Time Series Tool

Niclas Hellberg

May 26, 2010
Master's Thesis in Computing Science, 30 credits
Supervisor at CS-UmU: Oleg Seleznjev
Examiner: Per Lindström

Umeå University
Department of Computing Science
SE-901 87 UMEÅ
SWEDEN

Abstract

In the financial sector and financial trading it is generated and stored a massive amount of time-series every day. There exist a need of analyzing these time-series by traders and financial institutes. Especially in this case the need of comparing time-series for equality even if the series are not exactly equal. This problem occurs, for example, when data are migrated between systems and accuracy loss on data emerge.

The result is a graphical application that helps user to decide if the series are equal. The comparison is made upon a numerical solution that decides if the differences between two series are within acceptable limits.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Due to increasing storing capacity and computer performance, the ability to record and store a vast amount of information in the financial business every day is made possible. And the need of many record samples is essential. Data vendors like Reuters transmit more than 275,000 prices per day for foreign exchange spot rates alone[15]. On the stock markets all over the world daily close prices on stocks are recorded as well as indices, interest rates and buy/sell orders. This information plays an important role when economists is making risk analysis and forecasting of assets. Intra-day data, also referred to as *High-Frequency data*[15, 19] , is observations taken at a fine time scale. In this thesis the focus will be on intra-day information, buy and sell transactions and updates on equities where the time between events could be seconds or even milliseconds apart. When making analysis on this kind of data it is important that the recorded timestamps is correct and there is a need of a tool that check for this time-accuracy.

1.1 Nomura

Nomura Group has about 25000 employees and resides in Asia, Europe and America and amongst other things they are involved in Investment Banking. This thesis is based on a proposal offered by Nomura Sweden AB, a development center based in Umeå where they develop trading systems for the global financial market.

1.2 Report Outline

In the remaining part of this chapter the problem background are discussed. Chapter 2 explains the problem more exactly in detail and states what the goal with this thesis is. Chapter 3 presents a general study of time series and the analysis of them. The chapter gives a brief overview of the subject and explains some key features belonging to time series analysis.

The design and implementation are explained in Chapter 4; what kind of mathematical solution that was used on the time-series analysis, how the application was implemented and what tools that were used to complete the solution. Chapter 5 presents the resulting application, what solutions that were used and how to use some of the features. Chapter 6 lists a few limitations in the application and a discussion about how future work could

enhance the application are presented. A user guide for the application is added in the end of the report as an appendix.

1.3 Background

Buy and sell transactions is referred to as **trades** where a trade is generated when a stock exchange transaction is closed. In this trade: information of price, volume, time and so forth is registered. Updates on shares are also called **quotes** and contains information about *bid* and *ask* prices among other things. Bid-price is the price the buyer of the equity is willing to pay and the ask-price is what the seller is willing to sell it for. And when they have a match and agreement a transaction takes place and a trade is generated. These trades and quotes are the subject of this thesis and are called *tick-by-tick* data; every *tick* is one logical unit of information.

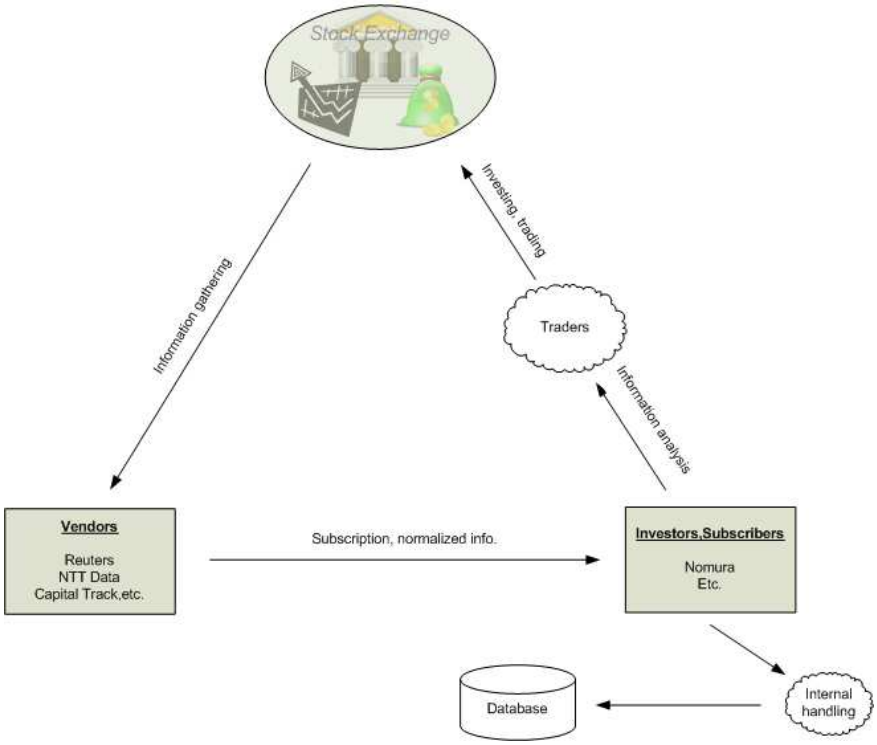

Figure 1.1: Illustration of the information flow on the Stock Market.

Besides the New York Stock Exchange (NYSE) there are many more stock exchange venues in the global finance. Due to the variety and different standards on the venues there is a need of a "middleman" (vendor) that aggregates the large amount of data and data standards. The data is then normalized and transformed to be more available and more useful for traders and investors globally. Examples of vendors are Thomson Reuters (USA)[1],

[1] `http://thomsonreuters.com/`

Bloomberg (USA)[2] , NTT Data (Japan)[3] and CapitalTrack (GB)[4]. Investors and traders such as Nomura can subscribe for daily, monthly or real-time data for a certain cost; as illustrated in Figure 1.1. This information can be used in a variety of ways but especially to extract time-series from this data and, as described in Chapter 3 on page 7, to make analysis of it and utilize this in future investments.

[2] `http://www.bloomberg.com/`
[3] `http://www.nttdata.co.jp/en/`
[4] `http://www.capitaltrack.net/new/`

Chapter 2

Problem Description

Nomura amongst other investors and traders use more than one vendor to subscribe for the same data, i.e. both Reuters and CapitalTrack offer information on the same equities registered on NYSE. Even though it is the same data it could be small changes in the event-time registration. Because of the very high pace of events, milliseconds apart, the compatibility between systems, different vendors, aren't really accurate, thus leading to differences in timing and timestamps. Also internally within Nomura´s systems, information is moved around to different areas where needed. During this migration and adaption to different local systems, truncation and other reformatting actions can occur with accuracy loss in time as a result. This in turn can affect the outcome of any analysis.

Date	Time	Symbol	Price	Volume	Source of Trade
09/26/2007	**10:29:53.851**	Eric_B	40.92	300	N
09/26/2007	**10:29:56.201**	Eric_B	41.01	600	N
09/26/2007	**10:29:56.201**	Eric_B	41.01	600	C
⋮	⋮	⋮	⋮	⋮	⋮

Table 2.1: Trade data example. Each tuple corresponds to one trade, a tick.

In Table 2.1 we can see an example of tick-by-tick data that by definition forms a time-series due to the successive time-stamps. When talking about time-series analysis and especially regression-analysis (see Chapter 3 on page 7), either univariate or multivariate series are used[17, 11]. As described in Chapter 3 on page 7 often when analyzing a time-series, data is matched against the time. For example if we plot price in Table 2.1 against time we have a univariate time-series with one independent variable. If we add volume simultaneously to price and plot against time we get a multivariate time-series; if adding volume makes any sense is another question. However in this thesis and this specific problem only the time-variable is of importance. All the variables are assumed to be exactly equal besides the durations, spacing of the corresponding time-stamps.

Nomura is in need of deciding if these time series are equal even though they differ in time. And when the data amount is so tremendous it would be a great help to have a tool that makes this automatically.

2.1 Goal

The aim of this thesis is to study time-series in general and particular in the area of the financial market; to define what properties that is associated with time-series and how they relate to the financial business; how to gain more understanding in the processes generating time-series and in what constraints that two time-series can be considered equal.

From this theoretical background the approach is to construct or adapt a method to compare time-series for equality. When the method is specified a tool has to be developed that uses this method to establish if two series are equal.

Chapter 3

Time series analysis

In this chapter the intention is to give a definition of what time series is and what properties that are associated with them; what techniques that are used and in what areas they are practiced, particularly the financial sector. Only some parts of the area are introduced in the rapidly growing field of time series modeling and analysis. Readers are expected to be familiar with the basics of mathematical statistics; however some areas are still mentioned in this chapter.

3.1 Time Series

A time series is a set of observations x_t, each one being recorded at specific time t[17]. A *discrete-time time series* is one in which the set T_0 of times at which observations are made is a discrete set, as is the case, for example, when observations are made at fixed time intervals. *Continuous-time time series* are obtained when observations are recorded continuously over some time interval, e.g., when $T_0 = [0, 10]$. In this thesis it is exclusively discrete-time time series that are being discussed.

There exist many various types of time series and in many various fields such as engineering, science, sociology, and economics. In economics and finance, time series are generated in a large quantity every day. For example, share prices that are recorded every close day[1] on stock market, average incomes in successive months or company profits in successive years and so on. In physical science, there exist amongst others time series in meteorology, marine science and geophysics. And from that area examples could be rainfall and air temperature observed at a specific time in successive days. Demographic time series are generated from, e.g., population measured annually or monthly birth totals in Sweden. These are only a few examples of the various types of time series that could be observed.

In Figure 3.1 we can see an example of a time series where it shows the total amount of energy consumption per month (GWh) in Sweden during the year 1974 through 2008[2]. Figure 3.2 is another example of the closing price every open day[3] of the Ericsson B equity

[1] *Close-day price* are the last observed price for that specific share on that day before the stock exchange is closed. The exact time can differ due to holidays.

[2] The data for the graph is collected from Nasdaq OMX, http://www.nasdaqomxnordic.com/.

[3] *Every open day* refers to the days that the stock exchange are open. Stock exchange are closed on weekends and other holidays.

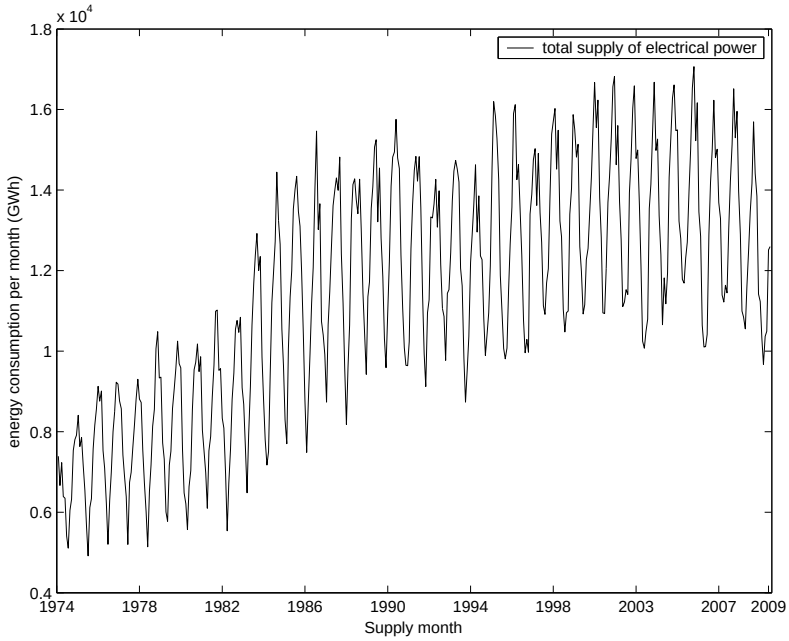

Figure 3.1: Energy consumption (GWh) per month in Sweden during 1974 through 2008.

at the OMX stock market exchange during the period 2000-2004[4].

From a given set of n observations it is convenient to rescale the time axis in such a way that the observations corresponds to the set of integers $\{1, 2, \ldots, n\}$. In the previous examples this means that (2001, January) corresponds to 1 and the time set for Figure 3.1 is $\{1, 2, \ldots, 431\}$. In the same way, (2006, day 1) corresponds to 1 and the time set for Figure 3.2 is $\{1, 2, \ldots, 999\}$.

3.2 Time Series Analysis

Besides from using time series as historical records of specified data it is also possible to gain more information about them by doing time series analysis[17]. With a model fitted to the data, it is possible to draw inferences from such series about its nature and mechanisms generating them. Depending in which field it is used, that information can be used in a variety of ways, for example, to forecast or estimate risk associated with trading on a specific equity on the stock market.

Other areas of practices could be to predict future sales using advertising expenditure data, or controlling future values of series by adjusting parameters, process control[20]. In a manufacturing process it is a problem to detect changes in performance of process. By measuring a variable that shows the quality of the process it is possible to plot against time and detect deviations. When deviations occur and the value stray too far from some historical average value, suitable corrections could be taken to control the process and the results.

It is assumed that an observed time series contains some form of a systematic pattern so

[4]The data for the graph is collected from Statistics Sweden, http://www.scb.se/.

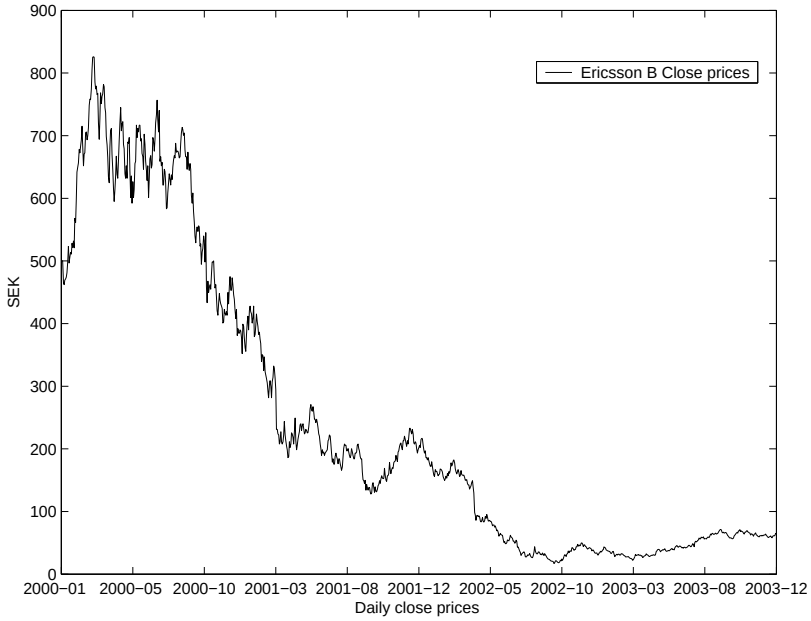

Figure 3.2: Closing price on Ericcson B share at OMX Stockholm (2000-2004)

the first step in the analysis is the selection of a suitable model (or class of models) for the data. Having selected a time series model the parameters of the model need to be estimated and its goodness of fit to the data has to be checked, thus checking how well the model values corresponds to the actual values.

Seasonal variation is apparent in many time series, such as sales figures and temperature reading, exhibit variation that is annual in period[5]. For example if we look at Figure 3.1, electric consumption is high in winter but lower in summer (e.g., many households use electric power to heat up their houses during the winter season.). **Trend** may be loosely defined as 'long-term change in the mean level'[5]. One difficulty with this is to decide what is meant by 'long-term'. For example climatic variables might have cyclic variation in a period of 50 years, and if only data is available for 20 years the oscillation might wrongly be taken for a trend. But if several hundreds of years were available the cyclic variation would be visible. For example, if we look at the Figure 3.1 again, there is an increasing trend from year 1974 through 2009. However we cannot decide if it is a cyclic behavior or not.

If the plot does not reveal the trend and seasonality enough smoothing can be used to enhance the visual pattern on a graph. Sometimes the seasonality and trend is not desirable. There exist methods that remove either trend, seasonality, or both.

3.2.1 Stochastic process

Stochastic process is a set of random variables that evolves through time[10, 5].

In time-series analysis, when dealing with successive observations they are usually *not* independent, and when analyzing, the successive time order of the observations has to be considered. Because the observations are dependant it is possible to predict future values with the help of previous observations. If we could predict the values exactly, the underlying model would be called *deterministic*. For example if we throw a ball from a slingshot with

a certain velocity and angle we could exactly determine where the ball should come down. But due to unknown variables such as random wind changes, spinning of the ball, and so on the ball does not land on the same spot every time. Therefore, many of models that explain observations have a stochastic process as one part, to cope with random behavior. Thus no exact prediction is possible but with help of past values a probability distribution can be used to take a good guess at future values.

3.2.2 Stationarity

When dealing with time series analysis, many of the methods and modeling procedures is dependant of the assumption that the time series is stationary. If the series is non-stationary, methods can be used to transform them to stationary. A time series is said to be stationary if its underlying stochastic process has a constant mean, variance, and autocorrelation through time. Thus it has a flat looking series, without trend and no cyclic behavior.

A time series $\{X_t, t \in \mathbb{Z}\}$ is strictly stationary if

$$(x_1, \ldots, x_n) \quad =^d \quad (x_{1+h}, \ldots, x_{n+h}) \tag{3.1}$$

for all integers h and $n \geq 1$. ($=^d$ is used to indicate that two random vectors have the same joint distribution function)

This is a very strong condition that is hard to verify empirically, therefore often a weaker type of stationarity is assumed[22]. A time series $\{X_t\}$ is weakly stationary if both the mean of X_t and the covariance between X_t and X_{t-h} are time-invariant; where h and t are arbitrary integers.

A time series $\{X_t, t \in \mathbb{Z}\}$ is weakly stationary if

(i) $\mu_x(t)$ is independent of t,

and

(ii) $Cov(X_t, X_{t-h})$ is independent of t for each h.

3.2.3 Smoothing

Data observations taken over time, consists of random variations and fluctuations that could make analysis difficult. There exist methods for reducing the confusing effect due to random variation. An often-used technique is "smoothing"[10]. This technique, when properly applied, reveals more clearly the underlying trend, seasonality. A general expression for the moving average is:

$$M_t = [X_t + X_{t-1} + ... + X_{t-N+1}]/N \tag{3.2}$$

Where t is the number of subset values from the time series set.

The Figure 3.3 shows the same data as in Figure 3.2 with the difference that a moving average function has been applied on the data. We can see a smoother curve with spikes not that overwhelming. By taking average value of adjacent observations, random fluctuations will have less influence on the mean value.

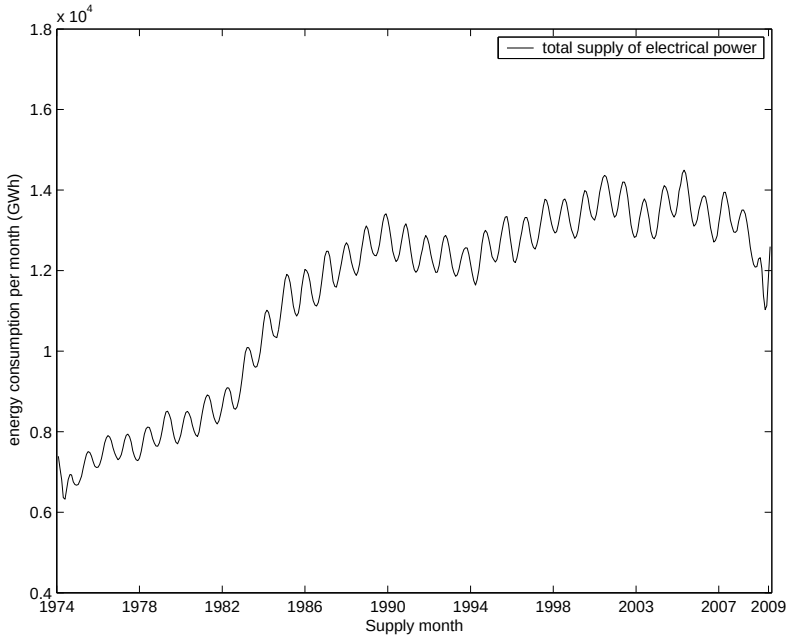

Figure 3.3: Same data as in Figure 3.2 with a Moving Average MA function applied to it.

3.2.4 Autocorrelation and Autocovariance

Let $\{X_t\}$ be a stationary time series. The autocovariance function (ACVF) of $\{X_t\}$ is

$$\gamma_X(h) = Cov(X_{t+h}, X_t) \tag{3.3}$$

The autocorrelation function (ACF) of $\{X_t\}$ is

$$\rho_X(h) \equiv \frac{\gamma_X(h)}{\gamma_X(0)} = Cor(X_{t+h}, X_t) \tag{3.4}$$

Covariance function is an important tool in the analysis of time series and especially autocovariance. Covariance is a measure of dependence between two or more random variables. In autocovariance only one variable is involved and thereby the inherent dependency is measured at different time distances apart[20].

Autocorrelation is used for example when fitting the order of p in an autoregressive model, see section 3.2.8 on the following page.

3.2.5 IID and White noise (WN(μ,σ^2))

A sequence of random variables is IID (independent and identically distributed) if each random variable has the same probability distribution as the others. It has no trend or seasonality and the random observations are simply independent and identically distributed with zero mean. A sequence of coin flips where we register if its head or tails is said to be IID.

White noise almost has the same characteristics as IID but the successive values are merely uncorrelated rather than independent. If successive values follow a normal distri-

bution (Gaussian distribution) and we have zero correlation it implies that the series is independent.

3.2.6 Random walk

A time series $\{X_t\}$ is a random walk if it satisfies

$$X_t = X_{t-1} + Z_t, \tag{3.5}$$

where X_0 is a real number denoting the starting value of the process and $\{Z_t\}$ is a white noise series[22]. Random walk is very common in the financial business, equity returns are assumed to follow this model.

3.2.7 Regression Analysis

Regression analysis is a widely used statistical method that investigates the relationships between variables. The goal is usually to seek the causal effect of one variable to another. Regression techniques have long been central to the field of economic statistics (econometrics).

3.2.8 Autoregressive processes (AR(p))

$\{X_t\}$ is said to be an AR(p) **autoregressive process of order p** if it is stationary and if

$$X_t - \phi_1 X_{t-1} - \ldots - \phi_p X_{t-p} = Z_t, \tag{3.6}$$

where $\{Z_t\} \sim \text{WN}(0,\sigma^2)$, $p \geq 0$.

3.2.9 Moving average processes (MA(q))

The MA(q) process $\{X_t\}$ is a moving average process of order q if

$$X_t = Z_t + \theta_1 Z_{t-1} + \ldots + \theta_q Z_{t-q} \tag{3.7}$$

where $\{Z_t\} \sim \text{WN}(0,\sigma^2)$ and $\theta_1, \ldots, \theta_q$ are constants. $q \geq 0$.

3.2.10 ARMA (Autoregressive-Moving average)

ARMA models are used to model the conditional expectation of the current observation X_t, given the past observations. ARMA[9] is a technique that mixes autoregressive and moving-average models. To achieve greater flexibility in fitting of actual time series, it is sometimes advantageous to include both autoregressive and moving-average terms in the model instead of a pure MA or AR process itself. The time series observations have to be stationary to use it with ARMA-modeling.

$\{X_t\}$ is an ARMA(p, q) process if $\{X_t\}$ is stationary and if for every t,

$$X_t - \phi_1 X_{t-1} - \ldots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \ldots + \theta_q Z_{t-q}, \tag{3.8}$$

where $\{Z_t\} \sim \text{WN}(0,\sigma^2)$ and the polynomials $(1-\phi_1 z-\ldots-\phi_p z^p)$ and $(1+\theta_1 z+\ldots+\theta_q z^q)$ have no common factors.

3.3 Financial time series analysis

Financial data are data related to the trading of financial assets such as shares, options, interests' rates and commodity prices. When dealing with that huge quantity of assets it is important to know what risks that are associated with it. When analyzing historical and present data the main concern are to evaluate the risks of investing in assets over time[22]. For example, when putting together a share portfolio it is a good idea to know the investment risks of it and have a balanced content. Financial data and the generated time series from them are often very uncertain and require special methods and models to analyze them.

3.3.1 Asset Returns

When looking at financial time series they are far from stationary and one way to deal with this problem is to analyze the returns of the assets, changes in price expressed as a fraction of the initial price. Instead of analyzing the prices, the returns have more favorable statistical properties that make it easier to use. There are several definitions of an asset return; two of them are presented below and they are also mentioned by Tsay[22] and Ruppert[20].

Net returns

Let P_t be the price of an asset at time t. The **net return** over the holding period from time $t-1$ to time t is

$$R_t = \frac{P_t}{P_{t-1}} - 1 = \frac{P_t - P_{t-1}}{P_{t-1}}. \tag{3.9}$$

The numerator, $P_t - P_{t-1}$, is the revenue or profit during the holding period, with a negative profit meaning a loss. The denominator, P_{t-1}, was the initial investment at the start of the holding period. Therefore, the net return can be viewed as the relative revenue or profit rate.

Log returns

Continuously compounded returns or **log returns**, are denoted by r_t and defined as

$$r_t = \ln(1 + R_t) = \ln \frac{P_t}{P_{t-1}} = p_t - p_{t-1}, \tag{3.10}$$

where $p_t = \ln(P_t)$.

If comparing net and log returns, one advantage with the continuously compounded return are that it is symmetric, while the net return are not. Positive and negative percent net returns are not equal. This means that an investment of \$100 that yields an net return of 50% followed by an net return of -50% will result in \$75, while an investment of \$100 that yields a logarithmic return of 50% followed by an logarithmic return of -50% it will remain \$100.

3.3.2 ARCH and GARCH modeling

ARCH (Autoregressive Conditional Heteroskedasticity) is dealing with randomly varying volatility and was first introduced by Engle in 1982[18]. Engle was one of the laureates of the Nobel Prize in Economics 2003 for the work on Autoregressive Conditional Heteroskedasticity[5]. Heteroskedasticity is meaning that the random variables in a stochastic process have different variances.

The process $\{X_t\}$ is said to be a ARCH(p) process of order p if it is stationary and if

$$X_t = \sigma_t Z_t, \qquad \sigma_t^2 = \alpha_0 + \alpha_1 X_{t-1}^2 + \ldots + \alpha_p X_{t-p}^2 \tag{3.11}$$

where $\{Z_t\}$ is a sequence of independant and indentically distributed (IID, see section 3.2.5) random variables with mean zero and variance 1, $\alpha_0 > 0$, $\alpha_j \geq 0$ for $i > 0$.

The basic idea in ARCH modeling is to capture the serial correlation or autocorrelation of volatility in time series[12]. It allows the conditional variance to change over time as a function of past errors leaving the unconditional variance constant[21]. With this technique it is possible to do forward-looking evaluations, forecasts of volatile series in a satisfactory way.

Ruey S. Tsay[22] mentions though that ARCH has some weaknesses. The model assumes that the volatility responds in the same way to positive as well as to negative shocks. But it is commonly known that financial assets respond differently to negative and positive shocks. The model does not tell anything about the underlying causes generating financial time series, which is one of the ideas with time series analysis. And ARCH models are likely to over predict the volatility because they respond slowly to large isolated shocks to the return series.

The **GARCH**(p, q), *Generalized* ARCH model is an extension of the ARCH model and was first introduced by Bolleslev in 1986[4].

3.3.3 Ultra High Frequency data

When analyzing time series data or financial time series data it is with equally spaced time intervals. For example, the closing price on a share every day we know that we have 28 to 31 observations in a month, or the index level at the end of every month renders in 12 observations in a year. When talking about intra-day data on the financial market it comes to thousands of observations. And these observations also come in an irregular dispersion, sometimes with second or even milliseconds apart. Intra-day data are also referred as the microstructure of finance market.

The volatility in intra-day financial data is apparent in the same way as long-term financial data. In intra-day though it appears a diurnal pattern, where it is systematically higher volatility in the beginning and the end of the trading day[7, 22].

Due to the irregularly spaced intervals of ultra high frequency data the usual methods used in econometrics cannot be applied. In Figure 3.4 we can see an example of trades of a share on the New York Stock Exchange (NYSE) where it clearly shows that the trades come in different durations. During these 14 min of time we can see that events can arise within seconds apart from each other whereas sometimes it can pass a minute before next trade

[5] `http://nobelprize.org/nobel_prizes/economics/laureates/2003/public.html`

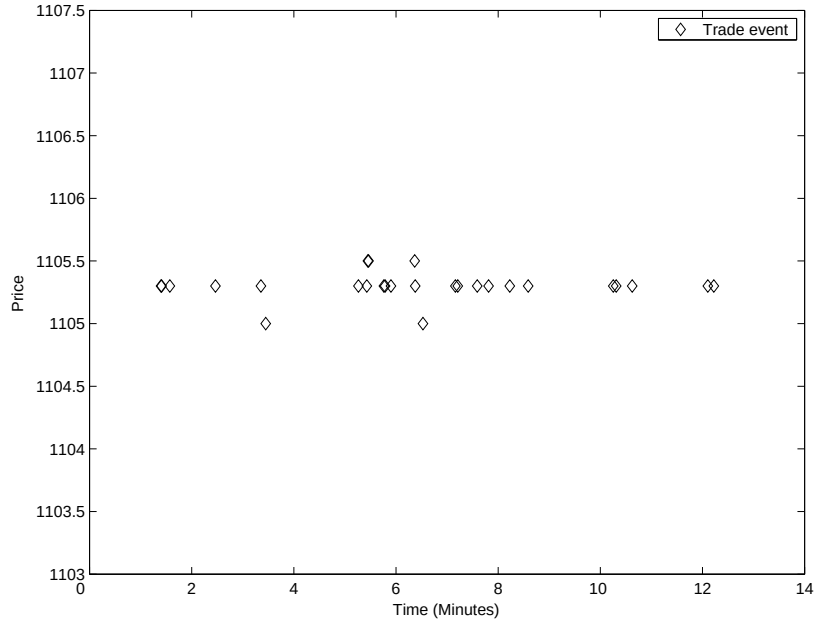

Figure 3.4: Share trade events on the New York Stock Exchange (NYSE). Showing the irregularly spaced times between incoming trades events during a short period of time (14 min).

occur. To cope with this irregularity, new methods have been developed that take this duration into account[19]. Otherwise fixed intervals could be used and some sort of interpolation has to be applied at the end of an interval if no observation is available for that interval. Or some averaging of the values contained inside for example a five minutes interval or just taking the last value in that five minutes interval. Another strategy might be to consider some other type of spacing, for example taking the every third trade regardless of what time spacing these have. However, by these techniques important information might get lost and new techniques have been developed to cope with this irregularity. One new method is the Autoregressive Conditional Duration (**ACD**) model whose explicit objective is the modeling of times between events and was first introduced by Engle and Russell in 1998[7]. The idea is to view the observations as point processes. These processes arrive on the time-axis in a randomly manner and therefore could be seen as stochastic variables. From that the durations between consecutive points could be modeled as a stochastic process. The ACD model is seen as the counterpart of the GARCH model for duration data because those two models share common features. In the same way the GARCH model capture the volatility, the ACD model in a similar way captures the duration clustering[22, 7]. Thus small (large) durations are followed by other small (large) durations. Longer durations usually imply lack of trading activities and thus signifying a period with no new information which in hand gives valuable information of the intraday activities.

Engle and Russell mention in[7] 1998 that the durations also have the same diurnal patterns mentioned before, thus the time between durations tend to be shortest near to the open and just prior to the close.

Chapter 4

Design and Implementation

The goal from the beginning was to develop a tool that could inspect and compare time-series or tick-by-tick data and control them for equality. The tool was intended to be used together with a graphical interface (GUI). In this Chapter the design and system is presented and the logic and math used to compare the time-series.

4.1 Numerical Approach

The actual question from the beginning was how to define two series being equal even though they weren't exactly equal mathematically. A prerequisite for this was to gain more understanding in general of time-series and the properties that is associated with them and especially within the financial area. In Chapter 3 on page 7 the in-depth study of time-series is accounted for and that was the starting point in solving the problem. Even though the problem with skews in time of time-series is well known in the financial market no specific research was found that addressed this particular situation.

The intention was to find any way or method in time-series analysis that could be used to identify that two time-series where equal, even though that the spacing in time of the two series differed. Most of the techniques and methods explained in Chapter 3 on page 7 are used to explain the behavior of the underlying mechanisms generating the series and to make forecasts. After studying time-series and the problem a bit further a numerical solution was chosen.

Original

Date	Time	Symbol	Bid Price	Ask Price	Bid Size	Ask Size
09/26/2007	10:45:55.**461**	VOD	41.03	41.05	62	58
09/26/2007	10:45:55.**461**	VOD	41.03	41.05	80	69
09/26/2007	10:45:55.**478**	VOD	41.04	41.05	23	82
09/26/2007	10:45:55.**479**	VOD	41.04	41.05	25	82
09/26/2007	10:45:55.**490**	VOD	41.04	41.05	87	99

Table 4.1: Quote data illustration. Each tuple corresponds to one quote, a tick. This table represents the original data.

In Table 4.1 we can see an illustration sample of how a time-series can look like when coming from a vendor. Different vendors can have different data sets, thus what kind of

Copy

Date	Time	Symbol	Bid Price	Ask Price	Bid Size	Ask Size
09/26/2007	10:45:55.**000**	VOD	41.03	41.05	62	58
09/26/2007	10:45:55.**000**	VOD	41.03	41.05	80	69
09/26/2007	10:45:55.**000**	VOD	41.04	41.05	23	82
09/26/2007	10:45:55.**000**	VOD	41.04	41.05	25	82
09/26/2007	10:45:55.**000**	VOD	41.04	41.05	87	99

Table 4.2: Quote data illustration. Each tuple corresponds to one quote, a tick. This table represents a copy of the data in table 4.1 on the preceding page. The diffrerence is that the time column has suffered a time-accuracy loss due to truncation.

.

information that the columns contains. Usually there is around 10-20 columns of data. As told before a quote is an update on an equity that settles the best price for the moment. The *bid-price* is what an investor is willing to pay and the *ask-price* is what the seller is willing to sell his share for. *Bid-size* is the number of shares that the buyer is willing to buy and the *ask-size* is the number of shares the seller is ready to let go. The *symbol* states what equity we are talking about, VOD in this case stands for Vodafone Group Plc. In other cases the symbol or equities is usually mixed up and thousands of other shares would be interleaved.

If we compare Table 4.1 on the preceding page with Table 4.2 we can see that they are almost equal besides from a number of milliseconds apart. Even if it was one millisecond off at one of the tuples they wouldn't be equal when testing for it. Nevertheless they are considered to be equal in this context. So the approach is just to compare every corresponding tuple to each other and check that every column has an exact match, besides the time. The time is numerically checked that the difference is within an acceptable range. That is:

$$abs(a - b) \quad < \quad c \tag{4.1}$$

Where a is the timestamp in the first table and b is the timestamp in the second table. And c is the value that is the acceptable difference between the two times. What the acceptable time difference is set to be are not fixed, in the tool implementation this could be altered, thus the time difference requirements could be different from one case to another.

4.2 Numerical Results

To get a picture of the whole time-series a few variables had to be summed up and these variables are also part of the results that is represented in the graphical interface. When calculating the time-differences milliseconds are used and in this case its milliseconds from midnight and forward. In the following table the variables are explained.

Let $\{X_{0t}\}$ and $\{X_{1t}\}$ be time-series, then:

To both have the total amount and absolute amount values is to be able to see if the residual is lagging at every point or if it is ahead at some point. I.e. if every time point is 1 ms behind the total and the absolute value should be the same, but if one or more values is ahead at some point there will be a difference between the total and the absolute values.

Total (T) is the total amount of difference that has been measured by adding the difference from every tuple or tick.	$T = \sum_0^t (X_{0t} - X_{1t})$
Absolute (T_a) is the total absolute amount of difference that has been measured by adding the difference from every tuple or tick.	$T_a = \sum_0^t \lvert(X_{0t} - X_{1t})\rvert$
Average is the total amount of difference that has been measured by adding the difference from every tuple or tick and then divided by the total number of tuples or ticks.	$A = \sum_0^t \frac{\lvert(X_{0t} - X_{1t})\rvert}{t}$

Table 4.3: The resulting variables from the time-series analysis.

4.3 Database

A *column*-based proprietary database was used. Most databases are *row*-based because it is often natural to retrieve the data row-by-row, where the data in the row is strongly related. When storing the information in memory the rows are stored successively after each other. If searching for, let's say, 20 rows out 1000 it is very fast to retrieve that data because it is grouped together in the memory storage. In a column-oriented database on the other hand each column is stored successively after each other. And in the same way to get 20 rows from this database takes a bit longer because each row are scattered over the memory storage. However if we are getting all the information from only one column, let´s say all the timestamps from Table 4.1 on page 17, it will go much faster because the information is gathered successively.

In the financial business it is very common to retrieve information column-wise instead of row-wise and that is the reason why column-based databases are widely used.

The communication and information retrieval with Nomura´s databases goes through their own libraries and layers. The syntax used to communicate with the proprietary database is very similar to the SQL standard with a few differences.

4.4 System Overview

The interface and logics are implemented in Java 6.0. Figure 4.1 show an overview of the application where the different parts can be seen as layers of dataflow. The time-series' is fetched by *DatabaseHandler*, where the *database-query* is built, through Nomura´s own database manager layers. The information that comes from the lower layers is in the form of JavaBeans[2] or more exact in this case like DynaBeans[1]. Javabeans is actually only plain old java objects but with some naming and structure conventions. They are used to encapsulate many objects into a single object (the DynaBean), so that they can be passed

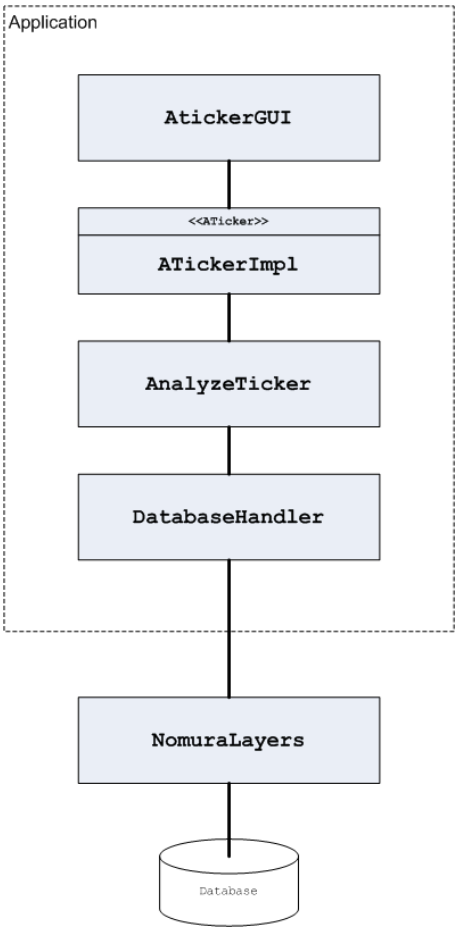

Figure 4.1: System Overview

around as a single bean object instead of as multiple individual objects.

The retrieved data is then extracted from the bean and then analyzed and compared in *AnalyzeTicker* where the result is then organized and passed back through *ATicker* to the graphical interface. Threading is also used in AnalyzeTicker to make it possible to do several analyses at the same time, since one analysis could take many hours and to be able to make an analysis in between it was deemed that threading was required.

The amount of data that is going to be analyzed is sometimes very large. To cope with this the data has to be divided and retrieved in smaller chunks. This will increase the time it takes to complete the task, but it was not a requirement that it should be very fast.

4.5 Graphical Interface (GUI)

The graphical user interface is as told before built in Java and the swing-toolkit is used. To get a different look at the swing-components, substance-look-and-feel (third-party) library[3] was used and the *Business Black Steel* skin.

Some tasks took a bit too long time to execute in the *Event-Dispatch-thread*, thus the

main thread that repaints the graphical interface had to be put in a worker thread. Following the introduction of Java SE 6.0 a new class *SwingWorker* has been added which makes it easier to handle tasks in the background. In this application the swingworker-class was used to cope with this problem.

Chapter 5

Results

In this chapter the outcome of the thesis are presented; the resulting application and how the requirements have been fulfilled. The intention from the beginning was to get a functioning application that could be used inside the Nomura Company that could compare time-series for equality even though they weren't exactly equal mathematically.

5.1 Graphical Design

There weren't any special requirements on the graphical user interface besides the fact that it should be graphical. When designing the interface the approach was to make it as easy as possible to use but at the same time to get all the necessary information and choices available and displayed to the user.

In Figure 5.1 we see the main tab which is the starting point. A tabbed pane was used to get as much space available for every function. Also if the application is extended in the future it is easy to add a new tab with new functionality.

In the main tab (*ATicker*) the first thing to do is to choose the date range and location in the left side of the pane. Location states what database that is to be used and which table in that database. Date range decides between what dates the time-series is going to be analyzed. This means that no more accuracy than daily series can be chosen at the moment; hence an interval of 1 hour can't be analyzed separately. Location has to be set for every time-series or ticker to be able to locate it. *Service* and *table* states on what service to search on and where in the database the ticker is located.

When date and location is set the table information is fetched with the help of '*Get Table*'. The checkboxes underneath every column-name indicates if the actual column should be included in the analysis or not. Sometimes the columns are redundant or of some other reason not wanted in the analysis. The same number of columns has to be chosen so that every column in ticker1 has a corresponding one in ticker2. If for some reason, when getting the columns, the tables don't match; drag-and-drop can be used. I.e. if the *val-column* in ticker1 is in column-position 2 and the *val-column* in ticker2 is in column-position 3, either one of the columns can be moved with the help of the mouse-pointer to a new position so that *val-column* in ticker1 has the same as *val-column* in ticker2.

The comboboxes *Date* and *Time* decides which of the columns that states the time and date of the time-series. The column names on date, time and symbols etc. is not consistent from case to case so that has to be chosen before a compare could take place. The *Filter* combobox decides what column to use with filtering. As said before the time-series from

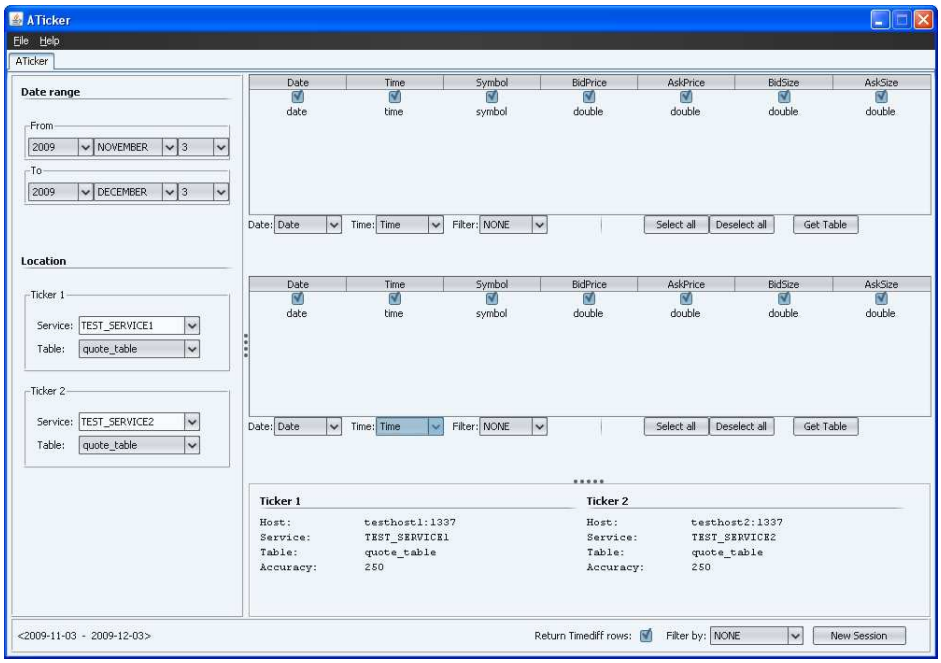

Figure 5.1: System Overview

different equities are interleaved and forms one big total series. If only one share is to be analyzed, filtering can be used to extract that one from the series.

When the rearranging and the selection of the columns are finished a *New Session* can be started. '*Return Timediff rows*' checkbox makes an alternative to return the rows that exceed the *Accuracy*-limit; to alter the accuracy-limit see User-guide in Appendix A on page 33. The accuracy-limit states when the time-difference between two ticks is not acceptable (see Equation 4.1 on page 18). The '*Filter By*' combobox decides what symbol or share that is going to be analyzed. If filter is not changed every row is going to be included in the analysis.

After a new session is started from the aticker-tab a new tab is created, an example is shown in Figure 5.2. For every new session a new tab is added and several analyses can be active at the same time. After a new aticker-tab has been created the session has to be started. In the bottom of the tab choices for Start, Stop and Pause is available.

The two tables in the middle of the panel contain the rows that have been returned from the finished analysis. Either the rows that have exceeded the time-accuracy-limit or it is the rows that contain mismatched cells, thus cells that are not equal to the corresponding ticker cells.

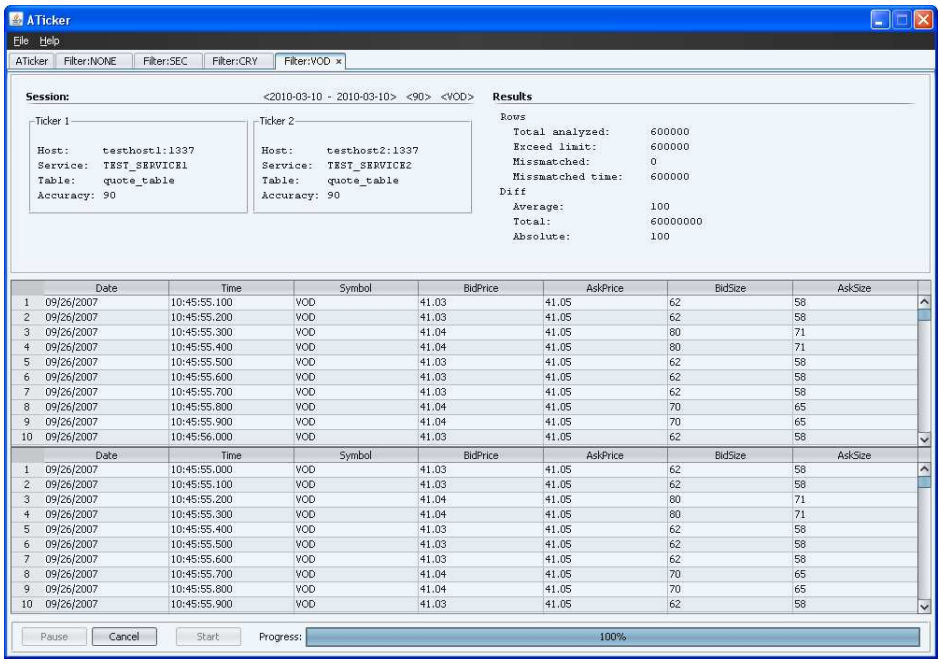

Figure 5.2: System Overview

At the left upper corner of the panel, information of the current session is displayed. And when the analysis is finished the conclusion of it is presented at the right upper corner. A few of the variables displayed are mentioned in Section 4.2 on page 18 and in addition to that the following are displayed:

Total analyzed:	Shows the total amount of rows that has been analyzed.
Exceed limit:	Show the number of rows that exceeded the time-limit constant.
Mismatched:	Shows the number of rows that have any cell, besides time, that does not match the corresponding cell in the other time-series.
Mismatched time:	Shows the number of rows where the time-stamps differ.

Table 5.1: Gui variables

See the User-guide for more details, Appendix A on page 33.

5.2 Time Series Equality

The original task was to compare two time-series for equality even though they weren't exactly equal. When setting up the environment described in the previous chapter a com-

parison is made and if the two series is assumed to be equal there can't be any mismatched rows and additionally there can't be any rows that exceed the time-limit-accuracy. If that holds the series is assumed to be equal.

Chapter 6

Conclusions

The goals for this thesis was to study time-series and the analysis of them and from that knowledge develop a tool that could compare time-series for equality. This project has resulted in a time-series tool that is able to compare time-series and to decide if these series are considered equal according to some specified criteria.

The definition of when two time-series are exactly equal in mathematical terms are when they don't differ at all. The time-series analyzed in this thesis do differ sometimes even though it is by very little. How little or how much they could differ in time to be considered equal isn't established in this thesis. This time accuracy limit is left as a setting and can be altered from one case to another.

No major implementation difficulties arose during the project, it was quite straight forward. When it comes to time-series and the analysis of it, the pre-knowledge was very limited at the beginning of the project. The scope of time-series and the properties of these are quite comprehensive and for this reason the in-depth study took some more time than planned; even though a numerical solution was used in the end.

6.1 Limitations

One limitation is, as have has been mentioned before, that when analyzing a time-series only whole days can be analyzed, thus only one hour can´t be chosen. This isn't a problem at the moment but if a need for it should arise, it will be easy to implement. Another thing is that some of the primitive data-types in the database had some problem of being converted to Java Beans in the lower layers. This affects the retrieval of the tables from the database and results in an error message. Partly for that reason, the decision of adding functionality to decide participant columns in the analysis was made.

6.2 Future work

One thing that has already been mentioned in limitations is to implement the ability to compare time-series that are shorter, intraday ranges. Another thing is to extend the functionality of the analysis. The current analysis is comparing for equality of time-series but with some extension, other types of analysis could be added. Because of the 'tabbed panes' in the graphical interface, it could be easy to extend the GUI with an extra functionality tab.

Chapter 7

Acknowledgements

I would like to thank my supervisor Håkan Lindkvist at Nomura Sweden AB for the support and help during the entire process of the thesis. I also would like to thank other personnel that helped me during my residence at Nomura.

I also would like to thank my internal supervisor, Oleg Seleznjev at the Department of Mathematics and Mathematical Statistics at Umeå University, for the support and help in Mathematics and the writing process.

References

[1] DynaBeans. http://commons.apache.org/beanutils/api/org/apache/commons/ beanutils/DynaBean.html (visited 2009-10-30).

[2] JavaBeans. http://java.sun.com/docs/books/tutorial/javabeans/index.html (visited 2009-10-30).

[3] Substance look and feel. https://substance.dev.java.net/ (visited 2009-10-30).

[4] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal Of Econometrics*, 31:307–327, 1986.

[5] Chris Chatfield. *The Analysis if Time Series*. Chapman & Hall/CRC, Florida, Boca Raton, 2004.

[6] J. E. Dennis, D. M. Gay, and R. E. Welsch. An adaptive nonlinear least-squares algorithm. *ACM Trans. Math. Software*, 7:348–368, 1981.

[7] Russell J. R Engle R. F. Autoregressive conditional duration: A new model for irregularly spaced transaction data. *Econometrica*, 66:1127–1162, 1998.

[8] J. Wolters G. Kirchgässner. *Introduction to Modern Time Series Analysis*. Springer Verlag, Berlin Heidelberg, 2007.

[9] Gregory C. Reinsel George E.P Box, Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control, third edition*. Prentice Hall, New Jersey, USA, 1997.

[10] Gwilym M. Jenkins George E.P Box. *Time Series Analysis, forecasting and control*. Holden-Day, San Fransisco, 1970.

[11] Jan Grandell. Time series analysis. http://www.math.kth.se/matstat/gru/5b1545/ts.pdf (visited 2009-10-30).

[12] A. Craig MacKinlay John Y. Campbell, Andrew W. Lo. *The Econometrics of Financial markets*. Princeton Univ. Press, Princeton, N.J., 1994.

[13] L. Lamport. Text Processing using LaTeX. http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/ (visited 2004-12-29).

[14] P. Lindström. Writing Thesis Reports at CS-UmU using LaTeX. Technical report, Dept. of Comp. Sc., Umeå University, Umeå, Sweden, not published.

[15] Ulrich A. Muller Richard B. Olsen Olivier V. Pictet Michel M. Dacorogna, Ramazan Gencay. *An introduction to high-frequency finance*. Academic Press, San Diego, USA, 2001.

[16] Maria Pacurar. Autoregressive Conditional Duration (ACD) Models in Finance: A Survey of the Theoretical and Empirical Literature. http://ideas.repec.org/a/bla/jecsur/v22y2008i4p711-751.html (visited 2009-10-30).

[17] R.A Davis P.J Brockwell. *Introduction to Time Series and Forecasting*. Springer Verlag, New York, 1996.

[18] Engle R.F. Autoregressive conditional heteroscedasticity with estimates of variance of united kingdom inflation. *Econometrica*, 50:987–1007, 1982.

[19] Jeffrey R. Russell Robert F. Engle. Analysis of High Frequency Financial Data. http://home.uchicago.edu/ lhansen/survey.pdf (visited 2009-10-30).

[20] David Ruppert. *Statistics and Finance: An Introduction*. Springer Verlag, New York, 2004.

[21] Kenneth F. Kroner T. Bollerslev, Ray Y. Chou. Arch modeling in finance. *Journal Of Econometrics*, 52:5–59, 1986.

[22] Ruey S. Tsay. *Analysis of Financial Time Series*. Wiley & Sons, Canada, 2002.

Appendix A

User's Guide

The following chapter explains the different parts in the graphical user interface; the order to choose the settings and the meaning of them. In Figure A.1 and A.8 on page 36 the two tab-panes, that is used throughout the analysis, is shown with red circle letters. Every red circle letters area is explained further in following subsections. Throughout the application description, duplicates occur on the settings, *Ticker 1* and *Ticker 2* settings. These are individual settings for the two time-series that is going to be analyzed.

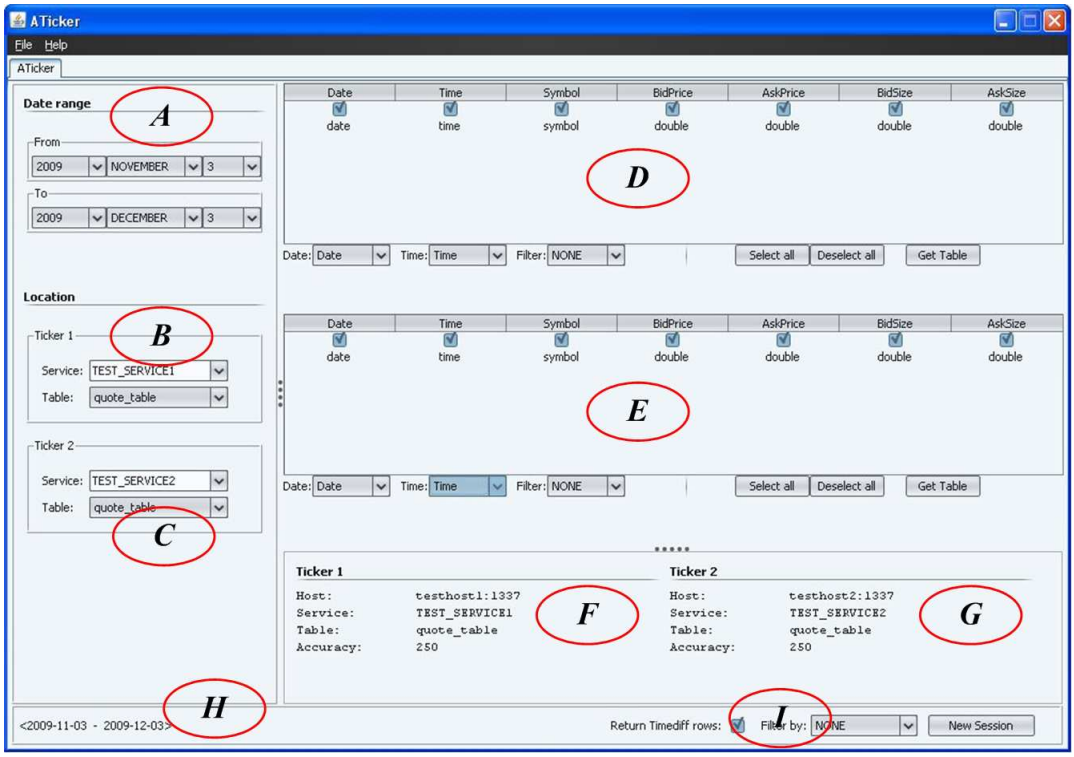

Figure A.1: ATicker Tab

A.1 ATicker tab

ATicker tab is the starting point when doing an analysis. The date range and where the time-series are located have to be decided. When the time-series is located they are represented in two tables ***D*** and ***E*** where the appropriate settings can be carried out. First of all to check if the two tables are the right ones and then to choose which of the columns that is going to be included in the analysis.

A.1.1 *A*

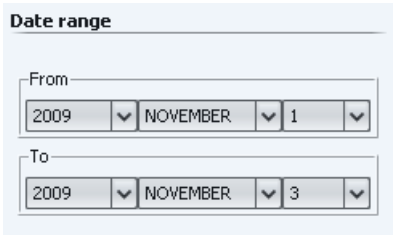

Figure A.2: Date range.

Decides between what dates that the time-series' is going to be analyzed.

A.1.2 *B*

Figure A.3: Location selection

The combo-boxes determine the location of the *Ticker 1* time-series. **Service** states what service to connect to and **Table** states which of the tables in the database to pick.

A.1.3 *C*

Same as section A.1.2 but for *Ticker 2*.

A.1.4 *D*

When the **Get Table** button is pressed, a table representation of the *Ticker 1* time-series is fetched and displayed in ***D***. The order of the columns can be changed with the help of drag-and-drop. This is made possible because the column-names are not always the same and the order of the columns in the database-tables might differ. The goal is to have the column in table ***D*** at the same position as the corresponding column in table ***E***.

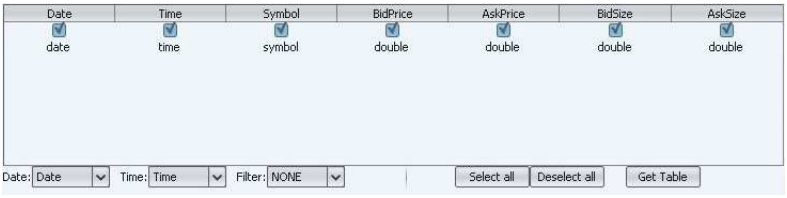

Figure A.4: Table representaion of the time-series

The *checkboxes* determines whether the column is to be included or not. Once again because the table-names are not consistent the columns that states the date and time of the time-series' has to be chosen; the **Date** and **Time**-combo-boxes determines this.

Select All sets all the checkboxes to true and **Deselect All** sets all the checkboxes to false. **Filter** decides if any filtering is going to be used and if so the filter column is chosen. *NONE* indicates no filtering.

A.1.5 *E*

Same as section A.1.4 on the facing page but for *Ticker 2*.

A.1.6 *F*

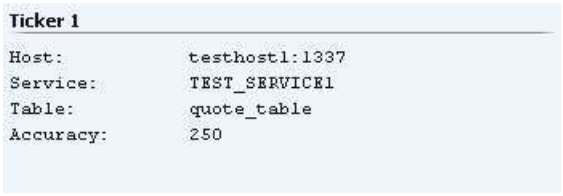

Figure A.5: Time-series information

Shows information of the settings in the properties dialog and what services and tables that are chosen for *Ticker 1*.

A.1.7 *G*

Same as section A.1.6 but for *Ticker 2*.

A.1.8 *H*

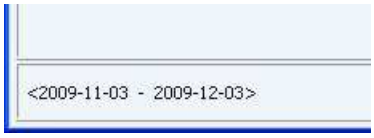

Figure A.6: Current date

Displays what date range that are chosen in ***A***.

A.1.9 *I*

Figure A.7: Start new session

New Session starts a new analysis session with the current settings in an **Analyze tab**. **Return Timediff rows** states if rows that exceed accuracy limit should be returned or not. When making the analysis and there exist rows that exceed the time accuracy limit, the whole table or a sample of it can be returned for inspection in the Analyze tab. **Filter by** combo-box is used to select the required filter-symbol when filtering is chosen.

A.2 Analyze tab

When a new session is started an Analyze tab is created. When the analysis is started and finished the result is returned in this tab, the numerical results and possibly the rows that are mismatched in some way.

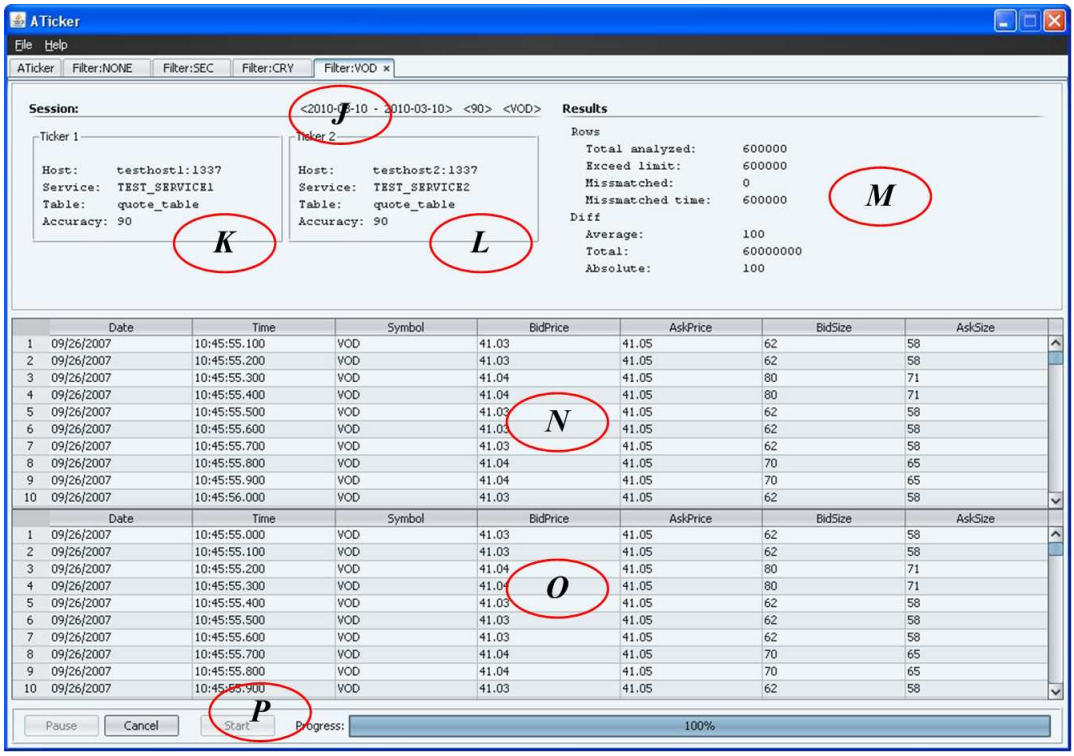

Figure A.8: Analyze Tab

A.2.1 *J*

Figure A.9: Session info

Displays the date range of the time-series to be analyzed, the time-accuracy setting and if and which filter-symbol that is used.

A.2.2 *K*

Figure A.10: Ticker information

Displays information of the first time-series: Ticker 1. The location of the time-series and what settings that are used in this specific session analysis.

A.2.3 *L*

Same as section A.2.2 but for *Ticker 2*.

A.2.4 *M*

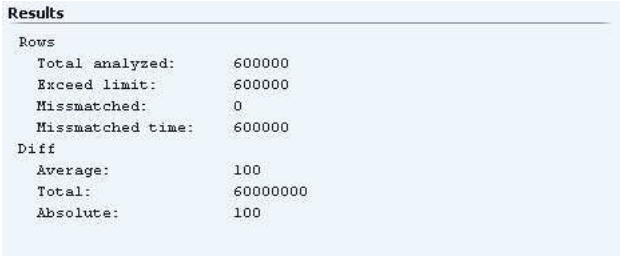

Figure A.11: Result variables pane

Displays the numerical results from the analysis. See Table A.1 on the next page for an explanation.

Rows	
Total analyzed:	Shows the total amount of rows that has been analyzed.
Exceed limit:	Show the number of rows that exceeded the time-limit constant.
Mismatched:	Shows the number of rows that have any cell, besides time, that does not match the corresponding cell in the other time-series.
Mismatched time:	Shows the number of rows where the time-stamps differ.
Diff	
Average:	Is the total amount of difference that has been measured by adding the difference from every tuple or tick and then divided by the total number of tuples or ticks. $A = \sum_0^t \frac{\|(X_{0t} - X_{1t})\|}{t}$
Total:	Is the total amount of difference that has been measured by adding the difference from every tuple or tick. $T = \sum_0^t (X_{0t} - X_{1t})$
Absolute:	Is the total absolute amount of difference that has been measured by adding the difference from every tuple or tick. $T_a = \sum_0^t \|(X_{0t} - X_{1t})\|$

Table A.1: Explanation of the result variables.

A.2.5 *N*

	Date	Time	Symbol	BidPrice	AskPrice	BidSize	AskSize
1	09/26/2007	10:45:55.100	VOD	41.03	41.05	62	58
2	09/26/2007	10:45:55.200	VOD	41.03	41.05	62	58
3	09/26/2007	10:45:55.300	VOD	41.04	41.05	80	71
4	09/26/2007	10:45:55.400	VOD	41.04	41.05	80	71
5	09/26/2007	10:45:55.500	VOD	41.03	41.05	62	58
6	09/26/2007	10:45:55.600	VOD	41.03	41.05	62	58
7	09/26/2007	10:45:55.700	VOD	41.03	41.05	62	58
8	09/26/2007	10:45:55.800	VOD	41.04	41.05	70	65
9	09/26/2007	10:45:55.900	VOD	41.04	41.05	70	65
10	09/26/2007	10:45:56.000	VOD	41.03	41.05	62	58

Figure A.12: Error-rows table

Shows a table that contains possible error rows from *Ticker 1*. Either there are rows that have mismatched cells in the corresponding time-series or rows that exceed the time-

accuracy limit. There is a limit of 20000 rows that could be returned to the table, if more error-rows exist then that they will be rejected. The intention is to be able to make a quick inspection of how and in what way the time-series differ and there is no need to load all the error-rows.

A.2.6 *O*

Same as section A.2.5 on the facing page but for *Ticker 2*.

A.2.7 *P*

Figure A.13: Process Bar

When a new session analyze has been created it has to be started, *Start*, before it returns any results. If necessary the analysis can be Canceled, *Cancel*, or paused, *Pause*, during the progress. The *Progress* bar checks the status during the analysis and gives an idea of how many rows there are left to analyze.

A.3 Properties Dialog

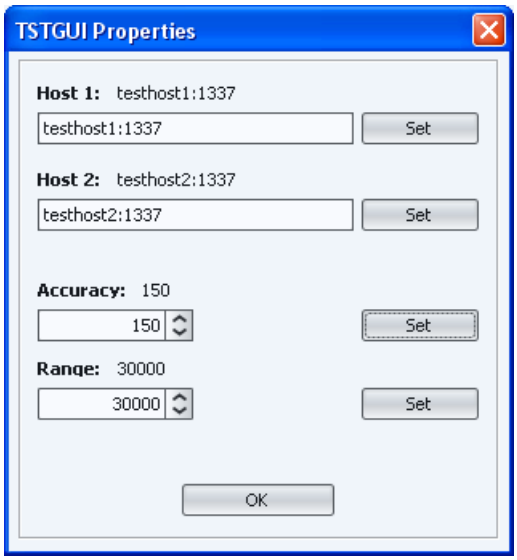

Figure A.14: Properties Dialog

Host 1 and **Host2** sets the hosts where the two time-series are located. **Accuracy** is given in milliseconds and sets the maximum time difference that the two time-stamps is acceptable to have. If the two time-stamps differ more than the selected value the row or tick are not considered to be equal. **Range** sets the number of beans or rows that are fetched from the database at one call. The speed will increase if a higher number is chosen but

will require more heap memory. If a higher value than default is chosen on range the heap size has to be increased on the runtime environment. This can be done with the following arguments to the java start command:

```
-Xms<initial heap size> - defines the initial java heap size

-Xmx<maximum heap size> - defines the maximum java heap size
```

The default maximum heap size is 128 MB. A good idea is to set both to the same value.