ISCTE Business School
Instituto Universitário de Lisboa

MISSING DATA IN TIME SERIES: ANALYSIS, MODELS AND SOFTWARE APPLICATIONS

Francesca Minglino

Dissertation submitted as partial requirement for the conferral of

Master in Management of Services and Technology

Supervisor:
Prof. José Dias Curto, ISCTE Business School, Departamento de Métodos Quantitativos para Gestão e Economia (DMQGE)

Supervisor:
Prof. Alberto Lombardo, Università degli Studi di Palermo, Dipartimento di Innovazione Industriale e Digitale

September 2018

ISCTE Business School
Instituto Universitário de Lisboa

MISSING DATA IN TIME SERIES: ANALYSIS, MODEL AND SOFTWARE APPLICATION

Francesca Minglino

- Spine example -

ABSTRACT

Missing data in univariate time series are a recurring problem causing bias and leading to inefficient analyses. Most existing statistical methods which address the missingness problem do not consider the characteristics of the time series when imputing the missing values and, most of all, do not allow the imputation in a univariate time series context. Moreover, just a few methods can be applied to all missing data patterns. Finally, no intuitive procedure addressing the missingness obstacle exists in the literature. In this work of investigation, an algorithm having the aim of filling in these gaps is presented; its main purpose is to find a procedure that gives reliable imputations of the missing values, i.e. not far from the true ones. To this aim, the reliability and robustness of the algorithm have been tested through the simulations campaigns approach. Its innovative feature is the combination of the ARMA models, used to impute the missing values through a forecast and a backcast approach, and the Expectation-Maximization algorithm, used to achieve the parameters convergence. This approach was evaluated through the RMSE and the MAPE metrics, which showed that the algorithm can be used in almost every model setting among the tested ones, with a good reliability. However, one of the main limitations of the introduced procedure is that the non-convergence of the algorithm could bring to biased imputations. The algorithm can be applied step by step by a common analyst, in a more intuitive way than the majority of other existing approaches.

Keywords: Missing Data; Univariate Time Series; ARMA Models; Expectation-Maximization Algorithm.

JEL Classification: C2, C6.

INDEX

FIGURE INDEX

TABLE INDEX

CHAPTER 1: INTRODUCTION

Time series data can be found in nearly every study field, going from healthcare to finance, from social science to the energy industry. Many researches focused on time series analysis (Box and Jenkins, 1970; Chatfield, 2000; Brockwell and Davis, 2016) since these data show some features allowing to extract a lot of information about the process which generated them. Nearly everywhere, however, when data is measured and recorded, some issues linked to the missing values occur. Various reasons can lead to missing data: for instance, values may not be properly collected or measured, values could be measured by considered unusable by the analyst. Real life examples can be the closure of the markets for one day, the malfunction of the sensor recording some movements, a communication error (Moritz et al., 2015). Missing data can present different patterns and frequencies (Rubin, 1976) and their presence can lead to various problems. Indeed, when a missingness mechanism occurs, further data processing and analysis steps cannot be pursued since the common software statistics are not able to handle the missing data. Furthermore, the absence of data reduces the statistical power of the tests, which refers to the probability that the test will reject the null hypothesis when it is false, and can cause bias in the estimation of parameters. Moreover, also the representativeness of the chosen sample could be put in question. These are the main reasons why missing data have to be replaced with reasonable values, which do not have to bring sharp modifications to its distribution and the data generating process. A lot of researches have been done in this field in order to find trustworthy techniques to fill the missing values in without damaging the distributional shape of the data.

Many researches focused on different missing data imputation techniques, among which the most known ones are the Single and Multiple Imputation (Rubin, 1976; Rubin, 1987), as well as the Maximum Likelihood Estimation through the Expectation Maximization algorithm (Dempster et al., 1977), the Bootstrap (Efron, 1994) and also the Expectation-Maximization with Bootstrapping (EMB) algorithm (Honaker and King, 2010). The strength of the application of these kinds of solution to the problem of missingness is that the output is a complete dataset, ready to be analysed through all the basic and more sophisticated statistics.

1.1 OPPORTUNITY FOR INVESTIGATION

In the context of the problem of the missing data in time series, in the existing literature only a limited number of researches have focused on the specific case of univariate time series

imputation. In fact, just a few software applications exist in this field. The imputation methods mentioned before are, indeed, thought to be applied to multivariate datasets. Specifically, in order to be applied to time series data, these methods have to be adapted to consider the interactions in time beyond those in space: the process generating a time series indeed shows its features over time, therefore the lagged values of a variable have their own weight in describing the following values (Box and Jenkins, 1970), together with the other variables' present and lagged values if the dataset under analysis is a panel. Furthermore, in order to handle a univariate time series, the above-mentioned methods cannot rely on inter-variables interactions since just a time series is being considered. They must be adapted again in order to rely only on interactions over time of a time series own values. Therefore, univariate time series are a special challenge in the field of missing data imputation. It is reasonable to tailor the imputation algorithms in order to take into account both the characteristics of the time series and the lack of inter-variables interactions.

Another consideration has to be made, about the existing imputation algorithms. These methods are usually of low intuitiveness and of difficult implementation for a common user, who is dealing with the missing data problem for the first time. Moreover, even to employ one of the few existing software applications to bypass the problem of the missing data, a common user has to study part of the literature about the topic to understand what the software is doing and how to interpret the output.

The combination of these two gaps in the state of the art, which are the few researches about the univariate time series with missing data scenarios and the need of a guided software implementation, represent the opportunity to start this investigation.

1.2 Objectives of the Research

The general aim of this dissertation is to build an intuitive guided software application for dealing with missing data in a univariate time series scenario. The idea of the guided application is that the user, following the procedure step by step, could be able to handle his/her missingness problem without the need of looking for an explanation elsewhere. Further, the algorithm on which the application is built has to rely on basic time series concepts in order to be widely used and understood by the common users. Of course, the algorithm should give back reliable results, that the analyst can use to his/her own purpose.

Therefore, the following specific objectives are addressed from this work of investigation:

1)Time Series Data Structure

1.1) Identify the principal aspects to be included in a time series analysis;

1.2) Identify the most widely used models to fit a time series;

1.3) Define the impact of the missingness on time series data.

2)Missing Data Imputation Methods

2.1) Analyse the principal objectives of the application of an imputation method;

2.2) Identify the requirements of univariate time series analysis;

2.3) Build a procedure based on the previous objectives' development;

2.4) Determine how to sketch the built procedure to meet the user's intuitiveness needed.

1.3 RESEARCH QUESTIONS

Starting from the specific objectives of the research, with the final aim to meet the general objectives explained in the previous paragraph, this dissertation will answer to the following research questions:

1)Time Series Data Structure

Q1: "Is it possible to preserve the process generating a univariate time series when missing data occur?"

2)Missing Data Imputation Methods

Q2: "Is it possible to apply the same imputation algorithm structure to any missingness case?"

Q3: "Is it possible to apply the same imputation algorithm structure to both univariate and multivariate time series scenarios?"

1.4 INVESTIGATION METHODOLOGY

This investigation started in October 2017 and finished in September 2018. The purpose of this exploratory work is to give a contribution to the research field of the univariate time series with missing data.

The investigation methodology applied in this work follows an inductive approach. The research started from the literature review, which brought to the identification of the main topics associated to the time series with missing data and to the definition of the propositions that were used in the work. After this step, the application of a method built on the basis of the already existing approaches was done on simulated data. Furthermore, the results of the application of the procedure were analysed in order to assess the goodness-of-fit of the method. The final step

of the investigation was the definition of a vademecum about the whole methodology to be applied in this kind of situations.

1.5 STRUCTURE OF THE THESIS

This work is organized in five chapters. Chapter 1 is the introduction, which explains the objectives of the investigation, as well as the research questions. In Chapter 2, the literature review is developed, divided in 5 Sections: in the first one the Time Series Data are described; in the second one the problem of the Missing Data is introduced, with an insight about the patterns they can follow; the third and fourth sections deal with the old and new methods (respectively) used to handle the missingness problem; in the fifth and final section, a summary of the literature and the definition of the propositions is done. Chapter 3 deals with the description of the methodology applied. In Chapter 4 the application of the method is shown, with an overview about the code used in the software R, the statistical analysis tool. In the same chapter, the results of the application are presented and discussed and the vademecum built for the users is shown. Finally, Chapter 5 presents the conclusions, in which an answer to the research questions is given and the contribution of the work is reported. In the same final chapter, also the limitations and the starting points for further research are explained.

CHAPTER 2: LITERATURE REVIEW

The aim of the literature review is to collect, analyse and resume in a critical way the existing literature about the topic being studied, identifying the theories and studies underlying it in an integrated form. Thanks to the literature review, it was possible to identify the gap whose solution is the aim of this study. This chapter is divided in five main parts: in the first one, an overview about time series, its characteristics and the ARMA models is done; the second part is dedicated to the Missing Data, therefore the problem arising with the missingness is explained, as well as the different patterns they can show; in the third part, the old models used to handle the missingness in a variable or a set of variables are analysed, highlighting the advantages and the limitations; in the fourth part, the new and more sophisticated models are analysed and compared. Finally, a summary of the most important findings from the literature analysed is composed.

Because of the complexity of the investigation topic, in this literature review, before even highlighting strengths and weaknesses, the description of the features and of the application of each method is introduced. This choice was made to give to the reader an overview about the differences among the existing methods, in order to focus afterwards on the conditions under which a method can be preferred to another one.

2.1 TIME SERIES

A discrete time series is a sequence of discrete-time data, taken at successive equally spaced points in time. Data in time series present some interesting features, useful for conducing statistical analysis whose aim is to extract information about the observed phenomenon and to forecast the next values that the series can present. For instance, assuming that the variable Y_t has been generated by a stochastic process, it is common to find a link between the variable at time *t* and its lagged values Y_{t-1}, Y_{t-2}, …, Y_{t-j}. For this reason, the past observations of a variable can be used to say something about their generating process.

A time series y_t can be regarded as a particular realization of a stochastic process Y_y (Brockwell and Davis, 2016) and so it presents a mean (μ_t), a variance (σ_t^2) and a covariance ($\gamma_{(t,s)}$) when they exist) defined as follows:

$E(Y_t) = \mu_t$ $\qquad t = 1, 2, \dots, T$

$V(Y_t) = E(Y_t-\mu_t)^2 = \sigma_t^2$ $\qquad t = 1, 2, \dots, T$

$Cov(Y_t, Y_s) = E[(Y_t-\mu_t)(Y_s-\mu_s)] = \gamma_{(t,s)}$ $\qquad t, s = 1, 2, \dots, T \qquad t \neq s$

When these moments are independent of *t*, the series is said to be stationary. If it is not, some transformations like the first difference and/or the logarithm have to be applied in order to obtain a stationary series. Usually, those are sufficient to this aim. In this work, heteroskedasticity issues are not taken into consideration: the conditional variance of the errors will be always assumed as constant. So, just homoskedastic time series will be considered in this investigation. Further studies could focus on this issue.
In the following paragraphs, the variations that a time series can present over time and some models dealing with stationary processes are treated.

2.1.1 COMPONENTS

A time series is rarely perfectly constant over time. Indeed, it usually presents some variations, and it can be decomposed in four components which may or may not exist at the same time. The components are:

1. Seasonal variation. This type of variation refers to cycles that repeat regularly over time, generally annually. It arises for many series, whether measured weekly, monthly or quarterly, when similar patterns of behaviour are observed at particular times of the year (Chatfield, 2000). An example are the retail sales, which tend to peak for the Christmas season and then decline after the holidays;

2. Trend. This type of variation is present when a series exhibits a growth or a decrease over time. It can be defined as a long-term change in the mean level. A time series can present a stochastic or deterministic trend, which can be linear, as well as quadratic, parabolic or any other shape. The trend is usually the result of long-term factors influence like changes in demographics, technology or customers habits;

3. Cyclical variation. This variation includes a cyclical variation which does not present regular repetitions over time and lasts more than one year. It is usually hard to estimate, since the distance in time between two cycles can be long and it is not constant. Examples include business cycles over periods of five or more years and the periodic variations in nature, for instance regarding temperature or the lifecycle of the beings;

4. Irregular fluctuations. This last variation is the part of the time series which has been 'left over' after trend, seasonality and other systematic effects have been removed. The irregular component of a time series is unpredictable, since it may be completely random. The irregular component catches also sudden variations such as an increase in the price of steel due to a strike in the factory, reflecting some extraordinary events.

The Figure 1 below shows graphically how a raw time series can be decomposed in its components, in the case in which it presents them all. Since the cyclical variations are hard to detect, the decomposition models include this component into the trend.

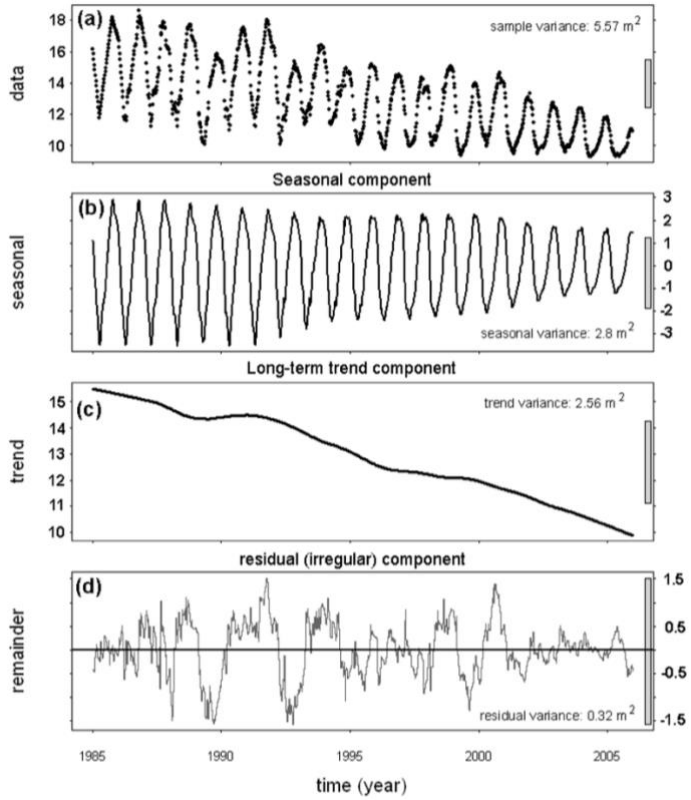

Figure 1: Decomposition of a time series

2.1.2 ARIMA MODELS

The ARIMA class of models is an important forecasting and time series analysis tool. These models are also known as the Box-Jenkins models since these two authors first introduced a method for implement them in an efficient algorithm. The acronym ARIMA stands for 'AutoRegressive Integrated Moving Average', where the Integrated part indicates the number of differences needed to turn the series into stationary. The other two components, the AutoRegressive and the Moving Average indicate whether the observation at time *t* depends on its lagged observations or on the previous shocks (errors), or on both. In an ARIMA(p, d, q) model, the letters p and q state the maximum lag order still influencing the variable at time t, respectively for AR(p) processes and MA(q) processes. Finally, the letter d regards the integrated part, as it was said previously.

In this work, as will be explained more in detail in Section 4.1.1, only stationary time series will be considered. Therefore, from now on, the ARIMA(p, 0, q) models, that can be written in the equivalent form ARMA(p, q), will be employed.

In order to set an ARMA model in a proper way, it is suggested to have more than 100 observations for each time series, although this could increase the cost of data collection. Despite the high number of observations needed, the ARMA models catch the behaviour of the time series in a better way in the short run (Box and Jenkins, 1970) and this is the main advantage in employing this class of models. Moreover, while building a proper time series model, the principle of parameters parsimony has to be considered. The Box-Jenkins models follow this principle and favour the one with the smallest possible number of parameters out of a number of suitable models. The simplest one, that still maintains an accurate description of inherent properties of the time series, is preferred to the more complex ones. Box and Jenkins (1970) introduced as main scope of the ARMA models the identification and the estimation of the model underlying a time series, with the final aim to perform a forecast of one or more future data. However, through the same identified and estimated model, also a back-forecast (or backcast) can be applied in order to find older values than the observed time series. The ARMA models represent the more flexible and easier to adapt approach to the time series analysis and it is the reason why this investigation prefers the application of this class of models rather than, for instance, the smoothing models one.

In Section 3.1.1, further details about the ARMA models applied in this investigation is presented.

2.2 MISSING DATA

In statistics, when no data value is stored for the variable in an observation, a missing datum or value occurs. Missing data are common in research in economics, sociology, and political science when governments choose not to, or fail to, report critical statistics. Sometimes, also data collection can cause missing values when, for instance, the researcher mistakes in data entry or collects the responses unproperly. Rubin (1976), for instance, introduces his paper presenting the problem of surveys addressed to families, which could not be located in the following years to repeat the same survey. Situations like this one create missing values.

When missing data are found in a time series, this causes a lot of troubles. Indeed, since the special feature of time-series data is that consecutive observations are usually not independent and so the order in which the observations are collected has to be taken into account, when a missingness occurs the analysis of the following observations becomes harder due to the lack of precious information.

In general, missing data affect the main objectives of time-series analysis (Chatfield, 2000):

1.Description. Describing the data using summary statistics and/or graphical methods is difficult since the most part of software statistics do not work when a time series presents missing data. A time plot of the data would show holes in the series;

2. Modelling. Finding a suitable statistical model to describe the data-generating process, based only on past values of that variable considered, loses power if some observations are unhelpful to this scope. The statistical model found could be different from the real one;

3. Forecasting. As a consequence of the previous two steps, forecasting the future values of the series presents biases and low efficiency, due to the difficulties in describing and modelling the series.

Summarizing what said, the goal of a statistical procedure should be to make valid and efficient inferences about a population of interest in order to use the extracted information to various aims. When looking for a way to find a value as close as possible to the missing one, some considerations have to be made to avoid damaging the inferences about the entire population. Let's consider for instance the easiest way to fill in the missing data, i.e. the common practice of mean substitution. The method consists in replacing each missing value with the mean of the observed values. This may accurately predict missing data but distort estimated variances and correlations (Shafer and Graham, 2002). A missing value treatment is embedded in the modelling, estimation, or testing procedure applied to the whole-time series and basic criteria for evaluating statistical procedures have been established by Neyman and Pearson (1933). For instance, let's assume that Q denotes a generic population quantity (parameter) to be estimated and that $\hat{Q}$ is an estimate of Q based on a sample of data. If the sample includes missing values, then the method that will be used to handle them should be considered part of the overall procedure for calculating $\hat{Q}$. If the whole procedure works well, then $\hat{Q}$ will be close to Q (Shafer and Graham, 2002), therefore showing small bias and standard deviation. Bias and variance are often combined into a single measure called mean square error (MSE), which is the average value of the squared distance $(\hat{Q} - Q)^2$ over repeated samples. However, even if the procedure gives a low MSE as output, one should avoid solutions that apparently solve the missing data problem but actually redefine the parameters or the population. For instance, the mean substitution gives a biased estimation of the variance of the process since the estimate is lower than it should be. This effect should be always controlled by the analyst.

2.2.1 PATTERNS

Going more in depth in the analysis of the missing data in a time series, survey methodologists have historically distinguished unit nonresponse from item nonresponse (Lesser and Kalsbeek, 1992): the first one occurs when the entire data collection procedure fails (for instance because the sampled person is not at home or refuses to participate, etc.) while the second one occurs when partial data are available (for instance the participant does not respond to certain individual items). In this work, only item nonresponses are analysed.

The missingness of the data can be related to the observed data, can be caused by a random process or can be linked to some specific reasons. Adopting the generic notation introduced by Rubin (1976), but adapting it to the time series domain, let's imagine a complete time series $Y_{t\,com}$, which can be partitioned in the observed part $Y_{t\,obs}$ and in the missing part $Y_{t\,mis}$ such that $Y_{t\,com} = (Y_{t\,obs}, Y_{t\,mis})$. Let's assume M as the missingness vector, which takes the value 0 if the datum is observed and 1 if it is missing. Rubin (1976) defined missing data to be Missing At Random (MAR) if the distribution of missingness does not depend on the missing data $Y_{t\,mis}$ such that

$$P(M/\,Y_{t\,com}\,) = P(M/\,Y_{t\,obs}\,) \qquad (1)$$

MAR means there is a systematic relationship between the propensity of missing values and the observed data, but not the missing data. Whether an observation is missing has nothing to do with the missing values, but it does have to do with the values of an individual's observed variables.

A special case of MAR occurs when the distribution of the missing data does not depend on $Y_{t\,obs}$ either. This case is defined Missing Completely At Random (MCAR) and it can be explained as

$$P(M/\,Y_{t\,com}\,) = P(M\,) \qquad (2)$$

MCAR, means there is no relationship between the missingness of the data and any values, observed or missing. The missing data points are a random subset of the data. There is nothing systematic going on that makes some data more likely to be missing than others.

MAR and MCAR are both considered ignorable because no information about the missing data itself has to be included when dealing with the missing data. Instead, the third pattern of missingness that could occur is called non-ignorable, since the missing data mechanism itself has to be modelled. This is the case of Missing Not At Random (MNAR) data, when there is a relationship between the propensity of a value to be missing and its own values. In this last

case, some model to explain the missingness has to be included. It is the most difficult situation to deal with.

Notice that Rubin (1976) definitions describe statistical relationships between the data and the missingness, not causal relationships.

The assumption of MCAR data is usually very strong to hold. For this reason, in most of the cases, the assumption of the analysts is MAR. However, using maximum likelihood procedures as explained in the following paragraphs, one may achieve good performances without being sure about the distribution of missingness.

Assuming that $Y_{t\,com}$ comes from a population distribution $P(Y_{t\,com};\ \theta)$, where θ is the set of $Y_{t\,com}$ parameters (which can be the ARMA ones), it is tempting to ignore the missing data and base all the statistical analysis on the distribution of the observed portion of data $P(Y_{t\,obs};\ \theta)$ (Shafer and Graham, 2002). This distribution could be obtained as

$$P(Y_{t\,obs};\ \theta) = \int P(Y_{t\,com};\ \theta)\, dY_{t\,mis} \tag{3}$$

as explained in Hogg and Tanis (1997) texts on probability theory. As Rubin (1976) explained, however, using this equation as base for the statistical procedures is not always correct, since it represents a proper sampling distribution for $Y_{t\,obs}$ only if missing data are MCAR and it is a correct likelihood if data are at least MAR. If data cannot be considered MAR, a distribution for M has to be chosen, since it would be the case of non-ignorable missingness. Let $P(M/Y_{t\,com}\,;\xi)$ be a model for *M*, where ξ is the set of parameters of the missingness distribution. The joint model for the whole set of data would depend both on θ and on ξ as follows

$$P(Y_{t\,obs}, M; \theta, \xi) = \int P(Y_{t\,com};\ \theta)\, P(M/\ Y_{t\,com}\ ;\ \xi)\, dY_{t\,mis} \tag{4}$$

The practical implication of MNAR is that the likelihood for θ now depends on an explicit model for *M*. Many restrictions and limits occur when dealing with unknown patterns of missing data. For this reason, it is important to find models that do not need strict assumptions about the missingness pattern, unless one is able to correctly identify the distribution of the missing data, and to try to use the whole dataset. Especially when the series is a time series, every observation has its own weight and omitting some values can bias the estimates of the parameters of the whole distribution.

2.2.2 IDENTIFICATION

There is no exact science about how to identify the pattern of missingness. Some empirical analysis however can be done by the analyst before starting the process of imputation of the missing values. The only way to distinguish between MAR and MNAR is to try to measure some of the missing values by asking the non-respondent to answer to some key survey items. This could mean also asking them the reason why they didn't answer. However, in most missing data situations, it is rare to have the possibility to directly speaking with the non-respondents. So, the most used practice is to analyse the type of data and try to get a conclusion. For instance, if the series regards the aggregate consumption of cocaine in the last 20 years and some of the data are missing, it is more likely that data are MNAR than MAR since people could not be willing to declare the number of grams of cocaine they consume. Or, if measuring the body pressure of an individual, values are less likely to be written down if the previous day observation is higher than 160, then data are more likely to be MAR instead of MNAR.

Distinguishing from MAR and MCAR is more difficult in a univariate scenario than in a multivariate one. Indeed, for multivariate datasets it is possible to apply the Little test (Little, 1988) or also another R package called MissMech (Jamshidian et al., 2014). Further studies about this topic have been done by Heitjan and Basu (1996) and, apart from evidencing the differences in the behavior of the results if data were MAR or MCAR, they highlighted that these two patterns of missingness still let the analysts to work with that dataset. For this reason, the two patterns are called "ignorable".

To conclude, what an analyst can easily do to continue his/her studies is to assume a pattern basing the assumption on his/her own experience about the data field or about the context of data collection.

2.3 OLD METHODS TO HANDLE MISSING DATA

In this section, the Old Methods to deal with the missing data are explained. The most known ones are the Complete Case Analysis, the Available Case Analysis and the Single Imputation.

2.3.1 COMPLETE CASE ANALYSIS AND AVAILABLE CASE ANALYSIS

The Complete Case Analysis, also known as Listwise Deletion, confines attention only to units that have observed values for all variables under consideration. Therefore, the method omits the missing data from the analysis, as if they had never been included among the observed values. This procedure does not give biased results only when data are MCAR. When the missing data are not MCAR, results from listwise deletion may be biased, because the complete

cases could be unrepresentative of the full population. If the departures from MCAR are not huge, then the impact of this bias might be unimportant, but in practice it can be difficult to judge how large these departures might be. Little and Rubin (1987) suggest to reduce biases from listwise deletion in some non-MCAR scenarios by applying weights to the considered observations so that their distribution is closer to the full population one. The only difficulty in this procedure is to find the right weights from the probabilities of responses, which is not always easy and unbiased.

Clearly, even if Listwise Deletion could give unbiased results if data are MCAR, when this method is applied to a time series more problems occur. For instance, if the observation interval is a month and a time series has 12 observations per year, eliminating one of these values because it is missing implies to slip one month forward the previous month observation (March observation is missing and it is deleted, February observation takes its place). No more seasonality studies can be made on this time series. For this reason, Complete Case Analysis won't be taken into consideration in this study.

Available-Case (AC) Analysis, in contrast to the Complete Case Analysis, only considers different sets of sample units for different parameters. This means that, without deleting any value, parameters are estimated from different sets of units, probably of different length. As regards to a time series for instance, if it presents 20 consecutive observed values, then a missing one, then other 10 consecutive observed values, one could estimate an ARIMA model splitting the series so to only consider the two strings of observed data. Of course, this procedure leads to difficulties in computing standard errors or other measures of uncertainty. The estimated parameters of the series could be very different among them and also from the original series parameters. Although Kim and Curry (1977) argue that the Available Case Analysis can improve the quality of the estimates if compared to the Complete Case Analysis, other studies (Little and Rubin, 1987; Little, 1992) do not agree with that. To conclude, also the Available Case Analysis won't be used in this work since the uncertainty about the parameters resulting from the analysis is too high to be handled.

2.3.2 Single Imputation

Single Imputation is a simple method introduced by Rubin (1987) which involves filling in a value for each missing value. This method presents two main features of interest, which turn it more attractive than the listwise deletion. First, after imputing the missing values, standard complete-data methods of analysis can be run on the complete series. As a consequence, dealing

with complete datasets allows all kinds of users to reach reasonable conclusions about the data being analysed, applying the statistical tools they already know in any dataset. Second, imputations can incorporate the data collector's knowledge which, in many cases involves better information and understanding of the process being studied. The data collector could also know the possible reasons creating nonresponses.

Single Imputation is potentially more efficient than complete case analysis because no units are omitted. The retention of the full sample helps to prevent loss of power resulting from a diminished sample size (Shafer and Graham, 2002). Moreover, if the observed data contain useful information for predicting the missing values, an imputation procedure can make use of it and maintain high precision. Indeed, if a time series presents trend and/or seasonality, this information can be used to impute a value closer to the real one.

Single Imputation however presents also some disadvantages: indeed, a single imputed value cannot consider the uncertainty due to the unknown missing value and the uncertainty due the imputation model itself. This can lead to underestimate the consequences of these two uncertainties, such as a too optimistic variance.

For instance, considering a Single Imputation model of mean substitution, this could impute reasonable values for the ones which were missing, since the best prediction of a sample of data is its own mean, but it gives back a too small variance. Indeed, the mean is calculated on the n_{obs} observed values of Y_t, instead of on the n total values. After imputing the missing values by substituting the mean of Y_{obs}, the variance of the whole series turns to be smaller than it should, by a factor of $\frac{n_{obs}}{n}$. This leads also to excessively large significance levels: this means that the estimated variance is biased.

It is generally more desirable to preserve the distribution of a variable. Survey methodologists, field in which nonresponses are common, have developed a wide array of single-imputation methods that more effectively preserve distributional shape (Madow, Nisselson, & Olkin, 1983). For instance, one of the procedures is the hot deck imputation which, in the case of univariate replaces each missing value by a random draw from the observed values. However, this method still distorts correlations and other measures of association. Both mean substitution and hot deck produce biased estimates for many parameters under any type of missingness. Another type of single imputation is the one based on conditional distribution which, under MAR assumptions, produces nearly unbiased estimates. Supposed that $y_{t\,com}$, where $y_{t\,com} = (y_{t\,obs}, y_{t\,mis})$, comes from the distribution $P(Y_{t\,com}; \theta)$, imputing from the conditional distribution means taking a draw from

$$P(Y_{t\,mis}/Y_{t\,obs}\,;\,\theta) = \frac{P(Y_{t\,obs}, Y_{t\,mis}; \theta)}{P(Y_{t\,obs}; \theta)} \quad (5)$$

where θ is actually $\hat{\theta}$, the estimator of θ obtained from $y_{t\,obs}$ (Schafer and Graham, 2002).

One application of the single imputation through the conditional distribution in the field of the time series is given by Kihoro et al. (2013). The idea of their procedure was to put together the ARIMA process followed by a time series with an iterative imputation method. The researchers, after having identified the ARIMA/SARIMA model which seemed to fit better with the series, estimated the parameters of the observed part of the variable. Through these parameters estimates, the first missing value was then filled in. The following step was to estimate again the parameters of the series, considering also the filled-in value, so to be able to fill in the second missing value and so on. Making an example, let's consider a time series y_t, presenting $m = m_1, \dots, m$ missing data and following ARMA(1,1) process. The parameters to be estimated are therefore θ_1 and φ_1, from the observed part of the data $y_{t\,obs}$. After estimating the two parameters, the first missing value y_{m_1} is imputed as $y_{m_1} = \theta_1 y_{m_1-1} + \varphi_1 e_{m_1-1}$. The parameters are estimated again including the imputed value of y_{m_1}. y_{m_2} is then imputed in the same way and the entire procedure ends when all the missing values are filled in.

Even if the method considers the process underlying the series, the simulations the researchers did to test the effectiveness of this procedure did not provide quite good results.

In the context of a time series, the most part of single imputation methods are not able to take into account the relationship existing among the data: if the time series is an autoregressive process, replacing a missing value with a random draw from the rest of the data does not reflect the nature of the variable. If replacing the missing value with the mean, the information about eventual trend or seasonality would be lost. Drawing from the conditional distribution as in the study of Kihoro et al. (2013) could instead preserve the nature of the time series but some modifications have to be made in order to reach better results.

To conclude, although the intuitive easiness of the single imputation class of methods, the defect of underestimating variability and of losing precious information about the nature of the time series is insurmountable and this is the reason why other more precise methods are usually preferred.

2.4 New Methods to handle Missing Data

A good missing data handling technique to has to satisfy three requirements:

- it should allow standard complete-data methods to be used;

- it should yield valid inferences that produce estimates and standard errors that reflect the reduced sample size as well as the adjustment for the observed-missing values differences;
- it should display the sensitivity of inferences to various plausible models for missingness.

The new methods to handle the missingness in a dataset try to achieve these requirements, that is the reason why they are usually preferred to the old ones. In this Section, the Maximum Likelihood Estimation with the Expectation-Maximization Algorithm, the Multiple Imputation, the Bootstrap and the EMB Algorithm are discussed.

2.4.1 Maximum Likelihood Estimation and Expectation Maximization Algorithm

The Maximum Likelihood can be used to estimate the parameters of a variable although the presence of missing data. Literature shows that this method gives good results and, most of all, can be used with any missing data pattern. A general optimization algorithm for ML in missing data problems was described by Dempster et al. (1977) in their influential article on the Expectation-Maximization (EM) algorithm. This brings the ML, together with the EM, to be widely used. In order to have an insight about how the ML works, let $y_{t\,com}$ denote a time series, decomposed as before in its two parts, the observed and the missing one such that $y_{t\,com} = (y_{t\,obs}, y_{t\,mis})$. Being Equation (3) the marginal probability density of $Y_{t\,obs}$, assuming that the missing data mechanism could be ignored, Little and Rubin (1987) defined the likelihood of θ as any function of θ proportional to $P(Y_{t\,obs}/\theta)$ as follows

$$L_{ign}(\theta/y_{t\,obs}) \propto P(Y_{t\,obs}/\theta) \tag{6}$$

When the missing data mechanism cannot be ignored (MNAR data) however, the distribution of $y_{t\,obs}$ becomes as in Equation (4) and so the likelihood function of θ becomes

$$L(\theta, \xi/y_{t\,obs}, M) \propto P(Y_{t\,obs}, M/\theta, \xi). \tag{7}$$

However, in many realistic applications, departures from MAR (intended as non-ignorable mechanism) are not large enough to effectively invalidate the results of a MAR-based analysis (Collins et al., 2001).

The principle of drawing inferences from a likelihood function is widely accepted. ML estimates $\hat{\theta}$, the value of θ for which Equation (6),(7) are highest and it has attractive theoretical properties just as it does in complete-data problems. Under rather general regularity conditions, if considering large samples, it tends to be approximately unbiased (Shafer and Graham, 2002). The MLE is also highly efficient by definition: as the sample size grows, its variance

approaches the theoretical lower bound achievable by any unbiased estimator (Hinkley and Cox, 1979).

Even if Likelihood methods are more attractive than single imputation or ad hoc techniques, they still present some limitations which are the assumptions they rest on. The first assumption is that the sample is large enough to obtain unbiased and normally distributed estimates. However, it should be taken into account that, when some data are missing, the sample should be even larger than usual since its size has been reduced. The second assumption is the parametric model for the complete data $P(Y_{t\,com};\,\theta)$ where the likelihood function comes from. The ML could not be robust when departures from the model assumptions occur.

Applying the Maximum Likelihood Estimation to an ARMA(p, q) model, taking the form of Equation (21), the set of parameters to be estimated is $\theta = (\emptyset_1, \emptyset_2, \dots, \emptyset_p, \theta_1, \theta_2, \dots, \theta_q, \sigma^2)$ (Hamilton, 1994). The conditional log likelihood is then

$$L(\theta) = log\, f_{y_t, y_{t-1}, \dots, Y_1/Y_0, \varepsilon_0}(Y_t, Y_{t-1}, \dots, Y_1/Y_0\,, \varepsilon_0; \theta) \qquad (8)$$

being Y_0 and ε_0 the initial expected values of Y_t and ε_t, respectively.

The Expectation-Maximization algorithm rectangularizes the dataset through an iterative procedure which aims to find the ML Estimates when the likelihood function cannot be easily constructed, therefore a simplified function involving the maximization of unknown parameters has to be used. The EM algorithm is mainly used to find stable and reliable estimates of the parameters of a variable, as shown in Dempster et al. (1977) paper, and even if it was not born in the missing data field, its potentialities have soon been recognized. The EM iterations can be thought as the first step of a missing data imputation procedure, since they can be used to define the parameters characterizing a time series, although the missingness. The following step is to impute the missing values through the estimated parameters. As shown in Horton and Kleinman (2007) paper, where 100 cancer prognostic studies, 81% of which presented missing values, were treated with different methods, those filled in through the EM algorithm gave excellent outcomes. This happens because the EM algorithm is the only one pursuing the convergence of the parameters repeating the iterations.

Both ML and EM are widely employed approaches in the missing data field, mainly thanks to their flexibility and easiness to use. Indeed, they can be applied in every missingness scenario, since no limitations have to be set about the parameters of the series, the ones that will be estimated, nor about the missing values pattern.

2.4.2 MULTIPLE IMPUTATION (MI)

A flexible alternative to likelihood methods for a variety of missing data problem has been proposed by Rubin (1987) and this method is called Multiple Imputation. It is built based on the single imputation from the conditional distribution of a variable or a set of variables but, thanks to its structure, it handles the problem of the imputation uncertainty much better than its predecessor (Honaker and King, 2010). As the single imputation, since the result of the procedure is a complete dataset, it allows the analysts to perform the standard statistics without any limitation after the imputation is completed. Moreover, also the multiple imputation allows to incorporate the analyst's own knowledge about the process being studied, which is a preciousness for the goodness of the final result.

Rubin's (1987) method works imputing for each missing value of every variable taken in consideration (the method was born to handle missingness in multivariate datasets) $m > 1$ imputations. After obtaining the *m* sets of imputed values and after substituting them to the missing values, *m* plausible alternative versions of the complete data are obtained. Each version is then analysed through the same statistics, whose results are then combined by simple arithmetic to obtain overall estimates and standard errors that reflect missing-data uncertainty as well as finite-sample variation. The Figure 2 below, extracted from Schafer and Graham (2002), shows a schematic representation of multiple imputation, if a multivariate dataset is considered.

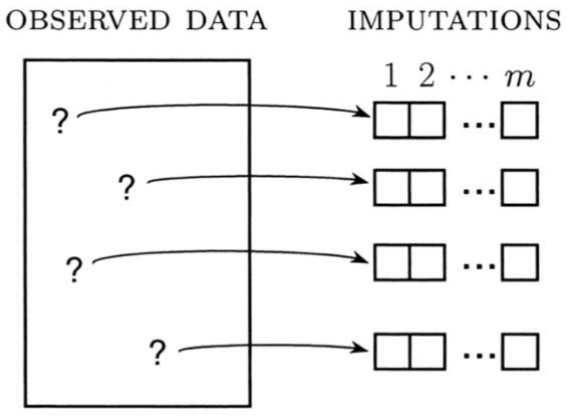

Figure 2: Schematic representation of Multiple Imputation

Unlike other Monte Carlo methods, with Multiple Imputation a large number of repetitions for precise estimates is not needed. Rubin (1987) showed indeed that the efficiency of an estimate based on *m* imputations is computed as $(1 + \frac{\lambda}{m})^{-1}$, where λ is the rate of missing information. For instance, with 5% missing information, if $m = 10$ imputations, the efficiency of the estimates would be $(1 + \frac{0{,}05}{10})^{-1} = 99\%$ efficiency.

As regards the combination of the results of the m analysis, Rubin (1987) proposed a method for a scalar parameter. He supposed Q to represent a population quantity to be estimated, $\hat{Q}$ and $\sqrt{U}$ being the estimate of Q and the standard error that would be used if there were no missing data in the series. The sample is assumed to be big enough to have normally distributed residuals, such that $\hat{Q} \pm 1{,}96\sqrt{U}$ reaches the 95% of coverage. After having imputed m set of missing values, m different versions of these two measures $[\hat{Q}^{(j)}, U^{(j)}]$, where $j = 1, 2, \dots, m$, are obtained. Rubin's (1987) overall estimate is simply the average of the m estimates,

$$\bar{Q} = m^{-1} \sum_{j=1}^{m} \hat{Q}^{(j)} \tag{9}$$

The uncertainty in Q can be split in two parts: the average within-imputation variance, which represents the uncertainty inherent in any estimation method, and the between-imputations variance, which represents the uncertainty due to the missing data treatment (Dong and Peng, 2013), calculated as follows

$$\bar{U} = m^{-1} \sum_{j=1}^{m} U^{(j)} \tag{10}$$

$$B = (m-1)^{-1} \sum_{j=1}^{m} [\hat{Q}^{(j)} - \bar{Q}]^2 \tag{11}$$

Rubin's (1987) combination process calculates the total variance as a modified sum of the two components

$$T = \bar{U} + (1 + m^{-1})\mathrm{B} \tag{12}$$

Having the overall estimate of the parameter $\bar{Q}$, and having its variance, confidence intervals and tests can be computed through a t-student approximation $T^{-\frac{1}{2}}(\bar{Q} - Q) \sim t_{\nu}$, where the ν degrees of freedom are given by

$$\nu = (m-1)\left[1 + \frac{\bar{U}}{(1 + m^{-1})\mathrm{B}}\right]^2 \tag{13}$$

The degrees of freedom may vary from $m - 1$ to infinity depending on the rate of missing information.

Making a step backwards, the imputations can be created through special algorithms (Schafer, 1997; Rubin and Schenker, 1986) but, in general, they should be drawn from a distribution for the missing data that reflects uncertainty about the parameters of the data model. As seen in the single imputation, it is better to impute the data starting from the conditional distribution $P(Y_{mis}|y_{obs};\, \hat{\theta})$, where $\hat{\theta}$ is an estimate derived from the observed data. MI extends this

procedure by first simulating m independent plausible values for the parameters, $\theta^{(1)}, \ldots, \theta^{(m)}$ and then drawing the missing data from $P(Y_{mis}|y_{obs};\ \hat{\theta}^{(t)})$, where $t = 1, \ldots, m$. Parameters, in Multiple Imputation, are treated as random rather than fixed.

One of the main disadvantages of the single imputation, i.e. the overconfidence coming from the standard analysis of just one complete dataset, is eliminated through the variation among the multiple imputations (Schafer and Graham, 2002). Indeed, as seen previously, the standard errors of the quantity of interest are incorporated in the variation across the estimates from each complete dataset.

The Multiple Imputation has the great advantage of preserve the important features of the joint distribution in the imputed values. It is quite easy also to maintain the distributional shape of the imputed variable (Schafer, 1997). For instance, if a variable is right skewed, it may be modelled on a logarithmic scale and then transformed back to the original scale after imputation.

Moreover, the two-step nature of multiple imputation has other two advantages if compared to one-step approaches: first, since it is possible to include other variables or important information in the imputation model, this can make the estimates even more efficient (property called "super-efficiency"); second, since the imputation model does affect only the missing values, the method turns to be less model-dependent (Schafer and Graham, 2002).

Like Maximum Likelihood, also Multiple Imputation relies on large-samples approximations, but when the sample has moderate size the approximations seem to work better for MI than for ML (Shafer and Graham, 2002). Moreover, MI relies on the assumption that missing data are Missing At Random (MAR), even if some studies implying MNAR data have been published (Glynn et al., 1993; Verbeke and Molenberghs, 2000). The approach of Multiple Imputation when dealing with MNAR data is slightly different since mixture models have to be used. This class of models assumes separate parameters for observed and missing data and it works repeatedly filling in missing values, estimating parameters using the filled-in data and then adjusting for variability between imputation. Mixture models provide a satisfactory and robust approach to inference, mostly for means and regression parameters, although they require a higher number of imputations for obtaining good results if compared to methods applied to ignorable nonresponse mechanism (Glynn et al., 1993).

Attention has to be paid when standard imputation models are applied to Time Series Cross Section (TSCS, Panel) data, since they can give absurd results (Honaker and King, 2010).

Indeed, the main characteristic of TSCS data, which are the smoothing behaviour that time series present and the possibly sharp changes in space, due to the cross-section structure, can bring the imputations to be far from the real values. Some studies in this field tried to solve this problem developing ad hoc approaches such as imputing some values with linear interpolation, means, or researchers' personal best guesses. These techniques often rest on reasonable intuitions: many national measures change slowly over time, observations at the mean of the data do not affect inferences for some quantities of interest, and expert knowledge outside their quantitative data set can offer useful information. The remaining missing data are then removed applying listwise deletion to let the analysis software work. However, Little and Rubin (1987) showed that, although relying on apparently reasonable assumptions, this type of procedure produces biased and inefficient inferences and confidence intervals. Honaker and King (2010) suggested to include, in the imputation model, the information that some variables can present smooth trends over time by creating these basis functions through polynomials, LOESS, splines etc. The two researchers also suggested to include the lags of the variable under analysis, if it is a time series. Finally, since a predictive model is being followed, the leads of the same variable can be included as well, so to use the future to predict the past. Honaker and King (2010) did not stop their analysis to these suggestions, rather they developed a new algorithm for handling TSCS data, whose name is EMB algorithm and which is explained in detail in the following sections.

To conclude, Multiple Imputation can be performed when the variable with missing data is a time series, provided that preventive measures are taken. Indeed, the nature of the series should be preserved, otherwise useful information would be missed. Therefore, in order to obtain good results, information about the smoothness of the time series, about the eventual seasonality and about eventual autoregression effects within the series have to be included in the imputation model, otherwise results would be far from being reliable.

2.4.3 BOOTSTRAP

The Bootstrap is another technique used to deal with missing data. A lot of bootstrapping applications have been developed in the field of missing values (Efron and Tibshirani, 1994; Schomaker and Heumann, 2016). Bootstrapping relies on random sampling with replacement. The basic idea of the bootstrapping is making inference about a population from a sample of data, which is resampled with replacement so to be able to make another inference on data coming from the same sample of a population. The process is population → sample →

resample. However, as the population is unknown, the true error in a sample statistic against its population value is unknown too. But, in bootstrap-resamples, the quality of inference of the true sample from resampled data is actually measurable, since the true sample itself acts like the population (Efron and Tibshirani, 1994). The inference of the true probability, considered the original data, is treated as analogous to the inference of the empirical distribution of the resampled data.

In order to make an example, let's imagine to be interested in measuring the average weight of people in Europe. Since the weight of every citizen cannot be easily measured, a sample of size N is extracted from the entire population and the average weight is computed. Only one inference can be done from this sample: bootstrapping it with replacement, it is easy to obtain another sample of the same size and obtain another inference. Repeating this procedure, a lot of times, it is possible to build confidence intervals to the parameter being inferenced (DiCiccio and Efron, 1996).

The bootstrapping technique presents the obvious advantages of being very simple both in theory and in practice and of being distribution-independent. Indeed, about the first advantage, it is a straightforward way to derive estimates, standard errors and confidence intervals also in the case of complex estimators of complex parameters (Efron and Tibshirani, 1994). Whether it is impossible to know the true confidence interval of a parameter of a distribution, the bootstrapping is asymptotically consistent and more accurate than other techniques using the sample variance to build the intervals. Moreover, about the second advantage, bootstrapping works because an indirect method to assess the properties of a distribution. Bootstrapping is also used to account for the distortions caused from a sample that could not be perfectly representative of the whole population.

Some limitations however come from the strict assumptions that must be made when undertaking a bootstrap analysis, for instance the independence of the samples or that the sample resampled is big enough to have zero probability of being exactly the same of the sample where it comes from.

The bootstrapping technique can be also applied to solve missing data problems. Indeed, as it is independent from the distribution of the population, it is also independent from the missing data mechanism: it can be easily applied to any missingness scenario.

For instance, let's consider F as a population of units Y_t, where $t = 1,2, \dots, T$, with T considered a big value. A missing data process, defined O_t results in a population G of partially observed values. If the value of a parameter θ_F of the population F has to be inferred, a random sample o of size n from G has to be considered, such that $o = (o_1, o_2, \dots, o_n)$. Some of those values

could be missing too. The empirical distribution $\hat{G}$ of o is then used to estimate θ_F, thanks to $\hat{\theta} = t(\hat{G})$ (Efron, 1994). From this point on, the bootstrapping procedure repeats the actual sampling, inference and estimation steps beginning with $\hat{G}$ which, unlike G is known. Unlimited number of bootstrap replications can be done and each of them involves drawing a bootstrap sample o^* from $\hat{G}$ building the empirical distribution corresponding to the o^* and calculating $\theta^* = t(\widehat{G^*})$.

The Figure 3 below, extracted from Efron (1994), shows in detail the process being descripted. The missing data problem does not affect directly *F*, rather than *G*. But, applying the bootstrap on these data, the missingness can be easily handled thanks to the number of inferences that can be done about the population through the sample-resample procedure.

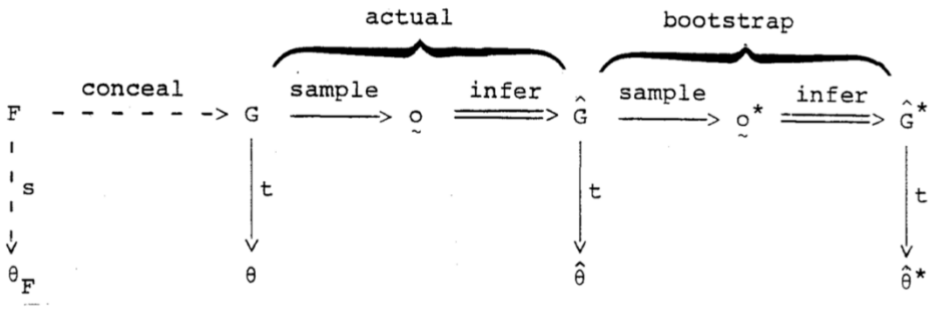

Figure 3: Bootstrap procedure in a concealment process

Bootstrapping techniques can be used also if, instead of only estimating the parameters of a distribution, the missing values themselves have to be filled in. So, bootstrapping could also be one of the steps of an imputation technique. Indeed, if considering a variable $Y_{t\,com}$, such that $Y_{t\,com} = (Y_{t\,obs}, Y_{t\,mis})$, for each of the m imputed variables $y_{t\,m}$, n bootstrap samples are drawn. Therefore, for each of the *m x n* datasets, the standard error of $\hat{\theta}_m$, i.e. the parameter being estimated from each imputed variable, can be easily computed (Schomaker and Heumann, 2016; Shao and Sitter, 1996). In this case, the bootstrap is used to assess the goodness of the process of multiple imputation.

2.4.4 EMB ALGORITHM

In this paragraph, an algorithm developed by Honaker and King (2010) is described. This method combines, in order to take the best of all techniques, the Expectation-Maximization approach and the Bootstrap, in the context of the Multiple Imputation. This method was also built in order to take into account the trends in time and the shifts in space that a Time Series Cross Section (TSCS) dataset presents, which represents an advantage to this method. The idea of this algorithm arose from the computational difficulty in taking random draws of the

parameters of the variable from its posterior density, in order to represent the estimation uncertainty of the problem caused by the multiple imputations. The difficulty derives from the huge number of draws of parameters to be taken if considering a TSCS database. This step of the procedure is therefore replaced by a bootstrapping algorithm, a much faster application. Moreover, the other algorithms previously used to this aim like the Imputation-Posterior (IP) Approach and the Expectation Maximization importance sampling (EMis) require hundreds of lines of computer code to implement, while bootstrapping can be implemented in just a few lines. Furthermore, the variance matrix of the parameters does not need to be estimated, importance sampling does not need to be conducted and evaluated as in EMis (King et al., 2001) and Markov chains do not need to be burnt in and checked for convergence as in IP.

In detail, the so-called EMB algorithm starts drawing *m* samples of size *n* with replacement from the data y_t, being *m* the number of imputations that will be done in the multiple imputation part of the process. This first step is done through the bootstrapping technique. After obtaining these samples, the Expectation-Maximization algorithm produces reliable point estimates of the set of parameters being considered. Finally, for each set of estimates, the original sample unit is used to impute the missing data through the estimated parameters, keeping the original observed values (Honaker and King, 2010). The final result is a set of *m* multiply imputed datasets that can be combined following Rubin's (1987) original rules, as already explained in the paragraph about the Multiple Imputation.

As explained by Efron (1994), the bootstrapped estimates of the parameters can be used in place of the draws from the posterior density because they have the right properties. Moreover, bootstrapping has lower order asymptotics than both IP and EMis approaches.

The EMB algorithm is a powerful technique because it combines the reliability of the Bootstrap together with the EM approach and the completeness of the Multiple Imputation. However, since it is thought to work with TSCS data, in order to apply it to a univariate time series, some adjustments in the software package Amelia II, i.e. the R-package created by Honaker and King (2010) to implement the EMB algorithm, have to be done. The procedure gives good and reliable results, especially if compared to those coming from the application of the single techniques being combined in the EMB algorithm.

2.4.5 METHODS THAT DEAL WITH MNAR DATA

When the MCAR and the MAR assumptions do not hold, so the mechanism of missingness has to be included in the model being used to handle the dataset, the most part of the previous

methods cannot be applied. Even if the bootstrapping for instance could be used to estimate the parameters of the distribution without caring about the missingness mechanism, it is always safer to first try to handle the non-ignorable missingness of the variable because it could contain a lot of information about the whole process generating the series.

There are two fundamentally different ways to include the missingness mechanism to the model: selection models and pattern-mixture models.

Selection models were first introduced by Heckman (1976) and they consist in two main steps: first, a distribution for complete data is specified and then a way in which the probability of a value being missing depends on the data is proposed. For instance, one could assume a normal distribution for the logarithm of the income in the population and a logit regression relating everyone's probability of responding to his/her own log-income (Honaker and King, 2010). Therefore, the selection model builds a joint distribution for the complete dataset $y_{t\ com}$ and the missingness vector *M*: a marginal distribution for $Y_{t\ com}$ and a conditional distribution for *M* are built, such that

$$P(Y_{t\ com}, M;\ \theta, \xi) = P(Y_{t\ com}; \theta) P(M / y_{t\ com}; \xi) \qquad (14)$$

where θ is the set of unknown parameters of the complete dataset and ξ is the set of unknown parameters of the distribution of the missingness mechanism, as seen in the previous sections. Finally, applying Equation (4), the likelihood function is obtained. In typical applications of selection models with dropout (Diggle and Kenward, 1994), for instance, researchers assume for the complete data $Y_{t\ com} = Y_1, Y_2, \dots, Y_T$ a normal distribution and assume that the probability of dropout at time *t* follows a logistic regression on both the previous and the current responses y_t, where $i = 1, \dots, t$, but not on the future responses.

Pattern-mixture models instead were first introduced by Little (1993) and work in a different way with respect to the selection models. This class of models classifies the variables through their missingness and describes the observed data within each group. It can be easily written as follows

$$P(Y_{t\ com}, M;\ \theta, \xi) = P(M; \eta) P(Y_{t\ com} / M; \nu) \qquad (15)$$

where η indicates the proportion of population in the various missingness groups and ν indicates the parameters of the data within the groups.

However, these methods present some limitations, such as the unverifiable assumptions underlying the estimation of ν, since a portion of data is hidden in every group presenting missing data. These assumptions are called "identifying restrictions" by Little (1993). Pattern-mixture models do not posit strict restrictions about the missingness mechanism; rather, they

want to describe the observed responses in every identified group and want to export this behaviour to the unobserved portion of the data.

To conclude, both the selection models and the pattern-mixture models handle the MNAR data problem within the model built for the entire dataset. However, when dealing with dataset in which MNAR situations can be anticipated, it is always better to mitigate its effect by changing a bit the study design. For instance, asking the respondent which is the probability of nonresponding to further attempts of data collection could add a covariate to the study which could convert the MNAR situation to a MAR one. If this scenario is achieved, it would immediately be easier to handle the eventual missingness.

2.5 SUMMARY OF THE LITERATURE REVIEW

After completing the critical literature review, it is important to summarize the most important concepts and topics of interest for this investigation, highlighting the most relevant aspects leading to the formulation of the propositions which will be used as a guide in the core of this work of investigation.

In the context of the time series analysis and definition (see Section 2.1), it emerged that the process generating a time series has to be caught in order to obtain good estimates and forecast. Moreover, in Box and Jenkins' (1970) procedures, also the back-forecast process was used to fill in the previous values of the observed ones in a time series. Therefore, if the model fits to the series, both forecast and backcast processes can be used on a time series. The first proposition arises:

P1: "The ARMA process generating a time series can be used to impute the missing values in a time series, using either a forecasting approach or a backcasting one".

In the scenario of the analysis of the missingness in a time series, it is difficult to identify for certain the pattern of the missing data since there is no unique and effective procedure to this aim (see Section 2.2). Therefore, the procedures being used to handle the missing data usually have to assume a pattern of missingness, otherwise they cannot hold. The second proposition arises:

P2: "A method has to rely on assumptions about the missingness pattern to be trustworthy".

Analysing the Old Methods, even though these theories were mostly applied to multivariate datasets where the variable time was not considered (see Sections 2.3.1 and 2.3.2), the comments can be extended to the time series field. These methods are not broadly used because

the use implies the modification of the parameters of the variable, the underestimation of the uncertainty linked to the application of the method or the loss of a lot of information about the process. In a time series application, these limitations are amplified because of the nature of the series. Therefore, a good method to be applied does not modify the underlying process generating the time series. The third proposition arises:

P3: "The parameters defining the ARMA process which generates a time series do not have to be modified by the imputation method".

Finally, from the analysis of the New Methods used to handle the missingness in a dataset, it emerged that the most urgent aspect to deal with is the estimation of the parameters of the variables considered in the study. Only with a good parameters estimation it is possible to impute good quality data when a missingness is observed. Therefore, as seen in the application of the EMB algorithm (see Sections 2.4.1, 2.4.2 and 2.4.4), which combines the Bootstrapping technique together with the EM algorithm in the context of the Multiple Imputation, it has been proven that the more reliable way to estimate the parameters of the distribution of a variable is the iterative procedure of Expectation-Maximization, which brings to the convergence of the estimates. Estimating the set of parameters thanks to this algorithm does not modify in any way the distribution of the variable. Therefore, since the EM algorithm is also independent from the missingness pattern, it is possible to extend this application to the time series field. Finally, the fourth proposition arises:

P4: "An iterative mechanism of parameters estimation until convergence preserves the characteristics of the ARMA process generating a time series".

CHAPTER 3: METHODOLOGY

This chapter introduces the research methodology of this work, which aims to describe how the investigation will be conducted, highlighting the methods and approaches used to get an answer to the research questions, on the basis of the propositions emerged from the literature review.

3.1 LITERATURE REVIEW

The systematic critical literature review was the first step of this dissertation. The output of the literature review was an in-depth analysis of the existing approaches to the time series with missing data problem (see Sections 2.1, 2.2, 2.3 and 2.4), from which some propositions have emerged (see Section 2.5). The propositions emerged from the literature are assumed as guidelines to put into writing a new approach to the missingness problem in univariate time series.

The new procedure finds its pillars also on some studies which have been conducted and, on some theories, explained by important researchers. For instance, the approach and the results of the study of Kihoro and Athianky (2013) involving the use of ARIMA models to the missing data imputation was used as a starting point to this investigation method. Furthermore, the Expectation Maximization approach was added to improve the outcome.

3.1.1 ARMA MODELS DEFINITION

This Section is dedicated to an insight about the ARMA models used as imputation step of the imputation algorithm. Therefore, in the following paragraphs the general definition of AR(p) processes, MA(q) processes and ARMA(p, q) ones is presented.

3.1.1.1 AUTOREGRESSIVE PROCESS OF ORDER p

Y_t is said to be an autoregressive process of order p (AR(p)) if it is a weighted sum of the last p observations plus a random error $\varepsilon_t \sim N(0, \sigma_\varepsilon^2)$. An AR($p$) can be written as follows

$$Y_t = c + \emptyset_1 Y_{t-1} + \emptyset_2 Y_{t-2} + \cdots + \emptyset_p Y_{t-p} + \varepsilon_t \quad (16)$$

or, using the backward shift operator B such that $BY_t = Y_{t-1}$ (Box et al., 2015), an AR(p) process is equal to

$$\left(1 - \emptyset_1 B - \emptyset_2 B^2 - \cdots - \emptyset_p B^p\right) Y_t = c + \varepsilon_t \quad (17)$$

A particular example of an AR process is the first-order case AR(1), where every observation only depends on the previous one. It is given by

$$Y_t = c + \emptyset Y_{t-1} + \varepsilon_t \quad (18)$$

and it is said to be stationary if $|\emptyset| < 1$. If $\emptyset = 1$, the process presents a unit root and it is called a "random walk".

It is possible to identify the order of an AR process through the Partial Autocorrelation Function (PACF), since an AR(p) process shows the useful property of having the values of PACF under the threshold for every lag greater than p. The ACF instead decreases exponentially to 0.

Every stochastic stationary process can be written in the form of an AR(p), where p could also be an infinite number, according to the Wold Theorem (Wold, 1948). This theorem says that a stationary process can always be split in a deterministic and a stochastic part, where this last one is a linear combination of infinite white noise processes. The strength of this decomposition is that every model can be written as a pure model, where only the p order has to be set. However, even if the two models (original and decomposed ones) would be equivalent, the model identification error should not be underestimated.

3.1.1.2 Moving Average Process of order q

Y_t is said to be a moving average process of order q (MA(q)) if it is a weighted sum of the last q shocks, given that $\varepsilon_t \sim N(0, \sigma_\varepsilon^2)$. A MA($q$) process can be written as follows

$$Y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \quad (19)$$

or, using again the backward shift operator B, it can be written in the alternative form

$$Y_t = \left(1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q\right)\varepsilon_t \quad (20)$$

A moving average process is always stationary, as first demonstrated by Box and Jenkins (1970). The PACF of a moving average process decreases exponentially to 0, while the ACF shows values over the threshold for q lags.

3.1.1.3 ARMA(p, q) Processes

A combination of an autoregressive process and a moving average one can be modelled through an ARMA model. The importance of an ARMA process is that many real datasets can be approximated through fewer parameters instead of using pure AR of MA models (Chatfield, 2000). An ARMA(p,q) process can be written ad follows

$$Y_t = c + \emptyset_1 Y_{t-1} + \ldots + \emptyset_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q} \quad (21)$$

where $\emptyset_1, \emptyset_2, \ldots, \emptyset_p$ are the parameters of the AR model and $\theta_1, \theta_2, \ldots, \theta_q$ are the parameters of the MA. This process can be written through the backward shift operator B, as previously, in the form

$$\emptyset(B) Y_t = \theta(B) \varepsilon_t, \quad (22)$$

where $\emptyset(B) = \left(1 - \emptyset_1 B - \cdots - \emptyset_p B^p\right)$ and $\theta(B) = \left(1 - \theta_1 B - \cdots - \theta_q B^q\right)$.

3.1.2 Expectation Maximization Algorithm

The key idea of EM is to solve iteratively an easier complete-data problem in order to solve a difficult incomplete-data scenario. Each iteration of the EM algorithm involves two steps which we call the Expectation step (E-step) and the Maximization step (M-step). In the first one, the missing data are filled in with a best guess under the current estimate of the unknown parameters. In the second one, parameters are re-estimated from both the observed and the filled-in data. The iterations continue until the parameters converge. This may take some time, mostly if missing data are a lot or if the initial best guess is more than a standard error away from the real parameters.

One way to conceptualize the convergence of the algorithm, at least to a local maximum of the likelihood space under regularity conditions (McLachlan and Krishan, 2008), is that the parameters estimated at the end of each iteration θ_t can be considered as the weighted sum between the true parameters contained within the observed data θ^{MLE} and the parameters θ_{t-1} coming from the expected values (filled-in data). The previous iterations parameters θ_{t-1}, used to build the expectations, could be far from the true values but they will only be given partial weight in the construction of θ_t (Honaker and King, 2010). Each sequential value of θ will be every iteration closer to the truth. The main advantage of the algorithm is that its convergence reliability can be demonstrated. Indeed, in every iteration the loglikelihood $l(\theta/y_{t\,obs})$ increases and, as it is bounded, the sequence $l\left(\theta^{(t)}/y_{t\,obs}\right)$ converges to a stationary value of $l(\theta/y_{t\,obs})$ (Little and Rubin, 1987; Tagare, 1988).

However, there can be two drawbacks in the use of the Expectation-Maximization algorithm. The first one is that the algorithm could be very slow to converge, with large fractions of missing information (Little and Rubin, 1987). The second one is that in some problems the M step could be difficult to implement. Different types of EM algorithm are used to solve these limitations.

In this investigation, the EM algorithm is implemented within the imputation mechanism, as the way to estimate the parameters of the series through which the implementation is performed.

3.2 Simulations Campaigns

After assessing the important information obtained from the literature, with a particular insight on the ARMA models, the following step was to build an algorithm for the specific case of the

univariate time series with missing data, on the basis of the Propositions and of the literature reviewed. The algorithm was tested thanks to the simulation campaigns approach, which aimed to verify if the phenomenon being analysed met the initial conditions of the process and which aimed to give to the results obtained the robustness needed to trust them and extract conclusions. The simulations were all printed in order not to lose precious information which could influence the overall evaluation of the algorithm. Moreover, since the algorithm was tested on more than a process, a needed comparison between the different outputs was performed.

In this work, no classic data collection was performed. Indeed, as said before, the data to be analysed were simulated through the software R, the analysis tool of this work. The simulations campaigns evaluation is the last step of the research methodology. However, last but not least, it represents the most important and delicate part. Indeed, it is useful not only to assess the goodness of the algorithm, although it is its first aim, but also to open the door to new researches and investigation, on the basis of the findings of the assessment performed.

3.3 Analysis Tool

The software used for the whole investigation application is R, which is a programming language and free software environment used for statistical computing and graphics. It is supported by the R Foundation for Statistical Computing. The two main applications of R are the development of statistical software and the analysis of the data. User-created *packages* improve the performances of R. In this work, packages like *forecast* and *stlplus* were used, as well as the more classic ones like *zoo*, *tseries* and so on. The Comprehensive R Archive Network (CRAN) includes all the packages which have been developed and implemented in R, together with a detailed explication with examples about how to apply them. This is a really useful tool for new and experienced users. See Venables et al. (2018) book "An Introduction to R" to have an overview about the software.

R has been chosen for the application part of this investigation since it allows to work with the time series and many packages and functions support missing data situations. Moreover, R communities and forums are big enough to find whatever is necessary.

CHAPTER 4: APPLICATION

This chapter presents the application of the algorithm built on the literature analysed in Chapter 2. In the first section of this chapter, the conceptual model at the basis of the algorithm is explained, with a highlight on the procedure followed to assess its goodness and, on the metrics (criteria), used to evaluate the results. In the second section of the chapter, the practical application on the software R is showed, together with the explication of the code used to perform the whole algorithm and assessment. In the third section, the results of the evaluation of the algorithm are discussed and summarized. Finally, the fourth and final section presents the steps and the code on R that the users have to follow in order to apply the algorithm on their time series.

4.1 THEORETICAL APPLICATION

This section aims to present the approach followed in the application part of the investigation. It is divided in five subsections, which represent the main steps of the procedure. It is important to highlight that, in the context of the implementation of a new approach to deal with the missingness in time series problem, a simulation campaign setting was employed. The simulations campaigns allow the researcher to test more than one application of the same algorithm, which gives to the results the needed robustness. Therefore, for each setting of conditions, 1000 simulations were run. The results are shown and discussed in Section 4.3.

4.1.1 TIME SERIES SIMULATION

The first step of the procedure was to find the time series to analyse. This investigation methodology did not employ real time series data because it would have been far slower to find thousands univariate time series responding exactly to the required parameters. Moreover, the simulation allows to exactly detect the precision of the procedure, thanks to the knowledge of the stochastic process generating the time series.

The stochastic processes taken into consideration in this work were defined by two different sets of parameters for each ARMA model simulated (AR(1), AR(2), MA(1), MA(2), ARMA(1,1), ARMA(2,1), ARMA(1,2) and (ARMA(2,2)). In the following Table 1, the employed sets of parameters are shown.

		Model Parameters			
		ϕ_1	ϕ_2	θ_1	θ_2
AR(1)	**1**	0,6			
	2	0,8			
AR(2)	**1**	0,9	-0,5		
	2	0,7	-0,6		
MA(1)	**1**			0,6	
	2			0,8	
MA(2)	**1**			0,5	0,3
	2			1,2	-0,3
ARMA(1,1)	**1**	0,6		0.6	
	2	0,8		0,8	
ARMA(2,1)	**1**	0,9	-0,5	0,6	
	2	0,7	-0,6	0,8	
ARMA(1,2)	**1**	0,6		0,5	0,3
	2	0,8		1,2	-0,3
ARMA(2,2)	**1**	0,9	-0,5	0,5	0,3
	2	0,7	-0,6	1,2	-0,3

Table 1: Parameters sets for each ARMA model

For every ARMA model showing an AutoRegressive part, the parameters have been chosen taking into consideration the stationarity of the model. It was considered important to check whether the algorithm worked better with more stationary time series rather than with less stationary ones, for every ARMA model. For instance, the AR(1) and ARMA(1,1) models where $\emptyset_1 = 0{,}6$ are more stationary than the others with $\emptyset_1 = 0{,}8$ since their root is more distant from 1. The root is calculated solving the Equation (23)

$$1 - \phi_1 z - \cdots - \phi_p z^p = 0 \qquad (23)$$

Applying the same reasoning to the other ARMA models, the set of parameters $\emptyset_1 = 0{,}9$ and $\emptyset_2 = -0{,}5$ represents a more stationary scenario than the set of parameters $\emptyset_1 = 0{,}7$ and $\emptyset_2 = -0{,}6$. As regards the MA coefficients, since the MA processes are always stationary, they have been chosen among the sets of coefficients found in the literature.

In this exploratory work of investigation, only stationary ARMA processes have been taken into consideration. This work represents indeed the first application of an algorithm showing these features, therefore it was considered more appropriate to test it on the simplest models with a limited number of parameters (≤ 4), before applying it on more complicated models. Then, the time series have been chosen stationary by default. Further researches could focus on the reliability of this procedure on time series showing a trend or a seasonal effect, if it can be concluded that the algorithm works well on the basic models.

4.1.2 Model Identification Step

In the context of this investigation, the identification step of the model generating the time series will be put apart. This investigation is mainly orientated toward the missing data imputation step, which is the reason why just a few guidelines will be left to the analyst dealing with this problem. Further researches will investigate about the model identification, to be added as first step of the developed imputation algorithm.

In order to identify the ARMA process generating the time series under analysis, a good practice is to check the literature to find out whether that type of data is usually determined by a specific class of models (for instance, if it follows a pure AR process in the most part of the cases). This would narrow the range of models to be tried in order to identify the closest one. If this empirical approach does not give any result, the analyst could try to compute the ACF and the PACF ignoring the missing data to see if there is an evidence of a pure AR process or a pure MA process.

If the user is not able to have an insight about the process generating the series through these tools, more models have to be tried, starting from the pure ones. The models can be compared through the Akaike Information Criterion (AIC), which estimates the relative quality of statistical models for a given set of data. It estimates the relative information lost by a given model: the less information it loses, the higher its quality (Akaike, 1974). Therefore, the AIC computed on different models provides a mean for model selection.

4.1.3 Missing Data Simulation

After completing the first step of the procedure, i.e. the time series simulations, the following step was to generate the missing data to be imputed. The literature showed that when dealing with MNAR data, since there is a specific reason why those data are missing, it is really difficult to catch the real value throughout an imputation algorithm. Results of MNAR data imputation are usually biased and far away from the real values. Due to this finding, this pioneer investigation focused only on MAR and MCAR data imputation, trying to broaden the application of the algorithm indiscriminately on both situations, even in the case of non-identified missingness mechanism. This choice is in line with the Proposition 2, stating that an assumption about the missingness pattern has to be made, since MNAR situations are treated differently from the other two, but it is also in contrast since the same algorithm is being applied on both MAR and MCAR data situations.

If the imputation procedure allows to find values close to the true ones in both MAR and MCAR data scenarios, then it could be applied (with some adjustments if necessary) also to MNAR

situations. Further researches could focus on this topic, so to be able to deal with the entire range of cases.

Two different missingness rates for every missing value pattern were hypothesized: 5% and 10%. These two rates were used to check the robustness of the algorithm under assessment. In the literature, no more than the higher (10%) percentage of missing data is handled, since for more than this amount of non-observed values the imputation is usually biased. However, if the algorithm allows the imputation of even the 10% of missing data showing a small error, this opens a window on the possibility of broadening the analysis to higher missingness rates.
Once defined the two missing data patterns and the two missingness rates for each pattern, it is clear that for every ARMA process hypothesized there will be 8 simulation blocks. Indeed, 2 conditions settings for every missingness rate for every missing data mechanism gives 4 simulations blocks for each set of parameters. Since two sets of parameters are considered for every model, 8 simulations blocks will be run for every ARMA model. Finally, since 8 different ARMA processes are considered, the overall number of simulations blocks was 64.

4.1.4 Algorithm Implementation

After the identification of the simulations to be run, the algorithm itself was built. Its basis rests on the study of Kihoro and Athiany (2013), which involved the use of the ARIMA models to impute the missing values. The two researchers, after having identified the ARIMA model which seemed to fit better with the series, estimated the parameters of the observed part of the variable. Through these parameters estimates, they filled in the first missing value. Then, the parameters of the series were estimated again, considering also the filled-in value. This way, the imputation of the second missing value was done and this iterative procedure was repeated until all the missing values were filled in.
The idea at the basis of the algorithm of this investigation is to use the parameters estimated through the observed part of the time series to fill in the missing data, as in the work of Kihoro and Athiany (2013) but, instead of imputing one missing value per step and then re-estimate the parameters, the non-observed data are imputed altogether, in one single step.
A common point with the study of the two researchers is that not all the missing values are imputed following a forecast approach. As in the finding of the Proposition 1 indeed, also the backcast can be used to fill in the missing data, according to the procedure explained in the work of Box and Jenkins (1970). The backcast approach is employed because, when simulating the missingness, some random missing values were found among the first observations of the

time series, i.e. the oldest ones. Therefore, in order to perform the imputation through an AR process, even older values were needed, but it was not possible to obtain them without simulating other values from the already simulated time series. The backcast solves this problem. Its application, considering for instance an AR(p) process, starts from the Equation (16), solved by Y_{t-p}, that is the furthest observation of the series. Equation (16) becomes as follows

$$Y_{t-p} = \frac{-Y_t + \emptyset_1 Y_{t-1} + \emptyset_2 Y_{t-2} + \cdots + \varepsilon_t}{\emptyset_p} \quad (24)$$

A different situation to handle arises when considering a Moving Average process, where the missing observations have to be imputed through the irregular part of the time series. The issue is that, since the observation is missing, also its irregular part will be missing. Therefore, following Box and Jenkins (1970) assumption of $E(\varepsilon_t) = 0$, the irregular part of missing values in the first observations of the time series was always considered to be equal to 0. Starting from this assumption, the backcast of the ARMA models presenting either the AR or the MA parts was only composed by the AutoRegressive component, since the residuals were considered 0 by default. The first year of observations was set as threshold between the backcast imputation approach and the forecast approach. Therefore, considering monthly data, the missing values occurring among the first 12 observations were backcasted, the others were forecasted.
Moreover, as explained in Section 4.1.1, only stationary time series were considered in this investigation. Therefore, since no difference is needed to turn the time series into stationary, the intercept of the time series can be added to the model. When instead the model is considering a difference transformation of the series, the intercept doesn't fit (Dickey and Fuller, 1979).

The most interesting part of the procedure introduced by this dissertation is not, however, the use of the ARMA models to impute the missing values. Indeed, in order to assure the robustness of the imputations performed, as suggested by Proposition 4, the Expectation-Maximization approach was implemented. The idea of the EM algorithm applied to this procedure was born in order to limit the uncertainty linked to the estimation of the parameters of the time series. As already said in the Section 3.2.1, the main strength of the EM algorithm is that, thanks to the sequence of the Expectation and the Maximization steps, it is able to bring the parameters of the variable under analysis to convergence. In this application, the E-Step is simply represented by the estimation of the parameters before every imputation; the M-Step instead follows the missing data imputation and it is represented by the re-estimation of the parameters of the time series considering also the filled in missing values. After this step, another E-Step is performed

and new values are imputed replacing the already filled-in values through a new set of estimated parameters. These two steps are repeated iteratively until the parameters of the last two steps converge.

The convergence of the parameters allows the user to be quite sure that imputing again some of the values of the time series, this would be defined by the same parameters in any case. This procedure can be considered a way to force the ARMA model to fit the series, with a set of parameters that does not change.

To conclude this part of the work, when the parameters converge, the iterations stop and the residuals of the fitted ARMA model can finally be checked. In order to use a fitted ARMA model, indeed, the residuals it produces should follow a white noise process and should not show any autocorrelation. To this aim, the Ljung-Box test (Ljung and Box,1978) is performed at the end of every cycle of iterations. This test belongs to the Portmanteau class of tests and its H_0 can be defined as follows:

H_0: The data has been generated by a white noise process

This hypothesis means that the correlations in the population from which the sample is taken are 0, therefore it can be concluded that any observed correlations in the data result from randomness of the sampling process, i.e. a white noise process. If the p-value of the Ljung-Box test is higher than 0,05, the H_0 cannot be rejected therefore the residuals of the model can be considered a WN process and the model being tested can be used for subsequent analysis.

In the following Section 4.1.4.1, the steps of the whole algorithm are outlined.

4.1.4.1 Algorithm Steps

The steps of the algorithm can be outlined as follows:

1. Parameters estimation on the basis of the only observed data;
2. Missing data imputation using the estimated parameters and the ARMA model hypothesized. For instance, considering a AR(1) process, if y_t is missing, it is imputed as $y_t = \phi_1 y_{t-1}$ if t does not fall among the first 12 observations (forecasting approach), or as $y_t = \frac{y_{t+1}}{\phi_1}$ if t falls among the first 12 observations (backcasting approach);
3. Parameters estimation considering also the imputed missing values (E-Step);
4. Imputation of the already filled-in values, through the new set of parameters;
5. Parameters re-estimation (M-Step);
6. Re-imputation of the already filled-in values;

7. Repetition of the steps 3-6 until the parameters estimated in the steps 1 and 3 or in the steps 3 and 5 converge;
8. Ljung-Box Test of the residuals.

4.1.5 Assessment Metrics

The final step of the procedure is to analyse the performances of the model being implemented trough the evaluation metrics. An important aspects of evaluation metrics is their capability to discriminate among model results. Two metrics have been chosen to this aim: the Root Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE). They are two very common metrics which are used together in order to catch different aspects of the model under evaluation (Moritz et al., 2015). The RMSE, given that $\bar{y}_t$ is the time series with the imputed values and y_t is the time series with the respective true values, is calculated as follows in Equation (25).

$$RMSE(\bar{y}_t, y_t) = \sqrt{\frac{\sum_{t=1}^{n}(\bar{y}_t - y_t)^2}{n}} \tag{25}$$

By definition, the RMSE should be as small as possible. This would mean that the difference between the imputed values and the true ones is small, i.e. the imputation algorithm is reliable enough.

The MAPE, given that $\bar{y}_t$ is again the time series with the imputed values and y_t is the time series with the respective true values, is calculated as follows in Equation (26).

$$MAPE(\bar{y}_t, y_t) = \frac{\sum_{t=1}^{n}(\frac{|\bar{y}_t - y_t|}{|y_t|})}{n} \tag{26}$$

In this investigation the MAPE is used to compare the simulations blocks, to perform a sort of sensitivity analysis and test the limitations of the application. However, MAPE values should be, as said for the RMSE, as small as possible, always with respect to the values interval of the time series, in order to be comfortable to say that the algorithm can be used in that specific case.

4.2 Software Application

This Section is dedicated to the practical application on R, the software used for the whole algorithm assessment. Therefore, for each of the various steps composing the procedure that brought to the algorithm itself, the code employed will be presented and explained. In order not to bore the reader, only the application of the procedure considering an AR(1) process, where $\emptyset_1 = 0{,}6$, will be reported. The other codes applications can be found in the Appendix.

4.2.1 TIME SERIES SIMULATION

The first code lines were dedicated to the simulation of the time series, which was created following a AR(1) process where $\emptyset_1 = 0{,}6$. The number of observations of the time series is 181 and it starts from January 2002 and ends on January 2017. The letter *n* represents the length of the time series.

```
ts.sim <- arima.sim(list(order = c(1, 0, 0), ar = 0.6), n = 181)
n <- length(ts.sim)
ts.orig <- ts(ts.sim, start=c(2002,1), end=c(2017,01), freq=12)
```

After defining the time series, the ARMA model fitting the original series is estimated. This allows to print the coefficients of the process, in order to compare them with the estimated ones at the end of the iterations which aim to reach the parameters convergence.

```
ARIMAorig <- arima(ts.orig, order=c(1,0,0))
```

4.2.2 MISSING DATA SIMULATION

Once obtained the time series to be analysed, the missing data have to be simulated. When simulating MAR data, a condition linking the observed data to the probability of missingness has to be chosen. In this investigation, the chosen condition that generates MAR data is that the difference between the observation lagged 6 times and the observation lagged 12 times is higher than 0,2 and that the difference between the observation at the time *i* and the observation lagged 6 times is higher than 0,1. If both conditions are respected, the data at time *i* would be set as missing. This means that it is more probable to find a missingness if the difference between these lagged observations increases over time. It is possible to talk about probability of missingness because a condition stopping this process is set: when the number of missing data simulated reaches the 5% or the 10%, dependently from the block of simulation considered, the process creating missing data is stopped. Therefore, there will be some data that are not missing even if the conditions about the differences are respected. The code lines below are the ones employed to generate MAR missing data. In the *if* condition, the two conditions to generate Not Available (NA) data and the condition to stop the process are set.

```
ytMAR <- ts.orig
for(i in seq(from=(frequency(ts.orig)+1), to=(n-(frequency(ts.orig))), by=2)){
  dif1 <- ytMAR[i+6]-ytMAR[i+12]
```

```
dif2 <- ytMAR[i]-ytMAR[i+6]
numNA <- sum(is.na(ytMAR))
if(!is.na(ytMAR[i]) | !is.na(ytMAR[i+6]) | !is.na(ytMAR[i+12])) {
if(dif1>0.2 & dif2>0.1 & (numNA < ((0.05*n)-1))) {
 (ytMAR[i] <- NA)}
 next
 }
}
```

When instead MCAR data are simulated, the process which generates them is far easier to compute since there is no link between the observed and the missing data as in MAR situations. Therefore, missing data occur randomly. Once defined the missingness rate to be obtained, a sample of that length is extracted from an interval of numbers going from 1 to *n*, being *n* the length of the time series considered. The values of the time series belonging to the sampled observations were substituted by NA data, i.e. missing ones. The code lines below are the ones employed to generate MCAR missing data

```
originalyt <- ts.sim
  ytMCAR <- ts.sim
  miss <- as.integer(0.05*(n))
  mcar <- sort(sample(1:(n), miss))
  ytMCAR[originalyt <- mcar] <- NA
```

4.2.3 Algorithm Implementation

Once the time series containing some missing values has been defined, a decomposition in its components has to be done in order to catch the irregular part. The irregular part is needed because, in the context of a MA process, each missing observation to be imputed needs the residuals to be calculated. The STL decomposition is performed to this aim. STL means Seasonal and Trend decomposition using Loess, where Loess is a method for estimating nonlinear relationships. The STL method was developed by Cleveland et al. (1990), it handles every seasonal period and performs an additive decomposition. Modifying the series components taking its logarithm, it is possible to do also a multiplicative decomposition. In this work, an additive decomposition was employed. In order not to bore the reader, the code lines from now on will only refer to a MAR situation but the procedure is exactly the same for MCAR data.

```
MAR.ts <- ts(ytMAR, start=c(2002,1), end=c(2017,01), freq=12)
ts.stl <- stlplus(MAR.ts, s.window="per", na.action=na.pass(MAR.ts))
residuals <- ts(ts.stl$data[,4], start=c(2002,1), end=c(2017,01), freq=12)
```

After isolating the irregular component of the series, the iterations that should bring the parameters of the time series to convergence can start. The maximum number of iterations to be performed, set to handle the unfortunate case of non-convergence, is 40, therefore *itermax* is set to 39. The tolerance, i.e. the difference value between two subsequent sets of parameters which allows the analyst to state the convergence of parameters, is set to 0,00001. Finally, the last preparatory step before starting the iterations is to split the vector *mar* or *mcar* containing the positions of the missing values in two parts: the first one (*mar1* or *mcar1*) includes all the missing values occurring when $t < 12$, which are the missing values found in the first year of observation, and the second one (*mar2* or *mcar2*) includes all the missing values occurring after the first year of observation, when $t > 12$. This operation is done in order to separate the missing values to be imputed through the backcast from the ones to be imputed through the forecast. In particular, the missing data in the positions defined by *mar1* or *mcar1* are backcasted, the others are forecasted.

```
iter <-1
itermax <- 39
tolerance <- 0.00001
newts <- MAR.ts
newts2 <- MAR.ts
mar <- which(is.na(MAR.ts))
lim <- 0
for(i in 1:length(mar)) {
 if(mar[i] < (frequency(ts.orig) + 1)) {
 lim <- lim + 1
 } next
}
mar1 <- mar[0:lim]
mar2 <- mar[(lim+1):length(mar)]
```

Finally, the iterative algorithm can be run. Therefore, the first iteration starts and, as a preliminary E-Step, the parameters of the fitted AR(1) model on the time series still containing missing values are estimated thanks to the *arima* function. The residuals of the series corresponding to the missing values are set equal to 0, since the process is an AutoRegressive

one. In a Moving Average scenario instead, only the residuals corresponding to the missing values included in *mar1* or *mcar1* would have been set equal to 0, the others would have been calculated solving by ε_t the Equation (20).

```
repeat{
 iter <-iter + 1
 AR1 <- arima((newts), order = c(1, 0, 0))
 coeff <- (AR1$coef[1])
 intercept <- (AR1$coef[2])
 residuals[newts2 <- mar] <- 0
 newts[newts2 <- mar1] <- (-intercept + newts[(newts2 <- mar1) + 1] - residuals[(newts2 <- mar1) + 1])/coeff
 newts[newts2 <- mar2] <- intercept + coeff*newts[(newts2 <- mar2)-1] + residuals[newts2 <- mar2]
```

The second part of the first iteration, which represents the real beginning of the EM algorithm, starts with the estimation of the parameters of the AR(1) process generating the series but, in this case, the time series considered by the *arima* function includes the filled-in missing data. This is the real first E-Step of the algorithm. The first iteration ends after the new imputation of the already filled-in missing values. The conditions which stop the loop are two for this AR(1) scenario: the cycle ends if the first estimate of $\emptyset_1$ and the second one converge, i.e. the difference between them is lower than the value of the tolerance, or if the iterations number is reaches its threshold value, that is 40 iterations. If one of these two conditions is true, the loop stops. Otherwise it continues with the following iteration, which repeats the already described sequence of steps. The new estimation of parameters is the M-Step of the algorithm. The code lines below show the second part of an iteration and the *if* condition that stops the loop.

```
 AR1new <- arima((newts), order = c(1, 0, 0))
 coeffnew <- (AR1new$coef[1])
 interceptnew <- (AR1new$coef[2])
 residuals[newts2 <- mar] <- 0
 newts[newts2 <- mar1] <- (-interceptnew + newts[(newts2 <- mar1) + 1] - residuals[(newts2 <- mar1) +
1])/coeffnew
 newts[newts2 <- mar2] <- interceptnew + coeffnew*newts[(newts2 <- mar2)-1] + residuals[newts2 <- mar2]
if((((abs(coeffnew-coeff)) < tolerance)) | (iter > itermax)) {
 break
 }
next
}
```

When the loop ends, the residuals of the last fitted AR(1) model have to be checked through the Ljung-Box test, shown below. For each simulation, a variable representing the number of times a process shows white noise residuals is computed, in order to know at the end of the 1000 simulations how many times the fitted model could have been used. In the same way, a variable counting the number of convergences for every simulation block was defined.

```
LBAR1_0.6_5MAR <- Box.test(AR1new$residuals, lag=30, type = "Ljung-Box", fitdf=2)
if(LBAR1_0.6_5MAR$p.value > 0.05) {
 WNresidAR1_0.6_5MAR <- WNresidAR1_0.6_5MAR + 1
}
if(iter < itermax) {
 convergenceAR1_0.6_5MAR <- convergenceAR1_0.6_5MAR + 1
 }
```

4.2.4 Assessment Metrics

To conclude, the last step of each simulation is the evaluation of the model through the two metrics defined in Section 4.1.5. The Root Mean Square Error and the Mean Absolute Percentage Error were computed as shown in the code lines below, where *ts.orig* is the original time series, the simulated one, and *newts* is the time series with the imputed missing values. The RMSE and the MAPE were then printed in the results matrix, to be analysed afterwards.

```
rmse <- ((sum((ts.orig-newts)^2))/length(newts))^0.5
mape <- (sum(abs((ts.orig-newts)/ts.orig)))/length(newts)
```

4.3 Results and Discussion

4.3.1 Results Structure

At the end of the 8 simulations blocks of each class of models (AR(1), AR(2), MA(1) etc), two kinds of tables containing the results to be analysed were printed:

1. The first class of tables included the output of each setting of conditions for each model considered. Therefore, 8 of these tables were printed for each model. In order to clarify what is meant with the word setting, the 8 settings of the AR(1) class of models were:
    a. $\emptyset_1 = 0{,}6$ with 5% MAR data;
    b. $\emptyset_1 = 0{,}6$ with 10% MAR data;
    c. $\emptyset_1 = 0{,}6$ with 5% MCAR data;
    d. $\emptyset_1 = 0{,}6$ with 10% MCAR data;

e. $\emptyset_1 = 0{,}8$ with 5% MAR data;
f. $\emptyset_1 = 0{,}8$ with 10% MAR data;
g. $\emptyset_1 = 0{,}8$ with 5% MCAR data;
h. $\emptyset_1 = 0{,}8$ with 10% MCAR data.

These tables had as many rows as the number of simulations per block, i.e. 1000, and $2\beta + 4$ columns, where β is the number of parameters to be estimated. Therefore, this table was a 1000x6 matrix if considering AR(1) or MA(1) processes, a 1000x8 matrix if considering AR(2), MA(2) or ARMA(1,1) processes, a 1000x10 matrix if considering ARMA(2,1) or ARMA(1,2) processes and a 1000x12 matrix if considering ARMA(2,2) processes. The number of parameters β is multiplied by 2 because also the original parameters, i.e. the ones estimated when the time series was still complete before simulating the missing data, were printed. Those parameters are printed in order to check whether the new ones are close to them. It is a way to verify if the parameters of the original time series are respected by the imputations: as highlighted by the Proposition 3, this is an important aspect of an imputation method.

The columns of these matrixes contained therefore the original and new parameters, the number of iterations the algorithm took to reach convergence, the RMSE value, the MAPE value and the p-value of the Ljung-Box test;

2. The second class of tables included a summary for every model considered. They are 8x8 matrixes, in which every row represents the 8 different settings for each class of models and the columns contain information about the mean, the variance and the standard deviation of RMSE and MAPE, as well as the convergence percentage rate and the WN residuals percentage rate for every setting of conditions.

The first class of tables will be partially shown along this Chapter, while the second class of tables will be widely commented in the following Section 4.3.2.

4.3.2 DISCUSSION

This section is dedicated to the presentation and the discussion of the results printed in the second class of tables.

Starting from the Table 2 below, it shows the results obtained for the AR(1) model settings.

	AR(1) Model Settings							
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**	**Convergence %**	**WN Residuals %**
0,6, 5% MAR	0,246604209	0,003539795	0,059496172	0,111871255	0,337300263	0,58077557	100%	90,7%
0,6, 10% MAR	0,277659856	0,004938889	0,070277233	0,264787906	14,02592437	3,745120074	100%	90,7%
0,6, 5% MCAR	0,228053402	0,003225485	0,056793358	0,267101823	8,108914001	2,847615494	100%	91,2%
0,6, 10% MCAR	0,326536977	0,003476737	0,058963865	0,57344274	41,80448992	6,465639173	100%	91,1%
0,8, 5% MAR	0,243760903	0,00447949	0,066928993	0,139418024	0,643602758	0,802248564	100%	91,3%
0,8, 10% MAR	0,292017317	0,006470982	0,080442416	0,179265982	0,48005116	0,692857243	100%	89,8%
0,8, 5% MCAR	0,340559263	0,004425825	0,066526876	0,338821001	1,540176262	1,24103838	100%	90,8%
0,8, 10% MCAR	0,343698477	0,004627283	0,068024133	0,379804405	2,339306469	1,529479149	100%	91,3%

Table 2: AR(1) Model Settings Results

For all the AR(1) settings considered, the means of the RMSE are all low values (<0,35), with a very low standard deviation. Furthermore, also the means of the MAPE are low (<0.6) with a much higher standard deviation. This shows that some imputations were far away from the true values, although the 100% rate of convergence. In this situation, users should check if the calculated intercept has a much higher value than the order of the observed part of the time series. It is quite evident, comparing MAPE means and standard deviations, that with MCAR data the algorithm for AR(1) models does not work as well as for MAR data. Furthermore, as could have been expected, the higher the missingness rate, the worse the output of the imputation. Finally, the different coefficient sets employed, i.e. the different stationarity rate, do not seem to affect the goodness of the model. In average, the 90,9% of the simulations gave AR(1) models with white noise residuals, which is a good result since only the 9,1% of the fitted models should be revised before being used for further analysis.

To conclude and to summarize, the algorithm can be used with AR(1)-hypothesized models, giving quite good results for both MAR and MCAR data, at any missingness rate. As already said, a tip for the user is to check the intercept of the fitted model and, if too high with respect to the other values of the time series, to subtract it from the imputations to improve the assessment metrics.

The Table 3 below shows the results obtained for the AR(2) model settings.

	AR(2) Model Settings							
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**	**Convergence %**	**WN Residuals %**
0,9 -0,5, 5% MAR	0,233197651	0,003403661	0,0583409	0,092397418	0,047889489	0,218836671	100,0%	91,4%
0,9 -0,5, 10% MAR	0,270175463	0,004404772	0,066368455	0,162740407	0,31827449	0,564158214	100,0%	91,5%
0,9 -0,5, 5% MCAR	6328,974846	19580724439	139931,1418	337,9521612	63706649,15	7981,644514	97,5%	93,2%
0,9 -0,5, 10% MCAR	132426,9429	6,96167E+12	2638498,347	28030,52773	4,21948E+11	649574,9289	93,5%	89,8%
0,7 -0,6, 5% MAR	0,23720976	0,00305422	0,055264999	0,288326837	10,49280792	3,239260397	100,0%	92,1%
0,7 -0,6, 10% MAR	0,282599489	0,004757886	0,068977428	0,185922355	2,684180368	1,63834684	100,0%	91,2%
0,7 -0,6, 5% MCAR	99771,1232	9,21942E+12	3036349,315	142149,6888	1,6667E+13	4082520,682	98,2%	92,9%
0,7 -0,6, 10% MCAR	1880,680666	2457275637	49570,91524	579,8539929	292244370,4	17095,15634	98,2%	93,2%

Table 3: AR(2) Model Settings Results

As can be immediately noticed, MCAR settings show really high means and standard deviations for both RMSE and MAPE. Since there where some simulations which did not reach convergence before the first 40 iterations, the non-convergent MCAR simulations were eliminated in order to analyse more sensitive results. These simulations can bring indeed to biased parameters imputations and, consequently, to bad missing data imputations. Moreover, if their elimination brings to an improvement of the results, this means that non-convergent simulations output should not be used by the user for further analysis. In the following Table 4, the results without the non-convergent simulations are shown.

	AR(2) Model Settings without Non-Convergent Simulations					
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**
0,9 -0,5, 5% MAR	0,233197651	0,003403661	0,0583409	0,092397418	0,047889489	0,218836671
0,9 -0,5, 10% MAR	0,270175463	0,004404772	0,066368455	0,162740407	0,31827449	0,564158214
0,9 -0,5, 5% MCAR	11,37587327	5608,814727	74,89202045	1,962543781	344,3521068	18,55672673
0,9 -0,5, 10% MCAR	12,11739173	7142,496356	84,513291	2,649317233	388,8513445	19,719314
0,7 -0,6, 5% MAR	0,23720976	0,00305422	0,055264999	0,288326837	10,49280792	3,239260397
0,7 -0,6, 10% MAR	0,282599489	0,004757886	0,068977428	0,185922355	2,684180368	1,63834684
0,7 -0,6, 5% MCAR	0,989536997	64,88481133	8,055110882	0,798772318	63,14627102	7,946462799
0,7 -0,6, 10% MCAR	0,983585155	151,934214	12,32615974	0,750373442	50,05518758	7,074969087

Table 4: AR(2) Model Setting Without Non-Convergent Simulations

It is quite clear that removing the non-convergent simulations the results improve a lot, even if they still show too high RMSE and MAPE values to be considered good. The first conclusion to be taken is therefore that the user is not recommended to use the imputed values of an AR(2) model if the algorithm does not converge.

Nevertheless, it has been noticed analysing the first class of tables, an example of which can be found in the Table 5 below, that also the simulations which take more than 10 iterations to reach the convergence usually give biased parameters and so bad imputations, as when the algorithm does not converge.

AR(2), 0,9 and -0,5, 5% MCAR								
Simulations	**Original Phi1**	**New Phi1**	**Original Phi2**	**New Phi2**	**Iterations Number**	**RMSE**	**MAPE**	**Ljung-Box test**
490	0,828291433	-0,005779337	-0,44479254	-0,005922006	12	392,3355656	160,2895274	1
263	0,905203606	-0,005035616	-0,451556266	-0,026655868	12	8,067265574	1,493087765	0,999612668
331	0,904367369	0,674249172	-0,492710016	-0,309090385	12	0,60717063	0,154645573	0,889891922
289	0,872815057	0,758723536	-0,458443693	-0,428394454	12	0,61619745	0,124451857	0,623220286
970	0,959446731	0,7464446	-0,54434325	-0,380212125	12	0,594103849	0,094195113	0,977265278
964	0,866050484	0,750546193	-0,478703359	-0,396239045	11	0,915030901	3,117743369	0,629371208
768	0,859447084	0,745546476	-0,408082577	-0,318506016	11	0,44896733	0,191728736	0,675455423
31	0,910201198	0,721248497	-0,383859724	-0,29140939	11	0,59573774	0,132246211	0,859488524
434	0,90079455	0,842803696	-0,524924049	-0,484784271	11	0,303660821	0,08595536	0,931809139
515	0,841174821	-0,012147521	-0,45510771	-0,012412076	10	1330,662569	99,2121041	0,012335965
659	0,879751629	-0,006151859	-0,545570188	-0,00622022	10	235,0564705	44,93210532	1

Table 5: AR(2) Process Simulations Example

Therefore, also in this case, user is not recommended to use the imputed values if the algorithm takes more than 10 iterations to converge. Moreover, it can also be concluded that the algorithm works better with the first set of coefficients, which represent a more stationary time series, if data are MAR. Instead, with less stationary series, as the ones defined by the second set of coefficients, better results are given if MCAR data occur. As for AR(1), a high percentage (91,9% in average) of simulations presents White Noise residuals from the fitted models.

To conclude and to summarize, the algorithm can be used with AR(2)-hypothesized models in every MAR data scenario with elevate reliability, both with 5% and 10% missingness rates. The use of the imputed values is instead not recommended if, in MCAR data settings, the algorithm reaches convergence with more than 10 iterations or if it does not converge at all. Finally, dependently from the type of missingness, the more stationary set of coefficients gives back better or worse results. Therefore, it cannot be concluded that the stationarity of the series influences in any way the goodness of the results.

The Table 6 below shows the results obtained for the MA(1) model settings.

	MA(1) Model Settings							
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**	**Convergence %**	**WN Residuals %**
0,6, 5% MAR	0,290689633	0,004148873	0,064411743	0,046790103	3,79E-05	6,16E-03	100,0%	90,6%
0,6, 10% MAR	0,322952796	0,005873894	0,07664133	0,058259025	0,000269306	0,016410547	100,0%	88,8%
0,6, 5% MCAR	0,257322571	0,004251737	0,065205346	0,052571949	0,00029698	0,017233108	100,0%	93,5%
0,6, 10% MCAR	0,368243341	0,003931062	0,062698182	0,116924363	0,070370897	0,265275135	100,0%	91,0%
0,8, 5% MAR	0,316992038	0,004651683	0,06820325	0,047292826	3,19E-05	0,005643646	100,0%	89,1%
0,8, 10% MAR	0,352729033	0,006560991	0,080999944	0,058579954	0,000249612	0,0157991	100,0%	88,0%
0,8, 5% MCAR	0,284792533	0,00462354	0,067996618	0,059718959	0,017406749	0,13193464	100,0%	93,3%
0,8, 10% MCAR	0,404370201	0,005023273	0,070875051	0,126964629	0,194430441	0,440942673	100,0%	91,5%

Table 6: MA(1) Model Settings Results

It is quite evident that all the considered MA(1) model settings give good results, with the means of the RMSE lower than 0,45 and very low standard deviations too; also MAPE means and standard deviations are very low (<0,2 and <0,45), in all the models. The simulations results do not show evident differences between the two sets of coefficients, although with a MA process the stationarity is not in discussion. Moreover, the algorithm seems to work slightly better with MAR data than to MCAR ones, but the difference is really small. All the simulations reached convergence and the 90,7% in average of the fitted models showed White Noise residuals.

It can be concluded that the algorithm can be used with MA(1)-hypothesized models, with quite reliable results for both MAR and MCAR data scenarios with any missingness rate and with any coefficients setting.

The Table 7 below shows the results obtained for the MA(2) model settings.

	MA(2) Model Settings							
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**	**Convergence %**	**WN Residuals %**
0,5 0,3, 5% MAR	0,285942889	0,004178287	0,064639672	0,046679578	3,71E-05	6,09E-03	100,0%	93,7%
0,5 0,3, 10% MAR	0,316404641	0,005763077	0,075914932	0,057552606	0,000251865	0,015870247	100,0%	93,3%
0,5 0,3, 5% MCAR	0,254041336	0,003766707	0,061373506	0,056753296	0,009995919	0,099979592	100,0%	94,5%
0,5 0,3, 10% MCAR	0,361401117	0,003954752	0,062886815	0,115104678	0,067324968	0,259470553	100,0%	94,4%
1,2 -0,3, 5% MAR	0,394699445	0,007116354	0,084358483	0,047386771	2,85E-05	0,005336806	99,9%	91,8%
1,2 -0,3, 10% MAR	0,446532365	0,009598239	0,097970603	0,061607324	0,000260577	0,016142412	100,0%	92,3%
1,2 -0,3, 5% MCAR	0,347588684	0,007216837	0,08495197	0,057342498	0,014983882	0,122408667	100,0%	93,8%
1,2 -0,3, 10% MCAR	0,501047697	0,00698684	0,08358732	0,109718797	0,016708219	0,129260276	99,9%	94,4%

Table 7: MA(2) Model Settings Results

As for the MA(1) models, also the MA(2) models do not show evident differences between the two sets of coefficients. Only one simulation in the blocks with 5% MAR data and 10% MCAR data for the second set of coefficients did not converge and their elimination did not improve the results, as shown in the following Table 8. This means that although the non-convergence, the algorithm can be used for MA(2) models, being sure of quite good imputations.

	MA(2) Model Settings Results without Non-Convergent Simulations					
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**
0,5 0,3, 5% MAR	0,285942889	0,004178287	0,064639672	0,046679578	3,70754E-05	0,006088956
0,5 0,3, 10% MAR	0,316404641	0,005763077	0,075914932	0,057552606	0,000251865	0,015870247
0,5 0,3, 5% MCAR	0,254041336	0,003766707	0,061373506	0,056753296	0,009995919	0,099979592
0,5 0,3, 10% MCAR	0,361401117	0,003954752	0,062886815	0,115104678	0,067324968	0,259470553
1,2 -0,3, 5% MAR	0,394503526	0,00707797	0,08413067	0,047389962	2,84713E-05	0,005335852
1,2 -0,3, 10% MAR	0,446532365	0,009598239	0,097970603	0,061607324	0,000260577	0,016142412
1,2 -0,3, 5% MCAR	0,347588684	0,007216837	0,08495197	0,057342498	0,014983882	0,122408667
1,2 -0,3, 10% MCAR	0,5010531	0,068001622	0,260771207	0,109722096	0,016708208	0,129260233

Table 8: MA(2) Model Settings Results Without Non-Convergent Simulations

The RMSE shows low enough mean values ($\leq$0,5) with even lower standard deviations for all the MA(2) models considered. Also MAPE values are low, in either means or standard deviations. Finally, it can be noticed that simulations with MAR data give better results than MCAR data ones, considering the same missingness rate. However, this difference is slight. The percentage of fitted models showing White Noise residuals is higher than for the previous models, presenting an average of 93,5%.

It can be concluded that the algorithm can be used also in MA(2)-hypothesized situations, with quite high reliability with any set of coefficients, any missingness rate and mechanism.

The Table 9 below shows the results obtained for the ARMA(1,1) model settings.

	ARMA(1,1) Model Settings							
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**	**Convergence %**	**WN Residuals %**
0,6 0,6, 5% MAR	0,269027365	0,005086809	0,071321871	0,096044126	0,242274718	4,92E-01	100,0%	89,6%
0,6 0,6, 10% MAR	0,301657154	0,00559248	0,074782883	0,154987909	0,847908577	0,920819514	99,9%	88,9%
0,6 0,6, 5% MCAR	0,242189335	0,004827481	0,069480074	0,139935483	0,42285402	0,650272266	99,9%	96,0%
0,6 0,6, 10% MCAR	0,364033824	0,006990956	0,083611937	0,279899479	0,991772626	0,995877817	99,9%	95,1%
0,8 0,8, 5% MAR	0,295784267	0,007765213	0,088120446	0,093025306	0,14702313	0,383435952	99,6%	89,0%
0,8 0,8, 10% MAR	0,360047864	0,011905699	0,109113241	0,143755462	0,17098964	0,413508936	99,5%	87,9%
0,8 0,8, 5% MCAR	0,2891798	0,007641183	0,087413859	0,189187524	2,294139312	1,514641645	99,9%	95,8%
0,8 0,8, 10% MCAR	0,447022771	0,012790157	0,113093575	0,992709465	270,3069006	16,44101276	99,8%	94,9%

Table 9: ARMA(1,1) Model Settings Results

For all the ARMA(1,1) models taken into consideration, they all show low RMSE means ad standard deviations (<0,45 and <0,12), as well as MAPE ones (<0,993 and <1,52). The only exception showing a high MAPE standard deviation (16,44) is the one with 10% MCAR data, with the second set of coefficients. In the following Table 10, the non-convergent simulations of all settings but the first one have been eliminated.

	ARMA(1,1) Model Settings without Non-Convergent Simulations					
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**
0,6 0,6, 5% MAR	0,269027365	0,005086809	0,071321871	0,096044126	0,242274718	4,92E-01
0,6 0,6, 10% MAR	0,301717484	0,00558884	0,074758544	0,155087797	0,847898599	0,920814096
0,6 0,6, 5% MCAR	0,241704472	0,004813218	0,069377362	0,139655331	0,421584266	0,649295207
0,6 0,6, 10% MCAR	0,361180224	0,006856725	0,082805343	0,278113656	0,984796316	0,992369042
0,8 0,8, 5% MAR	0,296052205	0,007747918	0,088022259	0,093283974	0,147448542	0,383990289
0,8 0,8, 10% MAR	0,360480204	0,011892747	0,109053873	0,144152485	0,171638411	0,414292663
0,8 0,8, 5% MCAR	0,288552214	0,007623853	0,087314677	0,188791986	2,289546302	1,513124682
0,8 0,8, 10% MCAR	0,442405087	0,012606976	0,112280793	0,984371697	267,8680531	16,36667508

Table 10: ARMA(1,1) Model Settings Results Without Non-Convergent Simulations

Eliminating the non-convergent simulations, the situation does not change (the low values remain low, that high value remains high). Therefore, it can be concluded that the non-convergence does not influence the overall results. In that specific case, analysing the block of simulations showing that high MAPE variance, some simulations reached convergence after 7-8 iterations and, although the low RMSE, showed a really high MAPE (around 440), as shown in the Table 11 below.

ARMA(1,1), 0,8 and 0,8, 10% MCAR								
Simulations	**Original Phi**	**New Phi**	**Original Theta**	**New Theta**	**Iterations Number**	**RMSE**	**MAPE**	**Ljung-Box test**
522	0,789970209	0,821351803	0,90509422	0,345463369	7	0,700022381	440,8614825	0,724464636
866	0,792682616	0,815935426	0,844354707	0,2304615	8	0,637560242	269,7462949	0,101711749
327	0,794615094	0,817870234	0,746134147	0,307440216	5	0,406102979	29,46440809	0,706265907
142	0,812925335	0,835458083	0,799924637	0,296861517	13	0,399074454	7,901654292	0,162935481
217	0,714588182	0,709385606	0,822364455	0,484422796	9	0,35700232	7,358382809	0,936557932
276	0,755835748	0,826329587	0,765567757	0,234210143	7	0,408064762	6,820124621	0,664411195
297	0,766815628	0,776147378	0,677901007	0,494752067	13	0,21324585	3,797023024	0,849522519
350	0,780176459	0,840078772	0,808402389	0,308020675	7	0,408328494	3,162206323	0,454491889

Table 11: ARMA(1,1) Process Simulations Example

As said for the AR(1) models, in this situation it is better to check the value of the intercept and subtract it from the imputed values if it is too high. Some of the simulations, moreover, although they reached convergence, did not succeed in the imputation of all the missing data. It happens when too many (>4) consecutive values are missing or when the values that are missing are the ones positioned around the threshold between the imputation through backcast and through forecast. In these situations, users should modify the algorithm to include the values at the threshold in the same imputation approach (i.e. both backcast or both forecast). Moreover, considering MCAR data, the algorithm works better for the first set of parameters (the more stationary series); considering the MAR data, the algorithm works slightly better for the second set of parameters (the less stationary series). Therefore, it cannot be concluded that the stationarity of the model influences the goodness of the results. An average of 92,2% of simulations showed White Noise Residuals from the fitted models.

To conclude and summarize, the algorithm can be used for ARMA(1,1)-hypothesized models unless too many consecutive values are missing. Modifying a bit the algorithm, it can be used if missing data lie around the threshold between the two imputation approaches. Furthermore, users should check if the estimated intercept of the fitted model shows a much higher value than the observed data of the time series and, in that case, subtract it from the imputations to achieve better results.

The Table 12 below shows the results obtained for the ARMA(2,1) model settings.

	ARMA(2,1) Model Settings							
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**	**Convergence %**	**WN Residuals %**
0,9 -0,5 0,6, 5% MAR	0,313163548	0,003948122	0,062834086	0,11075756	2,336126852	1,528439352	88,7%	90,5%
0,9 -0,5 0,6, 10% MAR	0,334283208	0,005365834	0,073251853	0,095461485	0,213184817	0,461719414	87,7%	91,2%
0,9 -0,5 0,6, 5% MCAR	0,862086858	283,8233703	16,84705821	0,96036141	634,0414467	25,18017964	93,8%	94,7%
0,9 -0,5 0,6, 10% MCAR	1736,601075	2940919923	54230,24915	58,06211871	3248604,252	1802,388485	88,3%	97,1%
0,7 -0,6 0,8, 5% MAR	0,370515013	0,004444846	0,066669677	0,081212687	0,085003726	0,291553985	96,3%	88,9%
0,7 -0,6 0,8, 10% MAR	0,396970808	0,004841891	0,069583698	0,08552712	0,069044323	0,262762865	95,9%	89,8%
0,7 -0,6 0,8, 5% MCAR	0,358641082	0,025573717	0,159917844	0,271263917	10,85168644	3,2941898	95,5%	95,5%
0,7 -0,6 0,8, 10% MCAR	33,18822376	1037140,09	1018,400751	2,262955845	3927,487625	62,66967069	92,7%	96,4%

Table 12: ARMA(2,1) Model Settings Results

It is quite evident that there are some too high values for RMSE and MAPE, in all the MCAR data situations. As done before, in order to analyse more sensitive results, the non-convergent simulations have been removed and the new results are shown in the following Table 13.

	ARMA(2,1) Model Settings without Non-Convergent Simulations					
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**
0,9 -0,5 0,6, 5% MAR	0,312231205	0,003418537	0,058468255	0,119160874	2,63015042	1,62177385
0,9 -0,5 0,6, 10% MAR	0,334433743	0,004799433	0,069277939	0,096662719	0,237365994	0,487202211
0,9 -0,5 0,6, 5% MCAR	0,317416799	0,013365298	0,115608383	0,143693731	1,210758777	1,100344845
0,9 -0,5 0,6, 10% MCAR	0,522499098	2,349519303	1,532814178	0,340721292	5,321146403	2,30676102
0,7 -0,6 0,8, 5% MAR	0,371806798	0,004357691	0,066012809	0,081678338	0,087399229	0,295633606
0,7 -0,6 0,8, 10% MAR	0,399586891	0,004613596	0,067923457	0,086419884	0,071627845	0,267633788
0,7 -0,6 0,8, 5% MCAR	0,352281335	0,007499853	0,086601694	0,278325371	11,29297049	3,360501524
0,7 -0,6 0,8, 10% MCAR	0,477325489	0,008906653	0,094375068	0,237106687	0,797892108	0,893248066

Table 13: ARMA(2,1) Model Settings Results without Non-Convergent Simulations

Removing the non-convergent simulations, the results in terms of means and standard deviations for both RMSE and MAPE are far better for the MCAR data scenarios, while it is the same for MAR data. Therefore, when considering MCAR data, the imputed values for ARMA(2,1) models should not be used for further analysis if the algorithm does not converge. The parameters, as well as the imputed values would be biased. Fitted models in MAR situations give a lower White Noise residuals percentage, if compared to MCAR situations: indeed, for MAR data the average percentage White Noise residuals rate is 90,1%, while for MCAR ones this value is 95,9%. From the results of Table 11 is not possible to conclude that the algorithm works better for one class of models rather than for the other one. Indeed, the stationarity of the simulated time series cannot be considered a discerning parameter, as well as the missingness rate.

To conclude and summarize, the algorithm can be used for all ARMA(2,1)-hypothesized models, with quite good results, unless in MCAR data settings if the algorithm does not converge.

The Table 14 below shows the results obtained for the ARMA(1,2) model settings.

	ARMA(1,2) Model Settings							
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**	**Convergence %**	**WN Residuals %**
0,6 0,5 0,3, 5% MAR	0,36930248	0,015269124	0,123568298	0,092174112	0,17317772	0,416146272	99,4%	95,6%
0,6 0,5 0,3, 10% MAR	0,421237984	0,015294663	0,123671595	0,105467009	0,067944978	0,260662575	99,6%	92,2%
0,6 0,5 0,3, 5% MCAR	346,7369657	119881595,5	10949,04541	619,6118866	383362198	19579,63733	99,7%	96,8%
0,6 0,5 0,3, 10% MCAR	0,486025185	0,28972221	0,538258497	0,686394345	100,6656586	10,03322773	98,4%	94,7%
0,8 1,2 -0,3, 5% MAR	0,442723916	0,01466217	0,121087447	0,133429715	1,030035226	1,014906511	99,8%	94,7%
0,8 1,2 -0,3, 10% MAR	0,541619176	0,025252429	0,15891013	0,286783398	10,42220326	3,228343733	99,7%	93,8%
0,8 1,2 -0,3, 5% MCAR	0,408878674	0,010981652	0,10479338	0,267651398	6,320308846	2,514022443	99,8%	94,7%
0,8 1,2 -0,3, 10% MCAR	0,588807889	0,014915161	0,122127644	0,862718456	215,0422261	14,66431813	99,3%	96,0%

Table 14: ARMA(1,2) Model Settings Results

The ARMA(1,2) settings with MAR data show quite low means and standard deviations for both RMSE and MAPE, with exception for the standard deviation of the MAPE of the 6th setting. All the settings with MCAR data instead show really high RMSE and MAPE values,

therefore the non-convergent simulations are removed from the analysis to check the eventual improvement. Results without the non-convergent simulations are shown in Table 15.

	ARMA(1,2) Model Settings without Non-Convergent Simulations					
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**
0,6 0,5 0,3, 5% MAR	0,368356124	0,01488775	0,122015366	0,092372235	0,174035599	0,417175741
0,6 0,5 0,3, 10% MAR	0,421024162	0,0150964	0,122867409	0,105387147	0,068106722	0,260972647
0,6 0,5 0,3, 5% MCAR	347,431165	120001476	10954,51852	620,8546458	383745555,7	19589,42459
0,6 0,5 0,3, 10% MCAR	0,457982188	0,011646135	0,107917258	0,662115396	100,8465779	10,04223968
0,8 1,2 -0,3, 5% MAR	0,442626719	0,014661492	0,121084648	0,133602364	1,031051921	1,015407268
0,8 1,2 -0,3, 10% MAR	0,540899175	0,025045857	0,158258828	0,287111083	10,44307417	3,231574566
0,8 1,2 -0,3, 5% MCAR	0,408950652	0,01098869	0,104826952	0,268039657	6,326561881	2,515265767
0,8 1,2 -0,3, 10% MCAR	0,586879652	0,014876193	0,121968002	0,863375349	215,2557211	14,67159573

Table 15: ARMA(1,2) Model Settings Results Without Non-Convergent Simulations

The elimination of the non-convergent simulations did not modify the results obtained in terms of RMSE and MAPE (the low values remained low, the high ones remained high). Therefore, the problem is not caused by the non-convergence of the algorithm, even if it could produce biased parameters. A deeper analysis of the first class of tables for this model was done and it was noticed that eliminating the observation 831 in the 5% MCAR data scenario with the first set of parameters, as shown in Table 16 below, considered to be an outlier with respect to the other simulations, means and standard deviations of both RMSE and MAPE decreased a lot and became really good results as shown in Table 17.

ARMA(1,2), 0,6, 0,5 and 0,3, 5% MCAR										
Simulations	**Original Phi**	**New Phi**	**Original Theta1**	**New Theta1**	**Original Theta2**	**New Theta2**	**Iterations Number**	**RMSE**	**MAPE**	**Ljung-Box test**
831	0,3597664	-0,002838937	0,633323764	-0,003018005	0,372552087	-0,005937601	20	346066,3787	618852,9773	1
902	0,592196559	0,562875957	0,417108855	0,355250368	0,372290746	0,297891479	8	0,398865145	31,68687554	0,093127947
945	0,735413133	0,768724033	0,276929697	0,189337167	0,156685776	0,09672732	4	0,243247488	4,835226052	0,2476127
437	0,657509502	0,633217217	0,537545508	0,442880455	0,274022197	0,302269679	4	0,268259434	3,010224523	0,123226273
323	0,276653338	0,131300976	0,841515159	0,688738789	0,463410692	0,424274326	6	0,415936874	2,345043143	0,141227389

Table 16: ARMA(1,2) Process Simulation Example

	ARMA(1,2) Model Setting without Observation 831					
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**
0,6 0,5 0,3, 5% MCAR	0,323788017	0,006737184	0,082080351	0,139663232	1,060229468	1,029674448

Table 17: ARMA(1,2) Model Setting Results Without Observation 831

Therefore, in these situations, it is always good to check the intercept of the fitted model and, as said before, if it is one or more orders bigger than the time series values, subtract it from the imputed values. The algorithm works better for the first set of coefficients, i.e. the more stationary one, comparing the means and the standard deviations of the MAPE of both sets of coefficients. Moreover, it gives slightly better results for MCAR data if the missingness rate is 5%; in reverse, it works better with MAR data if the missingness rate increases to 10%. The

convergence rate is quite high for all the simulation blocks, as well as the percentage average of White Noise residuals produced by the fitted models (94,8%).

To conclude and summarize, the algorithm can be used for ARMA(1,2)-hypothesized models but users have to pay attention to the eventual outliers, probably caused by a too high intercept value which has to be subtracted from the imputations. Moreover, the more stationary is the time series under analysis, the better the imputations.

The Table 18 below shows the results obtained for the ARMA(2,2) model settings.

	ARMA(2,2) Model Settings							
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**	**Convergence %**	**WN Residuals %**
0,9 -0,5 0,5 0,3, 5% MAR	0,380167684	0,023030343	0,151757513	0,103552963	0,231869798	0,481528606	95,0%	94,7%
0,9 -0,5 0,5 0,3, 10% MAR	0,432305939	0,031669971	0,177960588	0,229181889	6,401796498	2,530177167	92,3%	94,4%
0,9 -0,5 0,5 0,3, 5% MCAR	0,409272979	0,430397817	0,656047115	0,153649745	0,423477569	0,650751542	93,5%	97,5%
0,9 -0,5 0,5 0,3, 10% MCAR	4,81654056	15794,42404	125,675869	0,498623451	30,4114755	5,514660053	87,9%	95,3%
0,7 -0,6 1,2 -0,3, 5% MAR	0,457985733	0,011878475	0,108988416	0,101119838	0,221412481	0,470544877	98,5%	92,5%
0,7 -0,6 1,2 -0,3, 10% MAR	0,534810113	0,015900534	0,126097319	0,49811975	73,11880439	8,55095342	97,0%	92,3%
0,7 -0,6 1,2 -0,3, 5% MCAR	0,434643804	0,028615345	0,169160707	0,114200128	0,086176413	0,293558194	99,0%	97,5%
0,7 -0,6 1,2 -0,3, 10% MCAR	0,65334347	0,833907113	0,913185147	0,41510945	11,61354287	3,407864855	96,4%	97,5%

Table 18: ARMA(2,2) Model Settings Results

As can be immediately noticed, the algorithm found some problems in both 10% MCAR data settings, showing high RMSE and MAPE values. In these simulations blocks moreover, the non-convergence rate is quite high (87,9%). As for the other models, an attempt to improve the results is done through the elimination of the non-convergent simulations, as shown in the Table 19 below.

	ARMA(2,2) Model Settings without Non-Convergent Simulations					
	Mean RMSE	**Var RMSE**	**St Dev RMSE**	**Mean MAPE**	**Var MAPE**	**St Dev MAPE**
0,9 -0,5 0,5 0,3, 5% MAR	0,37676001	0,017265423	0,131397958	0,104979299	0,243405215	0,493361141
0,9 -0,5 0,5 0,3, 10% MAR	0,43029123	0,027376244	0,16545768	0,236263671	6,922263867	2,631019549
0,9 -0,5 0,5 0,3, 5% MCAR	0,362183306	0,019022278	0,137921274	0,149071498	0,439995735	0,663321743
0,9 -0,5 0,5 0,3, 10% MCAR	0,520096228	0,052734038	0,22963893	0,273228013	0,510752748	0,714669677
0,7 -0,6 1,2 -0,3, 5% MAR	0,457370625	0,011539752	0,107423238	0,101418537	0,224461477	0,473773656
0,7 -0,6 1,2 -0,3, 10% MAR	0,532874248	0,01533613	0,123839129	0,506406287	75,137327	8,66817899
0,7 -0,6 1,2 -0,3, 5% MCAR	0,428310358	0,01501761	0,122546357	0,113429731	0,086497598	0,29410474
0,7 -0,6 1,2 -0,3, 10% MCAR	0,599283672	0,034469236	0,185658923	0,599283672	0,034469236	0,185658923

Table 19: ARMA(2,2) Model Settings Results Without Non-Convergent Simulations

The elimination of the non-convergent simulations improved a lot the results obtained in terms of RMSE and MAPE in both 10% MCAR scenarios. Therefore, while the non-convergence of the algorithm should not be a problem in the other situations, even if it could produce biased parameters, in a 10% MCAR situation the imputations coming from non-convergent iterations do not have to be taken into account. The algorithm gives better RMSE values for the first set of coefficients, i.e. the more stationary one, in both MAR ad MCAR data situations. Moreover, the percentage average of White Noise residuals produced by the fitted models is quite high in both MAR and MCAR settings (93,5% and 97% respectively).

To conclude and to summarize, the algorithm can be used with ARMA(2,2)-hypothesized models, unless in 10% MCAR data situations the iterations do not reach convergence. In that case, users are not recommended to use the imputed values. Moreover, the more stationary is the time series under analysis, the better the imputations of the algorithm.

4.3.3 Results Summary

In this section, the results discussed in the Section 4.3.2 are schematized.

1. The algorithm works in any AR(1), MA(1), MA(2), ARMA(1,1) and ARMA(1,2) settings, since the missingness mechanism, as well as the missingness rate, does not influence the goodness of the results;
2. The estimated intercept of the fitted model has to be checked by the user and subtracted from the imputed values if it is one or more orders higher than the observed values of the time series, paying special attention to the AR(1), ARMA(1,1) and ARMA(1,2) settings;
3. The algorithm cannot be employed if too many consecutive values (≥ 4) are missing since it does not succeed with all the imputations;
4. The algorithm has to be slightly modified if missing values occur in the positions around the threshold between the two imputation approaches, i.e. forecast approach and backcast approach;
5. The algorithm gives better imputations in ARMA(1,2) and ARMA(2,2) settings if the time series under analysis is more stationary;
6. The algorithm gives high reliability results, considering AR(2), ARMA(2,1) and ARMA(2,2) models, in every MAR setting, both with 5% and 10% missingness rates;
7. The imputed values in AR(2), ARMA(2,1) and ARMA(2,2) models should not be used if in the MCAR data settings (both with 5% and 10% missingness rates for the first two models and with the 10% for the third one) the algorithm does not converge;
8. The imputed values considering an AR(2) model should not be used if in the MCAR settings the algorithm converges in more than 10 iterations;
9. The algorithm gives better results with a lower missingness rate in almost all the simulations.

It is interesting to notice that, until the order of the AutoRegressive part is lower than 2, the algorithm does not find convergence problems. However, it usually produces higher estimates than expected for the intercept. To conclude, the algorithm handles the MAR settings better than the MCAR ones and the influence of the stationarity degree of the time series considered is not of fundamental importance.

4.4 VADEMECUM FOR THE USER

In this section, the Vademecum the users should follow in order to successfully apply the created algorithm to their univariate time series presenting missing data is presented. The word vademecum takes its origins from the Latin word "vade mecum" which means "come with me". It is used to indicate a guide, a summary of what should be done or known about a specific topic. In this investigation, this word refers to all the steps the users have to follow to use the algorithm created, tested and discussed in the previous sections.

As already introduced in Section 4.1.2, the model identification step of the procedure will be assumed as done, therefore the implementation of the algorithm will start directly from the first steps. However, if the user still has doubts about the ARMA process his/her time series follows and he/she prefers to test more than one model, the Akaike Information Criteria have to be compared and the model showing the smallest one has to be chosen.

The algorithm presented in the following pages is only referred to an AR(1) process, in order not to bore the reader. In the Appendix, in Section 1, the algorithm adjusted for the other processes to be replaced in the step 6 of the Vademecum is shown.

The steps of the vademecum are defined as follows:

1. *Install the needed R-Packages*. The packages *forecast* and *stlplus* are needed for some of the steps of the algorithm.

```
library(forecast)
library(stlplus)
```

2. *Rename the Time Series*. After importing the time series on R, define *ts.orig* as the name of the time series under analysis, setting its starting and ending date, as well as its frequency. It is really important to set the frequency since it will be used in other steps. To rename the time series, replace *ts.sim* in the first code line with the name of the imported time series. *n* represents the length of the time series.

```
ts.orig <- ts(ts.sim, start=c(2002,1), end=c(2017,01), freq=12)
n <- length(ts.orig) #length of your time series
```

3. *Index the Missing Data*. After identifying where the missing values occur thanks to the function *which(is.na),* the vector containing the positions of the missing data has to be split in order to distinct the backcast imputation approach from the forecast imputation one. However, if missing values occur in symmetric positions around the threshold

between the two imputation approaches, the user should remove the "+1" in the *if* condition of the *for* loop if the AR part of the process is of order 1 or replace the "+1" with a "-1" or a "+2" if the AR part of the process is of order 2. Thanks to this adjustment, these missing values will be imputed through the same imputation method.

```
miss <- which(is.na(ts.orig))
lim <- 0
for(i in 1:length(miss)) {
  if(miss[i] < (frequency(ts.orig) + 1)) {
    lim <- lim + 1
  } next }
miss1 <- miss[0:lim]
miss2 <- miss[(lim+1):length(miss)]
```

4. *Isolate the Irregular Part of the Time Series.* Thanks to the STL decomposition, the irregular part of the time series can be isolated, to be used in the following steps.

```
ts.stl <- stlplus(ts.orig, s.window="per", na.action=na.pass(ts.orig))
residuals <- ts(ts.stl$data[,4], start=c(2002,1), end=c(2017,01), freq=12)
```

5. *Set the Initial Conditions of the Algorithm*. The maximum number of iterations allowed, as well as the tolerance can be changed from the settings of this investigation by modifying the numbers at the right of *itermax* and *tolerance*.

```
iter <-1
itermax <- 39 #maximum number of iteration allowed
tolerance <- 0.00001 #difference tolerance between sets of parameters
newts <- ts.orig
newts2 <- ts.orig
```

6. Run the Algorithm for the AR(1) process being hypothesized. Replace this step with the code shown in the Section 1 of the Appendix to deal with other ARMA processes.

```
repeat{
  iter <-iter + 1
  Fitted <- arima((newts), order = c(1, 0, 0))
  coeff <- (Fitted$coef[1])
  intercept <- (Fitted$coef[2])
  residuals[newts2 <- miss] <- 0
  newts[newts2 <- miss1] <- (-intercept + newts[(newts2 <- miss1) + 1] - residuals[(newts2 <- miss1) +
1])/coeff
 newts[newts2 <- miss2] <- intercept + coeff*newts[(newts2 <- miss2)-1] + residuals[newts2 <- miss2]
```

```
Fittednew <- arima((newts), order = c(1, 0, 0))
 coeffnew <- (Fittednew$coef[1])
 interceptnew <- (Fittednew$coef[2])
 residuals[newts2 <- miss] <- 0
 newts[newts2 <- miss1] <- (-interceptnew + newts[(newts2 <- miss1) + 1] - residuals[(newts2 <- miss1)
+ 1])/coeffnew
 newts[newts2 <- miss2] <- interceptnew + coeffnew*newts[(newts2 <- miss2)-1] + residuals[newts2 <-
miss2]
 if((((abs(coeffnew-coeff)) < tolerance)) | (iter > itermax)) {
  break
 }
 next
}
```

7. *Run the Ljung-Box Test*. Checking if the residuals of the fitted model follow a White Noise Process can be done through the Ljung-Box Test

```
LjungBox <- Box.test(Fittednew$residuals, lag=30, type = "Ljung-Box", fitdf=2)
if(LjungBox$p.value > 0.05) {
 print("White Noise Residuals")
} else {
 print("Non White Noise Residuals")
}
```

8. Check for Convergence, AIC and parameters Estimates. If $iter = itermax$, this means that the algorithm did not reach convergence.

```
print(iter)
print(Fittednew$aic)
print(Fittednew$coef)
```

Finally, after completing all the steps of the algorithm, the user should have a look at the list of results of Section 4.3.3 in order to be check whether his/her settings and results are suitable to be used for further analysis.

CHAPTER 5: CONCLUSIONS

In this final chapter, the principal conclusions of this investigation will be presented, keeping in mind the literature framework analysed, as well as the results obtained by the work. Therefore, an overview about the topic is the aim of the Section 5.1, which opens the chapter. After that, the Propositions will be commented and the answers to the Research Questions will be provided in Section 5.2. Finally, the last two sections aim to highlight the contributions of this work and the limitations and starting points for further investigations.

5.1 OVERVIEW

In the context of the missing data topic, one of the most difficult situations to handle happens when the missingness occurs in univariate time series. In the literature, although a lot of researchers published their works about the missing data problem, introducing good methods, procedures and algorithms that are still employed nowadays, just a part of them investigated about the more specific field of the time series data. And just a small part of them focused on the even more particular case of the univariate time series, where other time series could not be employed to solve the missingness problem. Time series are difficult to handle because the variable time has the highest importance, therefore a missingness causes much more losses of information than in other types of variables. For the same reason, it is harder to catch the true value of a missing observation through an imputation without modifying its influence on other observations in time. And, if this influence is modified, the entire time series setting is damaged.

The algorithm built, tested and assessed within this work of investigation aims to give a contribution to the univariate time series with missing data field, in particular creating a procedure which can be easily split in finite steps and applied by common analysts, who see the missingness as an obstacle to the continuation of their studies. Indeed, finding missingness in the dataset to analyse is a very widespread situation and, apart from the statisticians who investigate about this topic, the rest of the users is only interested in applying their statistics on the complete dataset. Therefore, an easy solution to be implemented should be made available. The algorithm described in this investigation can be applied to the most common processes generating the time series with, sometimes, some limitations. Indeed, the discussion of the results brought to the identification of the situations in which the algorithm can be used, having the certainty of obtaining good imputations and good parameters estimate, and of the situations

in which some precautions have to be taken. In this case, a solution or an action plan has been usually suggested, in order to contain the situations of doubts.

To conclude, thanks to this investigation it has been possible to face an arduous issue and to turn it into a much easier one. This work could represent, for many aspects, the starting point for many researches and studies, both about the exploration needed to broaden the action range of the algorithm and about its application to different kinds of situations. In Section 5.4, these aspects are analysed more in depth.

5.2 Answer to Research Questions

This section aims to explain how the Propositions emerged from the literature influenced the creation and the evaluation of the algorithm applied in this investigation and how the results of the algorithm answer to the research questions introduced in Chapter 1.

The Propositions emerged from the literature are the result of researches made in the field of the missing data in time series, but which were not directly applied to univariate time series. This is the reason why some of them were used as suggestions and guidelines for the explorative part of this dissertation like the Proposition 1, the Proposition 2 and the Proposition 4, while Proposition 3 content was taken as a further objective to be achieved.

The Proposition 1 and the Proposition 4, which state "The ARMA process generating a time series can be used to impute the missing values in a time series, using either a forecasting approach or a backcasting one" and "An iterative mechanism of parameters estimation until convergence preserves the characteristics of the ARMA process generating a time series" respectively, were used as basis of the algorithm object of this study. Indeed, the algorithm is a combination of these two Propositions contents since it imputes the missing values by using the ARMA Model which is supposed to have generated the time series and since it bases the phase of parameters estimation and missing data imputation on the iterative mechanism of Expectation-Maximization until the parameters convergence.

Instead, the Propositions 2, which states "A method has to rely on assumptions about the missingness pattern to be trustworthy", was partially followed. Indeed, no assumption was taken about the application of the algorithm on MAR data rather than MCAR data, while in order to apply the same algorithm on MNAR data the need for an assumption about the missingness mechanism was recognized.

Finally, Proposition 3, which states "The parameters defining the ARMA process which generates a time series do not have to be modified by the imputation method", was considered as an objective to be achieved by the created algorithm. However, the stability and the

robustness of the estimated parameters was considered a more important result to be achieved, even though the two objectives are usually coincident.

The Research Questions highlighted in the Chapter 1 found an answer through the application of the algorithm object of this investigation. The Research Questions were divided in two groups: the first one dealt with the Time Series Data Structure and included the first question Q1; the second one was about the Missing Data Imputation Methods and included the other two questions Q2 and Q3.

Q1: "*Is it possible to preserve the process generating a univariate time series when missing data occur?*"
The procedure and in particular the imputation algorithm assessed in this work of investigation aimed to answer affirmatively to this question. As can be checked in the tables showing all the simulations results, partially contained in Chapter 4, the original parameters and the new estimated ones are usually quite close to each other in the case of pure models, but they could be far away in more complicated models. The comparison between the original and new parameters however does not represent the real ability of the algorithm to catch the process generating the time series, which is given by the imputation of missing values as close as possible to the original ones. Indeed, as explained by the Wold Theorem in Section 2.1.2.1, many settings of parameters could be representative of the same time series. Therefore, the only true measure of the goodness of the imputation are the metrics used throughout the investigation, i.e. RMSE and MAPE, which should be as small as possible to be considered good.
The answer to the first question is that it is possible to preserve the process generating a univariate time series when missing data occur through the application of the algorithm of this work, in the situations in which its use is recommended (see Section 4.3.3). However, it has to be specified that the algorithm is applied relying on the assumption that the identification of the model has been correctly performed, which is a strong assumption to be done. Therefore, if the identification is not correct and a different process is hypothesized, the imputations cannot be trusted and the process generating the series could also not be preserved.

Q2: "*Is it possible to apply the same imputation algorithm structure to any missingness case?*"
The algorithm applied to different missingness mechanism settings gave quite similar and good results. The overall best results were obtained in MAR situations. After having had a look at

the results for all the hypothesized settings, it is possible to conclude that in the most part of the cases this algorithm can be applied indiscriminately to MAR and MCAR data situations, even when the missingness mechanism is unknown for the analyst. Therefore, the answer to this question is affirmative for the model settings considered, with some needed precaution (see Section 4.3.3). However, it should be considered that MNAR data situations have not been taken into account in this investigation. Further investigations could confirm the affirmative answer, or discern between MAR and MCAR data settings, where the algorithm can be applied, and MNAR data settings, where the algorithm needs to rely on other assumptions in order to work correctly.

Q3: "*Is it possible to apply the same imputation algorithm structure to both univariate and multivariate time series scenarios?*"

After having experienced the building, testing and evaluation steps of the algorithm of this investigation, it is possible to conclude that, due to the structure of the algorithm itself it is not possible to include other variables in the imputation step. Indeed, the whole procedure is calibrated in order to only take into account the weighted relationships among the observations over time. No influence from other variables, different from the one under analysis, can be considered, even if over time. Indeed, the algorithm imputes the missing values of a time series through the ARMA process which is assumed as generator of the time series. Therefore, in order to include the influence of another variable, a combination of ARMA processes should be developed. But this would impact negatively on the reliability of the imputations obtained, since there would be no distinction between the process which generated the time series and the eventual correlation between the values of two different time series.

To conclude, the answer to this question is that it is not possible to adapt the algorithm of this investigation, thought to be implemented in univariate time series, to the multivariate scenario.

5.3 CONTRIBUTION

The most important contribution of this work is the introduction of a new algorithm to be used in the univariate time series with missing data situations. The structure of this algorithm is different from the ones that already exist because it is a combination of the use of ARMA models and of the Expectation-Maximization approach, which has never been tested before. Even though further investigations can broaden the range of situations in which this algorithm can be applied, it represents an innovative way to approach a specific problem in the missing data in time series field. Researchers investigating about this topic could find this procedure

interesting and use it as a starting point for further studies. Moreover, the algorithm is also addressed, with the shape of a vademecum, to the analysts who find a missingness phenomenon in their time series and want to solve the problem in an easy and fast way.

Therefore, this work contributed to the **investigation** because it introduced a new way to deal with missing data in univariate time series and it contributed to the **practice** because it provided the users a way to solve the missing data problem in their univariate time series.

5.4 LIMITATIONS AND FURTHER INVESTIGATIONS

In the context of the presentation of the limitations and further investigations referred to this work, it is important to remember that this is an exploratory work and that it aimed to assess and evaluate a new algorithm. Therefore, some limitations of this study correspond to the suggestions given for further investigations. Indeed, not all the situations, settings and scenarios of possible applications were analysed, in order to focus on the most common aspects of the problem.

The main limitation of this study is the assumption about the identification of the model. Indeed, in order to proceed with the application of the algorithm, it has been assumed that the model underlying the time series was known by the user. It is quite evident that a part of the procedure is missing and that it gives back an optimistic view of the problem. Indeed, it is like to assume that users should have done the first step towards the correct imputation of the missing data. Further researches are therefore needed to cover the gap between the analyst's knowledge of the process and the correct application of the algorithm.

The other limitations of the work refer to the concept of broadening the application of the algorithm to more situations than the ones considered. Therefore, those are considered as suggestions for further researches, rather than limitations. For instance, this investigation put apart the stationarity issue of the time series, which is a very widespread setting in the field of the time series. Indeed, no trend and no seasonality effects have been considered in this work, although their inclusion would have improved the importance of the results obtained, most of all if positive. Further studies should include the trend and the seasonal effect typical of the time series in the algorithm.

Moreover, this investigation skipped the application of the algorithm on MNAR data situations because more effort from the user would have been required. MNAR data are indeed the most difficult situation to handle due to the logic mechanism linked to the missingness: therefore, an identification of the pattern is needed, when missing values are suspected to be MNAR.

Interesting further applications of the algorithm could fill this gap, allowing the algorithm to be used in all missingness situations.

Finally, the last suggestion for further investigations is also the hardest one and it deals with the heteroskedasticity of the time series. In this investigation, only homoskedastic time series have been taken into account. Therefore, two complementary researches should be done about this aspect: the first one regards the problem of the identification of the heteroskedasticity in a univariate time series showing missing data, while the second one regards the adjustment of the algorithm in order to catch the clusters and the volatility of the variance. To this aim, ARIMA ARCH, GARCH, EGARCH and IGARCH models should be employed, but this would complicate the simplicity of the algorithm.

To conclude, it is evident that this topic is full of clues and interesting starting points that should to be caught and analysed in depth. There is still a lot to find out.

REFERENCES

Akaike, H. 1974. A new look at the statistical model identification. ***IEEE transactions on automatic control***, 19 (6): 716-723.

Box, G. E. and Jenkins, G. M. 1970. ***Time series forecasting analysis and control***. San Francisco: Holden Day.

Box, G. E., Jenkins, G. M., Reinsel, G. C. and Ljung, G. M. 2015. ***Time series analysis: forecasting and control***. Hoboken: John Wiley & Sons.

Brockwell, P. J. and Davis, R. A. 2016. ***Introduction to time series and forecasting***. New York: Springer-Verlag.

Chatfield, C. 2000. ***Time-series forecasting***. New Jork: Chapman and Hall/CRC.

Cleveland, R. B., Cleveland, W. S., McRae, J. E. and Terpenning, I. 1990. STL: A Seasonal-Trend Decomposition. ***Journal of Official Statistics***, 6 (1): 3-73.

Collins, L. M., Schafer, J. L. and Kam, C. M. 2001. A comparison of inclusive and restrictive strategies in modern missing data procedures. ***Psychological methods***, 6 (4): 330.

Dempster, A. P., Laird, N. M. and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. ***Journal of the royal statistical society***, 39 (1): 1-38.

DiCiccio, T. J. and Efron, B. 1996. Bootstrap confidence intervals. ***Statistical science***, 11: 189-212.

Dickey, D. A. and Fuller, W. A. 1979. Distribution of the estimators for autoregressive time series with a unit root. ***Journal of the American Statistical Association***, 74 (366a): 427-431.

Diggle, P. and Kenward, M. G. 1994. Informative drop-out in longitudinal data analysis. ***Applied statistics***, 43 (1): 49-93.

Dong, Y. and Peng, C. Y. J. 2013. Principled missing data methods for researchers. ***SpringerPlus***, 2 (1): 222.

Efron, B. and Tibshirani, R. J. 1994. An introduction to the bootstrap**. *Journal of the American Statistical Association***, 89 (428): 436.

Ford, B. L. 1983. An overview of hot-deck procedures. ***Incomplete data in sample surveys***, 2 (Part IV): 185–207.

Glynn, R. J., Laird, N. M. and Rubin, D. B. 1993. Multiple imputation in mixture models for nonignorable nonresponse with followups. ***Journal of American Statistical Association***, 88: 984–993.

Hamilton, J. D. 1994. ***Time series analysis***. Princeton, New York: Princeton University Press.

Heckman, J. J. 1976. The Common Structure of Statistical Models of Truncation, Sample Selection and Limited Dependent Variables and a Simple Estimator for Such Models. ***NBER Chapters***, 5 (4): 475-492.

Heitjan, D.F. and Basu, S. 1996. Distinguishing `Missing at Random' and `Missing Completely at Random'. ***American Statistician,*** 50: 207-213.

Hinkley, D. V. and Cox, D. R. 1979. ***Theoretical statistics***. Chapman and Hall/CRC.

Honaker, J. and King, G. 2010. What to do about missing values in time-series cross-section data. ***American Journal of Political Science***, 54 (2): 561-581.

Horton, N. J. and Kleinman, K. P. 2007. Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models. ***The American Statistician***, 61 (1): 79-90.

Jamshidian, M., Jalal, S. J. and Jansen, C. 2014. MissMech: an R package for testing homoscedasticity, multivariate normality, and missing completely at random (MCAR). ***Journal of Statistical Software***, 56 (6):1–31.

Kihoro, J. and Athiany, K. 2013. Imputation of incomplete non-stationary seasonal time series data. ***Mathematical Theory and Modeling***, 3 (12):142–154.

Kim, J. and Curry, J. 1977. The treatment of missing data in multivariate analysis. ***Sociological***

Methods & Research, 6: 215-240.

King, G., Honaker, J., Joseph, A. and Scheve, K. 2001. Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation. ***American Political Science Review,*** 95 (1): 49–69.

Lesser, J. T. and Kalsbeek, W. D. 1992. ***Non-sampling error in surveys***. New York: John Wiley & Sons.

Little, R. J. A. 1988. A test of missing completely at random for multivariate data with missing values. ***Journal of the American Statistical Association***, 83 (404): 1198-1202.

Little, R. J. A. 1992. Regression with missing X's: A review. ***Journal of the American Statistical Association***, 87: 1227-1237.

Little, R. J. A. 1993. Pattern-mixture models for multivariate incomplete data. ***Journal of the American Statistical Association***, 88: 125–134.

Little, R. J. A., & Rubin, D. B. 1987. ***Statistical analysis with missing data***. New York: John Wiley & Sons.

Ljung, G. M. and Box, G. E. 1978. On a measure of lack of fit in time series models. ***Biometrika***, 65(2): 297-303.

Madow, W. G., Nisselson, H. and Olkin, I. 1983. ***Incomplete data in sample surveys. Vol. 1: Report and case studies,*** New York: Academic Press.

McLachlan, G. and Krishnan, T. 2007. ***The EM algorithm and extensions***. New York: John Wiley & Sons.

Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M. and Stork, J. 2015. Comparison of different methods for univariate time series imputation in R. ***Cologne University of Applied Sciences.*** Retrieved from https://arxiv.org/pdf/1510.03924.pdf.

Neyman, J. and Pearson, E. 1933. On the Problem of the Most Efficient Tests of Statistical Hypotheses. ***Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character***, 231: 289-337.

Rubin, D. B. 1976. Inference and missing data. ***Biometrika***, 63 (3): 581-592.

Rubin, D. B. 1987. ***Multiple imputation for nonresponse in surveys***. New York: John Wiley & Sons.

Rubin, D. B. and Schenker, N. 1986. Multiple imputation for interval estimation from simple random samples with ignorable nonresponse. ***Journal of the American statistical Association***, 81 (394): 366-374.

Schafer, J. L. 1997. ***Analysis of incomplete multivariate data***. London: Chapman and Hall/CRC.

Schafer, J. L. and Graham, J. W. 2002. Missing data: our view of the state of the art. ***Psychological methods***, 7 (2): 147.

Schomaker, M. and Heumann, C. 2016. Bootstrap Inference When Using Multiple Imputation. ***Statistics in Medicine***, 37 (14): 2252-2266.

Shao, J. and Sitter, R. R. 1996. Bootstrap for imputed survey data. ***Journal of the American Statistical Association,*** 91(435): 1278-1288.

Tagare, H. D. 1998. A Gentle Introduction to the EM Algorithm. Part I: Theory. Retrieved from:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.442.7750&rep=rep1&type=pdf

Venables, W. N., Smith, D. M. and the R Core Team, 2018. ***An Introduction to R***. Retrieved from https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf.

Verbeke, G. and Molenberghs, G. 2000. ***Linear mixed models for longitudinal data***. New York: Springer-Verlag.

Wold, H. O. 1948. On prediction in stationary time series. ***The Annals of Mathematical Statistics***, 19 (4): 558-567.

APPENDIX

1. ITERATIVE IMPUTATION STEP

1.1 AR(2) PROCESS

```
repeat{
  iter <-iter + 1
  Fitted <- arima((newts), order = c(2, 0, 0))
  coeff1 <- (Fitted$coef[1])
  coeff2 <- (Fitted$coef[2])
  intercept <- (Fitted$coef[3])
  residuals[newts2 <- miss] <- 0
  newts[newts2 <- miss1] <- (-intercept + newts[(newts2 <- miss1) + 2] - coeff1*newts[(newts2 <- miss1) + 1]-
residuals[(newts2 <- miss1) + 1])/coeff2
  newts[newts2 <- miss2] <- intercept + coeff1*newts[(newts2 <- miss2)-1]+coeff2*newts[(newts2 <- miss2)-2]
+ residuals[newts2 <- miss2]

  Fittednew <- arima((newts), order = c(2, 0, 0))
  coeff1new <- (Fittednew$coef[1])
  coeff2new <- (Fittednew$coef[2])
  interceptnew <- (Fittednew$coef[3])
  residuals[newts2 <- miss] <- 0
  newts[newts2 <- miss1] <- (-interceptnew + newts[(newts2 <- miss1) + 2] - coeff1new*newts[(newts2 <- miss1)
+ 1]- residuals[(newts2 <- miss1) + 1])/coeff2new
  newts[newts2 <- miss2] <- interceptnew + coeff1new*newts[(newts2 <- miss2)-1]+coeff2new*newts[(newts2 <-
miss2)-2] + residuals[newts2 <- miss2]
  if((((abs(coeff1new-coeff1)) < tolerance) & (((abs(coeff2new-coeff2))) < tolerance)) | (iter > itermax)) {
    break
  }
  next
}
```

1.2 MA(1) PROCESS

```
repeat{
  iter <-iter + 1
  Fitted <- arima((newts), order = c(0, 0, 1))
  coeff <- (Fitted$coef[1])
  intercept <- (Fitted$coef[2])
  residuals[newts2 <- miss1] <- 0
```

```
 newts[newts2 <- miss1] <- intercept + residuals[(newts2 <- miss1)] - coeff*residuals[(newts2 <- miss1) -1]
 residuals[newts2 <- miss2] <- - intercept + coeff*residuals[(newts2 <- miss2)-1]
 newts[newts2 <- miss2] <- intercept + residuals[newts2 <- miss2] - coeff*residuals[(newts2 <- miss2)-1]

 Fittednew <- arima((newts), order = c(0, 0, 1))
 coeffnew <- (Fittednew$coef[1])
 interceptnew <- (Fittednew$coef[2])
 newts[newts2 <- miss1] <- interceptnew + residuals[(newts2 <- miss1)] - coeffnew*residuals[(newts2 <- miss1)
-1]
 residuals[newts2 <- miss2] <- - interceptnew + coeff*residuals[(newts2 <- miss2)-1]
 newts[newts2 <- miss2] <- interceptnew + residuals[newts2 <- miss2] - coeffnew*residuals[(newts2 <- miss2)-
1]
 if((((abs(coeffnew-coeff)) < tolerance)) | (iter > itermax)) {
   break
 }
 next
}
```

1.3 MA(2) PROCESS

```
repeat{
 iter <-iter + 1
 Fitted <- arima((newts), order = c(0, 0, 2))
 coeff1 <- (Fitted$coef[1])
 coeff2 <- (Fitted$coef[2])
 intercept <- (Fitted$coef[3])
 residuals[newts2 <- miss1] <- 0
 newts[newts2 <- miss1] <- intercept + residuals[(newts2 <- miss1)] - coeff1*residuals[(newts2 <- miss1) -1] -
coeff2*residuals[(newts2 <- miss1) -2]
 residuals[newts2 <- miss2] <- - intercept + coeff1*residuals[(newts2 <- miss2)-1] + coeff2*residuals[(newts2 <-
miss2)-2]
 newts[newts2 <- miss2] <- intercept + residuals[(newts2 <- miss2)] - coeff1*residuals[(newts2 <- miss2) -1] -
coeff2*residuals[(newts2 <- miss2) -2]

 Fittednew <- arima((newts), order = c(0, 0, 2))
 coeff1new <- (Fittednew$coef[1])
 coeff2new <- (Fittednew$coef[2])
 interceptnew <- (Fittednew$coef[3])
 residuals[newts2 <- miss1] <- 0
 newts[newts2 <- miss1] <- interceptnew + residuals[(newts2 <- miss1)] - coeff1new*residuals[(newts2 <- miss1)
-1] - coeff2new*residuals[(newts2 <- miss1) -2]
```

```
 residuals[newts2 <- miss2] <- - interceptnew + coeff1*residuals[(newts2 <- miss2)-1] + coeff2*residuals[(newts2 <- miss2)-2]
 newts[newts2 <- miss2] <- interceptnew + residuals[(newts2 <- miss2)] - coeff1new*residuals[(newts2 <- miss2) -1] - coeff2new*residuals[(newts2 <- miss2) -2]
 if((((abs(coeff1new-coeff1)) < tolerance) & (((abs(coeff2new-coeff2))) < tolerance)) | (iter > itermax)) {
  break
 }
 next
}
```

1.4 ARMA(1,1) PROCESS

```
repeat{
 iter <-iter + 1
 Fitted <- arima((newts), order = c(1, 0, 1))
 coeffAR1 <- (Fitted$coef[1])
 coeffMA1 <- (Fitted$coef[2])
 intercept <- (Fitted$coef[3])
 residuals[newts2 <- miss] <- 0
 newts[newts2 <- miss1] <- (-intercept + newts[(newts2 <- miss1) + 1] - residuals[(newts2 <- miss1) + 1])/coeffAR1
 newts[newts2 <- miss2] <- intercept + coeffAR1*newts[(newts2 <- miss2) - 1] + residuals[(newts2 <- miss2)] - coeffMA1*residuals[(newts2 <- miss2) -1]

 Fittednew <- arima((newts), order = c(1, 0, 1))
 coeffAR1new <- (Fittednew$coef[1])
 coeffMA1new <- (Fittednew$coef[2])
 interceptnew <- (Fittednew$coef[3])
 residuals[newts2 <- miss] <- 0
 newts[newts2 <- miss1] <- (-interceptnew + newts[(newts2 <- miss1) + 1] - residuals[(newts2 <- miss1) + 1])/coeffAR1new
 newts[newts2 <- miss2] <- interceptnew + coeffAR1new*newts[(newts2 <- miss2) - 1] + residuals[(newts2 <- miss2)] - coeffMA1new*residuals[(newts2 <- miss2) -1]
 if((((abs(coeffAR1new-coeffAR1)) < tolerance) & (((abs(coeffMA1new-coeffMA1))) < tolerance)) | (iter > itermax)) {
  break
 }
 next
}
```

1.5 ARMA(2,1) PROCESS

```
repeat{
 iter <-iter + 1
 Fitted <- arima((newts), order = c(2, 0, 1))
 coeffAR1 <- (Fitted$coef[1])
 coeffAR2 <- (Fitted$coef[2])
 coeffMA1 <- (Fitted$coef[3])
 intercept <- (Fitted$coef[4])
 residuals[newts2 <- miss] <- 0
 newts[newts2 <- miss1] <- (-intercept + newts[(newts2 <- miss1) + 2] - coeffAR1*newts[(newts2 <- miss1) + 1]
- residuals[(newts2 <- miss1) + 2] + coeffMA1*residuals[(newts2 <- miss1) + 2])/coeffAR2
 newts[newts2 <- miss2] <- intercept + coeffAR1*newts[(newts2 <- miss2)-1] + coeffAR2*newts[(newts2 <-
miss2)-2] - coeffMA1*residuals[(newts2 <- miss2)-1] + residuals[newts2 <- miss2]

 Fittednew <- arima((newts), order = c(2, 0, 1))
 coeffAR1new <- (Fittednew$coef[1])
 coeffAR2new <- (Fittednew$coef[2])
 coeffMA1new <- (Fittednew$coef[3])
 interceptnew <- (Fittednew$coef[4])
 residuals[newts2 <- miss] <- 0
 newts[newts2 <- miss1] <- (-interceptnew + newts[(newts2 <- miss1) + 2] - coeffAR1new*newts[(newts2 <-
miss1) + 1] - residuals[(newts2 <- miss1) + 2] + coeffMA1new*residuals[(newts2 <- miss1) + 2])/coeffAR2new
 newts[newts2 <- miss2] <- interceptnew + coeffAR1new*newts[(newts2 <- miss2)-1] +
coeffAR2new*newts[(newts2 <- miss2)-2] - coeffMA1new*residuals[(newts2 <- miss2)-1] + residuals[newts2 <-
miss2]
 if((((abs(coeffAR1new-coeffAR1)) < tolerance) & ((abs(coeffAR2new-coeffAR2)) < tolerance) &
(((abs(coeffMA1new-coeffMA1))) < tolerance)) | (iter > itermax)) {
   break
 }
 next
}
```

1.6 ARMA(1,2) PROCESS

```
repeat{
 iter <-iter + 1
 Fitted <- arima((newts), order = c(1, 0, 2))
 coeffAR1 <- (Fitted$coef[1])
 coeffMA1 <- (Fitted$coef[2])
 coeffMA2 <- (Fitted$coef[3])
 intercept <- (Fitted$coef[4])
```

```
 residuals[newts2 <- miss] <- 0
 newts[newts2 <- miss1] <- (-intercept + newts[(newts2 <- miss1) + 1] - residuals[(newts2 <- miss1) + 1] + coeffMA1*residuals[(newts2 <- miss1)])/coeffAR1
 newts[newts2 <- miss2] <- intercept + coeffAR1*newts[(newts2 <- miss2)-1] + residuals[newts2 <- miss2] - coeffMA1*residuals[(newts2 <- miss2)-1] - coeffMA2*residuals[(newts2 <- miss2)-2]

 Fittednew <- arima((newts), order = c(1, 0, 2))
 coeffAR1new <- (Fittednew$coef[1])
 coeffMA1new <- (Fittednew$coef[2])
 coeffMA2new <- (Fittednew$coef[3])
 interceptnew <- (Fittednew$coef[4])
 residuals[newts2 <- miss] <- 0
 newts[newts2 <- miss1] <- (-interceptnew + newts[(newts2 <- miss1) + 1] - residuals[(newts2 <- miss1) + 1] + coeffMA1new*residuals[(newts2 <- miss1)])/coeffAR1new
 newts[newts2 <- miss2] <- interceptnew + coeffAR1new*newts[(newts2 <- miss2)-1] + residuals[newts2 <- miss2] - coeffMA1new*residuals[(newts2 <- miss2)-1] - coeffMA2new*residuals[(newts2 <- miss2)-2]
 if((((abs(coeffAR1new-coeffAR1)) < tolerance) & ((abs(coeffMA1new-coeffMA1)) < tolerance) & (((abs(coeffMA2new-coeffMA2))) < tolerance)) | (iter > itermax)) {
   break
 }
 next
}
```

1.7 ARMA(2,2) PROCESS

```
repeat{
 iter <-iter + 1
 Fitted <- arima((newts), order = c(2, 0, 2))
 coeffAR1 <- (Fitted$coef[1])
 coeffAR2 <- (Fitted$coef[2])
 coeffMA1 <- (Fitted$coef[3])
 coeffMA2 <- (Fitted$coef[4])
 intercept <- (Fitted$coef[5])
 residuals[newts2 <- miss] <- 0
 newts[newts2 <- miss1] <- (-intercept + newts[(newts2 <- miss1) + 2] - coeffAR1*newts[(newts2 <- miss1) + 1] - residuals[(newts2 <- miss1) + 2] + coeffMA1*residuals[(newts2 <- miss1)+1] + coeffMA2*residuals[(newts2 <- miss1)])/coeffAR2
 newts[newts2 <- miss2] <- intercept + coeffAR1*newts[(newts2 <- miss2)-1] + coeffAR2*newts[(newts2 <- miss2)-2] + residuals[newts2 <- miss2] - coeffMA1*residuals[(newts2 <- miss2)-1] - coeffMA2*residuals[(newts2 <- miss2)-2]
```

```
 Fittednew <- arima((newts), order = c(2, 0, 2))
 coeffAR1new <- (Fittednew$coef[1])
 coeffAR2new <- (Fittednew$coef[2])
 coeffMA1new <- (Fittednew$coef[3])
 coeffMA2new <- (Fittednew$coef[4])
 interceptnew <- (Fittednew$coef[5])
 residuals[newts2 <- miss] <- 0
 newts[newts2 <- miss1] <- (-interceptnew + newts[(newts2 <- miss1) + 2] - coeffAR1new*newts[(newts2 <- miss1) + 1] - residuals[(newts2 <- miss1) + 2] + coeffMA1new*residuals[(newts2 <- miss1)+1] + coeffMA2new*residuals[(newts2 <- miss1)])/coeffAR2new
 newts[newts2 <- miss2] <- interceptnew + coeffAR1new*newts[(newts2 <- miss2)-1] + coeffAR2new*newts[(newts2 <- miss2)-2] + residuals[newts2 <- miss2] - coeffMA1new*residuals[(newts2 <- miss2)-1] - coeffMA2new*residuals[(newts2 <- miss2)-2]
 if((((abs(coeffAR1new-coeffAR1)) < tolerance) & ((abs(coeffAR2new-coeffAR2)) < tolerance) & (((abs(coeffMA1new-coeffMA1))) < tolerance) & (((abs(coeffMA2new-coeffMA2))) < tolerance)) | (iter > itermax)) {
   break
 }
 next
}
```

ACRONYMS

ACA	Available Case Analysis
ACF	AutoCorrelation Function
AIC	Akaike Information Criterion
AR	Auto Regressive
ARCH	Auto Regressive Conditional Heteroskedasticity
ARIMA	Auto Regressive Integrated Moving Average
ARIMA	Auto Regressive Moving Average
CCA	Complete Case Analysis
CRAN	Comprehensive R Archive Network
CS	Cross Section
EGARCH	Exponential Auto Regressive Conditional Heteroskedasticity
EM	Expectation-Maximization
EMB	Expectation-Maximization with Bootstrap
GARCH	Generalized Auto Regressive Conditional Heteroskedasticity

IGARCH	Integrated Auto Regressive Conditional Heteroskedasticity
IP	Imputation Posterior
MA	Moving Average
MAPE	Mean Absolute Percentage Error
MAR	Missing At Random
MCAR	Missing Completely At Random
MI	Multiple Imputation
MLE	Maximum Likelihood Estimator
MNAR	Missing Not At Random
MSE	Mean Square Error
NA	Not Available
PACF	Partial AutoCorrelation Function
RMSE	Root Mean Square Error
SARIMA	Seasonal Auto Regressive Moving Average
SI	Single Imputation
STL	Seasonal and Trend decomposition using Loess
TS	Time Series
TSCS	Time Series Cross Section
WN	White Noise