FAKULTÄT FÜR INFORMATIK

der Technische Universität München

Master's Thesis in Informatik

Design of an Interactive and Web-based Software for the Management, Analysis and Transformation of Time series

Kehinde Fawumi

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatik

DESIGN OF AN INTERACTIVE AND WEB-BASED SOFTWARE FOR THE MANAGEMENT, ANALYSIS AND TRANSFORMATION OF TIME SERIES

KONZEPTION EINER INTERAKTIVEN WEB-BASIERTEN SOFTWARE ZUR VERWALTUNG, ANALYSE UND TRANSFORMATION VON ZEITREIHEN

Author:	Kehinde Fawumi
Supervisor:	Matthes, Florian; Prof. Dr. rer. nat.
Advisor:	Reschenhofer, Thomas; M.Sc.
Submission:	12.05.2015

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munchen, 12.05.2015 Kehinde Fawumi

Abstract

Every day, time series data are generated in large volumes in a wide range of applications in nearly every organization. However, the management and analysis of these data still pose a great challenge to end users who have little programming experience and little knowledge of time series analysis models. Although, many tools exist for time series analysis, a review of these tools shows that they are usually designed for data experts and analysts.

In this research, an interactive web based time series software is designed for ease of use by end users. The software design aligns with typical properties of an end user oriented software for managing, analyzing and transforming time series.

Firstly, this thesis reports on the current state of research on time series and their commonness in private and public spreadsheets. Time series are identified in real world spreadsheets and results show that 14 percent of spreadsheets in the EUSES corpus and Enrons corpus are time series. Then, a review of some existing time series tools is made. This review reveals that:

1. Only a few of these tools are easy to use for end users. Hitherto, most time series tools have been developed for usage by professional analysts and data scientists.
2. Most of the tools give poor support for the transformation of time series; which involves the reduction of time-stamped data to time series or the conversion of time series from one level of time frequency to another.

A set of functional requirements for the thesis software is then generated from the review of the existing tools. These requirements form the basis for the software design in this research. The usage scenarios of the time series software designed are illustrated using mockups. This clearly illustrates how users can work with the software to effectively manage, transform and analyze time series data.

List of Figures

List of Tables

Contents

Part 1: Introduction and Theory

1 Introduction

1.1 Motivation

The amount of digital information in the world today is unimaginably enormous and is growing even more rapidly every day. Large amounts of time-stamped data are being processed and measured on the internet daily. For instance, Facebook was revealed in a white paper that its users have uploaded more than 250 billion photos, and are uploading 350 million new photos each day. Also, Google manages over 20 petabytes of user-generated data every day. At its peak in year 2012, Amazon was selling 306 items every second. Wal-Mart, a retail giant, handles more than 1 million customer transactions every hour, feeding databases estimated at more than 2.5 petabytes (Big Data Statistics, 2012). All these examples justify the ubiquity of time-stamped data. Every day, 2.5 quintillion bytes of data (both time-stamped and otherwise) are created. These data come from digital pictures, videos, posts to social media sites, intelligent sensors, purchase transaction records, cell phone GPS signals, to name a few (Zicari, 2013).

Also, large volumes of data are being collected in almost every (scientific) field, usually in the form of time-stamped data and time series. Time series are collections of events or observations, predominantly numeric in nature, sequentially recorded on a *defined* regular or irregular time basis (Castillejos, 2006). Time series data are being generated at an unprecedented speed and volume in a wide range of applications in almost every domain. For example, daily fluctuations of the stock market, traces produced by a computer cluster, medical and biological experimental observations, readings obtained from sensor networks, position updates of moving objects in location-based services, etc. are all represented in time series (P.Wang, H.Wang, & W.Wang, 2011). Time series are thus becoming increasingly important in nearly every organization and industry, including banking, finance, telecommunication, medical sciences and transportation. Banking institutions, for instance, rely on the analysis of time series for forecasting economic indices, elaborating financial market models, and registering international trade operations. In medical sciences, analysis of time series nowadays covers a wide range of real-life problems including gene expression analysis and medical surveillance.

Notably, the availability of these enormous time series data has brought about huge advantages in many areas. There is no doubt that proper management and analysis of time series has the potential to become a driving force for innovation and decision making. If managed well, time series can be used to unlock new sources of economic value, provide fresh insights into science and hold governments to account. However, the easy management and analysis of these data still pose a great challenge to scientists worldwide. Despite the abundance of tools to capture and process all this information, issues like: ensuring data security, easy management of data, analysis of huge datasets, optimizing response time for data retrieval, protecting privacy etc. are still being faced as information is shared widely around the world.

Consequently, researchers, data analysts, statisticians and computer scientists are greatly interested in finding the optimal processes and tools for managing and analyzing time series. This has resulted in a large amount of research on new methodologies and technologies for managing, analyzing, transforming, and visualizing time series data. Perhaps one major research areas is in end-user development (Nardi & Miller, 1990). In order to validate the viability, utility and interactivity of time series tools being rolled out, it is imperative to empower end users as key players in the development of best management strategies for time series data. Most programs today are written not by professional software developers, but by people with expertise in other domains working towards goals for which they need computational support. End users simply utilize the functions within an application to automate programming tasks. For example, a teacher might write a grading spreadsheet to save time, or an interaction designer might use an interface builder to test some user interface design ideas (Andrew, Robin, Laura, & Alan, 2010).

Often, end-user programmers have a set of requirements which differs from those of professional developers. End-user programmers generally program for themselves, a friend, or a colleague. The end-user is the customer/user and is therefore programming to achieve a personal goal, not programming to fulfill someone else’s; and there are no communication issues. It is on the basis of this difference that researchers have begun to study end-user programming practices and invent new kinds of technologies that collaborate with end users to improve the quality of tools for data management (Burnett, 2009).

One notable application area of end user development is in spreadsheets. The spreadsheet interface is easily adaptable for end user development. More so, the interface is accompanied with a formula language which supports users with little or no formal training in programming. This adaptability of spreadsheets for end user development derives from two properties of their design:

- Computational techniques that match users' tasks and that shield users from the low-level details of traditional programming, and

- A table-oriented interface (grid which permits users to view, structure and display data), that serves as a model for users' applications.

The power of spreadsheets comes from the combination of these properties; they are dependent and each will require the other to solve the spreadsheet user's two basic problems: computation and presentation (Nardi & Miller, 1990).

This thesis derives from the reasons for the success of spreadsheets, and applies these general principles for the design of an interactive web-based tool for time series data management and analysis.

1.2 Problem Statement

Institutions often want to discover knowledge from their time-stamped data. For instance, Wal-Mart wants to perform location-based analysis of over 1 million hourly transactions; Facebook wants to forecast how many millions of photos users will upload in the future; etc. More simplified examples include: a grocery store performing an analysis on daily sales of a product (e.g. Rice); a farmer tracking the number of crops harvested daily etc.

Business owners are often interested in performing forecasting, for economic reasons. *We all want to have crystal balls that can predict the future*. However, such analysis or forecasting can be difficult because of the level of experience users have in using the available tools. Other common problems that institutions face are:

- Unavailability of skilled analyst to perform detailed analysis
- Need for frequent updates to the analyses
- Huge number of datasets/transactions to be analyzed
- Large number of data in each transaction
- Need to convert time-stamped data to time series before analysis
- Need to discover and understand the appropriate statistical models for analysis.

At best, a skilled analyst can analyze a single dataset or time series by using a combination of relevant models for time series analysis, software based on proven statistical theory, and personal knowledge or experience. However, the task of frequently generating large number of analyses requires some degree of automation and simplification.

This thesis aims to contribute to solving the above-identified challenges. It is imperative that a simple tool and process for performing essential time series management and analysis tasks (e.g. data processing, data organization, clustering, modelling, analysis, forecasting and visualization) are developed and deployed. Hitherto, such data tasks are performed by technical experts, professional statisticians, data analysts etc. using advanced methodologies which are only understood by specialists. As a result, many institutions pay highly for professionals and tools to manage and analyze their time series – activities which are frequently performed.

Most of the available software tools for time series management and analysis are generally not suitable for use by inexperienced users. The following assertion by (Beiwald, 2009) further justifies the validity of these challenges: "MATLAB programming language is weak for standardized data analysis and visualization. It sometimes doesn’t seem to be much more than a scripting language wrapping the matrix libraries. R is pretty good (scheme-derived, smart use of named args, etc.), but only if you can get past the bizarre language constructs and weird functions in the standard library. Everyone says SAS is very bad! Microsoft Excel, which is widely used for data analysis has been reported by users to be hideous for analysis and visualization of time series data."

The disadvantages highlighted above is not so different for others data analysis tools like: SAS/Econometrics and Time Series Software, Mathematica, MINITAB, Statistical Package for the Social Sciences (SPSS), Systat, DTREG Time Series Analysis and Forecasting, Weka, GMDH Shell, R (Programming Language), GRETL among others. In a later chapter of this thesis, I made a comprehensive evaluation of the most widely-used time series tools. I identified the core strengths and weaknesses of these tools with respect to their support and functionalities for working with time series. Our emphasis is on the usability of the tools for end users; users with little or no knowledge of time series modelling and analysis. The deductions from evaluating the functionalities of these time series analysis packages, while benchmarking against all desirable functionalities, leads to deriving specific functional requirements which builds up to the design of an interactive and web-based software for the management, analysis and transformation of time series.

The next section describes the goals of this research.

1.3 Goal of the Thesis

The principal objective of this research is to derive requirements for, and to design a user-oriented web-based software for use by non-techies in managing, transforming, analyzing and visualizing time series data.

To achieve this, this thesis aims to answer the following questions:

1. What are Time-series and what features distinguishes them from other data types?
   This research will focus on understanding time series, identifying their specific features and how they differ from other data types identified in past data analysis researches.
2. How common are time-series patterns in spreadsheets today?
   I will identify uses cases and examples of time series in various fields. I will identify

time series data from spreadsheet databases like the EUSES spreadsheet corpus[1], Enrons spreadsheets[2] and other related researches.

3. What are the current tools used for managing and analyzing time-series? What are their strengths and weaknesses?
   In a bid to grasp the requirements and essential functionalities for the web-based software being designed, I will study, evaluate and analyze the most popular and widely-used time series tools. These tools are evaluated with respect to the functionalities they provide for managing, transforming, analyzing and visualizing time series data.
4. What are the requirements for an end-user oriented application for managing, transforming and analyzing time-series?
   Essentially, this research aims to derive the functional requirements for an end-user oriented time series software. These requirements will form the basis for the later development of the time series software designed in this thesis. I will then identify specific use cases for the software and design mock-ups which will demonstrate how time series will be managed, transformed, analyzed and visualized by end-users.

1.4 Thesis Outline

In order to attain the above-stated goals and answer the research questions, this document is structured as thus.

The thesis contents are broadly grouped into two sections. In section one, I introduced the general theoretical concepts of time series and lay strong foundations for the remaining parts of this research. This section describes the distinctive features of time series and identified real world examples. Time series are classified into two broad categories: Univariate and Multivariate. They are further classified on the basis of their regularity and seasonality. This section also includes an overview of the basic objectives of time series analysis and forecasting.

The second section contains four chapters. This section reports on the core contributions of this project. Firstly, I reported on the frequency of occurrence of time series data in real world spreadsheets. I also made an analysis of time existing time series tools. The most widely-used time series tools are described and analyzed on the basis of their support for time series management, transformation, analysis and visualization. I then extrapolated the desired functionalities and requirements for the tools being designed in this thesis. These requirements are categorized based on their support for four main processes including: Managing Time Series, Transforming Time Series, Analyzing Time Series and Visualizing Time Series Data.

[1] EUSES Spreadsheet Corpus is a shared resource for supporting experimentation with spreadsheet dependability mechanisms. EUSES stands for End Users Shaping Effective Software

[2] Enrons Spreadsheet is a collection of over 15,000 spreadsheets used within the Enron Corporation.

In the penultimate chapter, I illustrated real world use cases of the user-oriented time series software. These use cases are illustrated using mockups which show how end users can easily interact with the software. These mockups express the major requirements defined for the software including both the structural aspects and the functional aspects of the time series software.

Conclusion and directions for future research are presented in the last chapter. The diagrammatic representation of the thesis structure is presented in Figure 1.1.

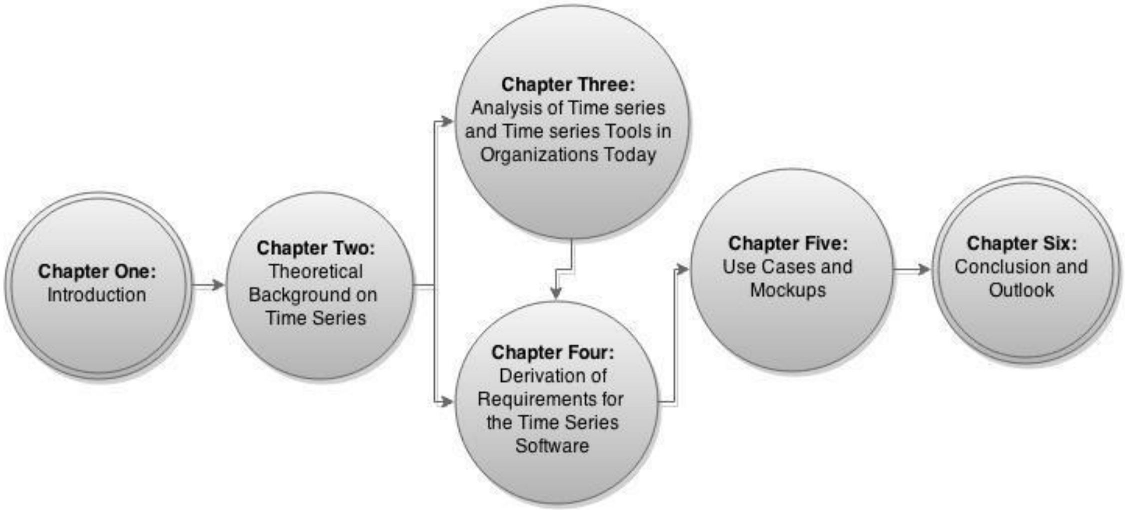

Chapter One: Introduction
Chapter Two: Theoretical Background on Time Series
Chapter Three: Analysis of Time series and Time series Tools in Organizations Today
Chapter Four: Derivation of Requirements for the Time Series Software
Chapter Five: Use Cases and Mockups
Chapter Six: Conclusion and Outlook

Figure 1.1: Diagrammatic representation of the thesis structure

2 Theoretical Background on Time Series

Today’s organizations in both the public and the private sectors are collecting large volume of data. Data is what businesses have been demanding for years in order to perform better analysis, make better decisions, and consequently become more competitive (Castillejos, 2006). Evidently, the possibility of such analysis and decision making depends most times on how data is collected and measured. Apparently, time scales provide the simplest and most sensible way of measuring collected data; this explains why time series are common in the world of business today.

Currently, time series underlie countless business activities. Businesses are thus often interested in analyzing and forecasting time series variables. Time series are used in statistics, signal processing, pattern recognition, econometrics, mathematical finance, weather forecasting, earthquake prediction, electroencephalography, control engineering, astronomy, communications engineering, and largely in any domain of applied science and engineering which involves temporal measurements (Hamilton, 1994).

This chapter presents the current state of research on time series. It covers discussion about the definitions of time series and how they differ from time-stamped data and other data types. This chapter also shows the different examples of time series and their properties and it describes the objectives for which time series analysis and forecasting are performed. In the last sections of the chapter, I described the different time scales for measuring time series data as well as the scenarios for which each scale can be used.

2.1 What is a Time series?

A time series is a sequence of data points, typically consisting of successive measurements or observations on quantifiable variable(s), made over a time interval (Cochrane, 2005). Usually the observations are chronological and taken at regular intervals (days, months, years), but the sampling could also be irregular.

Typical examples of time series also include historical data on sales, inventory, customer counts, interest rates, costs, etc. Time series data are also often seen naturally in many application areas including:

- Economics - e.g. monthly data for unemployment, hospital admissions, etc.
- Finance - e.g. daily exchange rate, share prices, etc.
- Environmental - e.g. daily rainfall, air quality readings.
- Medicine - e.g. ECG brain wave activity every 2^{-8} secs.

According to (Cochrane, 2005), time series can be represented as a set of observations X_T, each one being recorded at a specific time T; written as:

$$\{X_1, X_2, ...X_t\} \text{ or } \{X_T\}, \text{ where } T = 1, 2, ...t$$

If a time series has a regular pattern i.e. trend, then a value of the series should be a function of previous values. If X is the target value that is to be modelled and predicted, and X_t is the value of X at time t, then the goal is to create a model of the form:

$$X_t = f(X_{t-1}, X_{t-2}, X_{t-3}, \ldots, X_{t-n}) + e_t$$

Where X_{t-1} is the value of X for the previous observation, X_{t-2} is the value two observations ago, etc., and e_t represents noise that does not follow a predictable pattern (this is called a *random shock*). Values of variables occurring prior to the current observation are called *lag values*.

If a time series follows a repeating pattern, then the value of X_t is usually highly correlated with $X_{t-cycle}$ where $cycle$ is the number of observations in the regular cycle (DTREG, 2010). For example, monthly observations with an annual cycle often can be modeled by:

$$X_t = f(X_{t-12})$$

Generally, time series are not much different from the rest of econometrics. The major difference is that the variables are subscripted T rather than the normal-convention-usage i.

According to (Diggle, 1990), one simple method of describing a series is that of *classical decomposition*. Classical decomposition is of the notion that time series can be decomposed into four elements:

- Trend (T_t) — long term movements in the mean; Long term trend is typically modeled as a linear, quadratic or exponential function.
- Seasonal effects (I_t) — cyclical fluctuations related to the calendar;
- Cycles (C_t) — other cyclical fluctuations (such as a business cycles); an upturn or downturn not tied to seasonal variation. Usually results from changes in economic conditions.
- Residuals (E_t) — other random or systematic fluctuations.

The idea is to create separate models for these four elements and then combine them, either additively

$$X_t = T_t + I_t + C_t + E_t$$

or multiplicatively

$$X_t = T_t * I_t * C_t * E_t$$

2.2 Basic Concepts of Time Series

The review of the academic literatures shows that there is a vast literature on time series and their concepts. Time series can be categorized into two major classes namely: *univariate* or *multivariate*. *A univariate* time series is a sequence of measurements of the same variable collected over time. Most often, the measurements are sequence of events made at regular time intervals. An event is an ordered pair consisting of temporal value and an associated list of metadata (attributes) also known as header or general description (Dreyer, Kotz, & Schmidt, 1995). A typical univariate time series has the following format:

$$\{(t_1, \text{data-value}_1), (t_2, \text{data-value}_2), \ldots , (t_n, \text{data-value}_n)\}, \text{for } i= 1,2,\ldots,n$$

where data-value_i is the data value for the corresponding time t_i.

However, when a time series involves more than one variable, it is said to be multivariate. Most economic and financial information is structured in the form of multivariate time series. Multivariate time series can be further categorized into homogenous and heterogeneous multivariate time series, based on the relationships between the measured variables. *If a variable X is useful to predict future values of another variable Y, the multivariate time series is* *said to be homogeneous, else it is heterogeneous.* In homogenous multivariate time series, changes in one element in the observations vector of one variable imply corresponding changes in other variables that belong to the phenomenon under study.

Multivariate time series have the following general format:

{(t$_1$, < data-value$_{11}$, data-value$_{12}$, ... >), (t$_2$, < data-value$_{21}$, data-value$_{22}$, ... >), (t$_n$, < data-value$_{n1}$, data-value$_{n2}$, … >)}.

Time series can be further classified based on their regularity and seasonality. Classifying on the basis of regularity, time series can be: regular or irregular. These are defined by the duration between the timestamps of their elements. All regular time series have a predictable number of units between them, whereas irregular time series do not. For instance, an hourly reading from a thermometer produces a regular time series; while, a time series that records the time and amount of all ATM withdrawals from a bank account is an irregular time series (Castillejos, 2006).

When classified based on seasonality, a time series can be seasonal or non-seasonal. When a repetitive pattern is observed over some time horizon, the series is said to have seasonal behavior. Seasonal effects are usually associated with calendar or climatic changes. Seasonal variation is frequently tied to yearly cycles. Figure 2.1 shows a classification of time series.

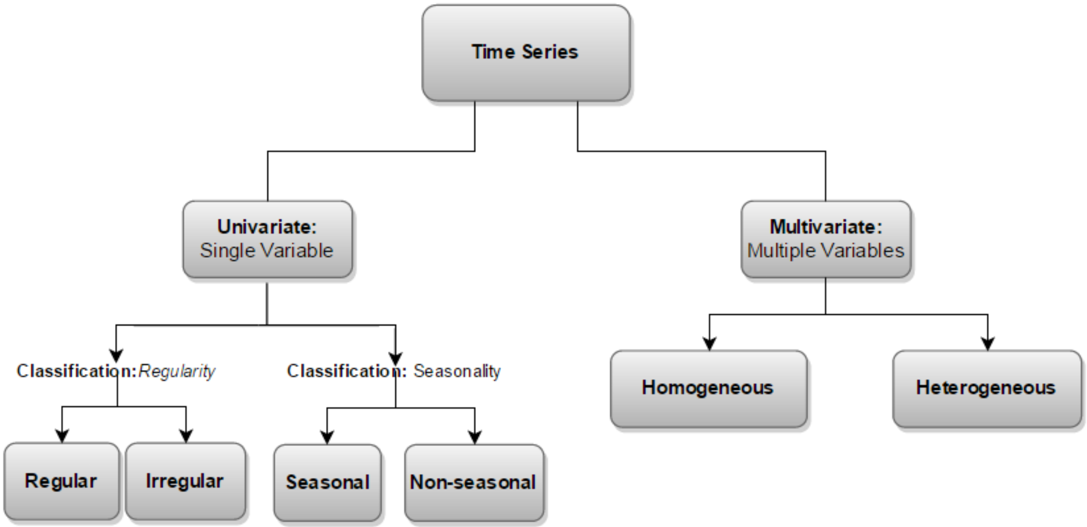


Figure 2.1: Classification of Time Series

The following is a brief definition of the commonly used terminologies in describing time series.

Stationary Data: This describes a time series variable which exhibits no significant upward or downward trend over time.

Non-stationary Data: A non-stationary time series data is a data with variable exhibiting a significant upward or downward trend over time.

Seasonal Data: This describes a time series variable exhibiting repeating patterns at regular intervals over time.

Time series analysis: Time series analysis is the process of using statistical techniques to model and explain a time-dependent series of data points. This involves methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data (Leonard & Wolfe, 2005). Time series analysis refers to problems in which observations which are collected at regular time intervals are correlated.

Time series forecasting: Time series forecasting is the process of using a model to generate predictions (forecasts) for future events based on known past events. The goal of forecasting is to project the underlying trend or pattern of the time series into the future as the most likely values for the data (Pentaho, 2013). Forecasting is the combining of knowledge from the past and future expectations with an estimated model to produce likely outcomes for the future. It enables more accurate predictions of the future to be made, reducing the uncertainty inherent in the decision-making process.

Regression analysis: Regression analysis is used essentially in such a way to test theories that the current values of one or more independent time series have some influences on the current value of another time series.

2.3 Time Series Properties

Time series have interesting features and properties that differentiate them from other data types. When compared to other data types, they behave in a different way. The following is a description of the most relevant properties of time series.

1. Time series data has a natural temporal ordering. They are generally written in a predefined order or some aggregated result based on the need of the user. This means access to data is usually alongside the time dimension, as, for instance, in the retrieval of all observations over a range of consecutive dates. Consequently, storage, retrieval, and update of time series data are not independent of each other (Castillejos, 2006). This differs from typical data mining/machine learning applications where each data point is an independent example of the concept to be learned.
2. The value of a time series in a time period is often affected by the values of variables in preceding periods, thus making the order in which the data occurs in the spreadsheet

very important. In essence, time series data are not commonly altered. A typical example of time series data is the daily reading of average wind speed. The data are recorded in intervals which are predefined and are normally not modified once they have been recorded. The data may be aggregated for further analysis but the aggregation is linked to a predefined granularity and sequential order.

3. Time series data are usually manipulated as one single object i.e. as a collection of data. This is because the order in which the data occurs in time series is very important because, unlike other data types, the ordering often represents the dependencies between the collected data. Thus, changing the order could change the meaning of the data. Consequently, manipulation of time series data puts more emphasis on aggregation operations on collections of data rather than on an individual data item (Lee & Elmasri, 1998).
4. Time series have a header i.e. a general description. The header contains all the metadata about the time series. Metadata can be information which allows the time series to be self-describing. The metadata can be information such as name, title, source, type of value, and type of time series. But more importantly is the date-time field which defines the dataset as a time series.
5. Data in time series are not necessarily identically distributed but they are dependent on preceding values. A stochastic model for a time series will generally reflect the fact that observations close together in time will be more closely related than observations further apart (Diggle, 1990).
6. In time series analysis, the past behavior of a variable is analyzed in order to predict its future behavior. By observing the different past states of a variable, the future states can be predicted through forecasting.

2.4 Examples of Time Series

In this section, real world examples of time series are illustrated based on the concepts explained and categorization of time series in section 2.2. Note however that the examples considered in this section are an extremely small sample from the multitude of time series encountered in various fields of engineering, science, sociology, economics etc. today. Our main purpose in this section is to describe the properties and types of time series using these real world examples.

2.4.1 Univariate Time Series with Trend

As defined in section 2.2, a univariate time series contains sequential measurements of a single variable. In this section, I illustrate a univariate time series with a uniform time distribution of measurements i.e. unvarying duration between the timestamps of the elements.

Figure 2.2 shows the population data of Nigeria in a 30-year interval from 1980 till 2010. The graph suggests the possibility of fitting a quadratic or exponential trend to the data. The quantitative time series data is univariate and data is finite.

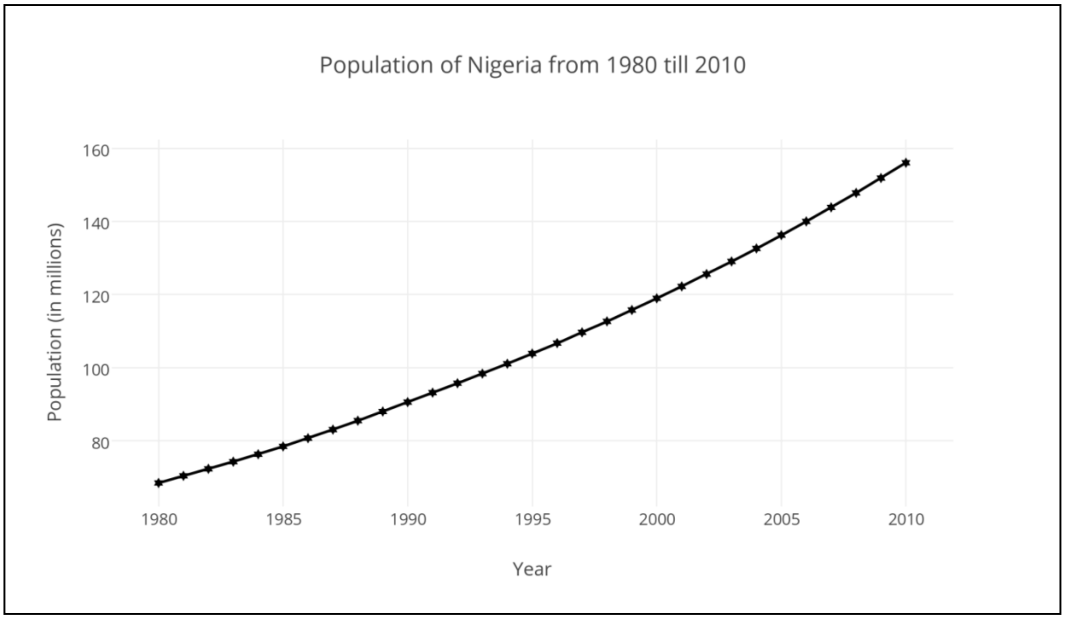


Figure 2.2: Univariate time series with trend
(population data of Nigeria from 1980 till 2010)

2.4.2 Univariate Time Series with Trend and Seasonality

Figure 2.3 illustrates a univariate time series data with a trend and a repetitive pattern in some time horizon i.e. seasonality. The times series also have a uniform time distribution. The data shows the monthly sales (in litres) of red wine by Australian winemakers from January 1980 till October 1991. In this case, the dataset consists of 142 observations, one for each month. Given a set of n observations made at uniformly spaced time intervals, it is often convenient to rescale the time axis in such a way that the set of time T_0 become the set of integers (1, 2, 3,…, n). In the present example this amounts to measuring time in months with January 1980 as month 1. Then T_0 is the set (1, 2, 3, …, 142). It appears from the graph that the sales have an upward trend and a seasonal pattern with a peak in July and a trough in January. The series is thus said to have seasonal behavior. This seasonal effect could be because of a major festival in Australia around July, or other social reasons. (Brockwell & Davis, 2002).

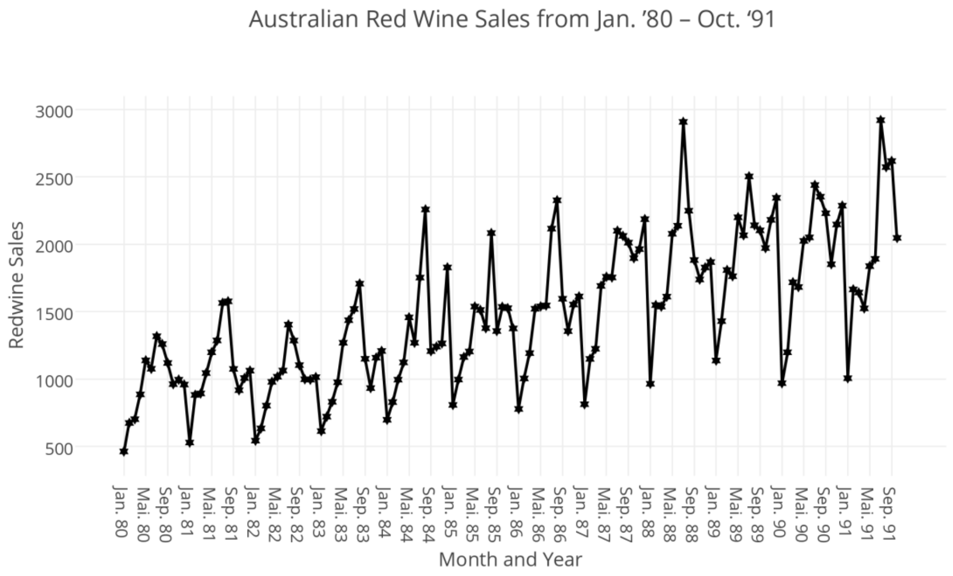


Figure 2.3: Univariate time series with trend and seasonality - uniform time distribution *(Australian red wine sales from January 1980 till October 1991)*

2.4.3 Univariate Time Series with no Trend and Seasonality

The annual strikes per year in the USA for the years 1951-1980 are shown in Figure 2.4. They appear to fluctuate erratically about a slowly changing level. There is no clear trend in the plot and no identifiable seasonality.

Figure 2.4: Univariate time series with no trend and seasonality *Number of Strikes per year in the USA 1951 - 1980*

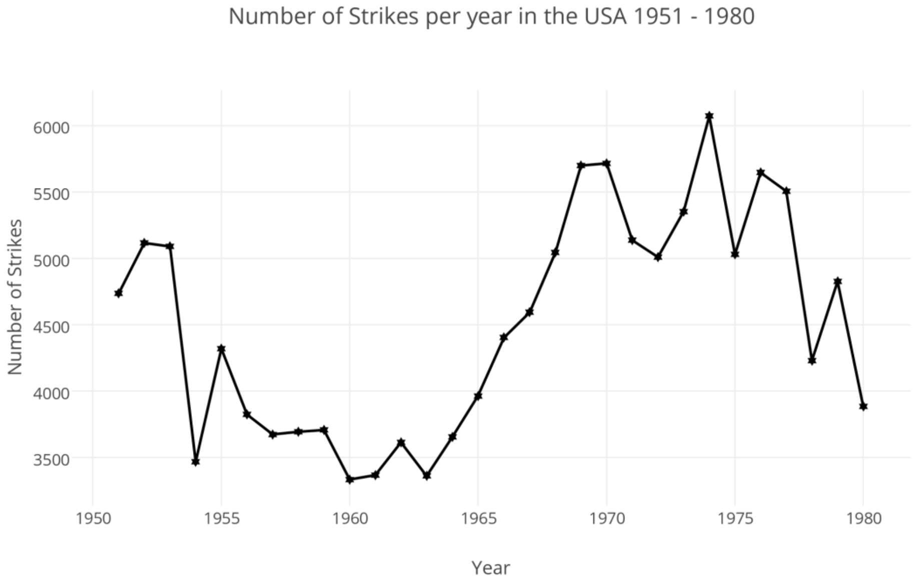


Figure 2.4: Univariate time series with no trend and seasonality Number of Strikes per year in the USA 1951 - 1980

2.4.4 Multivariate Time Series

A multivariate time series is defined as a sequence of measurements of more than one dependent or independent variables, collected over time. An example of an homogenous multivariate time series is illustrated here from (NIST/SEMATECH, 2012).

The data presented in figure 2.5 represents data from a gas furnace used for the production of CO_2 (carbon dioxide). Inside the gas furnace, air and methane were combined in order to obtain a mixture of gases containing CO_2 (carbon dioxide). The input series X_t is the **methane gas feed rate** and **the CO_2 concentration** is the output series Y_t. In this experiment 296 successive pairs of observations (X_t,Y_t) were collected from continuous records at 9-second intervals.

Note that there is a direct relationship between the feed rate of methane gas (i.e. X_t) and the concentration of CO_2 in the output gas (i.e. Y_t). From comparing the plots in figure 2.5, an inverse proportionality of the methane gas feed rate and the CO2 concentration is observed. This data shows a homogeneous multivariate time series with uniform time distribution. The plots of the input and output series are displayed below.

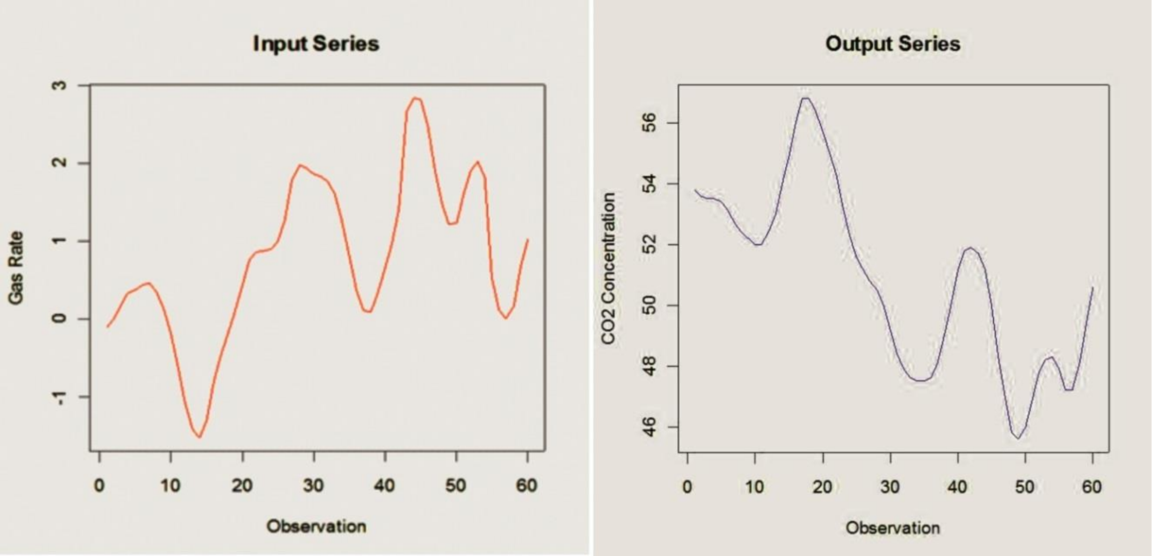


Figure 2.5: Multivariate homogenous time series (NIST/SEMATECH, 2012) Input and Output series of the CO2 gas furnace

2.5 Time Series Analysis

Time series data occurrences are becoming extremely valuable to the operations and development of modern organizations. Financial institutions, for example, rely on analysis of time series for forecasting economic conditions, developing and using complicated financial decision support models, and conducting international financial transactions. Likewise, public and private institutions are using time series data to manage and project the loads on their networks. More and more time series are used in this type of investigation and hundreds of thousands of time series that contain valuable economic and financial information are nowadays available both on and off-line (Castillejos, 2006). Thus, an understanding of the standard practices for time series analysis is appropriate for the reader. This section expatiates on the processes, practices and objectives of time series analysis. It also describes the uses and relevance of time series models as well as categorizes them.

Time series analysis accounts for the fact that data points taken over time may have an internal structure (such as autocorrelation, trend or seasonal variation) that should be accounted for. As defined earlier, time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. It involves the use of techniques for drawing inferences from time series data. Note however that one other main purpose for analyzing time series is forecasting. Forecasting is the application of a model to predict future values based on previously observed time series values.

In order to perform time series analysis, it is essential to set up hypothetical probability representation of the data. Such representation is called the Model. After the model has been appropriately determined, it is then possible to estimate parameters, check for goodness of fit to the data, and possibly to use the fitted model to enhance the understanding of the mechanism generating the series (Brockwell & Davis, 2002). Once a satisfactory model has been developed, it may be used in a variety of ways depending on the particular field of application.

The model may be used simply to provide a **compact description of the data**. One simple method of describing a series is that of *classical decomposition* discussed in section 2.1. I may, for example, be able to represent the Australian red wine sales data of section 2.4.2 as the sum of a specified trend, and seasonal and random terms.

2.5.1 Basic Objectives of Time series Analysis

The major approach to time series analysis is usually to determine a model that describes the pattern of the time series. Uses for such a model are:

- To describe the important features of the time series pattern.
- To explain how the past affects the future or how two time series can "interact".
- To forecast future values of the series.
- To possibly serve as a control standard for a variable that measures the quality of product in some manufacturing situations.

The goal of building a time series model is the same as the goal for other types of predictive models which is to create a model such that the error between the predicted value of the target variable and the actual value is as small as possible. The primary difference between time series models and other types of models is that lag values of the target variable are used as predictor variables, whereas traditional models use other variables as predictors, and the concept of a lag value doesn't apply because the observations don't represent a chronological sequence (Robert H. & David S., 2010).

Thus, the aim of time series analysis is to describe and summarize time series data, determine most suitable models, and make forecasts.

2.5.2 Time series Models

Time series models are used to describe the underlying data-generating process of a time series. The usual process to time series analysis and forecasting is:

1. Preprocess data for analysis. The preprocessing stage may involve some initial analysis steps e.g. plotting the data, determining time series characteristics such as trends, seasonality etc.
2. Determine suitable model for the preprocessed time series
3. Apply/fit model to time series data. Additionally, after fitting the time series model to the time series data, the fitted model can be used to determine departures (outliers) from the (assumed) data-generating process or forecast function components (future trend, seasonal or cycle estimates).
4. Perform forecasting and prediction of future values of time series.

In essence, time series models are used for predicting or forecasting the future behavior of variables (Brockwell & Davis, 2002).

According to (Leonard & Wolfe, 2005), other applications of time series models include separation (or filtering) of noise from signals, testing hypotheses such as global warming using recorded temperature data, predicting one series from observations of another. Time series models are most especially used to forecast time series. These forecasts can be used to predict future observations as well as to monitor more recent observations for anomalies using holdout sample analysis.

Time series models are also useful in simulation studies. For example, the performance of a reservoir depends heavily on the random daily inputs of water to the system. If these are modeled as a time series, then the fitted model can be used to simulate a large number of independent sequences of daily inputs. Knowing the size and mode of operation of the reservoir, the fraction of the simulated input sequences that cause the reservoir to run out of water in a given time period can be determined (Brockwell & Davis, 2002). This fraction will then be an estimate of the probability of emptiness of the reservoir at some time in the given period.

Models for time series data can have many forms and represent different stochastic processes. There are three broad classes of the models: the autoregressive (AR) models, the integrated (I) models, and the moving average (MA) models. These three classes depend linearly on previous data points. Combinations of these ideas produce autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) models. The autoregressive fractionally integrated moving average (ARFIMA) model generalizes the former three. Extensions of these classes to deal with vector-valued data are available under the heading of multivariate time series models and sometimes the preceding acronyms are extended by including an initial "V" for "vector", as in VAR for vector auto-regression. An additional set of extensions of these models is available for use where the observed time series is driven by some "forcing" time series (which may not have a causal effect on the observed series): the distinction from the multivariate case is that the forcing series may be deterministic or under the experimenter's control. For these models, the acronyms are extended with a final "X" for "exogenous" (Hamilton, 1994).

2.5.3 Classification of Time series Models

Time series models are generally classified based on their suitability for analyzing different types of time series data. Thus, models which are considered suitable for stationary time series are referred to as stationary time series models. This thesis classifies time series under the following heading (based on the prevalent classification in many literatures on time series):

- Stationary univariate time series models
- Non-stationary univariate time series models
- Stationary multivariate time series models
- Non-stationary multivariate time series models
- Structural Change and Nonlinear Models

Table 2.1 shows the categorization of 27 time series models. These models are not fully described in this thesis, as this is out of the scope of this work. Reader may refer to (Zivot, 2006) and (Cochrane, 2005) for more detailed description of these models.

Table 2.1: Categorization of Time Series Models

	Univariate Models	**Multivariate Models**	**Structural Change and Non-linear Models**
Stationary	Wold decomposition theorem	Dynamic simultaneous equations models	Tests for structural change with unknown change point
	Difference equations	Vector auto-regression (VAR) models	Estimation of linear models with structural change
	ARMA models	Granger causality	Regime switching models
	Box-Jenkins methodology	Impulse response functions	
	Model Selection	Variance decompositions	
	Forecasting methodology	Structural VAR models	
Non-stationary	Trend/Cycle decomposition	Spurious regression	
	Beveridge-Nelson decomposition	Co-integration	
	Deterministic and stochastic trend models	Granger representation theorem	
	Unit root tests	Vector error correction models (VECMs)	
	Stationarity tests	Structural VAR models with co-integration	
		Testing for co-integration	
		Estimating the co-integrating rank	
		Estimating co-integrating vectors	

2.5.4 A General Approach to Time series Modelling

In this section, a review of the general approaches to time series modelling is made. This also includes some important underlying characteristics to consider for time series modelling based on (Brockwell & Davis, 2002). Below is a step-wise overview of the way in which time series modelling can be carried out.

- Plot the time series and examine the main features of the graph, checking in particular whether there is:
  a. A trend, i.e. on average, do the measurements tend to increase (or decrease) over time?
  b. A seasonal component, i.e. is there is a regularly repeating pattern of highs and lows related to calendar time such as seasons, quarters, months, days of the week, and so on?
  c. Any apparent sharp changes in behavior,
  d. Any outlying observations. In regression, outliers are far away from the fitted line. With time series data, outliers are far away from the other data
  e. A long-run cycle or period unrelated to seasonality factors
  f. A constant variance overtime, or whether the variance is non-constant.
- Remove the trend and seasonal components to get *stationary* residuals (as defined in section 2.2). To achieve this goal it may sometimes be necessary to apply a preliminary transformation to the data. For example, if the magnitude of the fluctuations appears to grow roughly linearly with the level of the series, then the series will have fluctuations of more constant magnitude. There are several ways in which trend and seasonality can be removed, some involving estimating the components and subtracting them from the data, and others depending on differencing the data, i.e., replacing the original series $\{X_t\}$ by $\{Y_t := X_t - X_{t-d}\}$ for some positive integer *d*. Whichever method is used, the aim is to produce a stationary series, whose values shall be referred to as residuals.
- Choose a model to fit the residuals, making use of various sample statistics including the sample autocorrelation function briefly mentioned earlier in this section.
- Forecasting will be achieved by forecasting the residuals and then inverting the transformations described above to arrive at forecasts of the original series $\{X_t\}$.

It is often not a direct process to choose the model to fit a time series and analyze it, since there are usually many time series analysis and forecasting techniques, each with different characteristics. It is thus usually difficult to know which technique will be best for a particular data set. It is customary to try out several different techniques and select the one that seems to work best. To be an effective time series modeler, one needs to keep several time series techniques in one’s “tool box.” In the next section, I described a technique used in the selection of best-fit models for time series analysis.

2.5.4.1 Automatic Model Selection

Model selection is an important part of Time series analysis. However, the model that best fits a time series (for analysis and prediction operations) is often not easily determined. This problem is further complicated by the fact that there may be many time series, and no one model explains the data-generating process for all series. Automatic model selection is a technique that selects an appropriate time series model for a given time series.

There are many methods available for implementing automatic model selection. Below is a list of some of the methods.

- Model Selection Criteria
- Best Subsets Procedures
- Forward Inclusion
- Backward Deletion
- Forward, Backward Stepwise Model Selection (Stepwise Regression)

In this thesis, the model selection criteria method is used because it involves a wider search and it compares models in a preferable manner. For each time series, a list of candidate models can be chosen based on the time series characteristics (e.g., number of variables, seasonality, trend, etc.), and an appropriate time series model can be selected from the list of candidate models using the model selection criteria. The interest is in the model which best fits to the time series in terms of operations such as analysis and prediction.

Automatic model selection can reduce a single time series to a model specification. The selected model specification can then be fitted to the data. One way of applying model selection criteria is to score a single or multiple time series on a set of relevant criteria and then match the resulting score to all candidate models.

Table 2.2 below illustrates an example of automatic model selection for some time series (A, B, C, D and E).

Table 2.2: Automatic Model Selection

Series	Level Parameter	Trend Parameter	Season Parameter	Cycle Parameter	Model Specification Criteria
A	0.20	-	-	0.44	Cyclic
B	0.21	-	0.17	-	Seasonal
C	0.40	0.70	-	-	Trend
D	0.10	0.35	0.60	-	Seasonality with Trend
E	0.40	-	-	-	Level

In the example above, each time series are scored based on their characteristics. Then a final description is given for each series which is used as criteria for selecting the model that best fits for analysis and forecasting. For instance, one would consider best time series trend and seasonality model for series C and D.

2.6 Quantification of Qualitative Variables

Time series data are not always collected quantitatively, but also qualitatively. Usually, the approach to qualitative data collection and analysis is said to be methodical and allows for greater flexibility than in quantitative data collection (Abeyasekera, 2005). In qualitative data collection, data is collected in textual form on the basis of observation and interaction with the participants e.g. through participant observation, in-depth interviews and focus groups. However, it is important to have a numeric representation of the qualitative data in order to quantitatively describe and explain the event that those observations reflect. The quantification of qualitative time series data often precedes analysis and forecasting of the time series.

The task of representing time series data as discrete numerical figures is often times not straightforward as variables/information collected may be mainly qualitative. This usually depends on the project scenario, as it is sometimes easier to assign qualities, rather than quantities, to some variables of interest.

For example, suppose it is of interest to learn about people's perceptions of what poverty means for them - measured at different periods of the year. It is likely that the narratives that result from discussions across several communities will show some frequently occurring answers like experiencing periods of food shortage, being unable to provide children with a reasonable level of education, not owning a radio, etc. Such information can be extracted from the narratives and coded using some quantification approaches. These quantification approaches provide methods to represent qualitative information with discrete data. These can then be discussed more easily, unhindered by possible qualitative ambiguity.

There are many approaches for quantifying qualitative variables. However, the most popular approaches are in forms of ranking and scoring the variables using the measurement scales.
The four scales of measurements commonly used in statistical analysis are nominal, ordinal, interval, and ratio scales. These scales are briefly discussed below.

- **Nominal Scale**: This scale is based on a set of qualitative attributes. There is no criterion to order the items of a nominally scaled variable. Only the direct comparison ("is equal" and "is not equal") is possible and allowed. Numeric values are assigned to non-numeric variables. Examples include sex of a person, colour, trademark, species etc. However, the numeric values are not true numbers.
- **Ordinal Scale**: This scale refers to measurements that can be ordered in terms of "greater", "less" or "equal". Observations do not need to be equidistant. That is, ordinal scale measures variables where the order matters but the differences do not matter. The ordinal scale is therefore sometimes also called rank scale. Examples are percentile ranks, grades at school, ranks in a race etc.

  A typical description is in the case of letter grades. We don't really know how much better an A is than a D. We know that A is better than B, which is better than C, and so on. But is A four times better than D? Is it two times better? In this case, the order is important but not the differences. Other examples of variables measured on an ordinal scale include:
    - A question on 'How was your experience today?' rated on a scale of 1-10.
    - Job difficulty measure with options: hard, medium, easy
    - Order of finishing a race: first place, second place, and so on.
- **Interval scale:** It measures variables where the differences between the numbers do matter. Interval scales place objects in order and equal differences in value *which* denote equal differences in what is being measured. Examples include temperature (in C, F, or R), water level of a river. Interval scaled data can be transformed by a linear transformation of the type y = x + d without losing their characters (thus interval scales remain interval scales, under certain circumstances even a ratio scale may be achieved) (Abeyasekera, 2005).
- **Ratio Scale:** Ratio scale measures equally spaced units along the scale with a true zero point *and you can divide values*. Examples are temperature in K, weight, driving speed. Ratio scaled data can be transformed by the linear transformation $y = kx + d$

without losing their character. This transformation is applied, for example, when converting meters to inches. Similar interval measurement but also has a ‘true zero point’.

Many other procedures are available for dealing with qualitative information that can be coded either as binary variables, i.e. Yes/No, presence/absence type data, or as categorical variables, e.g. high, medium, low *access to regional facilities*, decreasing/static/increasing *dependence on forest resources*. If factors affecting qualitative features of the binary sort are to be explored, **logistic regression modelling** can be used (Abeyasekera, 2005).

2.7 Time-stamped data vs Time Series

Time-stamped data, also called transactional data, are data with date-and-time-stamps collected over time at no particular frequency. Time-stamped data are a common data type in the age of *Big Data*. The fact that computers can record all the actions a user takes means that a single user can generate thousands of data points alone in a day. For example, when people (users) visit a website or use an app, or interact with computers (or phones and other devices), their actions can be logged, and the exact time of their action recorded (Schutt, 2012). Below is an example of a time stamped data – a typical log data from a computer.

```
[Sun Mar  7 16:02:00 2004] [notice]
[Sun Mar  7 16:02:00 2004] [info] Server built: Feb 27 2004
13:56:37
[Sun Mar  7 16:02:00 2004] [notice] Accept mutex: sysvsem
(Default: sysvsem)
[Sun Mar  7 23:42:44 2004] [notice] [64.242.88.10]
[Mon Mar  8 00:11:22 2004] [info] [64.242.88.10] (104)
[Mon Mar  8 00:32:45 2004] [info] [64.242.88.10] (104)
[Mon Mar  8 00:40:10 2004] [info] [64.242.88.10] (104)
[Mon Mar  8 01:04:05 2004] [info] [64.242.88.10] (104)
[Mon Mar  8 08:14:15 2004] [info] [64.242.88.10] (104)
[Mon Mar  8 14:54:56 2004] [info] [64.242.88.10] (104)
[Tue Mar  9 13:49:05 2004] [info] [81.226.63.194]
[Tue Mar  9 08:15:21 2004] [info] [64.242.88.10] (104)
[Tue Mar  9 09:36:35 2004] [info] [64.242.88.10] (104)
[Tue Mar  9 13:36:06 2004] [info] [64.242.88.10] (104)
[Wed Mar 10 11:45:51 2004] [info] [24.71.236.129] (104)
[Wed Mar 10 18:52:30 2004] [info] [64.242.88.10] (104)
[Wed Mar 10 18:58:52 2004] [info] [64.242.88.10] (104)
[Thu Mar 11 20:04:35 2004] [info] [64.242.88.10] (104)
```

```
[Thu Mar 11 22:08:43 2004] [info] [64.242.88.10] (104)
[Thu Mar 11 22:09:44 2004] [info] [64.242.88.10] (104)
```

From the computer log data, we can identify the following parts:

- The time-stamp: This is the main field that defines any dataset as a time-stamped data e.g. `[Wed Mar 10 11:45:51 2004]`
- The value which includes:
  - Description e.g. `[notice]`
  - Client IP address e.g. `[64.242.88.10]`
  - Reference number: e.g. `(104)`

Other examples of time-stamped data include:

- Internet data
- Point of Sales (POS) data
- Inventory data
- Call Center data
- Trading data etc.

Businesses often want to analyze time-stamped data for trends and seasonal variation. Analyzing these time-stamped data can help business leaders make better decisions by listening to their suppliers or customers via their transactions collected over time. A business can have many suppliers and/or customers and may have a set of transactions associated with each one. However, the size of each set of transactions may be quite large, making it difficult to perform many traditional data-analysis tasks. Trend and seasonal statistical analysis of time-stamped data can help reduce the information contained in a single set of transactions to a small set of statistics. However, in order to analyze time-stamped data for trends and seasonality, statistics must be computed for each time period and season of concern (Leonard & Wolfe, 2005). The time-stamped data is then re-organized based on the computed time-period or frequency. This frequency may vary with the business problem. The output dataset is called time series.

Thus, time series data can be described as time-stamped data collected over time at a particular frequency. The frequency associated with the time series varies with the problem at hand. The frequency or time interval may be hourly, daily, weekly, monthly, quarterly, yearly, or many other variants of the basic time intervals. The choice of frequency is an important decision for data scientists. Consider two examples:

a. ATM daily withdrawals from a banking institution. Each transaction may be associated with the date and time (point in time) when the transactions are made. This can be represented with a daily time series i.e. frequency is daily.
b. The total monthly sale of a product during a year, say 2004, may be represented with a monthly time series. Each monthly sale can be associated with a specific month April, 2004 and so on.

Notice that even though both examples can be represented as time series, they have different time-granularities i.e. the first represents a daily time series and the second represents a monthly time series.

Some time series data may have an associated seasonal cycle or seasonality. For example, the length of seasonality for a monthly time series is usually assumed to be 12 because there are 12 months in a year. Likewise, the seasonality of a daily time series is usually assumed to be 7. The usual seasonality assumption may not always hold. For example, if a particular business's seasonal cycle is 14 days long, the seasonality is 14, not 7 (Leonard & Wolfe, 2005).

Time series are based on specific time granularity or frequency; consequently, in a time series application, it is essential to take into account the different time granularities (e.g. daily, weekly, quarterly, annual, etc.). Calendar units, such as months and days, clock units, such as hours and seconds, and specialized units, such as business days, tax reporting, and academic years, are typically used in defining the granularities for time series data (Castillejos, 2006).

The process of converting time-stamped data to time series is described using an example of a raw time-stamped data of an online shop by (Leonard & Wolfe, 2005). Firstly, the time-stamped data is plotted on a scatter plot, shown in Figure 2.6 (showing only the first year). The data is then grouped based on a monthly frequency. Figure 2.7 illustrates the "binding" of the time-stamped data in monthly intervals. Computations of the seasonal statistics (totals) for each month were then made in order to generate the time series. Figure 2.8 shows the totals for each month and Figure 2.9 shows the trend statistics (totals) for a single month (e.g. March).

It is important to note that there are standards methods used for the transformation of time-stamped data to time series. This thesis has used the MapReduce technique to reduce time-stamped data, associate frequencies to the data and convert the data to time series. MapReduce is generally defined as a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key (Dean & Ghemawat, 2010). A more detailed description of the MapReduce technique is made in the next section.

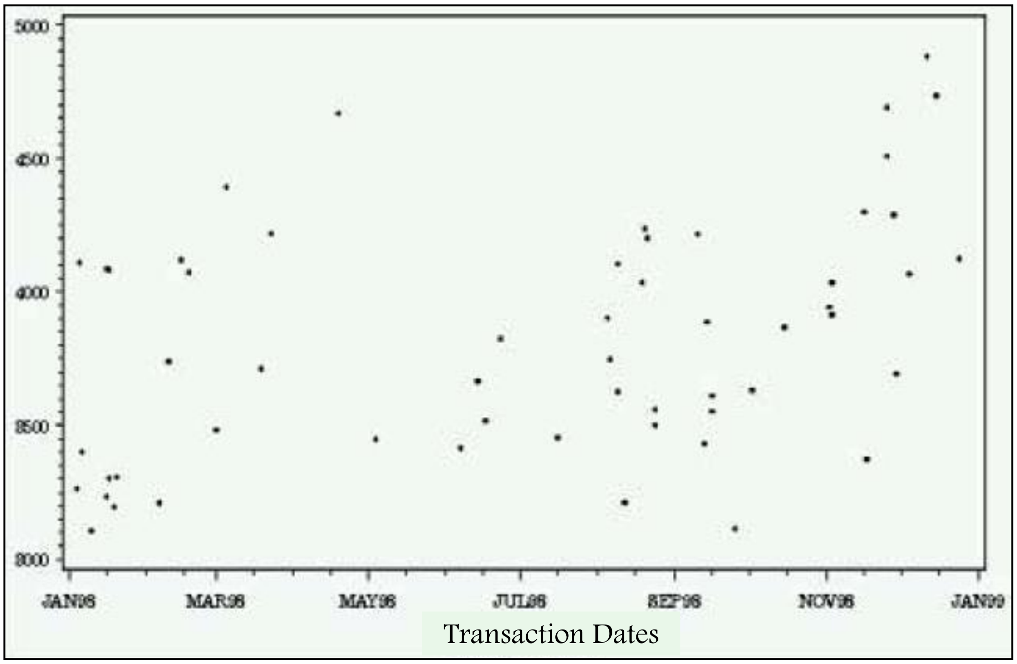


Figure 2.6: Historical Time-Stamped Data of an Online Shop (showing only first year)

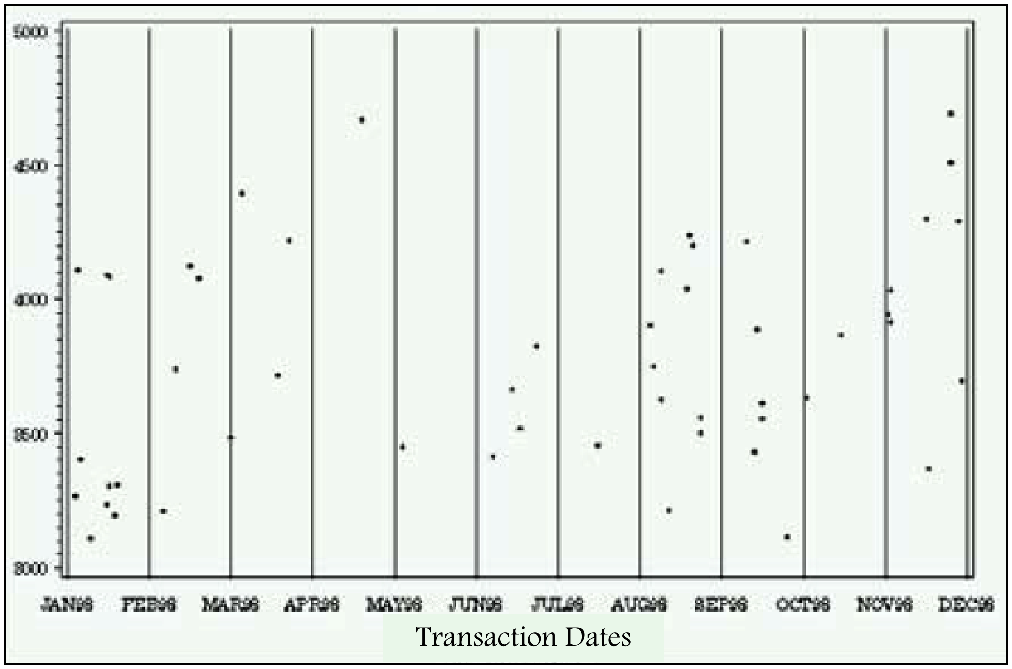


Figure 2.7: Time-Stamped Data analyzed on a Monthly Basis

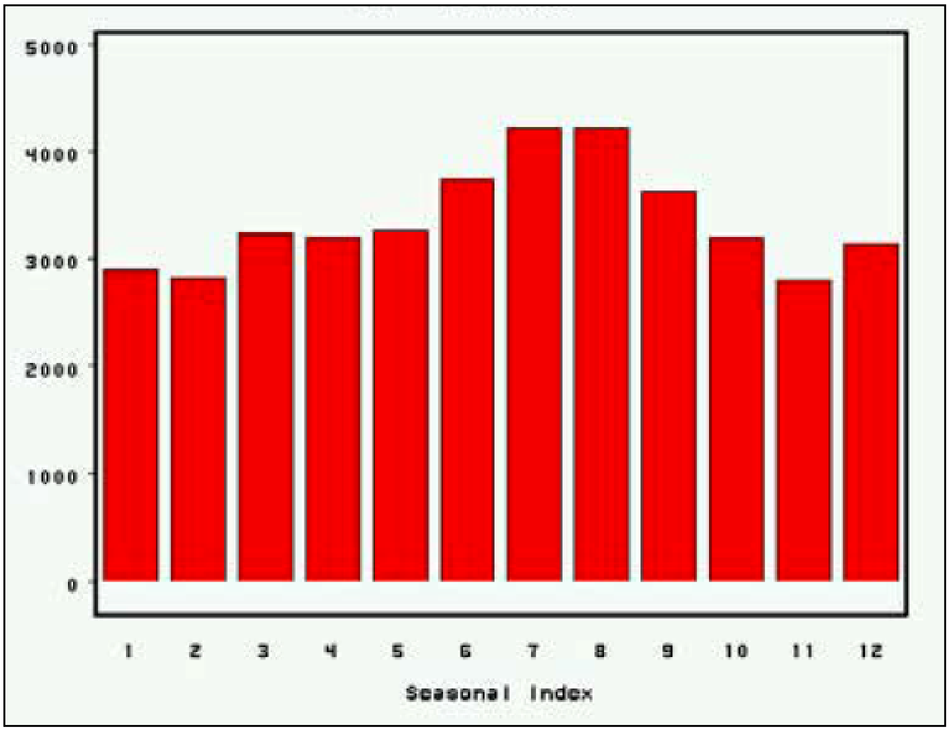


Figure 2.8: Seasonal Totals (12 Seasons)

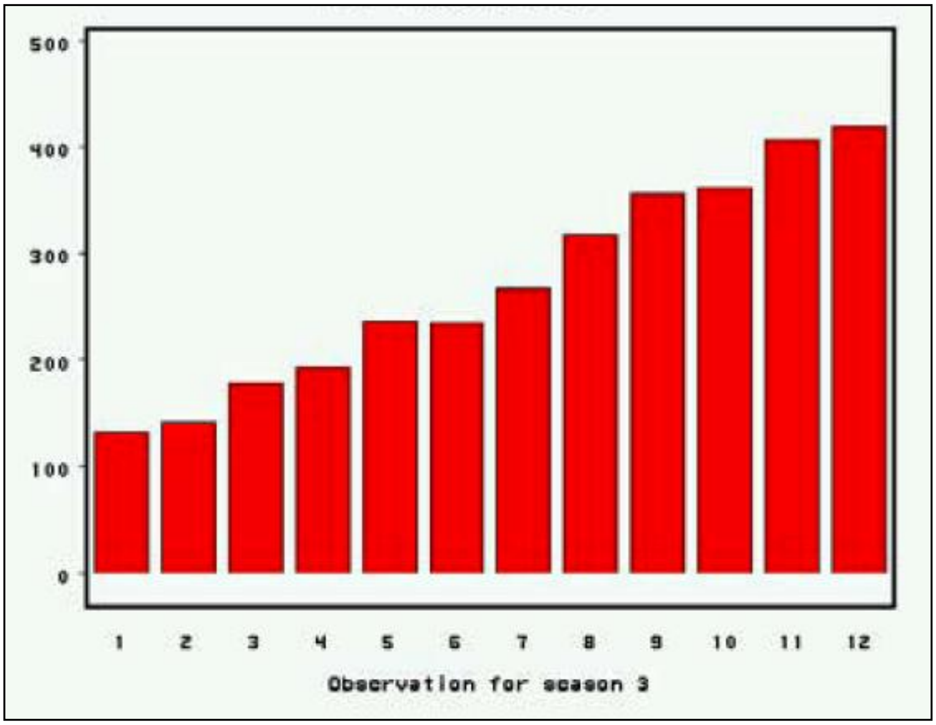


Figure 2.9: One Season's Totals (Season 3 of 12)

2.7.1 Transforming Time-stamped data to Time Series using MapReduce

MapReduce is a programming model for processing and generating large data sets. MapReduce was created by Google based on the parallel processing programming logic, written in Java. The Map Reduce programming model works on two parts – The Mapping part (done by the Mapper) and The Reduction part (done by the Reducer) (Dean & Ghemawat, 2010).

The Mapper works on the blocks of data available in the data nodes and tries to get the job done. One can think of Mapper as an individual worker (in the master-slave concept), working to get the data required from the client. Now the major task remains is to get the aggregate count of the results done by each Mapper. This work is done by the Reducer. The Reducer iterates over the entire result data and sends back a single output value. MapReduce allows programmers with no experience with parallel and distributed systems to easily utilize the resources of a large distributed system. A typical MapReduce computation processes many terabytes of data on hundreds or thousands of machines. Programmers find the system easy to use, and more than 100,000 MapReduce jobs are executed on Google's clusters every day (Shrivastava, 2012).

In the context of this thesis, MapReduce is applied to quantify qualitative variables, to convert time-stamped data to time series and to transform time series from one time frequency to another (e.g. transforming hourly time series to daily time series). The map function defines the mapping of a qualitative variable (or time-stamped data) to a numerical value, wherefore we get a quantitative variable; and the reduce function aggregates the quantitative variables accordingly. A detailed description of the exact application of MapReduce in this thesis is reported in chapters four and five.

The general flow of the MapReduce is shown in figure 2.10. The figure illustrates the overall process of the technique from the input phase to the output phase.

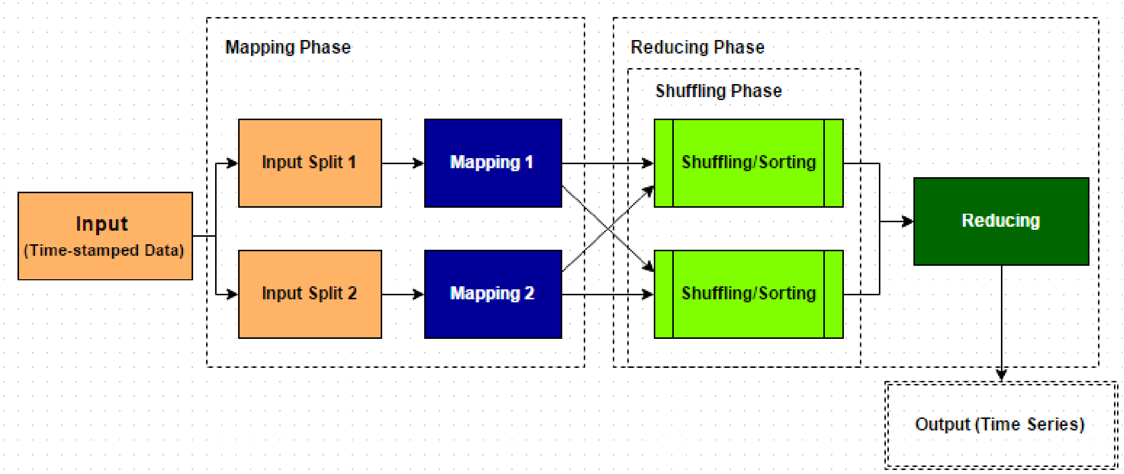


Figure 2.10: General flow of MapReduce Technique.

To help illustrate the MapReduce programming model, consider the problem of counting the number of occurrences of each word in a large collection of documents. From the illustration in fig 2.11, one can see that the user give something as the input. In this case the input is a question and its subsequent answer. The Map-Reduce program looks into given data and breaks the data into an intermediate stage. The intermediate stage consists of a key/value pair, which breaks the file data into many key-value pair data. The map function emits each word i.e. Key, plus an associated count of occurrences i.e. Value (e.g. '1' in this example).

The reduce function sums together all counts emitted for a particular word. MapReduce automatically parallelizes and executes the process on a large cluster of commodity machines. The runtime system takes care of the details of partitioning the input data, scheduling the process execution across a set of machines, handling machine failures, and managing required inter-machine communication (Shrivastava, 2012).

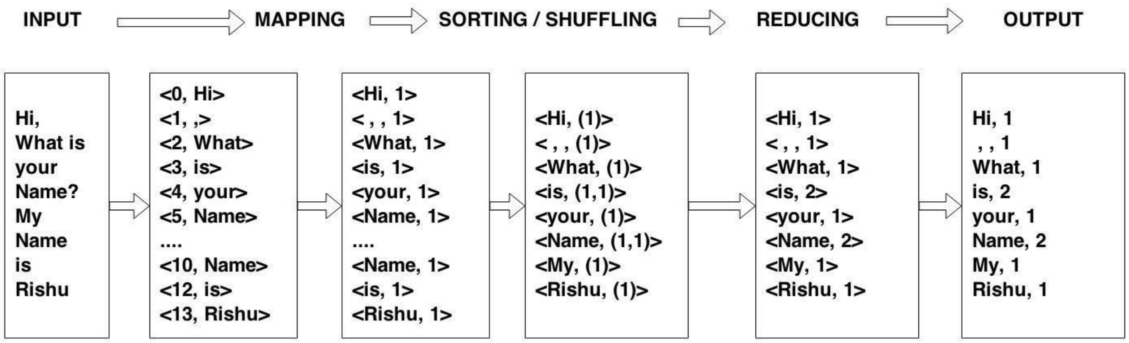


Figure 2.11: Illustrating the MapReduce Technique (source: (Shrivastava, 2012))

Part 2: Contributions of the Thesis

3 Analysis of Time series and Time series Tools in Organizations Today

In order to answer the research question on how common time-series data are in spreadsheets today, this chapter reports on the frequency of occurrence of time series data in real world spreadsheets - both in the private and public organizations. I also made an analysis of time existing time series tools in a later section. The most widely-used time series tools are described and analyzed on the basis of their support for time series management, transformation, analysis and visualization. This analysis will form the basis for extrapolating the desired functionalities and requirements for the thesis time series tools.

3.1 Time Series in Real-World Spreadsheets

Time series data are generally documented in spreadsheets. Thus, the datasets used as time series examples in this section are sourced from the spreadsheet databases: EUSES Corpus (Fisher & Rothermel, 2005) and the Enrons Spreadsheets (Hermans & Murphy-Hill, 2014) (refer to section 1.2 for definitions of these spreadsheet databases). There has been a significant amount of work on spreadsheets whose researches are based on the EUSES corpus: a set of 4,498 spreadsheets published in 2005. However, researchers have reported that most EUSES spreadsheets were obtained through the public world-wide-web and from textbook examples (Hermans & Murphy-Hill, 2014). The EUSES Corpus is thus missing a substantial set of closed source spreadsheets, that is, spreadsheets that were not intended to be made available to the public. According to (Hermans & Murphy-Hill, 2014), one of the reasons that EUSES is so commonly used, is that it is the best there is, there is no other corpus of similar size until the Enrons spreadsheets became existent. Some researchers have tried to get access to spreadsheets from industry, but companies are reluctant to share them. Firstly, the contents of the spreadsheets might hold confidential information, such as pricing models, which companies want to keep out of the hands of competitors or customers. Secondly, organizations are afraid detailed studies of their spreadsheets might reveal errors.

The identified setbacks of the EUSES Corpus do not affect the examples used in this thesis since the datasets are chosen from the combination of EUSES Corpus and the Enrons Spreadsheets. The Enrons Spreadsheets corpus is a newer spreadsheet corpus obtained from the industries. It is an industrial dataset of over 15,000 spreadsheets. It differs from the EUSES corpus in a number of ways:

- Although EUSES is a large spreadsheet corpus, it is relatively small by modern software repository standards; EUSES has about 4.5 thousand spreadsheets, while Sourceforge lists 350 thousand software projects[3] and OpenHub lists about 666 thousand software projects[4].
- The EUSES spreadsheets are open source spreadsheets and the spreadsheets were obtained from the internet and they lack a substantial set of closed source spreadsheets.
- The EUSES corpus is not publicly available. To use it, "you must be a researcher in the field of software engineering, end-user programming, human-computer interaction, or usability"[5], and even then, the researcher must explicitly ask for a copy by email. (Hermans & Murphy-Hill, 2014)

As part of the contributions of this thesis, I researched the frequency of occurrence of time series data in the public (i.e. EUSES Corpus) and private spreadsheets (i.e. Enrons spreadsheets). Out of a total of 20,000 spreadsheets, 5,000 spreadsheets were analyzed manually. The spreadsheets were checked for time series data through physical observation. The spreadsheets were categorized into 'Time Series' and 'Others' depending on whether it contains time series data in any of its sheets. This step was important and prerequisite to selecting the time series examples used in this section.

The criteria for detecting a time series data is not exactly direct as real world spreadsheets contain data which are formatted in different ways. Data are entered into spreadsheet in a way that suits the creator's needs. For instance, a date field may take different forms such as:

- Proper date format e.g. 15-Mar-2015
- Date and time e.g. 15-Mar-2015 12:00:00
- Criteria for identifying Spreadsheets
- An incomplete date e.g. only year, month, day etc.
- Seasonal divisions e.g. Summer, Winter, Autumn and Spring
- Other time or date divisions:
    - Week numbers e.g. Week 1 to Week 52
    - Month numbers e.g. Month 1 to Month 12
    - Quarter Numbers e.g. Quarter 1, Quarter 2 etc.
- Date grouping e.g. 15-Mar-2013 to 17-May-2014.

All these variations are taken into consideration while detecting time series in the spreadsheets.

[3] http://sourceforge.net/blog/sourceforge-myths/

[4] https://www.openhub.net/explore/projects

[5] http://eusesconsortium.org/resources.php

Despite these many variations, the underlying factor that defines a datasets as time series is the availability successive measurements or observations made over some time-based intervals.

The frequency of occurrence of time series data in spreadsheets are illustrated in the figures below. As shown in figure 3.1, time series data were found in 463 spreadsheets out of a total of 2000 spreadsheets analyzed from the EUSES corpus. This represents 23% of the spreadsheets. Figure 3.2 shows that only 7% of the 3000 Enrons spreadsheets analyzed contain time series data. Figure 3.3 shows the average percentage of time series data in the total 5000 spreadsheets analyzed. Overall 14% of the analyzed spreadsheets contain time series data in one or all of their sheets.

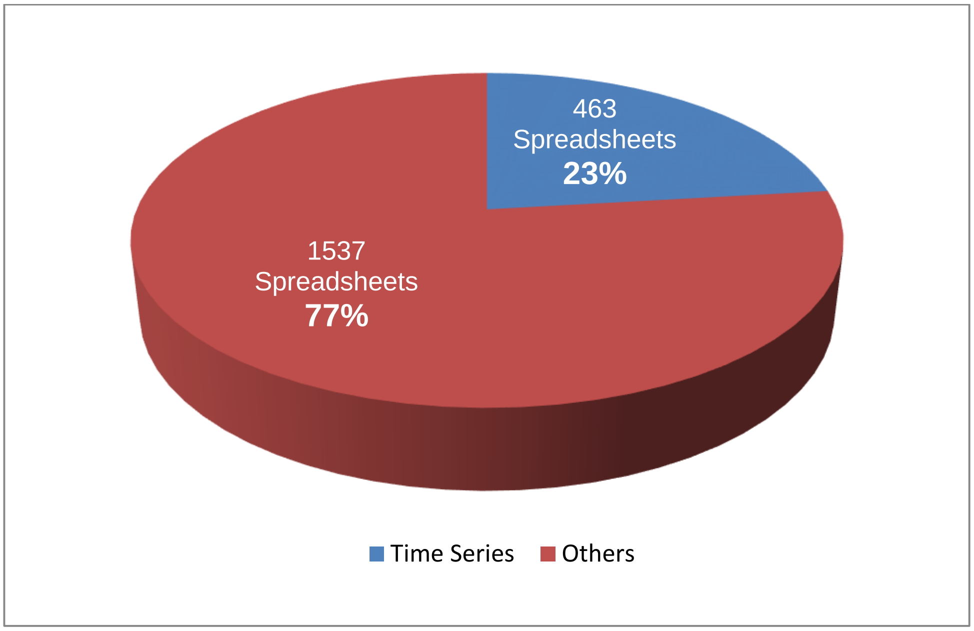


Figure 3.1: Frequency of occurrence of Time series data in EUSES Corpus (2000 Spreadsheets analyzed)

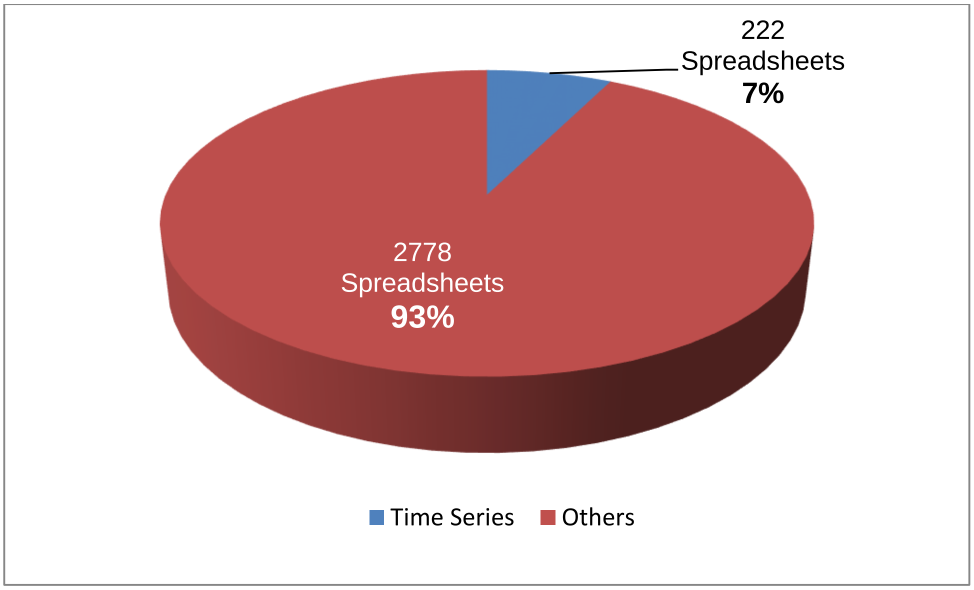


Figure 3.2: Frequency of occurrence of Time series data in Enrons Spreadsheet Corpus (3000 Spreadsheets analyzed)

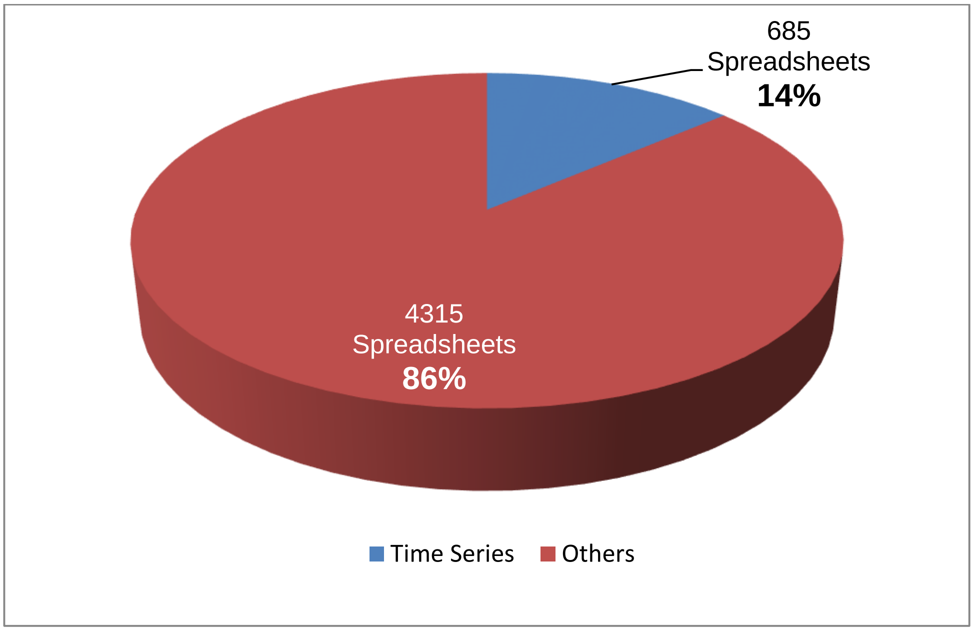


Figure 3.3: Average percentage of Time series data occurrence in spreadsheets (Total of 5000 Spreadsheets were analyzed)

3.2 Analysis of Time series Tools

The increase in the amount of time series data generated daily in both public and private sectors has necessitated the need for more efficient processes, methods and tools for analyzing the huge datasets. While much progress has been made in the development of time series tools, analysis and forecasting of time series data are still among the most important problems that analysts face across many fields today. These problems cut across many fields ranging from the natural sciences to finance, production operations and economics.

As a result, there is a widespread need for efficient time series software to manage and analyze these data in efficient ways. Many software applications have thus evolved in the recent years to meet up the challenge of time series analysis and forecasting. Some of these applications are domain-specific e.g. Aquarius time series (Aquatic Informatics Inc., 2014) for management of water time series, OpenEpi (Dean, Sullivan, & Soe, 2014) for Epidemiologic Statistics for Public Health; while other applications are not domain-specific and can be adapted to different domains depending on the needs e.g. MATLAB, Mathematica etc.

Another main challenge that is prevalent in the tasks of time series analysis is the skill-level required of an end-user in using the available tools. In most cases, a skilled analyst is needed to perform basic operations like processing of raw data, plotting of data and the analysis of time series. In effect, there is the need for large groups of people in a variety of fields who understands the advanced concepts of time series analysis and forecasting. This poses a great challenge since time series information is widely used in many establishments (like governmental agencies, fashion shops etc.) where end-users with little or no data analysis skills are found. It is thus essential to develop time series analysis software which are interactive and well suited for non-skilled end-users.

The technicalities of some analysis tools are also found to be unsuited for non-skilled end-users. For most existing tools, data has to be found, processed and even converted to the appropriate format prior to executing time series analysis. Data has to be professionally prepared before they can be used in these existing software packages.

In this section, I made an analysis of the most widely used tools for managing, analyzing, visualizing and forecasting time series data. A total of six tools were discussed in detail to enable the reader know the tools currently available for time series data management and analysis. Other time series tools are included in the comparison tables and analysis presented in the subsequent sections. The tools that are included in this section have been chosen on the basis of their popularity and user-base from various fields and analyzed on the basis of their support for time series management, transformation, analysis and visualization (McCullough & Vinod, 1999), (Zhu & Kuljaca, 2005) and (Zaslavsky, 2014).

3.3 Description of Tools

This thesis categorizes time series tools into domain-specific and general-purpose tools. Domain-specific time series tools are defined as tools designed to analyze time series data from a single domain e.g. water time series tools, geographical time series tools etc. On the other hand, a general purpose time series tools are those which can be adapted to analyze time series data from other fields. A general purpose tool can be referred to as domain-specific if used within a specific field, but a domain-specific tool cannot be regarded as a general purpose tool.

In this section, I briefly described one domain-specific web-based time series tool and a few general purpose time series tools. These descriptions are not only based on resources from tool developers but also include discussions and criticisms by end-users from various forums and social media. Emphasis is laid on the strengths and weaknesses of these tools in relation to how they support time series data management and analysis.

This section presents the summarized descriptions the following tools:

- OpenEpi
- MATLAB
- SAS / Econometrics and Time series Software (ETS)
- Microsoft Excel
- GMDH Shell
- R Language

In a later section, these aforementioned tools and eight others were analyzed and compared on the basis of the support they offer end users for working with time series.

3.3.1 OpenEpi: A Domain-Specific Time series Tool

OpenEpi is a free, web-based, open source, operating system-independent series of programs for analyzing time series data in medical fields e.g. epidemiology, biostatistics and public health among others. It provides a number of epidemiologic and statistical tools for summary data. OpenEpi was developed in JavaScript and HTML, and can be run in modern web browsers. The program can be run from the OpenEpi website or downloaded and run without a web connection. The source code and documentation is downloadable and freely available for use by other investigators. However, reviews from media organizations and in research journals reports that OpenEpi offers very little interactivity and poor front end designs for end users (Dean, Sullivan, & Soe, 2014).

OpenEpi was developed to perform analyses found in the DOS version of Epi Info modules StatCalc and EpiTable[6], to improve upon the types of analyses provided by these modules, and

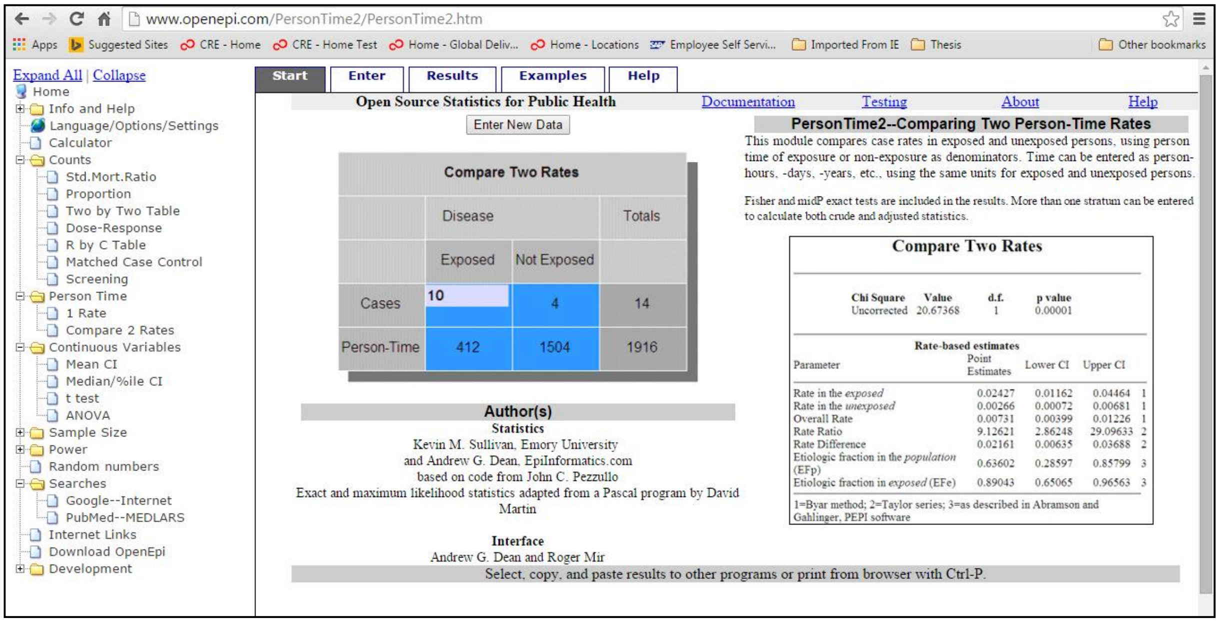
[6] StatCalc and EpiTable are modules in EPI, an earlier version of the OpenEPI. More details on http://www.owlnet.rice.edu/~mtomson/CEVE637/epi/epi.html

to provide a number of tools and calculations not currently available in Epi Info. It is the first step toward an entirely web-based set of epidemiologic software tools (OpenEpi, 2014). OpenEpi can be thought of as an important companion to Epi Info and to other programs such as SAS, SPSS, Stata, SYSTAT, Minitab and R language (see the subsequent sections for details of these programs).

Support for Time series Analysis:
For epidemiologists and other health researchers, OpenEpi performs a number of calculations and analysis based on tables not found in most epidemiologic and statistical packages. It focuses on time series data analysis and provides statistics for counts and measurements in descriptive and analytic studies, stratified analysis with exact confidence limits, matched pair and person-time analysis, sample size and power calculations, random numbers, sensitivity, specificity and other evaluation statistics, R x C tables, chi-square for dose-response, and links to other useful sites (OpenEpi, 2014).

Finally, OpenEpi also performs a test for trend, for both crude data and stratified data. Because OpenEpi is easy to use, requires no programming experience, and can be run on the internet, students can use the program and focus on the interpretation of results. Users can run the program in five languages including: English, French, Spanish, Portuguese or Italian.

However, OpenEpi provides very little support for implementing the standard time series models listed in chapter two. The software only focuses on the general statistical functionalities listed above.



Compare Two Rates

	Disease		Totals
	Exposed	Not Exposed	
Cases	10	4	14
Person-Time	412	1504	1916

Author(s)
Statistics
Kevin M. Sullivan, Emory University
and Andrew G. Dean, EpiInformatics.com
based on code from John C. Pezzullo
Exact and maximum likelihood statistics adapted from a Pascal program by David Martin

Interface
Andrew G. Dean and Roger Mir
Select, copy, and paste results to other programs or print from browser with Ctrl-P.

PersonTime2--Comparing Two Person-Time Rates

This module compares case rates in exposed and unexposed persons, using person time of exposure or non-exposure as denominators. Time can be entered as person-hours, -days, -years, etc., using the same units for exposed and unexposed persons.

Fisher and midP exact tests are included in the results. More than one stratum can be entered to calculate both crude and adjusted statistics.

Compare Two Rates

Chi Square	Value	d.f.	p value
Uncorrected	20.67368	1	0.00001

Rate-based estimates

Parameter	Point Estimates	Lower CI	Upper CI	
Rate in the *exposed*	0.02427	0.01162	0.04464	1
Rate in the *unexposed*	0.00266	0.00072	0.00681	1
Overall Rate	0.00731	0.00399	0.01226	1
Rate Ratio	9.12621	2.86248	29.09633	2
Rate Difference	0.02161	0.00635	0.03688	2
Etiologic fraction in the *population* (EFp)	0.63602	0.28597	0.85799	3
Etiologic fraction in *exposed* (EFe)	0.89043	0.65065	0.96563	3

1=Byar method; 2=Taylor series; 3=as described in Abramson and Gahlinger, PEPI software

Figure 3.4: OpenEpi - Time series analysis tool

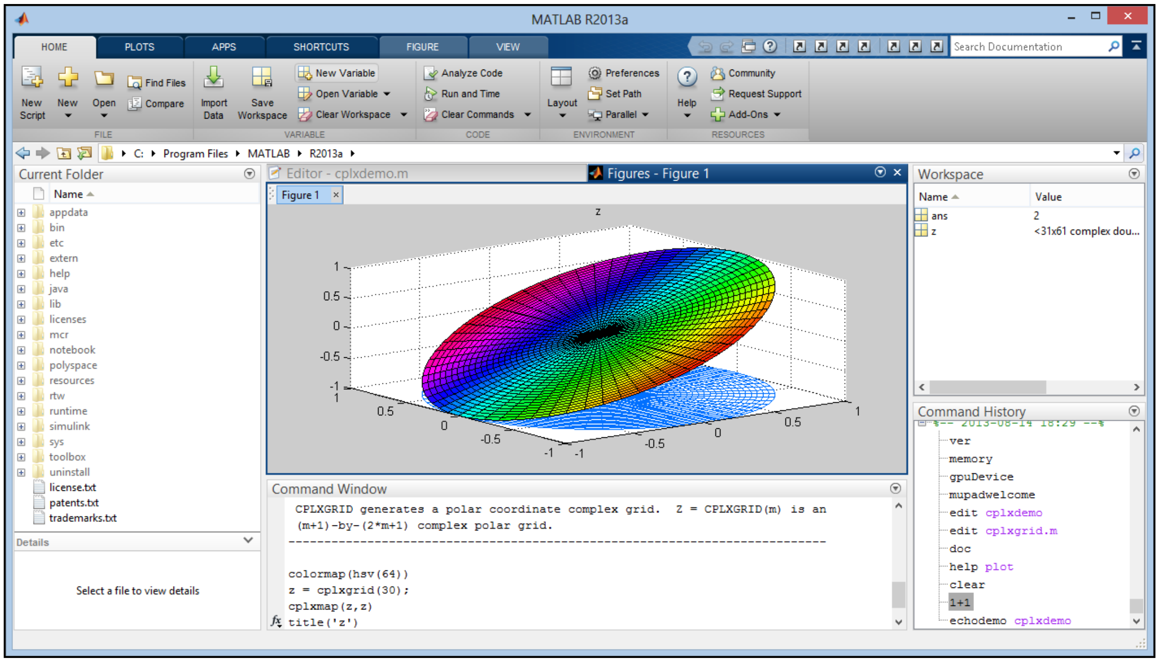
3.3.2 MATLAB:

MATLAB is a numerical computing environment and programming language. Maintained by the MathWorks, MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages (MATLAB, 2015). MATLAB is not a dedicated time series tool; it only offers specific classes and functions for managing, analyzing and forecasting time series.

The MATLAB application is built around the MATLAB language, and most use of MATLAB involves typing MATLAB code into the Command Window (as an interactive mathematical shell), or executing text files containing MATLAB code, including scripts and/or functions. This therefore makes MATLAB unsuited for end users with little or no programming skills. However, the tool provides good visualizations of time series data and has large set of advanced functionalities for data analysis, data management and forecasting.

Support for Time series Analysis:
MATLAB supports time series manipulations, plotting of functions and data, implementation of time series algorithms and models, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python. MATLAB have support for implementing almost many time series model. It provides specific functions for:

- Adding or deleting time series
- Manipulating time series objects
- Performing descriptive statistics for time series objects
- Querying and setting time series collection object properties
- Plotting time series collection objects, etc.



Name	Value
ans	2
z	<31x61 complex dou...

```
 CPLXGRID generates a polar coordinate complex grid.  Z = CPLXGRID(m) is an
 (m+1)-by-(2*m+1) complex polar grid.
------------------------------------------------------------------------------

colormap(hsv(64))
z = cplxgrid(30);
cplxmap(z,z)
title('z')
```

```
%-- 2013-08-14 18:29 --%
ver
memory
gpuDevice
mupadwelcome
edit cplxdemo
edit cplxgrid.m
doc
help plot
clear
1+1
echodemo cplxdemo
```

Figure 3.5: MATLAB - Time series analysis tool

3.3.3 SAS / Econometrics and Time series Software (ETS)

The Econometrics and Time series Software is a software suite developed by Statistical Analysis System (SAS) Institute for advanced analytics, data management, and predictive analytics of time series data. It is high-performance econometrics software which supports simple to advanced implementation of time series models. The software can perform simple to complex analysis of huge sets of time series data. It also provides support for time series forecasting (SAS Institute Inc., SAS/ETS Software, 2014).

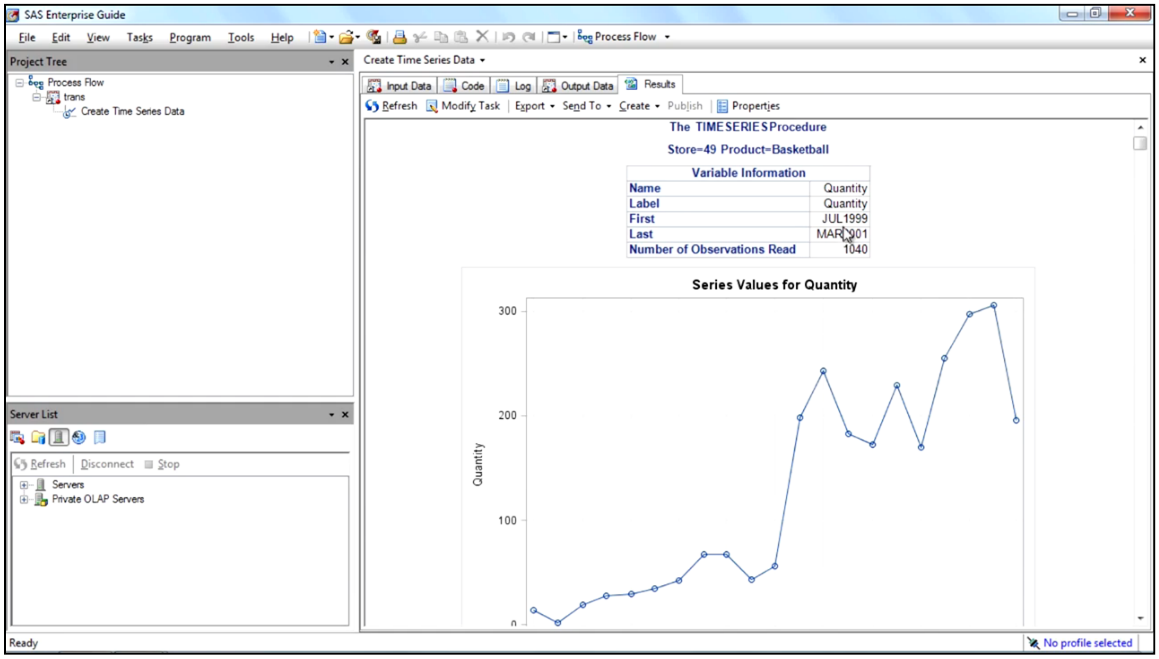
Although the SAS/ETS software offers advanced support for time series data, it is expensive and not easily accessible to students or small-scale establishments who are valid end users of the software. Users have also reported that the desktop-based software is not user friendly and could be difficult to operate (Leonard & Wolfe, 2005).

Support for Time series Analysis:
SAS/ETS software provides extensive facilities for analyzing time series and performing forecasts. SAS/ETS software includes a wide range of tools for analyzing time series data. One can estimate relationships and produce forecasts that make use of information in past values, independent or explanatory variables, and indicator or dummy variables. In addition, users can model and predict the autoregressive conditional heteroscedastic (ARCH) model or its

generalizations (GARCH)[7]. Additional tools provide regression analysis for linear models with distributed lags and time series cross-sectional regression analysis for panel data.

Users can also perform multiple regression in the presence of serially correlated error terms, fit models that allow for an error term generated by an autoregressive integrated moving-average (ARIMA) process, or use spectral analysis to decompose a series into cyclical components or to perform frequency domain tests (SAS Institute Inc., SAS/ETS Software, 2014).

The software includes a point-and-click application for exploring and analyzing univariate time series data. Users can use the automatic model selection facility to select the best-fitting model for each time series, or use the system's diagnostic features and time series modeling tools interactively to develop forecasting models customized to best predict your time series. The system provides both graphical and statistical features to help users choose the best forecasting method for each series. More so, many of the SAS/ETS procedures have options that facilitate the forecasting of time series variables.



The TIMESERIESProcedure

Store=49 Product=Basketball

Variable Information	
Name	Quantity
Label	Quantity
First	JUL1999
Last	MAR2001
Number of Observations Read	1040

Figure 3.6: SAS/ETS - Time series analysis tool

[7] The ARCH and GARCH are standard statistical models for time series analysis.

3.3.4 Microsoft Excel:

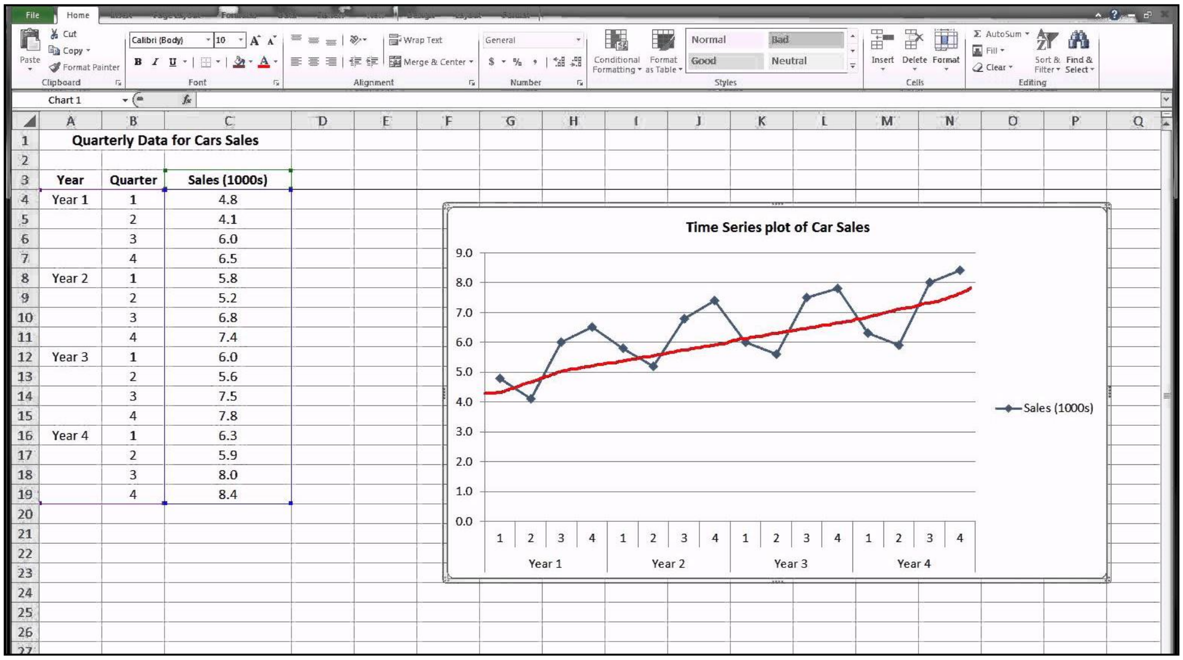
Microsoft Excel (MS Excel) is a spreadsheet application developed by Microsoft for Microsoft Windows, Mac OS, and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications. MS Excel can perform basic statistics, regression, correlation, ANOVA, and other statistical functionalities. The application has been a very widely applied spreadsheet application in almost every field as the industry standard for spreadsheets. Excel is part of Microsoft Office (Frye, 2010).

Support for Time Series Analysis:

Like MATLAB, MS Excel is not dedicated time series software, but it provides lots of features that are used for time series analysis. These features are amplified by add-in applications developed by third-party developers. Some of the add-ins developed for Microsoft Excel are discussed below (Time-Series Forecasting, 2014).

- **ForecastX Wizard** by John Galt Solutions, Inc.: It is designed for non-statisticians but offers a full range of features for advanced users. It can perform statistical forecasting (seasonal, non-seasonal, growth, slow-moving items, event modeling); calculate safety stock; build inventory plans etc.
- **PEERForecaster** by Delphus: An Excel Add-in with all the horsepower of a full-fledged forecast modeling tool without the overhead commonly associated with many forecasting solutions. The models include all the well-known techniques from simple smoothing, holt-trending, holt-winters seasonal models, and damped trend exponential smoothing models to the Box Jenkins ARIMA models[8].
- **XL Miner** by Cytel Software Corporation: All versions of XLMiner offer options for modeling time series data, smoothing time series data and XLMiner also provides special functions for partitioning time series data. The tool provides ARIMA, ACF (autocorrelations), PACF (partial autocorrelations) for modeling time series data. For smoothing time series data, XLMiner offers: Exponential, Double exponential, Moving average, Holt Winter, Hold Winter - no trend.
- **EZ Forecaster** by ParkerSoft.: Designed for users who needs to create business forecasts, ezForecaster is a powerful, yet remarkably easy-to-use time series forecasting add-in for Microsoft Excel. ezForecaster can automatically choose a suitable forecasting method using a wide variety of time series techniques, but also allows users to select a method manually.
- **NumXL** by Spider Financial. NumXL is an Excel Add-in that provides users an intuitive and powerful solution for time series analysis and forecasting. NumXL wraps common mundane calculations such as auto-correlation, log-likelihood, model fitting/calibration, residuals diagnosis, forecasting and much more, and into a simple extension of MS Excel. Users can use NumXL functions via Excel menus, entering them directly into workbook cells, or by using the Excel function wizard.

[8] Details of these models can be found in http://peerforecaster.com/index.htm



Quarterly Data for Cars Sales

Year	Quarter	Sales (1000s)
Year 1	1	4.8
	2	4.1
	3	6.0
	4	6.5
Year 2	1	5.8
	2	5.2
	3	6.8
	4	7.4
Year 3	1	6.0
	2	5.6
	3	7.5
	4	7.8
Year 4	1	6.3
	2	5.9
	3	8.0
	4	8.4

Figure 3.7: Microsoft Excel - Time series analysis tool

3.3.5 GMDH Shell

Group Method of Data Handling (GMDH) Shell is a forecasting-software that enables users of all types to easily and accurately forecast their data. It's developed by GMDH LLC - a privately held company founded in 2009 with an idea to build the best forecasting software. GMDH is a state of the art predictive modeling technology with accuracy and reliability proven by over 40 years of scientific research. GMDH Shell is powerful for forecasting time series for small businesses, traders and scientists.

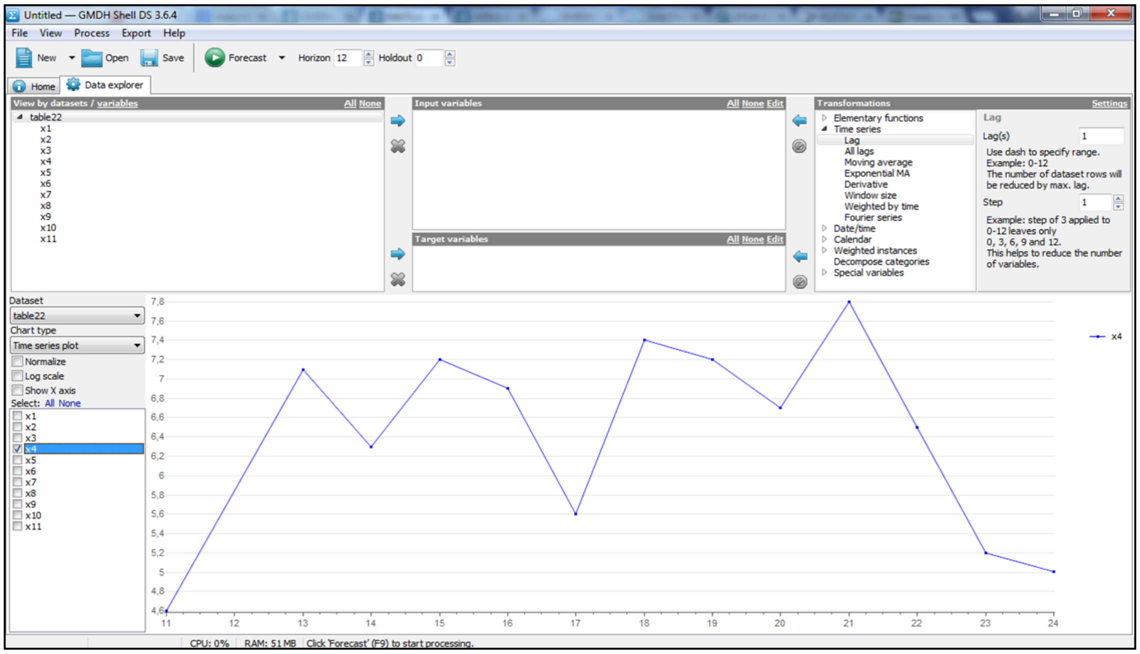
The tool provides an easy-to-use way to accurately forecast time series, create classifiers and regression models. Based on artificial neural networks, it allows users to easily create predictive models, as well as preprocess data with simple point-and-click interface. (GMDH Shell LLC., 2013).

Support for Time Series Analysis:

GMDH Shell provides good analysis and forecasting, it also features comprehensive data-manager tool for quick entering of input data, rich visualization capabilities and templates ready for instant analysis of time series data. GMDH shell also has a user-friendly interface and easy to use wizards for performing data imports, analysis and visualization of time series. The interface of the program isn’t overloaded with excessive details, so even a low experienced user can quickly begin using it. However, users may get confused while performing more complicated analysis (GMDH Shell LLC., 2013).

One the other hand, the tool is slightly expensive for students and small scale businesses. It is also slow for huge data sets. The tool responds slowly for datasets more than 5,000 records. For example, a dataset of about 10,000 rows is analyzed in 2 hours.

Overall, GMDH Shell provides about the most user-friendly interface and one of the most powerful end-user oriented time series analysis software on the market. With it, time series analysis and forecasting can be easily done by end users with little data analysis skills.



Horizon 12 Holdout 0

Lag(s) 1

Use dash to specify range.
Example: 0-12
The number of dataset rows will be reduced by max. lag.

Step 1

Example: step of 3 applied to 0-12 leaves only 0, 3, 6, 9 and 12.
This helps to reduce the number of variables.

Dataset: table22
Chart type: Time series plot

□ Normalize
□ Log scale
□ Show X axis

□ x1
□ x2
□ x3
☑ x4
□ x5
□ x6
□ x7
□ x8
□ x9
□ x10
□ x11

CPU: 0% RAM: 51 MB Click 'Forecast' (F9) to start processing.

Figure 3.8: GMDH Shell - Time series analysis tool

3.3.6 R Language

R is a free software environment for statistical computing and graphics. R is not developed specifically for time series analysis; however it supports many statistical analysis methods and functions for analyzing and forecasting time series data, and has good graphical display. It is open source. R is a whole language with its working bundled application as specially the "de facto" standard for data analysis and data mining (McLeod, Yu, & Mahdi, 2011). However, it is better suited for advanced users with programing skills and good understand of data analysis models.

Support for Time series Analysis:

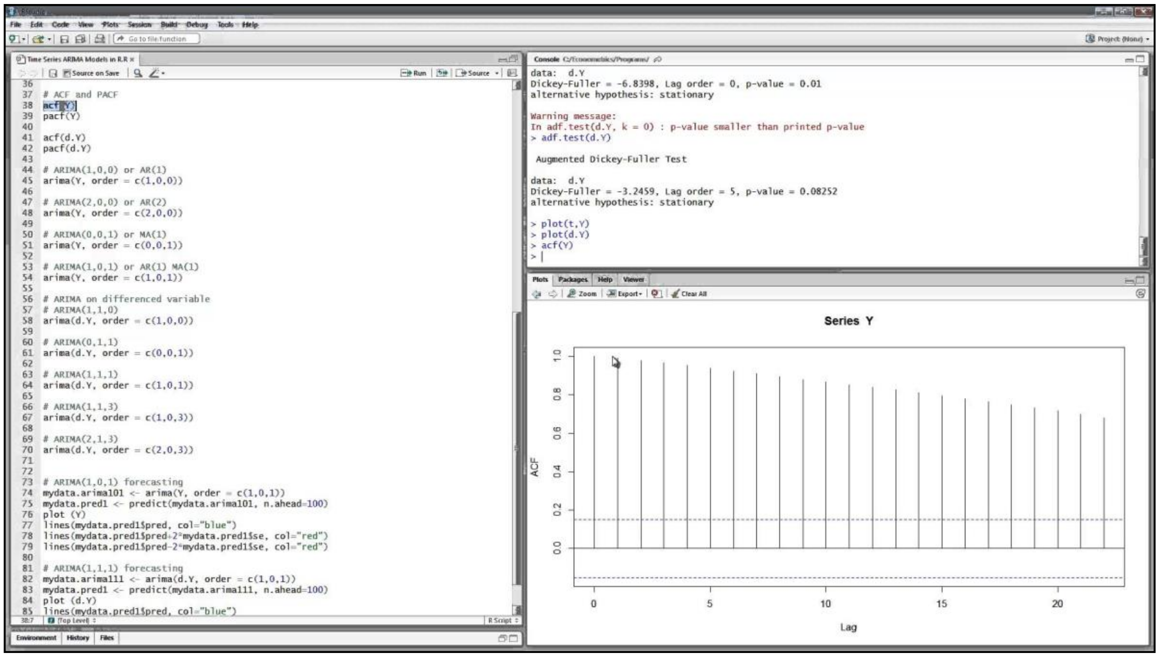
R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time series analysis, classification, clustering...) and graphical techniques, and is highly extensible. Base R ships with a lot of functionality useful for time series, in particular in the stats package. Base R contains substantial infrastructure for representing and analyzing time series data. The fundamental class is "`ts`" that can represent regularly spaced time series (using numeric time stamps). Hence, it is particularly well-suited for annual, monthly, quarterly data, etc.

Time series plots are obtained with `plot()` applied to `ts` objects. (Partial) autocorrelation functions plots are implemented in `acf()` and `pacf()`. Seasonal displays are obtained `usingmonthplot()` in stats and seasonplot in forecast. A few other time series classes include:

- As mentioned above, "`ts`" is the basic class for regularly spaced time series using numeric time stamps.
- The `zoo` package provides infrastructure for regularly and irregularly spaced time series using arbitrary classes for the time stamps (i.e., allowing all classes from the previous section). It is designed to be as consistent as possible with "`ts`". Coercion from and to "`zoo`" is available for all other classes mentioned in this section.
- The package `xts` is based on `zoo` and provides uniform handling of R's different time-based data classes.
- The class "`timeSeries`" implements time series with "`timeDate`" time stamps.
- The `forecast` package provides a class and methods for univariate time series forecasts, and provides many functions implementing different forecasting models including all those in the stats package.

R Language also supports frequency analysis of time series, decomposition and filtering, analysis of seasonality, analysis of nonlinear time series, dynamic regression models and other time series operations (McLeod, Yu, & Mahdi, 2011).

Even though R Language offers a lot of support for time series analysis, it requires medium-level to high-level programming skills for users to comfortably perform analysis and forecasting of time series. The tool is therefore not well suited for end users without sound programming skills.



```
# ACF and PACF
acf(Y)
pacf(Y)

acf(d.Y)
pacf(d.Y)

# ARIMA(1,0,0) or AR(1)
arima(Y, order = c(1,0,0))

# ARIMA(2,0,0) or AR(2)
arima(Y, order = c(2,0,0))

# ARIMA(0,0,1) or MA(1)
arima(Y, order = c(0,0,1))

# ARIMA(1,0,1) or AR(1) MA(1)
arima(Y, order = c(1,0,1))

# ARIMA on differenced variable
# ARIMA(1,1,0)
arima(d.Y, order = c(1,0,0))

# ARIMA(0,1,1)
arima(d.Y, order = c(0,0,1))

# ARIMA(1,1,1)
arima(d.Y, order = c(1,0,1))

# ARIMA(1,1,3)
arima(d.Y, order = c(1,0,3))

# ARIMA(2,1,3)
arima(d.Y, order = c(2,0,3))


# ARIMA(1,0,1) forecasting
mydata.arima101 <- arima(Y, order = c(1,0,1))
mydata.pred1 <- predict(mydata.arima101, n.ahead=100)
plot (Y)
lines(mydata.pred1$pred, col="blue")
lines(mydata.pred1$pred+2*mydata.pred1$se, col="red")
lines(mydata.pred1$pred-2*mydata.pred1$se, col="red")

# ARIMA(1,1,1) forecasting
mydata.arima111 <- arima(d.Y, order = c(1,0,1))
mydata.pred1 <- predict(mydata.arima111, n.ahead=100)
plot (d.Y)
lines(mydata.pred1$pred, col="blue")
```

```
data:  d.Y
Dickey-Fuller = -6.8398, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(d.Y, k = 0) : p-value smaller than printed p-value
> adf.test(d.Y)

 Augmented Dickey-Fuller Test

data:  d.Y
Dickey-Fuller = -3.2459, Lag order = 5, p-value = 0.08252
alternative hypothesis: stationary

> plot(t,Y)
> plot(d.Y)
> acf(Y)
> |
```

File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Project (None)
Time Series ARIMA Models in R.R
Source on Save
Run
Source
38:7
(Top Level)
R Script
Environment History Files
Console C:/Economics/Programs/
Plots Packages Help Viewer
Zoom Export Clear All
Series Y
ACF
Lag
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20

Figure 3.9: R Language - Time series analysis tool

3.4 A Generic Comparison of Time Series Tools

As established in the earlier sections, there are a good number of tools currently used in the field of time series analysis. (Beiwald, 2009) made a generic comparison of the some software for time series analysis. His deductions are shown in table 3.1 below.

(Beiwald, 2009) compared six tools on the basis of the main advantages and disadvantages they offer end users. His description and comparison of the tools aligns with the discussions in section 3.3 of this thesis. As can be deduced from the table, of all the tools, only R-Language is open source and relatively affordable and accessible to end users. Other tools are expensive and therefore not easily accessible. The R Language would have *"carried the day"* save the fact that it has a steep learning curve and thus undesirable by users with low programming skills. The most popular of the tools is the Microsoft Excel. According to (Beiwald, 2009), although MS Excel provides users with an easy and flexible interface for data analysis, it provides very little support for implementing standard time series analysis and forecasting models.

It is also notable that these tools are designed for skilled analysts in the fields of finance, statistics, engineering, business, science and government.

Table 3.1: Comparison of time series data analysis packages (Beiwald, 2009)

Name	Advantages	Disadvantages	Open source?	Typical Users
R	Library support; visualization	Steep learning curve	Yes	Finance; Statistics
Matlab	Elegant matrix support; visualization	Expensive; incomplete statistics support	No	Engineering
Excel	Easy; visual; flexible	Inability to handle large datasets	No	Business
SAS	Large datasets	Expensive; outdated programming language	No	Business; Government
Stata	Easy statistical analysis		No	Science
SPSS	Like Stata but more expensive and worse			

Another generic description of 13 time series tools is presented in table 3.2. The table describes the developers of each tool and shows which tools are web based. One can immediately infer that most of the tools available for time series analysis are not web based and this poses a great challenge, since web based software applications are seen to offer more advantages relative to stand-alone desktop software.

Unlike traditional applications, web-based applications are accessible anytime, anywhere, via a computer enabled with an Internet connection, putting the user in charge of where and when they access the application. Moreover, content can also be customized for presentation on any device connected to the internet, including PDAs, mobile phones, etc., further extending the user's ability to receive and interact with information. Other advantages include: cost savings, improved efficiency and user productivity and enhanced data security (Laidre, 2012).

This thesis advocates for the development of more web based tools for managing time series data such as one presented in chapters four and five. More justifications for web-based tools are discussed in chapter four.

Table 3.2: Comparison of time series data analysis packages II

Product	Developer	Date of Latest version	Open source	Software license	Web-based?
Gretl	The Gretl Team	September 20, 2014	Yes	GNU GPL	No
Mathematica	Wolfram Research	March 30, 2015	No	Proprietary	No
MATLAB	MathWorks	New releases twice per year	No	Proprietary	No
Minitab	Minitab Inc.	February 18, 2014	No	Proprietary	No
R	R Foundation	March 9, 2015	Yes	GNU GPL	No
SAS	SAS Institute	July, 2013	No	Proprietary	No
SPlus	Insightful Inc.	2010	No	Proprietary	No
SPSS	IBM	Aug 13	No	Proprietary	No
Stata	StataCorp	June 24, 2013	No	Proprietary	No
STATISTICA	StatSoft	November, 2010	No	Proprietary	No
OpenEpi	A. Dean, K. Sullivan, M. Soe	June 23, 2011	Yes	GNU GPL	Yes
Weka	Machine Learning Group at the University of Waikato	March 17, 2015	Yes	GNU GPL	No
Systat	Systat Software Inc.	May, 2012	No	Proprietary	No

3.5 Requirement-based Comparison of Existing Time series Tools

As already mentioned, there are a good number of tools being developed to ensure the accurate analysis of time series data. The important question however is whether these tools are usable for non-experienced end users. Do these tools meet the expected requirements for end-user software?

In this section, I identified five basic requirements for an end-user oriented time series tool, based on interviews and discussions of end-users on online forums and social media. Up to 300 statements and reviews from different end users were analyzed, to understand the most important requirements expected of these tools by end users. The result of this exercise, as shown in table 3.3, adds to one of the key contributions of this thesis, which is to define and clearly describe functional requirements for a web based, interactive software for managing, analyzing and transforming time series. These functional requirements are fully described in Chapter four of this thesis.

As shown in table 3.3, each criterion represents a condensed summary of requirements that users either praised or criticized about different tools. Each criterion is used to compare different tools in this section. The purpose is to understand how these tools fared as end user oriented software. Table 3.3 describes the five main criteria used in the comparison of these tools.

Table 3.4 shows the comparison of 14 time series tools on the basis of the criteria shown in table 3.3. It can be deduced from the table that most of the tools are only moderately user-friendly and others are essentially difficult. Only GMDH Shell provides an easy to use environment however, it is rated poor on its delivery of the non-functional requirements. The reason for this is explained in section 3.3.5. Also, the table shows that about 72% of the software compared only provides support for implementing analysis and prediction models of time series data. The other 28% provides support for all standard models for analysis, predictions and transformations for univariate and multivariate time series data. These include: MATLAB, SAS/ETS, Mathematica and R Language.

It is also interesting to discuss the cost and availability of these software to end users. The three open source software: Gretl, R Language and OpenEpi are free, while most of the other software are expensive with prices as high as 10,000 euros (in the case of SPSS).

Table 3.3: Description of criteria for requirement-based comparison of time series tools

Criteria	Definition / Description	Values
User friendliness	• Ease of use and learnability of tool to end users. e.g. use of 'click-and-go' buttons, least/no command-driven actions etc. • Use of intuitive and common patterns, interfaces and menus • Least dependency on user inputs for analysis and forecasting of time series	• 1- Difficult • 2 - Moderate • 3 - Easy
Front-end/GUI	• Good-looking front-end/interface designs • Use of high-standard look and feel elements for layout and flow, colours etc. • Quality of graphical representation for visualizing time series	• 1 – Poor • 2 – Fair • 3 – Good • 4 – Excellent
Support for time series analysis	• Level of support for time series models and tool integration with spreadsheets • **Basic**: Supports only analysis models • **Moderate**: Supports analysis and prediction models • **Advanced**: Support all standard models for analysis, predictions, transformations for univariate and multivariate Time series	• 1 – Basic • 2 – Moderate • 3 – Advanced
Non Functional Requirements	The non-functional requirements considered are: • Performance • Reliability • Security • Compatibility • Robustness	• 1 – Poor • 2 – Fair • 3 – Good • 4 – Excellent
Cost & Availability	This is calculated as an average of the offers by the developers • **High**: 1001euros and above • **Medium**: 51euros - 1000euros • **Low**: 1 - 50euros • **Free**: incurs no financial costs (e.g. Open source)	• 1 – High • 2 – Medium • 3 – Low • 4 – Free

Table 3.4: Requirement-based comparison of time series tools

Time Series Tool	User friendliness	Front-end/GUI	Support for time series analysis	Non-functional requirements	Cost & Availability
MATLAB	Moderate	Fair	Advanced	Good	High
SAS/ETS	Moderate	Good	Advanced	Good	Medium
MS Excel*	Moderate	Fair	Moderate	Fair	Low
Mathematica	Difficult	Fair	Advanced	Fair	Medium
MINITAB	Moderate	Fair	Moderate	Good	High
SPSS	Moderate	Fair	Moderate	Good	High
Systat	Moderate	Fair	Moderate	Fair	High
DTREG Analysis & Forecasting	Difficult	Poor	Moderate	Fair	High
Weka	Moderate	Fair	Moderate	Fair	Free
GMDH Shell	Easy	Fair	Moderate	Poor	Medium
R Language	Difficult	Poor	Advanced	Fair	Free
GRETL	Moderate	Fair	Moderate	Fair	Free
STATA	Difficult	Fair	Moderate	Fair	Medium
OpenEpi	Moderate	Poor	Moderate	Fair	Free

*MS Excel was analyzed on the basis of the time series add-ins discussed in section 3.3.4

3.6 Assessment of Time series Tools based on their Support for Thesis Objectives

Apart from reviewing the performances of time series tools from an end user perspective, an assessment of these tools on the basis of how they fulfil the thesis objectives was conducted. These objectives include: managing, transforming, analyzing and visualizing time series. The important question here is how many these tools satisfy the requirements for a user-oriented web based time series tool.

For each of the thesis objectives listed above, the tools were assessed using the following criteria:

- **No Support** – Thesis Objective is not supported
- **Poor** – There is no direct or strong support for functionality. It can however be adapted by end users manually or by programmatically combining software functions
- **Fair** – Thesis Objective is supported but is not easy to use by end users
- **Good** – Thesis Objective is supported and is easy to use by end users

This assessment has been based on the personal interaction and knowledge of each tool as well as knowledge from developers' manuals and reviews of the tools.

Table 3.5 shows the assessment of the tools. As a generic deduction, many of the tools are only fairly good for managing time series. Open Epi for instance has no support for preparing data input or handling errors in data. Once an inconsistency is discovered in data, the software fails. Many of the other tools also require that data is well prepared an error free before being introduced to the system. Another interesting discovery is that many of the tools do not directly support the transformation of time series data; which involves the conversion of time series from one time frequency to another. This is crucial to the work of this thesis as it focuses on software with support for MapReduce technique for transforming time-stamped and time series data.

While many of the tools offer sophisticated support for analyzing time series, they offer relatively fair visualizations of these analyses. As shown in the table, only SAS/ETS, GMDH Shell and Weka offer good visualizations – which involves the effective presentation technique and use of good graphics.

Table 3.5: Assessment of Time series Tools based on their Support for Thesis Objectives

Time Series Tool	Time series Management	Time series Transformation	Time series Analysis	Time series Visualization
MATLAB	**Fair**	**Fair**	**Good**	Fair
SAS/ETS	**Fair**	Good	Good	Good
MS Excel*	**Fair**	**Poor**	**Fair**	**Fair**
Mathematica	**Fair**	**Poor**	**Good**	**Poor**
MINITAB	**Fair**	**Poor**	**Good**	**Fair**
SPSS	**Fair**	**Poor**	**Fair**	**Poor**
Systat	**Good**	**Fair**	**Good**	**Fair**
DTREG Analysis & Forecasting	**Good**	**Fair**	**Good**	**Fair**
Weka	**Fair**	**No Support**	**Good**	**Good**
GMDH Shell	**Good**	**Good**	**Fair**	**Good**
R Language	**Fair**	**Fair**	**Good**	**Fair**
GRETL	**Good**	**Fair**	**Fair**	**Fair**
STATA	**Fair**	**Fair**	**Good**	**Fair**
OpenEpi	**Fair**	**No support**	**Fair**	**Poor**

3.7 Description of Thesis Tool based on the Defined Criteria

The work of this thesis aims to design an interactive web based time series tool which will fulfill the most important requirements for an end user oriented time series software in conformation with the thesis objectives.

The approach of this work is not to develop a perfect flawless time series software, however it is intended that the software will incorporate the researched requirements as well as best practices in the field of time series data management, analysis, prediction and transformation.

This section projects the software being designed in this thesis in the light of the comparison criteria used in sections 3.5 and 3.6. Having analyzed and assessed other time series software, the question is: how should the Thesis-Time-series-tool compare with the others tools using the exact same criteria? Tables 3.6 and 3.7 show a summary of the scoring of the tool on the basis of the stated criteria. These are explained below.

Table 3.6: Description of Thesis Time series Tools on its Support for Thesis Objectives

Time Series Tool	Time series Management	Time series Transformation	Time series Analysis	Time series Visualization
Thesis Time series Tool	Good	Good	Good	Good

Table 3.7: Description of Thesis Tool based on the End-user Requirements

Tool	User friendliness	Front-end/GUI	Support for time series analysis	Non-functional requirements	Cost & Availability
Thesis Time series Tool	Easy	Good	Moderate	Good	Free

Table 3.6 shows the thesis time series should offer good support all the thesis objectives. The contents of table 3.7 are explained as thus.

A. User Friendliness

The thesis time series tool should be easy to use and learned by users who have little or no knowledge of data analysis or programming skills. It should support standard methods like automatic model selection, automatic outlier detection etc.; such that analysis and forecasting of time series will require least dependency on user inputs.

B. Front-end/GUI

The tool should have eye-catching front-end designs with a high quality of graphical representation for visualizing time series. It should support the plotting of time series data on a line graph, frequency chart, scatter plot, histogram, auto correlation chart, surface chart, 3d bubbles and other effective ways of visualizing time series.

C. Support for Time series Analysis

The tool should provide a moderate level of support for time series models i.e. it should support analysis and prediction models for univariate and multivariate time series. The tools should also be integrated with a spreadsheet application like MS Excel such that data import and export to .xls, .xlsx, .csv formats is possible.

D. Cost & Availability

The tool should be free, web based and open source.

In the next chapter, the functional requirements for the thesis time series tool are described into more details. These requirements cut across the different areas of the software and include its support for managing, transforming, analyzing and visualizing time series data.

4 Requirements for an Interactive and Web-based Time Series Software

(Rouse, 2014) defines software requirements as the intended purpose and environment for software under development. These requirements are the services that a software system must provide to users and the constraints under which the software must operate. Software requirements are usually documented in a Software Requirement Specification (SRS); a document which fully describes what the software will do and how it will be expected to perform.

The success of any software design depends on the requirements gathered at the planning stage of the system's development life cycle. If the requirements are not thorough, it is almost impossible to build not only the right software, but to build the software right. Therefore, tons of time is spent in the requirements gathering phase before even beginning any design or implementation (Rosene, 2013). The case is the same for the work of this thesis.

Having analyzed the major state-of-the-art software currently being used for time series management and analysis, this thesis deduces the key characteristics that make existing time series software easy to use for end users. These characteristics, however unexpected, seem to excite end users about time series tools and thus forms the basis for the requirements discussed in this section.

A major concern for software engineers about requirements gathering is the source of the requirements. The question is often asked whether requirements should come from the end users or from the engineers themselves. Should the end user have a say in how specific software are designed and programmed? The answers depend on a number of variables one of which is the context and purpose of the software being designed. For an end-user oriented design, it is generally believed that requirements should come from the customers and users of the software, rather than from the engineers themselves. In this case, requirements gathered should consider the users' capabilities, needs and preferences (Andrew, Robin, Laura, & Alan, 2010).

Once gathered, these requirements are essential for creating design specifications which specify the internal behavior of the software. The design specifications are used to lay out the implementation strategy for ensuring that the software meets all of the requirements. This is done by assigning appropriate priorities to each requirement so that the highest priority requirements are taken care of before the low priority ones (Rosene, 2013).

Another factor that comes into play during requirements gathering phase of any software is the type of software being developed i.e. is the software a traditional desktop software or web-based software. To some extent, software requirements differ depending on whether software is web-based or not. Generally web-based development offers a good number of advantages for end user oriented software, hence its usage for this thesis. The benefits of web-based applications according to (Laidre, 2012) are listed below:

a. Easier to develop and maintained
b. Adaptable to increased workload i.e. easily extensible
c. With web-based applications, users access the system via a uniform environment—the web browser.
d. The user interface of web-based applications is easier to customize than it is in desktop applications. This makes it easier to update the look and feel of the application, or to customize the presentation of information to different user groups.
e. Web-based architecture makes it possible to rapidly integrate enterprise systems, improving work-flow and other business processes.
f. Security: Web-based applications are typically deployed on dedicated servers, which are monitored and maintained by experienced server administrators. This is far more effective than monitoring hundreds or even thousands of client computers, as is the case with new desktop applications.
g. Web-based software improves communications and coordination.

This chapter gives the reader a good understanding of the software being designed in this thesis. First I listed the uses of the software. Then, the overall functional requirements of the thesis software are described. These requirements are gathered by observing and analyzing the existing time series software. Many requirements were gathered but only some of them are considered relevant for this thesis. The selected requirements are based on how they support the software design as an end user orientation design.

After describing the functional requirements of the thesis software, I listed the specific technical features which the software must possess in order to satisfy the requirements. These technical features and their usage scenarios are illustrated in the later sections. More specific use cases and mock ups of the thesis software are described in Chapter five.

4.1 Description of Software Uses and Requirements Categorization

This section briefly describes the uses of the thesis software and explains the basis for categorizing the requirements.

The software is designed for use in a wide variety of fields like business and economics, government, academia, finance, geology etc. It is appropriate for end users that relate often with time series data e.g. medical researchers, biostatisticians, economists, sociologists, political scientists, bankers, geographers, psychologists, epidemiologists, geologists, social scientists and other researchers and general users needing to analyze time series data. The common theme relating the many application areas of the software is time series data i.e. the software is useful whenever it is necessary to analyze or predict processes that take place over time or to transform time-stamped data to time series (usually as preparation for further analysis).

In general, the thesis time series software is useful for:

- Time series data management
- Time series analysis and forecasting
- Transforming time-stamped data to time series
- Seasonal adjustment of time series data
- Plotting and reporting of trends and forecasts of time series values

The software is designed to be easy to use and learned. This makes it an appropriate tool for instruction of statistics and research methods in time series. A typical usage scenario is for an advanced student carrying out a doctoral–level research. The software can be used by the student to efficiently generate a fully reproducible analysis of the time series.

One major consideration in the design of the software was interactivity. Interactivity of the software for end users is a central attribute which the software is built upon. Software that is truly interactive should be simple for end users to learn and use. Users should quickly see the value the software brings, and be able to learn how to use the key features in a short amount of time.

This is however not to put away the importance of the utility of the software. Utility refers to the ability of the product to perform a task or tasks. The more tasks the product is designed to perform, the more utility it has. Both qualities (i.e. interactivity and utility) are necessary for an end user oriented development. Obviously, if a program is highly interactive but doesn't do anything of value, nobody will have much reason to use it. And users who are presented with a powerful program that is difficult to use will likely resist it or seek out alternatives.

The requirements of the software are thus gathered on the basis of these desired qualities. These requirements have been gathered from different sources and online reviews, and from an in-depth understanding of how the existing time series tools work. The sources and corresponding tools studied are shown in Table 4.1.

Table 4.1: Requirements Gathering - Tools Studied and Sources

Tool	Source(s)
1. SAS/ETS	• (SAS Institute Inc., SAS/ETS 9.1 Users Guide, 2004)
2. GMDH Shell	• (GMDH Shell LLC., 2013)
3. MATLAB	• (The MathWorks Inc., 2015)
4. GRETL	• (Cottrell & Lucchetti, 2015)
5. STATA	• (Baum, 2003) • (STATA, 2003)
6. DTREG	• (DTREG, Time Series Analysis, 2010)

In order to capture the main purposes of the thesis software, the requirements are grouped into four main categories namely:

A. **Requirements for time series data management**
This includes all requirements of the software that relate to the efficient management of time series data; including the description of the general interface and outlook of the software.

B. **Requirements for time series transformation**
These are requirements which support the transformation time-stamped data to time series, quantifying qualitative data inputs, as well as checking the time series output for correctness.

C. **Requirements for time series analysis**
These are requirements that ensure that software accurately analyzes time series and predict futures values with least dependency on user input.

D. **Requirements for time series visualization**
This includes all requirements which ensure standardized visualization and plotting of time series data as well as the persistent interaction between the time series data input and respective graphical representations.

As will be seen in later sections, the grouping of these requirements are not exclusively independent i.e. a requirement for managing time series data may also be relevant for transforming time-stamped data to time series, while some requirements for analyzing time series could also be important for visualizing the analysis output.

In the next sections, each of the four categories of functionalities is described and their main processes are illustrated.

4.2 Time series Data Management

The high level structure of the thesis software for managing time series is described in this section. Figure 4.1 illustrates the stages that a time series goes through in the thesis software. It consists of four main stages namely: Data Input, Transform, Analyze and Visualize. The time series data advances through the stages in a sequential order as shown in the figure. Each stage is a set of processes which are illustrated in more details in fig 4.2.

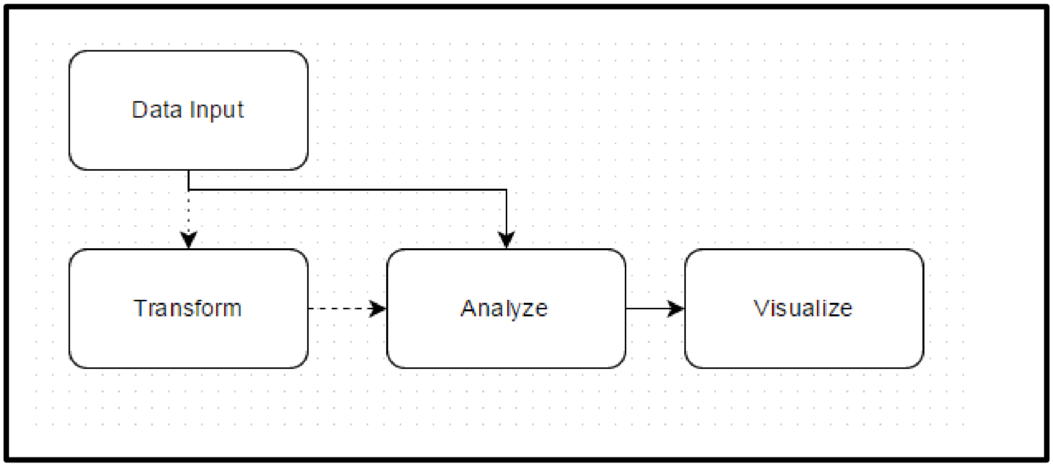


Figure 4.1: Stages of a time series in the thesis software

As can be seen in figure 4.2, the management of time series follows an intuitive workflow which simplifies the use of the software. Users advance through the system by first entering data into the system. Depending on the type of data entered, the user determines whether data will be transformed from time-stamped data input to time series, and then instructs the system accordingly. If data was transformed to time series, the user is presented with the time series output for verification. The transformation phase is skipped if input data is already in the right format of time series. Users can then perform analysis and forecasting of time series data.

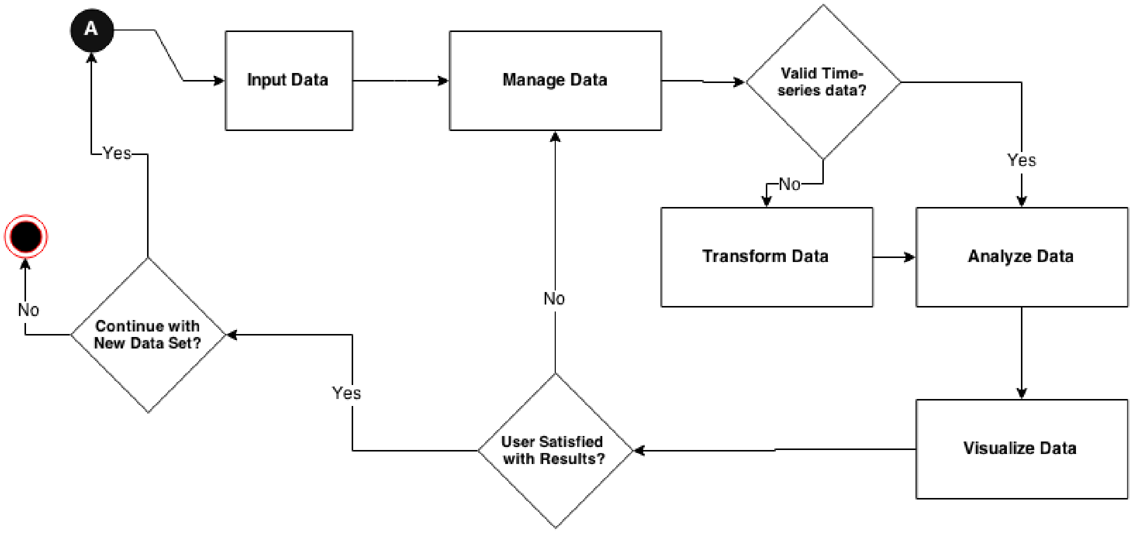


Figure 4.2: Time series Data Management – Activity Diagram

Table 4.2 shows the functional requirements provided by the software for managing time series data. The table also shows the existing tools or research from which the requirements are sourced.

Table 4.2: Requirements for managing time series

Functional Requirement	Source(s)
1. Create time series	• SAS/ETS (SAS Institute Inc., SAS/ETS 9.1 Users Guide, 2004)
2. Convert irregularly spaced data to equally spaced data and filling-in missing values.	• SAS/ETS (SAS Institute Inc., SAS/ETS 9.1 Users Guide, 2004) • (GMDH Shell LLC., 2013) • MATLAB
3. Read time series data recorded in different ways	• SAS/ETS (SAS Institute Inc., SAS/ETS 9.1 Users Guide, 2004)
4. Edit time series data from a spreadsheet interface	• (GMDH Shell LLC., 2013) • GRETL (Cottrell & Lucchetti, 2015)
5. Import and export time series in the following formats: xls, .xlsx, .txt, .csv.	• (Baum, 2003) • DTREG (DTREG, Time Series Analysis, 2010) • (STATA, 2003)

These requirements cut across the entire phases of the software operations: from the point time series is entered to the system to time it is analyzed and visualized. The software provides good support for data management. Datasets may be created and modified with a built–in spreadsheet data editor in the GUI. The software provides a comprehensive set of data–handling functions for both numeric and string variables with sophisticated parsing facilities for the latter. Users can work with data defined on different time–series frequencies. The software also has good integration with spreadsheet applications like Microsoft excel. Data can be imported and exported in different file formats.

Table 4.3 shows the functional requirements and the corresponding software features relating to the management of time series. The respective technical features identified in this section are meant to enable the software deliver effectively on the requirements for managing time series. For instance, for the software to support the import of time series data, it must have support for reading file formats such as .csv, .xls, .xslx and txt.

Table 4.3: Technical Features for managing time series

Functional Requirement	Supporting Technical Features
1. Create time series **2. Edit time series data from a spreadsheet interface** **3. Convert irregularly spaced data to equally spaced data and filling-in missing values.**	• Spreadsheet interface for data representation • Support for some spreadsheet functionalities such as: headers, formulas, text formatting, sorting and filtering. • Support for text editing functionalities. • Ability to auto-detect and describe of data characteristics e.g. data types, date field etc. • Users can add results from an external file (e.g. .xlsx) file to an active file. • Error Handling: e.g. Missing value imputation and Outlier detection.
4. Read time series data recorded in different ways **5. Import and export time series in the following formats: xls, .xlsx, .txt, .csv.**	• Support for CSV/XLS/XLSX and ODBC/OLEDB connections • Support for data import and export in the following formats: .csv, .xls, .xslx, .txt • Ability to detect and report time series aspects of a dataset or estimation sample. • Support for intelligent input feature i.e. software automatically detects the bounds of input data, header information, and the row/column orientation of the data. • Fill in gaps in time variable e.g. Auto fill for days, months, years where a trend is detected • Ability to convert irregularly spaced data to equally spaced data and filling-in missing values

4.3 Time series Transformation

One of the key features of the thesis software is the transformation of time-stamped data to time series. It has been discussed in previous sections that time-stamped data are often too cumbersome and disorganized for businesses to analyze and use for further decision making tasks. Usually, these time-stamped data must be converted to time series data before further analysis and forecasting can be done. Unfortunately, not many time series tools offer this functionality as already shown in Chapter two.

The process of converting time –stamped data to time series often involves the association of user-defined time series frequencies (e.g. Yearly, Quarterly, Monthly, Weekly, Daily, Hourly, or every minute) to the time-stamped data. The MapReduce technique has been used in this thesis for this process. Thereafter, a time series model of choice can be applied to perform further time series analysis and forecasting on the transformed data. Refer to Chapter two for a detailed description of MapReduce.

In figure 4.3, the process flow of transforming time-stamped data to time series with the thesis software is illustrated. [*Note that this is a generic process description; a specific use case is discussed in Chapter* five]. The process of performing time series transformation is designed such that users only give few inputs to the software which are basically (a) the input data and (b) the time frequency (or level of aggregation). The process uses an underlying MapReduce technique for the transformation.

At first, the user imports some time-stamped data to be transformed and further analyzed. The system then presents user with an interface where the time series frequency is set. The next phase involves an implicit application of the MapReduce algorithm to assign time frequencies and aggregate the data accordingly. The transformed data is then presented to the user for verification.

The functional requirements for transforming time series data are shown in table 4.4 below.

Table 4.4: Requirements for Transforming Time series

Functional Requirement	**Source(s)**
1. Convert time series data from one frequency to another (such as from weekly to monthly or vice versa). **2. Generating tabular reports for viewing the created time series and adjusting them for correctness.**	• SAS/ETS (SAS Institute Inc., SAS/ETS 9.1 Users Guide, 2004) • GMDH Shell (GMDH Shell LLC., 2013) • GRETL (Cottrell & Lucchetti, 2015)
3. Convert time-stamped data to time series **4. Represent text data with numeric values. If user inputs categorical variables with data values such as “Male”, “Female”, “Married”, “Single”, etc., there is no need for users to code them as numeric values.**	• Thesis Objectives

Table 4.5 lists the requirements and technical features of the thesis software for transforming time series data. One notable feature is the support for MapReduce libraries. The software incorporates the capabilities and advantages of the MapReduce technique, which are discussed in chapter two.

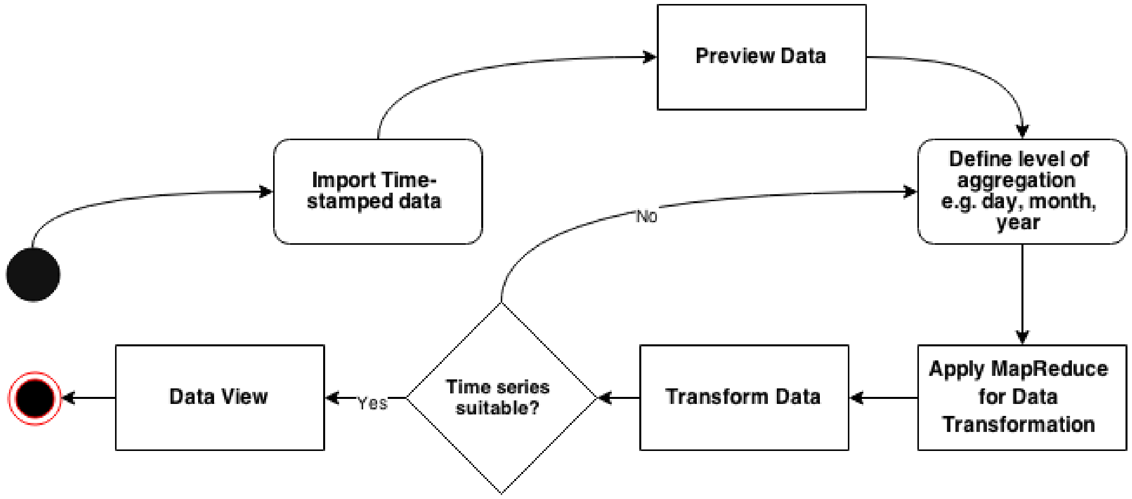

Preview Data
Import Time-stamped data
Define level of aggregation e.g. day, month, year
No
Apply MapReduce for Data Transformation
Transform Data
Time series suitable?
Yes
Data View

Figure 4.3: MapReduce Dataflow Diagram for Transforming Time-stamped Data to Time Series

Table 4.5: Technical Features for Transforming Time series

Functional Requirement	Supporting Technical Features
1. Convert time-stamped data to time series **2. Convert time series data from one frequency to another (such as from weekly to monthly or vice versa).** **3. Represent text data with numeric values. If user inputs categorical variables with data values such as “Male”, “Female”, “Married”, “Single”, etc., there is no need for users to code them as numeric values.** **4. Generating tabular reports for viewing the created time series and adjusting them for correctness.**	• Support for MapReduce libraries • Support for a wide range of time series frequencies: Yearly, Quarterly, Monthly, Weekly, Daily, Hourly and every minute. • Support for aggregation functions such as: Sum, Count, Average, Min and Max. • Spreadsheet interface for data representation • Support for data import and export in the following formats: .csv, .xls, .xslx, .txt • Ability to auto-detect and describe of data characteristics e.g. data types, date field etc.

4.4 Time series Analysis

The thesis software provides users with simplified processes for analyzing time series data. The software has good capabilities for time series analysis. End users can easily analyze univariate and multivariate time series, following the process illustrated in figure 4.4.

The analysis phase of the time series comes after the data is entered and properly structured. Analysis is done implicitly in three simple steps:

1. Automatic model selection
2. Confirmation/Change of preferred model
3. Selection of visualization type

The system supports the automatic model selection using system-defined model selection criteria (details are discussed in Chapter Two). The output of this step is a time series model which is suggested to the end user for the analysis. The user can accept or change the model (if required) in the second step. Once a model is selected, the next step is to select the visualization type and to edit the visualization variables such as axes, range of values, analysis variable (i.e. variable to be analyzed) e.t.c.

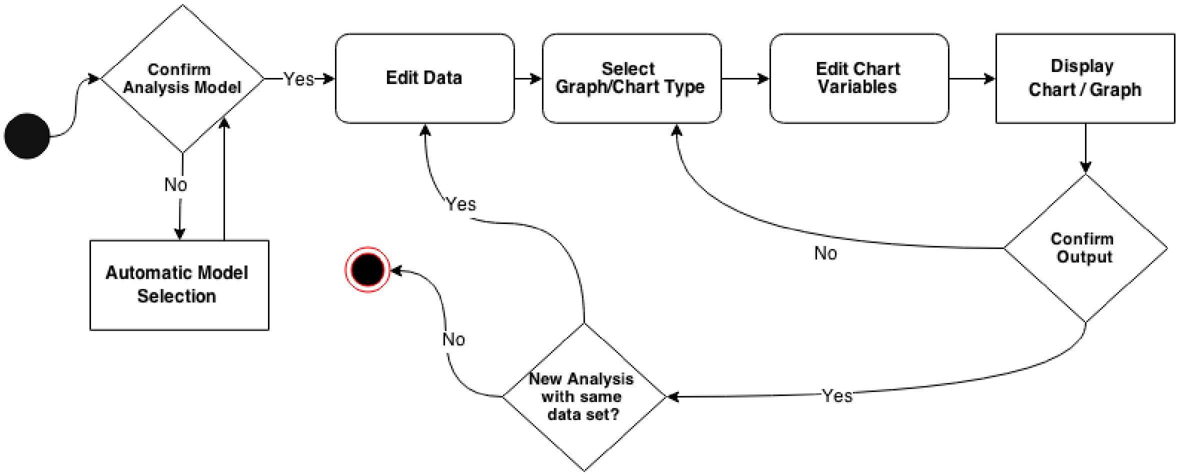


Figure 4.4: Time Series Analysis – Activity Diagram

The software is designed to support the major time series methods for analysis and forecasting. The specific sets of sophisticated time–series methods to be supported are based on the capabilities provided by other time series software like STATA, GMDH Shell and GRETL. These include: Box–Jenkins models; ARIMA, including ARMA–X models; a number of ARCH models; vector auto-regressions (VARs) and structural VARs, impulse response functions (IRFs); user–specified nonlinear least squares and maximum likelihood estimation capabilities; univariate regression models (Baum, 2003).

The main functional requirements of the thesis software for time series analysis and their sources are shown in table 4.6.

Table 4.6: Requirements for Analyzing Time series

Functional Requirement	**Source(s)**
1. Analyzing time series and generating a model showing how best to predict future time series values **2. Automatic selection of the analysis model which fits best to the time series to be analyzed** **3. Analyzing univariate and multivariate time series.**	• SAS/ETS (SAS Institute Inc., SAS/ETS 9.1 Users Guide, 2004) • (Baum, 2003) • GMDH Shell (GMDH Shell LLC., 2013) • GRETL (Cottrell & Lucchetti, 2015) • MATLAB • R
4. Forecasting future time series values **5. Possibility to use a wide variety of analysis and forecasting methods and models.**	• MATLAB • DTREG (DTREG, Time Series Analysis, 2010) • R
6. Guaranteeing that outputs of time series analysis are properly verified for precision	• DTREG (DTREG, Time Series Analysis, 2010) • R • STATA (STATA, 2003)

Table 4.7 lists corresponding technical features of the thesis software to satisfy these requirements for analyzing time series data.

Table 4.7: Technical Features for Analyzing Time series

Functional Requirement	**Supporting Technical Features**
1. Automatic selection of the analysis model which fits best to the time series to be analyzed **2. Analyzing time series and generating a model showing how best to predict future time series values** **3. Analyzing univariate and multivariate time series.**	• Automatic trial of all models on data inputs and choice of the one that best fits the input data. • Support for all main analysis and prediction Models e.g. ARIMA, ARMAX, Box–Jenkins models, ARCH models; vector auto-regressions (VARs) and structural VARs, impulse response functions (IRFs)and other dynamic regression models • Provision of diagnostic tools like a number of unit root tests, several frequency domain measures, tests for white noise and ARCH effects, for univariate time series • Support for moving average and nonlinear filters • Standard VAR estimation, structural VAR estimation, and the generation of diagnostic tests for multivariate time series • Clearing current model from memory / replace model
4. Forecasting future time series values **5. Possibility to use a wide variety of analysis and forecasting methods and models.** **6. Guaranteeing that outputs of time series analysis are properly verified for precision**	• Check the stability condition and correctness of time series model estimates • Support for Time Series smoothers and filter models e.g. Baxter–King time series filter, Butterworth time series filter, Christiano–Fitzgerald time series filter, Hodrick–Prescott time series filter • Post-estimation tools to estimate autocorrelations and autocovariances and to check stability condition of estimates • Econometric model forecasting • Support for conducting risk analysis seamlessly along with time series forecasting e.g. integration with Monte Carlo simulation • Support software libraries of time series estimators, diagnostic tools, and smoothers and filters • Support for calculating ancillary measures like fitted values, residuals, standardized residuals, studentized (jackknifed) residuals, standard errors of prediction, forecast and residual, and leverage estimates • Support for dynamic forecasts, forecast–error variance decompositions and impulse response functions in point and interval form • Adding estimation results to an existing forecast model

4.5 Time series Visualization

Most data are meaningless— unless visualized. Stepping beyond familiar visualizations like bar charts and pie charts, there are many approaches to visualizing data, from mapping (e.g., color coding a map to show patterns) to visualizing networks (e.g., the links time series events). Consequently, a variety of techniques have been designed published for visualizing data. This variety however makes it difficult for prospective users to select methods or tools that are useful for visualizing specific data (Aignera, Mikscha, Müllerb, Schumannc, & Tominski, 2007).

One major problem often confronted in visualizing data is the size of the dataset. Large datasets are often difficult to visualize with standard techniques given the limitations of current display devices. One major effective technique discussed in (Hao, Dayal, Keim, & Schreck, 2007) is a framework for intelligent time- and data-dependent visual aggregation of data along multiple resolution levels. The basic idea of the technique is that either data-dependent or application-dependent, display space is allocated in proportion to the degree of interest of data subintervals, thereby (a) guiding the user in perceiving important information, and (b) freeing required display space to visualize all the data. The framework can also accommodate any time series analysis algorithm yielding a numeric degree of interest scale. The technique has been applied to real-world data sets, compared with the standard visualization approach, and was found useful, scalable and appropriate for end user oriented designs (Hao, Dayal, Keim, & Schreck, 2007).

The problem of effectively visualizing (large) time series data sets was solved in this thesis time series software, by inculcating the properties of intelligent multiple resolution visualization technique to the requirements. The thesis software is designed to provide effective visualization support for long time-series data providing both focus and context. The software provides customizable charts, and plots of various sorts. Graphics files are produced in a native format and may be exported to formats like Portable Network Graphics, GIF, JPEG, PostScript, Encapsulated PostScript, Windows Metafile, Windows Enhanced Metafile, PDF, and PICT, depending on the user's needs.

Listed below are the main requirements for time series visualization. These requirements have been sourced mainly from GMDH Shell (GMDH Shell LLC., 2013) and GRETL (Cottrell & Lucchetti, 2015) since they offer good visualizations of time series data.

1. Standardized time series visualization
2. Viewable plots of the data, predicted versus actual values, prediction errors, and forecasts with confidence limits.
3. Adjustable time series and graphical representation.
4. Support for printing system output including spreadsheets and graphs.
5. Creating tabular reports such as balance sheets, and other row and column reports for viewing outputs of time series analysis.
6. Scalable to large data sets

Table 4.8 lists corresponding technical features of the thesis software to satisfy these requirements for visualizing time series data.

Table 4.8: Technical Features for Visualizing Time series

Functional Requirement	Supporting Technical Features
1. Standardized time series visualization **2. Viewable plots of the data, predicted versus actual values, prediction errors, and forecasts with confidence limits.** **3. Adjustable time series and graphical representation.** **4. Support for printing system output including spreadsheets and graphs.** **5. Creating tabular reports such as balance sheets, and other row and column reports for viewing outputs of time series analysis.** **6. Scalable to large data sets**	• Support for graphing tools to visualize data in the following chart types: line graphs, scatter plots, histograms, bar charts, bubble plots, area charts • Plot parametric autocorrelation and auto-covariance functions • Support for data & graphing diagnostic tools • Support for synchronous (client-side) visualization. Graph reloads as data input changes/is-edited or after some minutes • Report generation facility for creating tabular reports such as balance sheets, and other row and column reports for viewing outputs of time series analysis. • Front-end libraries for graphing and data visualizations e.g. arbor.js, cartodb, cubism.js, envision.js, google chart tools, etc. • Support identification of periodic structures in the data • Support for intelligent time- and data-dependent visual aggregation of data along multiple resolution visualization technique.

4.6 Overview of Software Requirements

This section gives a general overview of the functional requirements of the thesis software. The aim here is to clearly define and summarize all the functional requirements and set of capabilities of the thesis software. It derives from all the requirements discussed in the previous sections for managing, transforming, analyzing and visualizing time series data.

The requirements listed under the different categories are not exclusively specific to the categories e.g. a requirement for managing time series data may also be relevant for analyzing time series. Each category only reflects the requirements which are primary to its functionalities.

In Table 4.9, these functional requirements are listed (summarized to few words) and a tick (**X**) is placed to indicate all the categories where each requirement is relevant.

Table 4.9: Requirement-based comparison of time series tools

Functional Requirement / Software Feature	Time series management	Transforming time series	Time series analysis	Time series visualization
Create TS	X	X		
Auto-adjust TS	X	X		
Edit TS	X	X	X	X
Import and export TS	X	X		
Transform TS		X		
Frequency adjustment	X	X		
Quantify qualitative variables		X	X	
Report on Output		X	X	X
Analyze TS			X	
Automatic Model Selection			X	
Forecast TS			X	X
Variety of Models			X	X
Verify output	X	X	X	X
TS Visualization				X
Visualize multiple plots				X
Adjust graph				X
Print graph				X
Scalable plots				X

5 Use-cases and Mock Ups

The major contribution of this thesis is the design of an end-user oriented time series software. The software is designed based on the requirements and design specifications discussed in the previous chapter. Additionally, the tool incorporates the interactive and common features of existing time series software.

The thesis software is a time series transformation and analysis software designed for easy usage by end users. It is enriched with non-intimidating interfaces and sensible defaults for end users, and also provides possible configuration options for power users. The software is designed to support wizard-like interfaces to guide users through the analysis process. The extensive list of features (discussed in the previous chapter) can be divided into two categories:

- Data input and preparation
- Result analysis and reporting

As will be discussed in subsequent sections, data input and preparation is performed over three feature-rich screens: (1) Import/Input data, (2) describe data characteristics and (3) select models to run. The result analysis and reporting is performed using the visualization screens designed to display rich and clear interpretations of data.

This chapter describes the usage scenarios of the thesis software. It contains illustrations of the functionalities of the software with respect to managing, transforming, analyzing and visualizing time series. The usage-scenarios of these requirement categorizations are demonstrated using mockups to enable the reader have a visual understanding of the software design. An example of the analyses of rainfall average in Nigeria by a researcher in a public institution is illustrated to ensure proper understanding of the software.

5.1 Generic interface design of the thesis software

The interface of the software has been designed to intuitively enable end user interact with the system. As shown in figure 5.1, the interface is enriched with familiar and easy-to-use system elements to facilitate interactivity with the software. These include:

- **Top menu**: The top menu is arranged in the conventional way expected by users e.g. “File” menu comes first, followed by the “Edit” menu; and a menu like “Help” is positioned at the far right, following the usual convention that users are already used to.
- **Spreadsheet interface for data entry and manipulation**: Most users have good understanding of the tabular structure of data presentation used in spreadsheets. Thus the software design incorporates the spreadsheet interface. The spreadsheet can hold large sets of data and still present them in a clear manner. Furthermore, users can possibly work on multiple datasets by simply creating a new “tab” which creates another spreadsheet interface. This design also follows the conventions users are familiar with and makes multiple datasets easier to analyze and compare.
- **Text edit features**: Users can easily edit data using the edit features placed right above the spreadsheet. Conventional icons are used to represent each edit feature in order to simplify the understanding of their usage. Users who are familiar with text editing applications like WordPad, Microsoft Word etc. can easily interact with the interface.
- **Visualization pane**: The visualization pane is positioned below the spreadsheet to enable an overall presentation of the data and its visualizations in one view. This is a major advantage of the thesis software over many other existing applications. Often users need to click through many settings in order to activate the visualization screen, which is often presented in a separate window. See figure 5.1 for illustration.

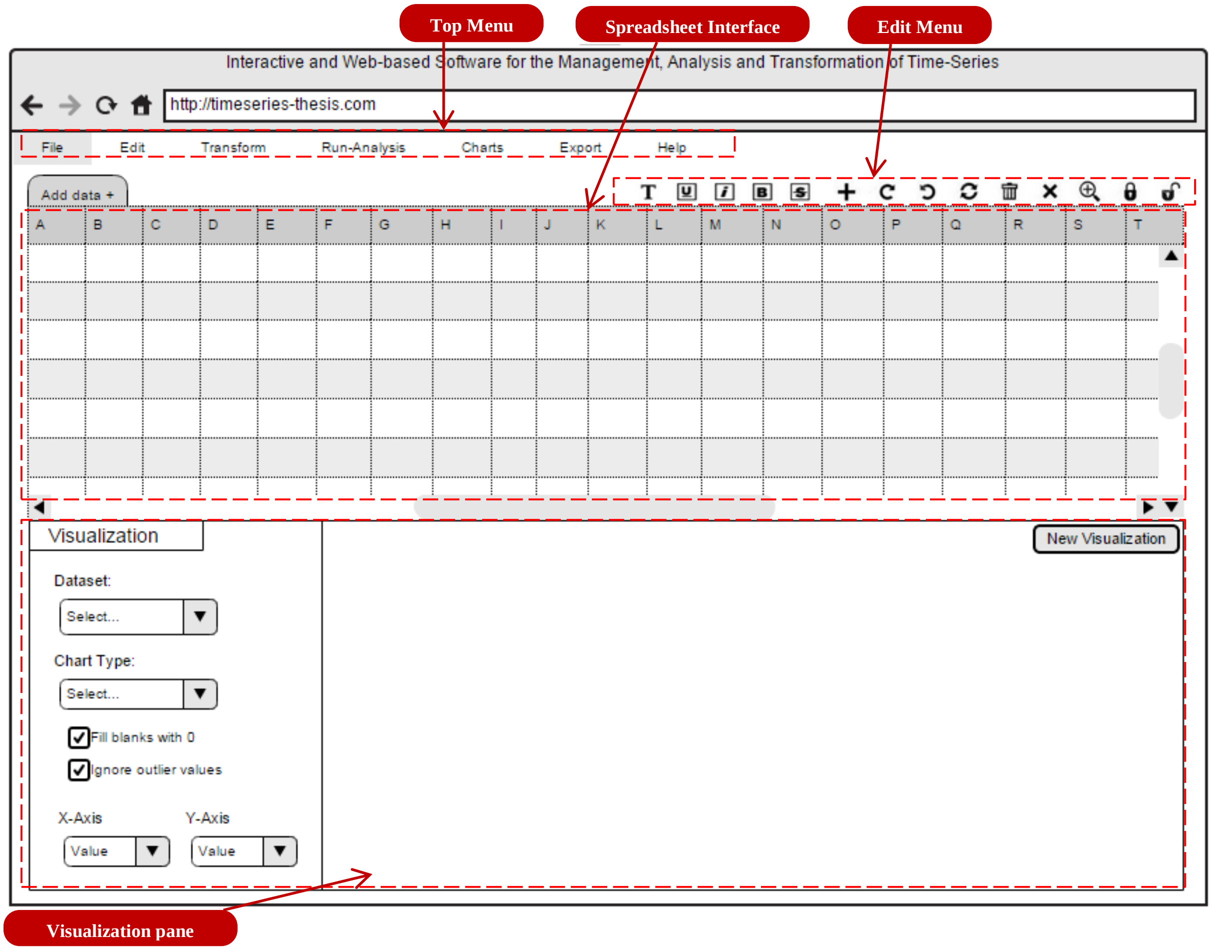


Figure 5.1: Generic interface design of the thesis software

5.2 Managing Time Series - Data Input Preparation

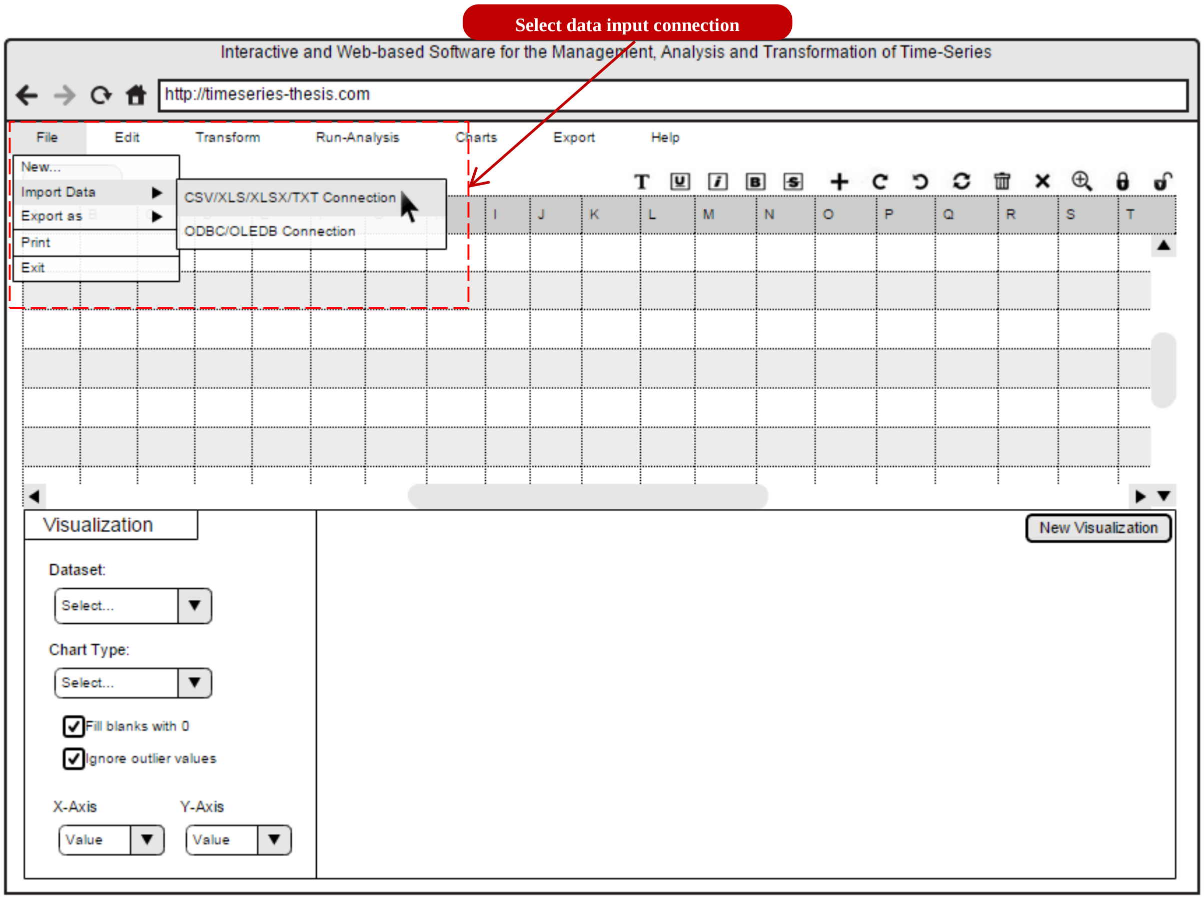
In order to illustrate the management of time series data in the thesis software, I illustrate here the wizard for data input and data preparation (for analysis). As mentioned earlier, this process is done in three steps which include:

- **Import/Input data**: This is the phase where the user inputs the time series data into the software. User can easily type the time series data directly on the spreadsheet, or copy it from another spreadsheet. However, user can import existing time series data stored in specific formats: .xls, .xlsx, .csv, and .txt. This is illustrated in fig 5.2. User also has the option to import data using the "ODBC/OLEDB" connection for data stored in other databases. Once the connection type is selected, user is presented with the typical browser window where the specific file is located (shown in fig 5.3). At this point, a couple of hidden data preparation actions are executed. Some of these are:

  - Intelligent data selection i.e. software identifies the complete range of data, orientation of the data, and the position of header and date ranges if existing.
  - Automatically identifies various types of periods like months in an year, dates, or quarters etc.
  - Fills in gaps (omitted values) and other discontinuous data issues (e.g., alternate rows or columns of data)

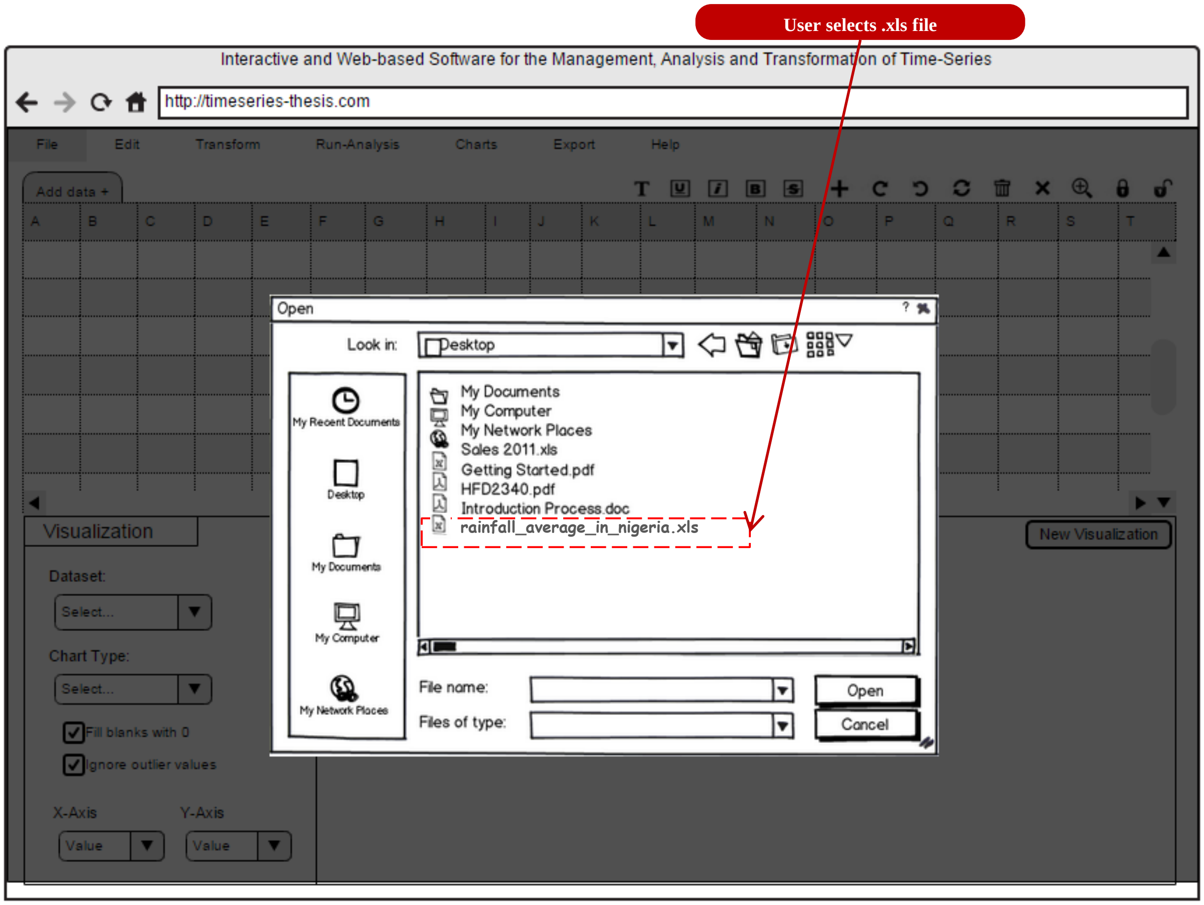
To illustrate this, let's consider a researcher in a government institution who is performing an analysis on rainfall average in Nigeria for a period of five months. As shown in fig 5.3, the researcher selects the time series data stored in .xls format.

- **Describe data characteristics**: After the time series data file (to be imported) is selected, user is presented with a screen to define the specific parts of the data to be analyzed. For instance, user can specify the specific row limits to be imported or define the date-time field if different from the system's automatic selection. Fig 5.4 shows the data from our example of the rainfall average in Nigeria. Data is presented to the researcher for verification.
- **Select models to run**: At this phase, user is presented a screen showing the time series models which can be used for analyzing the time series. This is discussed in details at the section for "time series analysis" in this chapter.



Select data input connection

Figure 5.2: Importing and preparing time series data



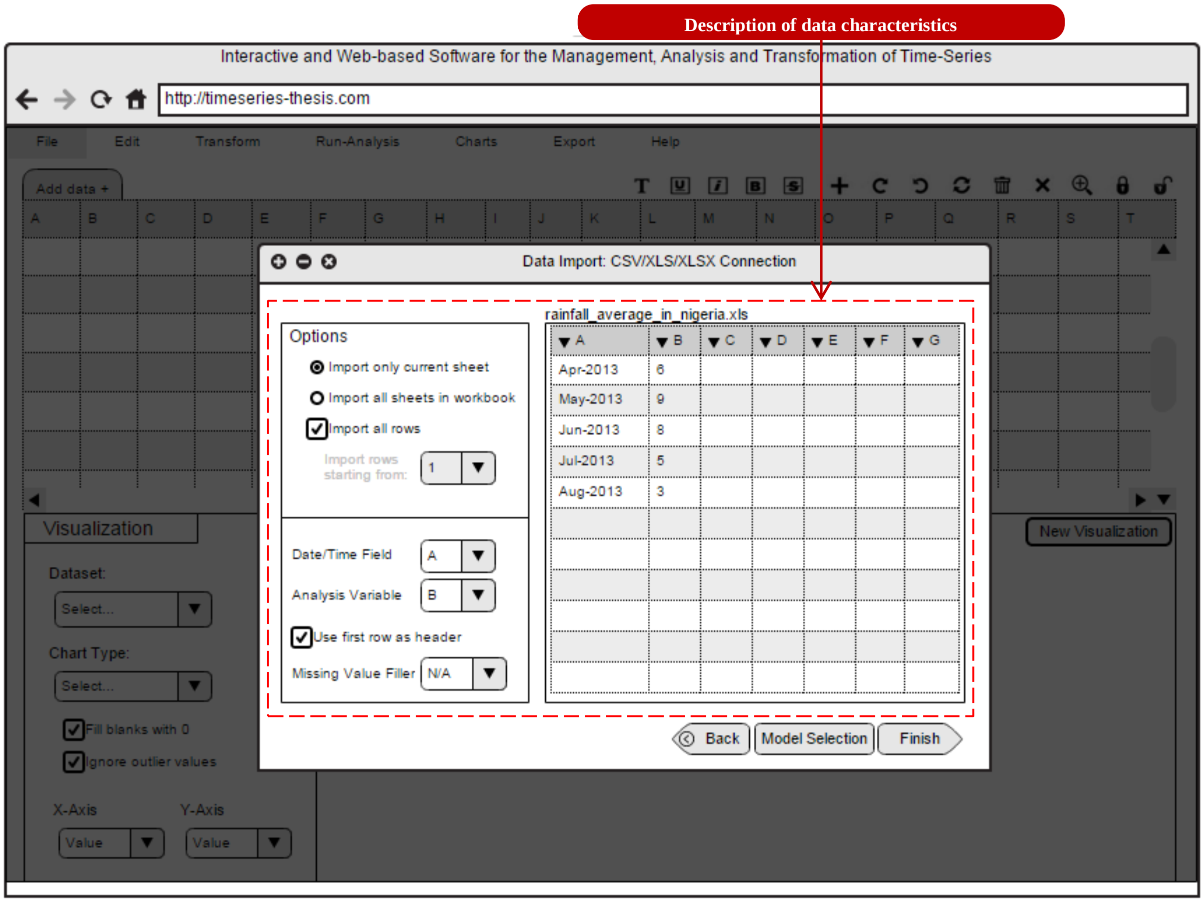
User selects .xls file

Interactive and Web-based Software for the Management, Analysis and Transformation of Time-Series

http://timeseries-thesis.com

Look in: Desktop

Figure 5.3: Browser window for importing and preparing time series data



Description of data characteristics
Interactive and Web-based Software for the Management, Analysis and Transformation of Time-Series
http://timeseries-thesis.com
File
Edit
Transform
Run-Analysis
Charts
Export
Help
Add data +
A B C D E F G H I J K L M N O P Q R S T
Data Import: CSV/XLS/XLSX Connection
Visualization
New Visualization
Back
Model Selection
Finish

Options

● Import only current sheet

○ Import all sheets in workbook

☑ Import all rows

Import rows starting from: 1

Date/Time Field A

Analysis Variable B

☑ Use first row as header

Missing Value Filler N/A

rainfall_average_in_nigeria.xls

▼ A	▼ B	▼ C	▼ D	▼ E	▼ F	▼ G
Apr-2013	6					
May-2013	9					
Jun-2013	8					
Jul-2013	5					
Aug-2013	3					

Dataset: Select...

Chart Type: Select...

☑ Fill blanks with 0

☑ Ignore outlier values

X-Axis Value

Y-Axis Value

Figure 5.4: Description of data characteristics

5.3 Transforming Time Series using MapReduce

The transformation of time series, either from time stamped data to time series or from one frequency level to another (e.g. weekly time series to monthly time series) is a key feature of the thesis software. This feature is designed using the MapReduce method, explained in the previous chapters. To illustrate this, let's take an example of a log data from a computer such as:

```
[Sun Mar  7 16:02:00 2004] [notice]
[Sun Mar  7 16:02:00 2004] [info]
[Sun Mar  7 16:02:00 2004] [notice]
[Sun Mar  7 16:05:49 2004] [info]
[Sun Mar  7 23:42:44 2004] [notice]
[Mon Mar  8 00:11:22 2004] [info]
[Mon Mar  8 00:32:45 2004] [info]
[Mon Mar  8 00:40:10 2004] [info]
[Mon Mar  8 07:54:30 2004] [notice]
[Mon Mar  8 08:14:15 2004] [info]
[Mon Mar  8 14:54:56 2004] [info]
[Tue Mar  9 13:49:05 2004] [info]
[Tue Mar  9 08:15:21 2004] [info]
[Tue Mar  9 08:37:23 2004] [info]
[Wed Mar 10 11:45:51 2004] [info]
```

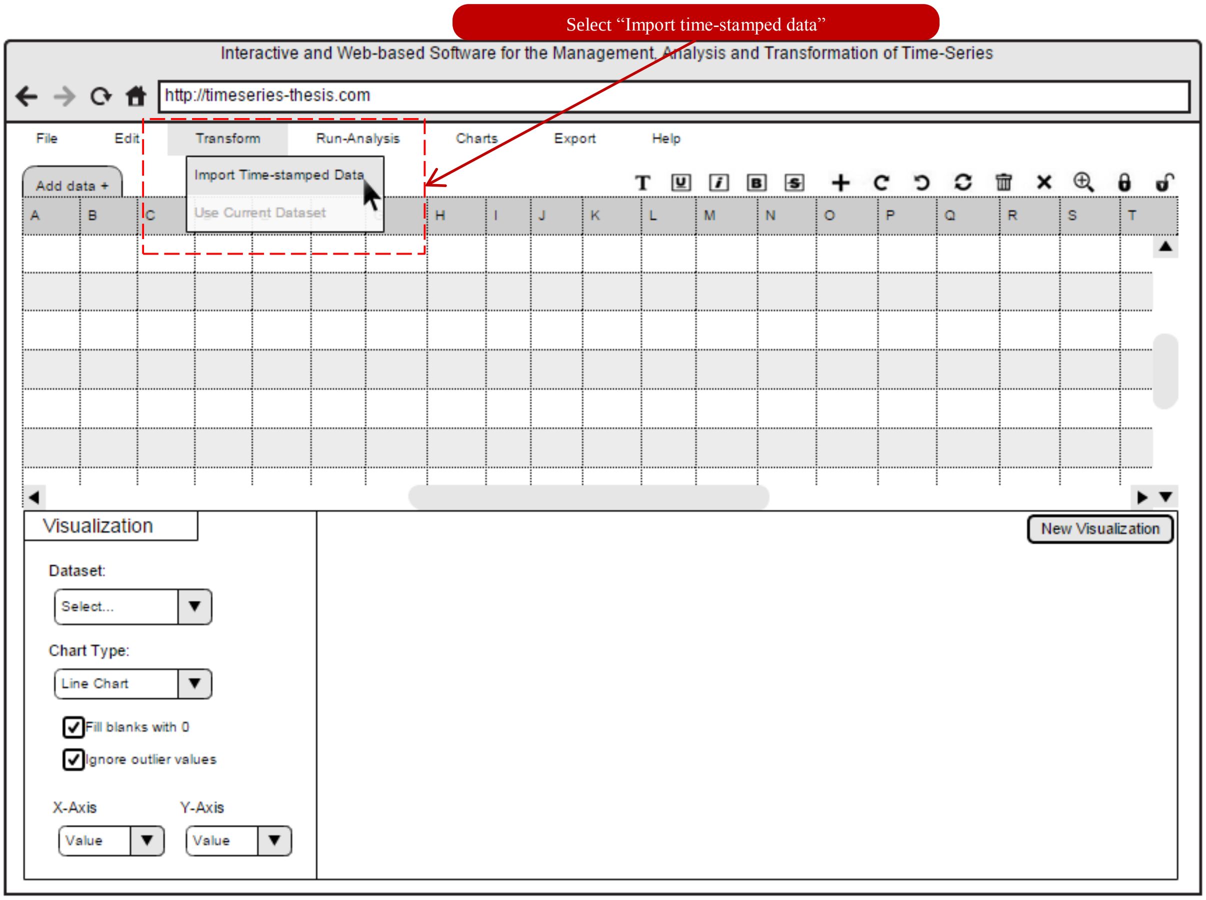
The process of transforming time series is completed in three simple steps:

- **Select and preview time-stamped data**: As illustrated in figure 5.5, user initiates the transformation process by selecting "Import time-stamped data". User then selects a file containing the data e.g. log data/file shown above. The software displays a preview of the time-stamped data to the user as shown in figure 5.6. User then verifies if data is correctly loaded.
- **Define reduction criteria**: At this phase, user can set a couple of parameters for the MapReduce. These include: time-stamp field, variable fields, reduction/aggregation function (e.g sum, count, min, max or average) and the resolution/frequency which can be per minute, weekly, monthly, quarterly and yearly. See figure 5.6 for illustration.
- **Apply MapReduce**: The transformation of data to time series is completed using the MapReduce, illustrated in figure 5.7. The MapReduce itself is in five phases described as thus:
    - *Input*: This refers to the data input to the MapReduce, i.e. the log file in this example.
    - *Splitting*: The splitting phase enables the technique to scale to large datasets. Input data is divided into chunks of data to enable parallel processing of the datasets.
    - *Mapping*: The Map function takes a series of key/value pairs, processes each,

and generates zero or more output key/value pairs. The input and output types of the map can be (and are often) different from each other. In our example: the *key* is the date e.g. "`Sun Mar  7`" and the *value* is the string: `{"notice", 1}.` The output (sample) of the mapping phase will be:

```
<"Sun Mar  7", {"notice", 1}>
<"Sun Mar  7", {"info", 1}>
<"Sun Mar  7", {"notice", 1}>
<"Sun Mar  7", {"info", 1}>
<"Sun Mar  7", {"notice", 1}>
```

- *Sorting*: After mapping phase, the sorting or shuffling of the data takes place using the *key*. The major reason is the availability of various servers or nodes for processing large datasets. This phase is less relevant for this thesis since our focus is not on processing big data.
- *Reduction*: Now comes the reduction phase, which accepts the data coming from the sorting/shuffling phase and combines the data into a smaller set of values using the reduce function set by the user (see figure 5.6). The reduce function takes the input values, sums them and generates a single and the final sum. This reduced data is then presented to the user for verification (shown in figure 5.8).



Select “Import time-stamped data”

Interactive and Web-based Software for the Management, Analysis and Transformation of Time-Series

http://timeseries-thesis.com

Dataset: Select...

Chart Type: Line Chart

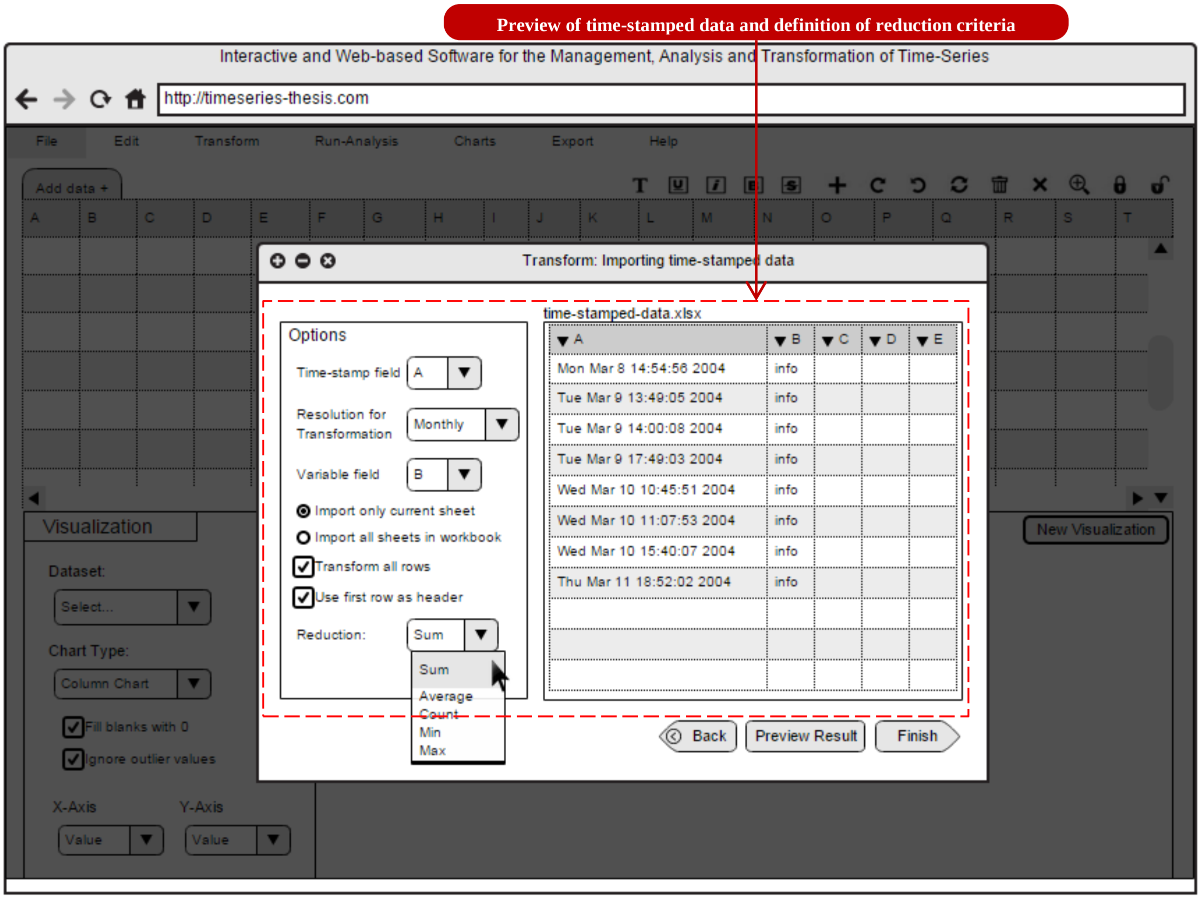
☑ Fill blanks with 0

☑ Ignore outlier values

X-Axis Value

Y-Axis Value

Figure 5.5: Selecting a time-stamped data



Preview of time-stamped data and definition of reduction criteria

Interactive and Web-based Software for the Management, Analysis and Transformation of Time-Series

http://timeseries-thesis.com

Transform: Importing time-stamped data

Options

Time-stamp field: A

Resolution for Transformation: Monthly

Variable field: B

● Import only current sheet

○ Import all sheets in workbook

☑ Transform all rows

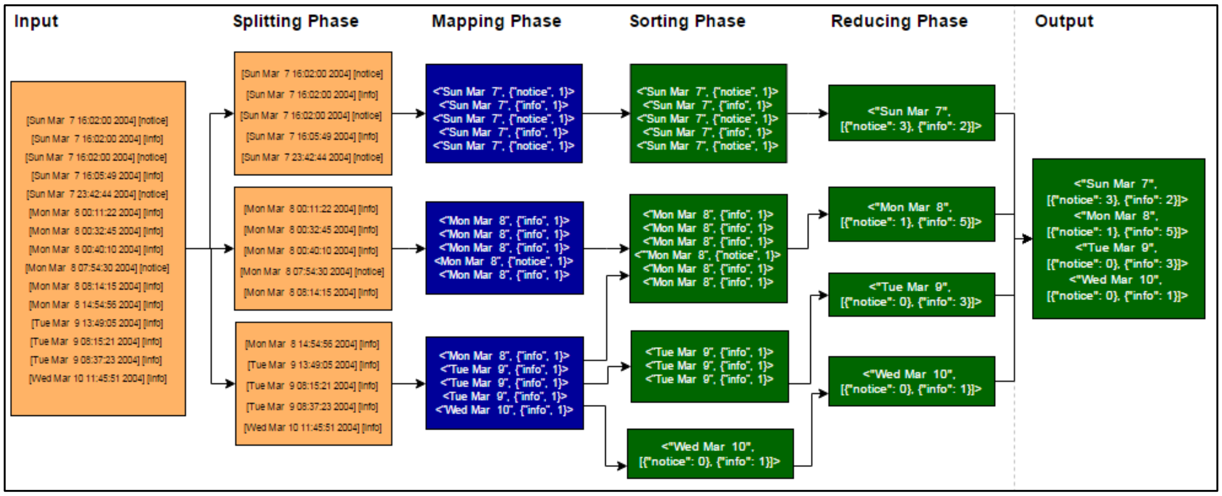
☑ Use first row as header

Reduction: Sum

time-stamped-data.xlsx

A	B	C	D	E
Mon Mar 8 14:54:56 2004	info			
Tue Mar 9 13:49:05 2004	info			
Tue Mar 9 14:00:08 2004	info			
Tue Mar 9 17:49:03 2004	info			
Wed Mar 10 10:45:51 2004	info			
Wed Mar 10 11:07:53 2004	info			
Wed Mar 10 15:40:07 2004	info			
Thu Mar 11 18:52:02 2004	info			

Figure 5.6: Previewing the time-stamped data



[Sun Mar 7 16:02:00 2004] [notice]
[Sun Mar 7 16:02:00 2004] [info]
[Sun Mar 7 16:02:00 2004] [notice]
[Sun Mar 7 16:05:49 2004] [info]
[Sun Mar 7 23:42:44 2004] [notice]
[Mon Mar 8 00:11:22 2004] [info]
[Mon Mar 8 00:32:45 2004] [info]
[Mon Mar 8 00:40:10 2004] [info]
[Mon Mar 8 07:54:30 2004] [notice]
[Mon Mar 8 08:14:15 2004] [info]
[Mon Mar 8 14:54:56 2004] [info]
[Tue Mar 9 13:49:05 2004] [info]
[Tue Mar 9 08:15:21 2004] [info]
[Tue Mar 9 08:37:23 2004] [info]
[Wed Mar 10 11:45:51 2004] [info]

[Sun Mar 7 16:02:00 2004] [notice]
[Sun Mar 7 16:02:00 2004] [info]
[Sun Mar 7 16:02:00 2004] [notice]
[Sun Mar 7 16:05:49 2004] [info]
[Sun Mar 7 23:42:44 2004] [notice]

[Mon Mar 8 00:11:22 2004] [info]
[Mon Mar 8 00:32:45 2004] [info]
[Mon Mar 8 00:40:10 2004] [info]
[Mon Mar 8 07:54:30 2004] [notice]
[Mon Mar 8 08:14:15 2004] [info]

[Mon Mar 8 14:54:56 2004] [info]
[Tue Mar 9 13:49:05 2004] [info]
[Tue Mar 9 08:15:21 2004] [info]
[Tue Mar 9 08:37:23 2004] [info]
[Wed Mar 10 11:45:51 2004] [info]

<"Sun Mar 7", {"notice", 1}>
<"Sun Mar 7", {"info", 1}>
<"Sun Mar 7", {"notice", 1}>
<"Sun Mar 7", {"info", 1}>
<"Sun Mar 7", {"notice", 1}>

<"Mon Mar 8", {"info", 1}>
<"Mon Mar 8", {"info", 1}>
<"Mon Mar 8", {"info", 1}>
<Mon Mar 8", {"notice", 1}>
<"Mon Mar 8", {"info", 1}>

<"Mon Mar 8", {"info", 1}>
<"Tue Mar 9", {"info", 1}>
<"Tue Mar 9", {"info", 1}>
<Tue Mar 9", {"info", 1}>
<"Wed Mar 10", {"info", 1}>

<"Sun Mar 7", {"notice", 1}>
<"Sun Mar 7", {"info", 1}>
<"Sun Mar 7", {"notice", 1}>
<"Sun Mar 7", {"info", 1}>
<"Sun Mar 7", {"notice", 1}>

<"Mon Mar 8", {"info", 1}>
<"Mon Mar 8", {"info", 1}>
<"Mon Mar 8", {"info", 1}>
<""Mon Mar 8", {"notice", 1}>
<"Mon Mar 8", {"info", 1}>
<"Mon Mar 8", {"info", 1}>

<"Tue Mar 9", {"info", 1}>
<"Tue Mar 9", {"info", 1}>
<"Tue Mar 9", {"info", 1}>

<"Wed Mar 10",
[{"notice": 0}, {"info": 1}]>

<"Sun Mar 7",
[{"notice": 3}, {"info": 2}]>

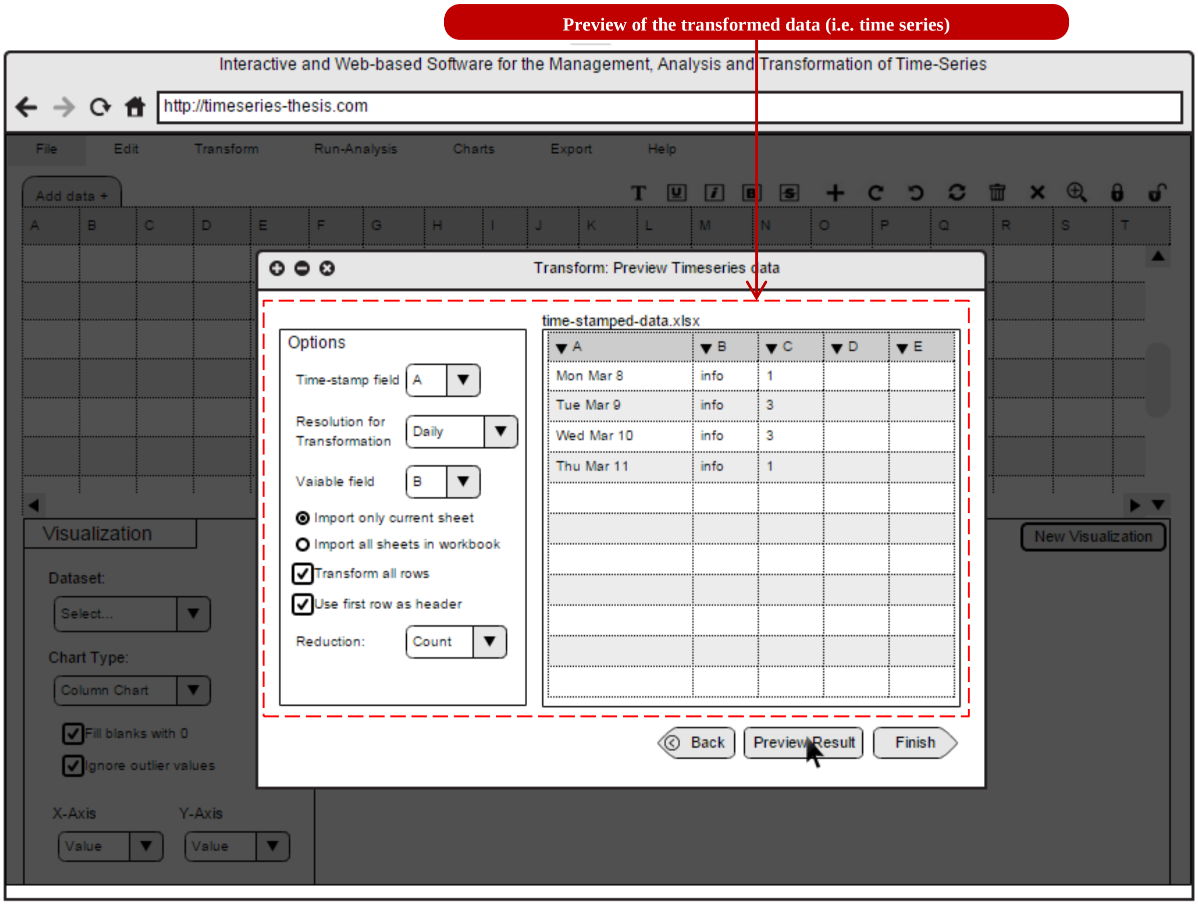
<"Mon Mar 8",
[{"notice": 1}, {"info": 5}]>

<"Tue Mar 9",
[{"notice": 0}, {"info": 3}]>

<"Wed Mar 10",
[{"notice": 0}, {"info": 1}]>

<"Sun Mar 7",
[{"notice": 3}, {"info": 2}]>
<"Mon Mar 8",
[{"notice": 1}, {"info": 5}]>
<"Tue Mar 9",
[{"notice": 0}, {"info": 3}]>
<"Wed Mar 10",
[{"notice": 0}, {"info": 1}]>

Figure 5.7: Transforming time-stamped data to time series using MapReduce



Preview of the transformed data (i.e. time series)

Interactive and Web-based Software for the Management, Analysis and Transformation of Time-Series

http://timeseries-thesis.com

Transform: Preview Timeseries data

Options

Time-stamp field A

Resolution for Transformation Daily

Vaiable field B

● Import only current sheet

○ Import all sheets in workbook

☑ Transform all rows

☑ Use first row as header

Reduction: Count

time-stamped-data.xlsx

A	B	C	D	E
Mon Mar 8	info	1		
Tue Mar 9	info	3		
Wed Mar 10	info	3		
Thu Mar 11	info	1		

Visualization

Dataset: Select...

Chart Type: Column Chart

☑ Fill blanks with 0

☑ Ignore outlier values

X-Axis Value

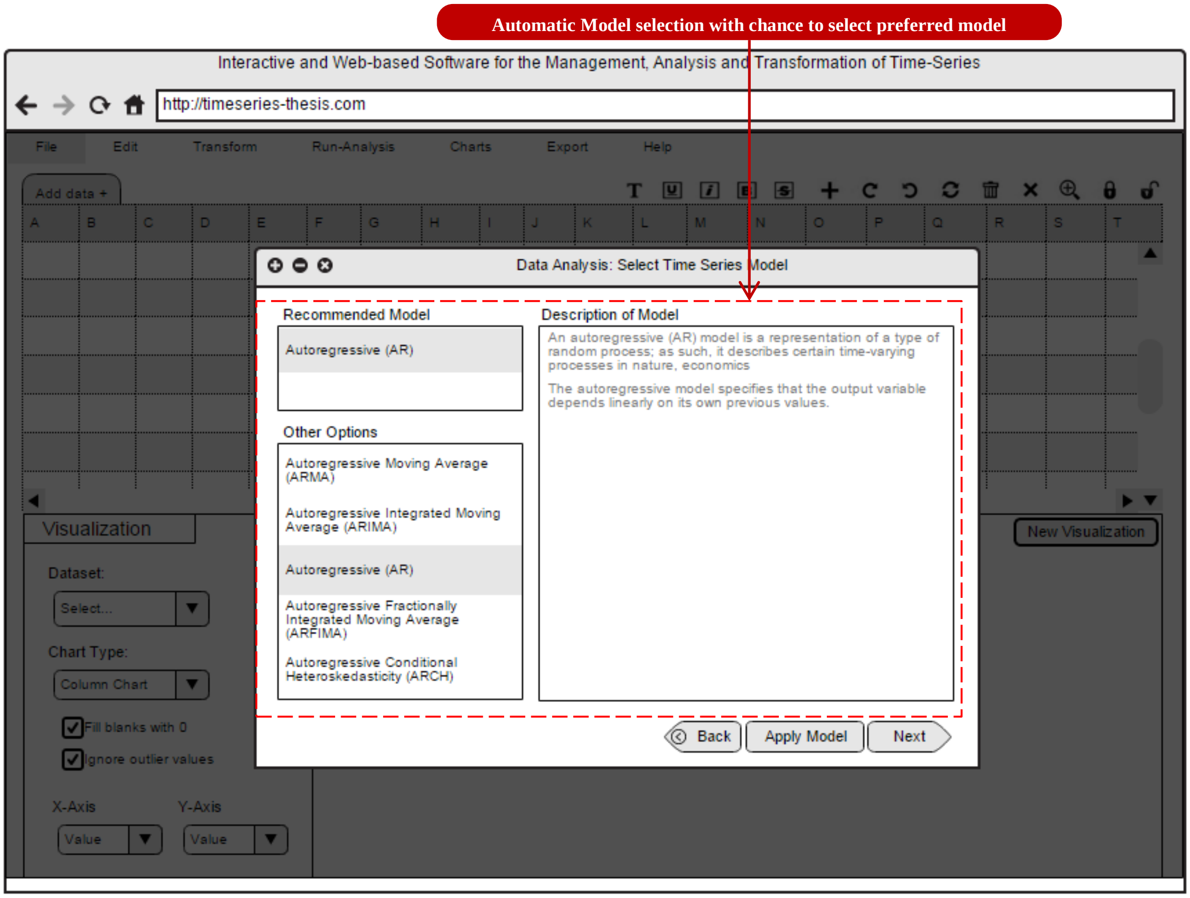
Y-Axis Value

Figure 5.8: Preview of transformed data (i.e. time series)

5.4 Analyzing Time Series

The process of analyzing time series data is simplified and designed for end users who have little or no knowledge about time series analysis and forecasting models. The software supports the automatic model selection using predefined selection criteria (discussed in chapter two). Thus the software automatically selects a best-fit time series analysis model based on predefined criteria. The goal of each model is to identify the underlying trend or pattern of the data and separate it from the "noise" (Raychaudhuri, 2010).

Analysis is done basically by applying the recommended model to the time series or by selecting another model depending on user's preference. This is illustrated in figure 5.9. The user however has the option of selecting from a list of other time series models. This is perhaps useful for power users who may at times prefer to use a particular analysis model other than the software's recommendation, possibly to achieve specific analysis objectives.



Automatic Model selection with chance to select preferred model

Interactive and Web-based Software for the Management, Analysis and Transformation of Time-Series

http://timeseries-thesis.com

Data Analysis: Select Time Series Model

Recommended Model: Autoregressive (AR)

Other Options:

- Autoregressive Moving Average (ARMA)
- Autoregressive Integrated Moving Average (ARIMA)
- Autoregressive (AR)
- Autoregressive Fractionally Integrated Moving Average (ARFIMA)
- Autoregressive Conditional Heteroskedasticity (ARCH)

Description of Model:

An autoregressive (AR) model is a representation of a type of random process; as such, it describes certain time-varying processes in nature, economics

The autoregressive model specifies that the output variable depends linearly on its own previous values.

Dataset: Select...

Chart Type: Column Chart

☑ Fill blanks with 0

☑ Ignore outlier values

X-Axis

Value

Y-Axis

Value

Figure 5.9: Selecting time series model for analysis

5.5 Visualizing time series

Perhaps the most important part of the software design (from an end-user perspective) is the visualization screen. Visualizations look better and attract more attention than the textual formats. Thus, the software design incorporates good visualizations of data which are:

- *Clear*: Users can easily understand graphs than numbers. Using visual presentation of numbers dramatically reduces confusion because the user does not need to process the numbers to be able to interpret them correctly.

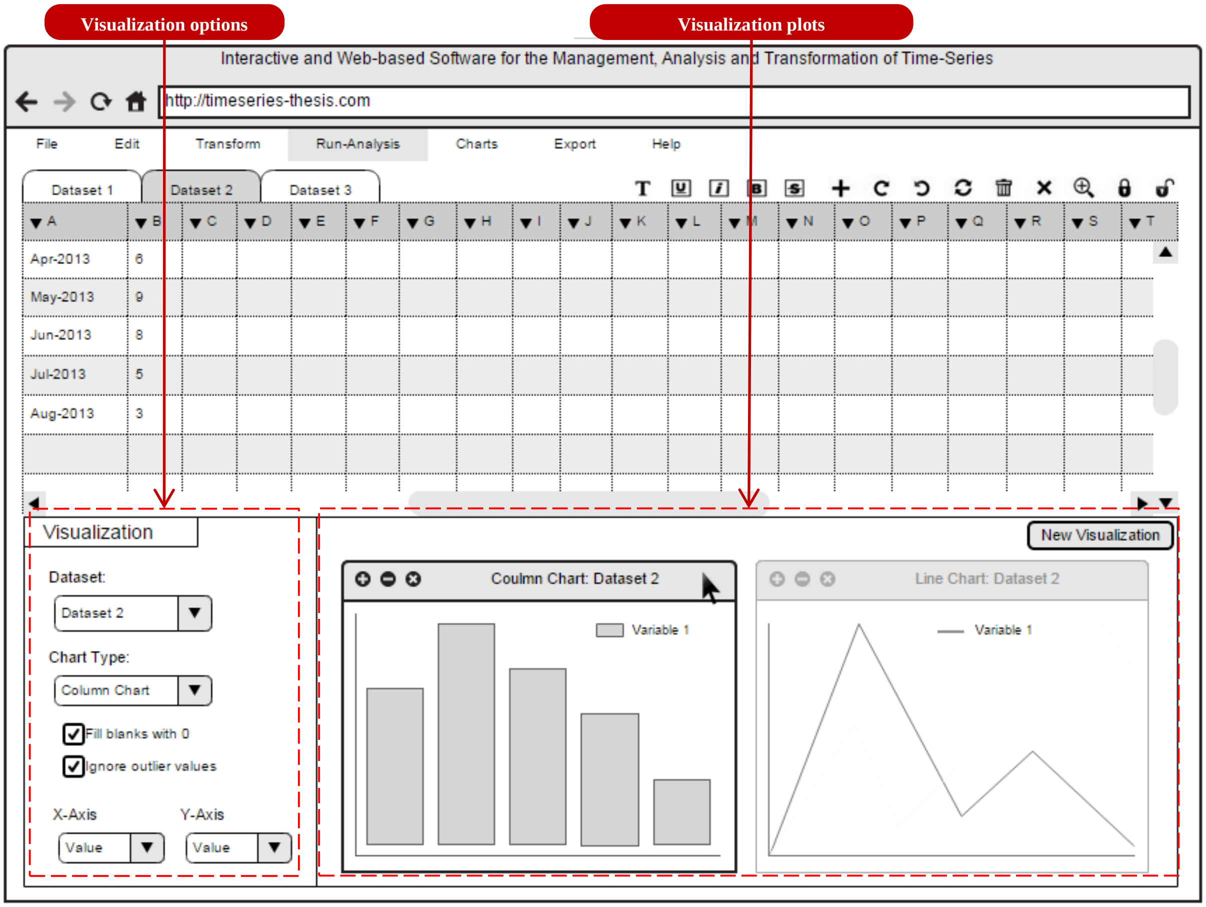
  The visualization pane of the thesis software makes up half of the interface. This way, users can see both the data and its visualization in one single screen.
- *Simple*: Visualization of data in itself is not complex and not ambiguous. This helps the audience quickly absorb and interpret the presented data.
- *Aesthetically appealing*: The graphics and presentation of the visualizations are designed to keep the user interested in the software.

As illustrated in figure 5.10, the visualization pane itself is divided into two sections: the visualization options and the plots. Visualization options are presented in well-arranged combo boxes, at the bottom-left of the screen. Users can:

- select datasets to visualize,
- select chart type
- determine how to fill in blanks/gaps in data
- determine how to handle data outliers[9]
- select fields to plot on x-axis and y-axis

The plots are displayed on the right part of the visualization pane. The flexible design enables users to visualize the same dataset in different chart types at the same time. Users can also visualize different datasets at the same time, usually to compare or combine time series characteristics. Figure 5.10 shows a column chart visualization of the data from our example of rainfall average in Nigeria.

[9] Outliers are data entries which do not follow the series trend i.e. far away from the other data



http://timeseries-thesis.com

Apr-2013	6
May-2013	9
Jun-2013	8
Jul-2013	5
Aug-2013	3

Dataset: Dataset 2

Chart Type: Column Chart

☑ Fill blanks with 0

☑ Ignore outlier values

X-Axis Value

Y-Axis Value

Figure 5.10: Visualizing time series

6 Conclusion and Outlook

The main motivation of this thesis lies in the fact that today's public and private organizations generate a large amount of data which are structured in form of time series. This creates a substantial need for effective management and analysis of these time series by end users who have little or no skills in analysis models for time series. This research is also motivated by the current discussions in literatures about end user software engineering (Burnett, 2009), (Rosene, 2013) and (Nardi & Miller, 1990). According to the literatures, end-user programmers have a set of requirements which differs from those of professional developers. Researchers advocate that new kinds of technologies should be invented to enable end users accomplish their requirements as well as improve the quality of tools for data management.

Firstly, this thesis reports on the current state of research on time series and their commonness in private and public spreadsheets. Times series is properly defined and illustrated with examples. Time series data has a natural temporal ordering. The value of a time series in a time period is often affected by the values of variables in preceding periods, thus making the order in which the data occurs in the spreadsheet very important. Time series are often confused with other data types like time-stamped data. Thus, this thesis describes the distinctive properties of time series and clarifies their differences from time-stamped data. An elaborate description of time series management and analysis models is also made.

As a way to identify time series in real world spreadsheets, existing spreadsheets corpuses for public spreadsheets (the EUSES corpus) and private spreadsheets (i.e. the Enrons corpus) were checked for occurrences of time series data. Results show that up to 14 percent of spreadsheets in these corpuses are time series.

An analysis of the tools currently being used for time series analysis and management was performed based on their support for analyzing, transforming and managing time series. Up to thirteen time series tools are reviewed. The strengths and the weaknesses of these tools are described, as well as their support for time series management and analysis. There are two important conclusions drawn from the review of these tools:

3. Only a few of these tools are easy to use for end users. Hitherto, most time series tools have been developed for usage by professional analysts and data scientists.
4. Most of the tools give poor support for the transformation of time series; which involves the reduction of time-stamped data to time series or the conversion of time series from one level of time frequency to another.

Besides, a web based time series software was designed for ease of use by end users. This represents a major contribution of this work to current research on time series. The time series software design aligns with typical properties of an end-user oriented software (Rosene, 2013), in our case for managing, analyzing and transforming time series. Prior to the design, a set of functional requirements of the software are presented, as well as the technical features of the thesis software. These requirements are as a result of the review and analysis of the existing tools currently being used for time series analysis.

The thesis software provides strong support for the transformation of time series. This is implemented with the MapReduce technique.

The usage scenarios of the time series software designed are illustrated using mockups and real world scenarios. The ultimate objective is to illustrate how users can work with the software to effectively manage, transform and analyze time series data.

As for the future, there is need for more research and development of web-based time series software. There are presently a few time series tools which are web-based. As established in this thesis, web based tools are generally better for end user oriented implementations because of the advantages they offer. It is hoped that the design made in this thesis is implemented in accordance with the requirements listed. This will be a good leap in terms of creating web-based time series tools.

Also, the design presented in this thesis for the transformation of time-stamped data to time series can be improved upon. Since the software is targeted at users with little knowledge of time series analysis, a more interactive interface can be provided - with possibility to track the progress of the data transformation i.e. where data comes from and its current state through the entire transformation process.

The system design can also leverage on the capabilities of pattern recognition. Users may want to understand the specific characteristics of a particular time series e.g. regularity, seasonality etc. The system should therefore be able to automatically identify the type of time series and in effect the category of analysis model to be used for the data.

Finally, since the scope of this thesis does not include handling of big data, the design presented in this thesis can be extended to handle huge sets of time series. This is particularly important for the future because of the fast rate at which data is growing in every field.

Bibliography

Abeyasekera, S. (2005). Quantitative Analysis Approaches to Qualitative Data: Why, When and How. *Statistical Services Centre, University of Reading, UK*, 1-2.

Aignera, W., Mikscha, S., Müllerb, W., Schumannc, H., & Tominski, C. (2007, February 7). Visualizing time-oriented data—A systematic view. *Computers & Graphics*, 5-7.

Andrew, J. K., Robin, A., Laura, B., & Alan, B. (2010). *The State of the Art in End-User Software Engineering.* USA.

Aquatic Informatics Inc. (2014). *Aquatic Informatics: Water Monitoring and Analysis Solution*. Retrieved March 2015, from Aquatic Informatics: http://aquaticinformatics.com/

Baum, C. F. (2003). *A review of Stata 8.1 and its time series capabilities.*

Beiwald, L. (2009). *Comparison of time series data analysis packages*. Retrieved March 2013, from AI and Social Science: http://brenocon.com/blog/2009/02/comparison-of-data-analysis-packages-r-matlab-scipy-excel-sas-spss-stata/

Brockwell, P. J., & Davis, R. A. (2002). *Introduction to Time Series and Forecasting, Second Edition.* USA: Springer.

Burnett, M. (2009). *What Is End-User Software Engineering and Why Does It Matter?* Corvallis, Oregon, USA: Springer-Verlag Berlin Heidelberg.

Castillejos, A. M. (2006). *Management of Time Series Data.* Canberra, Australia: University of Canberra.

Cochrane, J. H. (2005). *Time Series for Macroeconomics and Finance.* Chicago: Graduate School of Business, University of Chicago.

Cottrell, A., & Lucchetti, R. “. (2015). *Gretl User’s Guide.* Free Software Foundation.

Dean, A., Sullivan, K., & Soe, M. (2014, September). Retrieved March 2015, from OpenEpi: Open Source Epidemiologic Statistics for Public Health: http://www.openepi.com/Menu/OE_Menu.htm

Dean, J., & Ghemawat, S. (2010). *MapReduce: A Flexible Data Processing Tool.* USA: Communications of the ACM.

Diggle, P. (1990). *Time Series: A Biostatistical Introduction.* Oxford University Press.

Dreyer, W., Kotz, D. A., & Schmidt, D. (1995). Using the CALANDA Time Series. *ACM SIGMOD Intl. Conf.* San Jose, CA.

DTREG. (2010). *Time Series Analysis*. Retrieved March 2015, from DTREG: https://www.dtreg.com/methodology

Fisher, M., & Rothermel, G. (2005). The EUSES Spreadsheet Corpus: A Shared Resource for Supporting Experimentation with Spreadsheet Dependability mechanisms. *Proceedings of the Workshop on End-User Software Engineering*, (pp. 47–51).

Frye, C. D. (2010). *Stet by Step Microsoft Excel 2010.* Washington: Microsoft Press.

GMDH Shell LLC. (2013). *About GMDH Shell*. Retrieved March 2015, from GMDH Shell: https://www.gmdhshell.com/

Hamilton, J. (1994). *Time Series Analysis.* Princeton, United States: Princeton University Press, ISBN 0-691-04289-6.

Hao, M. C., Dayal, U., Keim, D. A., & Schreck, T. (2007). Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. *Eurographics/IEEE VGTC Symposium on Visualization* (pp. 27-34). Norrköping, Sweden: EUROVIS 2007.

Hermans, F., & Murphy-Hill, E. (2014). Enron’s Spreadsheets and Related Emails: A Dataset and Analysis. 1-2.

Laidre, A. (2012). *The Compelling Advantages of Web Based Business Software*. Retrieved March 2015, from iplanner.net: http://www.iplanner.net/business-financial/online/how-to-articles.aspx?article_id=software-web-based

Lee, J., & Elmasri, R. (1998). An EER Conceptual Model and Query language for Time Series Data. *Ling T.W., Ram S., and Lee M.L. (eds.): ER’98, LNCS 1507, Springer-Verlag, Berlin Hedelberg*, 22.

Leonard, M., & Wolfe, B. (2005). Mining Transactional and Time Series Data. *Data Mining and Predictive Modeling*, 1-3.

MATLAB. (2015). Retrieved March 2015, from Wikipedia: http://en.wikipedia.org/wiki/MATLAB

McCullough, B. D., & Vinod, H. D. (1999). The Numerical Reliability of Econometric Software. *Journal of Economic Literature*, 633-665.

McLeod, A. I., Yu, H., & Mahdi, E. (2011). *Time Series Analysis with R.* London: Elsevier.

Nardi, B. A., & Miller, J. R. (1990). The spreadsheet interface: A basis for end-user programming. *Human-Computer Interaction: INTERACT '90* (pp. 1-2). Amsterdam: North-Holland: Hewlett-Packard Laboratories.

NIST/SEMATECH. (2012). *e-Handbook of Statistical Methods.* USA: http://www.itl.nist.gov/div898/handbook/.

OpenEpi. (2014). *OpenEpi*. Retrieved March 2015, from OpenEpi: http://www.openepi.com/

P.Wang, H.Wang, & W.Wang. (2011). Finding Semantics in Time Series. *Microsoft Research Asia*.

Pentaho. (n.d.). *Time Series Analysis and Forecasting (with WEKA)*. Retrieved March 2015, from Pentaho: http://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka

Raychaudhuri, S. (2010). *Introducing Oracle Crystal Ball Predictor: a new approach to forecasting in MS Excel Environment.* USA: Oracle.

Robert H., S., & David S., S. (2010). *Time Series Analysis and its Applications (3rd Edition).* Springer.

Rosene, A. R. (2013). *End-User Software Engineering.* University of Winconsin-Platteville.

Rouse, M. (2014). *Software requirements specification*. Retrieved April 2015, from Tech Target: http://searchsoftwarequality.techtarget.com/definition/software-requirements-specification

SAS Institute Inc. (2004). *SAS/ETS 9.1 Users Guide.* Cary, NC, USA: SAS Publishing.

SAS Institute Inc. (2014). *SAS/ETS Software*. Retrieved April 2015, from SAS.com: http://support.sas.com/rnd/app/ets/index.html

Schutt, R. (2012). *Exploratory Data Analysis with Time-stamped Event Data*. Retrieved March 2015, from Columbia Datascience: http://columbiadatascience.com/2012/10/08/exploratory-data-analysis-with-time-stamped-event-data/

Shrivastava, R. (2012, September 27). *Big Data – Hadoop HDFS and MapReduce*. Retrieved March 2015, from CodeEmphasis: https://codemphasis.wordpress.com/tag/hadoop/

STATA. (2003). Introduction to timeseries. In STATA, *Time-Series Reference Manual* (pp. 1-5).

The MathWorks Inc. (2015, March). *Time Series*. Retrieved April 2015, from MathWorks: http://de.mathworks.com/help/matlab/time-series.html

Time-Series Forecasting. (2014). Retrieved February 2015, from Spreadsheet Analytics: https://sites.google.com/a/usfca.edu/business-analytics/business-function-analytics/forecasting

What are the Advantages of Data Visualisation. (2012). Retrieved March 2015, from Data Visualization and Presentation: http://www.uauug.org.uk/what-are-the-advantages-of-data-visualisation.html

Wikibon. (2012, August 1). *A Comprehensive List of Big Data Statistics*. Retrieved March 2015, from Wikibon Blog: http://wikibon.org/blog/big-data-statistics/

Zaslavsky, A. (2014). *Summary of Survey Analysis Software*. Retrieved March 2015, from http://www.hcp.med.harvard.edu/statistics/survey-soft/

Zhu, M. X., & Kuljaca, D. O. (2005). A Short Preview of Free Statistical Software Packages for Teaching Statistics to Industrial. *Journal of Industrial Technology*, 1-3.

Zicari, R. V. (2013). Chapter 3. Big Data: Challenges and Opportunities. In R. V. Zicari, *Big Data Computing* (p. 104).

Zivot, E. (2006). *Time Series Econometrics*. Retrieved March 2015, from http://faculty.washington.edu/ezivot/econ584/econ584.htm