Assignment 3: Trading Rule Back Testing

Dana DiVincenzo
6/27/21

Introduction

Behavioral finance is an area of study which focuses on how the emotions, biases, and cognitive limitations of investors can affect market outcomes. By studying behavioral finance, we are able to make better determinations on why the market is not fully efficient. Some examples of biases in the market are confirmation bias, experiential bias, loss aversion, and familiarity bias. Confirmation bias is when investors already hold a belief and are more likely to accept information that furthers that belief than information that goes against that belief. Experiential bias is when the investors recent memories or experiences makes them more likely to believe that those same events are more likely to occur again. Loss aversion is when investors put too high of a weight on avoiding market losses than they do on attaining market gains. Familiarity bias is when investors invest too much into sectors they know, preventing proper diversification and therefore increasing risk. It is important as an investor to be aware of these biases so you can prevent their impact as much as possible. *(Hayes, 2021)*

Back testing is a vital tool for investors - it is a method of analyzing the performance of a trading strategy applied to a time period in the past. By doing this, the investor can assess the strategy's behavior and see which parameters lead to the largest gains. However, back testing is not impervious to bias. Optimization bias, or continuously adding parameters to overfit the model, will lead to positive results within the simulation, but will cause poor results when applied to live data. Survivorship bias also leads to poor consequences. This is when the investor backtests on an incomplete range of assets relative to the assets they are actually interested in trading. Other types of bias exist in backtesting as well. However, when done correctly, backtesting can strengthen your trading strategy and give you a better performance. *(Tachev, 2021)*

In this assignment, I create functions to perform various back end tests, based on a DVI indicator. The DVI indicator was introduced in 2009 as a strong predictor for the S&P 500, a stock market index tracking the performance of the 500 largest companies listed on stock exchanges in the US. This indicator oscillates between 0 and 1, and a typical strategy is to go short if the market close is greater than .5, and go long otherwise. However, other DVI thresholds can be applied with success. Below, I show 3 functions, which apply back end testing to the Johnson and Johnson stock, with various date and DVI threshold parameters. The results of these simulations are shown and analyzed. *(Analyzing the DVI Indicator 2015)*

Function 1

This function, given a stock, date range, and DVI threshold (defaulted to 0.5) will simulate back end testing for that stock within that date range, trading based on the given DVI threshold. This function will return a summary of results containing the number of long trades, number of short trades, percent long trades, percent short trades, and the cumulative returns.

```
library(quantmod)
library(TTR)
library(dplyr)
library(PerformanceAnalytics)
library(ggplot2)
library(RColorBrewer)
library(knitr)
```

```
function1 <- function(ticker, begin, end, dvi_threshold = 0.5) {
  # formatting date variables
  begin <- as.Date(as.character(begin), format="%Y%m%d")
  end <- as.Date(as.character(end), format="%Y%m%d")

  # getting daily stock info between dates
  price <- getSymbols(ticker, auto.assign = F, from = format(begin, "%Y-%m-%d") , to = f
ormat(end, "%Y-%m-%d"), periodicity = "daily")
  price = price[,4]

  # creating DVI
  dvi <- DVI(Cl(price))

  # this calculates long or short position, lag position takes the previous date's posit
ion
  position <- ifelse(dvi$dvi < dvi_threshold, 1, -1)
  lag_position <- Lag(position)

  # getting returns
  ret <- ROC(Cl(price))
  ret <- ret[paste(begin, "/", end, sep = '')]

  dfxts <- merge(price, ret, dvi$dvi, position, lag_position)
  colnames(dfxts) <- c("Price", "Return", "DVI", "Position", "LagPosition")

  # dropping NAs
  dfxts <- na.omit(dfxts)

  # calculating summary outputs
  num_long <- length(which(dfxts$LagPosition == 1))
  num_short <- length(which(dfxts$LagPosition == -1))
  percent_long <- num_long / nrow(dfxts) * 100
  percent_short <- num_short / nrow(dfxts) * 100
  cum_ret <- as.numeric(Return.cumulative(ret*lag_position))

  retdf <- data.frame (num_long, num_short, percent_long, percent_short, cum_ret)
  colnames(retdf) <- c("TotalLong", "TotalShort", "PercentLong", "PercentShort", "Cumula
tiveReturn" )
  retdf
}
```

Function 2

This function, given a stock, a testing period in years, a date range, and a DVI threshold (defaulted to 0.5), will simulate back end testing for that stock within that date range, trading based on the given DVI threshold, similar to function 1. However, this function breaks the given date range down into groups of years the size of the given testing period. Then, the same summary from function 1 is outputted, but now displaying the average values for each group of years within the date range.

```
function2 <- function(ticker, testing_period, date_range, dvi_threshold = 0.5) {
  # making an empty df
  mean_data <- data.frame()

  # gets correct date range based on testing period
  end_year <- as.numeric(date_range[2]) - testing_period +1
  for (year in date_range[1]:end_year) {
    date_start <- (paste(year, '0101', sep = ''))
    date_end <- (paste(year + testing_period - 1, "1231", sep = ''))

    # getting the output from function 1 and then calculating the averages for each colu
mn based on the testing period
    period_sum <- function1(ticker, date_start, date_end, dvi_threshold)
    period_mean <- data.frame(period_sum$TotalLong / testing_period, period_sum$TotalSho
rt / testing_period,
                              period_sum$PercentLong, period_sum$PercentShort, period_su
m$CumulativeReturn / testing_period)
    mean_data <- rbind(mean_data, cbind(range=paste(year, '-', year + testing_period - 1
, sep = ''), period_mean))
  }
  colnames(mean_data) <- c("Range", "MeanTotalLong", "MeanTotalShort", "MeanPercentLong"
, "MeanPercentShort", "MeanCumulativeReturn" )
  mean_data
}
```

Function 3

This function, given a stock, date range, min and max values for the DVI threshold, and increments for the DVI threshold, will simulate back end testing for the given stock for each DVI increment between the max and min threshold values, over the chosen date range. The function will then output a table showing the DVI threshold, the total number of long trades, total number of short trades, and the cumulative returns for that threshold.

```
function3 <- function(ticker, begin, end, low_dvi, high_dvi, dvi_increment) {
  all_data <- data.frame()

  # goes through each DVI increment within the range and gets the number of long trades,
short trades, and cumulative returns
  for(current_inc in seq(low_dvi, high_dvi, by = dvi_increment)) {
    fun1_ret <- function1(ticker, begin, end, current_inc)
    current_inc_df <- data.frame(current_inc, fun1_ret$TotalLong, fun1_ret$TotalShort, f
un1_ret$CumulativeReturn)
    all_data <- rbind(all_data, current_inc_df)
  }
  colnames(all_data) <- c("DVIThreshold", "TotalLong", "TotalShort", "CumulativeReturn")
  all_data
}
```

Output and Analysis

In the following sections, I will run each function and discuss the outputs. Graphs will be included as necessary.

Function 1 Output

```
fun1 <- function1('JNJ','20140101','20171231',0.5)
kable(fun1, align = "lllll", caption = "Back End Testing Data Summary for JNJ from 2014
 - 2017")
```

Back End Testing Data Summary for JNJ from 2014 - 2017

TotalLong	TotalShort	PercentLong	PercentShort	CumulativeReturn
312	338	48	52	-0.1693098

Function 1 Analysis

We can see from the table above, that using our DVI threshold of .5, for the JNJ stock, our strategy resulted in 312 total long trades, and 338 total short trades. This translates to 48% long trades and 52% short trades. Unfortunately, this strategy did not seem to work well here, as the cumulative returns here are -0.1693098, so we would have lost money in this specific simulation. However, this negative result does not mean we should completely disregard this strategy. We can re-simulate using different date values and DVI thresholds and see how things change. Furthermore, although not shown in this specific report, it would likely help us to simulate this with more than one stock. We have to be careful, though, to not fine-tune the simulation too much, which could cause overfitting and poor future performance.

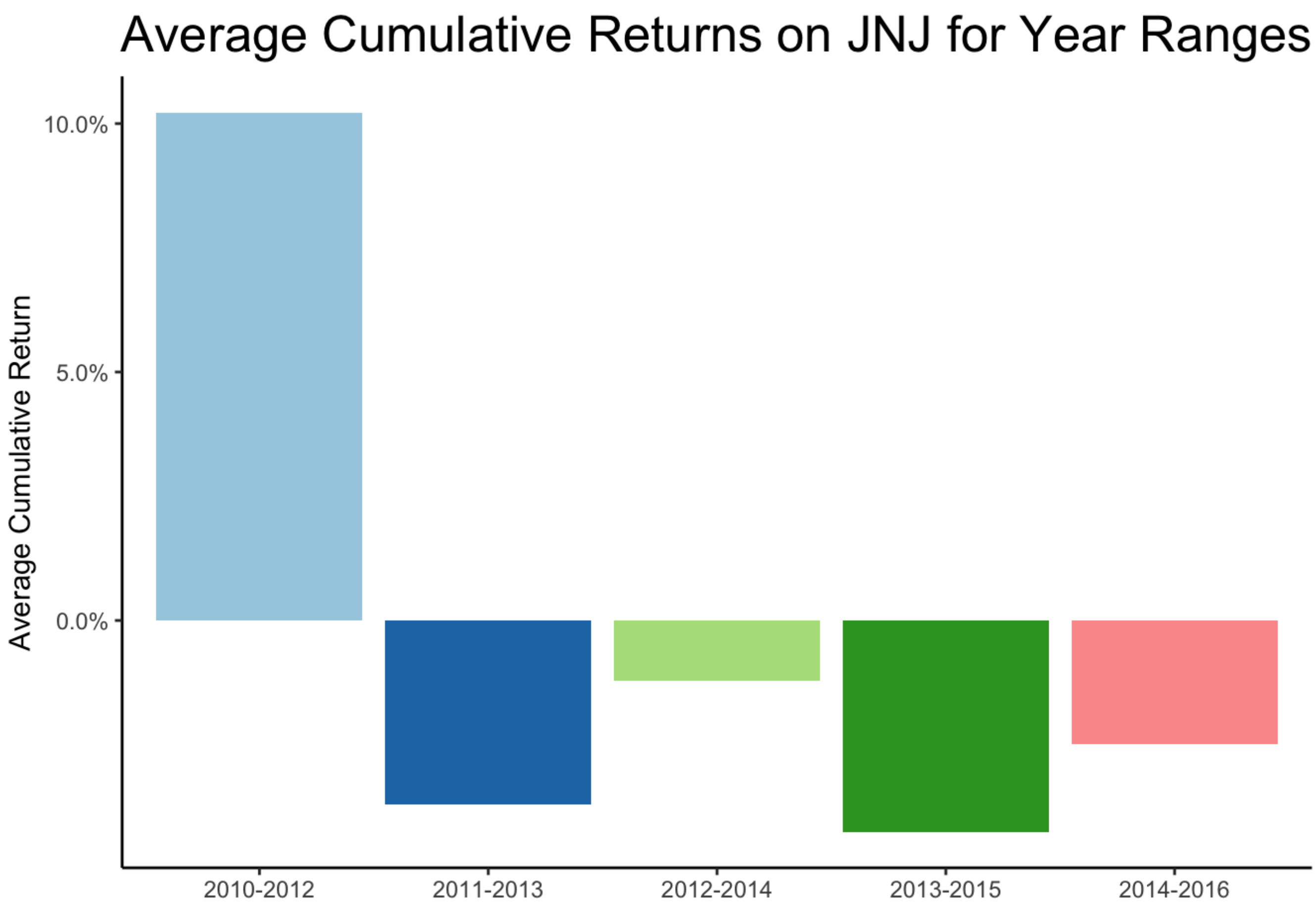
Function 2 Output

```
fun2 <- function2('JNJ', 3, c('2010', '2016'), 0.5)
kable(fun2, align = "lllll", caption = "Back End Testing Data Summary for JNJ, for 3 Yea
r Ranges from 2014 - 2017")
```

Back End Testing Data Summary for JNJ, for 3 Year Ranges from 2014 - 2017

Range	MeanTotalLong	MeanTotalShort	MeanPercentLong	MeanPercentShort	MeanCumulativeReturn
2010-2012	66.00000	66.00000	50.00000	50.00000	0.1022682
2011-2013	57.66667	74.33333	43.68687	56.31313	-0.0369614
2012-2014	74.66667	57.33333	56.56566	43.43434	-0.0120976
2013-2015	70.66667	62.00000	53.26633	46.73367	-0.0425442
2014-2016	65.33333	67.66667	49.12281	50.87719	-0.0248723

```
# graphing the data from function 2
ggplot(data = fun2) +
  geom_col(aes(x = Range, y = MeanCumulativeReturn, fill = Range)) +
  scale_fill_brewer(palette="Paired") +
  theme(legend.position="none", plot.title = element_text(hjust = 0.5, size = 19), pane
l.grid.major = element_blank(),
        panel.grid.minor = element_blank(), panel.background = element_blank(), axis.lin
e = element_line(colour = "black")) +
  labs(x = '', y = 'Average Cumulative Return',
       title = "Average Cumulative Returns on JNJ for Year Ranges") +
  scale_y_continuous(labels = scales::percent)
```



Function 2 Analysis

The table above shows us our return summary from function 1, however, we are now seeing the average values of each column for each group of 3 years. The most important column to look at is the Mean Cumulative Return column, so that is displayed graphically just above. We can see that the cumulative returns were negative for each group of years, besides the 2010-2012 group, which had a return of about 10%. These results are poor, however, it is based on only 1 stock during a 3 year period. Results may have been better if we used a variety of stocks, and a period of different length. Additionally, the DVI value could be adjusted to obtain different results. Although it is good to test out different parameters, we have to be careful to avoid over-tuning to get the results we want. For example, if we are looking for a 3 year long strategy, we should not run simulations on a 20 year period, thinking that the results will transfer over.

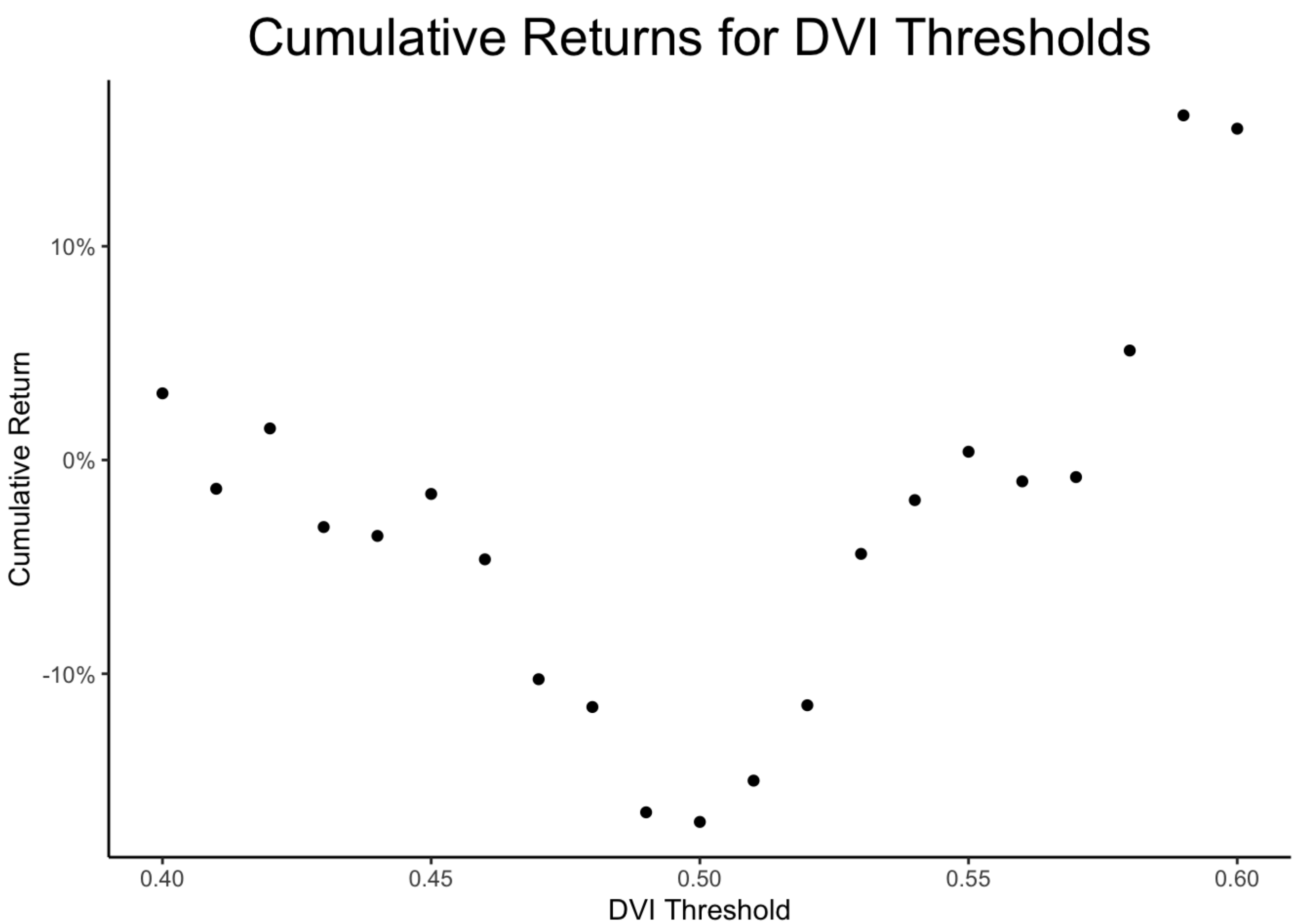
Function 3 Output

```
fun3 <- function3('JNJ','20140101','20171231',0.4,0.6,0.01)
kable(fun3, align = "lllll", caption = "Back End Testing Data Summary for JNJ, for Vario
us DVI Thresholds")
```

Back End Testing Data Summary for JNJ, for Various DVI Thresholds

DVIThreshold	TotalLong	TotalShort	CumulativeReturn
0.40	250	400	0.0311912
0.41	256	394	-0.0134603
0.42	263	387	0.0147721
0.43	269	381	-0.0313655
0.44	274	376	-0.0355116
0.45	281	369	-0.0158754
0.46	286	364	-0.0464695
0.47	292	358	-0.1025563
0.48	301	349	-0.1155554
0.49	306	344	-0.1648233
0.50	312	338	-0.1693098
0.51	320	330	-0.1499939
0.52	324	326	-0.1147299
0.53	336	314	-0.0439147
0.54	347	303	-0.0187680
0.55	354	296	0.0038693
0.56	360	290	-0.0099883
0.57	366	284	-0.0079930
0.58	373	277	0.0512157
0.59	379	271	0.1612314
0.60	389	261	0.1550146

```
# graphing the data from function 3
ggplot(fun3, aes(x = DVIThreshold, y = CumulativeReturn)) +
  geom_point() +
  theme(plot.title = element_text(hjust = 0.5, size = 20), panel.grid.major = element_bl
ank(),
        panel.grid.minor = element_blank(), panel.background = element_blank(), axis.lin
e = element_line(colour = "black")) +
  labs(x = 'DVI Threshold', y = 'Cumulative Return',
       title = "Cumulative Returns for DVI Thresholds") +
  scale_y_continuous(labels = scales::percent)
```



Function 3 Analysis

The table above shows us a return summary for each DVI threshold tested for the JNJ stock between the given dates. The thresholds range from .4 to .6, and increase in increments of .01. The values in the cumulative return column are also shown visually in the graph just above. Based on the graph, we can see that the DVI thresholds which gave us the best performance were the ones farthest away from .5. The DVI threshold which resulted in the greatest return is .59 with a return of 16.123%. As the thresholds get closer to .5, the cumulative returns generally decrease. These are very interesting results given that a DVI threshold of .5 is typically the standard, and it would be helpful to widen the range of DVI values and see if this trend increases at the extremes. Once again, this test was very limited as it ran over one stock for a short period of time, and changing those parameters could have very large impacts on our results. We also could make the DVI increment smaller if we wanted to see changes in cumulative returns more precisely.

Conclusion

Overall, the functions in this report were able to successfully perform back testing based on a DVI threshold. The simulation results outputted were generally poor and resulted in negative returns. However, these function only ran on one stock, for one general date period. There are many parameter options that could change the results, such as the dates, the range of the dates, the stock itself, the number of stocks provided, and the DVI threshold. These functions could also have used a different trading rule such as SMA. Basically, back testing is open to high amounts of tweaking. However, as mentioned several times in this report, this leaves back testing susceptible to bias from investors as cherrypicking stocks, date ranges, and thresholds to get very positive simulated results will not translate well when the strategy is implemented in real life.

Works Cited

Analyzing the DVI Indicator. Quintuitive. (2015, May 5). http://www.quintuitive.com/2013/11/30/analyzing-the-dvi-indicator/ (http://www.quintuitive.com/2013/11/30/analyzing-the-dvi-indicator/).
Hayes, A. (2021, June 10). Behavioral Finance Definition. Investopedia. https://www.investopedia.com/terms/b/behavioralfinance.asp (https://www.investopedia.com/terms/b/behavioralfinance.asp).
Tachev, V. (2021, March 5). Backtesting in Trading - A guide on how to Backtest a Trading Strategy. Earn2Trade Blog. https://blog.earn2trade.com/how-to-backtest/ (https://blog.earn2trade.com/how-to-backtest/).