

Upward and Downward Conditional Probabilities in 28 Currency Exchange Rates

A thesis submitted for the degree of Master of Science in
Algorithmic Trading

Pouyan Dinarvand

Supervisor: Professor Edward Tsang
Centre for Computational Finance and Economic Agents (CCFEA)
University of Essex

August 2016

Acknowledgment

Firstly, I would like to express my sincere gratitude to my supervisor Professor Edward Tsang for his support of my MSc thesis at the University of Essex and providing currency exchange rates data for this research. In addition, I would like to dedicate this thesis to my parents for their support and endless love.

Abstract

In this thesis, upward and downward conditional probabilities will be calculated for 28 currency pairs over 7 time periods ranging from 1-minute to 30-minute. The data used in this research are historical currency exchange rates from 27/9/2009 to 12/8/2015 that have minimum 1-minute time intervals. First, these raw data are converted to sequences of 0, 1 and 2 digits which represent downward, upward and no movements of exchange rates during a time period (e.g. 1-minute), respectively. Then, Bayes' rule is applied on converted data to calculate upward and downward conditional probabilities when a specific binary pattern (sequence of 0 and 1 digits) is observed in data. A C++ program is developed to automate all these processes.

We found that AUDNZD (Australian Dollar / New Zealand Dollar) has the highest conditional probability of 68.86% among other currency pairs. In addition, exchange rates are more predictable in lower time intervals and show resistance (reverting) behaviour during upward and downward movements. This resistance power decreases as the time interval increases from 1-minute to 30-minute.

Furthermore, we divide the 28 currency pairs into two groups called High Probability Currency Pairs (HPCPs) and Low Probability Currency Pairs (LPCPs) whether they have conditional probabilities of more or less than 60%. For each currency pair, conditional probabilities are shown on diagrams, which have applications in predicting the future direction of currency exchange rates and creating automated trading strategies for currency market.

Keywords: Conditional Probability, Bayes' Rule, Currency Market, Forex.

Table of Contents

1	INTRODUCTION	5
2	LITERATURE REVIEW	7
2.1	Currency Market	7
2.2	Bayes' Rule	8
3	METHODOLOGY	10
3.1	Data	10
3.2	Binary Patterns	14
3.3	Calculation of Conditional Probability	14
3.4	Experiment Design.....	17
4	RESULTS AND DISCUSSIONS.....	19
4.1	Low Probability Currency Pairs (LPCPs)	20
4.1.1	EURUSD (Euro / US Dollar)	20
4.1.2	USDJPY (US Dollar / Japan Yen).....	22
4.1.3	GBPEUR (Pound Sterling / Euro).....	23
4.2	High Probability Currency Pairs (HPCPs)	24
4.2.1	EURNZD (Euro / New Zealand Dollar).....	24
4.2.2	AUDNZD (Australian Dollar / New Zealand Dollar)	25
4.2.3	EURAUD (Euro / Australian Dollar)	27
4.3	Analysis of All Currency Pairs	27
5	CONCLUSIONS.....	30
6	FURTHER RESEARCH	31
7	REFERENCES	32
8	APPENDIX 1: CONDITIONAL PROBABILITY DIAGRAMS	34
9	APPENDIX 2: SOURCE CODE OF THE C++ PROGRAM.....	45

1 Introduction

What is the probability of an upward movement in currency exchange rates (e.g. Euro /U.S. Dollar) if an upward movement has already been observed in historical data? The primary goal of this research is to answer this question. In other words, we are interested to calculate the conditional probabilities of upward or downward movements in currency exchange rates based on an observed pattern in historical prices. The outcomes of this research have applications in predicting the future direction of currency exchange rates and creating automated trading algorithms.

According to the Efficient Market Hypothesis (EMH), market prices are efficient if all information is reflected in the prices [1], [2]. Therefore, it is impossible to predict the future prices by studying the historical prices. In other words, the price time series will be random walk and the probability that the price goes up or down is 50% (current price will be independent of the previous prices) [1], [2]. It is beyond the scope of this thesis to test whether EMH is true or false. In this research, we are interested to find conditional probabilities of greater than 60% in currency exchange rates.

Currency Market or Foreign Exchange market (Forex or FX) is a market where currencies of different countries (e.g. British Pound [GBP]), Euro [EUR], U.S. Dollar [USD]) can be exchanged with each other [3]. For example, a trader may buy GBP and sell USD at the same time. FX market runs 24 hours a day, operates 5 days a

week and is the most liquid market among other financial markets [4]. The details of FX market are discussed in section 2.1.

In this thesis, we use Bayes' rule [5] to calculate the conditional probabilities in FX rates. The concept of conditional probability is the fundamental block of this research, which is discussed in section 2.2. Bayes' rule is the foundation of this thesis, because it gives us a mathematical framework to calculate the probabilities of FX rates directions (upward and downward movements) given an observed pattern in historical FX rates.

Section 3 explains on the methodology of this thesis that is to calculate conditional probabilities in different currencies through the following sub-sections: data collection; conversion of raw FX to sequences of 0, 1, and 2 digits; binary patterns and experiment design. Next, the outcomes of the research are presented via upward and downward conditional probabilities diagrams in section 4. These diagrams show the likelihood that a FX rate (e.g. GBP) goes up or down in next time interval (e.g. 1-minute) based on an observed binary pattern in the past.

Although the main objective of this thesis is to calculate the conditional probabilities of upward and downward movements in FX rates, the finding of this thesis has applications in creating robust probability models that are capable of forecasting the future direction of FX rates. Plus, the results of this research are applicable to creating automated trading algorithms. Finally, the method developed in this thesis can be equally applied to detecting patterns in the prices of other financial asset classes such as stocks and commodities.

2 Literature Review

In the first section (2.1), details of FX market are discussed. In section 2.2, the concept of conditional probability and Bayesian statistics is presented.

2.1 Currency Market

Unlike stock exchange where there is a physical place that all trades (buying and selling stocks) occur, currency market or foreign exchange market (Forex or FX) does not have a central location and is called *Over The Counter* market (OTC) [6]. In FX market, currencies of different countries (e.g. Euro, U.S. Dollar and British Pound) are exchanged [3]. For example, it happens often that a trader buys Euro (EUR) while selling U.S. Dollar (USD) at the very same time, which means that the trader has bought EURUSD currency pair.

The market participants of FX mainly consist travellers, small or large companies, speculators, investors, investment banks, financial institutions, central banks and governments [7]. Each of these market participants have their own purpose for participating in currency market. For instance, travellers who want to travel from their home country to a foreign country need to convert their native currency to the destination currency. In addition, central banks or governments participate in FX market to control or modify the economic conditions [7].

In the FX market, the value of each currency is calculated in comparison with another currency and prices are quoted as a currency pair such as EURUSD (EUR/USD) [8]. In this example, EUR and USD are called the base and the counter currencies, respec-

tively [8]. In this thesis, we do not use “/” to separate base and counter currencies and show the symbol of currency pair without “/” (e.g. EURUSD). Another example of a currency pair is GBPUSD (British Pound/U.S. Dollar). When the value of this pair is 1.5, it means that £1 equals \$1.5.

In fact, exchange rates for each currency pair are quoted in bid and ask prices by market maker [9]. Market maker denotes a broker or a dealer that buys a financial instrument (e.g. currency pair) at the bid price and sells it at the ask price. The ask price is normally higher than the bid price and their difference is called *spread* [8].

In FX trading, it is possible to make profit regardless of the direction of the market [4]. In other words, when we expect a FX rate will go up (or down), we can buy (or sell) the FX rate to make profit in both directions. Another advantage of FX trading is that the market is open 24 hours a day, 5 days a week with daily trading volume of \$4 trillion [4]. However, there is a risk in FX trading due the price volatility in the FX market [10].

2.2 Bayes’ Rule

Bayes’ rule allows us to calculate conditional probability with a simple formula as shown in formula 2.1 [5].

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (2.1)$$

In this formula, $P(A \mid B)$ is the conditional probability of A given B. In other words, if B happens, $P(A \mid B)$ determines the likelihood that A will happen. The $P(A \cap B)$ is the joint probability of A and B happening together [5].

Bayes' rule is often applied on financial time series to test their inefficiency (predictability). For instance, tick-by-tick data of USDJPY were tested in [11], [12], [13] and [14]. These sources reported inefficiencies in high-frequency tick data. In addition, Rechenthin et al. [15] applied conditional probability in a popular U.S. stock for different intra-day time intervals and identified short-term predictability in high-frequency stock data.

Although there might be inefficiencies in high-frequency tick data as reported by the above sources, ordinary traders may not be able to use this information to create profitable trading strategies due to the transactions costs. Therefore, only high-frequency traders who may have access to high-tech infrastructures and pay less transaction costs in comparison with ordinary traders can make profit using this information.

In this thesis, we apply Bayes' rule on 28 currency pairs over intra-day timespans ranging from 1-minute to 30-minute to calculate upward and downward conditional probabilities. The details of our methodology are explained in the next section (section 3). The results of this research have applications in predicting direction of FX rates and creating automated trading systems. In addition, ordinary traders may find the outcomes of this thesis helpful in designing profitable trading strategies with holding periods of ranging from 1-minute to 30-minute.

3 Methodology

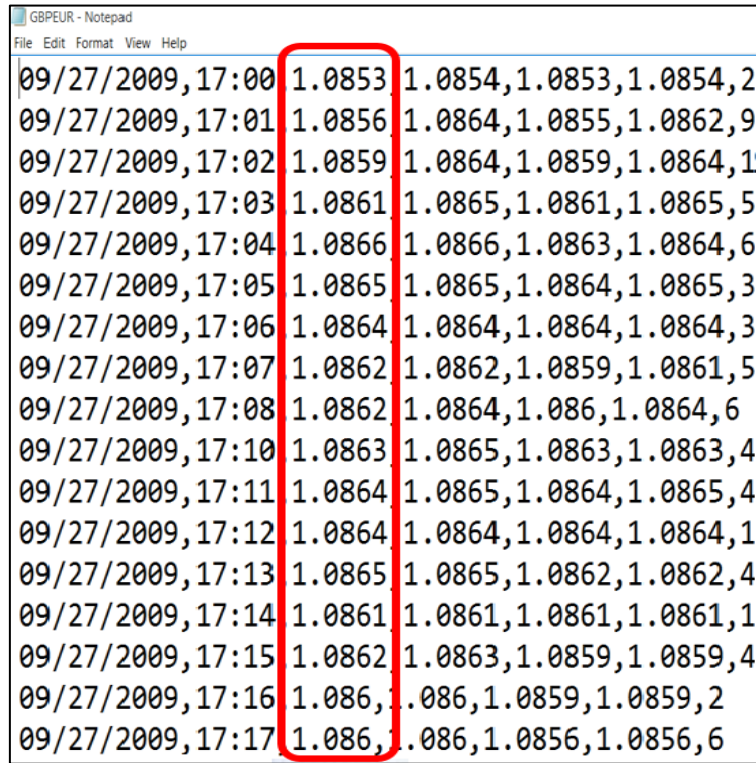
In this section, methodology of this research is discussed. First, the raw exchange rates will be summarised with sequences of 0, 1 and 2 digits which represent downward, upward and no movement, respectively. Details of data source and conversion procedure are explained in section 3.1. Section 3.2 describes how binary patterns (sequences of 0 and 1 digits) are created and utilised in this thesis. Then, we calculate the conditional probabilities of upward and downward movements in FX rates based on observed binary patterns in historical FX rates (section 3.3). Finally, details of experiment design to calculate conditional probabilities for different currency pairs over different time intervals are explained in section 3.4.

A C++ program is developed to automate the tasks of summarising raw FX rates with sequences of 0, 1 and 2 digits and then, calculating upward and downward conditional probabilities by Bayes' rule. The source code of this C++ program is provided in appendix 2.

3.1 Data

We need real prices in order to calculate the conditional probabilities of upward and downward movements in FX rates. The data (currency exchange rates) for this thesis are extracted from the exchange rates between 27/9/2009 and 12/8/2015. The data, which are provided by Professor Edward Tsang, have been purchased from www.kibot.com. These data are licenced for access by CSEE/CCFEA students. The data are in text format (.txt).

Figure 3.1 shows a sample picture of GBPEUR exchange rates. The time interval of the data is one minute. The first column is date. The second column is time stamp. The next four columns are open, high, low, and close (OHLC) FX rates and all of them are bid prices [16].



Date	Time Stamp	Open	High	Low	Close
09/27/2009	17:00	1.0853	1.0854	1.0853	1.0854, 2
09/27/2009	17:01	1.0856	1.0864	1.0855	1.0862, 9
09/27/2009	17:02	1.0859	1.0864	1.0859	1.0864, 1
09/27/2009	17:03	1.0861	1.0865	1.0861	1.0865, 5
09/27/2009	17:04	1.0866	1.0866	1.0863	1.0864, 6
09/27/2009	17:05	1.0865	1.0865	1.0864	1.0865, 3
09/27/2009	17:06	1.0864	1.0864	1.0864	1.0864, 3
09/27/2009	17:07	1.0862	1.0862	1.0859	1.0861, 5
09/27/2009	17:08	1.0862	1.0864	1.086	1.0864, 6
09/27/2009	17:10	1.0863	1.0865	1.0863	1.0863, 4
09/27/2009	17:11	1.0864	1.0865	1.0864	1.0865, 4
09/27/2009	17:12	1.0864	1.0864	1.0864	1.0864, 1
09/27/2009	17:13	1.0865	1.0865	1.0862	1.0862, 4
09/27/2009	17:14	1.0861	1.0861	1.0861	1.0861, 1
09/27/2009	17:15	1.0862	1.0863	1.0859	1.0859, 4
09/27/2009	17:16	1.086	1.086	1.0859	1.0859, 2
09/27/2009	17:17	1.086	1.086	1.0856	1.0856, 6

Figure 3.1. Sample picture of GBPEUR exchange rates.

In this thesis, only bid open FX rates as shown with a red rectangle on figure 3.1, are utilised to calculate conditional probabilities of upward and downward movements in FX rates. As we are interested in the direction of FX rates (up or down) and not in spot prices, we convert the raw FX rates to sequences of 0, 1 and 2 digits. We use the following notations for this conversion:

1. If price increases (goes up) during a certain time interval (e.g. 1-minute), it is represented by 1.

2. If price decreases (goes down) during a certain time interval (e.g. 1-minute), it is represented by 0.
3. If price stays at the same level during a certain time interval (e.g. 1-minute), it is represented by 2.

Figure 3.2 illustrates an example of this conversion for GBPEUR. 0, 1 and 2 digits represent downward, upward and no movements, respectively. We want to take into account the times when price stays at the same level and does not move. That is why 2 is used as a representation of no movement in FX rates during any given period.

```

111100021121010202102202201002210111021121
111010110200111110210022122102011121102120
010011021010211011010021100120101010210221
021010120101010102012102010102112011101010
001120212010111112001201011210220210102102
100010100010002101001110000121000010012110
021101202010111100200112211110000000111111
210021020110101112010210201221121201010110
220011120011010201100211021202100110101001
110201201121000021000010100012110110210110
000201011212201001010002111011010110201200
020101000120010102011021012102100012110101
121011022101001102100102010111022010010100
010011022101011011101000002120102000201110
120012101010101002200011011010012212010221

```

Figure 3.2. Summarising GBPEUR with sequences of 0, 1 and 2 digits.

Summarising raw FX rates as discussed in three above steps, has two advantages. First, FX data are summarised to sequences of 0, 1 and 2 digits which results in reducing the size of the original raw data. Therefore, further processing of FX rates be-

comes easier and less time consuming rather than processing the data manually. Second, upward and downward movements in FX rates are captured and we can use them to calculate upward and downward conditional probabilities in FX rates based on observed binary patterns (sequences of 0 and 1 digits) in historical data.

However, summarising the data manually is not practical. Because the sizes of the data files are too big, processing the data manually would consume a considerable time. In addition, we may introduce human errors if the data are processed by a human. In order to eliminate the human errors during conversion and reduce the duration of this process, a C++ program is coded (see `convertPriceToBinary` and `convertALLRealPricesToBinary` functions in Appendix 2).

The C++ program converts raw FX rates to sequences of 0, 1 and 2 digits automatically. After the converted data are generated, we can utilize them to calculate upward and downward conditional probabilities based on observed binary patterns in data. Binary patterns in this thesis are sequences of upward and downward movements in FX rates which are represented in sequences of 1 (up) and 0 (down) digits.

We use a binary pattern as a condition in Bayes' rule to calculate upward and downward conditional probabilities. For example, we want to calculate the chance that a FX rates goes up in the next time interval (e.g. 1-minute), if we observe three upward movements in data (pattern 111). The details of binary patterns and calculation of conditional probabilities are explained in sections 3.2 and 3.3, respectively.

3.2 Binary Patterns

Binary patterns are sequences of 0 and 1 digits. 0 and 1 represent downward and upward movements in FX rates, respectively. For example, pattern 100 means that FX rate goes up, down and down. The duration between each of these movements in this example is equal to the time interval used to convert raw FX rates (e.g. 1-minute).

Table 3.1 illustrates the binary patterns that are deployed in this research. The first column is the reference number of each pattern and the second is the binary pattern. The binary patterns are utilised as conditions in Bayes' rule to calculate upward and downward conditional probabilities. This concept is explained in the next section.

Table 3.1. Representation of binary patterns in this thesis.

Pattern Number	Binary Pattern	Definition
0	0	Down
1	1	Up
2	10	Up, Down
3	11	Up, Up
4	100	Up, Down, Down
5	101	Up, Down, Up
6	110	Up, Up, Down
7	111	Up, Up, Up

3.3 Calculation of Conditional Probability

The main goal of this thesis is to calculate the conditional probability that a FX rate (e.g. GBPUSD) will go up or down in the next time interval (e.g. 1-minute), when a specific binary pattern (from table 3.1) is observed in data. To achieve this goal, we apply Bayes' rule as discussed in section 2.2. Formulas 3.1 and 3.2 are derived from Bayes' rule (formula 2.1) for calculating upward and downward conditional probabilities, respectively.

$$P(\text{up} \mid \text{Binary Pattern}) = \frac{P(\text{up} \cap \text{Binary Pattern})}{P(\text{Binary Pattern})} \quad (3.1)$$

$$P(\text{down} \mid \text{Binary Pattern}) = \frac{P(\text{down} \cap \text{Binary Pattern})}{P(\text{Binary Pattern})} \quad (3.2)$$

$P(\text{up} \mid \text{Binary Pattern})$ and $P(\text{down} \mid \text{Binary Pattern})$ are the likelihoods that the price goes up and down, respectively, given an observed binary pattern in historical data. The binary pattern is a sequence of 0 (down) and 1 (up) digits as shown in table 3.1 which is utilised as condition in the above formulas. $P(\text{up} \cap \text{Binary Pattern})$ and $P(\text{down} \cap \text{Binary Pattern})$ are the joint probabilities that a specific binary pattern from table 3.1 is observed in data and then the price goes up and down, respectively. $P(\text{Binary Pattern})$ is the chance that a specific binary pattern is observed in data.

For example, to calculate the conditional probability of upward movements when pattern 100 (up, down, down) is observed in data, we use formula 3.3 which is derived from formula 3.1.

$$P(\text{up} \mid 100) = \frac{P(\text{up} \cap 100)}{P(100)} \quad (3.3)$$

In this example, the binary pattern (condition in Bayes' rule) is 100 and the direction that we want to calculate the conditional probability is upward. In other words, when we see up, down and down ward movements (pattern 100) in a FX rate, $P(\text{up} \mid 100)$ is the chance that the FX rate will go up in the next time interval (e.g. 1-minute). Picture of GBPEUR with pattern 100 is shown in figure 3.3. Pattern 100 is illustrated with red and blue rectangles. When this pattern happens, if the next digit is 1 (upward movement), the pattern is illustrated with a blue rectangle. Otherwise, it is shown with a red rectangle.

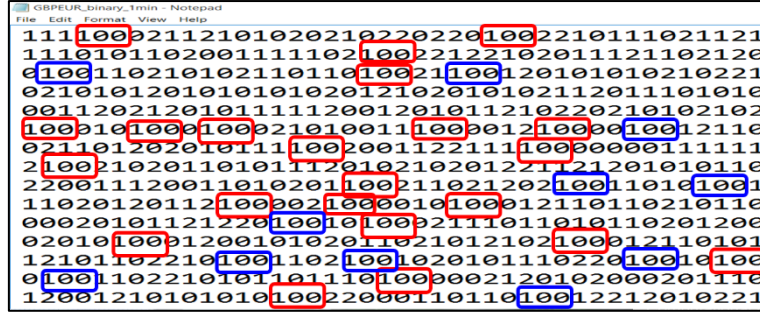


Figure 3.3. Pattern 100 in GBPEUR: blue rectangles illustrate $P(\text{up}=1 \cap 100)$, red rectangles show $P(\text{down}=0 \cap 100)$ or $P(\text{no movement}=2 \cap 100)$.

In this example (Figure 3.3), we want to calculate $P(\text{up} \mid 100)$. First, the total number of all rectangles (both blue and red) are counted. This number is equal to $P(100)$ which is put in the denominator of formula 3.3. Then we count the number of blue rectangles (situations where the FX rate goes up after pattern 100 happens). This number is equal to $P(\text{up} \cap 100)$ and is put in nominator of the formula 3.3 because pattern 100 happens and then the FX rate goes up. Next, number of blue rectangles ($P(\text{up} \cap 100)$) is divided by total number of rectangles ($P(100)$). The final result is equal to $P(\text{up} \mid 100)$.

The simple, straightforward and robust approach discussed above, enables us to explore different binary patterns in FX time series and find the upward and downward conditional probabilities for different FX rates. As mentioned previously in section 3.1, processing data manually is a time consuming task and may include human error. To overcome these obstacles, the same C++ program which is responsible for conversion of raw data (see section 3.1), also automates calculation of upward and downward conditional probabilities in FX rates. The source code of this program is provided in appendix 2 (see `searchPatternsInBinary` and `runStatisticalAnalysis` functions in appendix 2).

3.4 Experiment Design

The approach discussed in previous section (3.3) to calculate upward and downward conditional probabilities, is applied to 28 different currency pairs. The following 28 currency pairs are analysed in this research in order to calculate conditional probabilities:

- EURUSD (Euro / U.S. Dollar)
- EURNZD (Euro / New Zealand Dollar)
- AUDNZD (Australian Dollar / New Zealand Dollar)
- EURAUD (Euro / Australian Dollar)
- EURCAD (Euro / Canadian Dollar)
- EURJPY (Euro / Japan Yen)
- GBPEUR (Pound Sterling / Euro)
- GBPUSD (Pound Sterling / U.S. Dollar)
- GBPJPY (Pound Sterling / Japan Yen)
- GBPNZD (Pound Sterling / New Zealand Dollar)
- USDJPY (U.S. Dollar / Japan Yen)
- CADUSD (Canadian Dollar / U.S. Dollar)
- CADGBP (Canadian Dollar / Pound Sterling)
- CADJPY (Canadian Dollar / Japan Yen)
- CADNZD (Canadian Dollar / New Zealand Dollar)
- AUDUSD (Australian Dollar / U.S. Dollar)
- AUDCAD (Australian Dollar / Canadian Dollar)
- AUDGBP (Australian Dollar / Pound Sterling)
- AUDJPY (Australian Dollar / Japan Yen)
- NZDUSD (New Zealand Dollar / U.S. Dollar)
- NZDJPY (New Zealand Dollar / Japan Yen)
- CHFUSD (Swiss Franc / U.S. Dollar)
- CHFAUD (Swiss Franc / Australian Dollar)
- CHFCAD (Swiss Franc / Canadian Dollar)
- CHFEUR (Swiss Franc / Euro)
- CHFGBP (Swiss Franc / Pound Sterling)
- CHFJPY (Swiss Franc / Japan Yen)
- CHFNZD (Swiss Franc / New Zealand Dollar)

In addition, the exchange rates for each currency pair are converted to 7 time intervals including 1-minute, 2-minute, 3-minute, 5-minute, 10-minute, 15-minute- and 30-minute. Therefore, we have 196 (7×28) combinations. Then, these 196 summarised data are analysed with 8 binary patterns as shown in table 3.1. So we have 1568 ($7 \times 28 \times 8$) tests to calculate upward and downward conditional probabilities.

As mentioned in previous sections, in order to eliminate human errors and reduce time of data processing, we automate the process of calculating conditional probabilities. Therefore, the same C++ program, which converts data and calculates conditional probabilities for each currency pair (see sections 3.1 and 3.3), also calculates upward and downward conditional probabilities for all the 1568 combinations discussed above (see `runStatisticalAnalysis` function in appendix 2).

The outputs of the C++ program are upward and downward conditional probabilities for each of 28 currency pairs over 7 timespans based on 8 binary patterns as discussed in this section. These conditional probabilities are visualised on probability diagrams shown in section 4.

4 Results and Discussions

This section presents the final results of the thesis. As mentioned in previous sections, the main target of this research is to calculate the probabilities whether a FX rate will go up or down in the next time interval (e.g. 1 minute or 10-minute) when a specific binary pattern is observed in historical data. As we are interested to find probabilities of more than 60% in this research, the 28 currency pairs are divided into two groups based on 60% probability. The first group is named Low Probability Currency Pairs or LPCPs (section 4.1). This group contains currency pairs which have upward or downward conditional probabilities of less than 60% ($< 60\%$). The second group (section 4.2) which is named High Probability Currency Pairs or HPCPs, includes all FX rates that have upward or downward conditional probabilities of equal or more than 60% ($\geq 60\%$).

The results (conditional probabilities) for each currency pair are visualised on diagrams (e.g. figure 4.1). These diagrams show upward and downward conditional probabilities for each currency pair and have applications in predicting the future direction of a specific FX rate. The X-axis of each diagram displays binary patterns as shown in table 3.1 and Y-axis is the conditional probability in percentage. In addition, each diagram shows conditional probabilities over different periods ranging from 1-minute to 30-minute.

Finally, the results for all of the currency pairs analysed in this thesis, are compared with each other in order to find currency pairs which may be more predictable and may have more trading opportunities (see section 4.3). We can use the conditional probability diagrams to create trading signal generators in automated trading systems.

For example, if we know in advance that when a specific binary pattern happens, a FX rate has a higher chance to go up rather going down in the next minute, we can use this prior knowledge to buy the currency pair and expect the FX rate to rise.

4.1 Low Probability Currency Pairs (LPCPs)

Low Probability Currency Pairs (LPCPs) are currency pairs which have upward or downward probabilities of less than 60%. The probability diagrams are shown and discussed for LPCPs such as EURUSD, USDJPY and GBPEUR in this section. The diagrams for the rest of LPCPs are illustrated in the appendix 1.

4.1.1 EURUSD (Euro / US Dollar)

Figure 4.1 shows the conditional probability of upward and downward movements for EURUSD given a binary pattern is observed in data. The diagrams are shown for different time intervals ranging from 1-minute to 30-minute. For example, if we observe a downward movement of EURUSD in the last minute (we observe pattern 0), the chances that EURUSD goes up and down in the next minute are 49.55% and 37.84%, respectively and the rest 12.61% ($100 - 49.55 - 37.84$) is the probability that the FX rate stays at the same level. We can use these diagrams to find the probability whether EURUSD will go up or down in the next time interval (ranging from 1-minute to 30-minute).

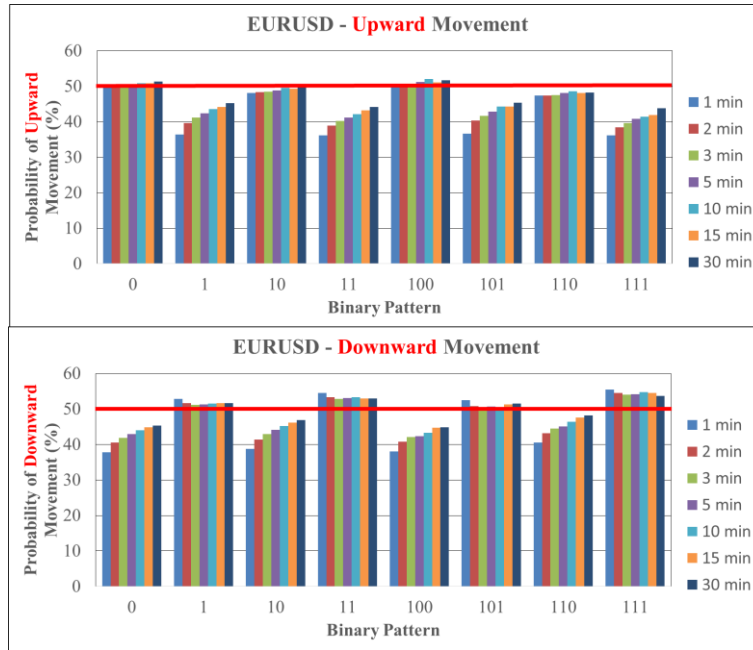


Figure 4.1. Probability of upward and downward movements for EURUSD.

The red line shown on diagrams is 50% line or random walk line. If the probability is lower than this level, the chance of EURUSD moving to that direction is low. Even if the probability is around 50%, it is random walk, which means that we cannot predict the next direction of EURUSD with high confidence. As shown in Figure 4.1, most of upward and downward probabilities for EURUSD are below or around 50% line. The probabilities of downward movements in 1-minute time intervals for EURUSD are 52.89%, 54.56% and 55.52% for patterns 1, 11 and 111, respectively. These are the highest probabilities calculated for EURUSD.

Furthermore, we can see from figure 4.1 that EURUSD shows a resistance behaviour toward last direction of the FX rate. In other words, when we see an upward movement in EURUSD (patterns that contain 1 and the end, e.g. 1, 11, 101 and 111), the probability that EURUSD goes down in the next time interval is higher than going up

and vice versa. In addition, when the conditional probabilities are below 50%, they increase as the time interval increases from 1-minute to 30-minute.

By taking all of the above points into account, we can conclude that EURUSD may be an efficient pair at least under our experiment design conditions because the probabilities are below or around 50%.

4.1.2 USDJPY (US Dollar / Japan Yen)

Figure 4.2 shows the conditional probability of upward and downward movements for USDJPY. All the probabilities (upward and downward) tend to move toward 50% random walk line. Therefore, USDJPY may be less predictable as the time interval increases.

The highest upward probability is 54.76% if pattern 100 happens in data (for 1-minute period). In contrast, USDJPY shows stronger resistant reactions when upward movements (patterns that have 1 at the end, e.g. 1, 11, 101 and 111) are observed in data. For instance, when pattern 111 happens in 1-minute period for USDJPY, the chance that USDJPY goes down in the next minute is 57.67%.

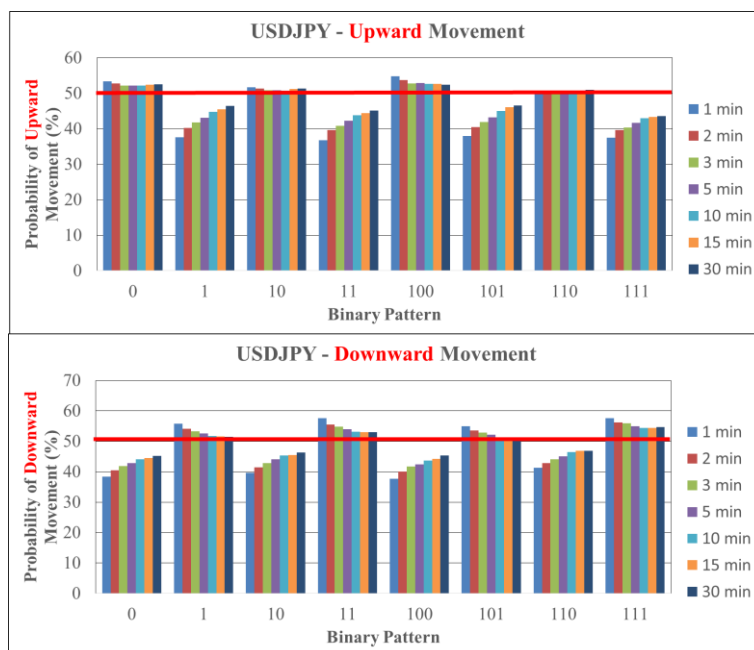


Figure 4.2. Probability of upward and downward movements for USDJPY.

4.1.3 GBPEUR (Pound Sterling / Euro)

Figure 4.3 shows the conditional probability of upward and downward movements for GBPEUR. Most of the probabilities are either around or below 50% random walk and they move closer to the random walk line when the time frame increases from 1-minute to 30-minute. The highest conditional probability of upward movement for this currency pair is 55.38% in 1-minute period when pattern 100 is observed in historical data.

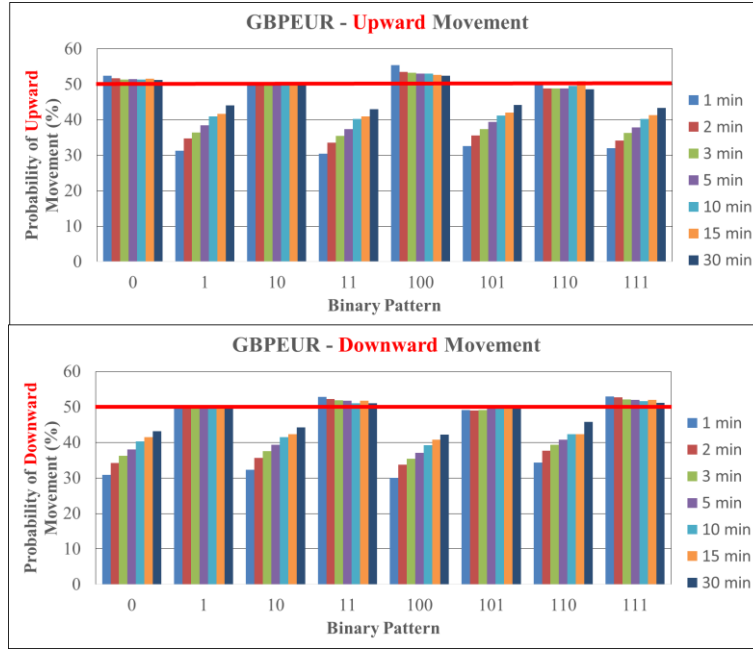


Figure 4.3. Probability of upward and downward movements for GBPEUR.

4.2 High Probability Currency Pairs (HPCPs)

High Probability Currency Pairs (HPCPs) are FX rates which have upward or downward probabilities equal or more than 60% such as EURNZD, AUDNZD and EURAUD. The conditional probability diagrams are shown for these FX rates in this section. The diagrams for other HPCPs are illustrated in appendix 1.

4.2.1 EURNZD (Euro / New Zealand Dollar)

Figure 4.4 displays the upward and downward probabilities for EURNZD. The highest upward probability is 59.84% when pattern 100 is observed in 1-minute period. When patterns 1, 11 and 111 are observed in 1-minute time interval, downward conditional probabilities are 59.73%, 63.49% and 64.32%, respectively. It means that the more the EURNZD moves upward, the chance that this currency pair moves down-

ward increases. In other words, EURNZD shows a resistant behaviour toward an upward trend. Furthermore, as the time interval increases from 1-minute to 30-minute, the probabilities move toward random walk line (50% line). Therefore, EURNZD may not be more predictable in higher periods (e.g. 30-minute) than lower time intervals (e.g. 1-minute).

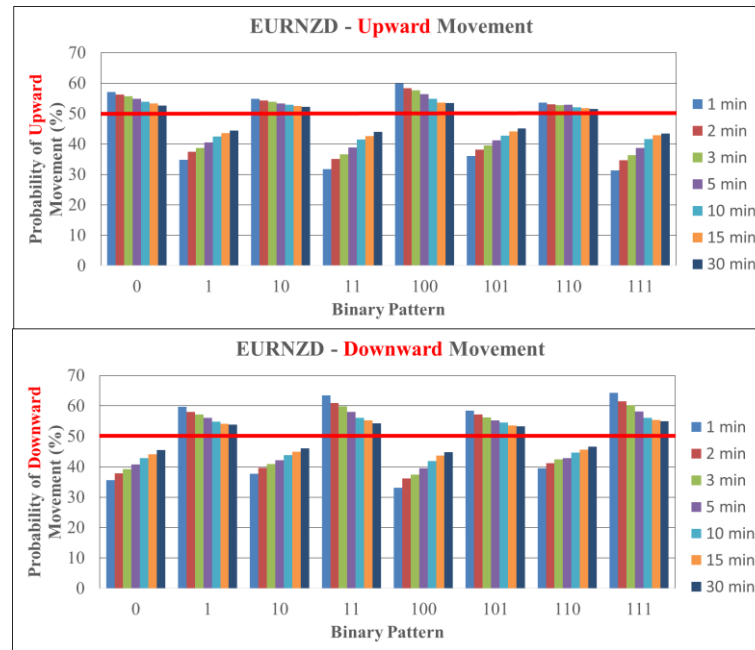


Figure 4.4. Probability of upward and downward movements for EURNZD.

4.2.2 AUDNZD (Australian Dollar / New Zealand Dollar)

Figure 4.5 illustrates the up and down probabilities for AUDNZD. The highest upward probabilities are 58.35%, 55.29% and 62.52% when patterns 0, 10, 100 are observed in 1-minute period. So AUDNZD shows a resistant reaction (AUDNZD has a higher chance of upward movement) when downward trend happens in 1 minute data. This resistant power decrease as the time interval increases from 1-minute to 30-minute. In addition, as the time interval increases from 1-minute to 30-minute, the probabilities move toward random walk line (50% line). Therefore, AUDNZD may

not be more predictable in higher periods (e.g. 30-minute) than lower time intervals (e.g. 1-minute).

When patterns 1, 11 and 111 are observed in 1-minute time interval, downward conditional probabilities are 61.09%, 66.94% and 68.86%, respectively. 68.86% is the highest probability observed among all the 28 currency pairs in this thesis. Therefore, the more the AUDNZD moves upward, the chance that this currency pair moves downward increases. In other words, AUDNZD shows a strong resistant behaviour toward an upward trend. Furthermore, the resistant power toward an upward trend is more powerful than toward a downward trend in AUDNZD under our experiment design conditions. It can be concluded that some patterns may exist in AUDNZD that can be used as trading opportunities.

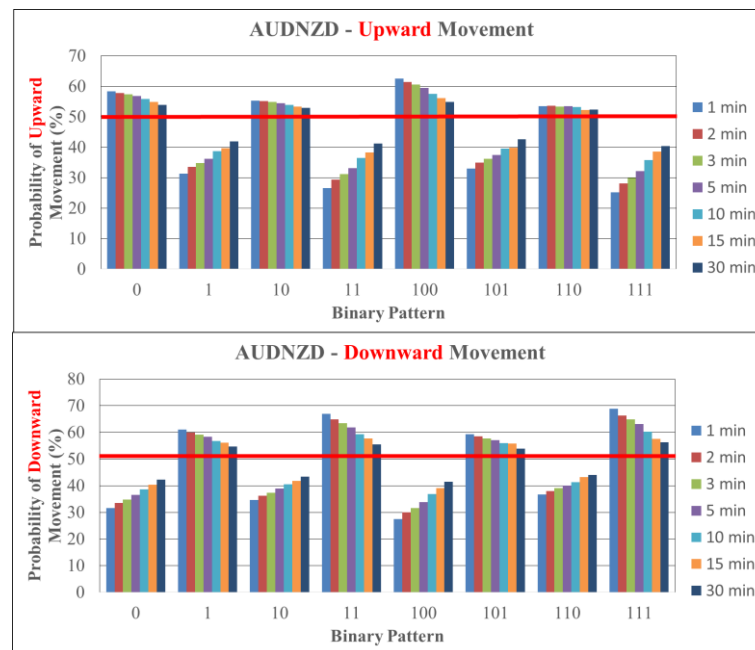


Figure 4.5. Probability of upward and downward movements for AUDNZD.

4.2.3 EURAUD (Euro / Australian Dollar)

Upward and downward probability diagrams for EURAUD are shown in figure 4.6. According to figure 4.6, EURAUD has a highest downward probability of 63.68% in 1-minute periods when pattern 111 is observed. In addition, conditional probabilities tend to move closer to 50% random walk line as the time interval increases.

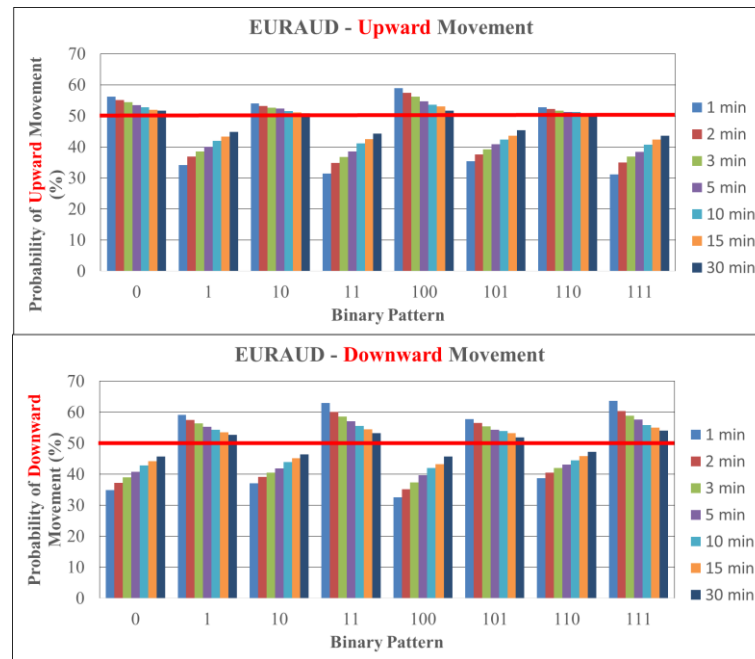


Figure 4.6. Probability of upward and downward movements for EURAUD.

4.3 Analysis of All Currency Pairs

Probability diagrams for some of the LPCPs and HPCPs are illustrated in previous sections (see sections 4.1 and 4.2). These diagrams for other currency pairs are shown in appendix 1. The maximum probabilities for 28 currency pairs are shown in table 4.1 and visualised in figure 4.7.

Table 4.1. Maximum conditional probabilities for 28 currency pairs.

Pair	% Max P(Direction Pattern)	Direction	Pattern	TIME INTERVAL
CHFNZD	56.99	Down	111	1-MINUTE
CHFJPY	59.74	Down	111	1-MINUTE
CHFGBP	60.40	Down	111	1-MINUTE
CHFEUR	61.98	Down	11	1-MINUTE
CHFCAD	51.74	Down	11	15-MINUTE
CHFAUD	60.47	Down	111	1-MINUTE
CHFUSD	55.56	Down	111	1-MINUTE
NZDJPY	62.37	Down	111	1-MINUTE
NZDUSD	56.65	Down	111	1-MINUTE
AUDJPY	61.49	Down	111	1-MINUTE
AUDNZD	68.86	Down	111	1-MINUTE
AUDGBP	56.38	Down	11	1-MINUTE
AUDCAD	62.32	Down	111	1-MINUTE
AUDUSD	55.03	Down	111	1-MINUTE
CADNZD	60.05	Up	100	1-MINUTE
CADJPY	61.25	Down	111	1-MINUTE
CADGBP	60.27	Down	111	1-MINUTE
CADUSD	53.99	Down	101	1-MINUTE
USDJPY	57.67	Down	111	1-MINUTE
GBPNZD	59.18	Down	111	1-MINUTE
GBPJPY	59.20	Down	111	1-MINUTE
GBPUSD	54.65	Down	111	1-MINUTE
GBPEUR	55.38	Up	100	1-MINUTE
EURNZD	64.32	Down	111	1-MINUTE
EURJPY	55.17	Down	111	1-MINUTE
EURCAD	62.59	Down	111	1-MINUTE
EURAUD	63.68	Down	111	1-MINUTE
EURUSD	55.52	Down	111	1-MINUTE

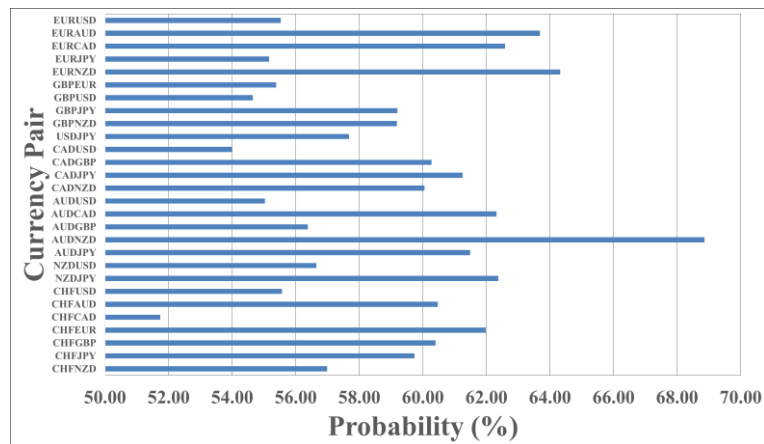


Figure 4.7. Visualisation of table 4.1: maximum conditional probabilities for 28 currency pairs.

HPCPs (currency pairs with maximum probability equal or more than 60%) are:

- CHFGBP
- CHFEUR
- CHFAUD

- NZDJPY
- AUDJPY
- AUDNZD
- AUDCAD
- CADNZD
- CADJPY
- CADGBP
- EURNZD
- EURCAD
- EURAUD

The rest of currency pairs are LPCPs (currency pairs with maximum probability less than 60%). The maximum probability calculated in this thesis is 68.86%, which is for AUDNZD. It is beyond the scope of this research to find the reason why AUDNZD has the highest conditional probability among all 28 currency pairs tested in this thesis. Therefore, further researches are needed to find the reason (see section 6).

In addition, the highest probabilities for most of the currency pairs (table 4.1) are downward probabilities when the binary pattern is 111 (in 1-minute time interval). In other words, when pattern 111 happens in most of the 28 currency pairs, FX rates have a high chance of moving downward in the next minute.

5 Conclusions

By applying Bayes' rule on FX rates, we calculated the upward and downward conditional probabilities for 28 currency pairs and divided them into two groups called LPCPs (low probability currency pairs) and HPCPs (high probability currency pairs). We can conclude the following statements by analysing the probability diagrams of each FX rate:

- The currency pair with the highest conditional probability is AUDNZD with downward probability 68.86% when pattern 111 is observed on 1-minute time interval.
- FX rates show resistance behaviour during upward and downward movements. In other words, when a FX rates goes up (or down) in 1-minute, it is more likely that it goes down (or up) in the next minute.
- As the time interval increases from 1-minute to 30-minute, the conditional probabilities tend to move closer to 50%. Therefore, FX rates may become more predictable in high-frequency periods such as 1-minute rather than 30-minute.
- Most of the currency pairs show their highest probabilities in 1-minute time interval. This conditional probability is for downward direction when pattern 111 is observed in 1-minute data.

The findings of this thesis have applications in following areas:

- Predicting the future direction of FX rates.
- Designing trading models based on probability diagrams for each FX rate.
- Implementing a fully automated trading system that generates trading signals based on probability diagrams for each FX rate.

6 Further Research

The approach used in this thesis can be applied to other asset classes such as stocks and commodities for calculating upward and downward conditional probabilities. Therefore, we can identify financial instruments which have more trading opportunities. Then we can use our findings to create automated trading systems that generate buy or sell signals based on the conditional probabilities that we have calculated in advance.

In addition, the time frames (e.g. 1-minute) used in this thesis can be modified to lower than 1-minute or higher than 30-minute. For example, we can use the method presented in this research to investigate the movement behaviour of a financial asset on high frequency periods (e.g. 1-second).

Furthermore, one interesting research topic is to find the reason why some financial assets have higher conditional probabilities than others. To achieve this goal, it may be helpful to investigate the market microstructure of a specific financial asset. In other words, by studying the pricing mechanisms, behaviours of buyers (demand side of the market) and sellers (supply side of the market) and their interactions, we may find the answer. This requires further research, which is beyond the scope of this thesis.

7 References

- [1] E. F. Fama, “The Behavior of Stock-Market Prices,” *Journal of Business*, vol. 38, no. 1, pp. 34-105, January 1965.
- [2] E. F. Fama, “Efficient Capital Markets: A Review of Theory and Empirical work,” *Journal of Finance*, vol. 25, no. 2, pp. 383-417, May 1970.
- [3] www.oanda.com, “Forex Trading Tutorial,” [Online]. Available: <https://www.oanda.com/forex-trading/learn/intro-to-currency-trading/>. [Accessed 10 August 2016].
- [4] www.oanda.com, “Benefits of Forex Trading,” [Online]. Available: <https://www.oanda.com/forex-trading/learn/intro-to-currency-trading/benefits/trading>. [Accessed 11 August 2016].
- [5] D. Barber, “Probabilistic Reasoning,” in *Bayesian Reasoning and Machine Learning*, Cambridge University Press, 2012, pp. 7-24.
- [6] www.oanda.com, “Evolution of an Open Forex Market,” [Online]. Available: <https://www.oanda.com/forex-trading/learn/intro-to-currency-trading/currency-market/evolution>. [Accessed 8 August 2016].
- [7] www.oanda.com, “Forex Market Participants,” [Online]. Available: <https://www.oanda.com/forex-trading/learn/intro-to-currency-trading/benefits/participants>. [Accessed 8 August 2016].
- [8] www.oanda.com, “Buying and Selling Currency Pairs,” [Online]. Available: <https://www.oanda.com/forex-trading/learn/intro-to-currency-trading/conventions/currency-pairs>. [Accessed 7 August 2016].
- [9] www.oanda.com, “Trading in The Forex Spot Market,” [Online]. Available: <https://www.oanda.com/forex-trading/learn/intro-to-currency-trading/conventions/spot-trading>. [Accessed 12 August 2016].
- [10] www.oanda.com, “Risks Involved in Forex Trading,” [Online]. Available: <https://www.oanda.com/forex-trading/learn/intro-to-currency-trading/benefits/risks>. [Accessed 11 August 2016].
- [11] T. Ohira, N. Sazuka, K. Marumo, T. Shimizu, M. Takayasu and H. Takayasu, “Predictability of Currency Market Exchange,” *Physica A: Statistical Mechanics and its Applications*, vol. 308, no. 1-4, p. 368–374, 2002.
- [12] N. Sazuka, “Analysis of Binarized High Frequency Financial Data,” *The European Physical Journal B. Condensed Matter and Complex Systems*, vol. 50, no. 1, p. 129–131, 2006.

- [13] Y. Hashimoto, T. Ito, T. Ohnishi, M. Takayasu, H. Takayasu and T. Watanabe, “Random Walk or A Run: Market Microstructure Analysis of the Foreign Exchange Rate Movements based on Conditional Probability,” National Bureau of Economic Research, Cambridge, Mass., USA, 2008.
- [14] M. Tanaka-Yamawaki, “Stability of Markovian Structure Observed in High Frequency Foreign Exchange Data,” *Annals of the Institute of Statistical Mathematics*, vol. 55, no. 2, p. 437–446, 2003.
- [15] M. Rechenthin and W. N. Street, “Using Conditional Probability to Identify Trends in Intra-day High-frequency Equity Pricing,” *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 24, p. 6169–6188, 2013.
- [16] www.kibot.com, “Frequently asked questions,” [Online]. Available: http://www.kibot.com/Support.aspx#historical_forex_data. [Accessed 10 August 2016].

8 Appendix 1: Conditional Probability Diagrams

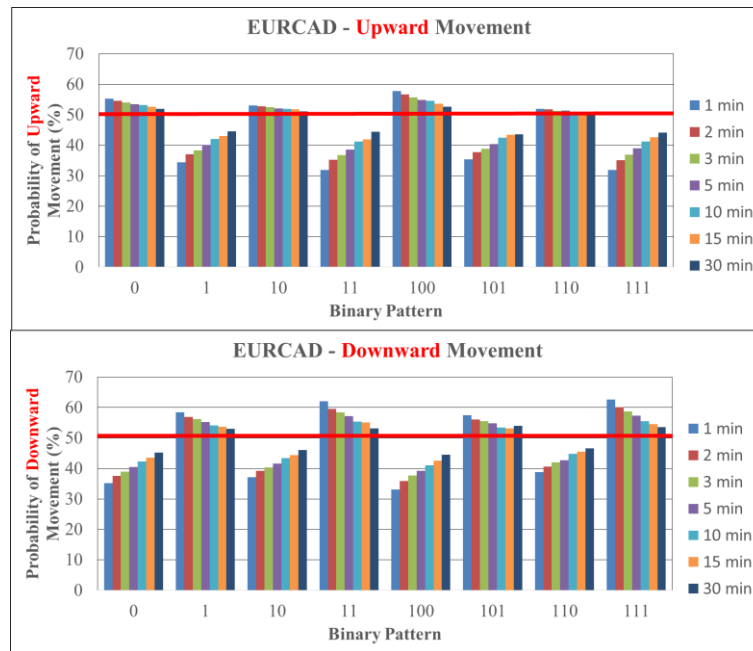


Figure 8.1. Probability of upward and downward movements for EURCAD.

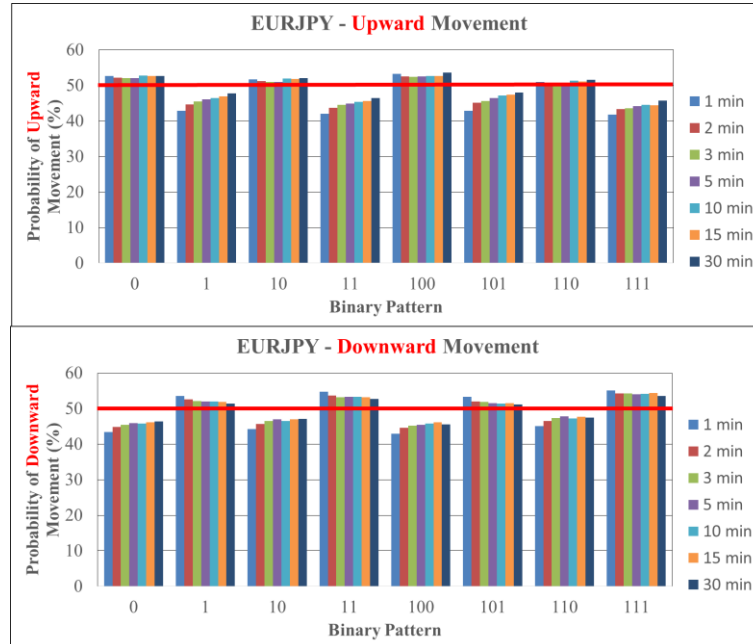


Figure 8.2. Probability of upward and downward movements for EURJPY.

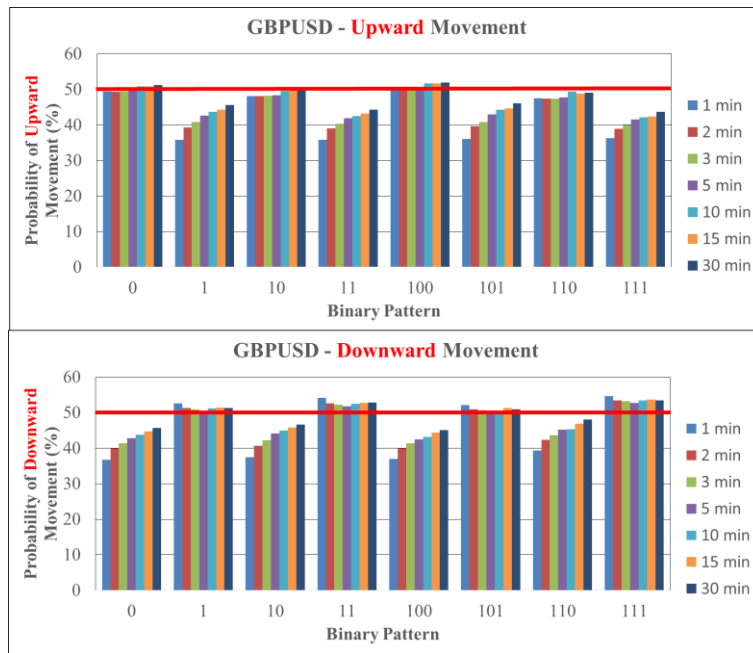


Figure 8.3. Probability of upward and downward movements for GBPUSD.

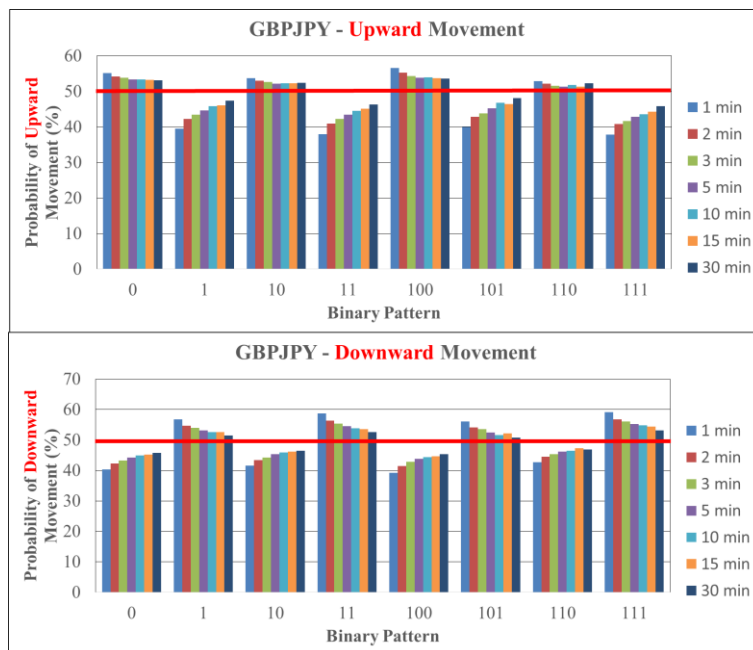


Figure 8.4. Probability of upward and downward movements for GBPJPY.

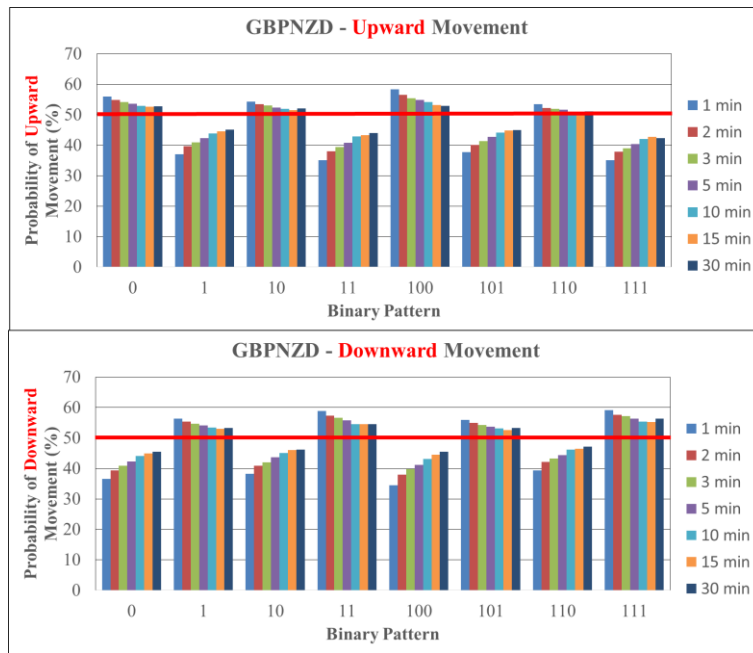


Figure 8.5. Probability of upward and downward movements for GBPNZD.

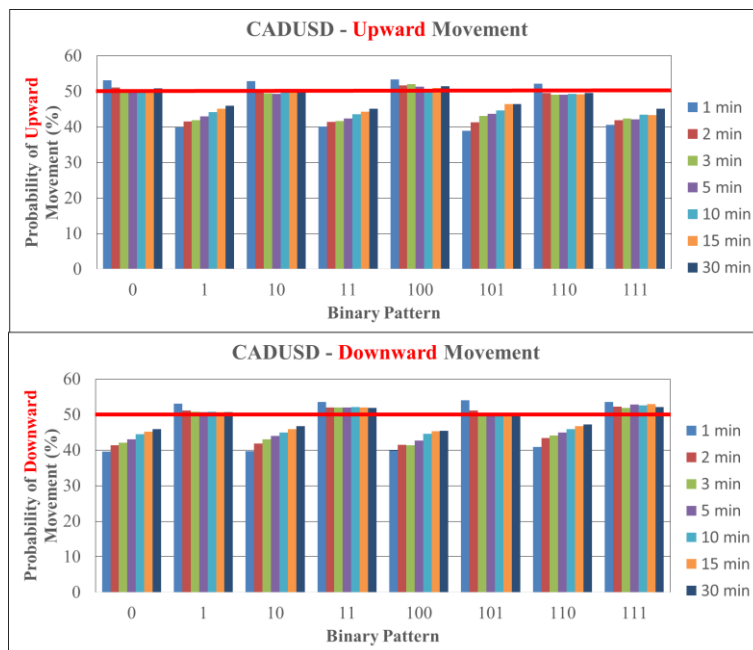


Figure 8.6. Probability of upward and downward movements for CADUSD.

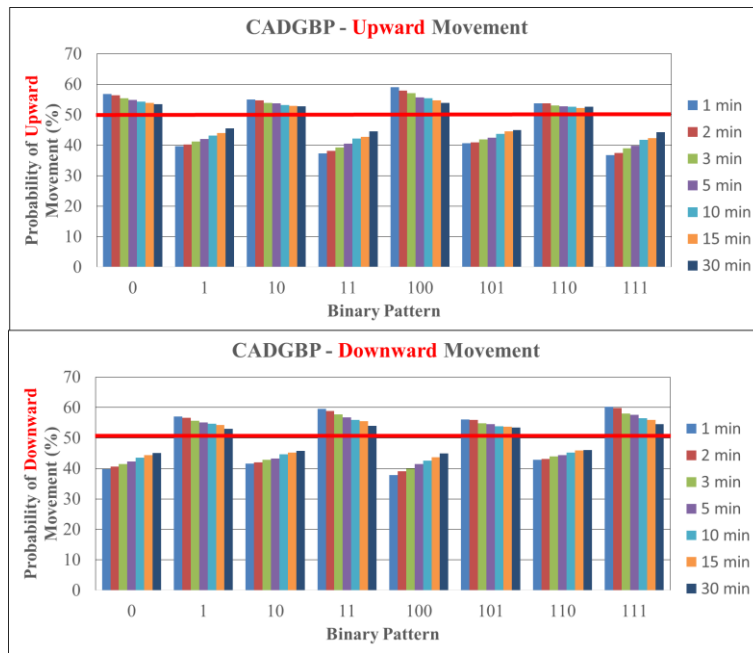


Figure 8.7. Probability of upward and downward movements for CADGBP.

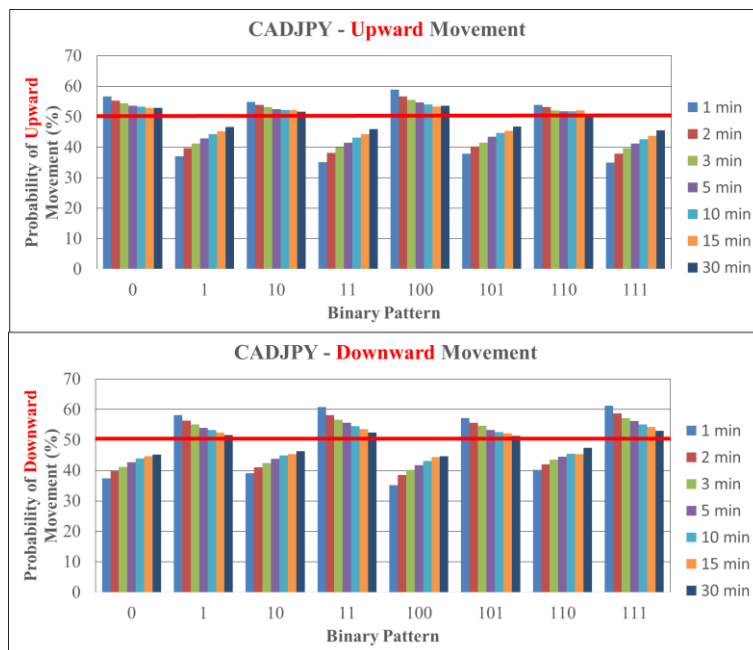


Figure 8.8. Probability of upward and downward movements for CADJPY.

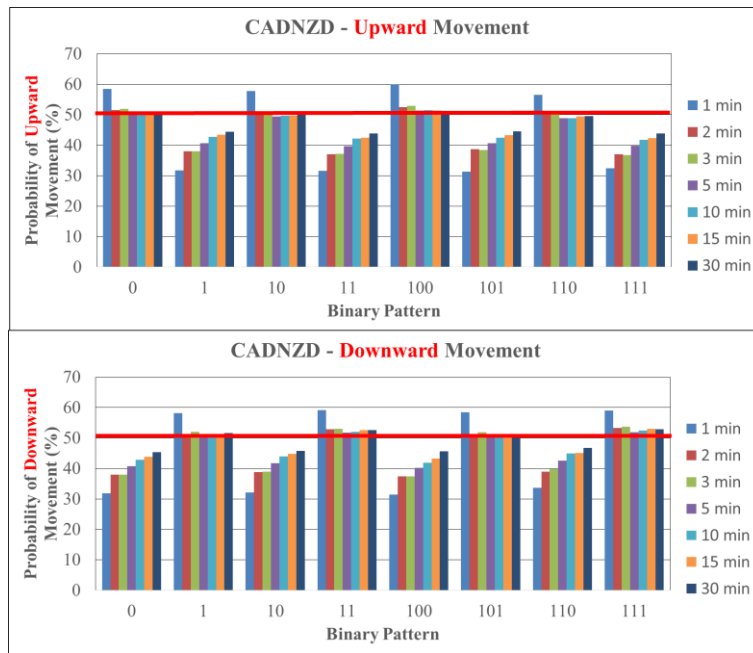


Figure 8.9. Probability of upward and downward movements for CADNZD.

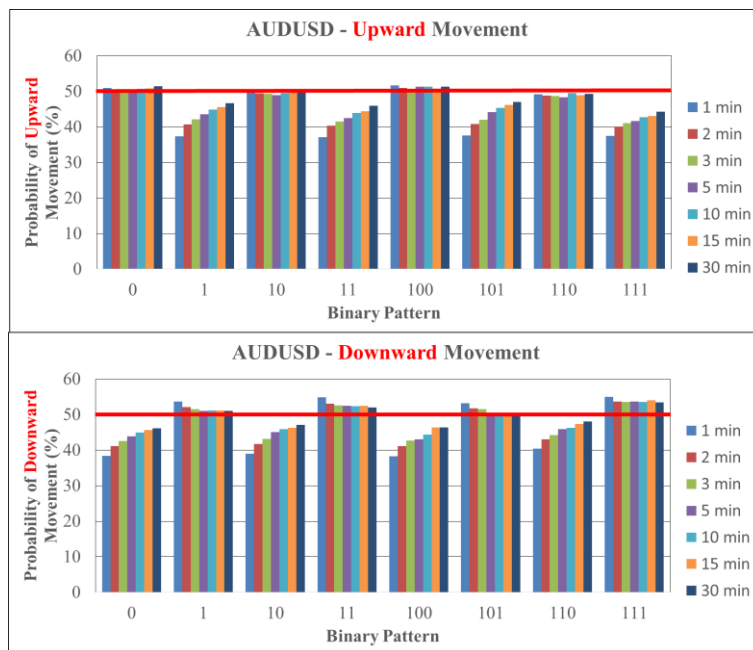


Figure 8.10. Probability of upward and downward movements for AUDUSD.

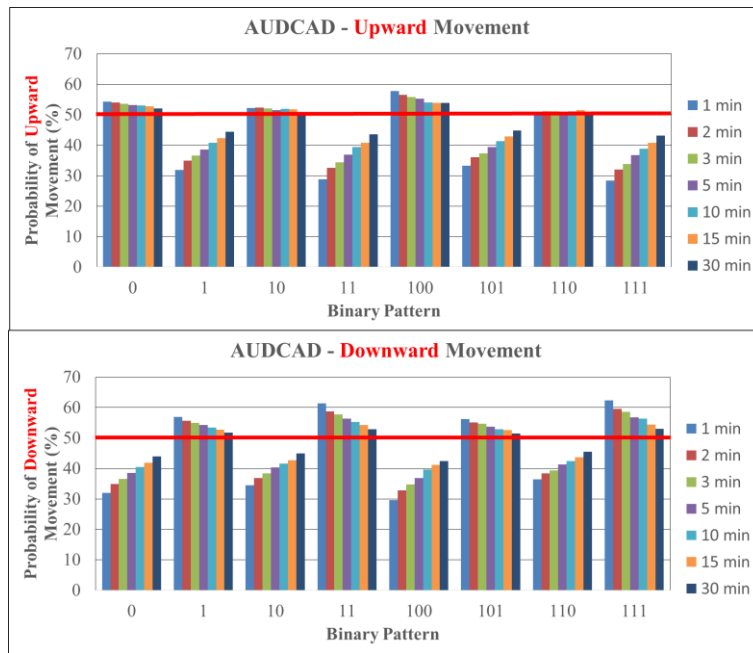


Figure 8.11. Probability of upward and downward movements for AUDCAD.

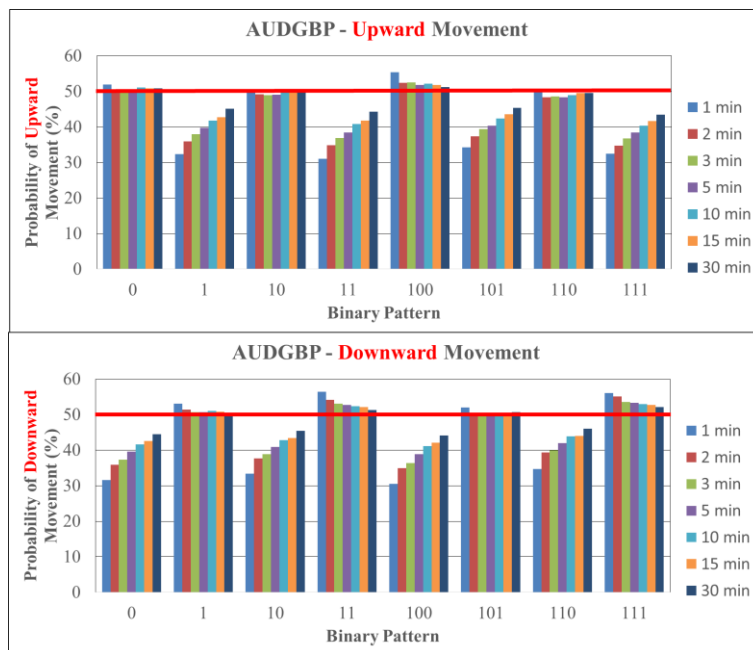


Figure 8.12. Probability of upward and downward movements for AUDGBP.

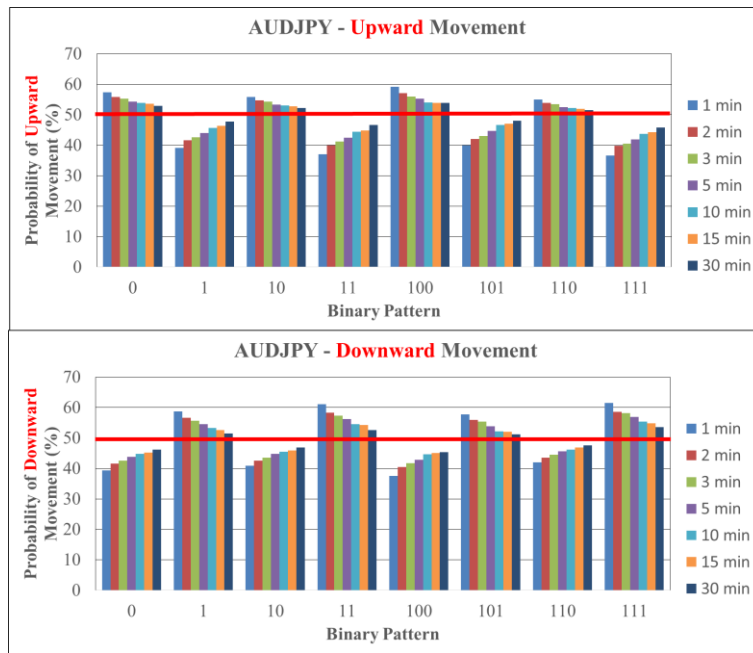


Figure 8.13. Probability of upward and downward movements for AUDJPY.

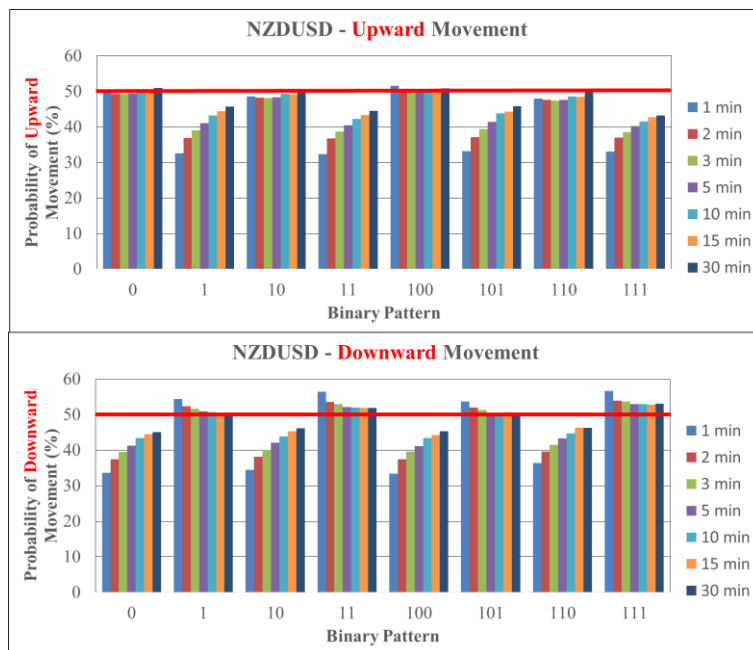


Figure 8.14. Probability of upward and downward movements for NZDUSD.

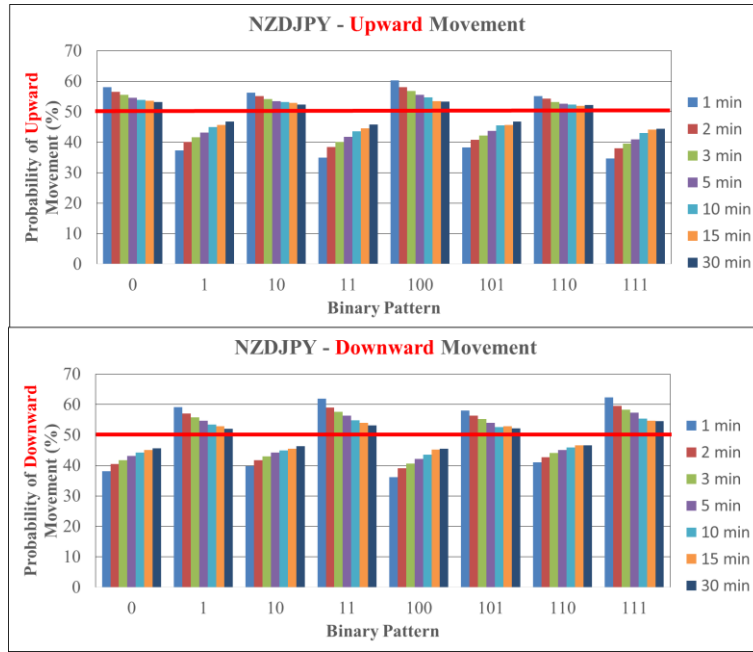


Figure 8.15. Probability of upward and downward movements for NZDJPY.

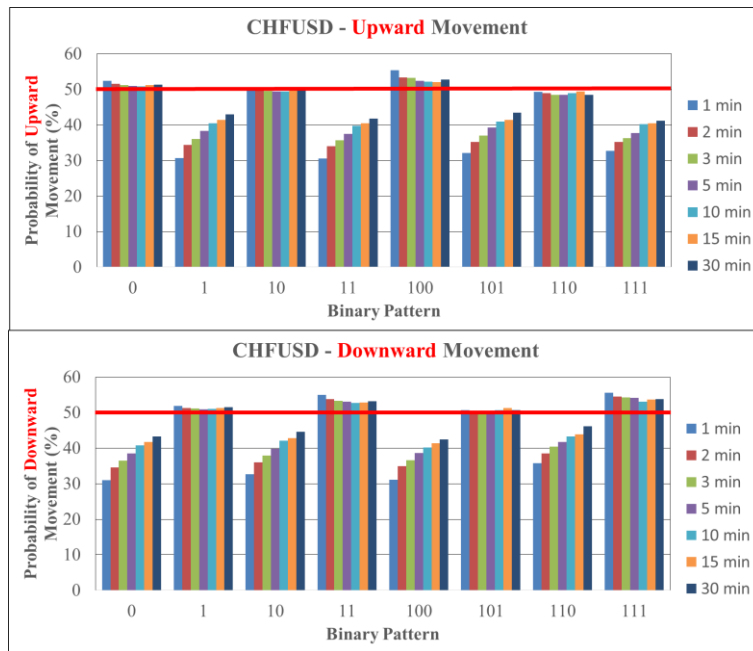


Figure 8.16. Probability of upward and downward movements for CHFUSD.

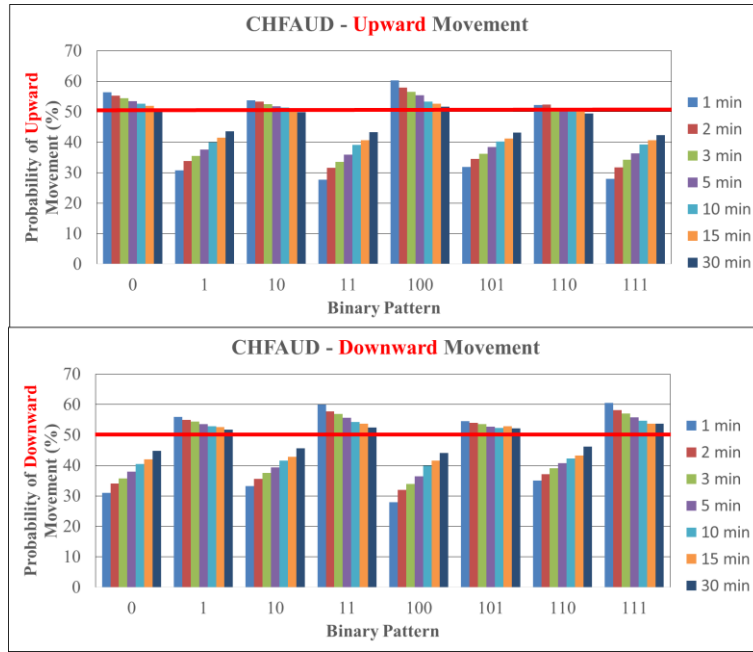


Figure 8.17. Probability of upward and downward movements for CHFAUD.

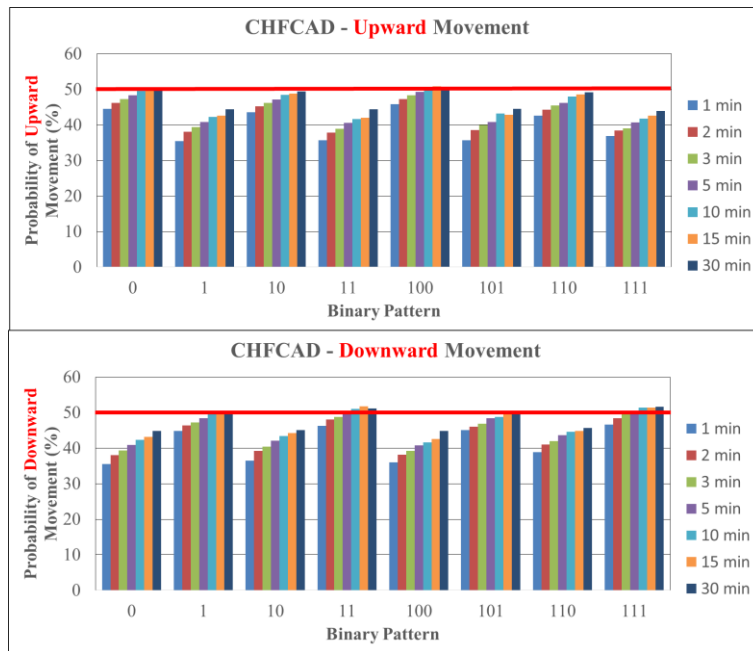


Figure 8.18. Probability of upward and downward movements for CHFCAD.

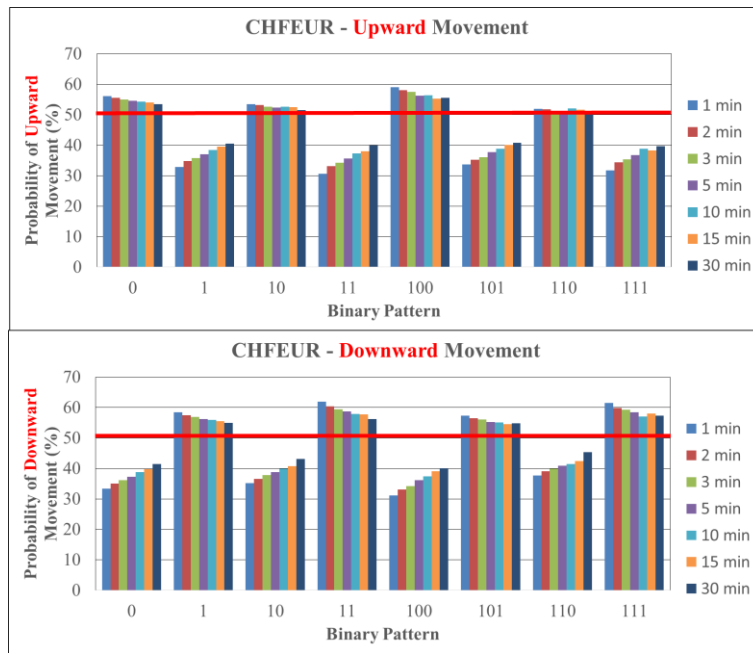


Figure 8.19. Probability of upward and downward movements for CHF EUR.

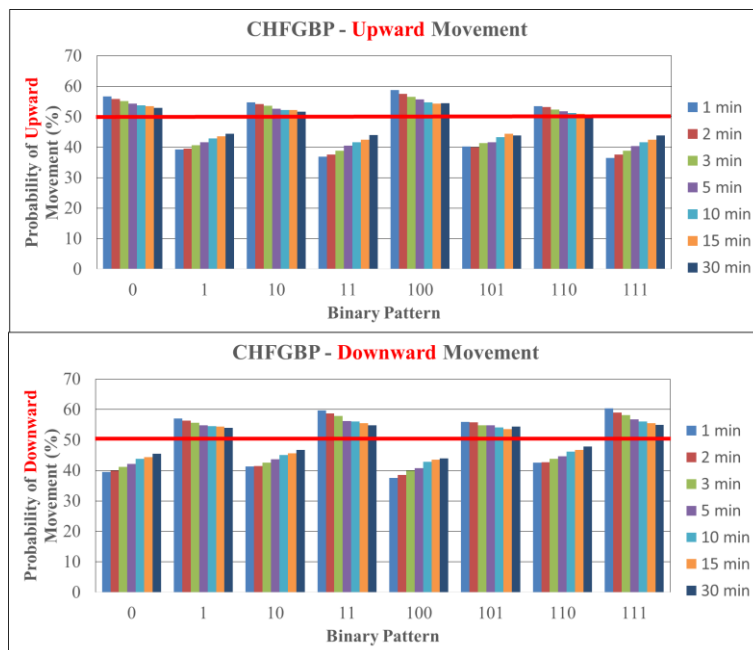


Figure 8.20. Probability of upward and downward movements for CHF GBP.

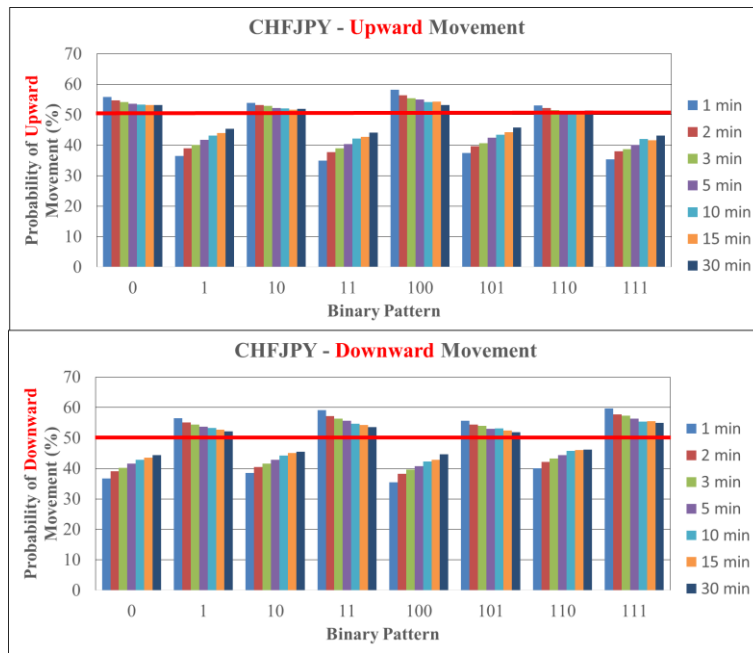


Figure 8.21. Probability of upward and downward movements for CHFJPY.

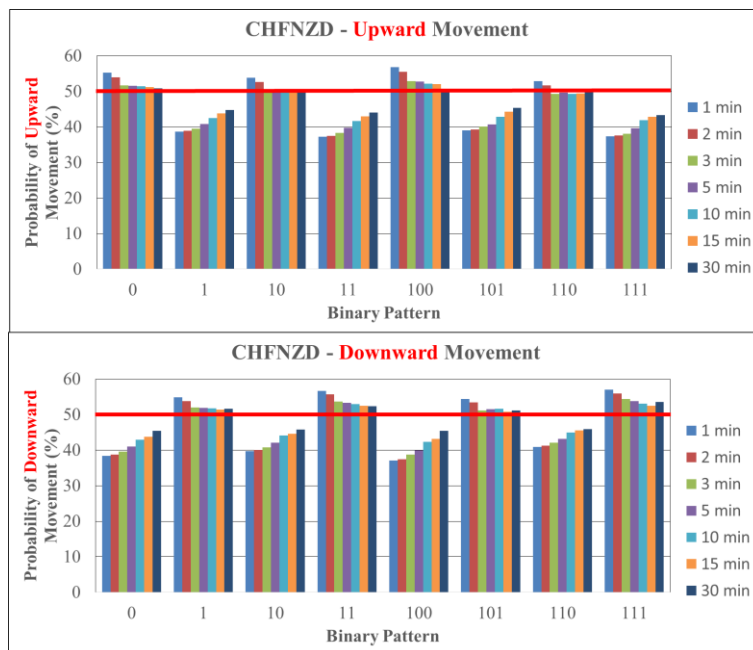


Figure 8.22. Probability of upward and downward movements for CHFNZD.

9 Appendix 2: Source Code of the C++ Program

/*

Author: Pouyan Dinarvand
All Rights Reserved
August 2016

This C++ program converts real FX rates (or random prices) to sequences of 0, 1, 2 digits:

0 -> downward price movement
1 -> upward price movement
2 -> no price movement

The name of the converted FX rates is in the following format:

-> currencyPair_binary_timeframe.txt (e.g. GBPUSD_binary_1min.txt)

Then, upward and downward conditional probabilities of seeing next 1(up) or 0(down) will be calculated based on previous observed binary patterns in historical data (e.g. $P(\text{up}|\text{pattern}=100)$) by applying Bayes rule.

Finally, the program shows the probabilities on screen in the following format:

-> pattern number, upward probability, downward probability

Note: All the data files must be in the same folder where main.cpp is.
Otherwise, the program may not function properly.

-> Compiler: GNU GCC Compiler
-> mingw32-g++.exe - std=c++11 - std=c++14
-> OS: Windows 10
-> IDE: CodeBlocks(16.01)

*/

```
#include <iostream>
#include <time.h>
#include <stdlib.h>
#include <vector>
#include <string>
#include <fstream>
```

```
using namespace std;
```

```
// Percentage number to filter the final results
double GOLDEN_PERCENTAGE = 10;
```

```
// ----- generate 01010 randomly -----
class BinaryRandom{
```

```
public:
```

```
    vector<long unsigned int> randomVector; // vector of 010221010
    long unsigned int numOfRandoms;         // length of randomVector
```

```
    // constructor & random generator
    BinaryRandom(long unsigned int numOfRandoms){
```

```

        this->numOfRandoms = numOfRandoms;

        srand(time(NULL)); // seed for random generator

        for(long unsigned int i=1; i<=this->numOfRandoms; i++){
            this->randomVector.push_back(rand()%3); // generate 0 or 1 or 2
        }

        cout<< " Binary Random numbers created." << endl;
    }

    // print randomVector
    void print(){
        cout << endl << " Binary Random numbers: " << endl;
        for(long unsigned int i=0; i< this->randomVector.size(); i++){
            cout <<this->randomVector[i];
        }

        cout<< endl << "End of Printing Binary Random Numbers" << endl;
    }
};

// ----- generate ONE pattern( without 2) (e.g. 0001001001) based on
// a number for just ONE CURRENCY PAIR -----
class Pattern{

public:
    int referenceNumber; // eg: 11 = 3 --> 3 reference number of pattern 11
    vector<unsigned int> patternVector; // vector 01010101 representing the pattern

    // Bayesian statistics of the pattern
    int numOfObservationInBinaryData; // how many time pattern was observed in binary data 010101

    // number of 1 observed in next forecast (one step forecast) when this pattern was seen
    int num_one_forecast;

    // number of 0 observed in next forecast (one step forecast) when this pattern was seen
    int num_zero_forecast;

    double percentageUpForecast; // 100* (num_one_forecast / numOfObservationInBinaryData)
    double percentageDownForecast; // 100* (num_zero_forecast / numOfObservationInBinaryData)

    // constructor
    Pattern(int referenceNumber){
        this->referenceNumber = referenceNumber;
        this->numOfObservationInBinaryData =0;
        this->num_one_forecast=0;
        this->num_zero_forecast=0;

        // convert integer n to binary code 010101
        int n = this->referenceNumber;
        while(n >=2){
            this->patternVector.insert(this->patternVector.begin(),n%2);
            n=n/2;
        }
        if (n == 1 || n==0){
            this->patternVector.insert(this->patternVector.begin(),n);
        }
    }

```

```

cout<< " Pattern " << this->referenceNumber << " created : ";

// print binary pattern
this->printPattern();

cout<<endl;

}

//Pattern(){ };// empty constructor

// does pattern observed from index start in binary random
bool isPatternFound(BinaryRandom& binaryRandomRef, long unsigned int startIndex=0){
    bool outcome = true;

    // check startIndex
    if(startIndex >=0 && startIndex <
        (long unsigned int) (binaryRandomRef.randomVector.size()- patternVector.size())){

        for(unsigned int i=0; i < this->patternVector.size();i++){
            if(this->patternVector[i] != binaryRandomRef.randomVector[i+startIndex] ){
                outcome = false;
                break;
            }
        }
    }
    else{
        outcome = false;
    }
    return outcome;
}

// does pattern observed from index start in real binary price
bool isPatternFound(vector<long unsigned int> & realBinaryPriceVectorRef,
    long unsigned int startIndex=0){

    bool outcome = true;

    // check startIndex
    if(startIndex >=0 && startIndex <
        (long unsigned int) (realBinaryPriceVectorRef.size()- patternVector.size())){

        for(unsigned int i=0; i < this->patternVector.size();i++){
            if(this->patternVector[i] != realBinaryPriceVectorRef[i+startIndex] ){
                outcome = false;
                break;
            }
        }
    }
    else{
        outcome = false;
    }
    return outcome;
}

// print Bayesian statistics found in binary random data regarding the pattern
void printPatternBayesianStatistics(){
    //cout<< "***** Statistics of Pattern: " << this->referenceNumber << " : ";
    //this->printPattern();
    // print format for excel file to draw graph

```

```

    cout << this->referenceNumber << ", "
        << this->percentageUpForecast
        << ", " << this->percentageDownForecast << endl;
    //cout<< " *****"
    /*
    cout<< endl << "number of observation : " << this->numOfObservationInBinaryData
        //<< endl << " - number of 1s observed in next forecast: " << this->num_one_forecast
        //<< endl << " - number of 0s observed in next forecast: " << this->num_zero_forecast
        //<< endl << " % probability of observing 1 in next forecast after seeing this pattern: "
        << endl << " % Upward: "
        << this->percentageUpForecast
        //<< endl << " % probability of observing 0 in next forecast after seeing this pattern: "
        << endl << " % Downward: "
        << this->percentageDownForecast << endl << endl;
    */

    //cout << " : "<<this->percentageUpForecast << endl;
}

// print pattern 01010101
void printPattern(){
    for(unsigned int j=0; j < this->patternVector.size(); j++){
        cout << this->patternVector[j];
    }
}

};

/* ----- convert 1-minute real price data to binary 010101 based on minuteInterval :
1min,5min,10min,15min,30min,60min and ... -----
open 1-min filename.txt (eg: EURUSD.txt), file must be in the same path and folder of main.cpp
price at each minute = open bid price at that minute
current price > previous price ---> 1
current price < previous price ---> 0
current price = previous price ---> 2
print the binary results 0101010 into a separate txt file called filename_binary_1min.txt
(eg: EURUSD_binary_1min.txt)
---> max minuteInterval = 60 because this function can only read minutes after :
*/
void convertPriceToBinary(string inputFileName, string outputFileName, int minuteInterval = 1){

    // max minuteInterval = 60
    if(minuteInterval>=60) minuteInterval = 60;
    ifstream inputStream(inputFileName);    // input (price 1 min OHLC)
    ofstream outputStream(outputFileName);    // output (Binary 101010101)
    char c;
    double previousPrice = 0, currentPrice;    // price
    string currentPriceString;    // need to be converted to double

    int previousTime = 0, currentTime;    // Time
    string currentTimeString;    // need to be converted to int

    if(! outputStream){
        cout << outputFileName << " (output file) cannot be opened !!!"<<endl;
    }

    if(! inputStream){
        cout << inputFileName << " (input file) cannot be opened !!!"<<endl;
    }
}

```



```

else{
    cout<< inputFile << " and " << outputFile << " opened successfully." << endl;
    bool doIHaveFirstPriceOfFile = false; // we do not have first price

    while(! inputStream.eof()){ // scan until the end of file

        currentPriceString="";
        currentTimeString= "";

        // locate minute : index in file
        do{
            if(! inputStream.eof()) inputStream>>c; else break;
        } while(c != ':');

        // get minute :00 to :59
        if(! inputStream.eof()) inputStream>>c; else break;
        currentTimeString = currentTimeString + c;
        if(! inputStream.eof()) inputStream>>c; else break;
        currentTimeString = currentTimeString + c;

        // 4th starts with open price (format data)
        if(! inputStream.eof()) inputStream>>c; else break;
        if(! inputStream.eof()) inputStream>>c; else break;

        currentPriceString = currentPriceString + c;

        // read open price
        do{
            if(! inputStream.eof()) inputStream>>c; else break;
            if(c == ',') break;
            currentPriceString = currentPriceString + c;
        } while(c != ',');

        currentPrice = stod(currentPriceString); // convert string price to real double
        currentTime = stoi(currentTimeString); // convert time to real int

        if (! doIHaveFirstPriceOfFile){
            previousPrice = currentPrice; // for the first time
            previousTime = currentTime;
            doIHaveFirstPriceOfFile = true; // just do it for ONE time
        }

        // check >= && <
        if( (currentTime >= previousTime && currentTime - previousTime >= minuteInterval)
            || (currentTime < previousTime && currentTime + 60 - previousTime >= minuteInterval)){

            if(currentPrice > previousPrice){ // upward >
                outputStream << '1'; // output is char
            }
            else if(currentPrice < previousPrice){ // downward <
                outputStream << '0'; // output is char
            }
            else{ // equal
                outputStream << '2'; // output is char
            }
        }
    }
}

```

```

        previousPrice = currentPrice;
        previousTime = currentTime;
    }

}

}

inputStream.close(); // close input stream
outputStream.close(); // close output stream

cout<< inputFileName << " and " << outputFileName << " closed successfully." << endl;

}

/* ----- open real binary prices and store it in reference vector for further analysis
inputFileName = filename_binary_1min.txt (eg: EURUSD_binary_1min.txt)
*/
void loadRealBinaryPrice(string inputFileName,vector<long unsigned int> & realBinaryPriceVector-
Ref){

    realBinaryPriceVectorRef.clear(); // clear memory

    ifstream inputStream(inputFileName); // open file that contains real binary 010101 prices

    char data;
    string s = ""; // empty string

    if(! inputStream){
        cout << inputFileName << " (input file) cannot be opened !!!" << endl;
    }
    else{
        cout<< inputFileName << " opened successfully." << endl;
        while(! inputStream.eof()){// scan until the end of file
            inputStream >> data;
            realBinaryPriceVectorRef.push_back((long unsigned int)stoi(s+data));
        }
    }

    inputStream.close(); // close input stream

    //cout<< inputFileName << " closed successfully and real binary prices loaded into memory." <<
endl;
}

// search all created patterns in random binary and print the final statistics of patterns
void searchPatternsInBinary(vector<Pattern> & patterns_vector_ref, BinaryRandom& binaryRan-
domRef){
    for(unsigned int j=0; j< patterns_vector_ref.size(); j++){

        // clear memory
        patterns_vector_ref[j].numOfObservationInBinaryData=0;
        patterns_vector_ref[j].num_one_forecast=0;
        patterns_vector_ref[j].num_zero_forecast=0;
        patterns_vector_ref[j].percentageUpForecast=0;
        patterns_vector_ref[j].percentageDownForecast=0;
    }
}

```

```

// search
for(long unsigned int i=0;
    i< (long unsigned int)(binaryRandomRef.randomVector.size()-
        patterns_vector_ref[j].patternVector.size()); i++){

    // pattern is found from index i
    if(patterns_vector_ref[j].isPatternFound(binaryRandomRef,i)){

        //update Bayesian statistics
        patterns_vector_ref[j].numOfObservationInBinaryData++;

        // update statistics regarding next forecast (one step forecast)
        //next forecast is 1
        if(binaryRandomRef.randomVector[i+patterns_vector_ref[j].patternVector.size()] == 1){
            patterns_vector_ref[j].num_one_forecast++;
        }
        else if(binaryRandomRef.randomVector[i+patterns_vector_ref[j].patternVector.size()] == 0){
            patterns_vector_ref[j].num_zero_forecast++; //next forecast is 0
        }

        // update percentage upward & downward one step forecast
        patterns_vector_ref[j].percentageUpForecast =
            ((double) patterns_vector_ref[j].num_one_forecast/
                patterns_vector_ref[j].numOfObservationInBinaryData) * 100;

        patterns_vector_ref[j].percentageDownForecast =
            ((double) patterns_vector_ref[j].num_zero_forecast/
                patterns_vector_ref[j].numOfObservationInBinaryData) * 100;

    }
}

// print Bayesian statistics of pattern j if there is
// good chance of upward or downward movement in next price movement
if(patterns_vector_ref[j].percentageUpForecast >= GOLDEN_PERCENTAGE||
    patterns_vector_ref[j].percentageDownForecast >= GOLDEN_PERCENTAGE){

    patterns_vector_ref[j].printPatternBayesianStatistics();
}
}
}

// search all created patterns in real binary and print the final statistics of patterns
void searchPatternsInBinary(vector <Pattern> & patterns_vector_ref, vector <long unsigned int> &
    realBinaryPriceVectorRef){

    for(unsigned int j=0; j< patterns_vector_ref.size(); j++){

        // clear memory
        patterns_vector_ref[j].numOfObservationInBinaryData=0;
        patterns_vector_ref[j].num_one_forecast=0;
        patterns_vector_ref[j].num_zero_forecast=0;
        patterns_vector_ref[j].percentageUpForecast=0;
        patterns_vector_ref[j].percentageDownForecast=0;

        // search
        for(long unsigned int i=0; i< (long unsigned int)(realBinaryPriceVectorRef.size()-

```

```

        patterns_vector_ref[j].patternVector.size());
        i++){
// pattern is found from index i
if(patterns_vector_ref[j].isPatternFound(realBinaryPriceVectorRef,i)){

    //update Bayesian statistics
    patterns_vector_ref[j].numOfObservationInBinaryData++;

    // update statistics regarding next forecast (one step forecast)
    //next forecast is 1
    if(realBinaryPriceVectorRef[i+patterns_vector_ref[j].patternVector.size()] == 1){
        patterns_vector_ref[j].num_one_forecast++;
    }
    else if(realBinaryPriceVectorRef[i+patterns_vector_ref[j].patternVector.size()] == 0){
        patterns_vector_ref[j].num_zero_forecast++; //next forecast is 0
    }

    // update percentage upward & downward one step forecast
    patterns_vector_ref[j].percentageUpForecast =
    ((double) patterns_vector_ref[j].num_one_forecast/
    patterns_vector_ref[j].numOfObservationInBinaryData) * 100;

    patterns_vector_ref[j].percentageDownForecast =
    ((double) patterns_vector_ref[j].num_zero_forecast/
    patterns_vector_ref[j].numOfObservationInBinaryData) * 100;

    }
}

// print Bayesian statistics of pattern j if there is
// good chance of upward or downward movement in next price movement
if(patterns_vector_ref[j].percentageUpForecast >= GOLDEN_PERCENTAGE||
    patterns_vector_ref[j].percentageDownForecast >= GOLDEN_PERCENTAGE){

    patterns_vector_ref[j].printPatternBayesianStatistics();
}
}
}

// convert all prices to binary based on t1, ... time intervals and create separate txt files
void convertALLRealPricesToBinary(vector <string> currencyPairsNameVector,
    vector <string> timeIntervalsVector){
    string inputFileName, outputFileName;

    for(unsigned int i=0; i < currencyPairsNameVector.size(); i++){
        for(unsigned int j=0; j< timeIntervalsVector.size(); j++){
            inputFileName = currencyPairsNameVector[i]+".txt";
            outputFileName =
            currencyPairsNameVector[i]+"_binary_" + timeIntervalsVector[j]+ ".min.txt";
            convertPriceToBinary(inputFileName,outputFileName,
                stoi(timeIntervalsVector[j])); // convert to binary
        }
    }
}

// perform statistical analysis on all currency pairs on all time intervals
void runStatisticalAnalysis(vector <string> currencyPairsNameVector,

```

```

        vector <string> timeIntervalsVector,
        vector <Pattern> & patterns_vector_ref,
        vector <long unsigned int> & realBinaryPriceVectorRef){
string binaryFileName; // txt file that contains 010101 of real prices

for(unsigned int i=0; i < currencyPairsNameVector.size(); i++){
    for(unsigned int j=0; j< timeIntervalsVector.size(); j++){
        binaryFileName =
            currencyPairsNameVector[i]+"_binary_" + timeIntervalsVector[j]+ ".min.txt";

        // load binary prices into memory
        loadRealBinaryPrice(binaryFileName, realBinaryPriceVectorRef);
        searchPatternsInBinary(patterns_vector_ref,realBinaryPriceVectorRef);

    }
}

}

int main()
{

    cout<< " Author: Pouyan Dinarvand " << endl
        << " All Rights Reserved " << endl
        << " August 2016 " << endl;

    cout<< " ----- Program Started ----- " << endl;
    //cout<< " Golden Percentage = " << GOLDEN_PERCENTAGE << endl;

    // ----- USER INPUT: PARAMETERS OF PROGRAM ----- //
    //long unsigned int numOfRandoms = 100;           // length of randomVector
    int numOfAllPatterns = 8;                        // total number of all binary patterns
    vector <Pattern> patterns_vector;                // vector of all patterns
    vector <Pattern> & patterns_vector_ref = patterns_vector; // Reference of vector of all patterns
    vector <string> currencyPairsNameVector;         // eg: EURUSD, GBPUSD

    // eg: 1, 2 , 5 , (minute) to convert real price to binary
    vector <string> timeIntervalsVector;

    // ----- test design : currency pairs for running experiment ----- //

    currencyPairsNameVector.push_back("EURUSD");
    currencyPairsNameVector.push_back("EURAUD");
    currencyPairsNameVector.push_back("EURCAD");
    currencyPairsNameVector.push_back("EURJPY");
    currencyPairsNameVector.push_back("EURNZD");
    currencyPairsNameVector.push_back("GBPEUR");
    currencyPairsNameVector.push_back("GBPUSD");
    currencyPairsNameVector.push_back("GBPJPY");
    currencyPairsNameVector.push_back("GBPNZD");
    currencyPairsNameVector.push_back("USDJPY");
    currencyPairsNameVector.push_back("CADUSD");
    currencyPairsNameVector.push_back("CADGBP");
    currencyPairsNameVector.push_back("CADJPY");
    currencyPairsNameVector.push_back("CADNZD");
    currencyPairsNameVector.push_back("AUDUSD");
    currencyPairsNameVector.push_back("AUDCAD");
    currencyPairsNameVector.push_back("AUDGBP");
    currencyPairsNameVector.push_back("AUDNZD");
    currencyPairsNameVector.push_back("AUDJPY");

```

```

currencyPairsNameVector.push_back("NZDUSD");
currencyPairsNameVector.push_back("NZDJPY");
currencyPairsNameVector.push_back("CHFUSD");
currencyPairsNameVector.push_back("CHFAUD");
currencyPairsNameVector.push_back("CHFCAD");
currencyPairsNameVector.push_back("CHF EUR");
currencyPairsNameVector.push_back("CHFGBP");
currencyPairsNameVector.push_back("CHFJPY");
currencyPairsNameVector.push_back("CHFNZD");

// ---- test design : time intervals (should be less than 59 min):
// 1 min, 2 min , ... 30 min for converting prices to binary 101010
timeIntervalsVector.push_back("1");
timeIntervalsVector.push_back("2");
timeIntervalsVector.push_back("3");
timeIntervalsVector.push_back("5");
timeIntervalsVector.push_back("10");
timeIntervalsVector.push_back("15");
timeIntervalsVector.push_back("30");

// ----- MAIN ALGO ----- //

// create and store all patterns in patternVector
for(int i=0;i<numOfAllPatterns;i++){
    patterns_vector.push_back(Pattern(i));
}

cout << " Output is in this format: pattern number, upward probability, downward probability "
    << endl;
// *** test with binary random data ***

// create BR object for testing program (not real data)
//BinaryRandom binaryRandom(numOfRandoms);

//BinaryRandom& binaryRandomRef = binaryRandom; // reference to binary random object
//binaryRandom.print();

// just use this function for one time to convert real prices to binary 01010121022
//convertALLRealPricesToBinary(currencyPairsNameVector, timeIntervalsVector);

// *** test with real data ***

// real binary format of prices
vector <long unsigned int> realBinaryPriceVector;

// reference of real binary price vector
vector <long unsigned int> & realBinaryPriceVectorRef = realBinaryPriceVector;

// search patterns in REAL random numbers and update and print Bayesian statistics
// for each pattern
runStatisticalAnalysis(currencyPairsNameVector, timeIntervalsVector,
    patterns_vector_ref, realBinaryPriceVectorRef);

// (optional)search patterns in binary RANDOM numbers and update and print Bayesian statistics
// for each pattern
//searchPatternsInBinary(patterns_vector_ref, binaryRandomRef);

cout << " ----- Program Finished ----- " << endl;
return 0;
}

```