Project 9: Strategy Evaluation

Yu-Wei Lai
ylai67@gatech.edu

1. INTRODUCTION

For this project, I am implementing a manual strategy where I develop trading rules, and a strategy learner, which uses machine learning methods to come up with trading actions. For both strategies, I am looking at three specific indicators, SMA (Simple Moving Average), MACD (Moving Average Convergence Divergence) histogram calculated by MACD line minus signal line, and BBP (Bollinger Bands Percentage), which will give insights on how to choose among three trading actions, "short", "out", or "long". The goal is to come up with trading solutions, to gain insights on both strategies, such as which works better or whether they work better than the benchmark solution, and to study other factors that played a role in this situation.

2. INDICATOR OVERVIEW

2.1 SMA

SMA is a commonly used indicator that involves calculating the average price of an asset over a specified period. For the manual strategy, I only calculated SMA when window size is 20, and for the strategy learner, I calculated SMA_10, SMA_30, and SMA_100, which stands for SMA when window size is 10, 30, and 100 days respectively.

2.2 MACD Histogram

MACD is another commonly used indicator and for this project, I have chosen to first calculate MACD by subtracting the longer period EMA (26 days) from the shorter EMA (12 days). Then, I calculate the signal, which is a 9-period EMA of the MACD line. Finally, I subtract signal from MACD to get MACD histogram, which is the indicator I used for both strategies.

2.3 BBP

BBP is an indicator derived from the commonly used BB (Bollinger Bands) indicator, which is calculated from SMA. For the manual strategy, I only calculated BBP when window size is 20, and for the strategy learner, I calculated BBP_10, BBP_30, and BBP_100, which stands for BBP when window size is 10, 30, and 100 days respectively.

3. MANUAL STRATEGY

3.1 Creating a Signal

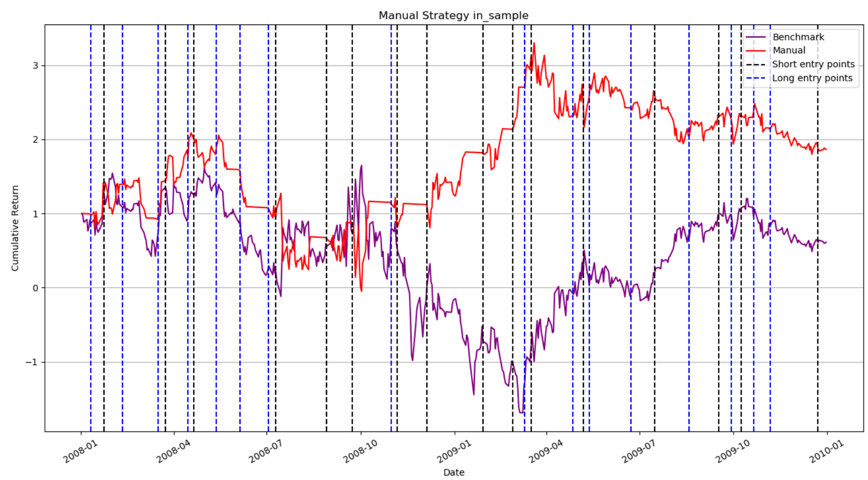
I combined the three indicators to create an overall signal using the following logic: For each day, I calculate three scores which are SMA score, MACD score, and BBP score, and by adding them together, I get a total score for that day. If the total score is greater than 3, I enter the long position; if it is less than -3, I enter the short position. If the total score is between -3 and 3, I enter the out position.

To calculate the SMA score, I first calculate the SMA value at the current date. If SMA is less than current price, I set SMA score as 1. If SMA is greater than current price, I set SMA score as -1. If they are equal, I set SMA score as 0 and therefore does not influence decision making. To calculate MACD score, I first calculate the MACD histogram value at the current date. If MACD histogram is greater than 0, I set the MACD score as -5; if MACD histogram is less than 0, I set the MACD score as 2. If MACD histogram is 0, I set the MACD score as 1. To calculate the BBP score, I first calculate the BBP value at the current date. If BBP is greater than 100, I set the BBP score as -1. If the BBP is less than 0, I set the BBP score as 1. If the BBP is between 0 and 100, I set the BBP score as 0.

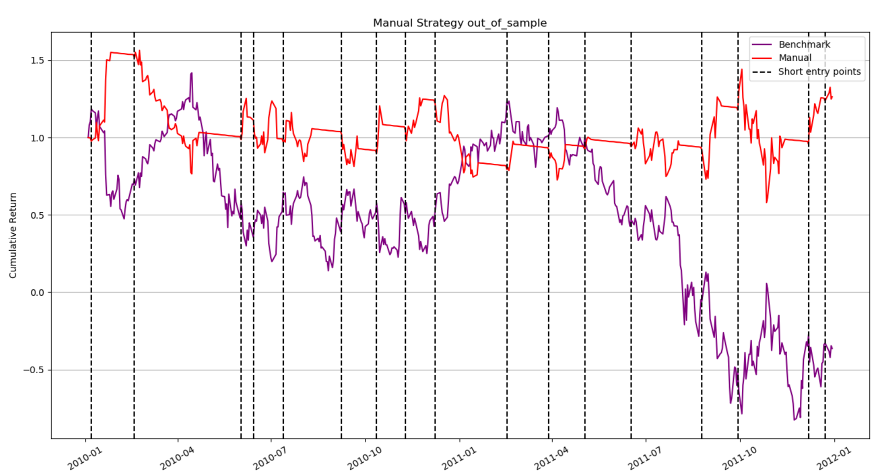
The boundaries, or the decisions I made on when to enter and exit certain positions, are based on the common boundaries for each indicator to reveal a buying or selling signal. By calculating individual scores for each of the indicators and assigning different scores for different ranges of indicator value, this strategy allows all three indicators to play a role in determining the position. Finally, the scores are assigned from a series of trial-and-error attempts, where I was finally able to find the set of values for each indicator that works best for my strategy and performs significantly better than the benchmark strategy where one simply buys and holds.

3.2 Comparing Performance in In-Sample and Out-of-sample Data

From the diagram, we can observe that the manual strategy for in-sample data (2008-01-01 to 2009-12-31) performs better than the benchmark most of the time, especially since the end of 2008 towards the end of the time period. As opposed to the benchmark strategy, the manual strategy proves to be profitable at the end.



From the diagram, we can observe that the manual strategy for out-of-sample data (2010-01-01 to 2011-12-31) performs better than the benchmark most of the time, except two items around April 2010 and in the beginning of 2011. At the end of the time period, the manual strategy proves to be much more profitable than the benchmark strategy. Manual strategy also has much lower standard deviation in both cases.



table_in_sample

	cr	adr	sddr
manual	0.8624831707549640	-0.012341553133966700	0.5317222606134040
benchmark	-0.3862947426436850	0.10182647815895000	5.470004686728080

table_out_of_sample

	cr	adr	sddr
manual	0.2639426229098630	0.002234213317259820	0.06010092767377120
benchmark	-1.3637092900864700	-0.036879079754878900	1.4423197623023300

3.2 Evaluating Strategy in Out-of-sample Data

In out-of-sample data, the manual strategy has continued to make profits, as opposed to the benchmark strategy that suffers from heavy losses. In addition, the manual strategy also has a positive average daily return and the standard deviation continues to be much lower than that of the benchmark strategy. This means that the manual strategy I have developed using in-sample data performs well with out-of-sample data as well. These differences, such as the cumulative return being higher for in-sample data, between in-sample and out-of-sample outcomes are likely due to the fact that the manual strategy is specifically developed with in-sample data, and the fact that standard deviation of daily returns is lower in out-of-sample data than in-sample is likely because prices are less volatile in the out-of-sample period, as the standard deviation for the benchmark strategy is also reduced, although not as significantly.

4. STRATEGY LEARNER

4.1 Framing the Trading Problem as A Learning Problem

To frame the trading problem as a learning problem, I applied the Q-Learner where the reward serves as an incentive for the learner to learn a trading policy and the indicators can be used to calculate a score that defines the "state" of the learner. During the learning stage, I follow the following steps: 1. Compute the current state, which includes holding and score from indicators. 2. Compute the reward for the last action. 3. Query the learner with the current state and reward to get an action while updating the Q table. 4. Implement the action the learner returned and update portfolio value. 5. Repeat the above steps for all the days in the in-sample time period. During the testing stage, I follow the exact steps to get actions and update portfolio values, but without updating the Q table, as training is not meant to happen during the testing stage.

4.2 Hyperparameters

These are the hyperparameters I used to instantiate the Q Learner:

```
self.learner = ql.QLearner(num_states=384,\
            num_actions = 3, \
            alpha = 0.2, \
            gamma = 0.9, \
            rar = 0.9, \
            radr = 0.99, \
            dyna = 100, \
            verbose=False)
```

num_states is set to be 384, since there are seven parameters (SMA 10 days, SMA 30 days, SMA 0 days, MACD histogram, BBP 10 days, BBP 30 days, and BBP 100 days) as well as the current position that is taken into consideration. Each of the seven parameters has a score of 0 to 1, and adds 2**n*score to the" state", while the current state adds 2*128, 1*128, or 0*128 to "state". Therefore, "state" can range from 0 to 383, so num_states equals 384. num_actions is set to be 3, as there are three positions (0 for short, 1 for out, 2 for long)

4.3 Discretizing Data

In order to discretize data, I first calculate 7 parameters: SMA 10 days, SMA 30 days, SMA 100 days, MACD histogram, BBP 20 days, BBP 30 days, and BBP 100 days. Then, I discretize them by assigning them values of 0 or 1. For example, on days when SMA_10 is greater than current price, the discretized SMA_10 is set to be 0 and 1 otherwise. The same thing is done for all SMAs. On days when MACD is greater than 0, the discretized MACD is set to be 1 and 0 otherwise. On days when BBP_10 is greater than 1, the discretized BBP is set to be 1 and 0 otherwise. The same thing is done with all BBPs. The position is set to be 0 for short, 1 for out, and 2 for long.

Finally, the state is calculated by the following:

```
BBP_50['BBP'] * 64 + SMA_30['SMA'] * 32 + SMA_10['SMA'] * 16 +
SMA_100['SMA'] * 8 + MACD * 4 + BBP_20['BBP'] * 2 +
BBP_100['BBP']
```

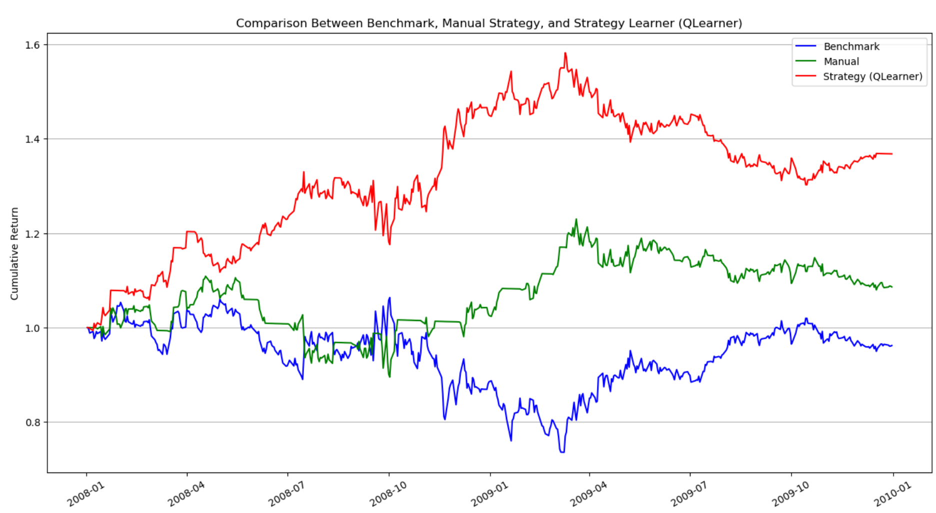
Therefore, the state can be an integer from 0 to 383, and each value for state unique defines the current positions as well as the value of discretized indicators.

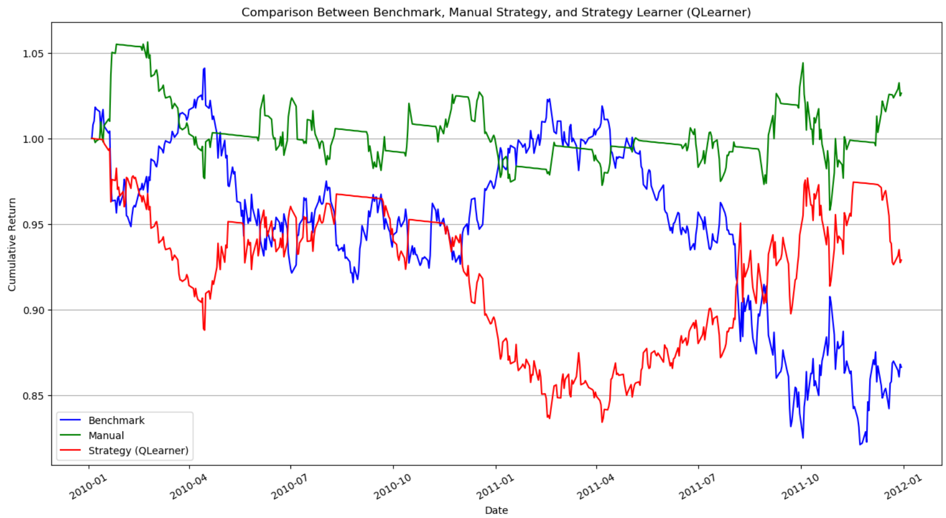
5. EXPERIMENT 1

5.1 Detailed Description of Experiment

Experiment 1 compares the manual strategy and strategy learner (Q Learner) in two situations, during the in-sample period (2008-01-01 to 2009-12-31) and out-of-sample period (2010-01-01 to 2011-12-31). The starting value is set to be 100000 like everywhere else in the project; commission is set to be 9.95 and impact is set to be 0.005. My initial assumption, as well as hypothesis, was that the strategy learner might not be able to perform as well as the manual strategy during the in-sample period, but it should be able to perform better than the manual strategy in the out-of-sample period. The reasoning I used to come up with this hypothesis is fairly simple: Manual strategy is developed for the in-sample period only based on trying different parameters and observing their outcomes; therefore, it may not work as well as the strategy learner which is supported by learning; also, as the number of experiences increase as time goes on, the strategy learner should have more data to base its decisions on, which makes it "smarter" in a sense and more advantageous during the out-of-sample period.

5.2 Analysis of Outcome





It is evident that the results were entirely opposite to my previous hypothesis. In the in-sample period, the strategy learner ends up performing better than the manual strategy. This is likely due to the fact that as time goes on in the in-sample data time period, it gains sufficient experiences, which might have caused the model to be slightly overfitted. In addition, the manual strategy ends up performing better than the strategy learner in the out-of-sample period, which is likely also due to the overfitted model and that the manual strategy is simply developed to work well with the chosen stock.

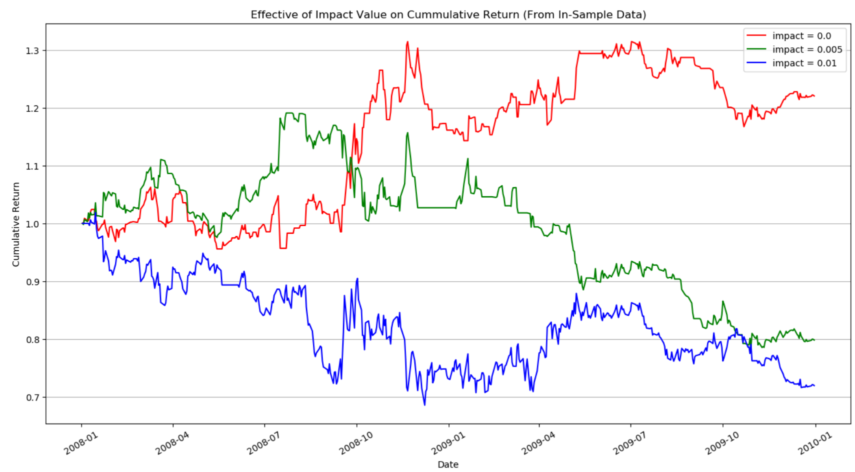
For this reason, this relative result can be expected every time with in-sample data, as learning or "memorizing" experiences in in-sample period allows the strategy learner to consistently outperform manual strategy with in-sample data.

6. EXPERIMENT 2

6.1 How Changing the Value of Impact Affects In-Sample Trading Behavior

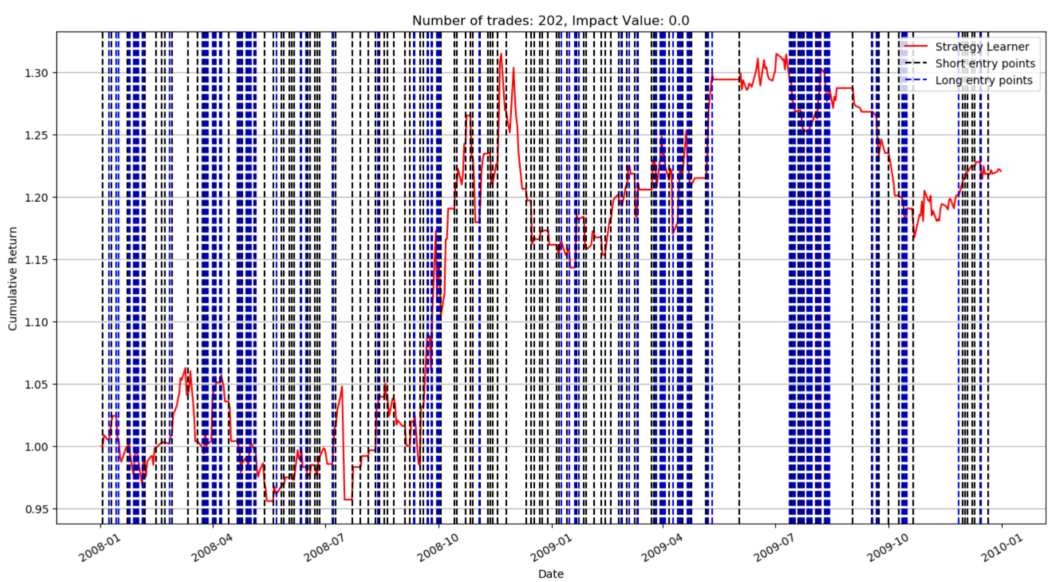
Experiment 2 is conducted on strategy learner during the in-sample period and seeks to explore the role in impact in in-sample trading behavior, where the strategy learner takes the same hyperparameters as in the previous section. Impact is the amount the price moves against the trader compared to the historical data at each transaction. My assumption is that lower impact will lead to better performance of the strategy learner and a greater number of trades. This is intuitive, as the fact that impact causes the price to move against the trader's desired direction means that it can erode profits and/or exacerbate losses. The two metrics I have used to test this hypothesis is cumulative return and number of trades.

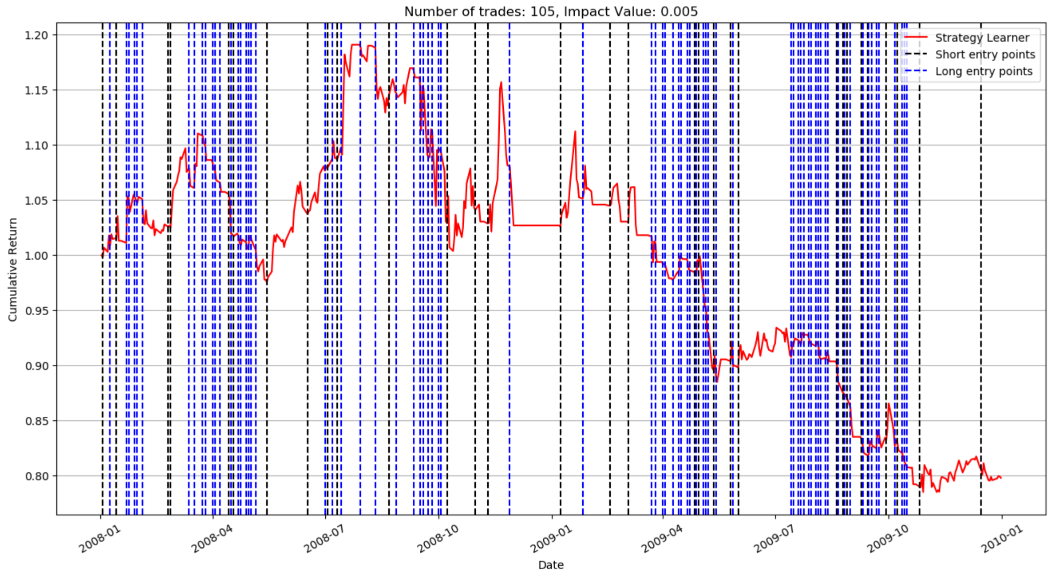
6.1.1 First Metric: Cumulative Return

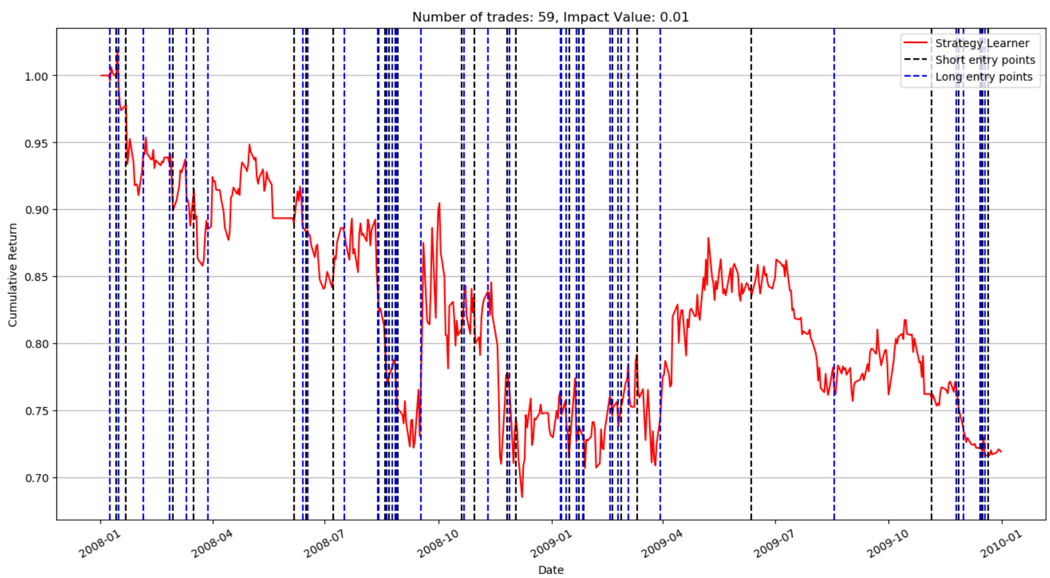


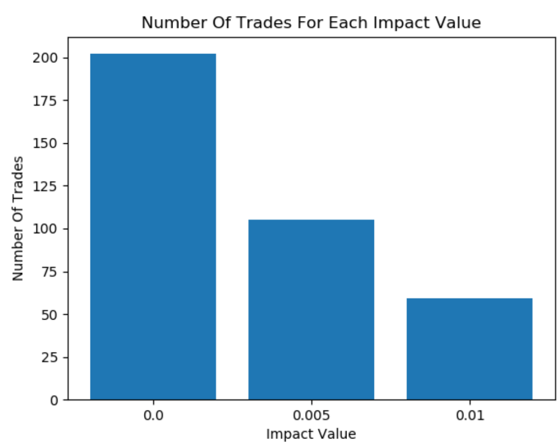
For the first part of the experiment, I calculated the normalized portfolio value at each date during the in-sample period using the strategy learner for impact = 0.0, 0.005, and 0.01 respectively. It can be seen that the cumulative return is highest when impact = 0.0 and lowest when impact = 0.01, with when impact = 0.05 in the middle. This result is closely in line with my previous hypothesis.

6.1.2 Second Metric: Number Of Trades




Number of trades: 105, Impact Value: 0.005
Strategy Learner
Short entry points
Long entry points
Cumulative Return
Date
1.20
1.15
1.10
1.05
1.00
0.95
0.90
0.85
0.80
2008-01
2008-04
2008-07
2008-10
2009-01
2009-04
2009-07
2009-10
2010-01


Number of trades: 59, Impact Value: 0.01
Strategy Learner
Short entry points
Long entry points
Cumulative Return
Date
1.00
0.95
0.90
0.85
0.80
0.75
0.70
2008-01
2008-04
2008-07
2008-10
2009-01
2009-04
2009-07
2009-10
2010-01


Number Of Trades For Each Impact Value
Number Of Trades
Impact Value
200
175
150
125
100
75
50
25
0
0.0
0.005
0.01

For the second part of the experiment, I calculated number of trades during the in-sample period using the strategy learner for impact = 0.0, 0.005, and 0.01 respectively. In particular, at each date, I determine the trading position using the strategy learner; if the position is “long” or “short” and not “out”, it is considered making a trade and I add 1 to the number of trades. It turns out that there are 202 trades when impact = 0, 105 trades when impact = 0.005, and 59 trades when impact = 0.10. This result is in line with my earlier hypothesis.