Optimized P_Econ

Dong Wang

2025-04-01

Data Collection

```r
library(tseries)
```

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

```r
library(zoo)
```

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

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

```r
library(quantmod)
```

```
## Loading required package: xts

## Loading required package: TTR
```

```r
library(TTR)
library(PerformanceAnalytics)
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 4.3.3

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

```r
library(xts)
```

GDP Data

```r
# GDP Data
getSymbols("GDP", src = "FRED", from = "2010-01-01", to = "2024-12-31")
```

```
## [1] "GDP"
```

```r
gdp_df <- data.frame(date = index(GDP), GDP = coredata(GDP))
monthly_dates <- seq(from = min(gdp_df$date), to = max(gdp_df$date), by = "month")

# Interpolating quarterly GDP to monthly
gdp_interp <- approx(x = gdp_df$date, y = gdp_df$GDP, xout = monthly_dates, method = "linear")

gdp_monthly_xts <- xts(gdp_interp$y, order.by = as.Date(gdp_interp$x))
colnames(gdp_monthly_xts) <- "GDP"
str(gdp_monthly_xts)
```

```
## An xts object on 2010-01-01 / 2024-10-01 containing:
##   Data:    double [178, 1]
##   Columns: GDP
##   Index:   Date [178] (TZ: "UTC")
```

```r
head(gdp_monthly_xts)
```

```
##                 GDP
## 2010-01-01 14764.61
## 2010-02-01 14838.87
## 2010-03-01 14905.94
## 2010-04-01 14980.19
## 2010-05-01 15033.41
## 2010-06-01 15088.39
```

```r
# Covering the last 2 values of GDP data to make its length align with those of others
last_date <- index(gdp_monthly_xts)[nrow(gdp_monthly_xts)]  # Last date
next_dates <- seq(from = last_date, by = "month", length.out = 3)[-1]  # Next 2 months

extended_gdp_xts <- xts(rep(tail(gdp_monthly_xts, 1), length(next_dates)), order.by = next_dates)
colnames(extended_gdp_xts) <- "GDP"

gdp_monthly_xts <- rbind(gdp_monthly_xts, extended_gdp_xts)
```

Interest Rate Data

```r
# Interest Rate Data (10-Year Treasury Rate)
getSymbols("DGS10", src = "FRED", from = "2010-01-01", to = "2024-12-31", periodicity = "monthly'
```

```
## [1] "DGS10"
```

```r
# Converting daily data to monthly
DGS10_monthly <- to.monthly(DGS10, indexAt = "lastof", OHLC = FALSE)
```

```
## Warning in to.period(x, "months", indexAt = indexAt, name = name, ...): missing
## values removed from data
```

```r
head(DGS10_monthly)
```

```
##            DGS10
## 2010-01-31  3.63
## 2010-02-28  3.61
## 2010-03-31  3.84
## 2010-04-30  3.69
## 2010-05-31  3.31
## 2010-06-30  2.97
```

```r
names(DGS10_monthly) <- "DGS10_monthly"
```

```r
# Index Adjustment
index(DGS10_monthly) <- index(gdp_monthly_xts)

# Checking if indices match
identical(index(gdp_monthly_xts), index(DGS10_monthly))
```

```
## [1] TRUE
```

Definition & Visualization of P_Econ

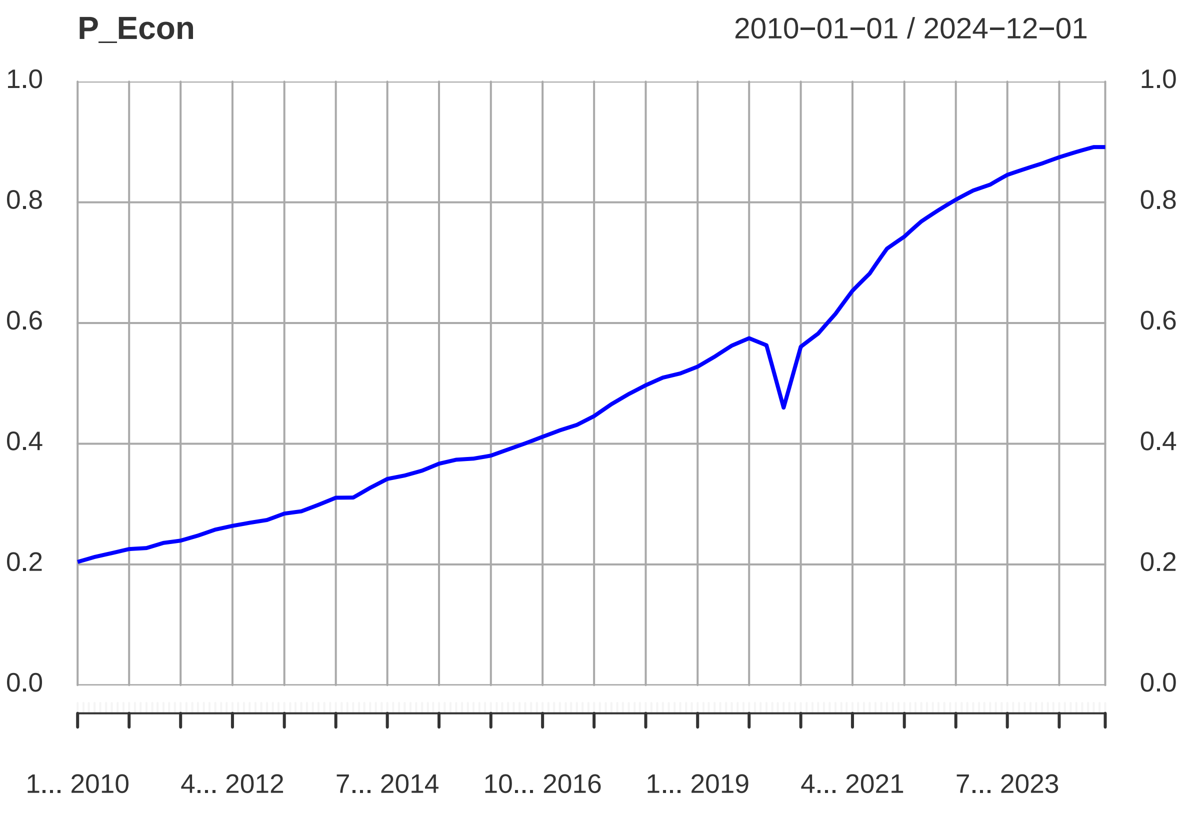
```r
# Sigmoid function
sigma <- function(x) {1 / (1 + exp(-x))}
```

```r
# Merging the 2 series in P_Econ
merged_xts_gdp_DGS10 <- merge(gdp_monthly_xts, DGS10_monthly, join = "inner")
colnames(merged_xts_gdp_DGS10) <- c("GDP", "DGS10")

# Computing the difference
merged_xts_gdp_DGS10$Diff <- merged_xts_gdp_DGS10$GDP - merged_xts_gdp_DGS10$DGS10

# Standardizing the difference
merged_xts_gdp_DGS10$Diff_Standardized <- scale(merged_xts_gdp_DGS10$Diff)
P_Econ_DGS10 = sigma(merged_xts_gdp_DGS10$Diff_Standardized)

# Visualization of P_Econ
plot.xts(P_Econ_DGS10, main = "P_Econ", ylim = c(0, 1), col = "blue", lwd = 2)
```



Optimizations of Lambdas for Each Asset

```
# Range for lambda
lambda_values <- seq(0.1, 10, by = 0.1)
```

NVDA

```
NVDA <- get.hist.quote(instrument = "NVDA",
                       start = "2010-01-01",
                       end = "2024-12-31",
                       quote = "AdjClose",
                       compression = "m")
```

```
## time series ends   2024-12-01
```

```
NVDA <- zoo(NVDA, order.by = as.Date(time(NVDA)))
```

```
score_list_NVDA <- data.frame()

for (lambda in lambda_values)
{
  # Returns & Trading signals
  P_Econ <- sigma(lambda * merged_xts_gdp_DGS10$Diff_Standardized)
  ret <- diff(log(NVDA))
  signals <- P_Econ[-1]

  # Trading strategy
  strategy_ret <- ifelse(signals > 0.3, ret, ifelse(signals < 0.1, -ret, 0))

  # Sharpe Ratio, Win/Loss Ratio, & Score
  sharpe <- as.numeric(SharpeRatio.annualized(strategy_ret, scale = 12, geometric = FALSE))
  wins <- sum(strategy_ret > 0, na.rm = TRUE)
  losses <- sum(strategy_ret < 0, na.rm = TRUE)
  win_loss <- ifelse(losses == 0, wins, wins/losses)
  score <- sharpe + win_loss

  # Storing in List
  score_list_NVDA <- rbind(score_list_NVDA, data.frame(Lambda = lambda,
                                                       Sharpe = sharpe,
                                                       WinLoss = win_loss,
                                                       Score = score))
}

# Finding the lambda value such that score is optimized
best_row <- score_list_NVDA[which.max(score_list_NVDA$Score), ]
print(best_row)
```

```
##   Lambda  Sharpe  WinLoss    Score
## 9    0.9 1.08738 1.938776 3.026156
```

SOXL

```
SOXL <- get.hist.quote(instrument = "SOXL",
                       start = "2010-01-01",
                       end = "2024-12-31",
                       quote = "AdjClose",
                       compression = "m")
```

```
## time series starts 2010-03-01
## time series ends   2024-12-01
```

```
SOXL <- zoo(SOXL, order.by = as.Date(time(SOXL)))
```

```
score_list_SOXL <- data.frame()

for (lambda in lambda_values)
{
  # Returns & Trading signals
  P_Econ <- sigma(lambda * merged_xts_gdp_DGS10$Diff_Standardized)
  ret <- diff(log(SOXL))
  signals <- P_Econ[-1]

  # Trading strategy
  strategy_ret <- ifelse(signals > 0.4, ret, ifelse(signals < 0.2, -ret, 0))

  # Sharpe Ratio, Win/Loss Ratio, & Score
  sharpe <- as.numeric(SharpeRatio.annualized(strategy_ret, scale = 12, geometric = FALSE))
  wins <- sum(strategy_ret > 0, na.rm = TRUE)
  losses <- sum(strategy_ret < 0, na.rm = TRUE)
  win_loss <- ifelse(losses == 0, wins, wins/losses)
  score <- sharpe + win_loss

  # Storing in List
  score_list_SOXL <- rbind(score_list_SOXL, data.frame(Lambda = lambda,
                                                         Sharpe = sharpe,
                                                         WinLoss = win_loss,
                                                         Score = score))
}

# Finding the lambda value such that score is optimized
best_row <- score_list_SOXL[which.max(score_list_SOXL$Score), ]
print(best_row)
```

```
##   Lambda    Sharpe  WinLoss    Score
## 4    0.4 0.3917412 1.359375 1.751116
```

XOM

```r
XOM <- get.hist.quote(instrument = "XOM",
                      start = "2010-01-01",
                      end = "2024-12-31",
                      quote = "AdjClose",
                      compression = "m")
```

```
## time series ends   2024-12-01
```

```r
XOM <- zoo(XOM, order.by = as.Date(time(XOM)))
```

```r
score_list_XOM <- data.frame()

for (lambda in lambda_values)
{
  # Returns & Trading signals
  P_Econ <- sigma(lambda * merged_xts_gdp_DGS10$Diff_Standardized)
  ret <- diff(log(XOM))
  signals <- P_Econ[-1]

  # Trading strategy
  strategy_ret <- ifelse(signals > 0.7, ret, ifelse(signals < 0.01, -ret, 0))

  # Sharpe Ratio, Win/Loss Ratio, & Score
  sharpe <- as.numeric(SharpeRatio.annualized(strategy_ret, scale = 12, geometric = FALSE))
  wins <- sum(strategy_ret > 0, na.rm = TRUE)
  losses <- sum(strategy_ret < 0, na.rm = TRUE)
  win_loss <- ifelse(losses == 0, wins, wins/losses)
  score <- sharpe + win_loss

  # Storing in List
  score_list_XOM <- rbind(score_list_XOM, data.frame(Lambda = lambda,
                                                     Sharpe = sharpe,
                                                     WinLoss = win_loss,
                                                     Score = score))
}

# Finding the lambda value such that score is optimized
best_row <- score_list_XOM[which.max(score_list_XOM$Score), ]
print(best_row)
```

```
##    Lambda    Sharpe  WinLoss    Score
## 23    2.3 0.6054196 1.777778 2.383197
```

CLS.TO

```r
CLS.TO <- get.hist.quote(instrument = "CLS.TO",
                         start = "2010-01-01",
                         end = "2024-12-31",
                         quote = "AdjClose",
                         compression = "m")
```

```
## time series ends   2024-12-01
```

```r
CLS.TO <- zoo(CLS.TO, order.by = as.Date(time(CLS.TO)))
```

```r
score_list_CLS.TO <- data.frame()

for (lambda in lambda_values)
{
  # Returns & Trading signals
  P_Econ <- sigma(lambda * merged_xts_gdp_DGS10$Diff_Standardized)
  ret <- diff(log(CLS.TO))
  signals <- P_Econ[-1]

  # Trading strategy
  strategy_ret <- ifelse(signals > 0.99, ret, ifelse(signals < 0.01, -ret, 0))

  # Sharpe Ratio, Win/Loss Ratio, & Score
  sharpe <- as.numeric(SharpeRatio.annualized(strategy_ret, scale = 12, geometric = FALSE))
  wins <- sum(strategy_ret > 0, na.rm = TRUE)
  losses <- sum(strategy_ret < 0, na.rm = TRUE)
  win_loss <- ifelse(losses == 0, wins, wins/losses)
  score <- sharpe + win_loss

  # Storing in List
  score_list_CLS.TO <- rbind(score_list_CLS.TO, data.frame(Lambda = lambda,
                                                           Sharpe = sharpe,
                                                           WinLoss = win_loss,
                                                           Score = score))
}

# Finding the lambda value such that score is optimized
best_row <- score_list_CLS.TO[which.max(score_list_CLS.TO$Score), ]
print(best_row)
```

```
##    Lambda    Sharpe WinLoss    Score
## 29    2.9 0.7817952       4 4.781795
```