

LaGuardia Community College

Trend analysis using bit strings

Introduction

The purpose of this paper is to illustrate how we can use bit strings to analyze trends in different markets and to try to take advantage of those trends to make predictions to, ultimately, make money.

Basically, we are going to try to invest in a market with the help of an algorithm, which is nothing more than a set of instructions, that parses data and looks for patterns that we specify beforehand. These patterns represent market fluctuations in the context of the data being analyzed.

Our algorithm is going to look for one pattern at a time and when it finds it, it triggers instructions to buy or sell what we are trading, allowing us to win or lose money. These instructions are also predefined by us.

We want to find out if there is one pattern of fluctuations that excels in all the markets. If we find said pattern, we can apply it to real world data and make a lot of money.

Algorithms

An algorithm is a step-by-step list of directions that need to be followed to solve a problem. The instructions should be simple enough so that each step can be done without thinking about it. Algorithms are often used to describe how a computer might solve a problem. But there are algorithms in the real world too. A recipe can be a type of algorithm. It tells what ingredients are needed to make the dish and what steps to follow. If the recipe tells exactly what to do without too much confusion, then it is an algorithm.

There is usually more than one way to solve a problem. There may be many different recipes to make a certain dish which look different but end up tasting the same when all is said and done. The same is true for algorithms. However, some of these ways will be better than others. If a recipe needs lots of complicated ingredients that you do not have, it is not as good as a simple recipe. When we look at algorithms as a way of solving problems, often we want to know how long it would take a computer to solve the problem using a particular algorithm. When we write algorithms, we like our algorithm to take the least amount of time so that we can solve our problem as quickly as possible.

String searching algorithms

String searching algorithms, sometimes called string matching algorithms, are an important class of string algorithms that try to find a place where one or several strings (also called patterns) are found within a larger string or text. In the context of this paper, we are going to analyze see how searching for these patterns help us to analyze and study markets.

Japanese Candlesticks

The Japanese began using technical analysis to trade rice in the 17th century. While this early version of technical analysis was different from the US version initiated by Charles Dow around 1900, many of the guiding principles were very similar:

- The "what" (price action) is more important than the "why" (news, earnings, and so on).
- All known information is reflected in the price.
- Buyers and sellers move markets based on expectations and emotions (fear and greed).
- Markets fluctuate.

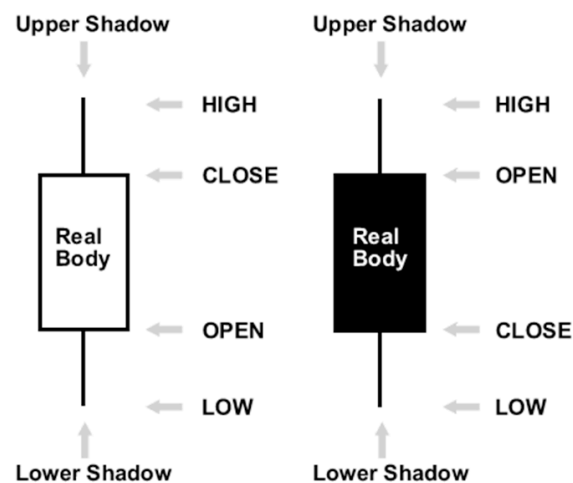
Trend analysis using bit strings

- The actual price may not reflect the underlying value.

According to Steve Nison, candlestick charting first appeared sometime after 1850. Much of the credit for candlestick development and charting goes to a legendary rice trader named Homma from the town of Sakata. It is likely that his original ideas were modified and refined over many years of trading eventually resulting in the system of candlestick charting that we use today.

Formation

In order to create a candlestick chart, you must have a data set that contains open, high, low and close values for each time period you want to display. The hollow or filled portion of the candlestick is called "the body" (also referred to as "the real body"). The long thin lines above and below the body represent the high/low range and are called "shadows" (also referred to as "wicks" and "tails"). The high is marked by the top of the upper shadow and the low by the bottom of the lower shadow. If the stock closes higher than its opening price, a hollow candlestick is drawn with the bottom of the body representing the opening price and the top of the body representing the closing price. If the stock closes lower than its opening price, a filled candlestick is drawn with the top of the body representing the opening price and the bottom of the body representing the closing price.



Compared to traditional bar charts, many traders consider candlestick charts more visually appealing and easier to interpret. Each candlestick provides an easy-to-decipher picture of price action. Immediately a trader can see compare the relationship between the open and close as well as the high and low. The relationship between the open and close is considered vital information and forms the essence of candlesticks. Hollow candlesticks, where the close is greater than the open, indicate buying pressure. Filled candlesticks, where the close is less than the open, indicate selling pressure.

Candlesticks can be used for any time frame, whether it be one day, one hour, 30-minutes - whatever you want! They are used to describe the price action during the given time frame.

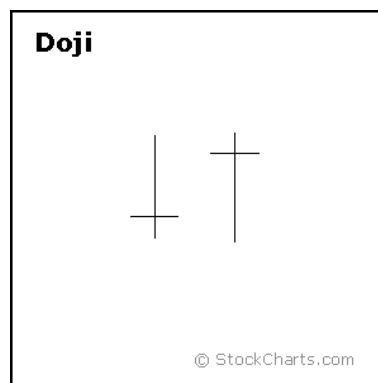
Candlesticks are formed using the open, high, low, and close of the chosen time period.

If the close is above the open, then a hollow candlestick (usually displayed as white) is drawn. If the close is below the open, then a filled candlestick (usually displayed as black) is drawn. The hollow or filled section of the candlestick is called the "real body" or body. The thin lines poking above and below the body display the high/low range and are called shadows. The top of the upper shadow is the "high". The bottom of the lower shadow is the "low".

For more on shadows, see appendix A.

Bodies

Long white candlesticks show strong buying pressure. The longer the white candlestick, the further the close is above the open. This indicates that prices increased considerably from open to close and buyers were aggressive.



Long black (filled) candlesticks show strong selling pressure. The longer the black candlestick, the further the close is below the open. This indicates that prices fell a great deal from the open and sellers were aggressive. In other words, the bears were grabbing the bulls by their horns and body-slammng them.

Doji

Doji are important candlesticks that provide information on their own and as components of in a number of important patterns. Doji form when a security's open and close are virtually equal. The length of the upper and lower shadows can vary and the resulting candlestick looks like a cross, inverted cross or plus sign. Alone, doji are neutral patterns. Any bullish or bearish bias is based on preceding price action and future confirmation. The word "Doji" refers to both the singular and plural form.

Ideally, but not necessarily, the open and close should be equal. While a doji with an equal open and close would be considered more robust, it is more important to capture the essence of the candlestick. Doji convey a sense of indecision or tug-of-war between buyers and sellers. Prices move above and below the opening level during the session, but close at or near the opening level. The result is a standoff.

For more on Doji Candlesticks, see appendix B.

Going long vs going short

Financial trading uses some phrases that are unique, including the terms that describe whether a trade was entered by buying or selling:

Long

When a day trader is in a long trade, they have entered a trade by buying a contract (or group of shares, or forex lot), and are hoping that the price will go up. Day traders will often use the terms buy and long interchangeably. Similarly, some trading software will have a trade entry button marked "Buy", where another might have a trade entry button marked "Long". The term long is often used to describe an open position, as in "long the YM", which would indicate that a trader had an active trade where they had bought a contract of the YM (Dow Jones) futures market.

Short

When a day trader is in a short trade, they have entered a trade by selling a contract (or group of shares, or forex lot), and are hoping that the price will go down. Day traders will often use the terms sell and short interchangeably. Similarly, some trading software will have a trade entry button marked "Sell", where another might have a trade entry button marked "Short". The term short is often used to describe an open position, as in "short the ZG", which would indicate that a trader had an active trade where they had sold a contract of the ZG (gold 100 troy ounce) futures market.

Methodology

For this project we are going to test different patterns on different markets and see which one performs best. The code is designed so that we can easily specify the pattern that is going to be looked for in our database of values. We can also specify if we want to go long or short.

For our calculations, we are going to analyze the following five markets:

- EURUSD (Euro, US Dollar)
- USDCAD (US Dollar, Canadian Dollar)
- USDJPY (US Dollar, Japanese Yen)
- XAU (Gold, US Dollar)
- SPX (S&P 500)

For more on the methodology, please refer to Appendix C.

Java Code

```
import java.io.File;
import java.io.FileNotFoundException;
import java.text.DecimalFormat;
import java.util.Scanner;
```

```
public class bitString {
```

Our main method consists only of two calls; one to read() the data and one to display() the results of that data.

```
    public static void main(String[] args) throws FileNotFoundException {
        read();
        display();
    }
```

Here we chose to either go long (private static int l=1) or short (private static int l=0).

```
private static int l=1; // 1= long 0=short
```

We also specify in the array p[] which pattern we are looking to match—like so: private static int[]p = {0, 0, 0, 0, 0, 0, 1, 1} or private static int[]p = {1, 1, 1, 1, 1, 1, 0, 0}.

```
private static int[]p = {0,0,0,0,0,0,1,1}; //00000011 is long, 11111100 is short
private static int size=0;
```

We create the arrays that will hold the values of the of cells in our CSV file.

```
private static double[] open=new double[1000000];
private static double[] close=new double[1000000];
private static double[] high=new double[1000000];
private static double[] low=new double[1000000];
private static int[] lowerLow=new int[1000000];
private static int[] higherHigh=new int[1000000];
private static int[] color=new int[1000000];
```

We also create counters to keep track of the number of wins and loses.

```
private static int in=0;
private static int out=0;
private static int match=0;
private static int win=0;
private static int lose=0;
private static double totalgain=0;
private static double totallose=0;
```

Always a good idea to throw an exception when dealing with manually inserting files.

```
public static void read() throws FileNotFoundException{
```

Creates the Scanner and starts parsing the cells into the arrays until the end of the file.

```
    Scanner in = new Scanner(new File("/Users/johan/EURUSD_hour.csv"));
    in.nextLine();
    while(in.hasNextLine()){
        String[] n= in.nextLine().split(",");
        open[size]=Double.parseDouble(n[3]);
        low[size]=Double.parseDouble(n[4]);
        high[size]=Double.parseDouble(n[5]);
        close[size]=Double.parseDouble(n[6]);
        lowerLow[size]=Integer.parseInt(n[7]);
        higherHigh[size]=Integer.parseInt(n[8]);
        size++;
    }
```

And we of course close the file and call the method search() to start looking for patterns.

```
        in.close();
        search();
    }
```

```
public static void search(){
    for (int i=0; i<size-4; i++){
```

Searches until it finds a 0, 0 which corresponds to the beginning of the pattern.

```
        if(lowerLow[i]==p[0] && higherHigh[i]==p[1]){
            int j=2;
            int k=i;
```

Next we check the next couple to see if it matches 0, 0, 0, 0, 1,1 which is the end of the pattern.

```
            while(j<p.length){
                if(lowerLow[k+1]==p[j] && higherHigh[k+1]==p[j+1]){
                    k++;
                    j=j+2;
                }
                else break;
            }
```

if it matches, we enter at the open of the next period. If it doesn't we break out of the loop and search for another 0,0.

```
            if(j>=p.length ){
                in=i+((p.length/2));
                match++;
                out=exit();
```

We then keep track of how much we make or lose. This is for when we go long.

```
                if(l==1){
```

Trend analysis using bit strings

```
double gain = open[in]- open[out];
System.out.println("in: " + (in+2) + " " +open[in]+ "\tout: "
+(out+2) + " " +open[out] + " gain: "+gain);
```

And increment the appropriate counters.

```

        if(gain>0){
            win++;
            totalgain=totalgain+gain;
        }
        else {
            lose++;
            totallose=totallose+gain;
        }
    }
}

```

Finally, we do the same for when we go short. First we log the gain or loss.

```

    }
    else{
        double gain = open[out]- open[in];
        System.out.println("in: " + (in+2) + " " +open[in]+ "\tout: "
+ (out+2) + " " +open[out] + " gain: "+gain);
    }
}

```

And again increment the counters.

```
if(gain>0){  
    win++;  
    totalgain=totalgain+gain;  
}  
else {  
    lose++;  
    totallose=totallose+gain;  
}  
}  
}  
}  
}
```

For our exit criteria, we chose to go the simple way: a method that exits the trade when we hit lower lows for the long and higher lows for the short.

```
static int exit(){
    int i=in;
    if(l==1){
        while(lowerLow[i]!=0)i++; // exit when lower low
    }
    else if(l==0){
        while(lowerLow[i]!=1)i++; // exit when higher low
    }
}
```



```

    }
    return i+2;
}

```

Lastly, the method that displays all the results. We also create a DecimalFormat object to get a nicer-looking output.

```

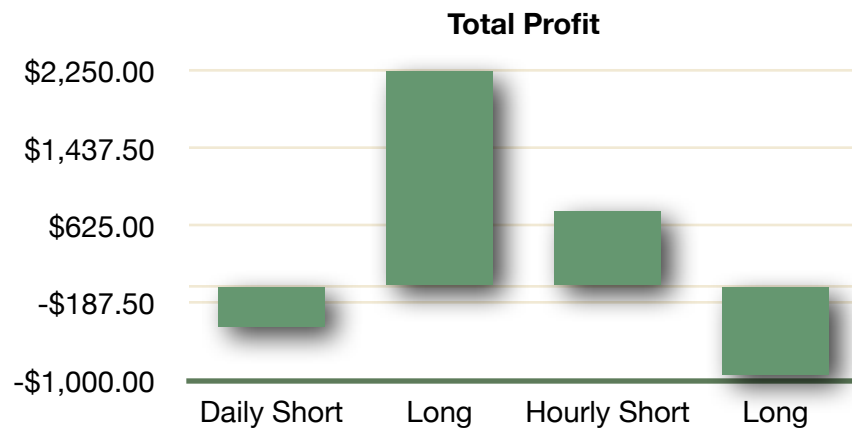
static void display(){
    System.out.println();
    System.out.println("matches: " +match);
    System.out.println("wins: " +win+ "\twin %: " +(win*100/match));
    System.out.println("loses: " +lose +"\tlose %: " +lose*100/match);
    System.out.println("Total Gain: "+totalgain);
    System.out.println("Total Loss: "+totallose);
    System.out.println("Profit: " +(totalgain+totallose));
    DecimalFormat df = new DecimalFormat("##.#####");
    System.out.println("Total pips per match: " +
(df.format((totalgain+totallose)/(match))));
}
}

```

Results

EURUSD

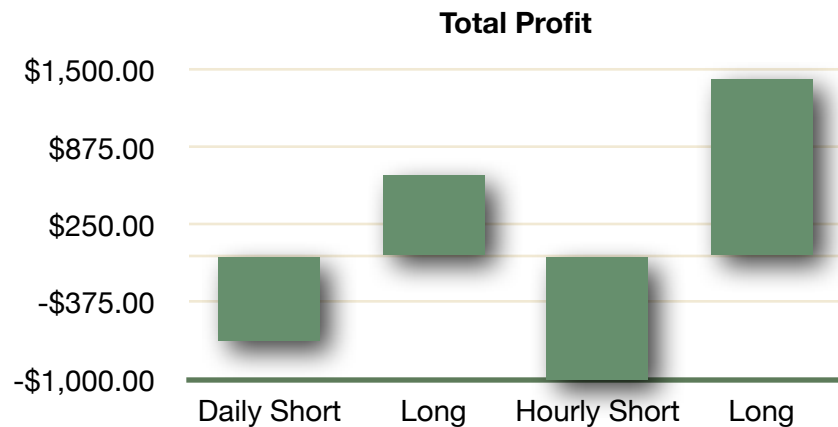
	Daily		Hourly	
	Short	Long	Short	Long
Matches	71	69	1454	1377
Wins	41	43	780	737
Loses	30	26	674	640
Win %	57.75%	62.32%	53.65%	53.52%
Lose %	42.25%	37.68%	46.35%	46.48%
Total Gain	3237	4975	15123	13420
Total Loss	-3661	-2298	-14335	-14357
Pips Per Match	-\$5.972	\$38.797	\$0.542	-\$0.680
Total Profit	-\$424.00	\$2,677.00	\$788.00	-\$937.00



Here we see that the most profitable trades by almost three times the second most profitable set lie on the daily pattern when we choose to go long. The algorithm encountered the pattern sixty nine times, executed the trade and made almost 38.9 average pips per trade. Even though the daily short pattern won 58% of the times in the 71 matches, the average pips per trade was -5.97, which means that it lost a big amount of pips in the 42% left.

USDCAD

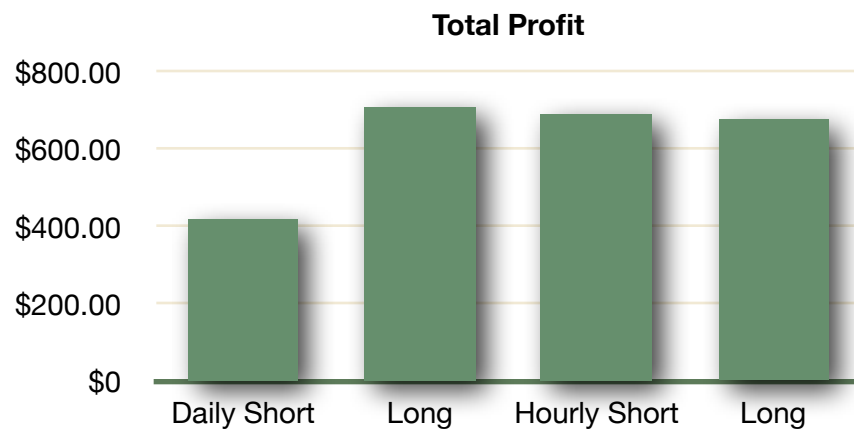
	Daily		Hourly	
	Short	Long	Short	Long
Matches	73	76	1379	1398
Wins	40	38	711	771
Loses	33	38	668	627
Win %	54.79%	50.00%	51.56%	55.15%
Lose %	45.21%	50.00%	48.44%	44.85%
Total Gain	2675	3141	13115	13491
Total Loss	-3358	-2490	-23641	-12058
Pips Per Match	-\$9.356	\$8.566	-\$7.633	\$1.025
Total Profit	-\$683.00	\$651.00	-\$10,526.00	\$1,433.00



Going short in the hourly period produced a massive loss using the pattern 1, 1, 1, 1, 1, 1, 0, 0. We had a lot of hits on the pattern for the hourly data. When going short we had 1379 patterns and 1398 when going long. Apparently, entering the trades by selling and waiting for the price to be low to buy didn't pay off. On the other hand, buying first and waiting for the price of the stock to go up happened very frequently and thus we did quite well.

USDJPY

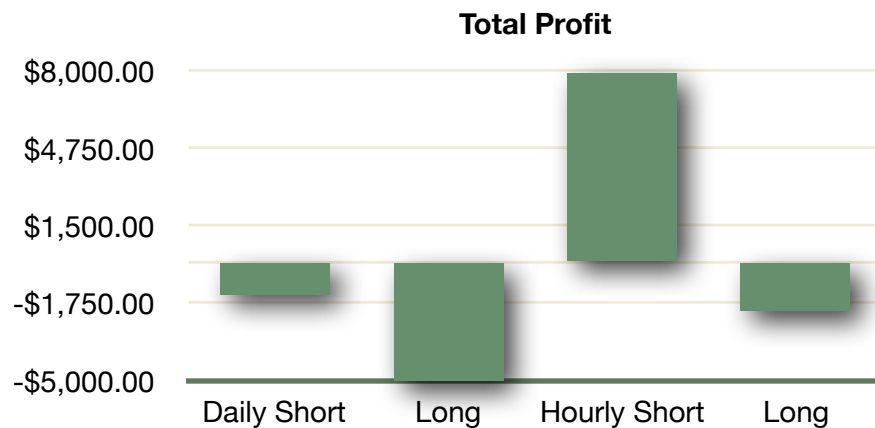
	Daily		Hourly	
	Short	Long	Short	Long
Matches	74	59	1473	1445
Wins	46	37	769	779
Loses	28	22	704	666
Win %	62.16%	62.71%	52.21%	53.91%
Lose %	37.84%	37.29%	47.79%	46.09%
Total Gain	3494	2518	12385	12615
Total Loss	-3074	-1808	-11692	-11937
Pips Per Match	\$5.676	\$12.034	\$0.470	\$0.469
Total Profit	\$420.00	\$710.00	\$693.00	\$678.00



Japanese Yen turned out to be a very nice investment, yielding generous profits in the daily as well as in the hourly data distribution when going short or long. The most profitable one was the daily when going long with an average gain of \$2518 per pattern and 12.03 average pips per match. The number of hits of the pattern we had on this dataset was 59 versus 74 on the short pattern. Going long produced less loss on the long trades (\$1808) when compared to the short trades (\$3074), which explains the difference in the total profit.

XAU

	Daily		Hourly	
	Short	Long	Short	Long
Matches	83	48	1339	1146
Wins	53	23	723	597
Loses	30	25	616	549
Win %	63.86%	47.92%	54.00%	52.09%
Lose %	36.14%	52.08%	46.00%	47.91%
Total Gain	51595	32541	154075	136484
Total Loss	-52966	-47646	-146136	-138540
Pips Per Match	-\$16.518	-\$314.688	\$5.929	-\$1.794
Total Profit	-\$1,371.00	-\$15,105.00	\$7,939.00	-\$2,056.00



It is clear that gold is a very dangerous market to invest in, at least with the data and the pattern we tested. The difference is incredibly high between the most profitable pattern versus the less profitable one. More than \$23,000 (\$7,939 - \$15,105) is not something I would like to bet in. In the daily long there weren't that many pattern repetition—we didn't even hit the 50 mark— but we sure lost a lot of money every time we lost a trade. The average pip per match lost was 314.68. Meanwhile, we won 54% of the 1339 times the pattern appeared in the hourly dataset when going short, yielding a final profit of \$7,939.

Conclusions

From the data we can certainly say that, as expected, there is not one algorithm that excels in all of the markets, but they are all entirely dependent of the market movements. Sometimes those movements played to our advantage and sometimes they don't.

The market that we would have liked to invest in is the US Dollar Japanese Yen. The percentage of winning versus losing was very comfortable and let us exit any period on any pattern with a nice stake of profit. Of course, in a real market we can't predict that the same would happen because the world economy is always moving in different directions. What we can certainly assert, though, is that we do not want to get involved in high risk markets like Gold.

Although it seemed like quite a challenge at the beginning of the term, dealing with bit string and pattern searching was easier than expected. Looking for these patterns inside thousands of numbers may seem quite complicated to a human but by writing code we let the computer do all the work so that we can focus only in analyzing the data spitted out by the machine (although we could've also let the computer analyze the data, but that is not inside the scope of this paper). The results were pretty clear, objective and informative.

Markets fluctuate according to dozens of factors and are very hard to predict. By abstracting away and disregarding a lot of these factors, we get a decent understanding of how they work and it makes it easy to analyze data that goes into them and see how we could've performed under specific scenarios.

References

What is a Japanese Candlestick? BabyPips.com LLC, 2012. Web. 23 Jan. 2012.

Williams, Peter. Data Structures. University of Sussex, 2006. Web. 30 Jul. 2006.

Yahoo! Finance. Yahoo! Inc., 2012. Web. 1 Jun. 2012.

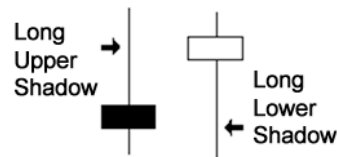
FXHISTORICALDATA.COM. FXHISTORICALDATA.COM, 2012. Web. 1 Jun. 2012.

Appendix A

Japanese Candlesticks — Shadows

The upper and lower shadows on candlesticks provide important clues about the trading session.

Long Shadows



Upper shadows signify the session high. Lower shadows signify the session low.

Candlesticks with long shadows show that trading action occurred well past the open and close.

Candlesticks with short shadows indicate that most of the trading action was confined near the open and close.

If a candlestick has a long upper shadow and short lower shadow, this means that buyers flexed their muscles and bid prices higher, but for one reason or another, sellers came in and drove prices back down to end the session back near its open price.

If a candlestick has a long lower shadow and short upper shadow, this means that sellers flashed their washboard abs and forced price lower, but for one reason or another, buyers came in and drove prices back up to end the session back near its open price.

Appendix B

Doji and Trend

The relevance of a doji depends on the preceding trend or preceding candlesticks. After an advance, or long white candlestick, a doji signals that the buying pressure is starting to weaken. After a decline, or long black candlestick, a doji signals that selling pressure is starting to diminish. Doji indicate that the forces of supply and demand are becoming more evenly matched and a change in trend may be near. Doji alone are not enough to mark a reversal and further confirmation may be warranted.

Long-legged Doji

Long-legged doji have long upper and lower shadows that are almost equal in length. These doji reflect a great amount of indecision in the market. Long-legged doji indicate that prices traded well above and below the session's opening level, but closed virtually even with the open. After a whole lot of yelling and screaming, the end result showed little change from the initial open.

Dragon Fly Doji

Dragon fly doji form when the open, high and close are equal and the low creates a long lower shadow. The resulting candlestick looks like a "T" with a long lower shadow and no upper shadow. Dragon fly doji indicate that sellers dominated trading and drove prices lower during the session. By the end of the session, buyers resurfaced and pushed prices back to the opening level and the session high.

Gravestone Doji

Gravestone doji form when the open, low and close are equal and the high creates a long upper shadow. The resulting candlestick looks like an upside down "T" with a long upper shadow and no lower shadow. Gravestone doji indicate that buyers dominated trading and drove prices higher during the session. However, by the end of the session, sellers resurfaced and pushed prices back to the opening level and the session low.

As with the dragon fly doji and other candlesticks, the reversal implications of gravestone doji depend on previous price action and future confirmation. Even though the long upper shadow indicates a failed rally, the intraday high provides evidence of some buying pressure. After a long downtrend, long black candlestick, or at support, focus turns to the evidence of buying pressure and a potential bullish reversal. After a long uptrend, long white candlestick or at resistance, focus turns to the failed rally and a potential bearish reversal. Bearish or bullish confirmation is required for both situations.

Appendix C

This historical exchange data that we are using is freely available at: <http://fxhistoricaldata.com/> and <http://finance.yahoo.com/>.

We are using both the daily and the hourly databases of values to see which one performs best. We are going to be looking for two types of patterns: 0 0 0 0 0 1 1 for long and 1 1 1 1 1 0 0 for short.