM with respect to SP500 during 2020-2022

The CAPM model is derived as

$$R_{i,t} - r_{f,t} = \alpha_i + \beta_i(\mu_{M,t} - r_{f,t}) + \epsilon_{i,t}$$

Here we take a fixed return of risk free asset, S&P500 as the market, and construct an tangency-portfolio.

```
library(quantmod)
start_date_2020 <- "2019-12-31"
end_date_2022 <- "2022-12-31"

getSymbols("JNJ", src = "yahoo", from = start_date_2020, to = end_date_2022)
```

```
## [1] "JNJ"
```

```
getSymbols("ABBV", src = "yahoo", from = start_date_2020, to = end_date_2022)
```

```
## [1] "ABBV"
```

```
getSymbols("PFE", src = "yahoo", from = start_date_2020, to = end_date_2022)
```

```
## [1] "PFE"
```

```
getSymbols("^GSPC", src="yahoo", from = start_date_2020, to = end_date_2022)
```

```
## [1] "GSPC"
```

```
getSymbols("^TYX", src="yahoo", from=start_date_2020, to = end_date_2022)
```

```
## Warning: ^TYX contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "TYX"
```

```
getSymbols("^SP500-352020", src = "yahoo", from = start_date_2020,
           to = end_date_2022)
```

```
## [1] "SP500-352020"
```

```
# daily return of assets
daily_PFE <- dailyReturn(PFE$PFE.Adjusted)
daily_JNJ <- dailyReturn(JNJ$JNJ.Adjusted)
daily_ABBV <- dailyReturn(ABBV$ABBV.Adjusted)

daily_GSPC <- dailyReturn(GSPC$GSPC.Adjusted)

daily_DRG <- dailyReturn(na.approx(`SP500-352020`[,6]))

daily_rf <- TYX$TYX.Adjusted / 100 / 365

daily_pf <- as.numeric(tan.vec)[1] * daily_PFE +
  as.numeric(tan.vec)[2] * daily_JNJ +
  as.numeric(tan.vec)[3] * daily_ABBV

# excess return of three companies
excess_return_PFE <- daily_PFE - daily_rf
excess_return_ABBV <- daily_ABBV - daily_rf
excess_return_JNJ <- daily_JNJ - daily_rf

excess_return_pf <- daily_pf - daily_rf

# excess return of industry
excess_return_DRG <- daily_DRG - daily_rf

# excess return of market
market_return <- daily_GSPC - daily_rf
```

```
# linear regression for three companies
CAPM_JNJ <- lm(excess_return_JNJ~market_return)
CAPM_ABBV <- lm(excess_return_ABBV~market_return)
CAPM_PFE <- lm(excess_return_PFE~market_return)

# linear regression for pf
CAPM_pf <- lm(excess_return_pf~market_return)

# linear regression for industry
CAPM_DRG <- lm(excess_return_DRG~market_return)

names <- c("Industry", "Pfizer", "AbbVie", "J&J", "Optimal Portfolio")
alpha <- c(round(as.numeric(CAPM_DRG$coefficients[1]), 3),
           round(as.numeric(CAPM_PFE$coefficients[1]), 3),
           round(as.numeric(CAPM_ABBV$coefficients[1]), 3),
           round(as.numeric(CAPM_JNJ$coefficients[1]), 3),
           round(as.numeric(CAPM_pf$coefficients[1]), 3))
beta <- c(round(as.numeric(CAPM_DRG$coefficients[2]), 3),
          round(as.numeric(CAPM_PFE$coefficients[2]), 3),
          round(as.numeric(CAPM_ABBV$coefficients[2]), 3),
          round(as.numeric(CAPM_JNJ$coefficients[2]), 3),
          round(as.numeric(CAPM_pf$coefficients[2]), 3))
p_value <- c(round(as.numeric(summary(CAPM_DRG)$coefficients[,4][1]), 3),
             round(as.numeric(summary(CAPM_PFE)$coefficients[,4][1]), 3),
             round(as.numeric(summary(CAPM_ABBV)$coefficients[,4][1]), 3),
             round(as.numeric(summary(CAPM_JNJ)$coefficients[,4][1]), 3),
             round(as.numeric(summary(CAPM_pf)$coefficients[,4][1]), 3))

table <- rbind(names, alpha, beta, p_value)
table
```

```
##         [,1]       [,2]     [,3]     [,4]    [,5]
## names   "Industry" "Pfizer" "AbbVie" "J&J"   "Optimal Portfolio"
## alpha   "0"        "0.001"  "0.001"  "0"     "0"
## beta    "0.576"    "0.574"  "0.592"  "0.539" "0.563"
## p_value "0.449"    "0.389"  "0.079"  "0.556" "0.175"
```

From the result we can find out that

- CAPM holds for the industry, the three companies and the optimal-weight-portfolio.
- ABBV has the highest expected beta value of 0.592, and J&J has the lowest expected beta value of 0.539.
- The industry is thriving with a beta value of 0.576.
- The optimal portfolio has a beta of 0.563, meaning that our constructed portfolio increases as the market increases, but is a little bit slower.
- Compared with 2017-2019, all companies and the optimal pf has lower beta, implying less correlation with the market.

7. VaR and ES

```r
non_parametric_var <- function(returns, p = 0.05) {
  -round(quantile(returns, probs = p, na.rm = TRUE)*100,3)
}

non_parametric_es <- function(returns, p = 0.05) {
  threshold <- quantile(returns, probs = p, na.rm = TRUE)
  -round(mean(returns[returns <= threshold], na.rm = TRUE)*100,3)
}

risk_metrics <- sapply(daily_returns, function(asset_returns) {
  c(
    VaR = non_parametric_var(asset_returns, p = 0.05),
    ES = non_parametric_es(asset_returns, p = 0.05)
  )
})
risk_metrics_df <- as.data.frame(t(risk_metrics))
colnames(risk_metrics_df) <- c("VaR (5%)", "ES (5%)")
risk_metrics_df <- cbind(Asset = rownames(risk_metrics_df), risk_metrics_df)
rownames(risk_metrics_df) <- NULL
print(risk_metrics_df)
```

```
##      Asset VaR (5%) ES (5%)
## 1 Industry    1.402   2.180
## 2      PFE    1.620   2.646
## 3      JNJ    1.476   2.722
## 4     ABBV    2.441   4.110
```

The industry index demonstrates the lowest risk among all assets, with a VaR of -1.40% and an ES of -2.18%. This aligns with the diversification benefits of an index, which aggregates the performance of multiple companies across sectors. Diversification reduces exposure to extreme losses, making the index a more stable investment option during downturns. The relatively lower ES highlights that even during the worst-case scenarios, the industry index mitigates losses more effectively compared to individual stocks.

Among the individual assets, Pfizer (PFE) shows a VaR of -1.62% and an ES of -2.65%, reflecting slightly higher downside risk than the industry but still moderate compared to other companies. Johnson & Johnson (JNJ) has a VaR of -1.48%, similar to Pfizer, but its ES of -2.72% indicates higher average losses during extreme market conditions. These results suggest that while both Pfizer and J&J have relatively stable risk profiles, they still exhibit notable susceptibility to market downturns compared to the broader industry index.

AbbVie (ABBV), with a VaR of -2.44% and an ES of -4.11%, presents the highest downside risk among the analyzed assets. Its higher VaR indicates a greater likelihood of extreme daily losses, while its significantly higher ES reflects the magnitude of average losses on the worst days. AbbVie's volatility and susceptibility to negative market conditions make it a riskier investment option, requiring a higher risk tolerance from investors.

In conclusion, the industry index is the most stable option, with the lowest VaR and ES, making it ideal for risk-averse investors. Pfizer and Johnson & Johnson exhibit moderate risk levels but underperform the index in terms of stability. AbbVie's high VaR and ES highlight its volatile nature and greater exposure to downside risk.

8. VaR and ES

```
risk_metrics <- sapply(daily_returns_2020, function(asset_returns) {
  c(
    VaR = non_parametric_var(asset_returns, p = 0.05),
    ES = non_parametric_es(asset_returns, p = 0.05)
  )
})
risk_metrics_df <- as.data.frame(t(risk_metrics))
colnames(risk_metrics_df) <- c("VaR (5%)", "ES (5%)")
risk_metrics_df <- cbind(Asset = rownames(risk_metrics_df), risk_metrics_df)
rownames(risk_metrics_df) <- NULL
print(risk_metrics_df)
```

```
##      Asset VaR (5%) ES (5%)
## 1 Industry    1.785   2.851
## 2      PFE    2.547   4.043
## 3      JNJ    1.887   3.149
## 4     ABBV    2.217   3.881
```

For the industry index, the VaR increased from -1.40% in 2017–2019 to -1.79% in 2020–2022, while the ES rose from -2.18% to -2.85%. This indicates that the industry experienced greater downside risk and larger average losses during extreme market events in the latter period. The industry index's increased risk metrics reflect the broader market instability during 2020–2022, driven by significant disruptions such as the COVID-19 pandemic. Nevertheless, the industry index remains the least risky asset compared to individual stocks, highlighting the continued benefits of diversification in reducing extreme losses.

Among the individual companies, Pfizer (PFE) exhibited the most significant increase in risk metrics. Its VaR rose from -1.62% to -2.55%, and its ES increased from -2.65% to -4.04%. This sharp increase likely reflects the heightened volatility and market sensitivity surrounding Pfizer during this period, as it played a central role in vaccine development and pandemic-related activities. Similarly, Johnson & Johnson (JNJ) also experienced increased risk, with its VaR rising from -1.48% to -1.89% and its ES from -2.72% to -3.15%. Although less severe than Pfizer's, this increase still underscores the broader trend of elevated market risk during 2020–2022.

Interestingly, AbbVie (ABBV) exhibited a slight improvement in its risk metrics. Its VaR decreased from -2.44% in 2017–2019 to -2.22% in 2020–2022, while its ES fell from -4.11% to -3.88%. Despite this improvement, AbbVie remains the riskiest asset among the individual companies, with the highest downside risk and largest average losses during extreme events. Its marginally better performance could be attributed to improved risk management or operational resilience during the pandemic.

Overall, the comparison highlights the increased volatility and downside risk during 2020–2022 across both the industry and individual assets. The industry index remains the safest option, offering the lowest VaR and ES values due to diversification. In contrast, individual companies, particularly Pfizer, faced heightened risks, reflecting their sensitivity to market-specific events.

9. Conclusion

The results from this analysis highlight some clear differences between the two time periods (2017–2019 vs. 2020–2022). The three individual asset series and the optimized portfolio both performed significantly better during 2020–2022 compared to 2017–2019. This is largely due to the pharmaceutical industry's unique position during the pandemic.

During the COVID-19 pandemic, companies like Pfizer and Johnson & Johnson became critical to global recovery efforts, driving unprecedented growth in the sector. AbbVie, while not directly involved in vaccine production, still benefited from the sector's overall momentum due to its strong immunology portfolio. The optimized portfolio successfully captured this growth while maintaining diversification, showing the advantages of applying portfolio optimization techniques during volatile periods.

In terms of market beta, 2020–2022 showed increased sensitivity to the market compared to the more stable relationship in 2017–2019. This shift is likely because healthcare stocks, traditionally considered defensive, turned into high-growth assets during the pandemic due to their central role in vaccine development and distribution. This change demonstrates the need to adapt investment strategies based on shifting sector dynamics.

Lastly, the tail risk analysis shows that 2020–2022 carried less downside risk for the portfolio compared to 2017–2019. While the pandemic initially introduced significant uncertainty, the pharmaceutical sector's resilience and eventual strong performance helped reduce extreme downside risk, especially for the efficient portfolio.

Overall, this analysis shows the importance of understanding sector-specific trends, especially during unprecedented events like the COVID-19 pandemic. The pharmaceutical industry's role as both a defensive and growth driver during this time highlights why diversification and timely portfolio adjustments are critical for managing risk and capturing opportunities.