Using Multiple Imputation to Simulate Time Series: A proposal to solve the *distance effect*.

SEBASTIAN CANO
Universitat Rovira i Virgili
Departament d'Economia
Av. Universitat 1, 43204 Reus
SPAIN
sebastian.cano@urv.net

JORDI ANDREU
Universitat Rovira i Virgili
Departament de Gestió d'Empreses
Av. Universitat 1, 43204 Reus
SPAIN
jordi.andreuc@urv.net

Abstract: Multiple Imputation (MI) is a Markov chain Monte Carlo technique developed to work out missing data problems, specially in cross section approaches. This paper uses Multiple Imputation from a different point of view: it intends to apply the technique to time series and develops that way a simpler framework presented in previous papers. Here, the authors' idea consists basically on an endogenous construction of the database (the use of lags as supporting variables supposes a new approach to deal with the *distance effect*). This construction strategy avoids noise in the simulations and forces the limit distribution of the chain to convergence well. Using this approximation, estimated plausible values are closer to real values, and missing data can be solved with more accuracy. This new proposal solves the main problem detected by the authors in [1] when using MI with time series: the previously commented *distance effect*. An endogenous construction when analyzing time series avoids this undesired effect, and allows Multiple Imputation to benefit from information from the whole data base. Finally, new R computer code was designed to carry out all the simulations and is presented in the Appendix to be analyzed and updated by researchers.

Key–Words: Missing data, Multiple Imputation, Time Series, MCMC, Simulation Algorithms, R Programming.

1 Introduction

Complex probability distributions, where the number of dimensions were a serious issue, were solved by physicists by simulation instead of direct calculation. One of the most important papers in that direction was published in 1953 [13], and was the beginning of a new fertile field. The main output of the article was the presentation of Metropolis Algorithm, that later will be generalized by Hastings [12]. The key innovation of the commented algorithm was the use of Markov chains to look for probability distributions, making the target distribution the limit distribution of the chain.

In the early years, Markov Chain Monte Carlo (MCMC) was only developed theoretically due to the lack of computational power to run these extraordinary complex algorithms. But since the late 1980s, the huge development of computers and technology allowed the empirical use of MCMC in many sciences. Nowadays, the empirical application of Markov Chain Monte Carlo is available in the great majority of technical software (Stata, R, SAS, SPlus).

As advanced before, MCMC implied a real and important revolution in multiple fields, not only in physics. We may find them in Mechanical Statistics, Bayesian Statistics or Reconstruction Image Theory for example. Within Bayesian Statistics, MCMC has been used to solve missing data problems (see [14]), application that is the main objective of this paper. Missing values in a data base represent a huge issue, because then data can not be analyzed directly. The absence of some values oblige researchers to decide how to deal with that situation: missings can be deleted or can be artificially substituted by a chosen value. The issue has been and still is a hot issue of investigation (see for example [5], [8],[7]). As can be seen in specialized literature, the decision taken by the researcher when facing missings is not innocuous, and introduces biases in calculations. To overcome this problem, Rubin proposed a new perspective. The author's idea was to combine MCMC algorithms such as *EM, Data Augmentation* or *Gibbs Sampling* to approximate the joint probability distribution of missing data and observed data. Applying Rubin's strategy it is possible to analyze the data base and the underlying missing structure to provide not only one alternative value to replace the missing value but (m) values. This simulation technique, known as Multiple Imputation (MI), offers (m) plausible values to fill in every empty cell. Once this variety of simulated results is available, MI faces another problem: this multiplicity of values must be managed and pooled. Therefore, special inference rules are designed to combine the simulated values and take into account the uncertainty.[1]

Literature regarding Multiple Imputation has been fo-

[1] The actual value is impossible to calculate, that's the reason why inference is necessary.

cused on cross section studies, where missing data is more likely to appear. In that direction, most applications of MI deal with surveys or incomplete cross section databases (see for example [4], [18]). In [1] the authors tried to apply MI in a different scenario. They tested MI with financial time series paying attention to how simulations change when one varies the main parameters of the technique. Author's wanted to see if simulated values really fit the financial time series.[2]. In case that values does not match the actual time series inference will lead to a wrong results and inference will be innacurate. In the cited paper, after almost 200 simulations [3], it was possible to draw some conclusions regarding MI sensitivity:

1. As expected by the theoretical framework, the difference between simulated values and real ones raises when the database suffers from a higher percentage of missing data. It is obvious to conclude the lower the available data (higher number of missings), the worse the estimation of plausible values (higher estimation error).

2. The estimation of plausible values becomes better increasing the number of imputations, but not in a significant way. In our empirical tests, the increase in the estimation accuracy does not worth the increase in computation time. Although from the theoretical framework the number of imputations seems a key parameter, empirical results seem not to support this importance. The simulation improvement using more than 40 imputations is negligible, because 80-90% error reduction is obtained using between 20 and 40 imputations.

3. Finally, estimated plausible values become worse when missings are distant values in time. This idea was called in our original paper *the distance effect*. Errors, when estimating distant values, raise exponentially due to the use of long time series to apply MI algorithms. Indeed, after a deep analysis, we can conclude now that the *distance effect* might be generated by an inappropriate design of the database when using MI with time series. A wrong data base structure leads to a faulty convergence of the algorithm, and therefore to *non plausible* simulations. So, a new perspective must be considered to improve MI performance when using the technique with time series.

In this paper, results from [1] are summarized and extended. Some graphs and tables are presented to a better understanding of the application of MI to time series. After some analysis of sensitivity to different parameters, this article proposes a solution to *the distance effect*, based on time series lags as supporting variables of main series. Doing the simulations in this fashion lead to better results and *the distance effect* not only decreases but almost disapears. As in many situations, the 'lag solution' brings a trade-off. Although the estimation of plausible values becomes better, higher multiplicity (more plausible values) is generated with this solution, and the necessity to pool results become overwhelming.

The paper is structured as follows: In section 2 we summarize methodology, paying attention to Markow chains, Markov Chain Monte Carlo, Gibbs Sampling and specially to Multiple Imputation. In section 3, conclusions of previous papers are presented, and the distance effect is deeply analized. Some graphs from simulations support the explanation. In section 4 a new point of view to deal with *the distance effect* is proposed. Section 5 developes empirical tests for this new approach, using different time series (economic and physic time series). Finally section 6 draws conclusions and section future research is proposed in section 7. An Appendix with the R code is presented.

2 Methodology Review

2.1 Markov Chains

A Markov chain, named after Andrey Markov, is a discrete time stochastic process which follows the Markov property. That means past and future status are independent from current status, formally this definition is written as,

$$Pr(X_{n+1} = X_n | x_n, ..., X_1 = x_1) = Pr(X_{n+1} | X_n = x_n)$$

The process followed by a Markov chain starts with a *state vector* called $\mathbf{u}$ that includes the probability values of different states. To go one step further $\mathbf{u}$ must be multiplied by the *transition matrix* $\mathbf{P}$, which includes every relation between all the possible states of the chain. So,

$$\mathbf{u}^{(n)} = \mathbf{u}^{(0)} \cdot \mathbf{P}$$

One of the most important properties of Markov chains consists on the calculation of a time independent transition matrix. Due to the nature of $\mathbf{P}$ is possible to look for a limit of the transition matrix, following the expression below:

$$\mathbf{W} = \lim_{n\to\infty} \mathbf{P}^n = \mathbf{P}^\infty \tag{1}$$

To carry on Multiple Imputation the calculus of $\mathbf{W}$ is a must, because the stationary matrix of the chain is the target distribution we are looking for. Also the chain cannot be absorbing, in this case $\mathbf{W}$ only gives information about the absorbing states. The limit then is as follows:

[2] The authors' original idea was to use new mathematical tools to estimate and predict future prices or financial values to improve Minimum Risk Index calculations developed in [3] and [2]

[3] 10 historical components of the *Dow Jones Industrial Average* were used in the simulation. Data for the period January 1962-December 2006 was downloaded from the Yahoo Finance Database. The authors used 541 monthly, 2347 weekly, and 11.328 daily observations to perform almost 200 simulations.

$$\lim_{n\to\infty} \mathbf{P}^n = \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2}$$

2.2 Markov chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a collection of methods to generate pseudorandom numbers via Markov Chains. MCMC works constructing a Markov chain which steady-state is the distribution of interest. Random Walks Markov are closely attached to MCMC. Indeed, this makes a division within the classification of MCMC algorithms. The well known Metropolis-Hastings and Gibbs Sampling are part of the Random Walk algorithms and their success depends on the number of iterations needed to explore the space, meanwhile the Hybrid Monte Carlo tries to avoid the random walk using hamiltonian dynamics.

The literature related to MCMC has raised in last decades due to the improvement of computational tools[4]. Following these improvements and developments, new fields for these methods have been discovered. For example, one can find MCMC applications in Statistical Mechanics, Image Reconstruction and Bayesian Statistics.

2.3 Gibbs Sampling

Gibbs Sampling, named after Josiah Willard Gibbs, is an MCMC algorithm created by Geman and Geman in 1984 [11]. Due to its simplicity it is a common option for those who implement Multiple Imputation in a software package. Furthermore, Gibbs Sampling has had a vital importance in the later development of Bayesian Inference thanks to BUGS software (Bayesian Inference Using Gibbs Sampling). Owing to this fact, some authors have suggested to rename the algorithm to Bayesian Sampling. The process of the algorythm is as follows: let $\pi(\theta)$ be the target distribution where $\theta = (\theta_1, \theta_2, ..., \theta_d)$. Also let $\pi_i(\theta_i) = \pi(\theta_i|\theta_{-i})$ be the conditional distributions for $i = 1, 2, ..., d$. Then, if the conditional distributions are available, we may approximate $\pi(\theta)$ through an iterative process. Gibbs Sampling is performed by 3 steps:

1. Choose the initial values at the moment j

$$\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_d^{(0)})$$

2. Calculate a new value of $\theta^{(j)}$ from $\theta^{(j-1)}$ by the following process,

$$\begin{aligned} \theta_1^{(j)} &\sim \pi(\theta_1|\theta_2^{(j-1)}, \cdots, \theta_d^{(j-1)}) \\ \theta_2^{(j)} &\sim \pi(\theta_2|\theta_1^{(j)}, \cdots, \theta_d^{(j-1)}) \\ &\vdots \\ \theta_d^{(j)} &\sim \pi(\theta_d|\theta_1^{(j)}, \cdots, \theta_d^{(j-1)}) \end{aligned}$$

3. Change the counter from j to $j + 1$ and go to the second step until the convergence is reached.

2.4 Multiple Imputation

The presence of missings means a big issue to process data. Every empty cell in a database is represented by software with *na* which cannot be treated until is replaced by a number or eliminated. In such scenario the literature has developed many approaches to deal with this problem. One traditional approach is *case deletion*, meaning the *na* is directly erased. Another solution is *single imputation*, that means the missing is substituted by a value selected by the researcher. This value can be for example the mean, the next or previous value, etc. Finally, a more complex solution to missing data is *Multiple Imputation* (MI). For a brief introduction to this technique see [15] and [17], and for a complete and detailed description see [14] and [16]. Multiple Imputation is a MCMC technique which tries to solve missing data problems in a different fashion. Instead of calculating missing values directly (as we do in single imputation), it carries many simulations to achieve *plausible values*. After this simulation, the researcher has many plausible values for every missing datum. This multiplicity of information needs to be summarized somehow, and special rules of inference are defined to pool the results.[5]

MI is a 3 stage process[6]:

imputation: The number m of imputations is set. The probability distribution $Pr(X_{mis}|X_{obs})$ is approximated through MCMC algorithms, where X_{mis} means missings and X_{obs} means observed data. Later on it will be used to Monte Carlo simulations.

analysis: Every simulated data set is analyzed using standard methods.

pool: At this point m results are available. They are combined with special inference rules.

Multiple Imputation performs fine when the data missing mechanism is random. To see that, the probability distribution of the dummy R (it represents the missing

[4] see [9] and [10]

[5] Inference rules calculate missing data uncertainty using degrees of freedom.

[6] MI stages can be found in Figure 1

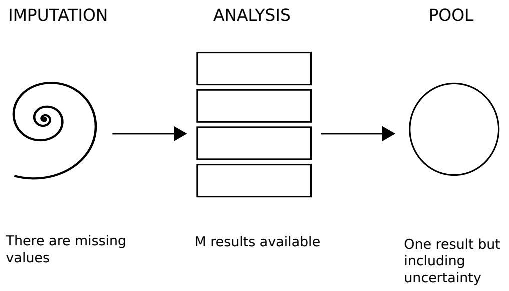

Figure 1: Three Multiple Imputation stages

data pattern) has to be analyzed. To do so, the connection between R (known information), missing information of the sample and a nuisance parameter ξ is studied through conditional probability,

$$Pr(R|X^{obs}, X^{mis}, \xi) \quad (3)$$

In case we have $R \cap X^{mis} = 0$ the missing data process is considered to be random. Nowadays, this analysis lacks a formal test to be sure about the missing data process.

3 Definition of the problem

Andreu and Cano (2008) [1] performed some tests with Multiple Imputation and time series. The authors designed a database with prices of 10 stocks of DJIA and perform several MI tests[7]. The main objective of the paper was to show MI accuracy when used with time series. To perform this sensibility analysis, the authors performed 200 simulations changing important parameters as the number of imputations, the length of the time series and the number of iterations. After this deep empirical study, convergence is shown to be reached only with a small number of iterations.[8] Secondly, MI accuracy is directly related with the number of missings the researcher is facing in the data base. Thirdly, results become better when forcing MI to perform with a higher number of imputations. Finally, one important issue was defined from the analysis: *the distance effect*, a problem that appears when simulating distant missing values. These main conclusions are presented in the next paragraphs.

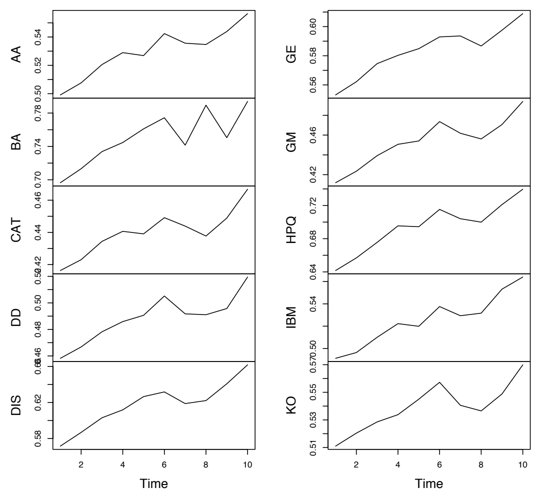

Figure 2: Errors when increasing the proportion of missing data. Weekly data.

3.1 MI estimations modifying imputation percentage

Multiple Imputation accuracy depends on the percentage of missing data we have in the data base. If missing data represent a 10% of the available dataset, plausible values provided by Multiple Imputation will be closer to real values than if the missing ratio is, for example 50%. Figure 2 shows Absolute Average Errors (AAEs) of weekly simulations increasing the percentage to simulate from 5 to 50%. It can be seen in the graph, as expected, that AAEs grow when the data base suffers from a higher missing ratio, although the increase is not linear.

3.2 MI estimations modifying number of imputations

Perhaps the number of imputations used in MI is the most important parameter to take into account. An increase in this number clearly benefits results. If more plausible values are generated, more values can be combined (pooled) and simulations are closer to real values. It can be seen in Figure 3. In this graph, the reduction of AAEs is clear. The plot shows AAEs of weekly data simulations for the entire period meanwhile we increase the number of imputations. For each time series it is possible to see how AAEs decrease when using more imputations in the simulation. According to our results, 80-90% of the error reduction is obtained using between 20 and 40 imputations. From an economic point of view and taking into account computational costs, it is worthless to use more than 40 imputa-

[7] Alcoa Inc (AA), Boeing Co (BA), Carterpillar Inc (CAT), Dupont (DD), Walt Disney (DIS), General Electric (GE), General Motors (GM), Hewlett Packard (HPQ), IBM and CocaCola (KO). Available data for 200 simulations are close prices for the period January 1962-December 2006. 541 monthly, 2.347 weekly and 11.328 daily observations are used in the calculations.

[8] Fast convergence is known to be one of the main properties of Markov Chains.

tions. The difference between a simulation using 40 and 1.000 imputations is tiny in AAEs, but huge in computational time (computational time increases 20 times when using 1.000 imputations).

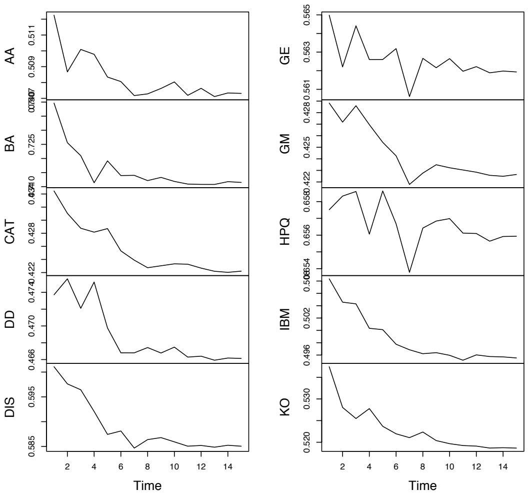

Figure 3: Errors when increasing the number of imputations. Weekly data.

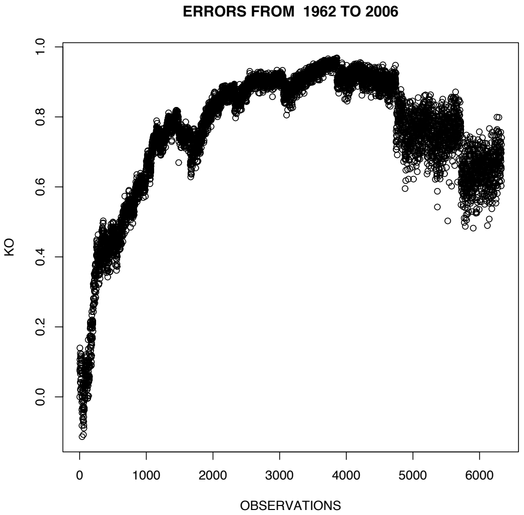

Figure 4: The Distance effect with CocaCola share's prices. Daily data.

3.3 MI estimations modifying time series length

Contrary to one of the most known principles of statistics, more data in our case might be negative. Using MI to estimate very distant missings (and using that way long

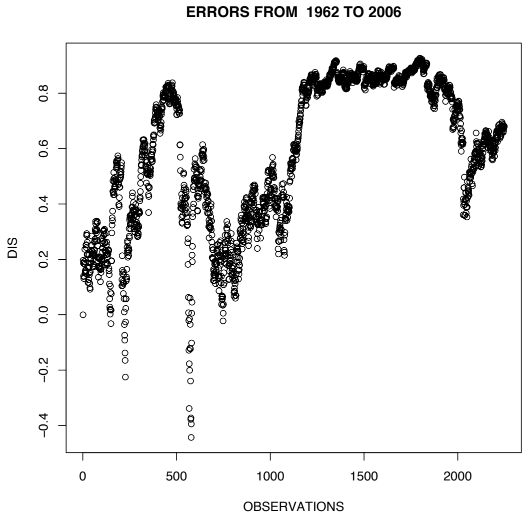

Figure 5: The Distance effect with Disney share's prices. Weekly data.

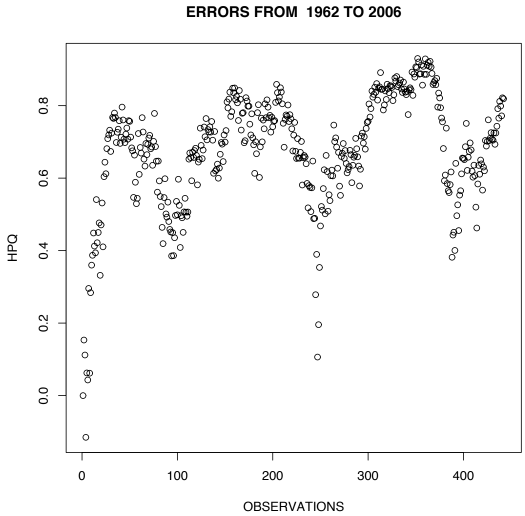

Figure 6: The Distance effect with Hewlett Packard share's prices. Monthly data.

time series) is dangerous to our purposes. Estimation errors grow when time series's length increases as can be observed in Figure 7. It is easy to see that this increase is exponentially, and disturbs MI estimations. The selected figure shows errors of weekly data simulations for the entire period and for each analysed time series, meanwhile we increase time series' length. Paying attention to the details, AAEs grow in all stocks between 5 and 90%, showing results are more sensible to this parameter than to percentage to impute. In Andreu and Cano (2008) [1] we called this problem *the distance effect*. From a theoretical point of view, these results can be explained that way. Multiple Imputation accuracy depends on the quality of available data. MCMC algorithm approximates the probability distribution function generating values of variables taking into account all the available information and correlations among variables. Augmenting the length of time series provides MI with more data to simulate missing values. The simulated value is worse if we accept structural breaks and changes in the probability distribution functions generating the time series are feasible. Giving further data as an input obliges the probability distribution to be the same during the period of analysis, and this is not necessarily true. Putting emphasis on *the distance effect*, the error analysis shows that after the 5^{th} or 6^{th} missing the simulation is not a plausible value, because the Markov chain does not converge to the right limit distribution, and error rises. In Figures 4, 5 and 6 the distance effect can be seen in detail. Figures show inicial missing estimations are good, so the error is close to the 0%. Error increases when MI tries to simulate more distant values, and this effect is similar using daily, weekly or monthly data. Usually, the distance effect becomes worse after the 6^{th} missing, and errors can increase from 0% to 100%.

4 A new point of view

The distance effect is a huge issue when using MI with time series. After conclusions in [1] and [6], some empirical tests were carried out. We conclude here that the problem could be solved applying a different approach. Multiple Imputation was designed for working on cross section databases. Making a design close to a cross section appearance seems not to work with time series, which is mainly due to the distance effect and the extraordinary increase in errors when estimating distant values. A new point of view is needed in order to use the technique with time series. In this new approach 2 issues need to be considered:

1. Proper construction of the Markov chain.

2. Noise from other variables of the database.

Let's see a normal time series (X), which is a matrix with t rows and one column,

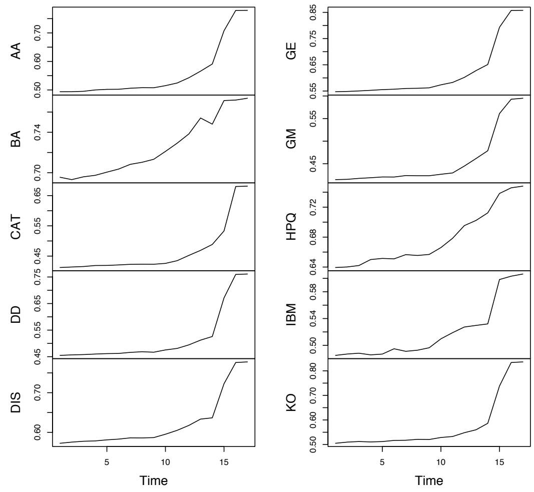

Figure 7: Errors when increasing time series lenght. Weekly data.

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_t \end{pmatrix}$$

One can add an auxiliary variable to the matrix, which is actually the first lag of X. We call the new data structure X^*, it has the shape,

$$X^* = \begin{pmatrix} x_2 & x_1 \\ x_3 & x_2 \\ x_4 & x_3 \\ \vdots & \vdots \\ x_t & x_{t-1} \end{pmatrix}$$

Arranging the time series in this fashion we let the values of the past influence the recent values. One can add as many artificial variables he may consider. Now let's think we have a missing value in our time series,

$$X = \begin{pmatrix} x_1 \\ x_2 \\ na \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ \vdots \\ x_t \end{pmatrix}$$

one can build the matrix X^* with two artificial variables,

$$X^* = \begin{pmatrix} na & x_2 & x_1 \\ x_4 & na & x_2 \\ x_5 & x_4 & na \\ \vdots & \vdots & \vdots \\ x_t & x_{t-1} & x_{t-2} \end{pmatrix}$$

Notice that now the missing value appears 3 times and is across one diagonal of the matrix. In a more complete case one might have a matrix like,

$$X^* = \begin{pmatrix} x_{t-2} & x_{t-3} & x_{t-4} & x_{t-5} \\ x_{t-1} & x_{t-2} & x_{t-3} & x_{t-4} \\ na & x_{t-1} & x_{t-2} & x_{t-3} \\ na & na & x_{t-1} & x_{t-2} \\ na & na & na & x_{t-1} \end{pmatrix}$$

Here one can identify 2 different submatrices, one is known information and the other one is missing information. Some considerations about the triangular matrix:

- if convergence is reached, values of the diagonal should be closer.
- the best simulation should be the one with more supporting information.
- when the missing is far away from the known information uncertainty will grow.

Now there are many plausible values which have to be pooled using Rubin's inference. The scalar of interest (Q) will be the value of the missing cell we are looking for. First we need calculate the average value of the scalar of interest,

$$\bar{Q} = \frac{1}{m}\sum_{t=1}^{m} \hat{Q}_{(t)} \tag{4}$$

and the total variance associated to Q,

$$T = \frac{1}{m}\sum_{t=1}^{m} \hat{S}_{(t)} + \left(1+\frac{1}{m}\right)\sum_{t=1}^{m} \frac{(\hat{Q}_{(t)} - \bar{Q})^2}{m-1} \tag{5}$$

Next step is to calculate the degrees of freedom for a small sample to carry out the inference,

$$\frac{dfc}{(1-f) - \dfrac{f-f^2}{m+1}} + \frac{dfc}{df} \tag{6}$$

After all this process inference based on t distribution can be calculated,

$$T^{0.5}(Q-\bar{Q}) \sim t_{df} \tag{7}$$

After doing this process information can be pooled. Now, there is a vector for each missing value with the following information,[9]

$$\begin{pmatrix} Lower\ value\ of\ CI \\ Central\ value\ (Q) \\ Upper\ value\ of\ CI \\ Degrees\ of\ freedom \end{pmatrix}$$

At this point one fact needs to be considered. In case one adds too many artificial variables the results might be faulty again, specially in those where frequency is low. Let's think we have annual frequency. If one adds for instance 15 artificial variables then simulated values may be smooth again. It is like saying that the value today is influenced by the value 15 years ago. We can see this fact in the figure below (Figure 8): the performance is raising until the optimal point, after that the performance decreases.

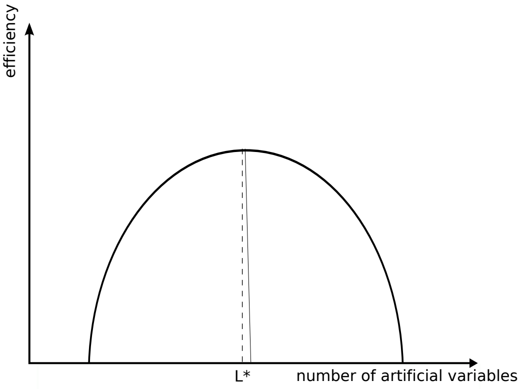

Figure 8: MI efficiency using Lags

5 Empirical tests

To illustrate what this paper has explained, we make some simulations with different time series. We use the **R** language to do the programming and to perform 5 tests. We use a library named *multiple imputation simulation for time series*.[10] There are 2 main commands in the library:

mists() performs the whole Multiple Imputation process and builds the X^* matrix. The sintax is: *mists(data, iterations, number of data simulations, number of artificial variables).*

[9] where *Upper* stands for left side of the confidence interval, *Q* stands for the average simulated value of the scalar of interest and *Upper* stands for right side of the confidence interval.

[10] This library has been programmed for the unpublished PhD Thesis "Imputación Múltiple: definición y aplicaciones", see [5].

rubin.value() pools the simulations for each missing value and makes the inference calculation. The syntax is: *rubin.value(object, missing number to make the inference).*

The tests have the following structure: first of all we run simulations with the mists() instruction and second we make the inference over each simulated value using 95% confidence level.[11]

5.1 Test 1

Time series: IBM prices. Frequency: Daily. Period: 2007. Sample: 260. Iterations: 50. Artificial variables: 9 Simulation: 7 last values.

```
mists(IBM,50,c(253:260),10)
```

Results in the following table,

Actual	Q	Lower	Upper	Df	Error
126,6	126,17	120,6	131,73	22	0,003
125,22	124,63	117,41	131,84	10	0,005
125,8	123,23	116,08	130,37	10	0,020
126,94	121,96	114,58	129,33	8	0,039
126,36	121,2	111,26	131,13	5	0,041
124,59	120,85	108,36	133,34	4	0,030
122,56	119,88	108,93	130,83	4	0,022

5.2 Test 2

Time series: Apple prices. Frequency: Daily. Period: 2007. Sample: 260. Iterations: 50. Artificial variables: 9. Simulation: 7 last values.

```
mists(Apple,50,c(253:260),10)
```

Results in the following table,

Actual	Q	Lower	Upper	Df	Error
173,56	170,24	163,72	176,75	31	0,020
176,73	170,30	164,36	176,23	27	0,036
179,30	168,99	162,22	175,75	16	0,058
179,32	168,52	161,45	175,59	19	0,060
175,74	169,27	161,67	176,87	12	0,037
175,39	167,66	151,89	183,43	4	0,044
173,53	167,55	138,48	196,62	2	0,034

5.3 Test 3

Time series: water temperature of the Pacific coast in USA. Frequency: 10 minutes. Period: July of 1974. Sample: 400. Iterations: 50. Artificial variables: 13. Simulation: 9 first values.

```
mists(aqua,50,c(1:9),14)
```

Results in the following table,

Actual	Q	Lower	Upper	Df	Error
17,7	18,14	16,52	19,75	67	-0,025
17,8	18,13	16,48	19,77	63	-0.019
17,7	18,11	16,41	19,83	59	-0.023
17,5	18,10	16,28	19,91	45	-0.034
18,6	18,01	16,10	19,92	47	0,032
18,3	18,23	16,31	20,10	43	0,004
18,2	18,34	16,36	20,31	39	-0,008
18,1	18,42	16,41	20,44	35	-0,018
18,6	18,38	16,36	20,40	34	0,012

5.4 Test 4

Time series: US output aggregate. Frequency: annual. Period: 1909 - 1949. Sample: 40. Iterations: 50. Artificial variables: 8. Simulation: 5 first values.

```
mists(usq,50,c(1:5),9)
```

Results in the following table,

Actual	Q	Lower	Upper	Df	Error
0,680	0,719	0,320	1,118	25	-0,057
0,652	0,722	0,332	1,121	20	-0,107
0,647	0,712	0,308	1,101	15	-0,100
0,616	0,710	0,290	1,134	10	-0,153
0,623	0,718	0,234	1,201	5	-0,152

5.5 Test 5

The last test in this paper is different from above. In this particular case we compare simulations obtained by the 'Lag methodology' with the ones obtained in [1]. Andreu and Cano (2008) found a pattern in the error which can be clearly noticed. To carry on this test we simulate again plausible values for the CocaCola Company time series.[12] Results for these simulations (in Figure 9) are clearly better if we compare them with the first 50 simulations obtained in Figure 4). Simulations do not show a raising pattern in the error, and statistics are quite satisfying. The following table shows the main statistics of Absolute Average Errors (AAEs) of test 5,

Mean	$0,0445$
Median	$0,0401$
St.Dev	$0,0312$
Minimum	$0,0018$
Maximum	$0,1212$

AAEs (and also median error) is around 4% with an standard deviation of 3%. Similar results can be obtained

[11] Several missing values have been simulated by the 'Lag approach'. Only 5 examples are provided here to show the application of this technique.

[12] The first two months in 1962 are simulated using daily frequency, 50 iterations and 14 artificial variables. Available sample: 260.

repeating the whole simulation for the entire period with all time series in [1]. It is possible to see in those simulations that AAEs decrease exponentially, showing the new perspective (the 'lag approach') indeed helps to improve the performance of Multiple Imputation. The proposed approach to use MI with time series seems to avoid two previously mentioned issues: proper construction of the Markov Chain and noise from other variables.

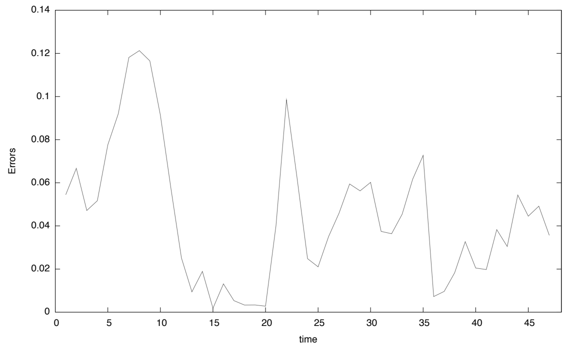

Figure 9: Errors in Test 5

6 Conclusion

Multiple Imputation is a MCMC technique developed to work out missing data problems via simulation of m simulated values. After this step, special inference rules are applied to calculate the uncertainty of missing data over the scalar of interest (it may be any value of the model or individual cells of the database). MI is basically used with cross-sectional data. In [1] the simulation accuracy depends on the time series length, concluding a *distance effect* appears disturbing MI.

In this article we made a further step using Multiple Imputation on time series. MI may be successfully applied on time series using the right data structure. Particularly, in this research an endogenous perspective has been used to design the database. This new perspective uses 'lags' as supporting variables for main time series, leading to a better construction of the Markov chain. After some empirical tests, the authors can conclude the 'lag perspective' seems to avoid noise in simulations, and allows MI to estimate better plausible values.

7 Future Research

Focusing on the performed tests, the actual value is inside the confidence interval, and Q is quite close to the real value. Even the estimation results are much better now than those obtained in previous papers, there are always things to improve. Data frequency really matters in MI simulations: it can be seen that more frequency improves simulations (best results are on water temperature, worst results are on annual frequency). In this situation, more uncertainty on the simulations appears, so the loss of freedom degrees is quite noticeable. More effort has to be put on this side to obtain a better application of MI.

Appendix

Code for the instruction mists()

```
mists ← function(x,y,z,l){
require(mice)

x → data
embed(data,l) → MRT
x[z] ← NA
embed(x,l) → MR

mice(MR,maxit=y) →MRS

complete(MRS,1) → M1
complete(MRS,2) → M2
complete(MRS,3) → M3
complete(MRS,4) → M4
complete(MRS,5) → M5

rbind(M1[z,l],M2[z,l],M3[z,l],M4[z,l],M5[z,l]) → Valor

data.frame(Valor)→V
names(Valor) ← P
   midsobj← list(
   data=(data[z]),
    imp=MRS,

   M1=as.matrix(M1[z,1:l]),
   M2=as.matrix(M2[z,1:l]),
   M3=as.matrix(M3[z,1:l]),
   M4=as.matrix(M4[z,1:l]),
   M5=as.matrix(M5[z,1:l]),


   return(midsobj)
   cat(agm(Valor))
}
```

Code for the instruction rubin.value()

```
rubin.value←function(x,y){

   mdm(x$M1)→c1
   mdm(x$M2)→c2
   mdm(x$M3)→c3
   mdm(x$M4)→c4
   mdm(x$M5)→c5

   sddm(x$M1)→cp1
   sddm(x$M2)→cp2
   sddm(x$M3)→cp3
   sddm(x$M4)→cp4
   sddm(x$M5)→cp5

   rbind(c[i]$STATSM[y])→mm
   rbind(cp[i]$STATSSD[y])→ss

   dim(x$M1)→D
   D1←D+1-y

   MIinference(mm,ss,D1)→inference
   print(inference)
   list(inference=inference)
}
```

References:

[1] J. Andreu and S. Cano. Finance forecasting: a multiple imputation approach. *Journal of Applied Mathematics - Aplimat*, 1(1), 2008.

[2] J. Andreu and S. Torra. Optimal market indices using value-at-risk: a first empirical approach for three stock markets. *Applied Financial Economics*, 19(14):1163–1170, 2009.

[3] J. Andreu and S. Torra. Market index biases and minimum risk indices. *WSEAS Transactions on Business and Economics*, 7(1):33–58, 2010.

[4] J. Barnard and X.L. Meng. Applications of multiple imputation in medical studies: from aids to nhanes. *Statistical Methods in Medical Research*, 8(1):17–36, 1999.

[5] S. Cano. Imputacion multiple: definicion y aplicaciones, tesis doctoral de la universitat rovira i virgili. Non Published, 2010.

[6] S. Cano and J. Andreu. Using multiple imputation to simulate time series. *Applied Computer and Applied Computational Science. Proceedings of the 9th WSEAS International Conference on Applied Computer and Applied Computational Sicence (ACACOS '10)*, pages 117–122, 2010.

[7] A. Dahabiah, J. Puentes, and B. Solaiman. Possibilistic pattern recognition in a digestive database for mining imperfect data. *WSEAS Transactions on Systems*, 8(2):229–240, 2009.

[8] A. Dahabiah, J. Puentes, and B. Solaiman. Possibilistic missing data estimation. *RECENT ADVANCES in ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES. Proceedings of the 9th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 173–178, 2010.

[9] J.S. Dapugnar. *Simulation and Monte Carlo*. Wiley, 2007.

[10] D. Gamerman and G. Lopes. *Markov Chain Monte Carlo*. Chapman and Hall, 2006.

[11] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions and the Bayesian Distribution of Images. *IEEE Trans. Pattern Anal. Machine Intell*, 6(6):721–741, 1984.

[12] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97, 1970.

[13] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21:1087, 1953.

[14] D.B. Rubin. *Multiple imputation for nonresponse in survey*. Wiley, 1987.

[15] D.B. Rubin. Multiple imputation after 18+ years. *Journal of the American Statistical Association*, 91, 1996.

[16] J.L. Schafer. *Analysis of Incomplete Multivariate Data*. London: Chapman and Hall / CRC Press, 1997.

[17] J.L. Schafer and J.W. Graham. Missing data: our view of the state of the art. *Psychological Methods*, 7(147-177), 2002.

[18] T. Siegl and P. Quell. Modelling specific interest rate risk with estimation of missing data. *Applied Mathematical Finance*, 12(3):283–309, 2005.