Hybrid & Technical Strategy Backtest, Optimizations, and Comparision.

Ji

2025-04-06

1. Load data,also cleaning

```
symbols <- c("NVDA", "SOXL", "XOM", "CLS.TO")
standard_cols <- as.character(c("Date", symbols))

library(readr)
library(openxlsx)

p_risk <- read_csv("daily_prisk_signals.csv")
```

```
## Rows: 3908 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (4): CLS.TO, SOXL, NVDA, XOM
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
p_ma_3_12 <- read_csv("final_PMA_3month&12month.csv")  # MA3/12
```

```
## Rows: 143 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (4): NVDA, SOXL, XOM, CLS.TO
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
p_ma_12_36 <- read_csv("final_PMA_12month&36month.csv")  # MA12/36
```

```
## Rows: 143 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (4): NVDA, SOXL, XOM, CLS.TO
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
p_econ <- read.xlsx("Optimized_P_Econ.xlsx")
monthly_close <- read_csv("monthly_close.csv")
```

```
## Rows: 143 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (4): NVDA, SOXL, XOM, CLS.TO
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
if(!("Date" %in% colnames(p_econ))) {
  p_econ$Date <- seq(as.Date("2010-01-01"), by = "month", length.out = nrow(p_econ))
}

p_risk <- p_risk[, as.character(intersect(colnames(p_risk), standard_cols))]
p_ma_3_12 <- p_ma_3_12[, as.character(intersect(colnames(p_ma_3_12), standard_cols))]
p_ma_12_36 <- p_ma_12_36[, as.character(intersect(colnames(p_ma_12_36), standard_cols))]
p_econ <- p_econ[, c("Date", symbols)]
monthly_close <- monthly_close[, c("Date", symbols)]


p_risk$Date <- as.Date(p_risk$Date)
p_ma_3_12$Date <- as.Date(p_ma_3_12$Date)
p_ma_12_36$Date <- as.Date(p_ma_12_36$Date)
p_econ$Date <- as.Date(p_econ$Date)
monthly_close$Date <- as.Date(monthly_close$Date)
```

2. p_risk: Convert daily to monthly average

```
# p_risk: daily → monthly
p_risk_monthly <- p_risk %>%
  mutate(YearMonth = floor_date(Date, "month")) %>%
  group_by(YearMonth) %>%
  summarise(across(all_of(symbols), ~mean(.x, na.rm = TRUE))) %>%
  rename(Date = YearMonth) %>%
  arrange(Date)

cat("\n p_risk_monthly range: [", min(p_risk_monthly$Date), "~", max(p_risk_monthly$Date), "]\n")
```

```
##
##  p_risk_monthly range: [ 14610 ~ 20179 ]
```

3. Check other datasets

```
cat("p_ma_3_12 range: [", min(p_ma_3_12$Date), "~", max(p_ma_3_12$Date), "]\n")
```

```
## p_ma_3_12 range: [ 15737 ~ 20058 ]
```

```
cat("p_ma_12_36 range: [", min(p_ma_12_36$Date), "~", max(p_ma_12_36$Date), "]\n")
```

```
## p_ma_12_36 range: [ 15737 ~ 20058 ]
```

```
cat("p_econ range:     [", min(p_econ$Date), "~", max(p_econ$Date), "]\n")
```

```
## p_econ range:     [ 14610 ~ 20058 ]
```

```
cat("monthly_close:    [", min(monthly_close$Date), "~", max(monthly_close$Date), "]\n")
```

```
## monthly_close:    [ 15737 ~ 20058 ]
```

4. Find common monthly dates

```
common_dates <- Reduce(intersect, list(
  as.character(p_risk_monthly$Date),
  as.character(p_ma_3_12$Date),
  as.character(p_ma_12_36$Date),
  as.character(p_econ$Date),
  as.character(monthly_close$Date)
))
common_dates <- as.Date(common_dates)

cat("\n common datas:", length(common_dates), "\n")
```

```
##
##  common datas: 143
```

```
print(head(common_dates))
```

```
## [1] "2013-02-01" "2013-03-01" "2013-04-01" "2013-05-01" "2013-06-01"
## [6] "2013-07-01"
```

5. Final synchronization

```
sync_and_sort <- function(df){
  df <- df[df$Date %in% common_dates, ]
  df[order(df$Date), ]
}

p_risk_sync     <- sync_and_sort(p_risk_monthly)
p_ma_3_12_sync  <- sync_and_sort(p_ma_3_12)
p_ma_12_36_sync <- sync_and_sort(p_ma_12_36)
p_econ_sync     <- sync_and_sort(p_econ)
monthly_close   <- sync_and_sort(monthly_close)

range_summary <- function(df) {
  paste0("[", min(df$Date, na.rm=TRUE), " ~ ", max(df$Date, na.rm=TRUE), "]")
}

cat("p_risk_sync:     ", range_summary(p_risk_sync), "\n")
```

```
## p_risk_sync:      [2013-02-01 ~ 2024-12-01]
```

```
cat("p_ma_3_12_sync:  ", range_summary(p_ma_3_12_sync), "\n")
```

```
## p_ma_3_12_sync:   [2013-02-01 ~ 2024-12-01]
```

```
cat("p_ma_12_36_sync: ", range_summary(p_ma_12_36_sync), "\n")
```

```
## p_ma_12_36_sync:  [2013-02-01 ~ 2024-12-01]
```

```
cat("p_econ_sync:     ", range_summary(p_econ_sync), "\n")
```

```
## p_econ_sync:      [2013-02-01 ~ 2024-12-01]
```

```
cat("monthly_close:   ", range_summary(monthly_close), "\n")
```

```
## monthly_close:    [2013-02-01 ~ 2024-12-01]
```

```
cat("\nFinal synchronization done. Proceeding to calculate f_technical and f_hybrid, then backtesting..
```

```
##
## Final synchronization done. Proceeding to calculate f_technical and f_hybrid, then backtesting...
```

```
w1 <- 0.5
w2 <- 0.5

f_technical_df <- p_ma_3_12_sync
for(sym in symbols){
  f_technical_df[[sym]] <- w1 * p_ma_3_12_sync[[sym]] + w2 * p_risk_sync[[sym]]
}
names(f_technical_df)[-1] <- paste0("f_tech_", symbols)

alpha <- 0.7
f_hybrid_df <- f_technical_df[, "Date", drop=FALSE]
for(sym in symbols){
  ft_sym <- f_technical_df[[paste0("f_tech_", sym)]]
  econ_sym <- p_econ_sync[[sym]]
  # hybrid = alpha * f_tech + (1-alpha)*p_econ
  fh_sym <- alpha * ft_sym + (1-alpha)*econ_sym
  f_hybrid_df[[paste0("f_hyb_", sym)]] <- fh_sym
}

cat(" f_technical & f_hybrid computed!\n")
```

```
##  f_technical & f_hybrid computed!
```

```
monthly_close <- monthly_close %>% arrange(Date)

logret_df <- monthly_close[, c("Date")]
for(sym in symbols){
  # Log return
  ret <- c(NA, diff(log(monthly_close[[sym]])))
  logret_df[[paste0("logret_",sym)]] <- ret
}

cat(" log returns computed!\n")
```

```
##  log returns computed!
```

```
head(logret_df)
```

```
## # A tibble: 6 x 5
##   Date       logret_NVDA logret_SOXL logret_XOM logret_CLS.TO
##   <date>           <dbl>       <dbl>      <dbl>         <dbl>
## 1 2013-02-01   NA            NA        NA            NA
## 2 2013-03-01   -0.00210      -0.127     0.0255       -0.0224
```

```
## 3 2013-04-01    0.0142       -0.0456   -0.00338       0.0444
## 4 2013-05-01    0.000632     -0.138     0.00266       0.0337
## 5 2013-06-01    0.00117       0.0674    0.0177       -0.118
## 6 2013-07-01    0.0153        0.181     0.0158        0.00248
```

ˆThese process are basically just for having Consistent data before we testing, optimizing and comparing. Since everyone have a different timeline, different column name, especailly for risk, risk has to be optimized based on daily since otherwise there are not many chance to trade signal, so instead I make it daily, and make it to normalized in monthly. and since we calculated different timeline, i made a logic that have intersect timeline.

now for backtesting, this is randomly decided parameters.

```r
library(PerformanceAnalytics)
library(xts)
library(ggplot2)

# Common parameters
threshold <- 0.6
symbols <- c("NVDA", "SOXL", "XOM", "CLS.TO")

# Define strategy backtest function
backtest_strategy <- function(sym, strategy_df, strategy_type) {
  f_col <- paste0(ifelse(strategy_type == "technical", "f_tech_", "f_hyb_"), sym)
  ret_col <- paste0("logret_", sym)

  df_test <- merge(strategy_df[, c("Date", f_col)],
                   logret_df[, c("Date", ret_col)],
                   by = "Date") |> arrange(Date)

  df_test$signal <- ifelse(df_test[[f_col]] > threshold, 1,
                           ifelse(df_test[[f_col]] < (1 - threshold), -1, 0))
  df_test$position <- dplyr::lag(df_test$signal, 1, default = 0)

  df_test$strat_ret <- df_test$position * df_test[[ret_col]]
  df_test$cum_ret <- cumsum(ifelse(is.na(df_test$strat_ret), 0, df_test$strat_ret))

  # Performance metrics
  mu <- mean(df_test$strat_ret, na.rm = TRUE)
  sd_ <- sd(df_test$strat_ret, na.rm = TRUE)
  sharpe_ <- ifelse(sd_ == 0, NA, mu / sd_)

  wins <- sum(df_test$strat_ret > 0, na.rm = TRUE)
  trades <- sum(!is.na(df_test$strat_ret))
  winrate_ <- ifelse(trades == 0, NA, wins / trades)

  equity_curve <- exp(df_test$cum_ret)
  equity_xts <- xts(equity_curve, order.by = as.Date(df_test$Date))
  max_dd <- PerformanceAnalytics::maxDrawdown(equity_xts)

  # Print result
  cat("\n=== Backtest:", sym, "-", strategy_type, "===\n")
  cat("Sharpe Ratio:", round(sharpe_, 4),
      " | Win Rate:", round(winrate_, 4),
      " | Max Drawdown:", round(max_dd, 4), "\n")
```

```r
  # Plot cumulative log return
  ggplot(df_test, aes(x = Date, y = cum_ret)) +
    geom_line(color = ifelse(strategy_type == "technical", "darkgreen", "blue"), linewidth = 1) +
    labs(title = paste0(sym, " - ", strategy_type, " Strategy"),
         x = NULL, y = "Cumulative Log Return") +
    theme_minimal()
}

# Run the loop for all symbols
for (sym in symbols) {
  print(backtest_strategy(sym, f_technical_df, "technical"))
  print(backtest_strategy(sym, f_hybrid_df, "hybrid"))
}
```

```
##
## === Backtest: NVDA - technical ===
## Sharpe Ratio: 0.0752  | Win Rate: 0.2324  | Max Drawdown: 0
```


```
##
## === Backtest: NVDA - hybrid ===
## Sharpe Ratio: 0.0046  | Win Rate: 0.1338  | Max Drawdown: 0
```


```
##
## === Backtest: SOXL - technical ===
## Sharpe Ratio: -0.0601  | Win Rate: 0.162  | Max Drawdown: 0
```


```
##
```

```
## === Backtest: SOXL - hybrid ===
## Sharpe Ratio: 0.041  | Win Rate: 0.1268  | Max Drawdown: 0
```


```
##
## === Backtest: XOM - technical ===
## Sharpe Ratio: 0.039  | Win Rate: 0.2606  | Max Drawdown: 0
```


```
##
## === Backtest: XOM - hybrid ===
## Sharpe Ratio: -0.0014  | Win Rate: 0.1549  | Max Drawdown: 0
```


```
##
```

```
## === Backtest: CLS.TO - technical ===
## Sharpe Ratio: -0.0076  | Win Rate: 0.1831  | Max Drawdown: 0
```


```
##
## === Backtest: CLS.TO - hybrid ===
## Sharpe Ratio: 0.1207  | Win Rate: 0.2042  | Max Drawdown: 0
```

Anyway, we can optimize based on grid search. for threshold and wieghts based on alpha, which is considering both winrate and sharp ratio.

```
optimize_threshold <- function(f_signal_df, logret_df, sym, strategy = "hybrid",
                               threshold_grid = seq(0.4, 0.7, by = 0.01), alpha = 0.7) {
  library(xts)
  library(PerformanceAnalytics)

  f_col <- if (strategy == "hybrid") paste0("f_hyb_", sym) else paste0("f_tech_", sym)
  ret_col <- paste0("logret_", sym)

  df <- merge(f_signal_df[, c("Date", f_col)],
              logret_df[, c("Date", ret_col)],
              by = "Date") |> arrange(Date)

  best_score <- -Inf
  best_thresh <- NA
  results <- data.frame()

  for (thresh in threshold_grid) {
    df$signal <- ifelse(df[[f_col]] > thresh, 1,
                        ifelse(df[[f_col]] < (1 - thresh), -1, 0))
    df$position <- dplyr::lag(df$signal, 1, default = 0)
    df$strat_ret <- df$position * df[[ret_col]]
    mu <- mean(df$strat_ret, na.rm = TRUE)
    sd_ <- sd(df$strat_ret, na.rm = TRUE)
    sharpe <- ifelse(sd_ == 0, NA, mu / sd_)
    wins <- sum(df$strat_ret > 0, na.rm = TRUE)
    trades <- sum(!is.na(df$strat_ret))
```

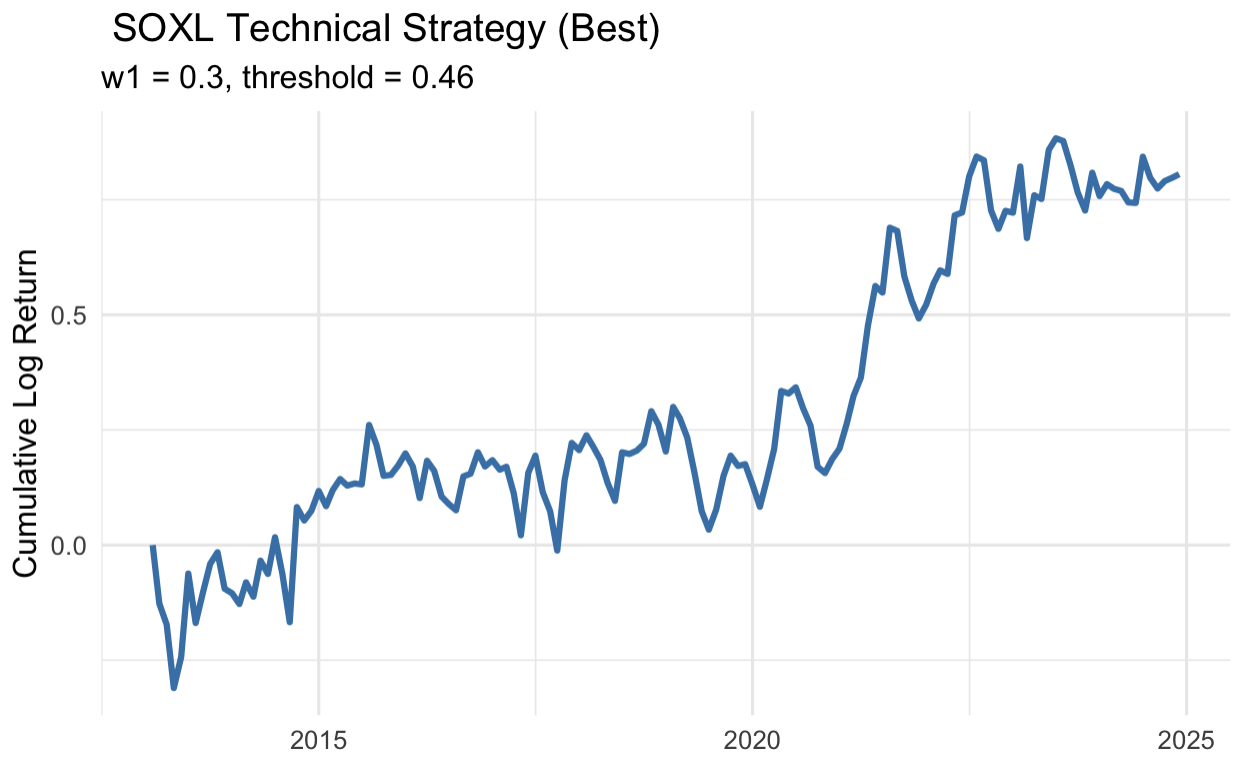
```
    winrate <- ifelse(trades == 0, NA, wins / trades)
    score <- 0.7 * sharpe + 0.3 * winrate

    results <- rbind(results, data.frame(threshold = thresh, Sharpe = sharpe,
                                         WinRate = winrate, Score = score))
    if (!is.na(score) && score > best_score) {
      best_score <- score
      best_thresh <- thresh
    }
  }

  list(best_threshold = best_thresh,
       best_score = best_score,
       full_results = results)
}
```

```
library(ggplot2)
library(xts)
library(PerformanceAnalytics)

symbols <- c("NVDA", "SOXL", "XOM", "CLS.TO")
threshold_grid <- seq(0.4, 0.7, by = 0.02)
w1_grid <- seq(0.1, 0.9, by = 0.1)

global_results <- list()

for (sym in symbols) {
  cat("\n Optimizing", sym, "...\n")
  results <- data.frame()

  for (w1 in w1_grid) {
    w2 <- 1 - w1

    f_tech_test <- p_ma_3_12_sync
    f_tech_test[symbols] <- w1 * p_ma_3_12_sync[symbols] + w2 * p_risk_sync[symbols]
    names(f_tech_test)[-1] <- paste0("f_tech_", symbols)

    for (threshold in threshold_grid) {
      res <- optimize_threshold(
        f_signal_df = f_tech_test,
        logret_df = logret_df,
        sym = sym,
        strategy = "technical",
        threshold_grid = threshold,
        alpha = 0.7
      )

      results <- rbind(results, data.frame(
        Symbol = sym,
        w1 = w1,
        w2 = w2,
        threshold = threshold,
        Score = res$best_score,
        Sharpe = res$full_results$Sharpe[1],
```

```
        WinRate = res$full_results$WinRate[1]
      ))
    }
  }

  global_results[[sym]] <- results

  #  Extract best parameter combination
  best_row <- results[which.max(results$Score), ]
  cat("", sym, ": Best Result\n")
  print(best_row)

  #  Generate signal using best parameters
  w1_best <- best_row$w1
  w2_best <- best_row$w2
  threshold_best <- best_row$threshold

  f_best <- p_ma_3_12_sync
  f_best[symbols] <- w1_best * p_ma_3_12_sync[symbols] + w2_best * p_risk_sync[symbols]
  names(f_best)[-1] <- paste0("f_tech_", symbols)

  #  Log return
  f_col <- paste0("f_tech_", sym)
  ret_col <- paste0("logret_", sym)

  df_plot <- merge(f_best[, c("Date", f_col)],
                   logret_df[, c("Date", ret_col)], by = "Date") |>
    arrange(Date)

  df_plot$signal <- ifelse(df_plot[[f_col]] > threshold_best, 1,
                           ifelse(df_plot[[f_col]] < (1 - threshold_best), -1, 0))
  df_plot$position <- dplyr::lag(df_plot$signal, 1, default = 0)
  df_plot$strat_ret <- df_plot$position * df_plot[[ret_col]]
  df_plot$cum_ret <- cumsum(ifelse(is.na(df_plot$strat_ret), 0, df_plot$strat_ret))

  #  Plot cumulative return graph
  p <- ggplot(df_plot, aes(x = Date, y = cum_ret)) +
    geom_line(color = "steelblue", linewidth = 1) +
    labs(title = paste0(" ", sym, " Technical Strategy (Best)"),
         subtitle = paste0("w1 = ", w1_best, ", threshold = ", threshold_best),
         x = NULL, y = "Cumulative Log Return") +
    theme_minimal()

  print(p)
}
```

```
##
##  Optimizing NVDA ...
##  NVDA : Best Result
##    Symbol  w1  w2 threshold     Score    Sharpe   WinRate
## 31   NVDA 0.2 0.8      0.68 0.2211374 0.1861319 0.3028169
```


```
##
##  Optimizing SOXL ...
##  SOXL : Best Result
##    Symbol  w1  w2 threshold     Score    Sharpe   WinRate
## 36   SOXL 0.3 0.7      0.46 0.2034184 0.0853663 0.4788732
```




```
##
##  Optimizing XOM ...
##  XOM : Best Result
##    Symbol  w1  w2 threshold     Score    Sharpe   WinRate
## 18    XOM 0.2 0.8      0.42 0.2933952 0.1625968 0.5985915
```


```
##
##  Optimizing CLS.TO ...
##  CLS.TO : Best Result
##     Symbol  w1  w2 threshold     Score   Sharpe   WinRate
## 130 CLS.TO 0.9 0.1      0.42 0.2720905 0.144234 0.5704225
```


```
library(ggplot2)
library(xts)
library(PerformanceAnalytics)

symbols <- c("NVDA", "SOXL", "XOM", "CLS.TO")

threshold_grid <- seq(0.4, 0.7, by = 0.02)
w1_grid <- seq(0.1, 0.9, by = 0.1)      # P_MA
alpha_grid <- seq(0.4, 0.9, by = 0.1)   # Hybrid

global_results_hyb <- list()

for (sym in symbols) {
  cat("\n Optimizing Hybrid Strategy for", sym, "...\n")
  results <- data.frame()

  for (w1 in w1_grid) {
    w2 <- 1 - w1

    # f_technical
    f_tech_temp <- p_ma_3_12_sync
    f_tech_temp[symbols] <- w1 * p_ma_3_12_sync[symbols] + w2 * p_risk_sync[symbols]
    names(f_tech_temp)[-1] <- paste0("f_tech_", symbols)

    for (alpha in alpha_grid) {
      # f_hybrid
      f_hyb_temp <- f_tech_temp[, "Date", drop=FALSE]
      for (s in symbols) {
        f_hyb_temp[[paste0("f_hyb_", s)]] <-
```

```
        alpha * f_tech_temp[[paste0("f_tech_", s)]] +
        (1 - alpha) * p_econ_sync[[s]]
    }

    for (threshold in threshold_grid) {
      res <- optimize_threshold(
        f_signal_df = f_hyb_temp,
        logret_df = logret_df,
        sym = sym,
        strategy = "hybrid",
        threshold_grid = threshold,
        alpha = alpha
      )

      results <- rbind(results, data.frame(
        Symbol = sym,
        w1 = w1,
        w2 = w2,
        alpha = alpha,
        threshold = threshold,
        Score = res$best_score,
        Sharpe = res$full_results$Sharpe[1],
        WinRate = res$full_results$WinRate[1]
      ))
    }
  }
}

global_results_hyb[[sym]] <- results

#  Extract and print optimal results
best_row <- results[which.max(results$Score), ]
cat("", sym, ": Best Hybrid Result\n")
print(best_row)

#   f_technical + f_hybrid
w1_best <- best_row$w1
w2_best <- best_row$w2
alpha_best <- best_row$alpha
threshold_best <- best_row$threshold

f_tech_best <- p_ma_3_12_sync
f_tech_best[symbols] <- w1_best * p_ma_3_12_sync[symbols] + w2_best * p_risk_sync[symbols]
names(f_tech_best)[-1] <- paste0("f_tech_", symbols)

f_hyb_best <- f_tech_best[, "Date", drop=FALSE]
for (s in symbols) {
  f_hyb_best[[paste0("f_hyb_", s)]] <-
    alpha_best * f_tech_best[[paste0("f_tech_", s)]] +
    (1 - alpha_best) * p_econ_sync[[s]]
}

f_col <- paste0("f_hyb_", sym)
```

```
  ret_col <- paste0("logret_", sym)

  df_plot <- merge(f_hyb_best[, c("Date", f_col)],
                   logret_df[, c("Date", ret_col)], by = "Date") |>
    arrange(Date)

  df_plot$signal <- ifelse(df_plot[[f_col]] > threshold_best, 1,
                           ifelse(df_plot[[f_col]] < (1 - threshold_best), -1, 0))
  df_plot$position <- dplyr::lag(df_plot$signal, 1, default = 0)
  df_plot$strat_ret <- df_plot$position * df_plot[[ret_col]]
  df_plot$cum_ret <- cumsum(ifelse(is.na(df_plot$strat_ret), 0, df_plot$strat_ret))

  #  Visualize cumulative return
  p <- ggplot(df_plot, aes(x = Date, y = cum_ret)) +
    geom_line(color = "darkgreen", linewidth = 1) +
    labs(title = paste0(" ", sym, " Hybrid Strategy (Best)"),
         subtitle = paste0("w1 = ", w1_best,
                           ",  = ", alpha_best,
                           ", threshold = ", threshold_best),
         x = NULL, y = "Cumulative Log Return") +
    theme_minimal()

  print(p)
}
```

```
##
##  Optimizing Hybrid Strategy for NVDA ...
##  NVDA : Best Hybrid Result
##    Symbol  w1  w2 alpha threshold     Score    Sharpe   WinRate
## 95   NVDA 0.1 0.9   0.9      0.68 0.2121874 0.1612738 0.3309859
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.1,  = 0.9, threshold = 0.68' in 'mbcsToSbcs':
## dot substituted for <ce>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.1,  = 0.9, threshold = 0.68' in 'mbcsToSbcs':
## dot substituted for <b1>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.1,  = 0.9, threshold = 0.68' in 'mbcsToSbcs':
## dot substituted for <ce>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.1,  = 0.9, threshold = 0.68' in 'mbcsToSbcs':
## dot substituted for <b1>
```


```
##
##  Optimizing Hybrid Strategy for SOXL ...
##  SOXL : Best Hybrid Result
##     Symbol  w1  w2 alpha threshold     Score    Sharpe   WinRate
## 117   SOXL 0.2 0.8   0.5      0.48 0.2225263 0.1066271 0.4929577
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.2,  = 0.5, threshold = 0.48' in 'mbcsToSbcs':
## dot substituted for <ce>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.2,  = 0.5, threshold = 0.48' in 'mbcsToSbcs':
## dot substituted for <b1>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.2,  = 0.5, threshold = 0.48' in 'mbcsToSbcs':
## dot substituted for <ce>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.2,  = 0.5, threshold = 0.48' in 'mbcsToSbcs':
## dot substituted for <b1>
```


```
##
##  Optimizing Hybrid Strategy for XOM ...
##  XOM : Best Hybrid Result
##    Symbol  w1  w2 alpha threshold     Score    Sharpe   WinRate
## 83    XOM 0.1 0.9   0.9      0.44 0.2799889 0.1464631 0.5915493

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.1,  = 0.9, threshold = 0.44' in 'mbcsToSbcs':
## dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.1,  = 0.9, threshold = 0.44' in 'mbcsToSbcs':
## dot substituted for <b1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.1,  = 0.9, threshold = 0.44' in 'mbcsToSbcs':
## dot substituted for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.1,  = 0.9, threshold = 0.44' in 'mbcsToSbcs':
## dot substituted for <b1>
```


```
##
##  Optimizing Hybrid Strategy for CLS.TO ...
##  CLS.TO : Best Hybrid Result
##     Symbol  w1  w2 alpha threshold   Score    Sharpe   WinRate
## 834 CLS.TO 0.9 0.1   0.8      0.42 0.28223 0.1496646 0.5915493
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.9,  = 0.8, threshold = 0.42' in 'mbcsToSbcs':
## dot substituted for <ce>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.9,  = 0.8, threshold = 0.42' in 'mbcsToSbcs':
## dot substituted for <b1>
```

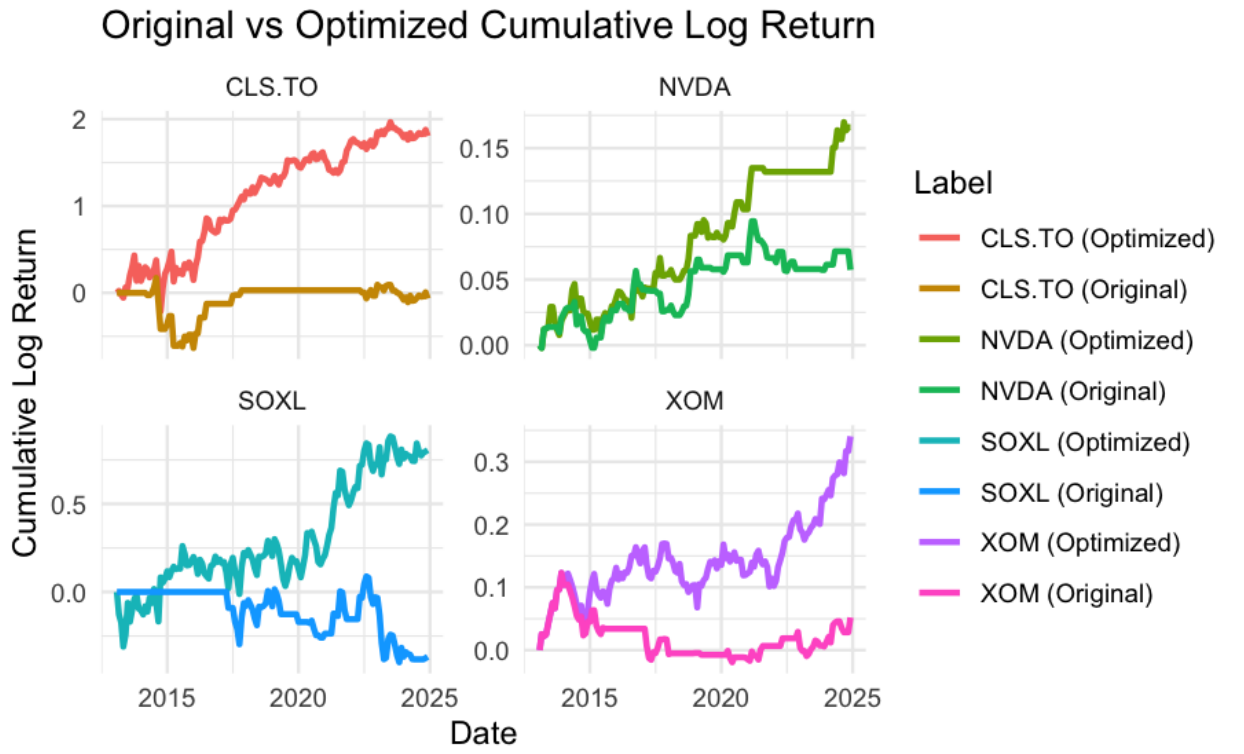
```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.9,  = 0.8, threshold = 0.42' in 'mbcsToSbcs':
## dot substituted for <ce>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'w1 = 0.9,  = 0.8, threshold = 0.42' in 'mbcsToSbcs':
## dot substituted for <b1>
```


```
library(dplyr)
library(ggplot2)
library(xts)
library(PerformanceAnalytics)

symbols <- c("NVDA", "SOXL", "XOM", "CLS.TO")

# Store original backtest results
original_results <- data.frame()
optimized_results <- data.frame()

# Visualization
cumrets_all <- list()

for (sym in symbols) {
  # threshold = 0.6
  f_col <- paste0("f_tech_", sym)
  ret_col <- paste0("logret_", sym)

  df_orig <- merge(f_technical_df[, c("Date", f_col)],
                   logret_df[, c("Date", ret_col)], by = "Date") |> arrange(Date)

  df_orig$signal <- ifelse(df_orig[[f_col]] > 0.6, 1,
                      ifelse(df_orig[[f_col]] < 0.4, -1, 0))
  df_orig$position <- dplyr::lag(df_orig$signal, 1, default = 0)
  df_orig$strat_ret <- df_orig$position * df_orig[[ret_col]]
  df_orig$cum_ret <- cumsum(ifelse(is.na(df_orig$strat_ret), 0, df_orig$strat_ret))

  sharpe_orig <- mean(df_orig$strat_ret, na.rm = TRUE) / sd(df_orig$strat_ret, na.rm = TRUE)
```

```
  winrate_orig <- sum(df_orig$strat_ret > 0, na.rm = TRUE) / sum(!is.na(df_orig$strat_ret))

  original_results <- rbind(original_results, data.frame(
    Symbol = sym,
    Strategy = "technical",
    Version = "Original",
    Sharpe = sharpe_orig,
    WinRate = winrate_orig
  ))

  cumrets_all[[paste0(sym, "_orig")]] <- data.frame(Date = df_orig$Date,
                                                    cum_ret = df_orig$cum_ret,
                                                    Label = paste0(sym, " (Original)"))

  #Backtest After Optimization
  best_row <- global_results[[sym]] |> filter(Score == max(Score))
  w1_best <- best_row$w1
  w2_best <- best_row$w2
  threshold_best <- best_row$threshold

  f_best <- p_ma_3_12_sync
  f_best[symbols] <- w1_best * p_ma_3_12_sync[symbols] + w2_best * p_risk_sync[symbols]
  names(f_best)[-1] <- paste0("f_tech_", symbols)

  df_opt <- merge(f_best[, c("Date", f_col)],
                  logret_df[, c("Date", ret_col)], by = "Date") |> arrange(Date)

  df_opt$signal <- ifelse(df_opt[[f_col]] > threshold_best, 1,
                     ifelse(df_opt[[f_col]] < (1 - threshold_best), -1, 0))
  df_opt$position <- dplyr::lag(df_opt$signal, 1, default = 0)
  df_opt$strat_ret <- df_opt$position * df_opt[[ret_col]]
  df_opt$cum_ret <- cumsum(ifelse(is.na(df_opt$strat_ret), 0, df_opt$strat_ret))

  sharpe_opt <- mean(df_opt$strat_ret, na.rm = TRUE) / sd(df_opt$strat_ret, na.rm = TRUE)
  winrate_opt <- sum(df_opt$strat_ret > 0, na.rm = TRUE) / sum(!is.na(df_opt$strat_ret))

  optimized_results <- rbind(optimized_results, data.frame(
    Symbol = sym,
    Strategy = "technical",
    Version = "Optimized",
    Sharpe = sharpe_opt,
    WinRate = winrate_opt
  ))

  cumrets_all[[paste0(sym, "_opt")]] <- data.frame(Date = df_opt$Date,
                                                   cum_ret = df_opt$cum_ret,
                                                   Label = paste0(sym, " (Optimized)"))
}

#  Compare Cumulative Return Graphs
cumrets_plot_df <- bind_rows(cumrets_all)
ggplot(cumrets_plot_df, aes(x = Date, y = cum_ret, color = Label)) +
  geom_line(linewidth = 1) +
```

```
  facet_wrap(~ gsub(" \\(.*\\)", "", Label), scales = "free_y")+
  labs(title = "Original vs Optimized Cumulative Log Return", y = "Cumulative Log Return") +
  theme_minimal()
```



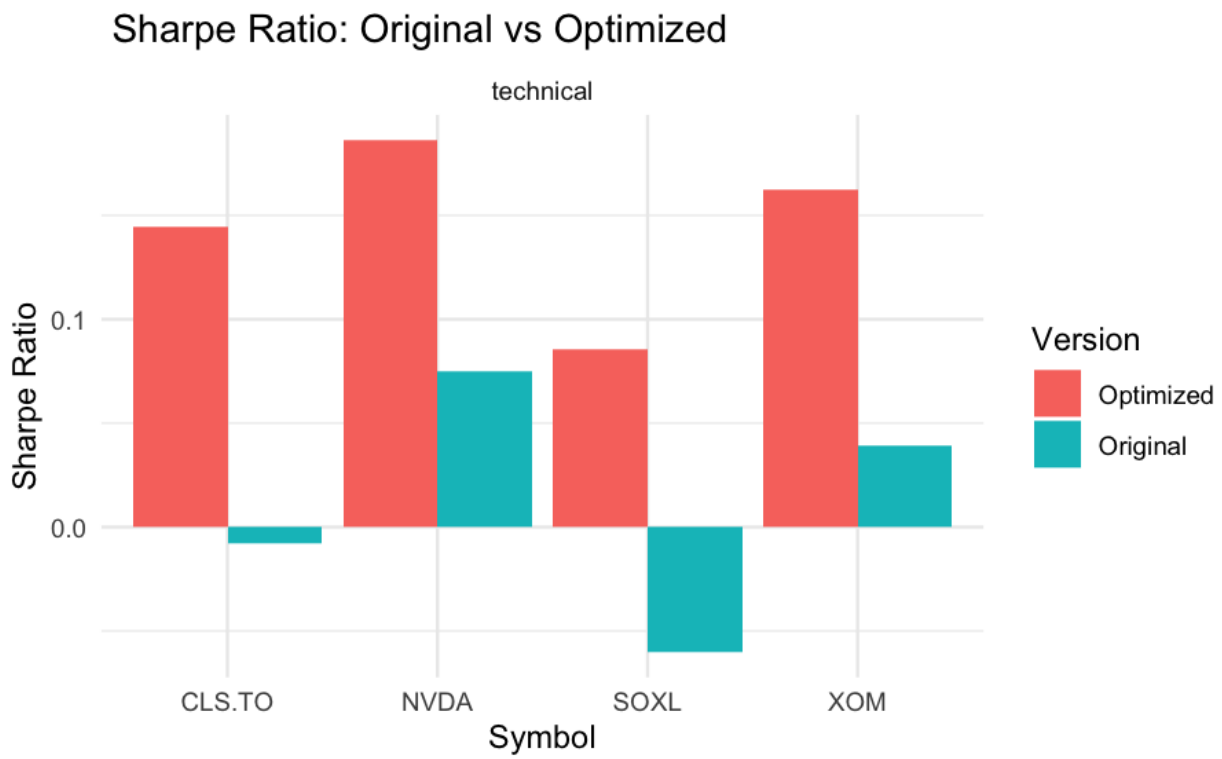

```
#  Compare Performance Metrics
perf_compare <- bind_rows(original_results, optimized_results)
print(perf_compare |> tidyr::pivot_wider(names_from = Version, values_from = c(Sharpe, WinRate)))
```

```
## # A tibble: 4 x 6
##   Symbol Strategy  Sharpe_Original Sharpe_Optimized WinRate_Original
##   <chr>  <chr>               <dbl>            <dbl>            <dbl>
## 1 NVDA   technical          0.0752            0.186            0.232
## 2 SOXL   technical         -0.0601            0.0854           0.162
## 3 XOM    technical          0.0390            0.163            0.261
## 4 CLS.TO technical         -0.00758           0.144            0.183
## # i 1 more variable: WinRate_Optimized <dbl>
```

we can see that optimized one is signifcantly got better in log return, but this deosnt fully explain that this algorithm is good, so lets check others.

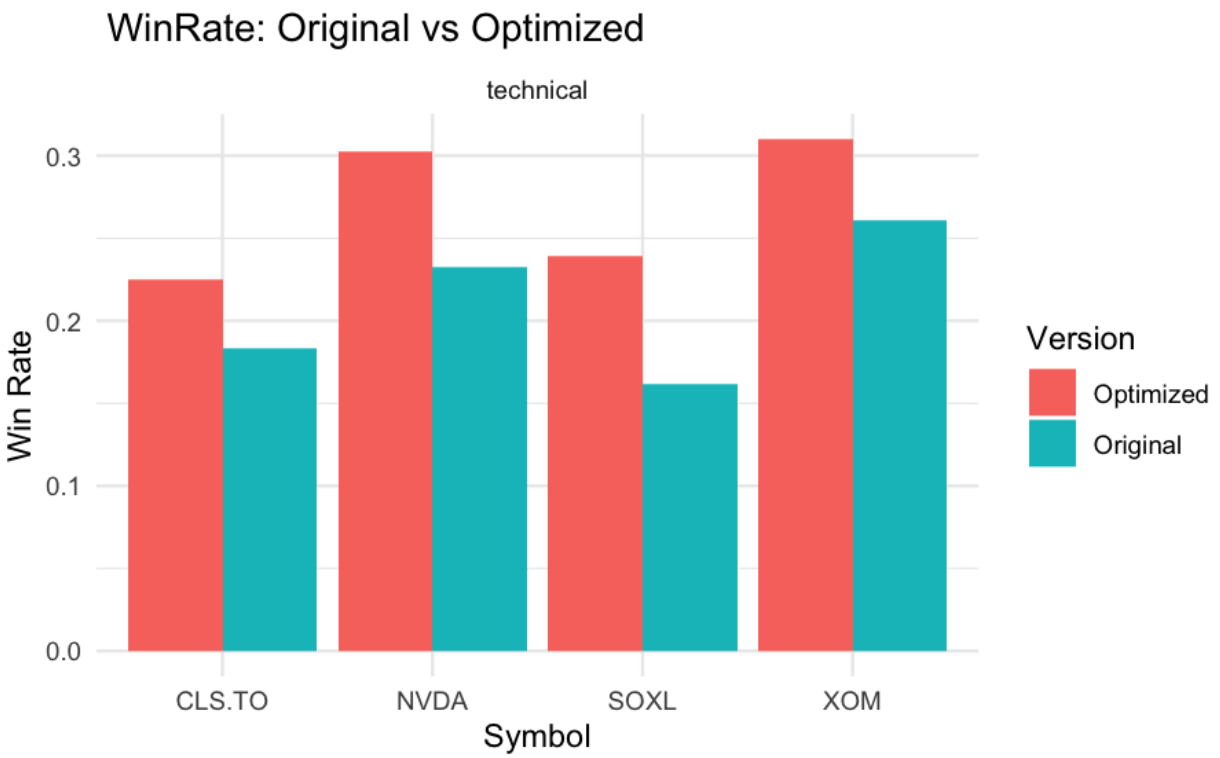
```
library(ggplot2)
library(tidyr)
library(dplyr)
performance_summary_df <- data.frame(
  Symbol = c("NVDA", "SOXL", "XOM", "CLS.TO"),
  Strategy = rep("technical", 4),
  Sharpe_Original = c(0.0752, -0.0601, 0.0390, -0.0076),
  Sharpe_Optimized = c(0.1861, 0.0854, 0.1626, 0.1442),
  WinRate_Original = c(0.2324, 0.1620, 0.2606, 0.1831),
  WinRate_Optimized = c(0.3028, 0.2394, 0.3099, 0.2254)
```

```
)

# change it to pivot_longer)
sharpe_long <- performance_summary_df %>%
  select(Symbol, Strategy, Sharpe_Original, Sharpe_Optimized) %>%
  pivot_longer(cols = c(Sharpe_Original, Sharpe_Optimized),
               names_to = "Version", values_to = "Sharpe")

# Format version names for display
sharpe_long$Version <- gsub("Sharpe_", "", sharpe_long$Version)

# Visualization
ggplot(sharpe_long, aes(x = Symbol, y = Sharpe, fill = Version)) +
  geom_col(position = "dodge") +
  facet_wrap(~ Strategy) +
  labs(title = " Sharpe Ratio: Original vs Optimized",
       y = "Sharpe Ratio", x = "Symbol") +
  theme_minimal()
```




```
# WinRate
winrate_long <- performance_summary_df %>%
  select(Symbol, Strategy, WinRate_Original, WinRate_Optimized) %>%
  pivot_longer(cols = c(WinRate_Original, WinRate_Optimized),
               names_to = "Version", values_to = "WinRate")

winrate_long$Version <- gsub("WinRate_", "", winrate_long$Version)

# Visualization
ggplot(winrate_long, aes(x = Symbol, y = WinRate, fill = Version)) +
```

```
  geom_col(position = "dodge") +
  facet_wrap(~ Strategy) +
  labs(title = " WinRate: Original vs Optimized",
       y = "Win Rate", x = "Symbol") +
  theme_minimal()
```



got way better, for both sharp ratio is way better than before, which means its very profitable with less volatility, and also win rate is higher too, which means its winning more times than original.

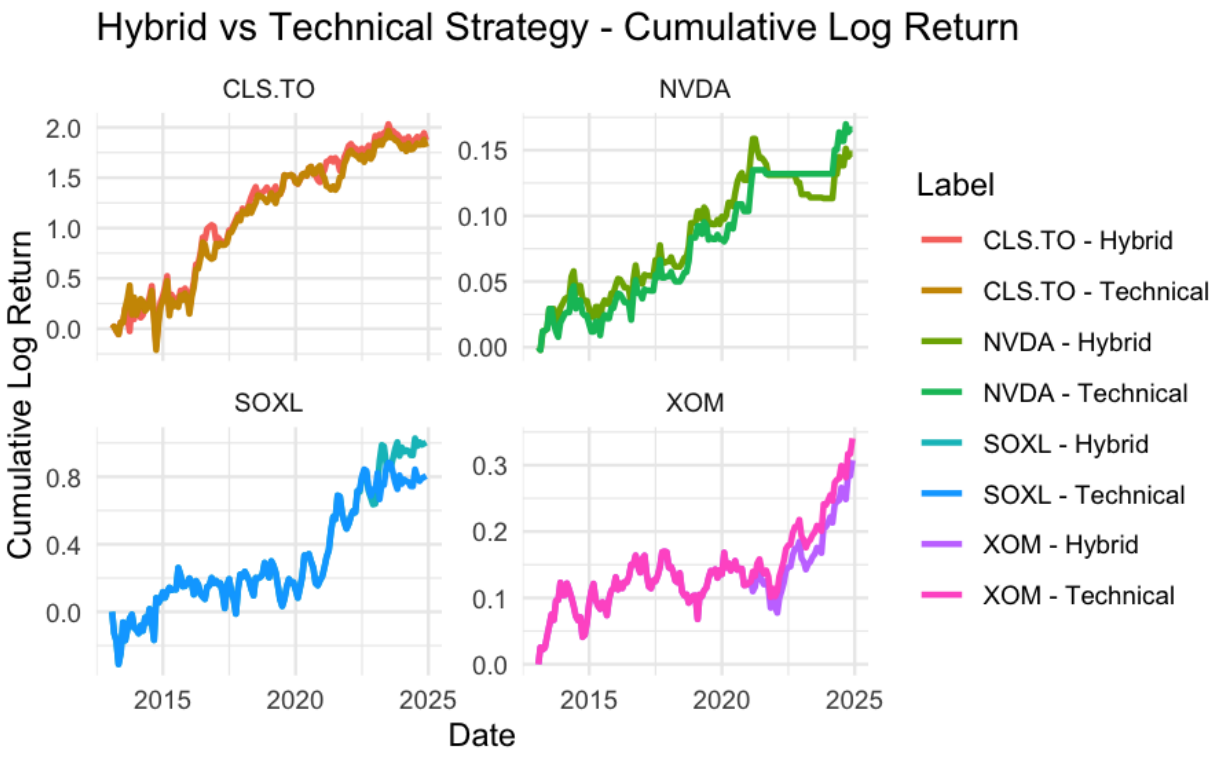
```
library(dplyr)
library(ggplot2)
library(xts)
library(PerformanceAnalytics)

symbols <- c("NVDA", "SOXL", "XOM", "CLS.TO")

compare_results <- data.frame()
cumret_list <- list()

for (sym in symbols) {
  cat("\n", sym, ": Comparing Technical vs Hybrid...\n")

  # Best Technical Result
  tech_row <- global_results[[sym]] |> filter(Score == max(Score))
  w1_t <- tech_row$w1
  w2_t <- tech_row$w2
  threshold_t <- tech_row$threshold

  f_tech <- p_ma_3_12_sync
  f_tech[symbols] <- w1_t * p_ma_3_12_sync[symbols] + w2_t * p_risk_sync[symbols]
```

```r
names(f_tech)[-1] <- paste0("f_tech_", symbols)

f_col_t <- paste0("f_tech_", sym)
ret_col <- paste0("logret_", sym)
df_t <- merge(f_tech[, c("Date", f_col_t)],
              logret_df[, c("Date", ret_col)], by = "Date") |> arrange(Date)
df_t$signal <- ifelse(df_t[[f_col_t]] > threshold_t, 1,
                      ifelse(df_t[[f_col_t]] < (1 - threshold_t), -1, 0))
df_t$position <- dplyr::lag(df_t$signal, 1, default = 0)
df_t$strat_ret <- df_t$position * df_t[[ret_col]]
df_t$cum_ret <- cumsum(ifelse(is.na(df_t$strat_ret), 0, df_t$strat_ret))
sharpe_t <- mean(df_t$strat_ret, na.rm=TRUE) / sd(df_t$strat_ret, na.rm=TRUE)
winrate_t <- sum(df_t$strat_ret > 0, na.rm=TRUE) / sum(!is.na(df_t$strat_ret))

cumret_list[[paste0(sym, "_Technical")]] <- data.frame(
  Date = df_t$Date, cum_ret = df_t$cum_ret, Label = paste0(sym, " - Technical"))

# Best Hybrid result
hyb_row <- global_results_hyb[[sym]] |> filter(Score == max(Score))
w1_h <- hyb_row$w1
w2_h <- hyb_row$w2
alpha <- hyb_row$alpha
threshold_h <- hyb_row$threshold

f_tech_h <- p_ma_3_12_sync
f_tech_h[symbols] <- w1_h * p_ma_3_12_sync[symbols] + w2_h * p_risk_sync[symbols]
names(f_tech_h)[-1] <- paste0("f_tech_", symbols)

f_hyb <- f_tech_h[, "Date", drop=FALSE]
for (s in symbols) {
  f_hyb[[paste0("f_hyb_", s)]] <- alpha * f_tech_h[[paste0("f_tech_", s)]] +
                                  (1 - alpha) * p_econ_sync[[s]]
}

f_col_h <- paste0("f_hyb_", sym)
df_h <- merge(f_hyb[, c("Date", f_col_h)],
              logret_df[, c("Date", ret_col)], by = "Date") |> arrange(Date)
df_h$signal <- ifelse(df_h[[f_col_h]] > threshold_h, 1,
                      ifelse(df_h[[f_col_h]] < (1 - threshold_h), -1, 0))
df_h$position <- dplyr::lag(df_h$signal, 1, default = 0)
df_h$strat_ret <- df_h$position * df_h[[ret_col]]
df_h$cum_ret <- cumsum(ifelse(is.na(df_h$strat_ret), 0, df_h$strat_ret))
sharpe_h <- mean(df_h$strat_ret, na.rm=TRUE) / sd(df_h$strat_ret, na.rm=TRUE)
winrate_h <- sum(df_h$strat_ret > 0, na.rm=TRUE) / sum(!is.na(df_h$strat_ret))

cumret_list[[paste0(sym, "_Hybrid")]] <- data.frame(
  Date = df_h$Date, cum_ret = df_h$cum_ret, Label = paste0(sym, " - Hybrid"))

# Store performance comparison
compare_results <- rbind(compare_results, data.frame(
  Symbol = sym,
  Sharpe_Technical = sharpe_t,
  WinRate_Technical = winrate_t,
```

```
    Sharpe_Hybrid = sharpe_h,
    WinRate_Hybrid = winrate_h
  ))
}
```

```
##
##  NVDA : Comparing Technical vs Hybrid...
##
##  SOXL : Comparing Technical vs Hybrid...
##
##  XOM : Comparing Technical vs Hybrid...
##
##  CLS.TO : Comparing Technical vs Hybrid...
```

```
#  Compare Cumulative Return Graphs
cumret_df <- bind_rows(cumret_list)
ggplot(cumret_df, aes(x = Date, y = cum_ret, color = Label)) +
  geom_line(linewidth = 1) +
  facet_wrap(~ gsub(" - .*", "", Label), scales = "free_y") +
  labs(title = "Hybrid vs Technical Strategy - Cumulative Log Return",
       y = "Cumulative Log Return") +
  theme_minimal()
```

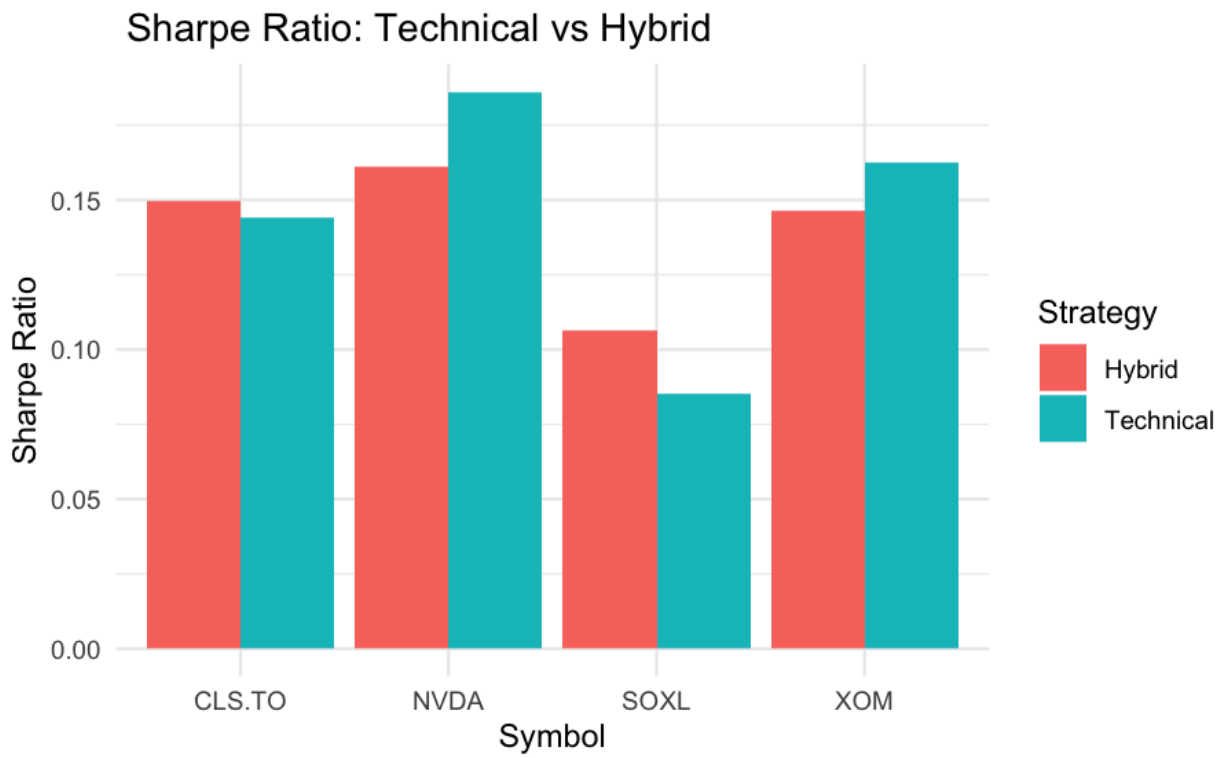



```
#  Performance comparison table
compare_results
```

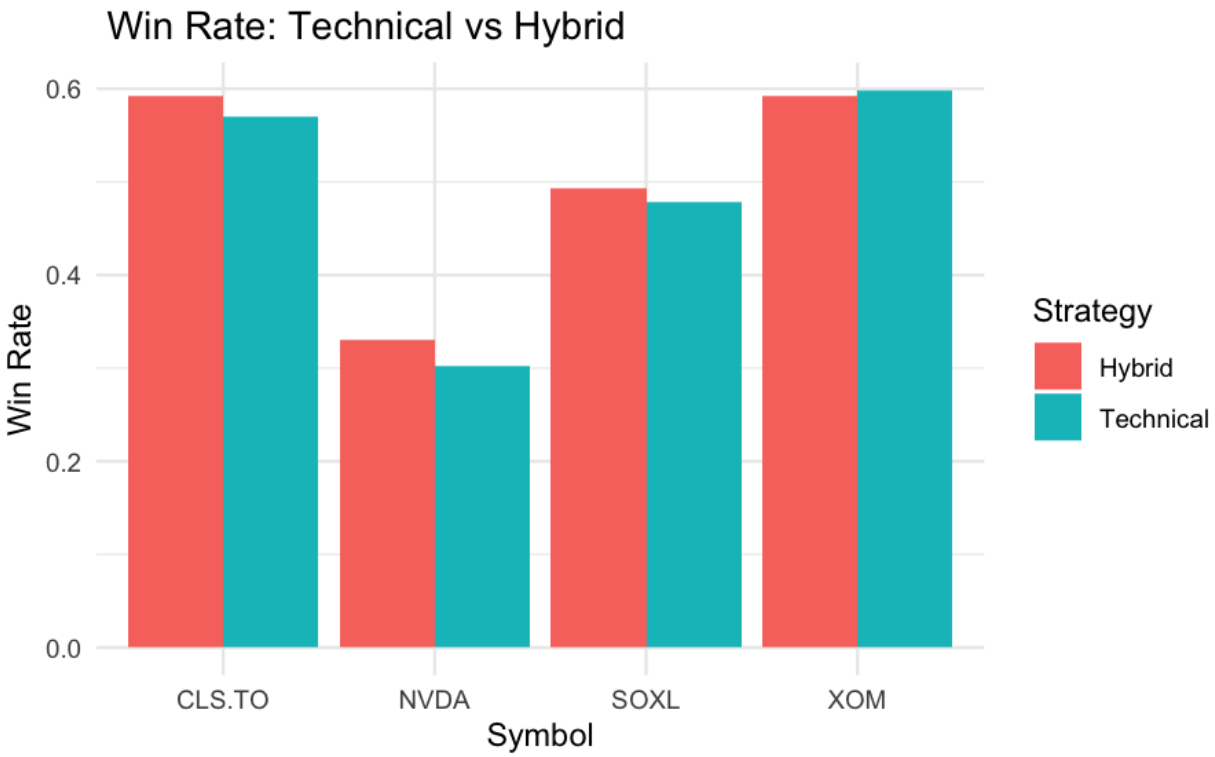
```
##   Symbol Sharpe_Technical WinRate_Technical Sharpe_Hybrid WinRate_Hybrid
## 1   NVDA        0.1861319         0.3028169     0.1612738      0.3309859
## 2   SOXL        0.0853663         0.4788732     0.1066271      0.4929577
## 3    XOM        0.1625968         0.5985915     0.1464631      0.5915493
```

```
## 4 CLS.TO        0.1442340         0.5704225      0.1496646       0.5915493
```

```
performance_comparison_df <- data.frame(
  Symbol = rep(c("NVDA", "SOXL", "XOM", "CLS.TO"), each = 2),
  Strategy = rep(c("Technical", "Hybrid"), times = 4),
  Sharpe = c(0.1861, 0.1613, 0.0854, 0.1066, 0.1626, 0.1465, 0.1442, 0.1497),
  WinRate = c(0.3028, 0.3310, 0.4789, 0.4930, 0.5986, 0.5915, 0.5704, 0.5915)
)
library(ggplot2)

ggplot(performance_comparison_df, aes(x = Symbol, y = Sharpe, fill = Strategy)) +
  geom_col(position = "dodge") +
  labs(title = " Sharpe Ratio: Technical vs Hybrid",
       y = "Sharpe Ratio", x = "Symbol") +
  theme_minimal()
```




```
ggplot(performance_comparison_df, aes(x = Symbol, y = WinRate, fill = Strategy)) +
  geom_col(position = "dodge") +
  labs(title = " Win Rate: Technical vs Hybrid",
       y = "Win Rate", x = "Symbol") +
  theme_minimal()
```



hybrid & technical comparision is more interesting, since boths are very similar,

Technical strategies show higher Sharpe Ratio on average. → In other words, they are slightly more efficient in terms of risk-to-return ratio.

On the other hand, Hybrid strategies show higher WinRate → In other words, they show a stable tendency to make profits in more than half of the trades.

Strategic Insight: For investors who want to avoid short-term or loss periods, Hybrid strategies with higher WinRate may be more stable.

For investors who value long-term returns and efficiency, Technical strategies with higher Sharpe may be more suitable.

Depending on the stock, there are cases where Hybrids are slightly ahead in all indicators, such as CLS.TO or SOXL.