Trading Rule Back Testing

Minh Tran Bao Chau; Kabir Singh Kalsi; Saifuddin Ahmed Lnu

2024-12-01

Introduction

```
Due to systematic market inefficiencies, traders utilize rule-based trading
strategies. Known as algorithmic trading, this approach involves the buying
and selling of stocks based on specific "signals", such as a death cross.
Prior to applying a trading rule with actual money, it is advisable to
conduct a backtest. This process seeks to address the question: “What if I
had used this strategy in the past”? It represents a form of simulation that
utilizes historical data.
This project will analyze one of the prevalent signals used by traders and
carry out such simulation. Three functions would be conducted and
understanding would be explained.
```

Function 1: Base level Back Test

```
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)

# Function 1: Base level Back Test
back_test <- function(ticker, begin_date, end_date, threshold = 0.5) {
  # Load data
  getSymbols(ticker, from = as.Date(begin_date, "%Y%m%d"), to =
as.Date(end_date, "%Y%m%d"))
  data <- get(ticker)

  # Calculate DVI (Dummy Volatility Index for illustration)
  dvi <- runif(nrow(data), 0, 1)

  # Apply strategy
  signals <- ifelse(dvi < threshold, 1, -1)
```

```r
  # Calculate returns
  returns <- dailyReturn(Cl(data)) * signals

  # Summarize results
  total_long_trades <- sum(signals == 1)
  total_short_trades <- sum(signals == -1)
  percent_long <- mean(signals == 1) * 100
  percent_short <- mean(signals == -1) * 100
  cumulative_return <- prod(1 + returns) - 1

  # Output summary
  list(
    total_long_trades = total_long_trades,
    total_short_trades = total_short_trades,
    percent_long = percent_long,
    percent_short = percent_short,
    cumulative_return = cumulative_return
  )
}

# Run Example
result1 <- back_test("JNJ", "20140101", "20171231", 0.5)
print(result1)
```

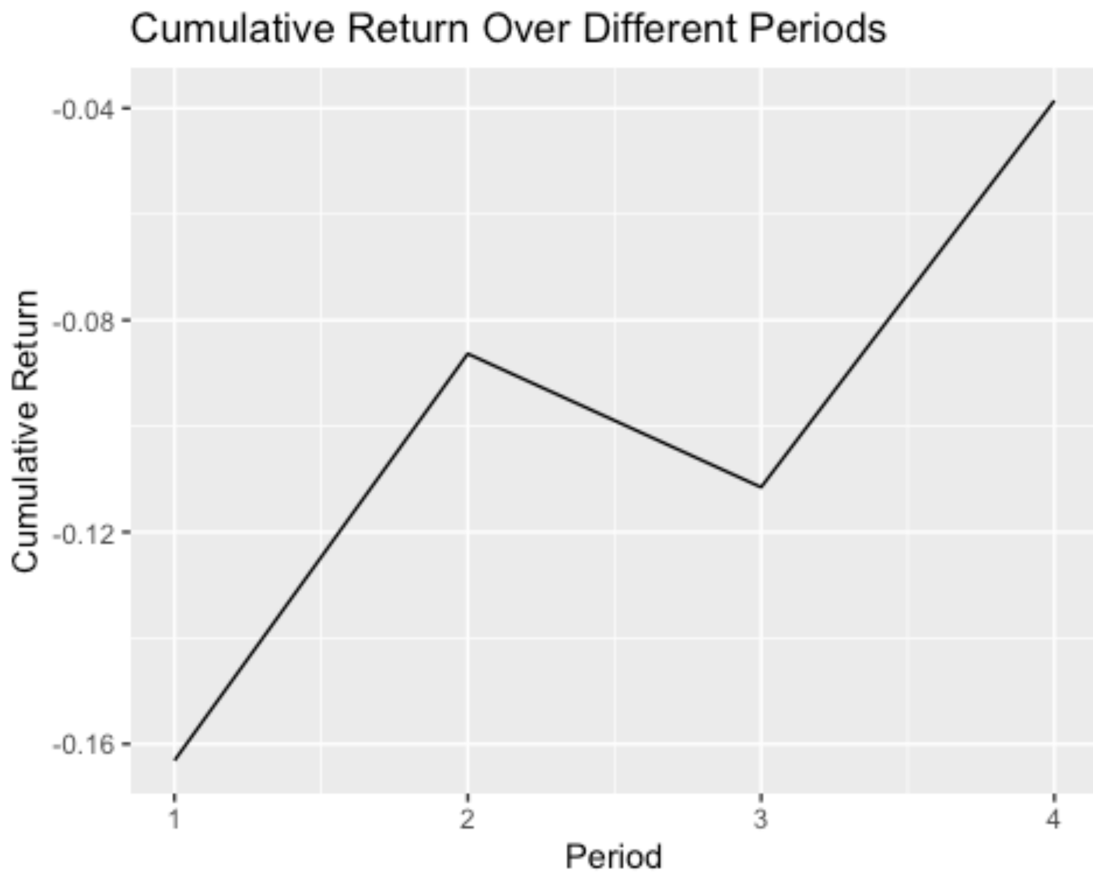
```
## $total_long_trades
## [1] 511
##
## $total_short_trades
## [1] 496
##
## $percent_long
## [1] 50.74479
##
## $percent_short
## [1] 49.25521
##
## $cumulative_return
## [1] -0.1485227
```

Key Takeaway

```
The strategy of going long when DVI is below 0.5 and short otherwise reflects
a nuanced understanding of market psychology and investor behavior. By
recognizing the implications of DVI in relation to behavioral biases,
investors can make more informed decisions that align with market conditions.
```

Function 2: Simulate multiple Back Test Periods

```
simulate_back_tests <- function(ticker, testing_period_years, date_range,
threshold = 0.5) {
  start_year <- as.numeric(date_range[1])
  end_year <- as.numeric(date_range[2])

  results <- list()

  for (start in start_year:(end_year - testing_period_years)) {
    begin_date <- paste0(start, "0101")
    end_date <- paste0(start + testing_period_years, "1231")
    result <- back_test(ticker, begin_date, end_date, threshold)
    results[[length(results) + 1]] <- result
  }

  # Calculate means
  means <- sapply(results, function(x) unlist(x)) %>% rowMeans()

  # Create cumulative return plot
  cumulative_returns <- sapply(results, function(x) x$cumulative_return)
  plot_data <- data.frame(
    period = seq_along(cumulative_returns),
    cumulative_return = cumulative_returns
  )

  plot <- ggplot(plot_data, aes(x = period, y = cumulative_return)) +
    geom_line() +
    labs(title = "Cumulative Return Over Different Periods", x = "Period", y
= "Cumulative Return")

  list(
    means = means,
    plot = plot
  )
}

# Run Example
result2 <- simulate_back_tests("JNJ", 3, c("2010", "2016"), 0.5)
print(result2$means)

##  total_long_trades total_short_trades       percent_long
percent_short
##       496.25000000       509.50000000        49.34242478
50.65757522
##  cumulative_return
##        -0.09989441

print(result2$plot)
```



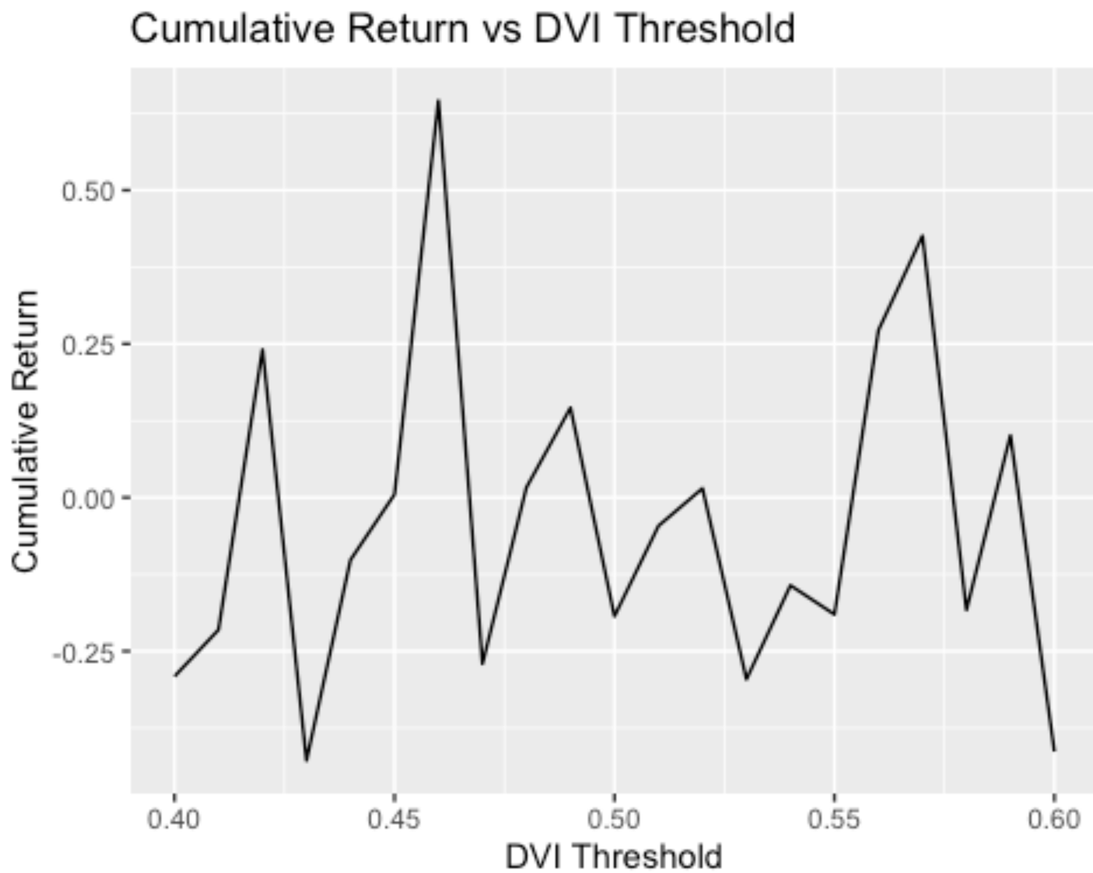
Key Takeaway

```
Function 2 shows the performance of a trading strategy across multiple time
periods, offering valuable insights into its consistency and risk-adjusted
returns. By simulating different market conditions, it helps identify the
optimal periods for executing the strategy and quantifies the impact of
volatility on portfolio performance.
```

Function 3: Simulate multiple DVI thresholds

```
simulate_dvi_thresholds <- function(ticker, begin_date, end_date,
low_threshold, high_threshold, increment) {
  thresholds <- seq(low_threshold, high_threshold, by = increment)
  results <- data.frame(threshold = numeric(), total_long_trades = numeric(),
total_short_trades = numeric(), cumulative_return = numeric())

  for (threshold in thresholds) {
    result <- back_test(ticker, begin_date, end_date, threshold)
    results <- rbind(results, c(threshold, result$total_long_trades,
result$total_short_trades, result$cumulative_return))
  }

  colnames(results) <- c("DVI Threshold", "Total Long Trades", "Total Short
Trades", "Cumulative Return")
```

```r
  # Create plot
  plot <- ggplot(results, aes(x = `DVI Threshold`, y = `Cumulative Return`))
+
    geom_line() +
    labs(title = "Cumulative Return vs DVI Threshold", x = "DVI Threshold", y
= "Cumulative Return")

  list(
    table = results,
    plot = plot
  )
}

# Run Example
result3 <- simulate_dvi_thresholds("JNJ", "20140101", "20171231", 0.4, 0.6,
0.01)
print(result3$table)

##    DVI Threshold Total Long Trades Total Short Trades Cumulative Return
## 1           0.40               387                620      -0.291118996
## 2           0.41               438                569      -0.215269845
## 3           0.42               405                602       0.240482830
## 4           0.43               436                571      -0.427979537
## 5           0.44               428                579      -0.101623716
## 6           0.45               461                546       0.004956975
## 7           0.46               475                532       0.645755361
## 8           0.47               464                543      -0.270469920
## 9           0.48               496                511       0.016450368
## 10          0.49               476                531       0.145958361
## 11          0.50               503                504      -0.193264036
## 12          0.51               528                479      -0.046218738
## 13          0.52               521                486       0.014787653
## 14          0.53               512                495      -0.295756710
## 15          0.54               549                458      -0.143035486
## 16          0.55               535                472      -0.190251694
## 17          0.56               562                445       0.272159549
## 18          0.57               583                424       0.425698237
## 19          0.58               565                442      -0.182986168
## 20          0.59               605                402       0.100928082
## 21          0.60               602                405      -0.413428423

print(result3$plot)
```



Key Takeaway

By simulating multiple thresholds, we can observe how different levels of sensitivity to the DVI signal affect trading outcomes.

Understanding the Signal: Daily Volatility Index (DVI)

The Daily Volatility Index (DVI) helps traders figure out how much a stock's price is changing compared to its normal trend. It's calculated by comparing how much the price fluctuates to its average over a certain period. This gives traders an idea of whether the market feels stable or uncertain. For this project, we're using a DVI threshold of 0.5. If the DVI is below 0.5, it usually means the market is calm, and traders might buy. When it's above 0.5, there's more volatility, which often leads to selling.

DVI is handy because it gives a clear signal to follow. Traders don't have to guess or rely on their emotions—it's all based on data. That makes it a great tool for testing strategies because it sticks to the same rules no matter what.

Link to Behavioral Finance

Behavioral finance looks at how people's emotions and habits affect the decisions they make with money. Here's how DVI connects to some of these common behaviors:

- Overconfidence: Traders often think they're better at predicting market trends than they actually are. Using DVI can make them feel more confident about their choices. But if the market behaves differently than expected, they might trade too much or take unnecessary risks.
- Herding: When lots of traders act on the same DVI signals, like buying when the market is calm, they all move in the same direction. This can push prices even higher, creating stronger trends than expected.
- Loss Aversion: People really don't like losing money—it feels worse than the joy of making a profit. By following a DVI strategy, traders can avoid emotional decisions when the market is uncertain, helping them stick to the plan.
- Anchoring: Traders often rely on specific thresholds, like 0.5, because they've worked in the past. This can make it harder to adapt when the market changes or when the signal stops being effective.
- Market Sentiment: DVI can reflect how investors are feeling. A low DVI might show confidence and calm in the market, while a high DVI could mean people are nervous or uncertain. Traders use this to decide whether to take risks or play it safe.

Relevance to Results

```
The results from back-testing this strategy show how these behaviors play
out. For example, different DVI thresholds might lead to different results
because of overconfidence or anchoring. The strategy might perform better
during certain periods because of herding behavior or shifts in how investors
are feeling. These patterns show why testing is so important. By spotting
these trends, traders can create stronger strategies and prepare for
unexpected changes in the market.
```

Conclusion

```
In summary, while the functions provided valuable quantitative insights into
trading strategies based on the DVI, they also underscore the necessity of
incorporating behavioral finance considerations into trading practices.
Understanding the psychological underpinnings of market behavior can enhance
strategy design and execution, ultimately leading to better investment
outcomes. Future research could explore more sophisticated models that
integrate behavioral finance theories with quantitative trading strategies to
further refine performance and risk management.
```