The EM Algorithm for a Multivariate Regression Model: including its applications to a non-parametric regression model and a multivariate time series model[1]

Philip Kokic[2]

Working Paper Series
No. 4, May 2002

QANTARIS GmbH

Hostatostraße 25
D-65929 Höchst
Frankfurt am Main, Germany
Tel.: ++49-69/3140 2311
Fax: ++49-69/3140 2323
Email: qantaris@freenet.de

[1] EUREDIT WP 5.7: Part B of deliverables D 5.7.1 and D 5.7.2

[2] The author wishes to thank Dennis Hauser for his help in preparing the computer programs for this part of the EUREDIT project.

Abstract

In this paper we present an EM algorithm for the imputation of missing data in a multivariate linear regression model. The method is then applied to the imputation of missing financial time series data in a varieties of ways: using straight forward linear regression of the log return price series against a common set of indexes, a nonparametric form of this, and finally one using a lagged covariates as well as the set of indexes, i.e. a multivariate AR regression model.

Key words: Imputation, AR model, multivariate linear regression, non-parametric regression

1 Introduction

The purpose of this paper is to describe the EM algorithm for estimating the parameters in a multivariate regression model. The model is then applied to the Eur$^{\text{F}}$dit time series panel data set in a variety of ways.

The multivariate model has the advantage that it can easily take covariates in to account. It is therefore expected to outperform the simpler approaches covered in the earlier working paper by Kokic (2001).

We begin with a technical description of the model and theory underlying the EM algorithm in this case. In the following section we re-derive some results that have already been obtained by Little and Rubin (1987), but in a considerably more convenient form for the current application.

2 The Multivariate Regression Model

Suppose that there are n observations in the dataset, and for the i^{th} observation, $i = 1, \ldots, n$, y_i is a $(k \times 1)$ vector response variable, and x_i a $(p \times 1)$ vector of explanatory variables. We assume that y_i is related to x_i according to the multivariate regression model:

$$y_i = B'x_i + \varepsilon_i, \tag{2.1}$$

where $\varepsilon_i \sim \text{NID}(0, \Sigma)$ (i.e. independent multivariate normal random variables), B is a $(p \times k)$ matrix of unknown regression coefficients and Σ is a $(k \times k)$ correlation matrix (unknown).

For convenience we define $X = (x_1, \ldots, x_n)'$, and $Y = (y_1, \ldots, y_n)'$. The (complete data) maximum likelihood estimates (MLEs) of B and Σ are:

$$\hat{B} = (X'X)^{-1}X'Y = (X'X)^{-1}\sum_i x_i y_i', \text{ and}$$

$$\hat{\Sigma}_M = n^{-1}\sum_i (y_i - \hat{B}'x_i)(y_i - \hat{B}'x_i)'.$$

We use the unbiased estimate of Σ instead:

$$\hat{\Sigma} = (n-p)^{-1}\sum_i (y_i - \hat{B}'x_i)(y_i - \hat{B}'x_i)'. \tag{2.2}$$

It is straight forward to show that $\hat{B} = B + \hat{R}$, where $\hat{R} = (X'X)^{-1}\sum_i x_i\varepsilon_i'$, and so

$$y_i - \hat{B}'x_i = \varepsilon_i - \hat{R}'x_i \tag{2.3}$$

For simplicity and without loss of generality, assume that

$$n^{-1}X'X = n^{-1}\sum_i x_i x_i' = I. \tag{2.4}$$

Then,

$$\begin{aligned}
\hat{R} &= n^{-1}\sum_i x_i\varepsilon_i', \\
\hat{B} &= B + n^{-1}\sum_i x_i\varepsilon_i', \text{ and} \qquad (2.5)\\
\hat{\Sigma} &= (n-p)^{-1}\sum_i (y_i - \hat{B}'x_i)(y_i - \hat{B}'x_i)' \\
&= (n-p)^{-1}\sum_i (\varepsilon_i - \hat{R}'x_i)(\varepsilon_i - \hat{R}'x_i)' \\
&= (n-p)^{-1}\sum_i \varepsilon_i\varepsilon_i' - n(n-p)^{-1}\hat{R}'\hat{R} \\
&= (n-p)^{-1}\sum_i (1 - n^{-1}x_i'x_i)\varepsilon_i\varepsilon_i' - n^{-1}(n-p)^{-1}\sum_{i\neq j} x_i'x_j\varepsilon_i\varepsilon_j'. \qquad (2.6)
\end{aligned}$$

One additional result that also follows from (2.4) and will be used in the following is:

$$n^{-1}\sum_i x_i'x_i = \text{trace}(n^{-1}XX') = \text{trace}(n^{-1}X'X) = p. \tag{2.7}$$

3 The EM Algorithm for the Regression Model

Suppose that some of the y values are missing or partly missing, but that we still wish to estimate the mean and covariance matrix[3]. To do this we use the EM algorithm. There are two steps involved in the EM algorithm, the E-step and M-step. By taking the conditional expectations of the $\hat{B}$ and $\hat{\Sigma}$ estimates, given the observed data and the current estimates, we can avoid the need to separately determine the M-step. In other words, to apply the EM algorithm all we require is computable expressions for:

$$E(\hat{B} \mid \text{observed data, } B, \Sigma \text{ known}), \text{ and } E(\hat{\Sigma} \mid \text{observed data, } B, \Sigma \text{ known}) \tag{3.1}$$

Given that B is known in (3.1), when y_i is missing ε_i is missing, and vice versa, thus we can use the results at (2.5) or (2.6) above to obtain closed form expressions for (3.1). To derive these, some simple results are first established (or presented).

Let us denote the conditioning at (3.1) by E^*. Let $\varepsilon = (\varepsilon_1', \varepsilon_2')' \sim N(0, \Sigma)$, where ε_1 is the missing data and ε_2 is the observed data. Also let

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

[3]Since, in the end, we need to perform imputation, it is necessary to assume the x_i have no missing values. If the only objective is to estimate B and Σ, then there is no need to impose this condition.

where the partition is according to the missing/nonmissing data. Assuming non-informative non-response, from standard results for the multivariate normal distribution:

$$E^*(\varepsilon_1 \mid \varepsilon_2) = \begin{cases} \Sigma_{12}\Sigma_{22}^{-1}\varepsilon_2 = \beta'\varepsilon_2, & \text{if } \varepsilon_2 \text{ is not empty, and} \\ 0, & \text{otherwise,} \end{cases}$$

$$E^*(\varepsilon_1\varepsilon_1' \mid \varepsilon_2) = \begin{cases} \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} + \beta'\varepsilon_2\varepsilon_2'\beta, & \text{if } \varepsilon_2 \text{ is not empty, and} \\ \Sigma, & \text{otherwise.} \end{cases}$$

where $\beta = \Sigma_{22}^{-1}\Sigma_{21}$. Let $\beta^* = (\beta, I_{22}) = \Sigma_{22}^{-1}\Sigma_2$, where $\Sigma_2 = (\Sigma_{21}, \Sigma_{22})$. Thus,

$$E^*(\varepsilon \mid \varepsilon_2) = \begin{cases} \begin{pmatrix} \beta' \\ I_{22} \end{pmatrix} \varepsilon_2 = \beta^{*\prime}\varepsilon_2, & \text{if } \varepsilon_2 \text{ is not empty, and} \\ 0, & \text{otherwise,} \end{cases} \tag{3.2}$$

$$\begin{aligned} E^*(\varepsilon\varepsilon' \mid \varepsilon_2) &= \begin{pmatrix} \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} & 0 \\ 0 & 0 \end{pmatrix} + E^*(\varepsilon \mid \varepsilon_2)E^*(\varepsilon \mid \varepsilon_2)' \\ &= \Sigma - E\left\{E^*(\varepsilon \mid \varepsilon_2)E^*(\varepsilon \mid \varepsilon_2)'\right\} + E^*(\varepsilon \mid \varepsilon_2)E^*(\varepsilon \mid \varepsilon_2)' \qquad (3.3) \\ &= \Sigma - \begin{pmatrix} \beta' \\ I_{22} \end{pmatrix} \Sigma_{22} \begin{pmatrix} \beta & I_{22} \end{pmatrix} + E^*(\varepsilon \mid \varepsilon_2)E^*(\varepsilon \mid \varepsilon_2)' \\ &= \Sigma + \begin{pmatrix} \beta' \\ I_{22} \end{pmatrix} (\varepsilon_2\varepsilon_2' - \Sigma_{22}) \begin{pmatrix} \beta & I_{22} \end{pmatrix}, \\ &= \begin{cases} \Sigma + \beta^{*\prime}(\varepsilon_2\varepsilon_2' - \Sigma_{22})\beta^*, & \text{if } \varepsilon_2 \text{ is not empty, and} \\ \Sigma, & \text{otherwise.} \end{cases} \qquad (3.4) \end{aligned}$$

where I_{22} is an identity matrix.

For subsequent developments (3.3) is a more convenient form for algebraic representation of the final solution, while (3.4) is more convenient for algorithmic purposes. For observation i let ε_i^* also denote the observed data (note that the subscripts of the observed and missing data may be different for each observation). Also let

$$C_i = \begin{cases} E^*(\varepsilon_i \mid \varepsilon_i^*)E^*(\varepsilon_i \mid \varepsilon_i^*)' - E\left\{E^*(\varepsilon_i \mid \varepsilon_i^*)E^*(\varepsilon_i \mid \varepsilon_i^*)'\right\}, & \text{if } \varepsilon_2 \text{ is not empty, and} \\ 0, & \text{otherwise.} \end{cases} \tag{3.5}$$

This is the second term in (3.3) for the i^{th} observation. Then from (2.5),

$$E^*(\hat{B}) = B + E^*(\hat{R}) = B + n^{-1}\sum_i x_i E^*(\varepsilon_i'|\varepsilon_i^*), \tag{3.6}$$

and from (2.6), (2.7), (3.3) and (3.5),

$$\begin{aligned}
E^*(\hat{\Sigma}) &= (n-p)^{-1}\sum_i (1-n^{-1}x_i'x_i)E^*(\varepsilon_i\varepsilon_i' \mid \varepsilon_i^*) \\
&\qquad - n^{-1}(n-p)^{-1}\sum_{i\neq j} x_i'x_j E^*(\varepsilon_i \mid \varepsilon_i^*)E^*(\varepsilon_j' \mid \varepsilon_j^*) \\
&= (n-p)^{-1}\sum_i (1-n^{-1}x_i'x_i)(\Sigma + C_i) - n(n-p)^{-1}E^*(\hat{R}')E^*(\hat{R}) \\
&\qquad + n^{-1}(n-p)^{-1}\sum_i x_i'x_i E^*(\varepsilon_i \mid \varepsilon_i^*)E^*(\varepsilon_i' \mid \varepsilon_i^*) \\
&= \Sigma + (n-p)^{-1}\sum_i C_i - n(n-p)^{-1}E^*(\hat{R}')E^*(\hat{R}) \\
&\qquad - n^{-1}(n-p)^{-1}\sum_i x_i'x_i E\left\{E^*(\varepsilon_i \mid \varepsilon_i^*)E^*(\varepsilon_i' \mid \varepsilon_i^*)\right\} \\
&= \Sigma + (n-p)^{-1}\sum_i C_i - n(n-p)^{-1}E^*(\hat{R}')E^*(\hat{R}) + O_p(n^{-2}), \text{ as } n\to\infty, \quad (3.7)
\end{aligned}$$

provided that for all i and n, $x_i'x_i$ is bounded above. Together (3.6) and (3.7) specify algebraically how the EM algorithm would operate: in the r^{th} iteration step, the B and Σ used to determine the various terms on the right hand side of these two expressions are replaced by their current estimates, $\tilde{B}_{r-1}$ and $\tilde{\Sigma}_{r-1}$ say. Note that β_i must be computed separately for each observation i in the definition of C_i and $E^*(\varepsilon_i \mid \varepsilon_i^*)$. Also, note that (3.7) is not guaranteed to produce a positive definite result, so it may be necessary to introduce a ridge parameter to ensure that this is the case. More details will be given section 4.

First, though, let us briefly examine the impact of assumption (2.4) on the estimating equations in the EM algorithm in this case. The equations that are used in the algorithm are (3.2), (3.4), (3.5), (3.6) and (3.7). Clearly, the only term in these equations affected by the removal of assumption (2.7) is the definition of $E^*(\hat{R})$ in (3.6) and (3.7), and this simply becomes:

$$E^*(\hat{R}) = (X'X)^{-1}\sum_i x_i E^*(\varepsilon_i'|\varepsilon_i^*). \quad (3.8)$$

After estimating the model parameters one needs to impute the missing values. To perform this task we simply utilise equations (2.1) and (3.2) together with the final estimates $\tilde{B}$ and $\tilde{\Sigma}$ of B and Σ, respectively:

$$\tilde{y}_i = \tilde{B}'x_i + E^*(\varepsilon_i|\varepsilon_i^*). \quad (3.9)$$

4 Specification of the Algorithm

The algorithm is briefly as follows. We do not give full specifications as these are contained in the associated MatLab source code where these procedures are implemented.

1. Set $r = 1$ for the first iteration of the EM algorithm.
2. Define the default values of B and Σ:

(a) If default values are available denote these by $B_D \equiv (b_{D1}, \ldots, b_{Dk})$ and $\Sigma_D \equiv (\sigma_{Djl})$.

(b) When no default values are available set $B_D = 0$ and $\Sigma_D = I$.

3. Construct an initial estimate of B:

   (a) Determine $\tilde{b}_j$ by univariate regression[4] of the y_j against X using the observed set of data for the j^{th} component only.

   (b) When the j^{th} univariate regressions cannot be performed (e.g. no observations in y_j) use the default value from B_D: $\tilde{b}_j = b_{Dj}$.

   (c) Combine these to form the initial estimate $\tilde{B}_1 = (\tilde{b}_1, \ldots, \tilde{b}_k)$.

4. Construct an initial estimate of Σ:

   (a) Using the residuals from the univariate regressions at (3a) for components j and l, construct an estimate $\tilde{\sigma}_{0jl}$ based on the non-missing values in common.

   (b) When there are no observations in common set $\tilde{\sigma}_{0jl} = \sigma_{Djl}$.

   (c) Let $\tilde{\Sigma}_1 = (\tilde{\sigma}_{0jl}) + \lambda_2 I$, where $\lambda_2 \geq 0$ is large enough to ensure that the estimate is positive definite[5].

5. Increment r and update the EM algorithm estimates of B and Σ:

   (a) $r \leftarrow r + 1$

   (b) For each observation i compute $E^*(\varepsilon_i|\varepsilon_i^*)$ and C_i. This involves the following steps:

      i. Using $B = \tilde{B}_{r-1}$ in (2.1), compute ε_i, and then remove the missing elements to form ε_i^*.

      ii. If ε_i^* is empty set $E^*(\varepsilon_i|\varepsilon_i^*) = 0$ and $C_i = 0$, then start processing the next observation.

      iii. Delete the rows of $\tilde{\Sigma}_{r-1}$ corresponding to the missing elements of y_i to form $\tilde{\Sigma}_{r-1,2}$. Now delete the columns of this corresponding to the missing elements of y_i to form $\tilde{\Sigma}_{r-1,22}$.

      iv. Compute $\beta_{r-1}^* = \tilde{\Sigma}_{r-1,22}^{-1}\tilde{\Sigma}_{r-1,2}$, and then expression (3.2) and the second term in (3.4).

   (c) Compute (3.8)[6] and as at (3.6) add this to $\tilde{B}_{r-1}$ to obtain $\tilde{B}_r$

   (d) Ignoring the $O_p(n^{-2})$ term and using $\tilde{\Sigma}_{r-1}$ in place of Σ, compute the right-hand side of expression (3.7) to obtain $\tilde{\Sigma}_r$.

   (e) As at step 4c, add $\lambda_3 I$ to $\tilde{\Sigma}_r$, where $\lambda_3 = \max\{\delta - e_{\min,r}, 0\}$ and $e_{\min,r}$ is the minimum eigenvalue of $\tilde{\Sigma}_r$.

6. Repeat the entire process from step 5 until the estimates of B and Σ converge.

7. Finally, using expression (3.9), impute the missing y-values.

[4] It may be necessary to use univariate ridge regression to overcome collinearity problems, see Marquardt and Snee (1975), where the ridge parameter λ_1 is set to a small positive constant.

[5] If the minimum eigenvalue of $\tilde{\Sigma}_1$ is $e_{\min,1}$ say, then set $\lambda_2 = \max\{\delta - e_{\min,1}, 0\}$, where δ is a small positive constant. Prior to this computation it may be necessary to make $\tilde{\Sigma}_1$ symmetric.

[6] As at step 3a, it may be necessary to use ridge regression to deal with multicollinearity. In this case it would make sense to use the same value of the ridge regression parameter

5 Application of the EM Algorithm to Financial Time Series Data

We consider three different ways of imputing financial time series data with the EM algorithm: using stock indexes and exchange rates as the covariates in the linear regression model in section 5.2, a nonparametric regression form of this model in section 5.3, and finally in section 5.4 using lagged variables and indexes as covariates. It is also possible to consider a non-parametric form of the last model, and we describe it briefly in passing, but it is not developed any further in this paper.

To be consistent with the last-value carried forward imputation method, which has the best overall performance out of all simple methods examined (Kokic 2001), we propose that a log-return pre-transformation of the data be performed. This has implications for the post-processing of the data, see section 5.1. For a brief overview of the data see the documentation provided with the financial time series CD for the Eur$^{\text{E}}$dit project.

For convenience, we shall use standard MatLab notation for sub-indexing of matrixes throughout this section (The MathWorks, Inc. 1999).

5.1 Pre- and post-transformation of the data

Let us denote the price (or index) time series by $\{P_{ti}, t = 1, \ldots, T\}$ where t is time (days) and $i = 1, \ldots, I$ is an instrument or index label, and let $P = (P_{ti})$ be the matrix of all these values. In all cases $P_{ti} \in \mathbb{R}^+ \cup \{\cdot\}$, i.e. the values are either positive real numbers, or missing, denoted by "$\cdot$". In the Eur$^{\text{E}}$dit project the dimension of P is 1304×99, that is there are 1304 daily values for 99 time series.

Since these time series include weekends and public holidays when no prices can be observed, and because no missing observations are allowed in the explanatory data, all rows with completely missing data are first removed from the price matrix P. These rows may be added back in once the imputation of the missing prices has been completed.

Preprocessing consists of taking the log returns of each time series:

$$Z_{ti} = \begin{cases} log(P_{t+1,i}/P_{ti}) & \text{if both values are non-missing, and} \\ \cdot & \text{otherwise.} \end{cases} \tag{5.1}$$

Note that this results in one less observation in each time series, and if there is a continuous subsequence, $t = l + 1, \ldots, l + m$, say, of m missing values in $\{P_{ti}\}$, this results in a subsequence of $m + 1$ missing values in $\{Z_{ti}\}$ when $0 < l < T - m$, and exactly m missing values when $l = 0$ or $T - m$. However, for reasons of consistency, the imputed values $\tilde{Z}$ in this subsequence should be constrained to add to the log return of the non-missing values just before and after the subsequence:

$$\sum_{t=l}^{l+m} \tilde{Z}_{ti} = log(P_{l+m+1,i}/P_{li}), \text{ if } 0 < l < T - m. \tag{5.2}$$

But for the imputation methods we are considering there is no guarantee that this constraint will be satisfied. For example, the last-value carried forward imputation

technique $\tilde{Z}_{ti} = 0$ which in general does not satisfy the above constraint. Thus we propose the following simple post-imputation adjustment:

$$\tilde{\tilde{Z}}_{ti} = \begin{cases} \tilde{Z}_{ti} + \frac{1}{m+1}\{log(P_{l+m+1,i}/P_{li}) - \sum_{t=l}^{l+m} \tilde{Z}_{ti}\}, & \text{if } 0 < l \leq t \leq l+m < T, \\ \tilde{Z}_{ti}, & \text{if } l = 0 \text{ or } l = T-m. \end{cases} \quad (5.3)$$

Note that the $\{\tilde{\tilde{Z}}_{ti}\}$ will satisfy expression (5.2). One minor disadvantage of this adjustment is the need for revision of the imputed values. However, there is only one revision required, and this is made once a new price can be observed.

Finally, these imputed values must be transformed back to the original scale. The manner this is performed depends on the arrangement of and types of missing values in the price time series. There are essentially two types to consider: type 1. normal missing values for which imputation should be performed, and type 2. naturally missing values for which imputation should not be performed (e.g. beyond the expiry date of an instrument). We now present an algorithm for performing the back-transformation.

1. For $t = 1, \ldots, T-1$, set $\tilde{\tilde{P}}_{1i} = P_{1i}$,

$$\hat{Z}_{ti} = \begin{cases} \tilde{\tilde{Z}}_{ti}, & \text{if } Z_{ti} = \cdot, \\ Z_{ti}, & \text{otherwise, and} \end{cases}$$

$$\tilde{\tilde{P}}_{t+1,i} = \begin{cases} P_{t+1,i} & \text{if } P_{t+1,i} \neq \cdot\text{(type 1)}, \\ \cdot\text{(type 1)}, & \text{if } P_{t+1,i} = \cdot\text{(type 1) and } \tilde{\tilde{P}}_{ti} = \cdot \text{ and} \\ \tilde{\tilde{P}}_{ti}\exp(\hat{Z}_{ti}), & \text{otherwise.} \end{cases}$$

2. For $t = T, \ldots, 2$, set

$$\tilde{\tilde{P}}_{t-1,i} = \begin{cases} \tilde{\tilde{P}}_{t-1,i} & \text{if } \tilde{\tilde{P}}_{t-1,i} \neq \cdot\text{(type 1)}, \\ \cdot\text{(type 1)}, & \text{if } \tilde{\tilde{P}}_{t-1,i} = \cdot\text{(type 1) and } \tilde{\tilde{P}}_{ti} = \cdot \text{ and} \\ \tilde{\tilde{P}}_{ti}\exp(-\hat{Z}_{t-1,i}), & \text{otherwise.} \end{cases}$$

5.2 Imputation using stock and exchange rate index covariates

It is possible to partition Z into several parts: $Z = Z^{(1)}, \ldots, Z^{(5)}$, where $Z^{(1)}$ are US shares, $Z^{(2)}$ are UK shares, $Z^{(3)}$ are UK bonds (Gilts), $Z^{(4)}$ are UK derivatives and $Z^{(5)}$ are stock indexes and exchange rates. In the Eur$^{\text{E}}$dit project the dimensions of these matrixes are 1303×9, 1303×6, 1303×36, 1303×36 and 1303×12, respectively. We may subdivide each $Z = Z^{(j)}$ into blocks covering one year or other time-spans of data. Denote the resulting submatrix of data by $Z = Z^{(uj)}$, where $u = 1, \ldots, U$ represent the time partition, and $j = 1, \ldots, 5$ represents the data partition. In the case $U = 1$, $Z^{(1j)} = Z^{(j)}$. For the imputation model (2.1) we suggest two separate specification of Y and X be used.

Specification R.1:

1. Perform the pre-transformation (5.1) on P to obtain $Z = [Z^{(1)}, \ldots, Z^{(5)}]$

2. Set $Y = [Z^{(1)}, Z^{(2)}, Z^{(3)}, Z^{(4)}]$, $X = Z^{(5)}$.
3. Impute using the EM algorithm in section 4 to obtain $\tilde{Z}$.
4. Perform the post-transformation (5.3) on the imputed values to obtain $\tilde{\tilde{Z}}$.
5. Transform back using the algorithm presented in section 5.1 to obtain the imputed values $\tilde{\tilde{P}}$.

Specification R.2:

1. Perform the pre-transformation (5.1) on P to obtain $Z = [Z^{(1)}, \ldots, Z^{(5)}]$
2. Separately for each $u = 1, \ldots, 5$ and $j = 1, \ldots, 4$ perform the following:
   (a) Set $Y = Z^{(uj)}$ and $X = Z^{(u5)}$
   (b) Impute using the EM algorithm in section 4.
3. Perform the post-transformations in section 5.1 to obtain the imputed values $\tilde{\tilde{P}}$.

5.3 Imputation using a nonparametric regression model

Nonparametric regression is relatively straightforward generalisation of the regression approach. With financial data it makes sense to apply the method using time as the smoothing variable, and lagged observations only as this will minimise the number of revisions required for missing observations to one (if the post-imputation adjustment (5.2) is applied). It would be possible to use an exponential weighting in the smoothing window as well, but for simplicity in the current situation we only use fixed weights. The approach proceeds as follows.

Specification NP:

1. Perform the pre-transformation (5.1) on P to obtain $Z = [Z^{(1)}, \ldots, Z^{(5)}]$
2. Set $j = 1$, the length of the estimation window to $h = 100$[7], and $w = (1, \ldots, h)$.
3. Set $Y_{\text{ini}} = Z^{(j)}$ and $X_{\text{ini}} = Z^{(5)}$.
4. Using the EM algorithm on Y_{ini} and X_{ini}, estimate B and Σ, but don't impute any missing values. Denote the parameter estimates by B_D and Σ_D, respectively.
5. Set $Y = Y_{\text{ini}}(w, :)$ and $X = X_{\text{ini}}(w, :)$.
6. Apply the EM algorithm using Y, X and the default parameters B_D and Σ_D. Replace B_D and Σ_D by the EM algorithm estimates $\tilde{B}$ and $\tilde{\Sigma}$, respectively. Save the imputed values: $Y_{\text{imp}}^{(j)} \leftarrow \tilde{Y}$.
7. Move the estimation window forward one step: $w \leftarrow w + 1$.
8. Set $Y = Y_{\text{ini}}(w, :)$ and $X = X_{\text{ini}}(w, :)$.

[7] Other values should be tested, e.g. $h = 50, 100, 150, 200, 250$, and only the value giving the best result used in the end.

9. Apply the EM algorithm using Y, X and the default parameters B_D and Σ_D. Replace B_D and Σ_D by the EM algorithm estimates $\tilde{B}$ and $\tilde{\Sigma}$, respectively. Append the imputed values from the last row of $\tilde{Y}$ to $Y_{\text{imp}}^{(j)}$: $Y_{\text{imp}}^{(j)} \leftarrow [Y_{\text{imp}}^{(j)}; \tilde{Y}(h,:)]$[8].

10. Repeat the algorithm from step 7 until the end of Y_{ini} is reached (i.e. there are no more values in Y_{ini} left to impute).

11. Repeat the algorithm from step 2 for $j = 2, 3$ and 4.

12. Set $\tilde{Z} = [Y_{\text{imp}}^{(1)}, \ldots, Y_{\text{imp}}^{(4)}, Z^{(5)}]$.

13. Perform the post-transformations in section 5.1 to obtain the imputed values $\tilde{\tilde{P}}$.

5.4 Imputation using a multivariate AR regression model

In this subsection we show how to apply the EM imputation algorithm (section 4) to two types of time series models. The first is a multivariate AR model with covariates. To avoid collinearity problems we propose that a simple lag 1 structure be used (i.e. only include lags of one point back in time). Specification of the model and the algorithm is given below and is referred to as the MARX1 specification (multivariate auto-regressive with an X-covariate).

The second model for imputation is referred to as AR5X. For this model we propose the use of a univariate response variables, the covariates are the same x-variables as used in MARX1, and include 4 lagged values of the response variable, but exclude lagged values of the x-variables.

Due to efficiency of markets we would not expect strong relationships with the lag covariates, and so it would be surprising if the two models proposed significantly outperform either of the regression models for imputation proposed in section 5.2. However, this fact is still worth testing. If results show that these models improve imputation performance significantly, then it may be worth considering how to apply the EM-algorithm approach to ARMA models, but this is a difficult problem and so won't be address in the current paper. We now present the two algorithm specifications.

Specification MARX1:

1. Perform the pre-transformation (5.1) on P to obtain $Z = [Z^{(1)}, \ldots, Z^{(5)}]$.

2. Set $j = 1$ and let T be the number of rows in Z.

3. Initially set $Y = Z^{(j)}$ and $X = Z^{(5)}$.

4. Apply the EM algorithm to form an initial imputed data set $\tilde{Y}$[9].

5. Define the additional covariate $Z^{(6)} = \tilde{Y}(1 : T-1, :)$ and $\tilde{y}_1 = \tilde{Y}(1, :)$.

6. Set $Y = Z^{(j)}(2 : T, :)$, $X = \left[Z^{(5)}(2 : T, :), Z^{(5)}(1 : T-1, :), Z^{(6)}\right]$.

[8] When the last row of Y has no missing values then it is more efficient to immediately return to step 7, rather than unnecessarily performing the EM algorithm

[9] This step is necessary as the covariate matrix must not have any missing values.

7. Apply the EM algorithm to Y and X to form the current imputed data set $\tilde{Y}$.
8. Set $Z^{(6)} = [\tilde{y}_1; \tilde{Y}(1:T-2,:)]$[10].
9. Repeat from step 6 until convergence.
10. Set $\tilde{Z}^{(j)} \leftarrow [\tilde{y}_1; \tilde{Y}]$.
11. Repeat the algorithm from step 3 for $j = 2, 3$ and 4.
12. Set $\tilde{Z} = [\tilde{Z}^{(1)}, \ldots, \tilde{Z}^{(4)}, Z^{(5)}]$.
13. Perform the post-transformations in section 5.1 to obtain the imputed values $\tilde{\tilde{P}}$.

Specification AR5X:

1. Perform the pre-transformation (5.1) on P to obtain $Z = [Z^{(1)}, \ldots, Z^{(5)}]$.
2. Set $m = 1$, let T be the number of rows in Z and M the number of columns in $[Z^{(1)}, \ldots, Z^{(4)}]$.
3. Initially set $Y = Z(:, m)$ and $X = Z^{(5)}$.
4. Apply the EM algorithm to form an initial imputed data set $\tilde{Y}$.
5. Define the additional covariate $Z^{(6)} = [\tilde{Y}(1:T-5), \tilde{Y}(2:T-4), \tilde{Y}(3:T-3), \tilde{Y}(4:T-2), \tilde{Y}(5:T-1)]$ and let $\tilde{y}_1 = \tilde{Y}(1:5)$.
6. Set $Y = Z(6:T, m)$ and $X = \left[Z^{(5)}(6:T,:), Z^{(6)}\right]$.
7. Apply the EM algorithm to Y and X to form the current imputed data set $\tilde{Y}$.
8. Set $\tilde{Y} \leftarrow [\tilde{y}_1; \tilde{Y}]$.
9. Set $Z^{(6)} = [\tilde{Y}(1:T-5), \tilde{Y}(2:T-4), \tilde{Y}(3:T-3), \tilde{Y}(4:T-2), \tilde{Y}(5:T-1)]$.
10. Repeat from step 6 until convergence.
11. Set $\tilde{Y}^{(m)} \leftarrow \tilde{Y}$.
12. Repeat the algorithm from step 3 for $m = 2, \ldots, M$.
13. Set $\tilde{Z} = [\tilde{Y}^{(1)}, \ldots, \tilde{Y}^{(M)}, Z^{(5)}]$.
14. Perform the post-transformations in section 5.1 to obtain the imputed values $\tilde{\tilde{P}}$.

[10]The influence of the first observation should diminish quickly as the number of iterations of the algorithm increases

6 Assessment results

Analysis of the data was performed using the financial panel/time series data from the Eur$^{\text{E}}$dit project. For a full description of this data and how missing observations were generated see the documentation associated with the data. A total of 87 daily time series covering the time period from the beginning of 1995 to the end of 1999 were used in the analysis, 36 of which are bond time series. The various algorithms described in section 5 were applied to the data. For all methods it was found that the convergence of the EM algorithm was extremely fast and reliable, usually requiring less than 5 iterations. Unfortunately, this was not the case for the derivatives. The EM method, in all cases, required over 100 iterations for the derivative instruments, and sometimes they did not converge at all. It was therefore decided to remove the 36 derivative instruments from the analysis, and to deal with these separately in the following section (section 7). Thus, all the results in this section exclude the 36 derivative instruments. In addition, for purposes of comparison, the last-value carried forward (LVCF) method was also included in the analysis. In an earlier paper Kokic (2001) found that the LVCF method worked best out of all the simple imputation methods (these included linear interpolation, the Black-Scholes pricing formula for options and term structure pricing for bonds).

Assessment was performed on the basis of two criteria, distributional accuracy and predictive accuracy as defined in Chambers (2000). Note that a fuller set of assessments will be performed in a later stage of the Eur$^{\text{E}}$dit project. *In all cases assessment was performed on the pretransformed log-return data because, on practical grounds, this is the most sensible to use.* In addition, observations where the log return of the non-missing data equals zero were excluded from the analysis, because they have already been imputed at their original source and it would bias the results in favour of the LVCF technique if they were included in the assessment. In fact, excluding these observations only had an impact on some of the distribution assessment results.

For the first assessment criterion the Wald statistic was used, see expression (14) of Chambers (2000). Specifically, this statistic and the corresponding p-value, computed on the basis of a χ^2 approximation, was determined over all imputed observations separately for each time series. The resulting set of p-values were then summarised using box plots as shown in figures 1-5 in the appendix. Note that in these figures small values of p close to zero indicate a significant departure from preservation of distribution. For predictive accuracy expression (19) in Chambers (2000) with $w_i = 1$ was used. This statistic can be interpreted as the average error of imputation. In effective it is a relative measure because the log-return data is a rate of change variable. Again the statistic was computed separately for each time series and then the set of results were summarised using box plots (see figures 6-10).

To briefly describe these results let us begin by looking at distributional accuracy. Examining figures 1 and 5, one immediately sees that the simple LVCF method performs worse of all and, not surprisingly, all methods perform worse the greater degree of missingness. The method that holds up best against this downward trend with increasing degree of missingness is R1, while the worst by far is LVCF. The remaining methods perform almost equally as well and are only slightly worse than the R1 method. In particular, there is no evidence of any additional benefits from the more sophisticated non-parametric approaches or the time series approaches compared to the two regression

techniques.

In terms of predictive accuracy there are very little differences between the results. Figure 6 shows that the LVCF method is slightly worse than the other approaches, otherwise there is little to distinguish the remain imputation methods. All methods predict bonds more accurately than shares (Figures 9 and 10). In terms of degree-of-missingness (figures 7 and 8) there seems to be little, if any, improvement over the LVCF technique.

7 Option pricing results

As mentioned in the previous section, applying the EM-algorithm directly to the log-returns of the option prices was not successful. The algorithm typically required several hundred iterations to converge, and often it did not converge at all. The solution to this problem is to apply the EM-algorithm directly to the missing volatilities data. The reason for doing this is the following. In banks the Black-Scholes pricing formula is almost exclusively used to price European call and put options, and to be effective this formula requires accurate estimates of the strike-to-underlying ratio and the volatility index. The first term can usually be estimated accurately, whereas the second is more problematic. The volatilities are normally estimated by inverting the Black-Scholes pricing formula, but this can only be done when the derivative price is known. Thus one obtains missing volatilities exactly where there are missing derivative prices[11]. We used three methods for imputing the missing volatilities, and hence obtain imputed prices: the standard basic method as described in Kokic (2001) (BSBASE), carrying the last volatility forward approach (BSLVCF), and using the EM algorithm together with a more sophisticated multivariate regression model applied to the log returns of the volatilities (BSEM). For simplicity, we only included an intercept term in the regression model as it made little sense to include the log-returns of the index data $Z_{(5)}$.

The results presented in Appendix A.3 clearly demonstrate the enormous improvement in performance of both the BSLVCF and BSEM approaches compared to the BSBASE method. Overall the BSLVCF method seems best of all, although for predictive accuracy BSEM works slightly better when the degree-of-missingness is medium or high (see figure 12). It is somewhat surprising that the BSEM approach is outperformed by the simpler BSLVCF method in terms of distributional accuracy (figure 11). Perhaps the reason for this is that the multivariate normality assumption underlying the BSEM approach is not valid and hence outliers are adversely affecting its performance.

References

Chambers, R. (2000). Evaluation Criteria for Statistical Editing and Imputation. EUREDIT working paper, University of Southampton, Southampton, UK.

Kokic, P. (2001). Standard methods for imputing missing values in financial panel/time series data. Working paper 2, QANTARIS GmbH, Frankfurt am Main.

[11]Note that the missing stock prices have been imputed by the LVCF method. Although a more sophisticated approach could have been used, this was expected to have little impact on the results because usually the locations of the missing option price and the missing underlying stock prices do not correspond with each other.

Little, R. J. A. and D. B. Rubin (1987). *Statistical Analysis with Missing Data*. New York: John Wiley & Sons.

Marquardt, D. and R. Snee (1975). Ridge Regression in Practice. *American Statistician 29*(1), 3–20.

The MathWorks, Inc. (1999). *MatLab The Language of Technical Computing: Using MatLab Version 5*. Natick, USA: The MathWorks, Inc.

A Figures

A.1 Wald statistics

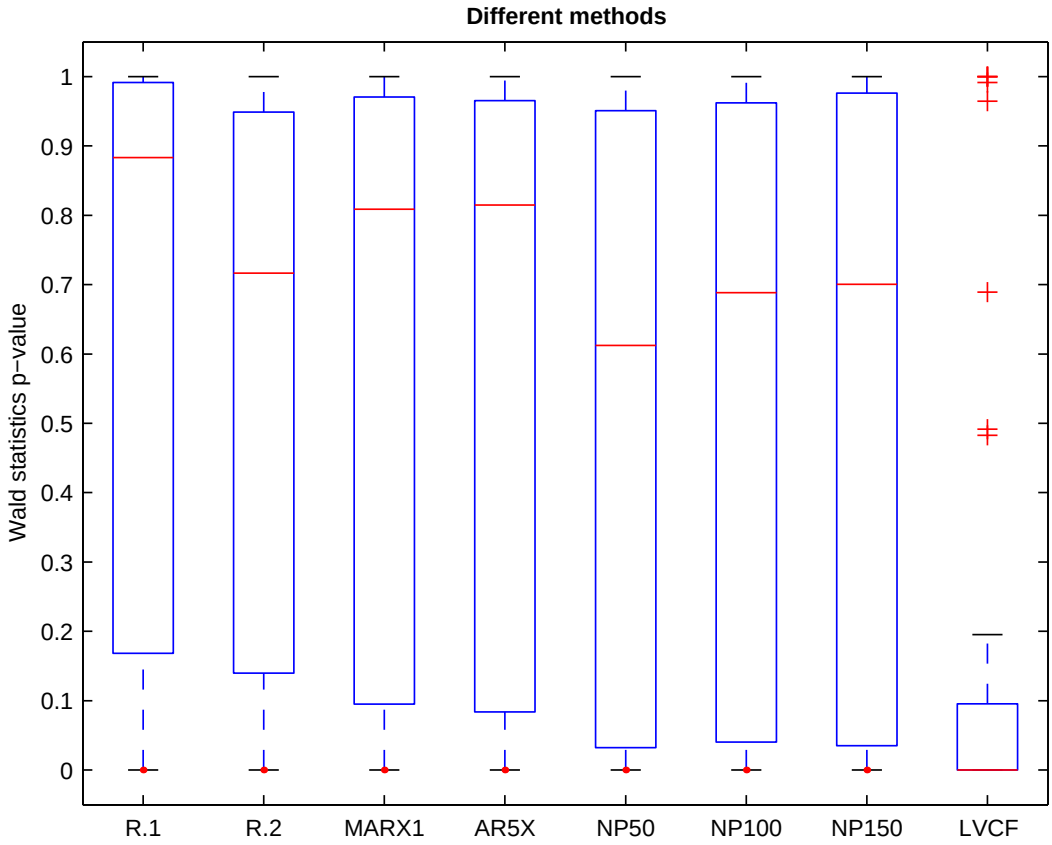

Figure 1: Distributional accuracy of the log-return imputed values

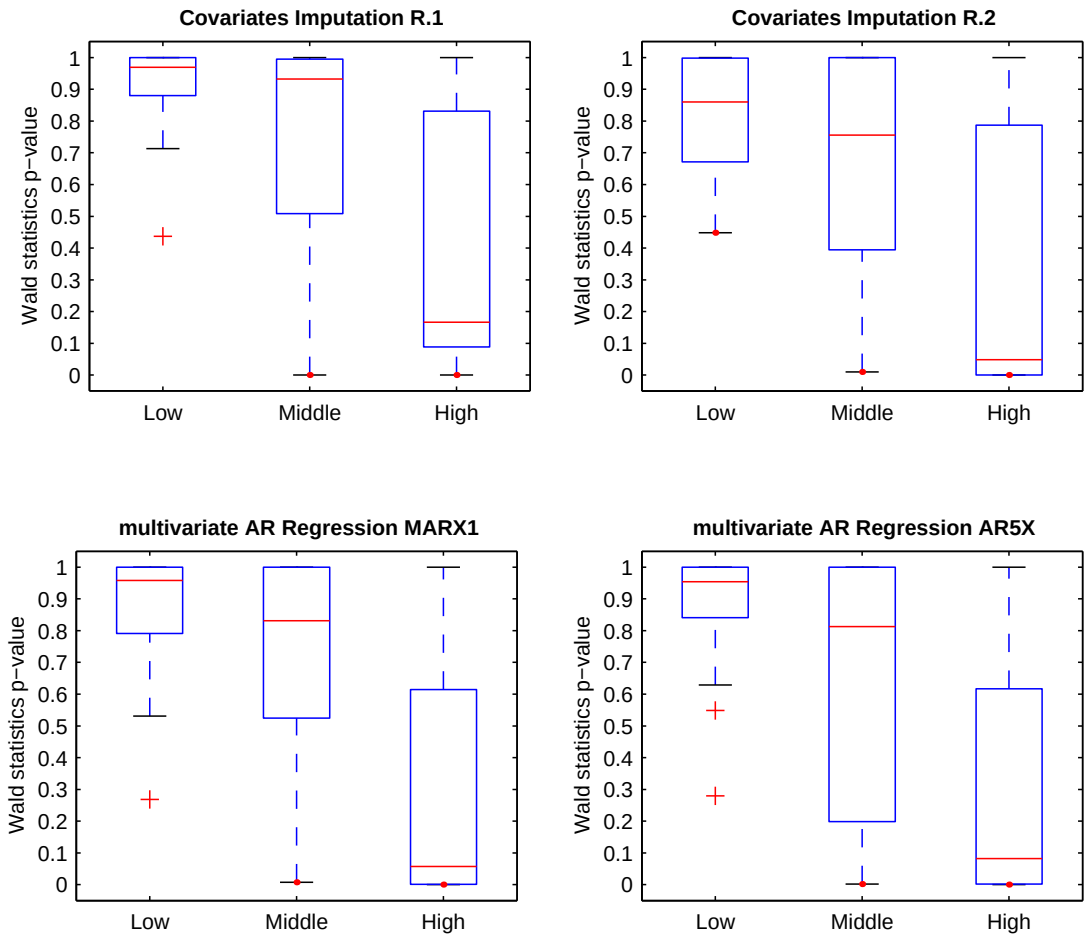

Figure 2: Distributional accuracy of the log-return imputed values by degree-of-missingness

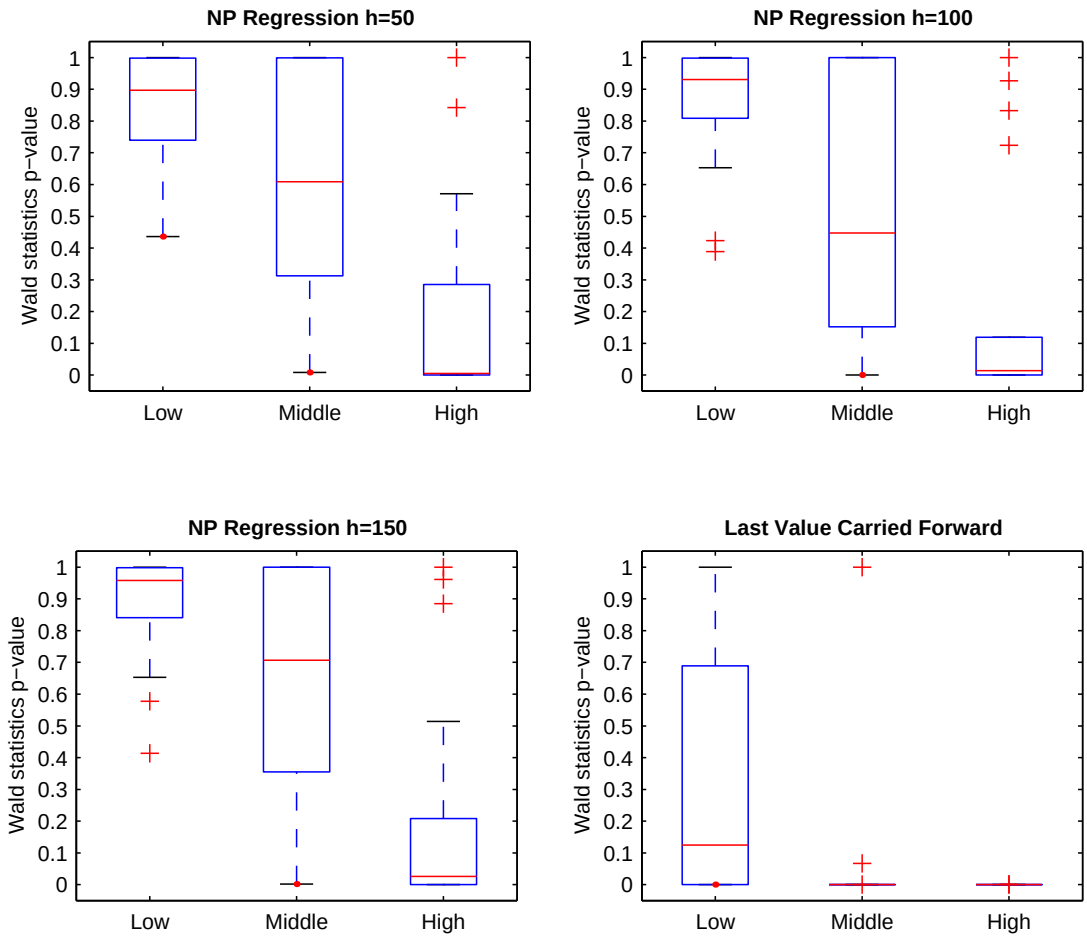

Figure 3: Distributional accuracy of the log-return imputed values by degree-of-missingness

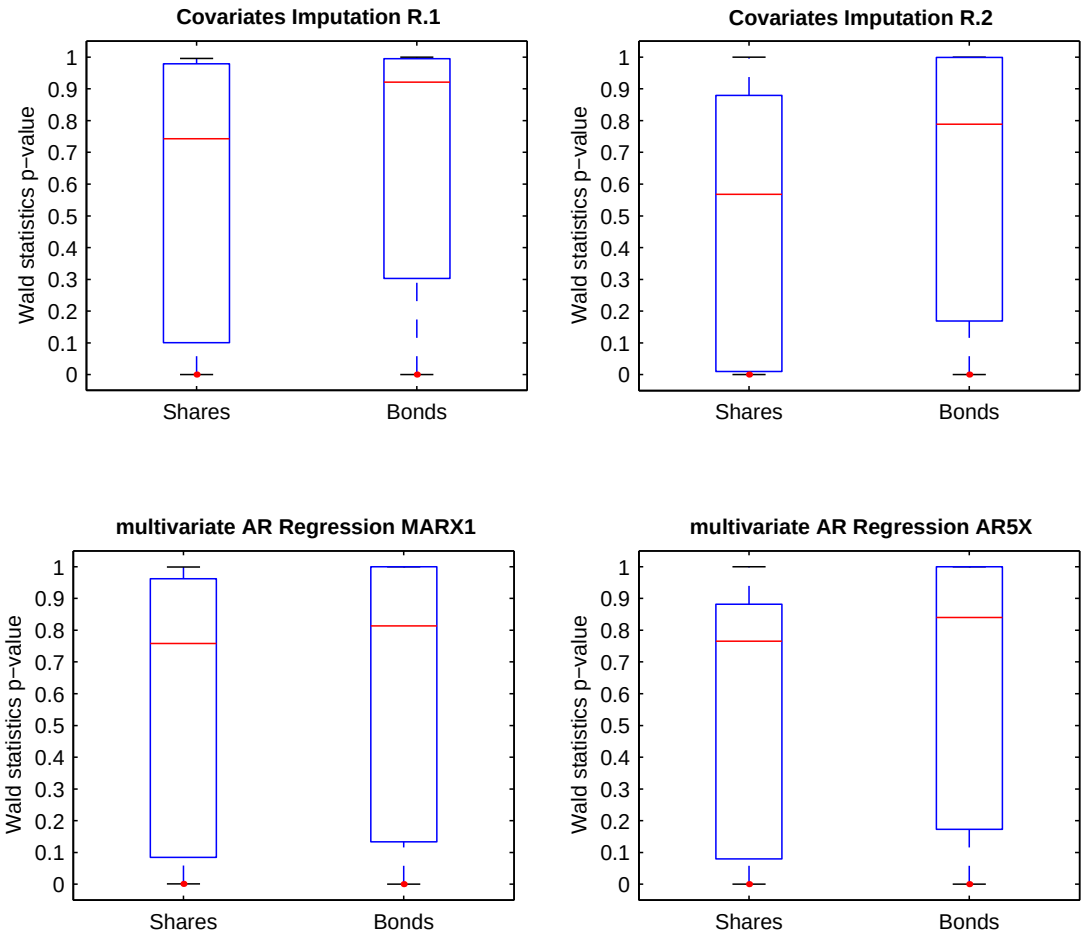

Figure 4: Distributional accuracy of the log-return imputed values by instrument type

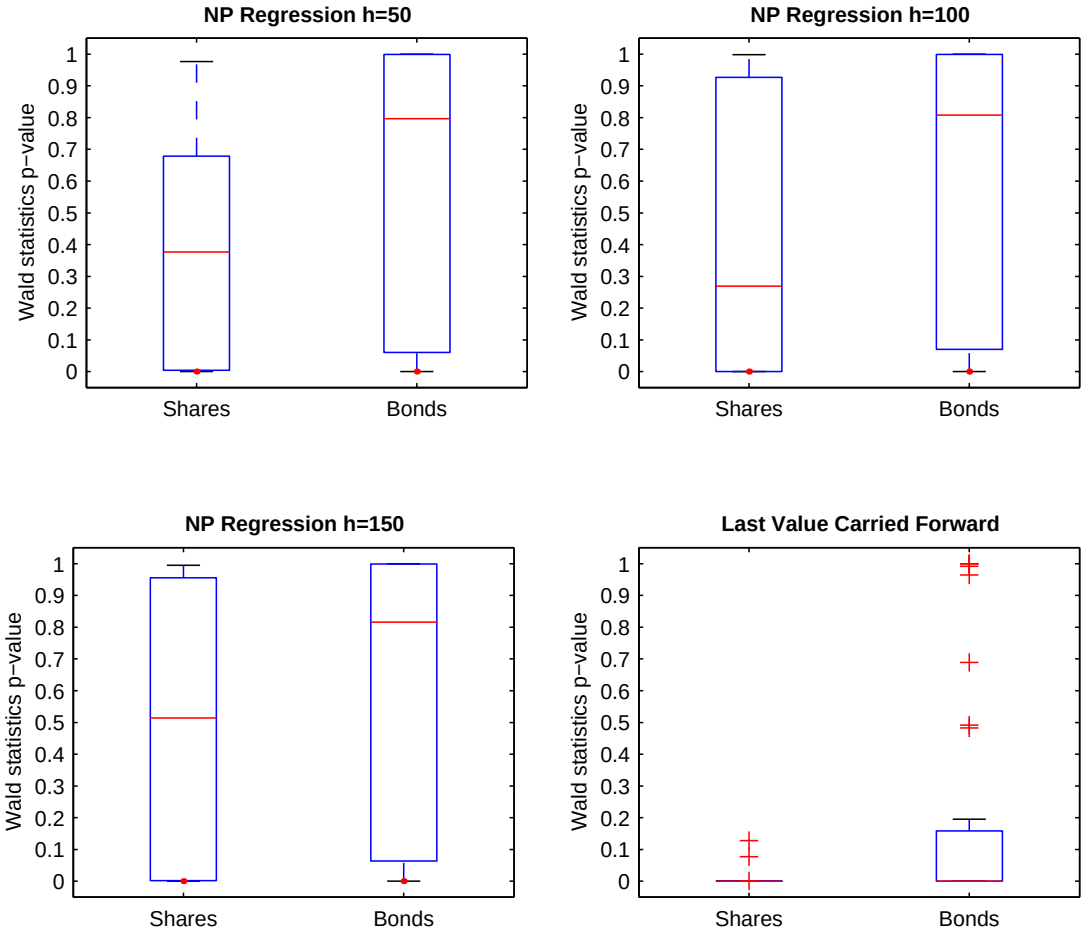

Figure 5: Distributional accuracy of the log-return imputed values by instrument type

A.2 Distance statistics

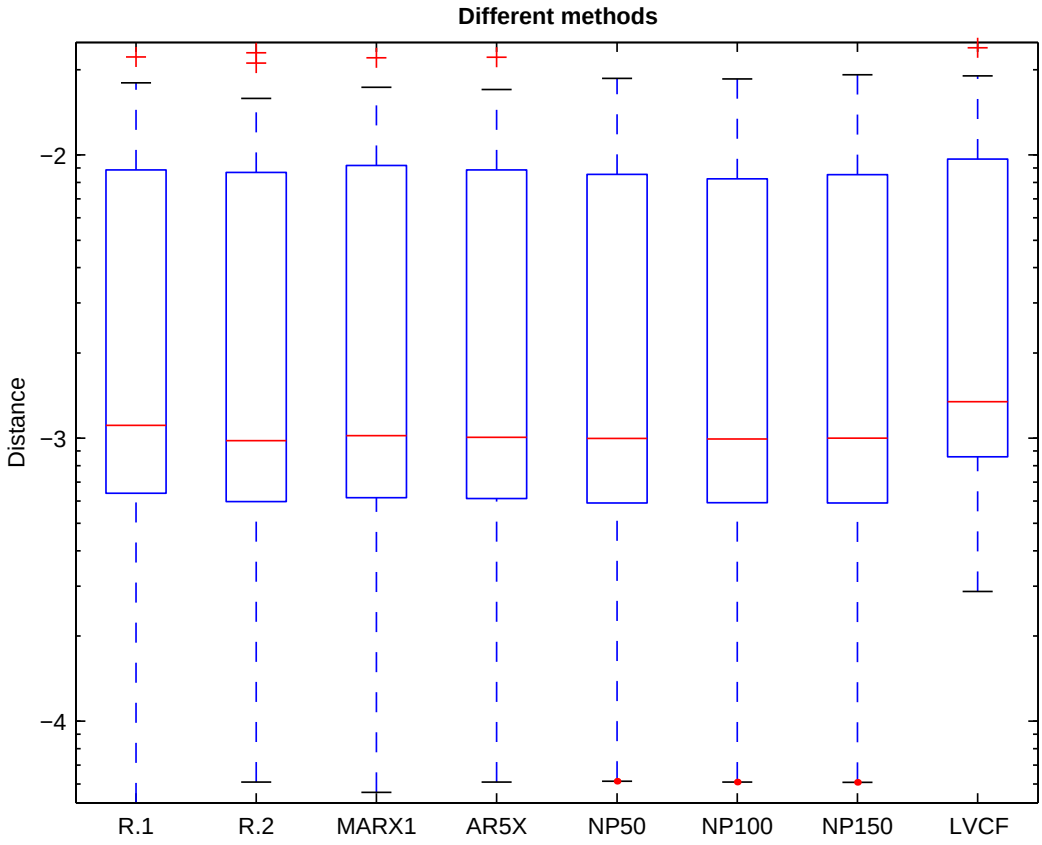

Figure 6: Predictive accuracy of the log-return imputed values

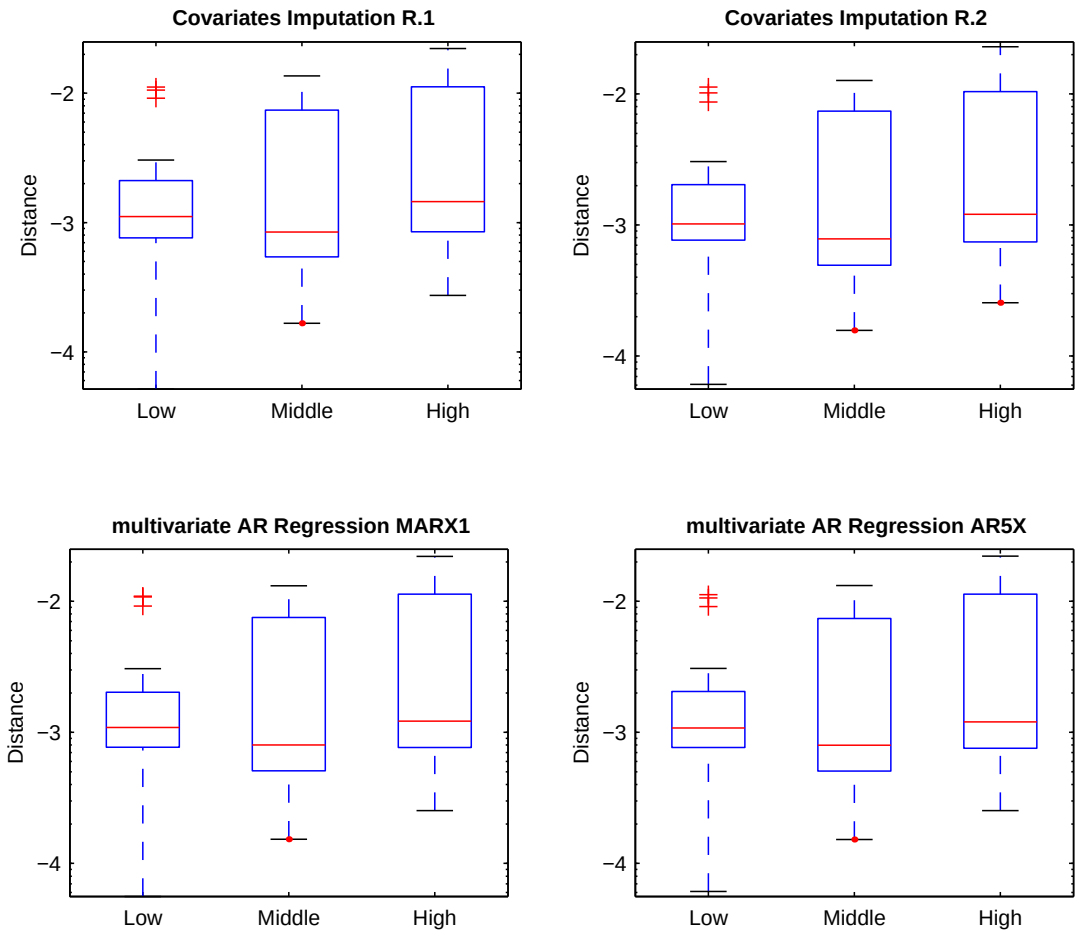

Figure 7: Predictive accuracy of the log-return imputed values by degree-of-missingness

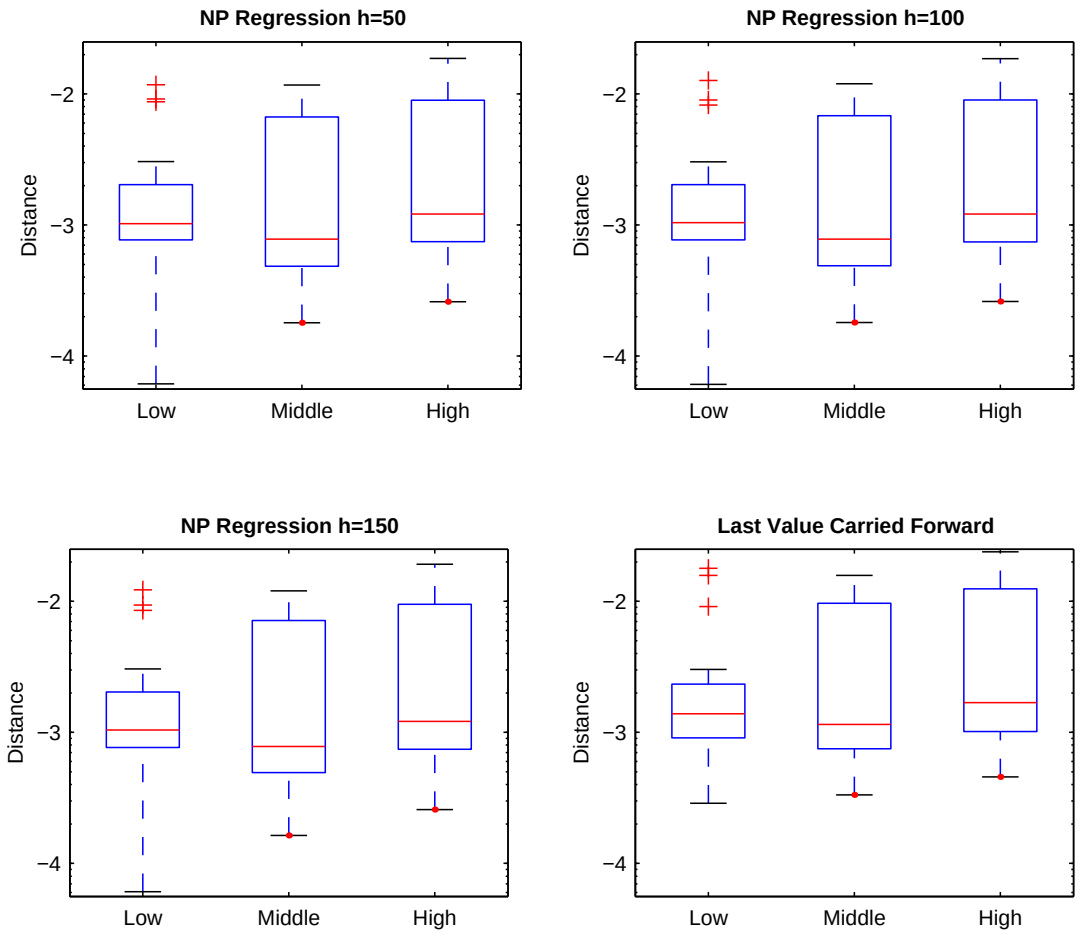

Figure 8: Predictive accuracy of the log-return imputed values by degree-of-missingness

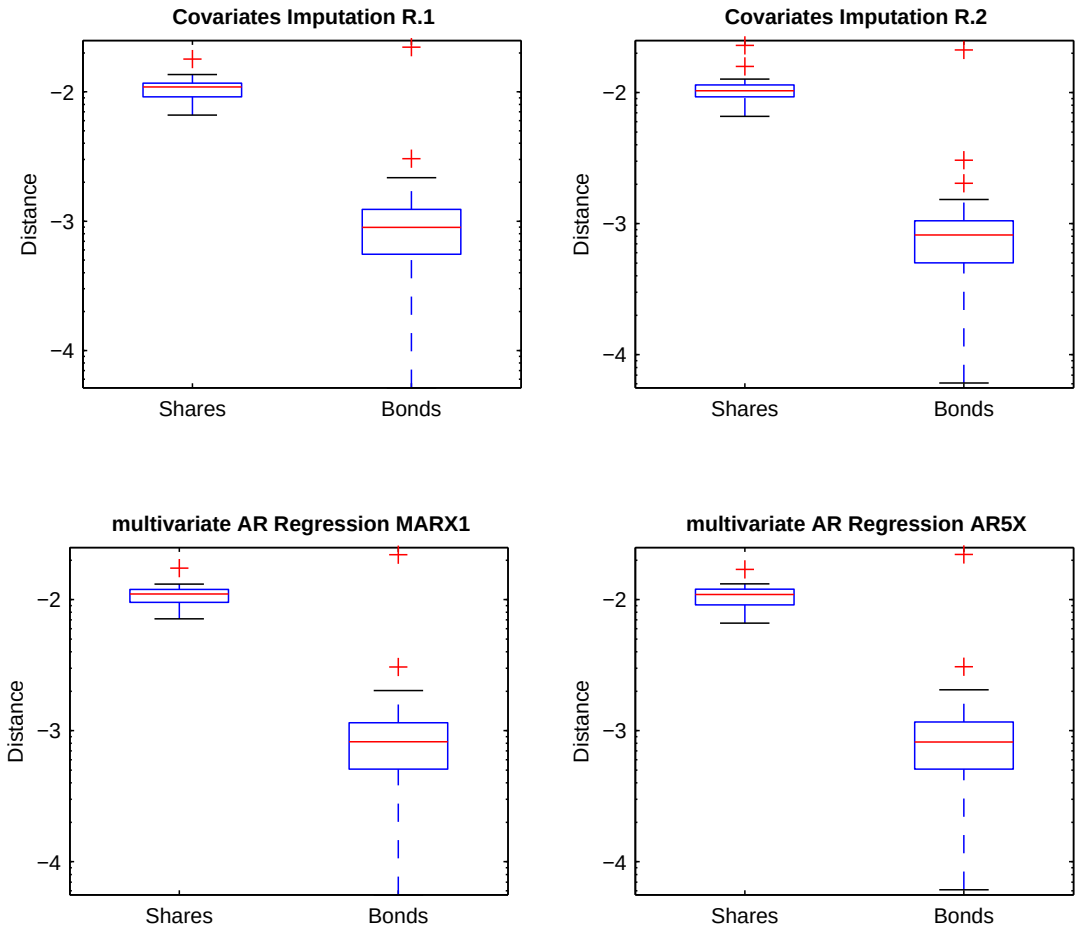

Figure 9: Predictive accuracy of the log-return imputed values by instrument type

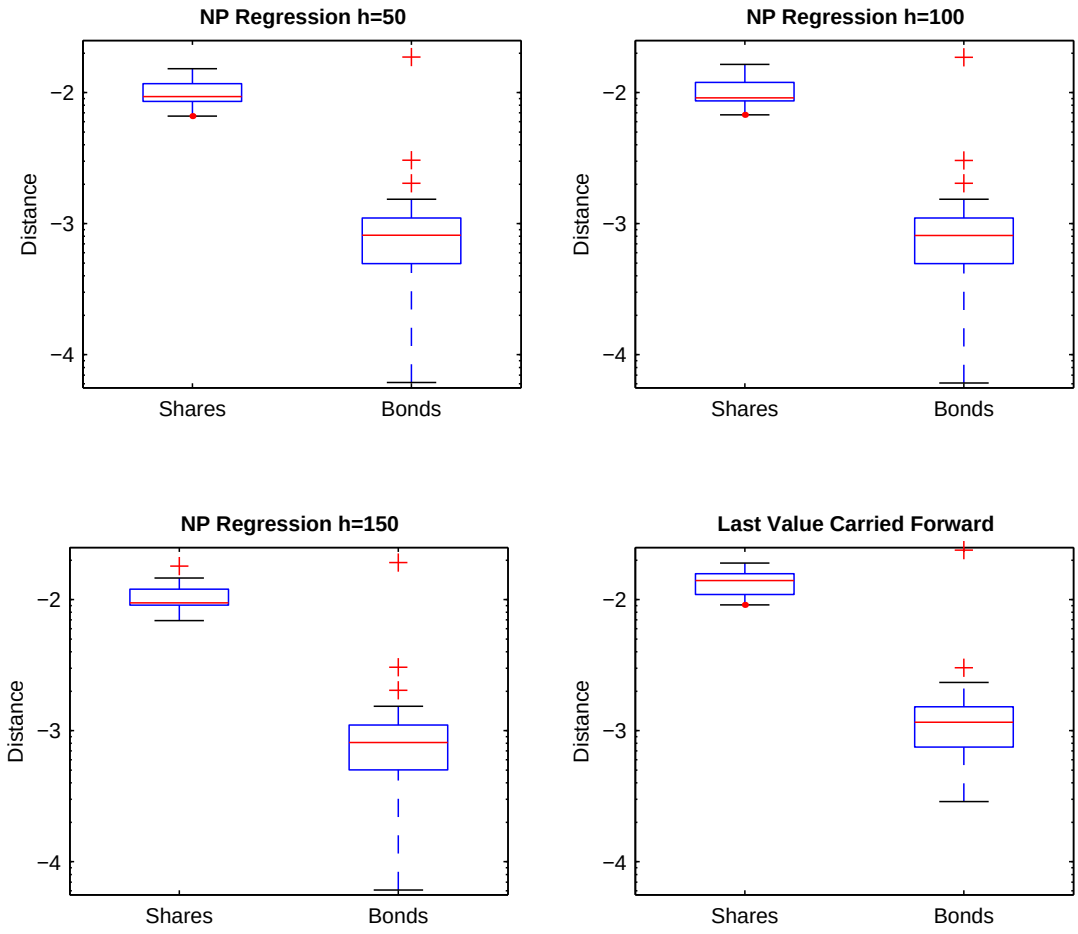

Figure 10: Predictive accuracy of the log-return imputed values by instrument type

A.3 Option pricing results

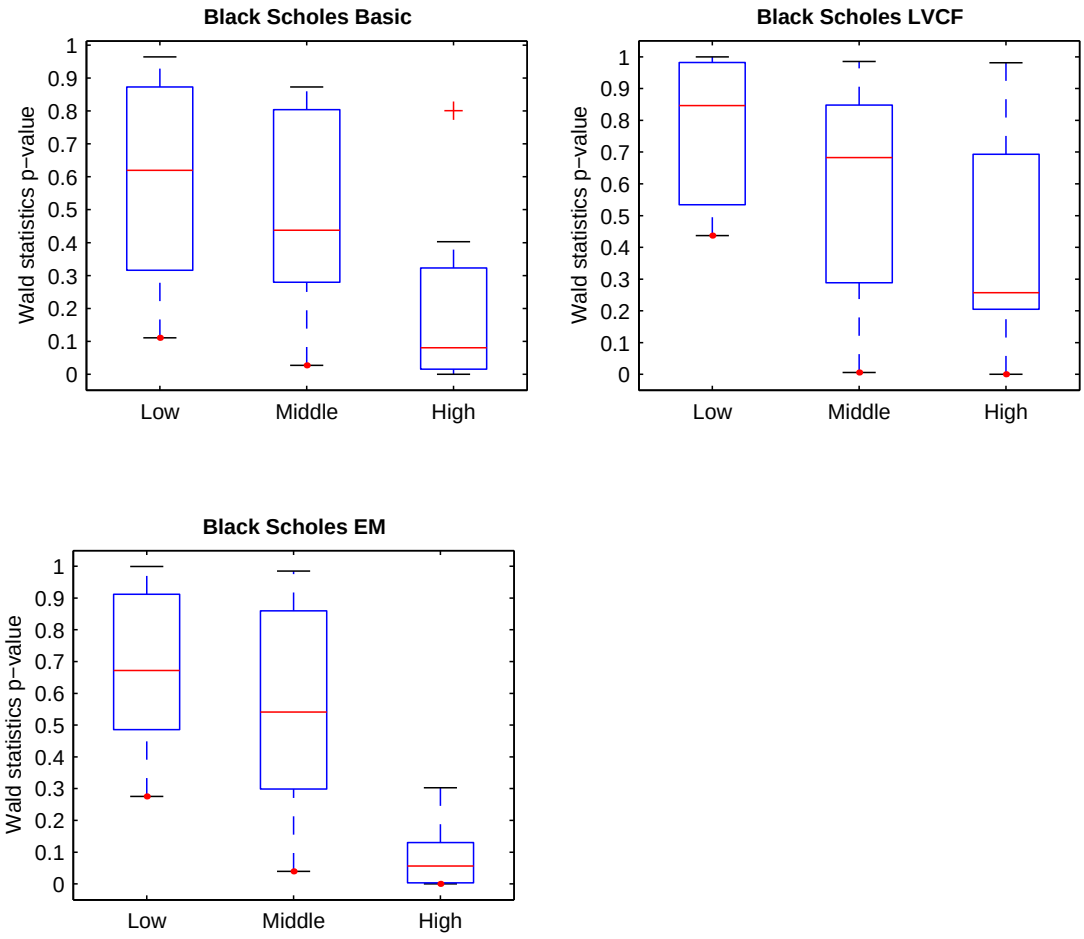

Figure 11: Distributional accuracy for option pricing methods by degree of missingness

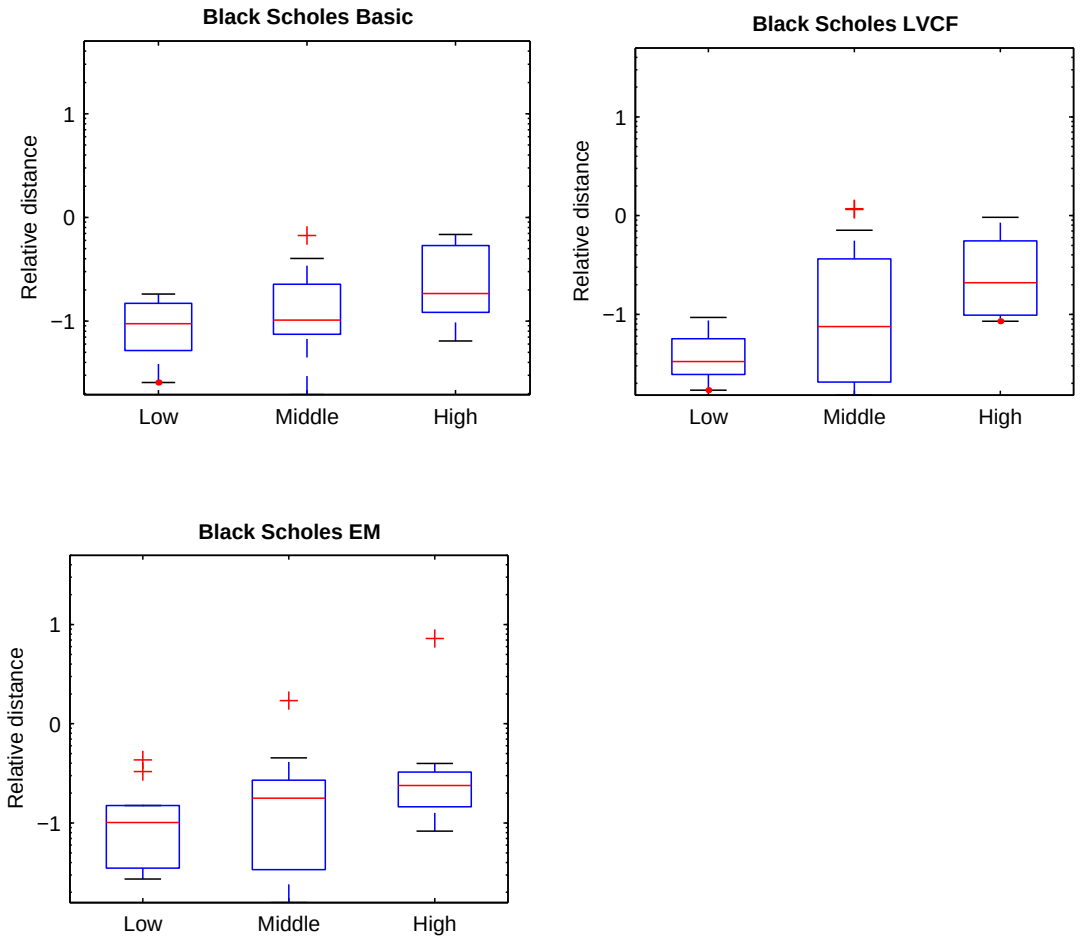

Figure 12: Predictive accuracy for option pricing methods by degree of missingness