Three Essays in Econometrics:
Multivariate Long Memory Time Series
and
Applying Regression Trees to Longitudinal Data

by

Rebecca J. Sela

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Statistics

New York University

May 2010

Clifford Hurvich, Jeffrey Simonoff

Dedication

For my husband, Amitai, my son, Jonah, and the rest of my family for their support and love in my research and beyond.

Acknowledgements

Many people have provided guidance and support for this dissertation. My two advisors, Clifford Hurvich and Jeffrey Simonoff, have been the source of inspiration and interesting conversations over the last six years. I also thank my the members of my committee, Robert Engle, William Greene, and Gary Simon, for their suggestions. Many thanks also go to Norm White and the others at NYU who have kept the Stern Grid running, allowing me to run thousands of hours of simulations for my various projects. Rohit Deo, Foster Provost, and attendees of numerous conferences (the Stern-Wharton Conference on Business in Statistics, Quantitative Social Science Research Using R Conference at Fordham University, and the Joint Statistical Meetings in 2008 and 2009) have provided helpful comments on the work in this dissertation.

My husband, Amitai Sela, has been a source of love, humor, and support, keeping me sane and balanced as I wrote this dissertation. He has also provided invaluable technical support, allowing me to work at home as well as at NYU. My son, Jonah Sela, born just before my dissertation research began, has kept me grounded and laughing throughout the process. Many, many thanks also go to my family and friends for their love and support over the last few years. Rita Sela has cared for Jonah and helped with household chores, letting me focus on my research. Karen Paul was a source of teaching inspiration. My parents, Chuck and Carolyn Paul, have listened to my adventures in research, providing moral support. Katherine Randle has also cared for Jonah. Many friends and family, including Amy Finkbiner, Amitai Sela, Rita Sela, and the entire Paul family, have read earlier versions of the papers in this dissertation, providing helpful comments.

Abstract

The first two chapters of this dissertation discuss multivariate long memory models. First, we discuss two distinct parametric multivariate time-series models. We discuss the implications of the models and describe an extension to fractional cointegration. We describe algorithms for computing the covariances of each model, for computing the likelihood and for simulating from each model. These algorithms are much more computationally efficient than the existing algorithms and are equally accurate, making it feasible to model multivariate long memory time series and to simulate from these models. We use maximum likelihood to fit models to data on goods and services inflation in the United States.

Second, we present a semiparametric model for bivariate long-memory time series that allows for power law behavior in the coherency and powers of the frequency in the phase. We describe the implications of a power law in the coherency and of powers of the frequency in the phase on the time-domain behavior of the time series and provide time domain examples. We prove the consistency of the averaged periodogram estimator for estimating the power law in the cross-spectrum and coherency. We prove that the very-narrow-band least squares estimator of the cointegrating parameter is not affected by power laws in the phase and coherency. We apply our methods to money supply data and to high and low stock prices.

The final chapter presents a methodology that combines the flexibility of tree-based estimation methods with the structure of random effects models for longitudinal data. We apply the resulting model and estimation method, called the RE-EM tree, to state traffic fatality rates and to pricing in online transactions. We also perform extensive simulation experiments to show that the estimator improves predictive performance relative to regression trees without random effects and is comparable or superior to using linear models with random effects.

Contents

List of Figures

List of Tables

1 Introduction

This dissertation consists of three papers dealing with different facets of econometrics. The first two chapters deal with different facets of multivariate long memory time series, while the third applies data mining methods to longitudinal data.

Univariate long memory time series have been widely discussed in literature. However, very few papers had previously considered models for two or more long memory time series. Chapter 2 takes a parametric approach: it describes two distinct multivariate extensions of the ARFIMA model to the multivariate case. It then presents efficient algorithms for calculating the covariance sequences of the two models, for simulating from the two models (and, in fact, many multivariate Gaussian models), and for computing the Gaussian likelihood. In contrast, Chapter 3 uses a semiparametric approach: it decomposes the cross-spectrum of a bivariate time series into the auto-spectra, the phase and coherency. It then presents a semiparametric model that allows for a wide range of behaviors in the phase and coherency, including power laws in the phase and infinite group delay at zero frequency.

Chapter 4 is quite different from the other two chapters, applying regression trees to longitudinal data. Most previous work on regression trees has focused on cross-sectional data. The fact that longitudinal data includes multiple observations per individual allows for improved estimation and prediction, if a model uses the longitudinal structure. RE-EM trees combines the flexibility of regression trees with individual-specific effects to account for the structure of the data. This allows for great flexibility in modelling and improves predictive performance.

2 Computationally Efficient Gaussian Maximum Likelihood Methods for Vector ARFIMA Models

2.1 Introduction

While time series often come in groups that could be analyzed together, much time series work focuses on the analysis of univariate time series. This has led to the creation of a wide variety of models that can handle many types of correlation structure, including long memory processes which have slowly-decaying autocorrelations (see Granger and Joyeux [1980] and Hosking [1981] for some of the earliest work in this area). In the case of multiple stationary time series, the most widely-used model is a vector autoregressive-moving average (ARMA) model, in which the autocorrelations of each component series and therefore the cross-correlations between pairs of series decay exponentially fast. Such a restriction on the autocorrelations has been found to be too strong in a variety of univariate cases. Instead, many authors suggest applying a long memory model such as an fractionally integrated ARMA (ARFIMA) model, to such time series (see Baillie [1996] for a discussion of applications to geophysical sciences, macroeconomics, prices, and more). In this paper, we discuss two vector versions of the ARFIMA model, both of which are multivariate generalizations of the traditional univariate ARFIMA model.

To make a time series model suitable for practical use, it is desirable to be able to determine its covariance structure, estimate its parameters through maximum likelihood, and simulate from it. Ideally, all of these tasks must be done both quickly and precisely. In the case of univariate and multivariate ARMA models,

the conditional likelihood function, in which some initial values of the time series are assumed to be fixed, provides a simple approximation to the full likelihood function. The application of the EM algorithm of Dempster et al. [1977] to the state-space representation of a multivariate ARMA process provides an alternative estimation method. (See Hamilton [1994, chapter 11 and section 13.4] for more information.) However, neither of these methods is applicable to long memory models, because one cannot condition on a finite number of observations and because long memory models do not have state space representations (Baillie [1996] and others). For univariate ARFIMA models, more recent work [Bertelli and Caporin, 2002, Deo et al., 2006, Davies and Harte, 1987] has found efficient methods for computing the autocovariances of an ARFIMA process, computing the likelihood function of an ARFIMA process, and simulating from an ARFIMA process. Previously, Sowell [1989b,a] described exact methods for computing the covariances from one particular type of vector ARFIMA model and for computing the exact likelihood and simulating from general multivariate processes. However, his calculation methods are often slow, with the likelihood and simulation calculations taking $O(T^2)$ time, where T is the number of observations in the dataset; reliance on these algorithms makes the use of vector ARFIMA models prohibitively expensive for large datasets. In this paper, we present methods which will accomplish the tasks of computation and simulation fast enough to make the use of vector ARFIMA models more practical.

Beyond the application of the newly proposed methods to estimating vector ARFIMA models, our algorithms for computing the quadratic form and for simulation are applicable to any multivariate time series for which the covariance structure is known. This provides additional value to people who wish to compute the quadratic form of or to simulate from a multivariate time series that does not

have state space representations or other methods for exact computation.

To make our notation precise, suppose that we observe $k = 1, ..., K$ time series over $t = 1, ..., T$ periods, with X_{kt} denoting the t^{th} period of the k^{th} time series and $X_t = (X_{1t}, ..., X_{Kt})'$. In this paper, unless stated otherwise, we will assume that all time series are stationary with zero mean. We will consider these observations grouped either by series or by time. In the former case, we will write $X = (X_{1\cdot}, ..., X_{K\cdot})'$, where $X_{k\cdot} = (X_{k1}, ..., X_{kT})'$. In the latter case, we will write $\tilde{X} = (X_1', ..., X_T')'$. Notice that $X = P\tilde{X}$, where P is a permutation matrix. Suppose we have a model for X described by a vector of parameters, θ. Define $\Omega(\theta)$ as the $KT \times KT$ matrix, $\text{Cov}(X)$. Note that $\Omega(\theta)$ consists of K^2 blocks, with the (i, j) block equal to the $T \times T$ matrix containing $E(X_{i\cdot}X_{j\cdot}')$. Since the multivariate process is stationary, each block is Toeplitz, with the same number along each diagonal. Alternatively, we may consider $\tilde{\Omega}(\theta) = \text{Cov}(\tilde{X})$. Then, $\tilde{\Omega}(\theta)$ consists of T^2 blocks containing $\text{Cov}(X_t, X_{t-r})$, arranged in a Toeplitz fashion, so that the blocks along each diagonal are identical. We may then write the Gaussian log likelihood as:

$$l(\theta|X) = -\frac{1}{2}\log|\Omega(\theta)| - \frac{1}{2}X'\Omega(\theta)^{-1}X \tag{2.1}$$

$$= -\frac{1}{2}\log|\tilde{\Omega}(\theta)| - \frac{1}{2}\tilde{X}'\tilde{\Omega}(\theta)^{-1}\tilde{X} \tag{2.2}$$

We will discuss how to compute the autocovariances which could be used to create $\Omega(\theta)$ and $\tilde{\Omega}(\theta)$ in section 2.5, how to compute the term containing the quadratic form in section 2.6, and how to approximate the determinant term in section 2.7.

In section 2.2 we provide some background on long memory processes and a discussion of two distinct models that appear as we move from the univariate case to the multivariate case. Section 2.3 describes block circulant and block Toeplitz matrices, which are the basis of many of the methods we will use. In section 2.4, we discuss existing computational methods that have been applied to maximum

likelihood estimation in multivariate ARFIMA models. In sections 2.5, 2.6, and 2.7, we present computationally efficient methods for the distinct tasks in estimating vector ARFIMA models with maximum likelihood: computing covariances, computing the quadratic form in the likelihood function, and computing the determinant in the likelihood function. The methods for computing the covariances and computing the quadratic form are extensions of univariate algorithms which have been discussed in the time series literature, and we review those methods in the corresponding sections. In section 2.8, we discuss simulating from a vector ARFIMA process. This section also includes a description of the existing method for univariate ARFIMA processes that our algorithm extends. After presenting these methods, we discuss the performance of the maximum likelihood estimator in section 4.6. We apply our estimator to econometric and meteorological data in section 2.10. Section 4.7 concludes.

2.2 Long Memory Processes

2.2.1 Univariate ARFIMA Processes

A univariate long memory process with differencing parameter, d, is one in which the autocovariances, $\omega(r)$, decay at a hyperbolic rate; that is, $\lim_{|r|\to\infty} \frac{\omega(r)}{|r|^{2d-1}}$ is constant. Equivalently, a univariate long memory process is a process in which the spectral density, defined as $f(\lambda) = \frac{1}{2\pi}\sum_{r=-\infty}^{\infty}\omega(r)\exp(-ir\lambda)$, obeys $f(\lambda) \sim C|1-e^{-i\lambda}|^{-2d}$ when λ is near 0. We must have $0 \leq |d| < \frac{1}{2}$, for this spectrum to be integrable and for the process to be stationary; the process is said to have short memory when $d = 0$ and long memory for any $0 < |d| < \frac{1}{2}$. Long memory processes have long been studied in the literature. (See Granger and Joyeux [1980] and Hosking [1981] for early work on long memory and Brockwell and Davis [1993, section 13.2] or Baillie [1996] for more background.)

The simplest case of long memory is fractionally integrated white noise, $\{y_t\}$. Fractionally integrated white noise is defined by $(1-L)^d y_t = \epsilon_t$, where ϵ_t is white noise with variance σ^2, and L is the lag operator, $Lx_t = x_{t-1}$. Even though d is not an integer, we can define $(1-L)^d$ by the binomial expansion:

$$\begin{aligned}(1-L)^d &= \sum_{j=0}^{\infty}(-1)^j\binom{d}{j}L^j \\ \binom{d}{j} &= \frac{d(d-1)\cdots(d-j+1)}{j!}\end{aligned}$$

The spectral density of $\{y_t\}$ is given by $f_y(\lambda) = \frac{\sigma^2}{2\pi}|1-e^{-i\lambda}|^{-2d}$. The coefficients of the infinite order autoregressive representation, the infinite order moving average representation, and the autocovariances of $\{y_t\}$ are available in closed form [Brockwell and Davis, 1993, see, for example, [Theorem 13.2.1].

ARFIMA models are a more general class of univariate long memory processes. A time series, $\{x_t\}$, follows an $ARFIMA(p,d,q)$ process if it can be written as $a(L)(1-L)^d x_t = b(L)\epsilon_t$, where $a(L)$ and $b(L)$ are lag polynomials of degree p and q respectively. We generally assume that $a(L)$ and $b(L)$ have no common roots and that all of their roots are outside the unit circle. Together with the assumption that $|d| < \frac{1}{2}$, these conditions ensure that $\{x_t\}$ is a stationary and invertible process. Notice that we may think of $\{x_t\}$ in two different ways that are equivalent in the univariate case but will not be equivalent for multivariate models. First, $\{x_t\}$ is an $ARMA(p,q)$ process driven by fractionally integrated white noise, which can be written as:

$$a(L)x_t = b(L)[(1-L)^{-d}\epsilon_t]$$

Second, we may describe $\{x_t\}$ as an ordinary $ARMA(p,q)$ process which has been fractionally integrated:

$$x_t = (1-L)^{-d}\left(\frac{b(L)}{a(L)}\epsilon_t\right)$$

Since the composition of linear filters is commutative in the univariate case, the two descriptions are identical.

2.2.2 Vector ARFIMA Processes

The composition of linear filters does not commute in the multivariate case, so there are multiple possible extensions of a univariate ARFIMA process to a vector ARFIMA process. In this paper, we will focus primarily on models with autoregressive but not moving average components, because of the additional complications associated with moving average components, particularly in a multivariate setting (see Dunsmuir and Hannan [1976, page340] for a description of the structure needed to identify the parameters in vector ARMA models). Because a vector ARMA model can be written as a vector AR model [Hamilton, 1994, page 259], many of our results generalize to models with MA components; we will identify cases in which that occurs.

Let $A(L) = A_0 + A_1 L + ... + A_p L^p$, where A_0 is the $K \times K$ identity matrix, I_K, and $A_1, ..., A_p$ are any matrices such that $|A(L)|$ has all of its roots outside the unit circle. If $p = 1$, this condition is equivalent to the requirement that A_1 has all of its singular values less than 1, or equivalently that all of the eigenvalues of $A^T A$ are less than one. Let $D(L)$ be the diagonal matrix with diagonal entries $(1 - L)^{d_1}, ..., (1 - L)^{d_K}$, where $d_1, ..., d_K \in (-\frac{1}{2}, \frac{1}{2})$, to ensure stationarity and invertibility. Let $\{\epsilon_t\}$ be a sequence of K-variate white noise, with $E(\epsilon_t \epsilon_s') = 0$ when $t \neq s$ and $E(\epsilon_t \epsilon_t') = \Sigma$, with Σ positive definite. Given the parameters $D(L)$, $A(L)$ and Σ, we may define two distinct vector ARFIMA models; versions of the models including moving average components were presented by Lobato [1997].

In the first model, called Model A by Lobato, we have:

$$A(L)D(L)X_t = \epsilon_t$$

We may understand the properties of the process, X_t, by defining it in two steps. First, define $X_t = D(L)^{-1}Z_t$, so that $X_{k\cdot} = (1-L)^{-d_k}Z_{k\cdot}$. Then, assume that $\{Z_t\}$ follows a vector autoregressive (VAR) model, $A(L)Z_t = \epsilon_t$. Combining these two parts, we see that X_t is a fractionally integrated vector autoregression, which we will called a FIVAR model in this paper. When we wish to specify p and $\vec{d} = (d_1, ..., d_K)$, we will call this a $FIVAR(p, \vec{d})$ model.

Permuting the matrices $A(L)$ and $D(L)$ gives us what Lobato calls Model B:

$$D(L)A(L)X_t = \epsilon_t$$

This model is a vector autoregressive model, $A(L)X_t = Y_t$, driven by fractionally integrated white noise, $D(L)^{-1}\epsilon_t$. In this paper, we will refer to this model as a $VARFI(p, \vec{d})$ model, where p is the order of the lag polynomials in $A(L)$ and $\vec{d}$ is the vector of differencing parameters, as before.

In the univariate case, these two models are identical. Since the composition of $A(L)$ and $D(L)$ is not necessarily commutative, however, these models differ in most cases when $K > 1$. The distinction between the models is also apparent when we write down the the spectral densities of the models:

$$\begin{aligned} f_{FIVAR}(\nu) &= \frac{1}{2\pi} D(e^{-i\nu})^{-1} A(e^{-i\nu})^{-1} \Sigma (A(e^{-i\nu})^{-1})^* (D(e^{-i\nu})^{-1})^* \\ f_{VARFI}(\nu) &= \frac{1}{2\pi} A(e^{-i\nu})^{-1} D(e^{-i\nu})^{-1} \Sigma (D(e^{-i\nu})^{-1})^* (A(e^{-i\nu})^{-1})^* \end{aligned}$$

These spectral densities are identical when the matrices describing the linear filters, $D(\cdot)$ and $A(\cdot)$, commute. In particular, they are identical when $D(L)$ is a scalar multiple of the identity matrix; this occurs when all of the series have equal differencing parameters. Also, they are identical when $A(L)$ and Σ are both diagonal; in that case, the individual series, $X_{k\cdot}$, are uncorrelated univariate ARFIMA series.

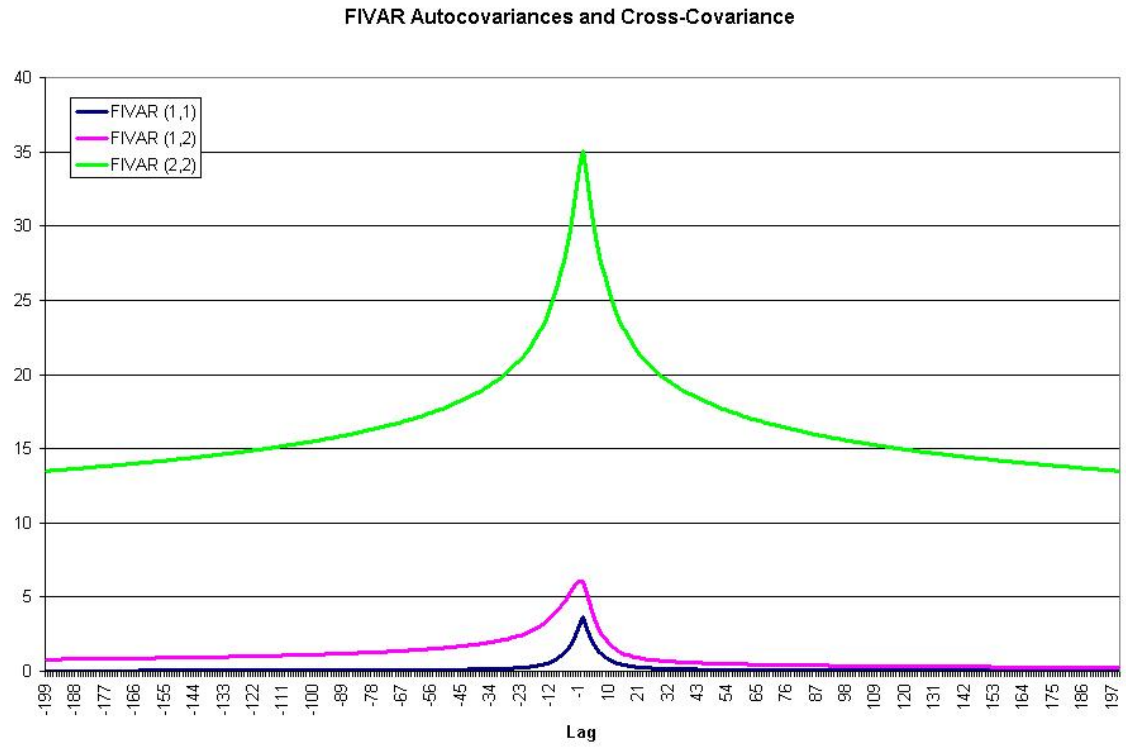

Figure 1: The theoretical autocovariance sequences and cross-covariance sequence of a $FIVAR(1, \vec{d})$ process with parameters $d = (0.1, 0.4)$, $A_1 = (0.7, 0.1, 0.2, 0.6)$, and $\Sigma = (1, .5, .5, 2)$ for lags -199 to 199.

In Figures 1, 2, and 3, we plot the autocovariance sequences and cross-covariance sequences of $FIVAR(1, \vec{d})$ and $VARFI(1, \vec{d})$ processes with identical $A(L)$, Σ, and d. The covariance sequences differ dramatically. The autocovariance sequences of the two variables decay more rapidly in the VARFI process than in the FIVAR process. The cross-covariance sequences show an even larger difference; the FIVAR process shows much more asymmetry in the cross-covariances.

Besides producing different autocovariance sequences, the two models differ in their implications; a FIVAR model cannot produce anything like fractional coin-

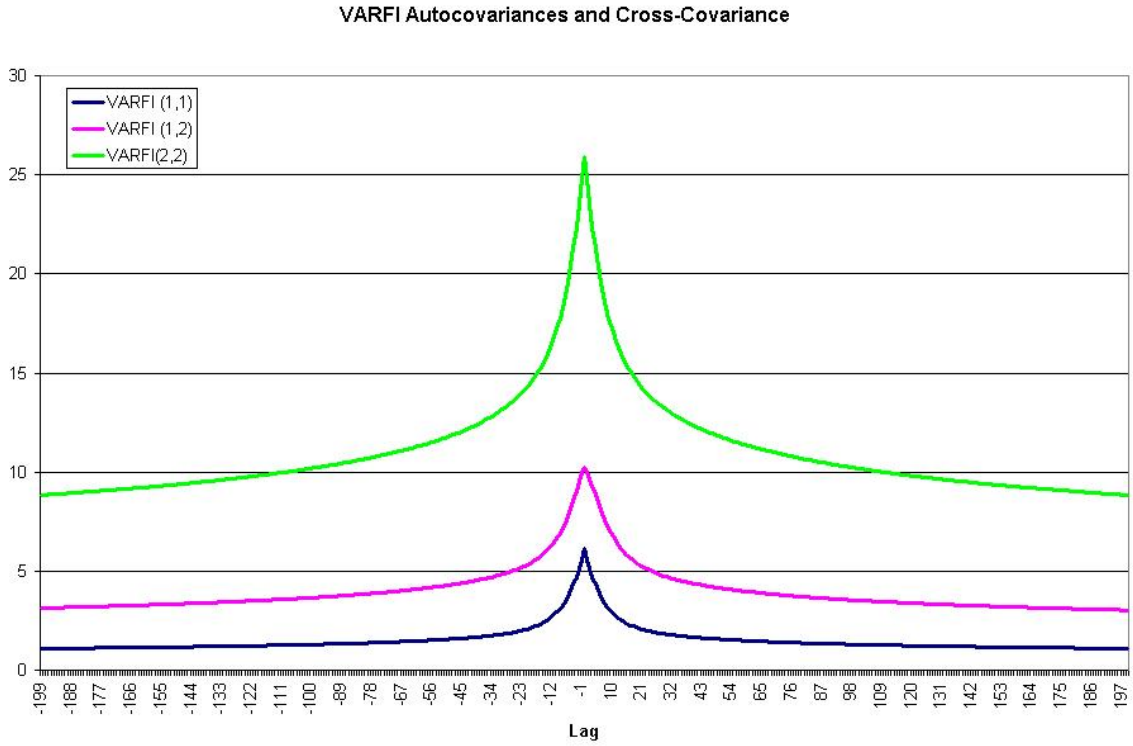

Figure 2: The theoretical autocovariance sequences and cross-covariance sequence of a $VARFI(1, \vec{d})$ process with parameters $d = (0.1, 0.4)$, $A_1 = (0.7, 0.1, 0.2, 0.6)$, and $\Sigma = (1, .5, .5, 2)$ for lags -199 to 199.

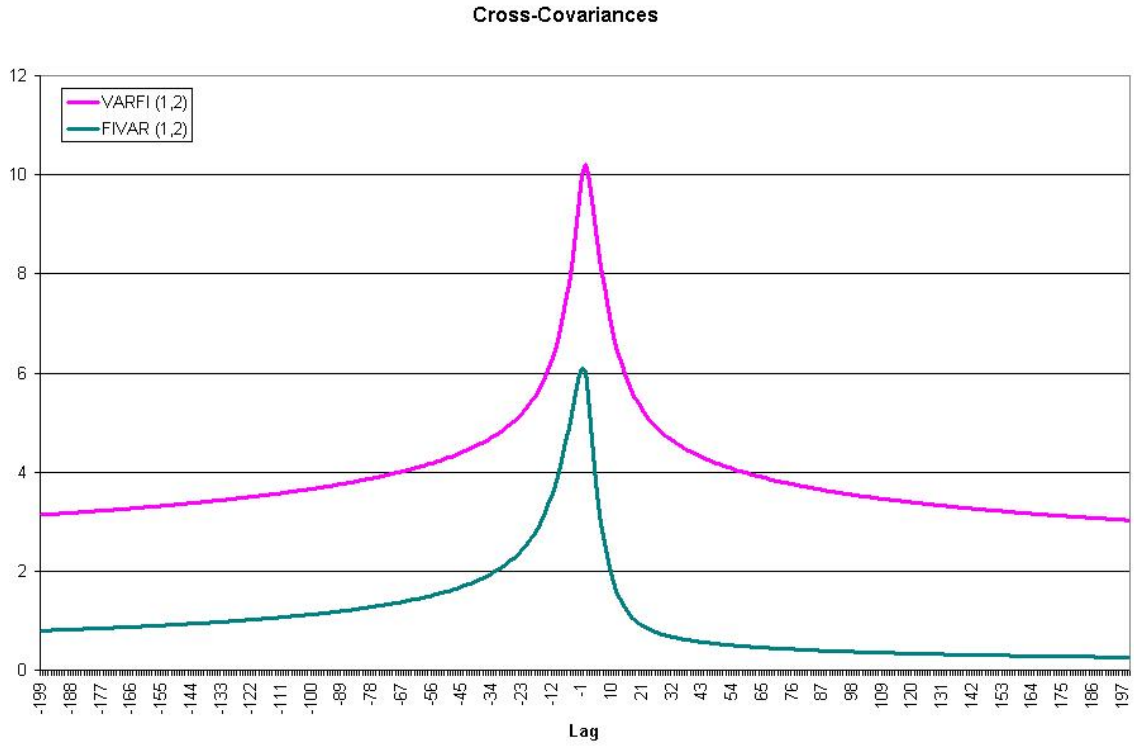

Figure 3: The theoretical cross-covariance sequences of a $FIVAR(1, \vec{d})$ process and a $VARFI(1, \vec{d})$ process, both with parameters $d = (0.1, 0.4)$, $A_1 = (0.7, 0.1, 0.2, 0.6)$, and $\Sigma = (1, .5, .5, 2)$ for lags -199 to 199.

tegration because the stationary VAR series are integrated separately. However, in most cases, a VARFI model will have linear combinations of X_t and up to p lags which are integrated of a lower order. Extending the analysis of Lobato [1997, page 141] from the bivariate case to a general multivariate case, we give a simple formula that describes the cointegrating relationships. Let $A_{\cdot,k}(L)$ be the k^{th} row of $A(L)$. Then, $A_{\cdot,k}(L)X_t = (1-L)^{d_k}\epsilon_{kt}$. Suppose that $d_k < \max(\vec{d})$ and at least two elements of $A_{\cdot,k}(L)$ are non-zero, and that the corresponding X_{kt} are integrated of order $\max(\vec{d})$. Then $A_{\cdot,k}(L)X_t$ is a linear combination of present and past variables which is fractionally integrated of a lower order than the individual variables are. This relationship will include both present and past values of the variables; since A_0 is the identity matrix, exactly one variable will enter with its present value.

True fractional cointegration occurs when there is some vector, a, such that $a'X_t$ is fractionally integrated of a lower order than any of the elements of X_t. Unlike the relationship we found for VARFI models, this relationship depends only on contemporaneous values of X_t. To produce true cointegration in a FIVAR model, we must include an additional linear filter in our description of the series (Sowell [1989a] uses this method as well). We motivate this addition through the simple bivariate fractional cointegration model of Robinson and Hualde [2003], Hualde and Robinson [2007]. Their model can be written as:

$$D(L)VX_t = \epsilon_t \tag{2.3}$$

where $V = \begin{pmatrix} 1 & -\nu \\ 0 & 1 \end{pmatrix}$; unlike them, we do not assume that $\epsilon_{kt} = 0$ when $t < 0$, because we consider only stationary cases. We may generalize the formulation in (2.3) by applying the V matrix to a FIVAR model, yielding a cointegrated FIVAR

model:

$$A(L)D(L)VX_t \;=\; \epsilon_t$$

where V is a matrix with ones along the diagonal and which is block diagonal according to which sets of series are cointegrated; see Sowell [1989a, section 5] for more details. In the case of a bivariate model, Sowell defines $V = \begin{pmatrix} 1 & 0 \\ \nu & 1 \end{pmatrix}$. In our analysis of cointegration in sections 2.5.4 and 2.7.4, any parameterization of the cointegrating matrix could be used. In our parameter estimation, we will use the parameterization of Sowell for identification. Then, the spectral density of multivariate cointegrated time series is given by:

$$f_{coint}(\nu) \;=\; \frac{1}{2\pi}V^{-1}D(e^{-i\nu})^{-1}A(e^{-i\nu})^{-1}\Sigma(A(e^{-i\nu})^{-1})^*(D(e^{-i\nu})^{-1})^*(V^{-1})^*$$

One could also introduce the V matrix into a VARFI model, though it would not generally lead to cointegration, because the series $A(L)^{-1}D(L)^{-1}\epsilon_t$ may all have the same order of integration even before the addition of V.

Thus far, we have assumed that all series have mean zero. In practice, it is likely that each time series will have an unknown mean. Consider the series $Y_t = X_t + \vec{\mu}$, where X_t follows one of the models with mean zero discussed above. A variety of possibilities exist for the estimation of the parameters of X_t and the estimation of μ. A common approach in the literature [for example Brockwell and Davis, 1993, page 238] is to subtract the sample mean from each time series $Y_{k\cdot}$, and to proceed with estimation based on the demeaned observations. However, the variance of the sample mean of a long memory process is $O(\frac{1}{n^{1-2d}})$, where d is the differencing parameter; thus, when $d > 0$, the variance declines more slowly than the traditional short memory variance of the mean, $O(\frac{1}{n})$. Despite these problems, demeaning is

straightforward, and we will use this method in our data analysis. An alternative method is to use restricted maximum likelihood (REML) as described by Harville [1977b]. In REML, the data is transformed to remove nuisance parameters, and then maximum likelihood is applied to the transformed data. In the case where the means of the individual time series are the only nuisance parameters, it is enough to take the first difference of each time series individually. Because differencing decreases each d_k by one, this method should be applied when we may assume that the original data has each $d_k \in (0.5, 1.5)$. One could also include the mean directly as part of maximum likelihood estimation. We will not pursue REML or inclusion of the mean further in this paper.

It is more common in the literature [for example Sowell, 1989a, Hosoya, 1996, Martin and Wilkins, 1999, **?**, Ravishanker and Ray, 1997, 2002] to analyze FIVAR models. Tsay [2007] is a notable exception. We will present algorithms for computing the covariances of FIVAR and VARFI models in sections 2.5.2 and 2.5.3, respectively. In section 2.7.1, we present an algorithm for approximating $|\Omega|$ which can be used with either FIVAR or VARFI models; section 2.7.3 contains a second algorithm which can be used only for VARFI models. The algorithms which we will present for computing the quadratic form, $X'\Omega^{-1}X$, and simulating from a multivariate time series apply to either of the models, because they depend only on knowing the covariance structure.

2.3 Block Circulant and Toeplitz Matrices

We begin by discussing some properties of circulant and Toeplitz matrices. These properties will be integral to many of the computational methods we will present. First, we recall the definitions of these two types of matrices. A Toeplitz matrix is one in which all of the elements along each diagonal are constant. That is, the

value of element A_{ij} depends only on $i-j$. The covariance matrix of a sequence of observations of a univariate time series, $x = (x_1, ..., x_T)'$, is a symmetric Toeplitz matrix. In general, Toeplitz matrices need not be symmetric. A circulant matrix is a matrix in which each row shifts the elements of the previous row one space to the right and moves the right-most element to the beginning of the row; a circulant matrix is a special case of a Toeplitz matrix.

Once we begin to consider multiple time series, we must use block matrices, that is, a matrices that can be partitioned into square blocks, each of which has a certain property. A block circulant matrix is a matrix which can be partitioned into blocks, each of which is a circulant; a block Toeplitz matrix is defined analogously. In this paper, we will generally consider $KT \times KT$ matrices which can be partitioned into K^2 Toeplitz or circulant blocks, each of which is of dimension $T \times T$.

In this section, we will present suggestions for the storage of block circulant and block Toeplitz matrices and algorithms for computing powers of block circulant matrices and for multiplying by block circulant and block Toeplitz matrices. Most of these algorithms are well-known; the algorithm for computing powers of circulant matrices is a new generalization of an algorithm presented by Chan and Olkin [1994] for computing inverses of block circulant matrices.

2.3.1 Efficient Storage

In this and the following sections, we discuss how we can use the properties of Toeplitz and circulant matrices to make the operations of the algorithms more efficient. In this section, as an introduction to the structure of these matrices, we discuss the simplest way: the repeated elements in each kind of matrix mean that there are more efficient ways to store them than just writing down all the elements.

The most obvious way to store a block circulant matrix would be to store all

K^2T^2 elements. However, because a circulant is completely defined by its first row, it is sufficient to store the first row of each block in a $K \times K \times T$ array, which is a dramatic reduction in the required storage space when T is large. In fact, one can store any T elements which uniquely define the first row of a circulant; we actually store the Fourier transform of the first row, as we will discuss in section 2.3.3.

In addition, it is not efficient to store the entire block Toeplitz matrix, Ω, since it would require the same large amount of space. Any Toeplitz matrix can be completely described by the first row and first column, and we store those elements instead of storing the entire matrix. In particular, we specify the Toeplitz matrix by the vector of elements:

$$[a(T-1,0),...,a(1,0),a(0,0),a(0,1),...,a(0,T-1)],$$

where we number the rows and columns starting at 0. When this Toeplitz matrix is the (i,j) block of Ω, we may describe the elements in relation to the covariances of $\{X_{it}\}$ and $\{X_{jt}\}$.. Note that, in block $A_{i,j}$, the $(0,r)$ element is $\omega_{ij}(-r) = \mathrm{Cov}(X_{i,t}, X_{j,r+t})$, and the $(r,0)$ element is $\omega_{ij}(r) = \mathrm{Cov}(X_{i,t}, X_{j,t-r})$. Thus, the elements of the first row and column as ordered above are simply

$$[a(T-1,0),...,a(1,0),a(0,0),a(0,1),...,a(0,T-1)] = [\omega_{ij}(T-1),...,\omega_{ij}(-(T-1))]$$

To describe a block Toeplitz matrix, we combine all of these vectors of length $2T-1$ into a three-dimensional array of size $K \times K \times (2T-1)$, in which each $K \times K$ layer is $\omega(r) = \mathrm{Cov}(X_t, X_{t-r})$ for $r = -(T-1),...,(T-1)$.

2.3.2 Computing a power of a block circulant matrix

As we will see, the methods for computing the quadratic form and for simulation both depend on computing a power of a block circulant matrix; the quadratic form requires computing an inverse, while simulation requires computing a square

root. In this section, we describe a fast way to compute an arbitrary power of a matrix, assuming that it is well-defined. The algorithm given in this section is a generalization of the one given by Chan and Olkin [1994], which describes only how to compute the inverse.

We first describe how the α^{th} power of a block circulant matrix, C, could be computed in theory. Let C_{ij} be the (i, j) block of C. The eigenvalue decomposition of that block is $C_{ij} = F^* \Lambda_{ij} F$, where F is the Fourier matrix with entries $F_{jk} = \frac{1}{\sqrt{T}} \exp\left(\frac{2\pi jk\sqrt{-1}}{T}\right)$ and Λ_{ij} is the diagonal matrix with diagonal equal to the Fourier transform of the first row of C_{ij} [see, for example, Brockwell and Davis, 1993, section 4.5]. Throughout this section, when we refer to the eigenvalues of a circulant, we order them as in the Fourier transform of the first row of C_j.

We now consider the matrix, C, as a whole. Let L be the $KT \times KT$ matrix consisting of the diagonal blocks, Λ_{ij}. We may write $C = (I \otimes F^*)L(I \otimes F)$, where $\otimes$ is the Kronecker product. Since $(I \otimes F)^* = (I \otimes F^*)^{-1}$, we may write $C^\alpha = (I \otimes F^*)L^\alpha(I \otimes F)$. Thus, it remains only find an expression for L^α.

Notice that L consists of K^2 blocks of size $T \times T$, each of which is zero except on the diagonal. Therefore, we may find a permutation matrix, P, such that $L = PBP'$, where B is a matrix with T blocks of size $K \times K$ along the diagonal and zeroes everywhere else. In particular, we choose P such that the t^{th} block along the diagonal of B consists of the t^{th} elements along the diagonal of each block in L; this moves all K^2T non-zero elements of L to the blocks along the diagonal of B. This is the same permutation matrix described in the introduction. The resulting blocks are not necessarily diagonal or Toeplitz. (See Chan and Olkin [1994, section three] for more details, particularly the graphic on page 94.)

Consider the spectral decomposition, $V_B \Lambda_B V_B^{-1}$, of B. Since B is block diagonal, we may choose V_B to be block diagonal as well. Combining this decomposition

with $(I \otimes F)$ and P yields the eigenvector decomposition of C:

$$C = (I \otimes F^*)PV_B\Lambda_B V_B^{-1}P^{-1}(I \otimes F^*)^{-1}$$

The spectral decomposition allows us to compute powers of C in a simple form. To do this, we first find B^α using the spectral decomposition for each block separately. (Though there no structure on the individual blocks in B, finding the eigenvalues is not computationally intensive if K is small.) We then find that $L^\alpha = PB^\alpha P'$. Since B^α is a block diagonal matrix and P is the same permutation matrix, L^α has the same diagonal block structure as L. Multiplying by $(I \otimes F^*)$ and $(I \otimes F)$, we find the formula for C^α:

$$C^\alpha = (I \otimes F^*)PB^\alpha P'(I \otimes F)$$

Not only does this give a method for computing C^α in theory, but it also shows that C^α is block circulant.

In a small number of cases, a block, B_r, of the matrix B might be defective, so that it has no spectral decomposition. While this means that general powers of B_r cannot be computed, algorithms exist for computing B_r^α for certain α. The inverse, $\alpha = -1$, can be computed using Gaussian elimination, as long as B_r is invertible. When $\alpha = \frac{1}{2}$, the algorithm of Denman and Beavers [1976] can be used to compute a square root. These are the two cases which will be required in this paper. When all of the B_r have spectral decompositions, the algorithm we have presented can be used for any α.

Though the formula above gives a straightforward method for describing C^α, it is not efficient to write down all K^2T^2 elements of C^α nor to multiply by permutation matrices. Instead, we create a $K \times K \times T$ array, Γ, to completely describe C in a way that makes computation simpler. First, we consider what is in each block, B_{rr}, of the block diagonal matrix, B. For a permutation matrix which moves the

r^{th} diagonal element of L_{ij} to the (i,j) location in the r^{th} block, the (i,j) element of B_{rr} is the r^{th} eigenvalue of C_{ij}. Thus, as we compute the eigenvalues for each block, C_{ij}, we may store them as $\Gamma(i,j,\cdot)$, so that each column of Γ corresponds to the eigenvalues of one block of C. Once Γ has been stored in this way, B_{rr} is simply $\Gamma(\cdot,\cdot,r)$. Define $\tilde{\Gamma}$ as the array that stores the elements of C^{α} in the same fashion. Then, since B is block diagonal, $\tilde{\Gamma}$ is obtained from Γ by computing the power of each layer, $\Gamma(\cdot,\cdot,r)$. This yields the following algorithm for obtaining the eigenvalues of the blocks of C^{α}:

Algorithm 2.1 Computing a Representation of a Power of a Block Circulant Matrix

- *Create two* $K \times K \times T$ *arrays,* Γ *and* $\tilde{\Gamma}$*, for storage.*
- *Loop over all pairs,* (i,j)*, with* $i = 1,...,K$ *and* $j = 1,...,K$*:*
  - *Set* $\Gamma(i,j,\cdot)$ *to the Fast Fourier Transform of the first row of* C_{ij}*.*
- *For* $r = 1,..,T$*, set* $\tilde{\Gamma}(\cdot,\cdot,r) = [\Gamma(\cdot,\cdot,r)]^{\alpha}$*.*

The resulting array holds the eigenvalues of the individual blocks of the power of the circulant preconditioner, which can be used for multiplication by C^{α} as shown in the next section.

2.3.3 Efficient Multiplication Methods

Multiplying a $T \times T$ matrix by a $T \times 1$ vector, v, requires $O(T^2)$ steps in general. If, however, the matrix, G, is a circulant, we can speed up this multiplication to $O(T \log T)$ steps, again using the fact that $G = F^{*}\Lambda F$. The following algorithm an be used for efficient multiplication by a circulant:

Algorithm 2.2 Multiplication by a Circulant, $G = F^{*}\Lambda F$.

- *Compute Fv as the Fourier transform of v.*
- *Compute ΛFv by multiplication by a diagonal matrix.*
- *Compute $F^*\Lambda Fv$ as the inverse Fourier transform of the previous result.*

Computing the Fourier transforms in the first and third steps takes $O(T \log T)$ operations, while the second step takes only $O(T)$ operations. In total, this multiplication takes $O(T \log T)$ time.

Algorithm 2.2 can also be used to compute Av, where A is a Toeplitz matrix and v in any vector. The extension to Toeplitz matrices requires circulant embedding. First, we create a $2T \times 2T$ circulant matrix, $\tilde{A}$, with diagonal blocks equal to A and off-diagonal blocks filled in with the elements of A necessary to make the matrix into a circulant. That is, if we number the row and column indices from 0 as before, the first row of $\tilde{A}$ is $[A(0,0), A(0,1), ..., A(0,T-1), A(0,0), A(T-1,0), ..., A(1,0)]$, and the circulant structure defines the remaining elements of $\tilde{A}$. Second, we extend v to a vector of length $2T$, $\tilde{v}$, by appending T zeroes to the end. We may then use Algorithm 2.2 to compute $\tilde{A}\tilde{v}$. Then, the first T elements of $\tilde{A}\tilde{v}$ are identical to the elements of Av.

Multiplication by circulant and Toeplitz matrices may be extended to multiplication by block circulant and block Toeplitz matrices. This takes advantage of the block-Toeplitz and block-circulant structures to reduce the number of operations required for multiplication to $O(K^2 T \log T)$ steps. Consider the general block matrix, B, with $T \times T$ blocks, B_{ij}, and vector, v, of length TK, partitioned into K subvectors, v_k, of length T. Then, we compute:

$$\begin{pmatrix} B_{11} & \cdots & B_{1K} \\ \vdots & \ddots & \vdots \\ B_{K1} & \cdots & B_{KK} \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_K \end{pmatrix} = \begin{pmatrix} B_{11}v_1 + \cdots + B_{1K}v_K \\ \vdots \\ B_{K1}v_1 + \cdots + B_{KK}v_K \end{pmatrix}$$

If the blocks of B are circulants, then each of the multiplications can be computed using the method for multiplying by a circulant. If the blocks of B are Toeplitz, then each of the multiplications can be computed using circulant-embedding. Computing K^2 such multiplications and then adding them up to get the final vector will take $O(K^2 T \log T)$ steps. We will see the usefulness of these multiplication methods in section 2.6.

2.4 Previous Computational Methods for Multivariate Models

2.4.1 Existing approximations to the likelihood of vector ARFIMA models

The most commonly used approximation to the likelihood in the frequency domain is the Whittle approximation first given in Whittle [1963]. The estimation is based on the periodogram matrix,

$$I(\lambda) = \frac{1}{2\pi T} \sum_{t=1}^{T} \sum_{s=1}^{T} X_t X_s' \exp(i\lambda(t-s))$$

According to Dunsmuir and Hannan [1976], the log likelihood is approximately a constant plus:

$$-\frac{T}{2} \log |\Sigma| - \frac{1}{2} \sum_{j=1}^{T} tr\left(f^{-1}\left(\frac{2\pi j}{T}\right) I\left(\frac{2\pi j}{T}\right)\right)$$

where $tr(\cdot)$ is the trace operator and f is the spectral density described in 2.2.2. Hosoya [1996] discusses this approximation in more detail. The first term uses the approximation $|\Omega| = T|\Sigma|$ of Grenander and Szego [1958], which Dunsmuir and Hannan [1976, page 344] note might not work well for small T even in the ARMA case, but which is very easy to compute. We discuss this approximation

and some possible modifications in more detail in section 2.7.1. In the univariate case, Hannan [1970, chapter 6, section 6] notes that the second term is based on the approximation $\Omega^{-1} \approx F\Lambda F^*$, where F is the Fourier matrix in section 2.3.2 and Λ is a diagonal matrix with $\frac{2\pi}{\sigma^2} f\left(\frac{2\pi j}{T}\right)$ at the (j,j) location. Notice that this approximates the Toeplitz matrix Ω by a circulant matrix. [See also Brockwell and Davis, 1993, Proposition 4.5.2.] Since $f(0)$ may be infinite in long memory models, this approximation may not be as accurate for vector ARFIMA models.

In the time domain, Luceno [1996] finds an approximation to the quadratic form in the likelihood expression; he neglects the determinant because he says (page 605) that importance declines in relation to the importance of the quadratic form for large T. As we will show in section 4.6, inclusion of an accurate approximation to the determinant can be quite important in the sample sizes we consider. He then finds an asymptotic approximation to Ω^{-1} in terms of "inverse-transpose" autocovariances, δ_i. These inverse-transpose autocovariances are defined by $\delta_i = \delta'_{-i} = \sum_{j=0}^{\infty} \pi'_{t+j} \Sigma^{-1} \pi_j$, for $i \geq 0$, where π_j are the $AR(\infty)$ coefficients and Σ is the innovation covariance for the process X_t. Using these inverse-transpose autocovariances, an exact expression for the quadratic form is given by:

$$X'\Omega^{-1}X \;=\; tr(\delta_0 P_0) + 2\sum_{i=1}^{\infty} tr(\delta_i P_i)$$

where he defines

$$P_i \;=\; \sum_{t=-\infty}^{\infty} \hat{X}_{t+i}\hat{X}'_t$$

and $\hat{X}_t$ is the observed series for $t = 1, ..., T$ and the forecast or backcast, that is, $E(X_t|X_1, ..., X_T)$, of the series otherwise. While this expression is exact, the forecasts and backcasts may be costly to compute, and the exact sum must be truncated for computational purposes. Therefore, Luceno recommends approxi-

mating:

$$P_i \approx \begin{cases} \sum_{t=1}^{T-i} X_{t+i} X_t' & 0 \leq i \leq T-1 \\ 0 & T \leq i \end{cases}$$
$$P_i = P_{-i}', i < 0$$

This approximation has an error which of the order $\frac{1}{T}$. Using the expression given above, the quadratic form can be computed by truncating the infinite sums. Luceno notes that the inverse-transpose autocovariances "frequently" are of the same model type as the original autocovariances (page 608); that is, the inverse-transpose autocovariances of a scalar ARFIMA model will be the autocovariances of a different ARFIMA model. However, he does not give a general method for computing the inverse-transpose autocovariance sequence, which makes them infeasible for the general case.

Martin and Wilkins [1999] avoid the likelihood functions altogether by applying indirect estimation to FIVAR models. In this approach, they estimate a $VAR(2)$ using the data and then find parameter values for a FIVAR model that lead to simulated data with identical estimates in a $VAR(2)$. We do not pursue this approach, though we note that indirect estimation would benefit from the simulation algorithm we propose in section 2.8.

2.4.2 Existing exact likelihood algorithms for vector ARFIMA models

The most comprehensive set of exact methods for maximum likelihood estimation for vector ARFIMA models can be found in two papers of Sowell [1989a,b]. The second paper presents algorithms of computing the autocovariances of a vector ARFIMA process of the FIVAR type, while the first paper presents methods for computing the inverse and determinant of a block Toeplitz matrix, which could be associated with any multivariate process.

First, we discuss Sowell's (1989a) algorithm for computing the autocovariances of a FIVAR process. Consider the autocovariances of a $FIVAR(p, \vec{d})$ process, where $\frac{B(L)}{a(L)}$ is the moving average representation of the vector $ARMA(p, q)$ part of the model, with $B(L)$ a matrix of lag polynomials of order at most $(K-1)p+q$ and $a(L)$ a scalar lag polynomial of order at most $H = Kp$. Let v_{ij} be the (i, j) entry of the cointegration matrix described in section 2.2.2. Sowell (1989a) finds that $\omega_{ij}(s) = \text{Cov}(X_{i,t}, X_{j,t-s})$ can be written as:

$$\omega_{ij}(s) = \sum_{l=-M}^{M} \sum_{m=1}^{H} \sum_{n=1}^{K} \sum_{r=1}^{K} v_{in} v_{jr} \psi_{ij}(l) \zeta_m C(d_i, d_j, H + l - s, \rho_m)$$

with

$$C(w, v, h, \rho) = \Gamma(1-w-v)\left(\rho^{2H} \sum_{m=0}^{\infty} \frac{\rho^m (-1)^{h+m}}{\Gamma(1-w+h+m)\Gamma(1-v-h-m)} + \sum_{n=1}^{\infty} \frac{\rho^n (-1)^{h-n}}{\Gamma(1-w+h-n)\Gamma(1-v-h+n)}\right)$$

and where the ρ_n, ζ_n, and $\psi_{ij}(l)$ satisfy:

$$\begin{aligned} a(\xi) &= \prod_{j=1}^{H} (1 - \rho_j \xi) \\ \zeta_j &= \frac{1}{\rho_j \prod_{i=1}^{H} (1 - \rho_i \rho_j) \prod_{m=1, m \neq j}^{H} (\rho_j - \rho_m)} \\ \psi_{ij}(l) &= \sum_{h=1}^{K} \sum_{t=1}^{K} \sum_{s=\max(0,l)}^{\min(M, M-l)} \Sigma_{ht} B_{ih}(s) B_{jt}(s-l) \end{aligned}$$

These sums must be evaluated using the hypergeometric function, which has no closed form in general [Weisstein, 2008]. While this gives an exact expression for the covariances, the sums are slow to evaluate, as we will show in section 4. Furthermore, Sowell's method does not apply to VARFI models.

An alternative method to compute the covariances of either a FIVAR or a VARFI model is to use the relationship between the spectral density and the

autocovariances. For any multivariate time series with cross-spectral density, f, we may compute the autocovariance function as:

$$\omega(h) = \int_{-\pi}^{\pi} e^{ih\lambda} f(\lambda) d\lambda$$

[See, for example, Brockwell and Davis, 1993, section 11.6.] This gives a straightforward method for computing the autocovariance sequence for either type of model. However, as we will show in Tables 2 and 4, it is also a computationally intensive method.

Sowell [1989b] describes methods to compute the determinants, inverses, and simulated realizations of any stationary multivariate process, including a vector ARFIMA process. In this paper, Sowell uses a version of the Durbin-Levinson algorithm [see also Brockwell and Davis, 1993, Proposition 11.4.1] to decompose the autocovariance matrix $\Omega = \mathrm{Var}(\tilde{X})$, where $\omega(j) = \mathrm{Cov}(X_t, X_{t-j})$, into a series of matrices that are useful for computation.

Algorithm 2.3 Sowell/Durbin-Levinson Covariance Matrix Decomposition [Sowell, 1989b]. *Set the initial values:*

$$\begin{aligned} v(0) &= \bar{v}(0) = \omega(0) \\ D(1) &= \omega(1) \\ \bar{D}(1) &= \omega(-1) \end{aligned}$$

For $n = 1, ..., T$ and $k = 1, ..., n$, compute the following quantities iteratively:

$$\begin{aligned}
A(n,n) &= D(n)\bar{v}(n-1)^{-1} \\
\bar{A}(n,n) &= \bar{D}(n)v(n-1)^{-1} \\
A(n,k) &= A(n-1,k) - A(n,n)\bar{A}(n-1,n-k) \\
\bar{A}(n,k) &= \bar{A}(n-1,k) - \bar{A}(n,n)A(n-1,n-k) \\
v(n) &= \omega(0) - \sum_{j=1}^{n} A(n,j)\omega(-j) \\
\bar{v}(n) &= \omega(0) - \sum_{j=1}^{n} \bar{A}(n,j)\omega(j) \\
D(n+1) &= \omega(n+1) - \sum_{j=1}^{n} A(n,n-j)\omega(j) \\
\bar{D}(n+1) &= \omega(-n-1) - \sum_{j=1}^{n} \bar{A}(n,n-j)\omega(-j)
\end{aligned}$$

Because all of the $A(n,k)$ must be computed, finding this decomposition requires $O(T^2)$ operations. General algorithms for determinants, inverses, and Cholesky decompositions for general matrices are $O(T^3)$, which means that using this algorithm is an improvement. However, an algorithm which is $O(T^2)$ is still quite slow for many applications. Given this decomposition, various quantities of interest become quite straightforward to compute. The determinant of Ω is simply

$\prod_{t=0}^{T-1} |v(t)|$. The inverse is given by $\Omega^{-1} = \bar{\beta}\bar{\beta}'$, where

$$\bar{\beta} = \begin{pmatrix} I_K & -\bar{A}(1,1)' & -\bar{A}(2,2)' & \cdots & -\bar{A}(T-1,T-1)' \\ 0 & I_K & -\bar{A}(2,1)' & \cdots & -\bar{A}(T-1,T-2)' \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \cdots & & 0 & I_K \end{pmatrix} \times \begin{pmatrix} \bar{v}(0) & 0 & 0 & \cdots & 0 \\ 0 & \bar{v}(1) & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & 0 & \bar{v}(T-1) \end{pmatrix}^{-1/2}$$

Notice that computation of the quadratic form, $X'\Omega^{-1}X$, using this representation would require an additional $O(T^2)$ steps, even if the decomposition were already known. Finally, Sowell points out (Result 3) that one can simulate from the distribution of X can be done by drawing a vector $U = (U_1', ..., U_T')'$ of length KT and then defining $X_1 = \bar{v}(0)^{1/2}U_1$ and $X_t = \sum_{j=1}^{t-1} \bar{A}(t-1, t-j)X_j + \bar{v}(t-1)^{1/2}U_t$. This simulation method also requires $O(T^2)$ steps, which would be particularly problematic if many samples were drawn. All of these methods are exact. However, the computations required are daunting when T is large. In fact, Doornik and Ooms [2003] say that this method is "still rather time consuming" for a dataset in which K is 2 and T is 121. In order for exact maximum likelihood to be feasible for estimating multivariate ARFIMA models, a faster algorithm is needed.

Tsay [2007] applied Sowell's algorithms to VARFI processes and using Sowell's (1989a) expression for the autocovariances of a $VARFI(0, \vec{d})$ process. While this work avoids the slow computations of autocovariances that plagues Sowell's (1989a) algorithm for computing the covariances of a FIVAR process, it does not address the slowness of the Cholesky decomposition.

Chung [2001] presents a method for calculating the impulse response function

of a FIVAR process. Also, Ravishanker and Ray [1997, 2002] discuss Bayesian methods for estimating from and forecasting FIVAR processes. We do not pursue either of these computations further.

2.5 Computing Autocovariances

In this section, we present algorithms for computing the autocovariances of both types of vector ARFIMA processes. First, as background, we describe the univariate splitting algorithm of Bertelli and Caporin [2002] for computing the autocovariances of a univariate ARFIMA model. In sections 2.5.2 and 2.5.3, we present fast algorithms for computing the autocovariance sequences of both FIVAR and VARFI processes. After we detail each algorithm, we will show the speed in practice and compare it to existing algorithms.

2.5.1 Computing the autocovariances of a univariate ARFIMA model

To compute the autocovariance sequence, $\omega(j)$, of an $ARFIMA(p,d,q)$ process with $d \in (-\frac{1}{2}, \frac{1}{2})$, Bertelli and Caporin [2002] write the covariances as the infinite convolution of the autocovariances, $\xi(j)$, of an $ARMA(p,q)$ process, and the autocovariances, $\phi(j)$, of an $ARFIMA(0,d,0)$. Both of these autocovariance sequences have closed forms or can be computed quickly. Then, the $ARFIMA(p,d,q)$ autocovariances can be written as :

$$\omega(j) = \sum_{h=-\infty}^{\infty} \xi(h)\phi(j-h)$$

Because the autocovariances of an ARMA model decay exponentially fast, they recommend setting the $\xi(h)$ to 0 for $|h| > M$ for large M. A larger value of M may be chosen to increase the accuracy. Then, the computation of these convolutions for $j = 0, ..., T$ can be done quickly using the Fast Fourier Transform. This gives a

fast and accurate method to compute the autocovariance sequence in the univariate case.

2.5.2 FIVAR Covariances

To generalize the univariate splitting algorithm to a FIVAR process, we use the two-step definition of a FIVAR discussed in section 2.2.2. In this section, for complete generality, we allow for a moving average component as well as an autoregressive component. First, we define Z_t as a vector ARMA process, so that $A(L)Z_t = B(L)\epsilon_t$, with $\text{Cov}(\epsilon_t) = \Sigma$. We assume that $A(L)$ and $B(L)$ both have all of their roots outside the unit circle. If this model were used for estimation, we would also require that $A(L)$ and $B(L)$ fit the identifiability conditions of Dunsmuir and Hannan [1976]. Let $\xi(h) = E(Z_{t+h}Z_t')$ be the autocovariance sequence of Z_t. The full model, $A(L)D(L)X_t = B(L)\epsilon_t$, can be written as $D(L)X_t = Z_t$. We may write $X_t = \sum_{j=0}^{\infty} C_j Z_{t-j}$, where C_j is a diagonal matrix with (k,k) element equal to $\psi(j, d_k) = \frac{\Gamma(j+d_k)}{\Gamma(j+1)\Gamma(d_k)}$ and Γ is the gamma function. If Z_t were white noise, this would be the moving average expansion of an $ARFIMA(0, d_k, 0)$ process. Using this "moving average" expansion, we find an expression for the autocovariances of $X_{\cdot t}$:

$$\begin{aligned}
\omega(h) &= \text{Cov}(X_t, X_{t-h}) && (2.4)\\
&= \text{Cov}\left(\sum_{i=0}^{\infty} C_i Z_{t-i}, \sum_{j=0}^{\infty} C_j Z_{t-j-h}\right) && (2.5)\\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} C_i \text{Cov}(Z_{t-i}, Z_{t-h-j}) C_j' && (2.6)\\
&= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} C_i \xi(h+j-i) C_j' && (2.7)
\end{aligned}$$

We now focus on the (k,l) entry of $C_i\xi(h+j-i)C_j'$. Let $\xi_{kl}(h)$ be the (k,l) entry of $\xi(h)$, that is, $\xi_{kl}(h) = E(Z_{k,t+h}Z_{l,t})$. Since C_i and C_j are both diagonal matrices,

the (k,l) entry of $C_i\xi(h+j-i)C_j'$ is $\psi(i,d_k)\psi(j,d_l)\xi_{kl}(h+j-i)$. Using this, we find an expression for the (k,l) entry of $\omega(h)$:

$$\begin{aligned}
\omega_{kl}(h) &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\psi(i,d_k)\psi(j,d_l)\xi_{kl}(h+j-i) && (2.8)\\
&= \sum_{m=0}^{\infty}\sum_{j=m}^{\infty}\psi(j-m,d_k)\psi(j,d_l)\xi_{kl}(h+m) && (2.9)\\
&= \sum_{m=0}^{\infty}\xi_{kl}(h+m)\left(\sum_{j=m}^{\infty}\psi(j,d_l)\psi(j-m,d_k)\right) && (2.10)
\end{aligned}$$

where the second equality follows from the substitution $m = j - i$ and an interchange of the order of summation. The inner sum is the cross-covariance of an $ARFIMA(0,d_k,0)$ process and an $ARFIMA(0,d_l,0)$ process that are driven by common white noise. Writing this cross-covariance in terms of the integral of the cross-spectrum, we find that:

$$\begin{aligned}
\sum_{j=m}^{\infty}\psi(j,d_l)\psi(j-m,d_k) &= \frac{1}{2\pi}\int_0^{2\pi}(1-e^{-i\lambda})^{-d_k}(1-e^{i\lambda})^{-d_l}e^{i\lambda m}d\lambda && (2.11)\\
&= \frac{\Gamma(1-d_k-d_l)(-1)^m}{\Gamma(1-d_k-m)\Gamma(1-d_l+m)} && (2.12)\\
&= \frac{\Gamma(1-d_k-d_l)\Gamma(d_k+m)}{\Gamma(d_k)\Gamma(1-d_k)\Gamma(1-d_l+m)} && (2.13)
\end{aligned}$$

where the last two equations follow from Sowell [1989a, Appendix II and Appendix III, equation IV.2]. Notice that this agrees with the usual expression for the autocovariance of an $ARFIMA(0,d_k,0)$ process when $d_k = d_l$ [see, for example, Brockwell and Davis, 1993, Theorem 13.2.1]. For notational convenience, we write $\phi_{lk}(h) = \frac{\Gamma(1-d_k-d_l)\Gamma(d_k+h)}{\Gamma(d_k)\Gamma(1-d_k)\Gamma(1-d_l+h)}$. Note that $\phi_{kl}(h) = \phi_{lk}(-h)$, as must be true for any cross-covariances.

Following Bertelli and Caporin [2002], we consider the finite approximation to the outer sum in (2.10), by setting $\xi_{kl}(m) = 0$ for all $|m| > M$. Because the autocovariance sequence of a vector ARMA decays exponentially fast, we may choose a

relatively small M to approximate the process to a given degree of accuracy. Our choice of M depends on the parameters of the ARMA process; if $\xi(h)$ is the autocovariance sequence of an $MA(q)$ process, then we may choose $M = q$ to compute the autocovariances exactly. Otherwise, we must choose an M which accounts for how quickly the autocovariances of the vector ARMA process decay.

Because any stationary and invertible vector $ARMA(p, q)$ process can be written as a vector $AR(1)$ [see Hamilton, 1994, page 259 for details], we focus on the $AR(1)$ process,

$$Z_t = A_1 Z_{t-1} + \eta_t$$

where A_1 is a $K \times K$ matrix such that all of its eigenvalues lie inside the unit circle. Notice that rewriting an ARMA process as an $AR(1)$ may lead to an innovation variance which is positive semi-definite but not positive definite. The computations presented in this section do not depend on Σ being positive definite, so this does not pose a problem. Then, the autocovariance sequence, $\xi(h)$, satisfies [Hamilton, 1994, page 265]:

$$\begin{aligned} vec(\xi(0)) &= (I_{K^2} - A_1 \otimes A_1)^{-1} vec(\Sigma) \\ \xi(h) &= A_1^h \xi(0), h > 0 \\ \xi(-h) &= \xi(h)' \end{aligned}$$

where h is a positive integer, vec is the vectorization operator, $\otimes$ is the Kronecker product, and I_{K^2} is a $K^2 \times K^2$ identity matrix. Let $G = \max_{k,l} \phi_{kl}(0)$. Let $\| \cdot \|$ be the Euclidean matrix norm, $\|Q\|_2$, where $\|Q\|_2$ is the maximum singular value of Q [see Heath, 2002, sections 3.6 and 4.7 for background]. This is equal to the square root of the largest eigenvalue of $Q^T Q$, which ensures that $\|A_1\| < 1$ as long as the $VAR(1)$ process defined by A_1 is stationary. Then, we may bound the norm

of the error in truncating the infinite sum by:

$$\begin{aligned} 2G\sum_{m=M}^{\infty}\|\xi_{kl}(m)\| &= 2G\sum_{m=M}^{\infty}\|A_1^m\xi(0)\| \\ &\leq 2G\sum_{m=M}^{\infty}\|A_1\|^m\|\xi(0)\| \\ &= 2G\|\xi(0)\|\frac{\|A_1\|^M}{1-\|A_1\|} \end{aligned}$$

Thus, once we know $\xi(0)$ and A_1, we can choose M such that the norm of the error does not exceed a chosen value, δ. In particular, we must have:

$$M \geq \frac{\log(1-\|A_1\|)+\log(\delta)-\log(G)}{\log(\|A_1\|)}+1$$

Once M has been chosen, it remains to compute the sequences $\phi_{kl}(h), h = -M-T, ..., M+T$ and $\xi_{kl}(h), h = -M, ..., M$ and their convolutions for each (k,l). Using the naive method of summing all the products directly would require $O(M^2+MT)$ operations. Instead, for larger values of M, we recommend using the Fast Fourier Transform to speed up the process to $O((M+T)\log(M+T))$ operations for each of the K^2 convolutions. In most cases, $M > \log T$; exceptions may occur when the eigenvalues of F are far from the unit circle or T is very large; we suggest checking this condition so that the faster convolution method is used. Since there are K^2 convolutions, using an efficient method is particularly important.

Combining all of these considerations yields the splitting algorithm for a FIVAR process:

Algorithm 2.4 Computing $FIVAR(1,\vec{d})$ covariances to tolerance δ.

- *Set G to be the maximum singular value of $\phi_{kl}(0)$ and compute the maximum singular value of A_1, $\|A_1\|$.*

- *Set M to be the smallest power of two greater than $\frac{\log(1-\|A_1\|)+\log(\delta)-\log(G)}{\log(\|A_1\|)}+1$.*
- *Compute the covariances, ξ, for a $VAR(1)$ for lags $-M$ to M.*
- *Compute the cross-covariances, ϕ, for ARFIMA processes with differencing parameters $\vec{d}$ for lags $-(M+T)$ to $M+T$.*
- *If $M \geq \log T$, compute the convolution of ξ_{ij} with ϕ_{ij} for $i = 1, ..., K$ and $j = 1, ..., K$ using the Fast Fourier Transform:*
  - *Append enough zeroes to ξ_{ij} and ϕ_{ij} so that the total length is the smallest power of two which is greater than $length(\xi_{ij}) + length(\phi_{ij})$. Call these series $\tilde{\xi}$ and $\tilde{\phi}$.*
  - *Compute the inverse Fast Fourier Transforms of $\tilde{\xi}$ and $\tilde{\phi}$ and multiply them together element-by-element.*
  - *Compute the Fast Fourier Transform of the result.*
  - *Return the first $(length(\xi_{ij}) + length(\phi_{ij}) - 1)$ elements of the result.*
  - *Extract the middle covariances from the result.*
- *If $M < \log T$, then compute the convolutions by summing all the terms directly.*

If the $VAR(1)$ process has been created from a vector $ARMA(p, q)$ process, the autocovariances of the original process are the autocovariances computed using the method above for the observed series.

Though we must truncate the sum, this method can be used to compute the autocovariances to any level of precision; more precision simply requires a larger choice of M. In Table 1, we give the computed values of some autocovariances

Lag	Splitting	Sowell
0	(3.658217, 6.04877, 6.048769, 35.02676)	(3.658217, 6.04877, 6.048769, 35.02676)
1	(3.103113, 5.530935, 6.094733, 33.952608)	(3.103113, 5.530935, 6.094733, 33.952608)
10	(0.7597274, 1.855598, 3.9196162, 25.501238)	(0.7597274, 1.855598, 3.9196162, 25.501238)
100	(0.06346564, 0.3674387, 1.12644985, 15.4985175)	(0.06346564, 0.3674387, 1.12644985, 15.4985175)

Table 1: Computed values for the autocovariances of a FIVAR process with $d = (0.1, 0.4)$, $A_1 = (0.7, 0.1, 0.2, 0.6)$, and $\Sigma = (1, .5, .5, 2)$, using the Sowell (1989a) method and the splitting method.

based on the splitting method and based on Sowell's (1989a) method. The two computed results are almost identical to at least five figures.

The running time of the FIVAR splitting algorithm depends on two factors. First, as the largest singular value of A_1 moves arbitrarily close to the unit circle, M will grow infinitely large. Second, given a fixed A_1 and therefore a fixed value of M, the running time will initially grow as $O(T \log T)$ as long as $M > \log T$, and will then grow linearly with T once it is faster to use direct summation instead of the Fast Fourier Transform. Notice that both the Sowell [1989a] method and the method using integrals described in section 2.4.2 grow linearly with T. Furthermore, Sowell's method depends on computing an infinite sum in which the summands decay as $\rho_1, ..., \rho_K$, which turn out to be the eigenvalues of A_1 in this case. Thus, both Sowell's method and our method slow down as the roots of $I - A_1 L$ approach the unit circle. In Tables 2 and 3, we report the total elapsed

T	Splitting	Sowell	Integral-Based Method
4	0.028	0.354	4.988
8	0.029	0.672	11.025
16	0.029	1.330	*
32	0.028	2.629	*
64	0.029	4.470	*
128	0.030	8.423	*

Table 2: Processing time needed to compute the autocovariances of a FIVAR process with $d = (0.1, 0.4)$, $A_1 = (0.7, 0.1, 0.2, 0.6)$, and $\Sigma = (1, .5, .5, 2)$, using the Sowell (1989a) method, the integral-based method and the splitting method presented in section 2.5.2. * indicates that the integral covariances for higher lags did not converge.

processor time in seconds as reported by the R `system.time` function to compute the autocovariances in various cases. This table shows that our method is much faster than either of the competing methods, for a range of T and for autoregressive matrices with singular values both near and far from the unit circle.

Now that we have seen that the splitting algorithm yields the same results as Sowell's algorithm in a fraction of the time, we will use only the splitting algorithm to compute covariances in the remainder of this paper.

2.5.3 VARFI Covariances

To use the splitting algorithm with a VARFI process, we first consider the spectral density of X_t. We begin with a $VARFI(1, \vec{d})$ in which $A(L) = I - A_1 L$, where $A(L)$ has all of its roots outside of the unit circle. In this case, we also assume that A_1 is not a defective matrix, so that it has K unique eigenvectors [see, for

Maximum Singular Value	Our Time	Sowell Time	Maximum Difference	M Required
0.8	0.028	4.552	9.808×10^{-10}	119
0.9	0.038	5.970	7.827×10^{-10}	263
0.95	0.058	8.154	7.504×10^{-10}	564
0.99	0.246	14.733	5.955×10^{-8}	3138
0.995	0.577	14.848	1.413×10^{-5}	6479
0.999	4.701	14.775	0.001285	34324

Table 3: Processing time needed to compute the autocovariances of a FIVAR process with $T = 64$, $d = (0.1, 0.4)$, $\Sigma = (1, .5, .5, 2)$, and $A_1 = \alpha(0.7, 0.1, 0.2, 0.6)$, where α is a scalar chosen to vary the maximum singular value. The fourth column shows the maximum absolute difference between Sowell's (1989a) method and our method over all 64 autocovariances. The last column shows the value of M required by the splitting algorithm. Times are the mean processing time needed for 100 repetitions of the calculation.

example Heath, 2002, chapter 4]. Though this requirement will cause the method not to apply for certain matrices, defective matrices are quite rare and therefore of little concern.

We first write the autocovariances of X_t in terms of the spectral density:

$$
\begin{aligned}
f_X(\lambda) &= (I - A_1 e^{-i\lambda})^{-1} D(e^{-i\lambda})^{-1} \Sigma D(e^{i\lambda})^{-1} (I - A_1^* e^{i\lambda})^{-1} \\
&= \left(\sum_{r=0}^{\infty} A_1^r e^{-i\lambda r} \right) \times \\
&\quad \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1K} \\ \Sigma_{21} & \Sigma_{22} & \ddots & \Sigma_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{K1} & \Sigma_{K2} & \cdots & \Sigma_{KK} \end{pmatrix} \bullet \\
&\quad \begin{pmatrix} (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_1} & (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_2} & \cdots & (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_K} \\ (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_2} & (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{-d_2} & \ddots & (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{-d_K} \\ \vdots & \vdots & \ddots & \vdots \\ (1-e^{-i\lambda})^{-d_K}(1-e^{i\lambda})^{-d_1} & (1-e^{-i\lambda})^{-d_K}(1-e^{i\lambda})^{-d_2} & \cdots & (1-e^{-i\lambda})^{-d_K}(1-e^{i\lambda})^{-d_K} \end{pmatrix} \\
&\quad \times \left(\sum_{s=0}^{\infty} (A_1^*)^s e^{i\lambda s} \right) \\
&= \sum_{r=0}^{\infty} \sum_{s=0}^{\infty} A_1^r \times \\
&\quad \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1K} \\ \Sigma_{21} & \Sigma_{22} & \ddots & \Sigma_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{K1} & \Sigma_{K2} & \cdots & \Sigma_{KK} \end{pmatrix} \bullet \\
&\quad \begin{pmatrix} (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_1} & (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_2} & \cdots & (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_K} \\ (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{-d_2} & (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{-d_2} & \ddots & (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{-d_K} \\ \vdots & \vdots & \ddots & \vdots \\ (1-e^{-i\lambda})^{-d_K}(1-e^{i\lambda})^{-d_1} & (1-e^{-i\lambda})^{-d_K}(1-e^{i\lambda})^{-d_2} & \cdots & (1-e^{-i\lambda})^{-d_K}(1-e^{i\lambda})^{-d_K} \end{pmatrix} \\
&\quad \times (A_1^*)^s e^{i\lambda(s-r)}
\end{aligned}
$$

where $\bullet$ denotes the Hadamard (element-wise) matrix product. Let $A_1 = V_A \Lambda V_A^{-1}$

be an eigenvalue decomposition of A_1. For notational convenience, define:

$$
\begin{aligned}
Q(\lambda) &= V_A^{-1} \begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1K} \\ \Sigma_{21} & \Sigma_{22} & \ddots & \Sigma_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{K1} & \Sigma_{K2} & \cdots & \Sigma_{KK} \end{pmatrix} \bullet \\
&\begin{pmatrix} (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_1} & (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_2} & \cdots & (1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_K} \\ (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{-d_2} & (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{-d_2} & \ddots & (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{-d_K} \\ \vdots & \vdots & \ddots & \vdots \\ (1-e^{-i\lambda})^{-d_K}(1-e^{i\lambda})^{-d_1} & (1-e^{-i\lambda})^{-d_K}(1-e^{i\lambda})^{-d_2} & \cdots & (1-e^{-i\lambda})^{-d_K}(1-e^{i\lambda})^{-d_K} \end{pmatrix} \\
&\times (V_A^*)^{-1}
\end{aligned}
$$

Using this notation, we describe the autocovariances of X_t:

$$
\begin{aligned}
\omega(h) &= \int_{-\pi}^{\pi} f_X(\lambda) e^{ih\lambda} d\lambda \\
&= \int_{-\pi}^{\pi} \sum_{r=0}^{\infty} \sum_{s=0}^{\infty} V_A \Lambda^r Q(\lambda) (\Lambda^*)^s V_A^* e^{-i\lambda(r-s)} e^{ih\lambda} d\lambda \\
&= V_A \left(\sum_{r=0}^{\infty} \sum_{s=0}^{\infty} \Lambda^r \left(\int_{-\pi}^{\pi} Q(\lambda) e^{-i\lambda(r-s-h)} d\lambda \right) (\Lambda^*)^s \right) V_A^*
\end{aligned}
$$

Notice that $\int_{-\pi}^{\pi} Q(\lambda) e^{-i\lambda(r-s-h)} d\lambda$ is $V_A^{-1}(\Sigma \bullet \phi(r-s-h))(V_A^*)^{-1}$, where $(\Sigma \bullet \phi(r-s-h))$ is the r^{th} autocovariance of a $VARFI(0, \vec{d})$; this can be computed using the expression in equation (2.13) above. Let $H_{ij}(r)$ be the (i,j) element of $V_A^{-1}\phi(r-s-h)(V_A^*)^{-1}$. Let Λ_{ii} be the (i,i) element of Λ. Then, the (i,j) element of the inner sum is:

$$
\sum_{r=0}^{\infty} \sum_{s=0}^{\infty} \Lambda_{ii}^r \bar{\Lambda}_{jj}^s H_{ij}(h+s-r) = \sum_{u=-\infty}^{\infty} L_{ij}(u) H_{ij}(h-u) \tag{2.14}
$$

where $\bar{\Lambda}_{jj}$ is the complex conjugate of Λ_{jj} and

$$
L_{ij}(u) = \begin{cases} \frac{\Lambda_{ii}^u}{1-\Lambda_{ii}\bar{\Lambda}_{jj}} & u \geq 0 \\ \frac{\bar{\Lambda}_{jj}^{|u|}}{1-\Lambda_{ii}\Lambda_{jj}} & u < 0 \end{cases}
$$

After the sums in (2.14) have been calculated for each lag and each $i = 1, ..., K$ and $j = 1, ..., K$, the matrix of sums for each u must be multiplied by V_A and V_A^* to find the covariances of the original process.

As before, we want to approximate the sums above by sums with a finite number of terms. Since each $L_{ij}(u)$ decays exponentially quickly, we again choose M so that $\sum_{u=M+1}^{\infty} L_{ij}(u) < \delta$ for a given tolerance δ and all i, j. Let $G = \max V_A^{-1}\phi(0)(V_A^{-1})^*$, where the maximum is taken over all of the entries in the product. Let $|\Lambda_{j^*j^*}|$ be the absolute value of the largest eigenvalue. Then, $L_{ij}(u) \leq \frac{\Lambda^u_{j^*j^*}}{1-|\Lambda_{j^*j^*}|^2}$, and we may bound the sum of the omitted terms:

$$\begin{aligned}\sum_{u=M+1}^{\infty} H(h-u)L_{ij}(u) &\leq G\sum_{u=M+1}^{\infty}\frac{\Lambda^u_{j^*j^*}}{1-|\Lambda_{j^*j^*}|^2} \\ &= G\frac{\Lambda^{M+1}_{j^*j^*}}{(1-|\Lambda_{j^*j^*}|^2)(1-|\Lambda_{j^*j^*}|)}\end{aligned}$$

Thus, we may choose $M > \frac{\log\delta+2\log(1-|\Lambda_{j^*j^*}|)+\log(1+|\Lambda_{j^*j^*}|)}{|\log\Lambda_{j^*j^*}|-\log G}$ to ensure that the sum of the omitted terms is less than δ. As in the computation of the autocovariances of FIVAR processes, we suggest using the fast Fourier transform to compute the convolutions in the case where $M > \log T$. This yields the following algorithm:

Algorithm 2.5 Computing the Covariances of a VARFI process to tolerance, δ.

- *Compute the eigenvalue decomposition,* $A_1 = V_A\Lambda V_A^{-1}$ *and find* j^* *such that* $\Lambda_{j^*j^*}$ *is the largest eigenvalue.*
- *Set* G *to be the maximum entry of* $V_A^{-1}\phi(0)(V_A^{-1})^*$.
- *Set* M *to be the smallest power of two greater than* $\frac{\log\delta+2\log(1-\Lambda_{j^*j^*})+\log(1+\Lambda_{j^*j^*})}{\log\Lambda_{j^*j^*}-\log G}$.
- *For* $i = 1, ..., K$, $j = 1, ..., K$, *and* $u = -M, ..., M$, *compute* $L_{ij}(u)$ *using equation (2.15).*

- *Compute the cross-covariances, $\phi(r)$, for ARFIMA processes with differencing parameters $\vec{d}$ from lags $r = -(M+T), ..., M+T$.*

- *For $r = -(M+T), ..., M+T$, compute $H(r) = V_A^{-1}\phi(r)(V_A^{-1})^*$.*

- *If $M \geq \log T$, compute the convolution of L_{ij} with H_{ij} for $i = 1, ..., K$ and $j = 1, ..., K$ using the Fast Fourier Transform:*

  - *Append enough zeroes to L_{ij} and H_{ij} so that the total length is the smallest power of two which is greater than $(length(L_{ij}) + length(H_{ij}))$. Call these series $\tilde{L}$ and $\tilde{H}$.*

  - *Compute the inverse Fast Fourier Transforms of $\tilde{L}$ and $\tilde{H}$ and multiply them together element-by-element.*

  - *Compute the Fast Fourier Transform of the result.*

  - *Return the first $(length(L_{ij}) + length(H_{ij}) - 1)$ elements of the result.*

  - *Extract the middle covariances, from $-T$ to T, from the result.*

- *If $M < \log T$, then compute the convolutions by summing all the terms directly.*

- *Pre-multiply the matrix for each lag by V_A and post-multiply the matrix for each lag by V_A^*.*

Like the algorithm for FIVAR covariances, this algorithm runs in $O(\min(M^2 + MT, (M+T)\log(M+T))$. In Table 4, we compare the processing time needed for this method to the time needed to use the integral definition of the autocovariance sequence. As in the FIVAR case, using the integral definition of the covariances requires dramatically more computing time, despite the fact that it is $O(T)$.

One disadvantage to this computational method for VARFI covariances is that V_A must be inverted. While this is generally fast for small K, it makes the computed covariances sensitive to the condition number of V_A. In particular, we have found that when V_A is close to singular, many of the computed covariances are zero, even though their exact values are nowhere near 0. This can occur when A_1 differs by a minute amount from a multiple of the identity matrix. This consideration should inform the choice of initial values in VARFI maximum likelihood estimation.

To extend this method for computing covariances to a $VARFI(p, \vec{d})$ model, we rewrite that model as a $VARFI(1, \vec{d^{\#}})$ model. Suppose $A(L) = I - A_1 L - ... - A_p L^p$. Let $X_t^{\#} = vec(X_t, ..., X_{t-p+1})$, and

$$A_1^{\#} = \begin{pmatrix} A_1 & A_2 & ... & A_p \\ I_K & 0 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

$$d^{\#} = \begin{pmatrix} d \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\Sigma^{\#} = \begin{pmatrix} \Sigma & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Then, $D^{\#}(L)(I - A_1^{\#} L)X_t^{\#} = \epsilon_t$ is a $VARFI(1, \vec{d^{\#}}, 0)$ process, and the first K series follow the original $VARFI(p, \vec{d}, 0)$ model. As before $\Sigma^{\#}$ is not generally positive definite, but this does not pose a problem for computing autocovariances.

T	Splitting	Integral-Based Method
4	0.031	8.362
8	0.031	18.025
16	0.032	37.511
32	0.035	85.087
64	0.040	216.706
128	0.050	623.426

Table 4: Time needed to compute the autocovariances of a VARFI process with $d = (0.1, 0.4)$, $A_1 = (0.7, 0.1, 0.2, 0.6)$, and $\Sigma = (1, .5, .5, 2)$ using the integral definition and using the splitting algorithm presented in section 2.5.3. Times are the mean processing time needed over 100 repetitions of the calculation.

Thus, the method presented above generalizes to any $VARFI(p, \vec{d}, 0)$ model with finite p, where we extract the relevant autocovariances as we did in section 2.5.2. We do not extend this method to models with moving average components.

2.5.4 Cointegrated Systems

Consider the cointegrated FIVAR model, $A(L)D(L)VX_t = \epsilon_t$. Define the process, Y_t, by $A(L)D(L)Y_t = \epsilon_t$. Then,

$$\begin{aligned} \text{Cov}(X_t, X_{t-j}) &= \text{Cov}(V^{-1}Y_t, V^{-1}Y_{t-j}) \\ &= V^{-1}\text{Cov}(Y_t, Y_{t-j})(V^{-1})' \end{aligned}$$

Since Y_t is a FIVAR process, its autocovariance sequence can be computed using Algorithm 2.4 above. Then, we may compute the autocovariances of X_t by multiplying each autocovariance by V^{-1} and $(V^{-1})'$, which takes $O(T)$ additional steps.

2.6 Computing the Quadratic Form

To compute the quadratic form, $X\Omega^{-1}X$, in the expression for the likelihood in equation (2.1), we apply the preconditioned conjugate gradient algorithm. The application of preconditioned conjugate gradient algorithms to univariate long memory time series began with Deo et al. [2006]; related theoretical results are available in Chen et al. [2006]. The algorithm which we present in this section was developed by Chan and Olkin [1994], but this is its first application to multivariate long memory time series.

We begin this section with some background on the PCG algorithm. We then describe how we can apply it most efficiently to multivariate time series, and finally discuss the computational cost of these methods.

2.6.1 The Preconditioned Conjugate Gradient Algorithm

Preconditioned conjugate gradient methods have been used extensively in solving systems of linear equations of the form $\Omega y = b$ where Ω is symmetric and positive definite (in this section, we rely heavily on Shewchuk [1994]; see his write-up for more details). The conjugate gradient method and the preconditioned conjugate gradient method are based on using the residual error at at each iteration to choose a search direction and the optimal distance in that direction. These methods can be applied to any system in which Ω is symmetric and positive definite.

Algorithm 2.6 Conjugate Gradient Algorithm [Shewchuk, 1994, see, for example,]. *Let a tolerance, δ, be given. Let the initial value, $y_{(0)}$, be a vector of zeroes.*

Initialize:

$$\begin{aligned} d_{(0)} &= b - \Omega y_{(0)} \\ r_{(0)} &= b - \Omega y_{(0)} \end{aligned}$$

Iterate through the following steps until $\|r_{(i)}\| < \delta$.

$$\begin{aligned} \alpha_{(i)} &= \frac{r'_{(i)} r_{(i)}}{d'_{(i)} \Omega d_{(i)}} \\ y_{(i+1)} &= y_{(i)} + \alpha_{(i)} d_{(i)} \\ r_{(i+1)} &= r_{(i)} - \alpha_{(i)} \Omega d_{(i)} \\ \beta_{(i+1)} &= \frac{r'_{(i+1)} r_{(i+1)}}{r'_{(i)} r_{(i)}} \\ d_{(i+1)} &= r_{(i+1)} + \beta_{(i+1)} d_{(i)} \end{aligned}$$

This algorithm chooses a direction, $d_{(i)}$, which is conjugate, or Ω-orthogonal, to all the previous search directions; that is, $d'_{(i)} \Omega d_{(j)} = 0$ when $i \neq j$. The choice of direction is based on Gram-Schmidt conjugation. Each search direction, $d_{(i)}$, is linearly independent, and the distance chosen for each search direction, $\alpha_{(i)}$, is optimal. That is, the resulting residual is conjugate to the search direction used to compute it. Because of this, if there were no roundoff error, each search direction would be linearly independent and used at most once. Thus, with infinite precision, the algorithm would always converge to exactly the true solution in a number of steps at most the dimension of Ω.

Because computers have finite precision, we say that the algorithm has converged when the distance from the computed answer to the true value is less than some tolerance level. In this algorithm, the search directions with the largest steps are used first, so convergence in this sense takes fewer steps than would be required with infinite precision. This is an important property when the dimension of Ω is

large. To be precise, the error in the i^{th} iteration, $e_{(i)} = y_{(i)} - y$, satisfies [Shewchuk, 1994, page 36]:

$$\sqrt{e'_{(i)}\Omega e_{(i)}} \leq 2\left(\frac{\sqrt{\kappa(\Omega)}-1}{\sqrt{\kappa(\Omega)}+1}\right)^i e'_{(0)}\Omega e_{(0)} = 2\left(1-\frac{2}{\sqrt{\kappa(\Omega)}+1}\right)^i e'_{(0)}\Omega e_{(0)}$$

where $\kappa(\Omega)$ is the condition number of the matrix Ω, defined as the ratio of the largest to the smallest eigenvalue of Ω. Given any tolerance level and initial error, we can solve for i to find an approximate number of iterations required for convergence within that tolerance level. Such an analysis shows that the required number of iterations is $O(\sqrt{\kappa(\Omega)})$. This shows the importance of the condition number of Ω to the computational complexity of this algorithm.

When the condition number is large, we can "precondition" the matrix in order to reduce the condition number. This is based on solving the system of linear equations, $C^{-1}\Omega y = C^{-1}b$, where C approximates Ω but has an inverse which is easy to compute. This method is effective when $\kappa(C^{-1}\Omega) << \kappa(\Omega)$. However, one does not simply apply the conjugate gradient method to the system $C^{-1}\Omega y = C^{-1}b$, since the product $C^{-1}\Omega$ is not generally symmetric or positive definite. Instead, consider the matrix E such that $EE' = C$. Then, $\kappa(C^{-1}\Omega) = \kappa(E^{-1}\Omega(E^{-1})')$, and the latter matrix is symmetric and positive definite. Thus, we could solve the system $E^{-1}\Omega(E^{-1})'\hat{y} = E^{-1}b$ for $\hat{y}$, and then compute $y = (E^{-1})'\hat{y}$; this is called the transformed preconditioned conjugate gradient algorithm.

Using this version of the algorithm would require computing E. Instead, we define $\hat{r}_{(i)} = E^{-1}r_{(i)}$ and $\hat{d}_{(i)} = E'd_{(i)}$. We can substitute these into the conjugate gradient algorithm above to arrive at the untransformed preconditioned conjugate gradient (PCG) algorithm.

Algorithm 2.7 Preconditioned Conjugate Gradient Algorithm [Shewchuk, 1994, see, for example,]. *Let a tolerance, δ, be given. Let $x_{(0)}$ be a vector of zeroes.*

Initialize:

$$\begin{aligned} r_{(0)} &= b - \Omega x_{(0)} \\ d_{(0)} &= C^{-1} r_{(0)} \end{aligned}$$

Iterate through the following steps until $\|r_{(i)}\| < \delta$.

$$\begin{aligned} \alpha_{(i)} &= \frac{r'_{(i)} C^{-1} r_{(i)}}{d'_{(i)} \Omega d_{(i)}} \\ x_{(i+1)} &= x_{(i)} + \alpha_{(i)} d_{(i)} \\ r_{(i+1)} &= r_{(i)} - \alpha_{(i)} \Omega d_{(i)} \\ \beta_{(i+1)} &= \frac{r'_{(i+1)} C^{-1} r_{(i+1)}}{r'_{(i)} C^{-1} r_{(i)}} \\ d_{(i+1)} &= C^{-1} r_{(i+1)} + \beta_{(i+1)} d_{(i)} \end{aligned}$$

Note that this algorithm requires multiplying vectors by C, which can be one of the more computationally intensive steps of the process. Therefore, we choose C to make the multiplication more tractable. In the case of multivariate time series with $K << T$, we choose C to be block circulant. This choice allows us to take advantage of all the computational methods designed for circulants described in section 2.3. For an extensive review of the PCG algorithm for Toeplitz and block-Toeplitz matrices, see Chan and Ng [1996].

2.6.2 The Choice of Preconditioner

A good preconditioner, C, must approximate Ω. In addition, its inverse must lend itself to efficient multiplication. We choose to use the "level 1" preconditioners of Chan and Olkin [1994]. In this section, we describe the preconditioner and how to compute it.

We begin by writing our block Toeplitz matrix, Ω, in terms of its blocks:

$$\Omega = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1K} \\ A_{21} & A_{22} & \cdots & A_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ A_{K1} & A_{K2} & \cdots & A_{KK} \end{pmatrix}$$

Define $A_{ij}(r)$, for $r = -(T-1), ..., (T-1)$, to be the element along the r^{th} subdiagonal away from the main diagonal, where a negative r corresponds to diagonals in the lower triangle of the matrix and a positive r corresponds to diagonals in the upper triangle of the matrix. That is, we write:

$$A_{ij} = \begin{pmatrix} A_{ij}(0) & A_{ij}(1) & \cdots & A_{ij}(T-1) \\ A_{ij}(-1) & A_{ij}(0) & \cdots & A_{ij}(T-2) \\ \vdots & \vdots & \ddots & \vdots \\ A_{ij}(-(T-1)) & A_{ij}(-(T-2)) & \cdots & A_{ij}(0) \end{pmatrix},$$

As mentioned in the section 2.3.1, $A_{ij}(r) = \omega(-r)$; this allows us to relate the elements of this matrix to the properties of the underlying time series. We approximate Ω by approximating its individual blocks. The approximation we use here is T. Chan's (1988) optimal circulant preconditioner, $circ(A_{ij})$. This preconditioner is the circulant matrix with first row consisting of entries $c_0 = A_{ij}(0)$ and $c_r = \frac{rA_{ij}(-(T-r))+(T-r)A_{ij}(r)}{T}$, $r = 1, ..., T-1$. Combining the preconditioners for all of the blocks yields the following block circulant matrix:

$$C = \begin{pmatrix} circ(A_{11}) & circ(A_{12}) & \cdots & circ(A_{1K}) \\ circ(A_{21}) & circ(A_{22}) & \cdots & circ(A_{2K}) \\ \vdots & \vdots & \ddots & \vdots \\ circ(A_{K1}) & circ(A_{K2}) & \cdots & circ(A_{KK}) \end{pmatrix}.$$

With this theoretical preconditioner, we can now apply the methods we discussed in section 2.3. First, we store only the first row of each block of the pre-

conditioner. Second, we find a representation for C^{-1} using the inversion method for block circulant matrices described in Algorithm 2.1. Finally, we multiply by Ω and by C^{-1} using the fast multiplication methods discussed in 2.3.3.

2.6.3 Computational Cost

Chan and Olkin [1994] show that the algorithm described in the previous two sections has a set-up cost of $O(K^2T\log T + K^3T)$ to compute the preconditioner and a cost of $O(K^2T + KT\log T)$ per iteration. In most multivariate time series applications, K is generally fixed and much smaller than $\log T$, so the relevant costs are $O(K^2T\log T)$ and $O(KT\log T)$.

As we mentioned in 2.6.1, the number of iterations required for convergence depends on the condition number of the matrix. By preconditioning, we hope to reduce that ratio so that convergence is faster. In the case of a covariance matrix based on a univariate long memory model, Chen et al. [2006] show that the condition number of the preconditioned matrix grows as $O(\log^3 T)$, which implies that the overall algorithm with $K = 1$ runs in $O(T\log^{5/2} T)$ time. Chan and Olkin [1994] run numerical experiments in which the r^{th} diagonal (for $r = -(T-1), ..., T-1$) of the j^{th} block has element $\frac{1}{(j+1)^{1.1}+(|r|+1)^{1.1}}$ or $\frac{1}{(j+1)^{2.1}+(|r|+1)^{2.1}}$. In their experiment, they find that their preconditioner dramatically reduces the number of iterations, but sometimes increases the number of operations because of the additional multiplications.

In Tables 5 and 6, we report the condition number, before and after preconditioning, for the covariance matrices associated with FIVAR and VARFI processes. Preconditioning dramatically reduces the condition number in both cases. A simple regression of $\log(\kappa(C^{-1}\Omega))$ on $\log(\log(T))$ using the data in those tables produces slope estimates of 1.238 (standard error 0.0492) and 1.259 (standard error

T	$\kappa(\Omega)$	$\kappa(C^{-1}\Omega)$	$\log(\kappa(\Omega))$	$\log(\kappa(C^{-1}\Omega))$
4	782.7286	11.5169	6.6628	2.4438
8	1749.7115	22.7916	7.4672	3.1264
16	3322.2824	32.5125	8.1084	3.4816
32	5952.1906	38.2324	8.6915	3.6437
64	10454.6722	42.2234	9.2548	3.7430
128	18250.7736	56.7711	9.8120	4.0390
256	31801.4260	71.3439	10.3673	4.2675
512	55382.3246	83.8753	10.9220	4.4293

Table 5: Condition number of autocovariance matrices for a $FIVAR(1, \vec{d})$ process with parameters $d = (0.1, 0.4)$, $\Sigma = (1, 0.5, 0.5, 2)$, and $A_1 = (0.6, -0.1, 0.2, 0.8)$, for a range of T.

0.0404) for FIVAR and VARFI processes, respectively. We also plot $\log(\kappa(C^{-1}\Omega))$ versus $\log(\log(T))$ in Figures 4 and 5; these plots show that the relationship is approximately linear. Based on the slope estimate and the linear relationship in the plots, the conditioned number of the preconditioned matrix appears to grow approximately as $O(\log^{5/4} T)$.

In the case of cointegrated FIVAR series, we may bound the condition number of Ω in terms of the cointegrating matrix and the properties of the underlying FIVAR series. Let Ω_0 be the covariance matrix of the series before they are cointegrated; this is the covariance matrix associated with the FIVAR process, Y_t, described in section 2.5.4. Let V be the cointegrating matrix as before. Then,

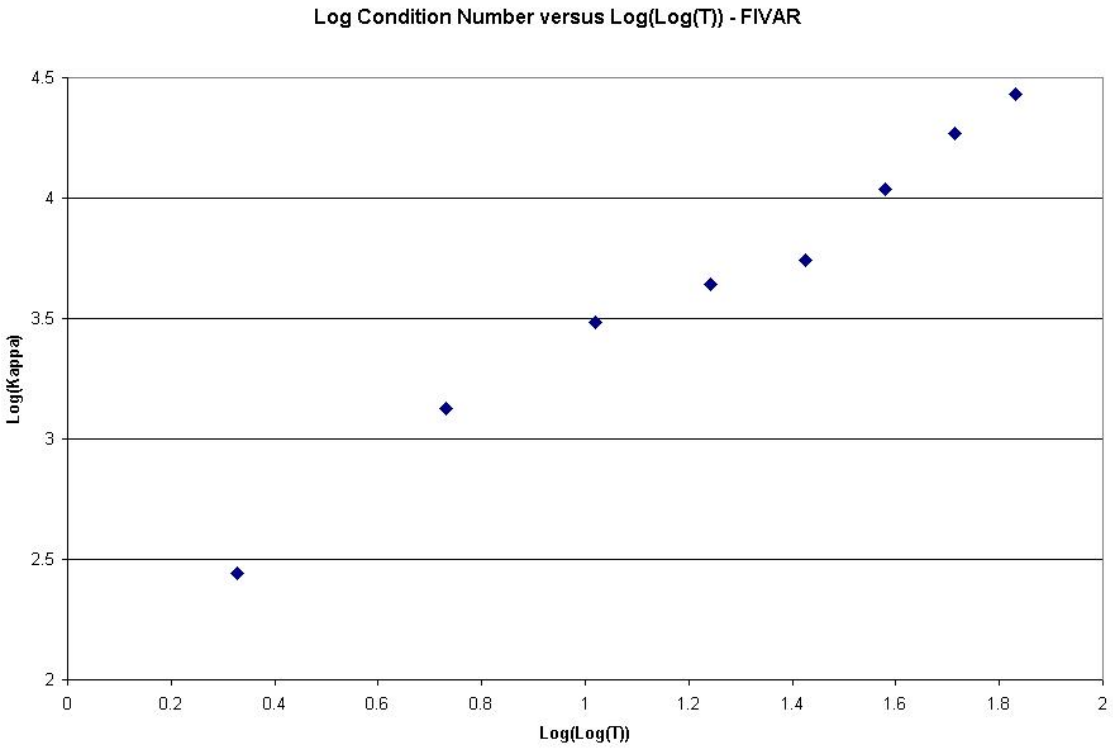

Figure 4: Plot of the logged condition number versus $\log(\log(T))$ for a $FIVAR(1, \vec{d})$ process with parameters $d = (0.1, 0.4)$, $\Sigma = (1, 0.5, 0.5, 2)$, and $A_1 = (0.6, -0.1, 0.2, 0.8)$.

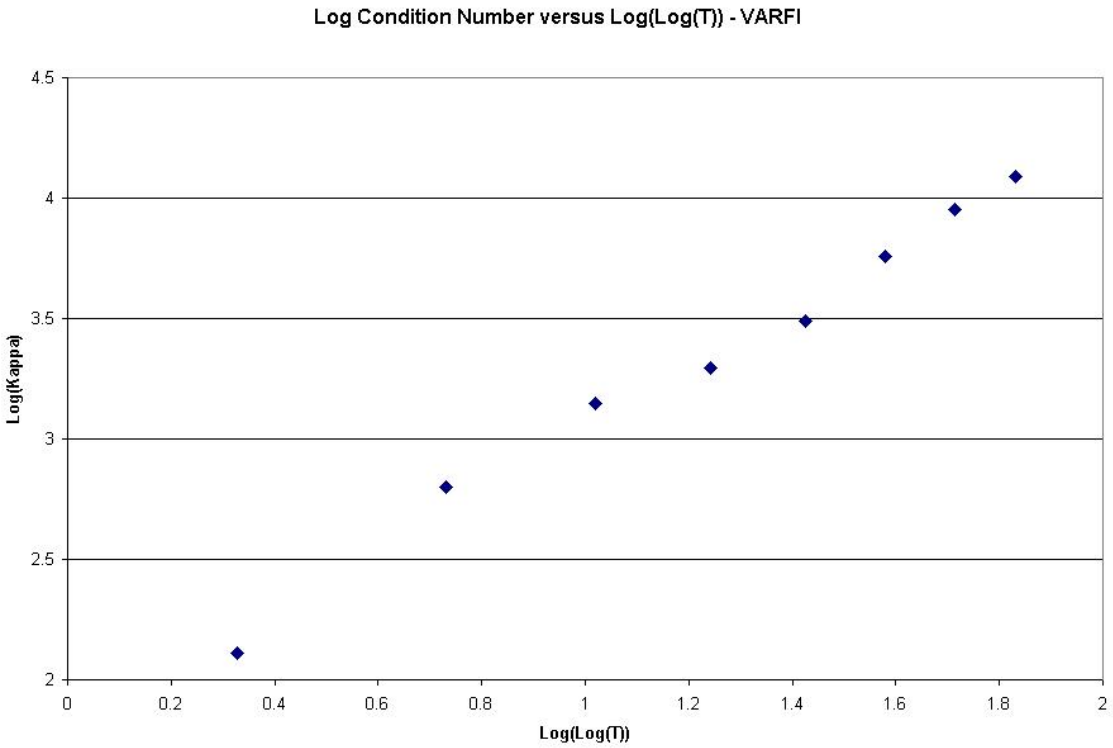

Figure 5: Plot of the logged condition number versus $\log(\log(T))$ for a $VARFI(1, \vec{d})$ process with parameters $d = (0.1, 0.4)$, $\Sigma = (1, 0.5, 0.5, 2)$, and $A_1 = (0.6, -0.1, 0.2, 0.8)$.

T	$\kappa(\Omega)$	$\kappa(C^{-1}\Omega)$	$\log(\kappa(\Omega))$	$\log(\kappa(C^{-1}\Omega))$
4	688.5823	8.2331	6.5346	2.1082
8	1537.3445	16.4618	7.3378	2.8010
16	2892.7798	23.2247	7.9700	3.1452
32	5123.9246	26.9049	8.5417	3.2923
64	8907.3649	32.7076	9.0946	3.4876
128	15417.6091	42.8939	9.6433	3.7587
256	26681.6308	52.1514	10.1917	3.9542
512	46214.2378	59.8457	10.7410	4.0918

Table 6: Condition number of autocovariance matrices for a $VARFI(1, \vec{d})$ process with parameters $d = (0.1, 0.4)$, $\Sigma = (1, 0.5, 0.5, 2)$, and $A_1 = (0.6, -0.1, 0.2, 0.8)$, for a range of T.

$\Omega = (V^{-1} \otimes I)\Omega_0((V')^{-1} \otimes I)$. Applying the definition of a condition number,

$$\begin{aligned}
\kappa(\Omega) &= \max_{x \in \mathbb{R}^{KT}} \frac{\|\Omega x\|}{\|x\|} \\
&= \max_{x \in \mathbb{R}^{KT}} \left(\frac{\|\Omega x\|}{\|\Omega_0((V')^{-1} \otimes I)x\|} \times \frac{\|\Omega_0((V')^{-1} \otimes I)x\|}{\|((V')^{-1} \otimes I)x\|} \times \frac{\|((V')^{-1} \otimes I)x\|}{\|x\|} \right) \\
&\leq \max_{x \in \mathbb{R}^{KT}} \left(\frac{\|\Omega x\|}{\|\Omega_0((V')^{-1} \otimes I)x\|} \right) \times \max_{x \in \mathbb{R}^{KT}} \left(\frac{\|\Omega_0((V')^{-1} \otimes I)x\|}{\|((V')^{-1} \otimes I)x\|} \right) \\
&\quad \times \max_{x \in \mathbb{R}^{KT}} \left(\frac{\|((V')^{-1} \otimes I)x\|}{\|x\|} \right) \\
&= \kappa(V^{-1} \otimes I)\kappa(\Omega_0)\kappa((V')^{-1} \otimes I) \\
&= \kappa(V)^2 \kappa(\Omega_0)
\end{aligned}$$

Using Sowell's (1989a) representation of bivariate cointegration with $V = \begin{pmatrix} 1 & 0 \\ \rho & 1 \end{pmatrix}$, $\kappa(V)$ is larger for larger values of $|\rho|$. This shows that both the cointegrating matrix and the autocovariance sequence of the associated FIVAR process affect the condition number of the resulting covariance matrix.

In addition to looking at the condition numbers, we can compare the processing time of this algorithm to the processing time needed to compute the quadratic form using the method of Sowell [1989b]. We time the Sowell method in two parts. First, the sequence of matrices, $v(n), d(n), A(n,k)$, must be computed. Then, those matrices must be used to compute the quadratic form itself. In Table 7, we present the processing time needed to compute the quadratic form using the PCG algorithm and Sowell's method. While the two methods are comparable for very small samples, we see that the PCG algorithm is almost ten times faster than Sowell's method at a sample size as small as 64. In Figure 6, we can also see that the time needed to use Sowell's method dwarfs the time needed for PCG; this figure also confirms that the processing time needed for Sowell's method grows quadratically with the sample size. In Figure 7, we plot only the processing time needed for the PCG method. As we expect from the discussion of condition numbers above, the PCG processing time seems to grow at less than a quadratic rate.

2.6.4 Relationship to Periodogram

In this section, we extend the analysis of Chen et al. [2006] to show that the block-circulant preconditioner is related to the expected value of the cross-periodogram of the multivariate time series, X_t. Let $I(\nu_s)$ be the $K \times K$ cross-periodogram of the vector X_t, where $\nu_s = \frac{2\pi s}{T}$ is the s^{th} Fourier frequency. Then, we may write $I(\nu_s)$ and its expectation in terms of the sample cross-covariance and its expectation [for example Brockwell and Davis, 1993, p. 443]:

$$\begin{aligned} I(\nu_s) &= \sum_{r=-(T-1)}^{T-1} \hat{\omega}(r) \exp(-ir\nu_s) \\ E(I(\nu_s)) &= \sum_{r=-(T-1)}^{T-1} E(\hat{\omega}(r)) \exp(-ir\nu_s) \end{aligned}$$

T	Sowell Setup	Sowell Quadratic Form	PCG
8	0.007	0.009	0.007
16	0.021	0.021	0.010
32	0.079	0.051	0.015
64	0.276	0.141	0.026
128	1.041	0.417	0.049
256	4.076	1.350	0.090
512	16.149	4.686	0.176
1024	64.07194	17.758	0.351
2048	255.430	67.672	0.730

Table 7: Processing time used to compute $X\Omega^{-1}X$ where X is a vector of ones and Ω is the autocovariance matrix for the $FIVAR(0, \vec{d})$ with $d = (0.1, 0.4)$ and $\Sigma = (1, 0.5, 0.5, 2)$. All times are the mean processing time as measured by R, over 100 repetitions.

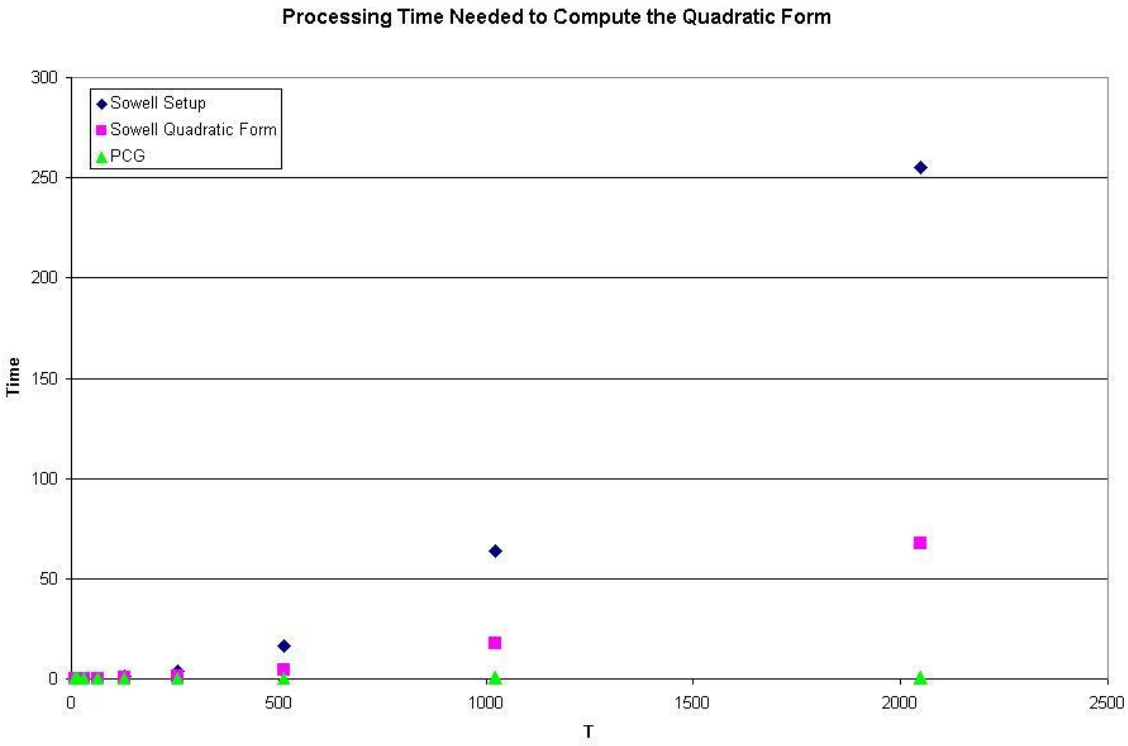

Figure 6: Processing time needed for quadratic form computation methods for various sample sizes.

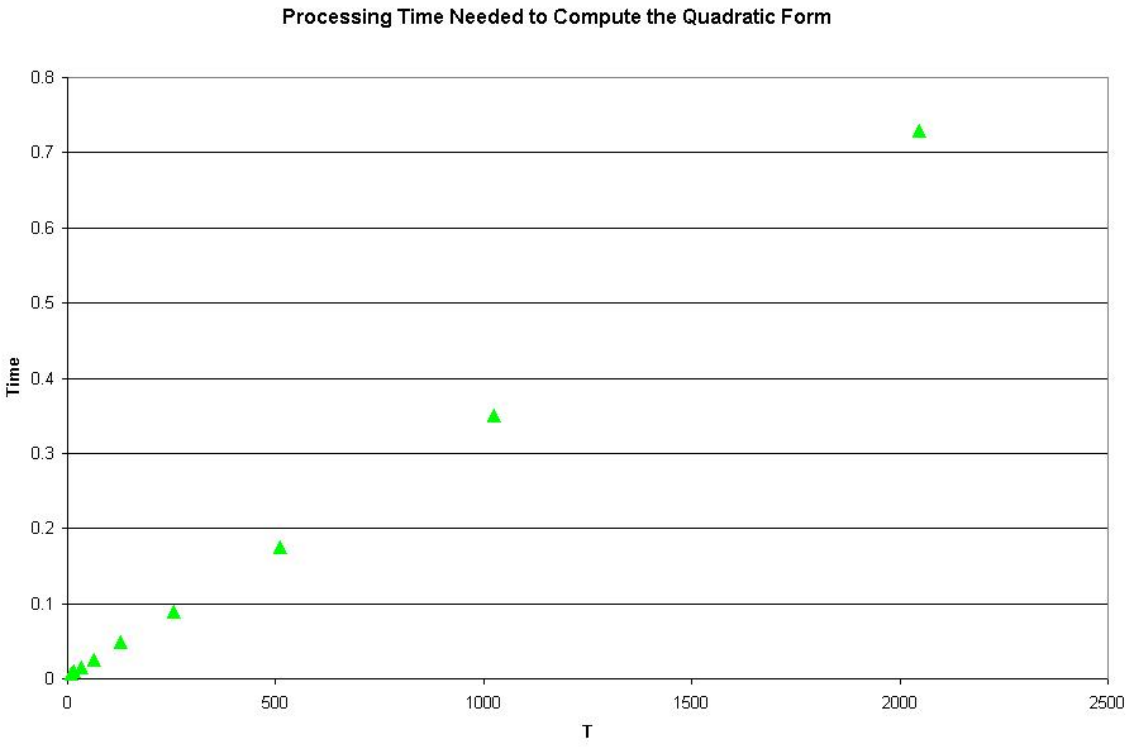

Figure 7: Processing time needed for quadratic form computation using the PCG algorithm.

where $\hat{\omega}(r)$ is the sample cross-covariance of x_t at lag r, defined as:

$$\hat{\omega}(r) = \begin{cases} \frac{1}{T}\sum_{t=1}^{T-r} X_{t+r}X_t' & 0 \leq r \leq T-1 \\ \frac{1}{T}\sum_{t=-r+1}^{T} X_{t+r}X_t' & -T+1 \leq r < 0 \end{cases}$$

(This definition differs slightly from Brockwell and Davis [1993, page 407] because we do not subtract off the sample mean.) Note that $E(\hat{\omega}(r)) = \frac{T-|r|}{T}\omega(r)$. Meanwhile, we may compute the elements of Chan and Olkin's preconditioner for the (i,j) block and its eigenvalues, $\lambda_{ij}(s)$, in terms of the covariances, $\omega_{i,j}(r)$:

$$\begin{aligned} c_r &= \frac{1}{T}\left(r\omega_{ij}(-(T-r)) + (T-r)\omega_{ij}(r)\right) \\ \lambda_{ij}(s) &= \sum_{r=0}^{T-1}\frac{r}{T}\omega_{ij}(-(T-r))\exp(-ir\nu_s) + \sum_{r=0}^{T-1}\frac{T-r}{T}\omega_{ij}(r)\exp(-ir\nu_s) \\ &= \sum_{q=-(T-1)}^{-1}\frac{T-|q|}{T}\omega_{ij}(q)\exp(-iq\nu_s) + \sum_{r=0}^{T-1}\frac{T-r}{T}\omega_{ij}(r)\exp(-ir\nu_s) \end{aligned}$$

where the last line follows from the substitution $q = -(T-r)$. Notice that the last line equals the (i,j) element of $E(I(\nu_s))$. Thus, the s^{th} eigenvalue of the (i,j) block of the preconditioner equals the expected value of the cross-periodogram of X_{it} and X_{jt} at ν_s. This corresponds to the results found in the univariate case, given in Chen et al. [2006, section 4].

2.6.5 Prediction

The preconditioned conjugate gradient algorithm which we have discussed can also be applied to efficiently compute the best linear predictor of multivariate processes. Notice that, for any Gaussian time series and lead time $h > 0$,

$$E(X_{T+h}|X) = E(X_{T+h}) + Cov(X, X_{T+h})Cov(X)^{-1}(X - E(X))$$

The preconditioned conjugate gradient algorithm can be used to compute $Cov(X)^{-1}(X - E(X))$, and the remaining multiplication can be computed in $O(TK)$ time. This

gives an efficient prediction computation based on the full sample and known covariance structure, which allows us to avoid computing an autoregressive approximation.

2.7 Computing the Determinant

Let $\Omega(T)$ be the covariance matrix of T observations of any multivariate process, X_t, that has an infinite moving average representation driven by innovations, ϵ_t, that have covariance matrix $\Sigma = E(\epsilon_t \epsilon_t')$. As before, let the autocovariance matrix of X_t at lag r be $\omega(r)$. According to Sowell [1989b], we may write:

$$\begin{aligned} |\Omega(T)| &= \prod_{r=0}^{T-1} |v(r)| \\ v(r) &= v(0) - \Upsilon(r)' \Omega(r)^{-1} \Upsilon(r) \\ \Upsilon(r) &= \begin{pmatrix} \omega(-1) \\ \vdots \\ \omega(-r) \end{pmatrix} \end{aligned}$$

$v(r)$ is the prediction variance of X_t given $X_{t-1}, ..., X_{t-r}$. Notice that we may use the PCG algorithm presented in section 2.6 K times to compute $\Omega(r)^{-1}\Upsilon(r)$, by using PCG on each column of $\Upsilon(r)$ separately. This means that $v(r)$ can be computed efficiently for any particular value of r. However, computing all of the $v(r)$ using the PCG algorithm would be slower than the $O(T^2)$ time required by Sowell's (1989b) method presented in section 2.4.2. Instead, we use our knowledge of $|v(r)|$ as a function of r and consider a variety of methods which may allow us to approximate the determinant in less processing time.

We begin by noting a few facts about $|v(r)|$ as a function of r. These facts hold for any multivariate time series with a moving average representation. First,

$|v(r)|$ is a non-increasing function, since

$$\begin{aligned} |v(r)| &= |\mathrm{Var}(X_t|X_{t-1},...,X_{t-r})| \\ &\geq |\mathrm{Var}(X_t|X_{t-1},...,X_{t-r},X_{t-r-1})| \\ &= |v(r+1)| \end{aligned}$$

Second, $|v(r)|$ is bounded below by $|\Sigma|$, since ϵ_t is uncorrelated with all past observations. Third, we can use the equations given in Sowell's algorithm to find that:

$$\begin{aligned} v(r)-v(r-1) &= -\sum_{j=1}^{r} A(r,j)\omega(-j) + \sum_{j=1}^{r-1} A(r-1,j)\omega(-j) \\ &= \left(\sum_{j=1}^{r-1}(A(r-1,j)-A(r,j))\omega(-j)\right) - A(r,r)\omega(-j) \\ &= \left(\sum_{j=1}^{r-1} A(r,r)\bar{A}(r-1,r-j)\omega(-j)\right) - A(r,r)\omega(-j) \\ &= A(r,r)\bar{D}(r) \\ &= A(r,r)\bar{A}(r,r)v(r-1) \\ \frac{|v(r)|}{|v(r-1)|} &= |I - A(r,r)\bar{A}(r,r)| \end{aligned}$$

This result is the analog of the result in the univariate case that $\frac{v(r)}{v(r-1)} = 1-\phi_r^2$, where ϕ_r is the r^{th} partial autocorrelation [for example, Brockwell and Davis, 1993].

In addition, we have found empirically for both FIVAR and VARFI models that $|v(r)|$ is quite smooth as a function of r, when $r > 0$. This smoothness does not hold as well at $|v(0)|$, since the inclusion of the first lagged value in predictions reduces the prediction variance quite dramatically because of the long memory. (See Figure 8 for an example.) This observation, together with the theoretical facts about $|v(r)|$, inform our choice of methods to compute the determinant.

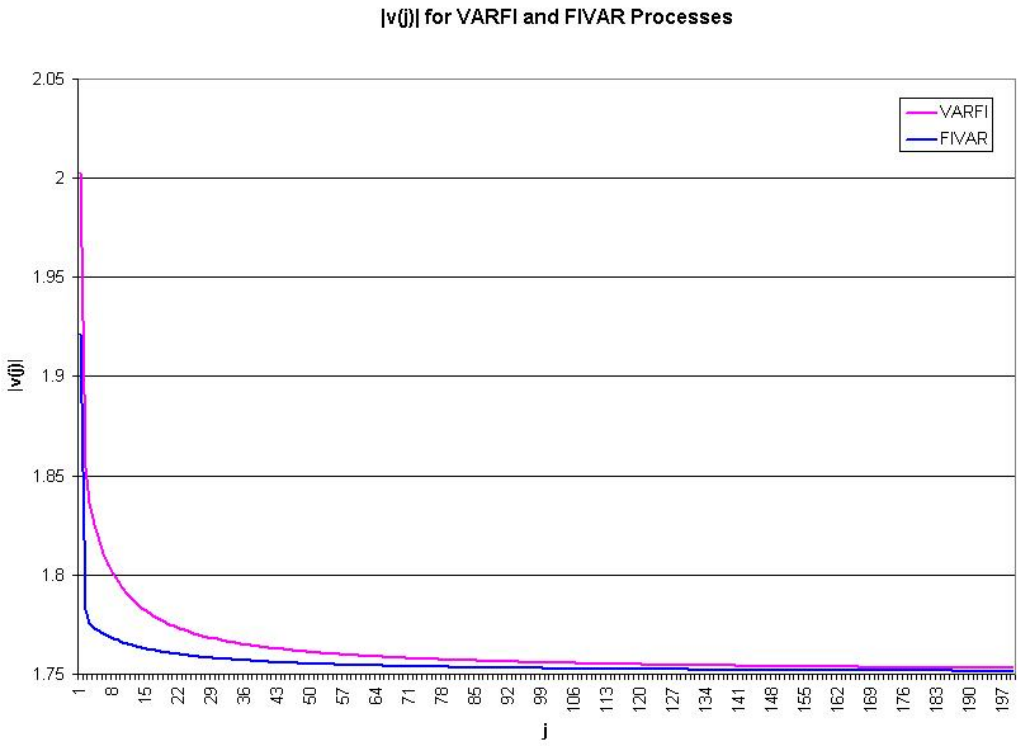

Figure 8: $|v(r)|$ for VARFI and FIVAR processes for r ranging from 1 to 199.

In the univariate case, Chen et al. [2006] suggest using an asymptotic approximation given by Bottcher and Silbermann [1999]. Also in the univariate case, Rohit Deo (private communication) has proposed an exact computational method, which generalizes to VARFI processes but not to FIVAR processes; we discuss the generalization in section 2.7.3. Sowell's (1989b) decomposition can be used to compute the exact determinant for univariate and multivariate processes, but it requires $O(T^2)$ time. In this section, we will discuss two alternatives to Sowell's method. First, we describe asymptotic approximations. Second, we discuss approximations that use curve-fitting and show the effectiveness of this method. In the final two sections, we describe ways to compute VARFI determinants and the determinants of cointegrated systems; these two methods would be exact if we had exact expressions for the determinants of the covariance matrices associated with $VARFI(0, \vec{d})$ or FIVAR processes respectively. None of the possible alternatives is exact. However, as we will see in section 4.6, using the approximation we describe instead of Sowell's exact determinant does not change parameter estimates by very much while it does speed up computation.

2.7.1 Asymptotic approximations to determinants

In this section, we will discuss two asymptotic approximations to the determinant. In our presentation, we will use two different notations for different types of asymptotic formulas. First, we write $f(T) \sim g(T)$ if $\lim_{T\to\infty} \frac{f(T)}{g(T)} = 1$. In some cases, we will also consider $\lim_{t\to\infty} \frac{f(T)}{f(T-1)}$. Notice that, if $f(T) \sim g(T)$ and

$\lim_{T\to\infty} g(T) \neq 0$:

$$\begin{aligned}\lim_{T\to\infty} \frac{f(T)/f(T-1)}{g(T)/g(T-1)} &= \lim_{T\to\infty} \left(\frac{f(T)}{g(T)}\right)\left(\frac{g(T-1)}{f(T-1)}\right) \\ &= \lim_{T\to\infty} \left(\frac{f(T)}{g(T)}\right) \lim_{T\to\infty} \left(\frac{g(T-1)}{f(T-1)}\right) \\ &= 1\end{aligned}$$

and $\frac{f(T)}{f(T-1)} \sim \frac{g(T)}{g(T-1)}$. If we take the logarithm of $\frac{f(T)}{g(T)}$, we have $\log f(T) = \log g(T) + o(1)$.

We now consider asymptotic approximations to either the overall determinant, $|\Omega(T)|$, or to the individual $|v(r)|$. In the univariate case, Bottcher and Silbermann [1999] give an asymptotic formula for the overall determinant of a univariate ARFIMA process . Taking the ratio of the approximations for r and $r-1$ yields an approximation for $|v(r)|$ in the univariate case. This approximation is:

$$\begin{aligned}|v(r)| &\sim |\Sigma| \exp\left(\frac{d^2}{r-1}\right) \\ \log|v(r)| &= \log|\Sigma| + \frac{d^2}{r-1} + o(1)\end{aligned}$$

Torsten Ehrhardt (private communication) found that this asymptotically correct formula can be extended to the multivariate case by replacing d^2 by a different constant. In the case of a $VARFI(0,\vec{d},0)$ or $FIVAR(0,\vec{d},0)$ model where Σ is invertible, he has worked out the expression for this constant. Let δ be the $K \times K$ diagonal matrix with $e^{-2\pi i d_1}, ..., e^{-2\pi i d_K}$ along the diagonal. Define

$$U = \Sigma\delta^*\Sigma^{-1}\delta$$

Let $e^{2\pi i u_1}, ..., e^{2\pi i u_K}$ be the eigenvalues of U. Since $|U| = |\Sigma||\delta^{-1}||\Sigma^{-1}||\delta| = 1$, we must have $1 = \prod_{k=1}^K e^{2\pi i u_k} = e^{2\pi i \sum_{k=1}^K u_k}$. While it is always possible to choose the u_k so that $\sum_{k=1}^K u_k = 0$, Ehrhardt's expression might not hold if the u_k chosen are not the principal branch logarithms. However, Ehrhardt conjectures that

for any choice where $|Re(u_k) - Re(u_j)| < 1$, the method will continue to hold. Given choices for u_k which obey this condition and which sum to 0, we have $|v(r)| \sim |\Sigma| \exp\left(\frac{\sum_{k=1}^{K} d_k^2 - \frac{1}{2}\sum_{k=1}^{K} u_k^2}{r-1}\right)$. We find that, when there is no short memory component, Ehrhardt's approximation improves monotonically as n increases, as seen in Figures 9 and 10. In addition, the error in the approximation is always of the same sign. The assumption that the error is monotonically decreasing allows us to bound the error in the approximation beyond a certain point. However, Ehrhardt's approximation does not give us a way to reduce the error beyond the initial approximation because there are no higher order terms. In fact, even if the term in the exponent is not correct, the approximation will eventually be close, since both the approximation and the true values tend toward $|\Sigma|$; in this case, the errors might not decrease monotonically and the approximation might not be as accurate for small n; we can see in this in Figure 10, looking at the two lines with $A_1 \neq 0$. However, as we see in Figure 10, Ehrhardt's approximation does not work well for small values of r.

We also consider the simpler approximation of the determinant of the covariance matrix by $|\Sigma|^T$, as suggested by Dunsmuir and Hannan [1976] and others. This is equivalent to approximating $|v(r)|$ by Σ for all r. This approximation ignores the $\exp\left(\frac{\sum_{k=1}^{K} d_k^2 - \frac{1}{2}\sum_{k=1}^{K} u_k^2}{r-1}\right)$ term of Ehrhardt's approximation. This term does not exist in short memory cases, since each $d_k = u_k = 0$. Furthermore, in the case of a $VAR(p)$ process, $v(r) = \Sigma$ for all $r \geq p$, because the prediction error based on the previous p observations is simply the next innovation, ϵ_t. In that case,

$$\begin{aligned} |\Omega(T)| &= \prod_{r=0}^{T-1} |v(r)| \\ &= |\Sigma|^{T-p} \prod_{r=0}^{p-1} |v(r)| \end{aligned}$$

Thus, for vector autoregressions, computation of the exact determinant requires

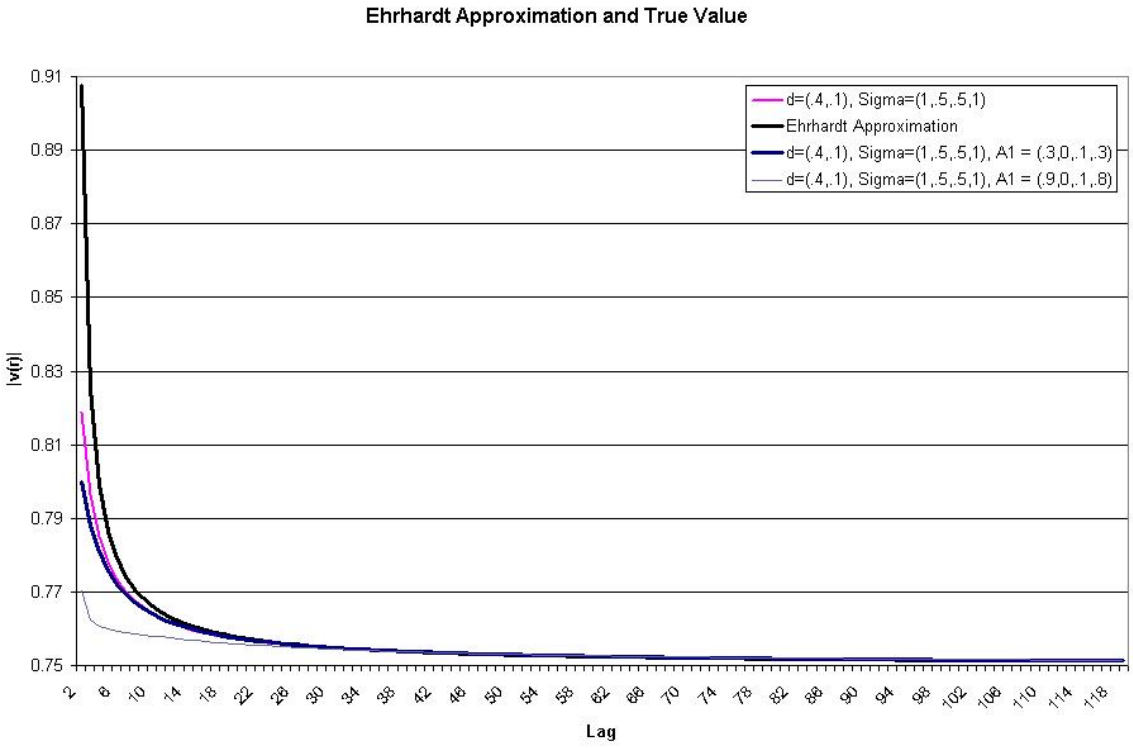

Figure 9: The Ehrhardt approximation to $|v(r)|$ and the true values for $|v(r)|$ for a variety of FIVAR processes.

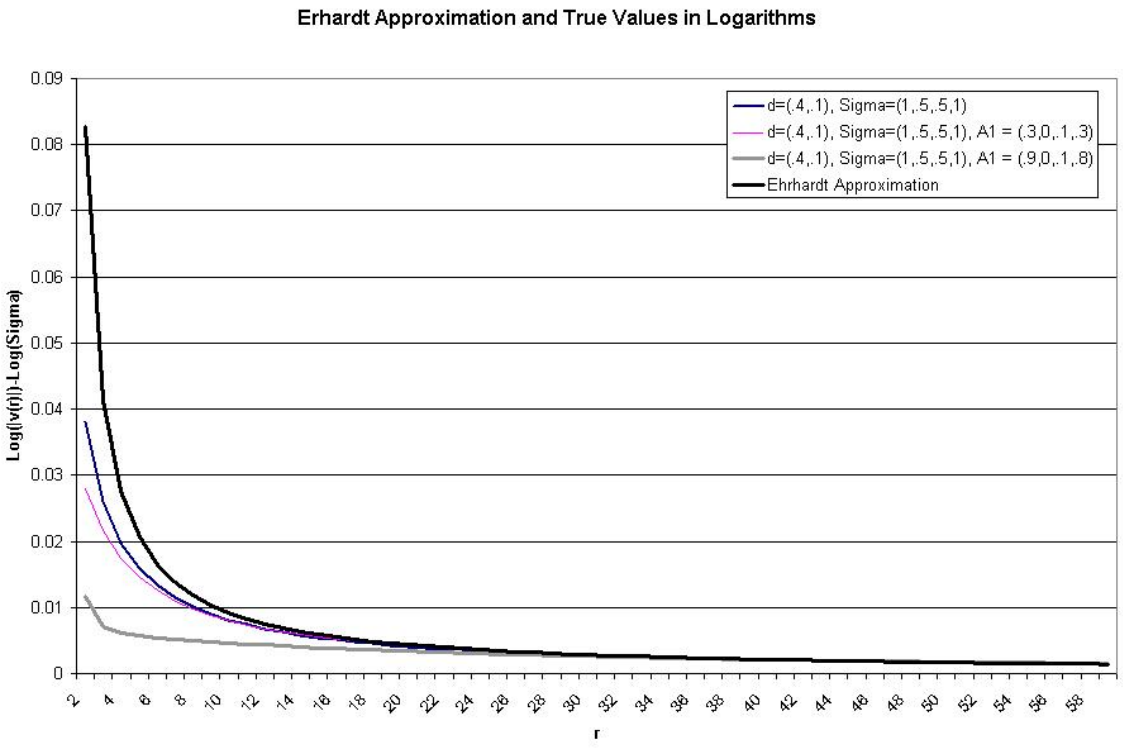

Figure 10: The Ehrhardt approximation to $|v(r)|$ and the true values for $|v(r)|$ for a variety of FIVAR processes, divided by $|\Sigma|$ and then logged.

the computation of only a fixed number of initial $|v(r)|$. Furthermore,

$$\begin{aligned}\lim_{T\to\infty}\frac{1}{T}\log|\Omega(T)| &= \lim_{T\to\infty}\frac{1}{T}\left((T-p)\log|\Sigma|+\sum_{r=0}^{p-1}\log|v(r)|\right)\\ &= \log|\Sigma|\end{aligned}$$

For this reason, it seems reasonable to approximate $|\Omega(T)|$ by $|\Sigma|^T$ for vector autoregressive models, even though the term containing $\log|\Omega(T)|$ is not divided by T in the expression for the likelihood. In a more general univariate case, under the conditions of a theorem of Grenander and Szego [1958, page 76], we must have:

$$\lim_{T\to\infty}\frac{\Omega(T)}{|\Sigma|^T}=C$$

where C is a constant that depends on the moving average representation of X_t. Even in this simple case, the assumption that $C=1$ will not be accurate. This approximation has additional problems in the long memory case. One of the conditions of Grenander and Szego's theorem is that the spectral density, f, is differentiable and that the derivative, f', obeys:

$$|f'(x_1)-f'(x_2)|<K|x_1-x_2|^\alpha$$

with $K>0$ and $0<\alpha<1$; this excludes the case of long memory. The approximations offered by Dunsmuir and Hannan [1976] and Luceno [1996] for more general models assume that this limit continues to hold. However, the Ehrhardt approximation shows that $\log|v(r)|=\log|\Sigma|+\frac{\sum_{k=1}^K d_k^2-\frac{1}{2}\sum_{k=1}^K u_k^2}{r-1}+o(1)$. Since $\sum_{r=0}^T\frac{1}{r-1}$ diverges as $T\to\infty$, the approximation of $|\Omega(T)|$ by $|\Sigma|^T$ may not be good enough for estimation. Our results in section 2.9.2 confirm this.

2.7.2 Determinant approximations using curve-fitting

Instead of using an asymptotic approximation, we consider using regression and curve-fitting as a way to interpolate between a few computed values of $|v(r)|$.

We expect that the best fits will come from functions which are decreasing and have a finite asymptotic value, so that they can mimic the known behavior of $|v(r)|$. For such a method to be feasible, we must be able to find a fit that is reasonably accurate based on computing only a subset of the $|v(r)|$ exactly, using either Sowell's method or PCG. We focus on fitting:

$$r\sqrt{|v(r)|} \;=\; \alpha + \beta r$$

This relationship is equivalent to:

$$|v(r)| \;=\; \beta^2 + \frac{2\alpha\beta}{r} + \frac{\alpha^2}{r^2}$$

which is decreasing and smooth in r. In this formulation, β^2 is able to adjust to match the asymptotic value of $|v(r)|$.

We will combine curve-fitting with the application of Sowell's method to an initial set of points. Though Sowell's method is too slow to use to compute $|v(r)|$ for all $r = 0, ..., T-1$ when T is large, it can be used for some of the initial points, $r = 0, ..., S$, where the curve may be hardest to fit and the approximation is least accurate. As long as the initial segment of points used with Sowell's method grows more slowly than T, we can use this method to compute some of the determinants of the prediction variances exactly without much additional computational cost.

Our current method combines a regression with the application of Sowell's method. First, we apply Sowell's method to compute $|v(r)|$ for $r = 0, ..., S$, for some S (we use 32 in our program). Then, we use PCG to compute $|v(T-1)|$. We then regress $r\sqrt{|v(r)|}$ on r for $r = 1, .., S, T-1$. Using the fitted line, we estimate $|v(r)|$ for all the points where $|v(r)|$ is unknown.

Algorithm 2.8 *Approximating $|\Omega(T)|$ through curve-fitting.*

1. *Use Sowell's algorithm (Algorithm 2.3) to compute $|v(r)|$ for $r = 0, ..., S$.*

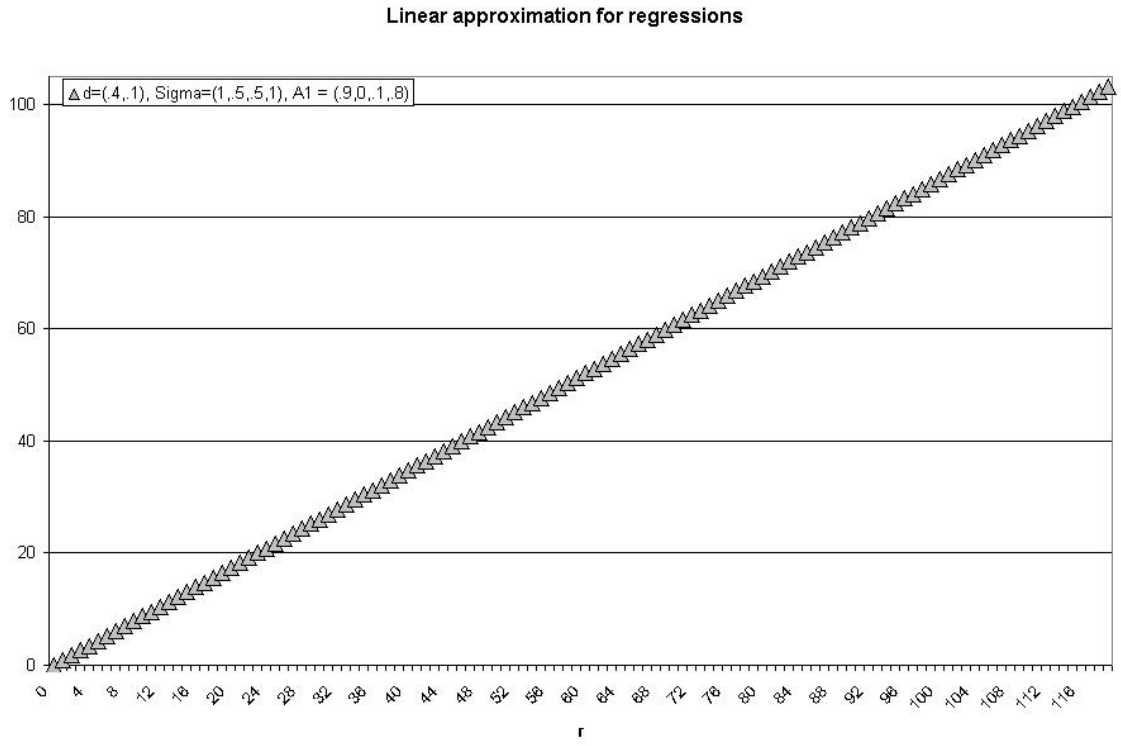

Figure 11: A plot of r versus $r\sqrt{|v(r)|}$ for the FIVAR process with $d = (0.1, 0.4)$, $\Sigma = (1, 0.5, 0.5, 2)$ and $A_1 = (0.7, 0.1, 0.2, 0.9)$.

2. *Compute* $|v(T-1)|$ *using the PCG algorithm:*

   (a) *Set* Υ *to be the* $KT \times K$ *matrix which stacks the autocovariance matrices,* $\omega(-1), ..., \omega(-r)$.

   (b) *Set* G *to be a* $KT \times K$ *matrix.*

   (c) *For* $i = 1, ..., K$, *compute the* i^{th} *column of* G *as* $\Omega^{-1}\Upsilon(\cdot, i)$ *using the PCG algorithm, where* $\Upsilon(\cdot, i)$ *is the* i^{th} *column of* Υ.

   (d) *Compute* $v(T-1) = \Upsilon' G$.

   (e) *Compute* $|v(T-1)|$.

3. *Regress* $r\sqrt{|v(r)|}$ *on* r *for the points* $r = 1, .., S, T-1$.

4. *Compute the fitted values,* $\widehat{|v(r)|}$ *for* $r = S+1, ..., T-2$ *based on the fitted values from the regression.*

5. *Sum the logarithms of* $|v(0)|, ...|v(S)|, |\widehat{v(S+1)}|, ..., |\widehat{v(T-2)}|, |v(T-1)|$ *to find the approximate log determinant.*

While this method is ad hoc, Tables 8 and 9 show that it performs well for both FIVAR and VARFI models. The approximation is closest when A_1 is far from the unit circle, but our approximate log determinant is within 0.5 of Sowell's exact log determinant even in the case where $A_1 = \begin{pmatrix} .7 & .2 \\ .1 & .9 \end{pmatrix}$, which has one eigenvalue greater than 0.97. The approximation is better for VARFI than for FIVAR models. The difference in computing time between Sowell's exact method and our regression-based approximation is quite large; when $T = 1000$, Sowell's algorithm takes almost 70 times longer than our approximation. Furthermore, we will see in Section 2.9.2 that the maximum likelihood estimates for the parameters based on using this determinant are close to those from Sowell.

T	A_1	d	Sowell Time	Sowell Value	Reg. Time	Reg. Value	Naive
250	(0,0,0,0)	(.4,.1)	3.966	141.7575	0.292	141.7568	139.9
250	(.4,.2,.1,.6)	(.4,.1)	3.950	143.6495	0.311	143.6363	139.9
250	(.7,.2,.1,.9)	(.4,.1)	3.978	151.4243	0.395	151.2217	139.99
500	(0,0,0,0)	(.4,.1)	18.359	281.7858	0.683	281.7827	279.8
500	(.4,.2,.1,.6)	(.4,.1)	18.302	283.7176	0.751	283.6769	279.8
500	(.7,.2,.1,.9)	(.4,.1)	18.184	291.8804	1.215	291.4227	279.8
1000	(0,0,0,0)	(.4,.1)	74.211	561.7179	0.798	561.7127	559.6
1000	(.4,.2,.1,.6)	(.4,.1)	73.016	563.6902	0.864	563.623	559.6
1000	(.7,.2,.1,.9)	(.4,.1)	74.146	572.2505	1.146	571.5228	559.6158
250	(0,0,0,0)	(.4,.49)	3.883	145.9179	0.313	145.9187	139.9
250	(.4,.2,.1,.6)	(.4,.49)	3.895	148.6055	0.320	148.5785	139.9
250	(.7,.2,.1,.9)	(.4,.49)	3.880	157.7377	0.552	157.6283	139.9
500	(0,0,0,0)	(.4,.49)	18.134	286.1003	0.734	286.1026	279.8
500	(.4,.2,.1,.6)	(.4,.49)	18.167	288.7922	0.797	288.7112	279.8
500	(.7,.2,.1,.9)	(.4,.49)	18.198	298.052	1.633	297.8051	279.8
1000	(0,0,0,0)	(.4,.49)	73.729	566.18648	0.858	566.19019	559.6
1000	(.4,.2,.1,.6)	(.4,.49)	72.877	568.88358	0.901	568.75156	559.6
1000	(.7,.2,.1,.9)	(.4,.49)	74.253	578.28725	1.267	577.86903	559.6

Table 8: The computed value of the log determinant and the processing time required to do the computation using Sowell's algorithm and using the regression-based approximation. The naive approximation is $\log |\Sigma|^T$. All models are FIVAR processes with $\Sigma = (1, .5, .5, 2)$. Times are the mean time taken over 100 repetitions of the calculation.

T	A_1	d	Sowell Time	Sowell Value	Reg. Time	Reg. Value	Naive
250	(0,0,0,0)	(.4,.1)	3.930	141.7575	0.294	141.75678	139.9
250	(.4,.2,.1,.6)	(.4,.1)	4.004	143.0659	0.320	143.05746	139.9
250	(.7,.2,.1,.9)	(.4,.1)	3.919	147.4836	0.346	147.44006	139.9
500	(0,0,0,0)	(.4,.1)	17.905	281.7858	0.679	281.78269	279.8
500	(.4,.2,.1,.6)	(.4,.1)	17.998	283.0938	0.738	283.06462	279.8
500	(.7,.2,.1,.9)	(.4,.1)	17.919	287.5041	0.868	287.40505	279.8
1000	(0,0,0,0)	(.4,.1)	73.589	561.7179	0.797	561.71271	559.6
1000	(.4,.2,.1,.6)	(.4,.1)	72.954	563.0257	0.841	562.97756	559.6
1000	(.7,.2,.1,.9)	(.4,.1)	73.227	567.4326	0.905	567.27000	559.6
250	(0,0,0,0)	(.4,.49)	3.949	145.9179	0.304	145.91866	139.9
250	(.4,.2,.1,.6)	(.4,.49)	3.922	148.0327	0.320	148.00202	139.9
250	(.7,.2,.1,.9)	(.4,.49)	3.894	153.6547	0.538	153.57598	139.9
500	(0,0,0,0)	(.4,.49)	17.932	286.1003	0.737	286.10259	279.8
500	(.4,.2,.1,.6)	(.4,.49)	17.871	288.2132	0.795	288.1236	279.8
500	(.7,.2,.1,.9)	(.4,.49)	18.020	293.8121	1.531	293.64902	279.8
1000	(0,0,0,0)	(.4,.49)	72.464	566.1865	0.853	566.1902	559.6
1000	(.4,.2,.1,.6)	(.4,.49)	72.054	568.2984	0.889	568.1529	559.6
1000	(.7,.2,.1,.9)	(.4,.49)	72.932	573.8849	1.213	573.6140	559.6

Table 9: The computed value of the log determinant and the processing time required to do the computation using Sowell's method and using the regression-based approximation. All models used $\Sigma = (1, .5, .5, 2)$ and a VARFI process.

Ideally, we also wish to move from this approximation to an approximation which can be made as close as desired with some additional computations; to accomplish this, we must find a way to bound the approximation error and reduce the error if desired. The approximation method we will present does not do this.

2.7.3 An alternative way to compute the determinant of a VARFI process

We now consider an alternative way to compute the covariances of the $VARFI(1, \vec{d})$ process, $X_.$. This algorithm for computing the determinant is a generalization of a univariate algorithm given by Rohit Deo (private communication). This method would be exact if we could compute the determinant of a $VARFI(0, \vec{d})$ process exactly. Since our approximation is close for such processes, we expect this approximation will also be close.

As before, let Ω be the covariance matrix of X, and $\omega(h) = \mathrm{Cov}(X_t, X_{t-h})$ be the $K \times K$ autocovariance matrix at lag h. Define a new process, W_t, by:

$$\begin{aligned} W_1 &= X_1 \\ W_t &= X_t - A_1 X_{t-1} \end{aligned}$$

Then, $W = (W_1', ..., W_T')'$ can be written as $W = BX$, where $|B| = 1$. Thus, $|\mathrm{Var}(W)| = |B'\Omega B| = |\Omega|$, and it is sufficient to compute $|\mathrm{Var}(W)|$. Notice that:

$$\mathrm{Var}(W) = \begin{pmatrix} \omega(0) & C' \\ C & \Phi(T-1) \end{pmatrix}$$

where $\omega(h)$ is the autocovariance of the original process, $\Phi(T-1)$ is the covariance matrix of a $VARFI(0, \vec{d}, 0)$ process of length $T-1$ and C is the $K(T-1) \times K$

matrix given by:

$$C = \begin{pmatrix} \omega(1) - A_1\omega(0) \\ \vdots \\ \omega(T-1) - A_1\omega(T-2) \end{pmatrix}$$

Using a formula for the determinant of a partitioned matrix [Sowell, 1989b], we compute:

$$|\mathrm{Var}(W)| = |\Phi(T-1)| \cdot |\omega(0) - C'\Phi(T-1)^{-1}C|$$

The first term must be computed using the method given in the previous section. The product $\Phi(T-1)^{-1}C$ can be computed using the PCG algorithm K times, once for each column of C. Then, since $\omega(0) - C'\Phi(T-1)C$ is a $K \times K$ matrix, computation of the determinant can be done quickly using standard methods.

2.7.4 Determinants of Cointegrated Systems

Let $\gamma(j)$ be the autocovariance sequence of a cointegrated system. Using the results from section 2.5.4, we know that $\gamma(j) = V^{-1}\omega(j)(V^{-1})'$, where $\omega(j)$ is the autocovariance sequence of the corresponding FIVAR process. Let

$$\Gamma(T) = (V^{-1} \otimes I)\tilde{\Omega}(T)((V^{-1})' \otimes I) \tag{2.15}$$

$$|\Gamma(T)| = |V|^{-2T}|\Omega(T)| \tag{2.16}$$

If we use a lower triangular representation with ones along the diagonal for the cointegrating relationship, then $|V| = 1$, and the determinant of the covariance matrix of a cointegrated system equals the determinant of the covariance matrix of the system before it is cointegrated. Even if we do not impose a restriction that implies that $|V| = 1$, this computation in equation (2.16) takes $O(1)$ time once $|\Omega(T)|$ is known.

2.8 Efficient Simulation

In this section, we present an efficient algorithm for simulating from a vector ARFIMA process with normally distributed innovations. Our approach extends the method proposed by Davies and Harte [1987]. Wood and Chan [1994] described the algorithm for a univariate time series in more detail and extended the algorithm to spatial time series in multiple dimensions but not to multivariate time series. The algorithm described in this section may be applied to other stationary multivariate time series, assuming that the conditions described are met.

As before, let Ω be the covariance matrix of the vector containing T periods of a stationary K-variate time series, where the data is grouped by series. The underlying idea of this algorithm is to embed Ω in a covariance matrix for a random vector which it is easy to simulate.

Recall that Ω is a block Toeplitz matrix, with K^2 blocks of size $T \times T$. Let $C(\Omega)$ be a block circulant embedding of Ω, where each block, $C_{ij}(\Omega)$, is of dimension M, with $M \geq 2T - 1$ and odd. We set the first row of $C(\Omega)$ equal to $\omega_{ij}(0), ..., \omega_{ij}(\frac{M-1}{2}), \omega_{ij}(-\frac{M-1}{2}), ..., \omega_{ij}(-1)$. Unlike the circulant embedding used in section 2.3.3, this embedding does not include a second diagonal with $\omega_{ij}(0)$.

Because $C(\Omega)$ is a block circulant matrix, we can apply the results of section 2.3.2 to write it as:

$$C(\Omega) = (I \otimes F^*)PB(\Omega)P'(I \otimes F)$$

where $B(\Omega)$ is a matrix with M blocks, $B_1, ..., B_M$, of size $K \times K$ along the diagonal. Using this representation, $C(\Omega)^{1/2}$ is straightforward to compute, using either the eigenvalue decomposition of each B_r or the algorithm of Denman and Beavers [1976]. Notice, however, that $C(\Omega)$ need not be a positive definite matrix. If it is not, the algorithm below will not apply, since $C(\omega)$ must be a covariance matrix for simulation. However, Wood and Chan [1994, proposition 2] notes that, in the

cases they consider, there is always a sufficiently large M such that the circulant embedding of size $M \times M$ will be positive definite. We have also found that omitting the second diagonal of $\omega_{ij}(0)$ generally results in a matrix that is positive definite. Because we do not repeat $\omega_{ij}(0)$ but we do repeat $\omega_{ij}(r)$ for every other r, M must be odd. For the efficiency of the fast Fourier transform, we recommend choosing M such that it has many small factors; choosing M to be a power of three allows it to be odd and have many factors. All of these considerations yield the following algorithm for computing B, which is a specialization of Algorithm 2.1 to this case:

Algorithm 2.9 Preparation for simulation using block circulant embedding.

1. *Choose* $M = 3^R$, *where* $3^{R-1} < 2T - 1 \leq 3^R$.

2. *Compute the* $K \times K$ *autocovariances,* γ, *at lags* $-\frac{M-1}{2}, ..., 0, ..., \frac{M-1}{2}$.

3. *Set the first row of each block of the circulant embedding equal to* $\omega_{ij}(0)$, ..., $\omega_{ij}(\frac{M-1}{2})$, $\omega_{ij}(-\frac{M-1}{2})$, ..., $\omega_{ij}(-1)$.

4. *Compute the inverse fast Fourier transform of each first block's first row. This yields* $B_r(i, j)$.

5. *For each* $r = 1, .., M$, *compute the eigenvalue decomposition of* B_r. *If any of the eigenvalues are negative, set* M *to* $3M$ *and return to step 2. Otherwise, compute* $B_r^{1/2}$ *and store the result.*

Given $C(\Omega)$, we require an algorithm to simulate a random vector with that covariance matrix. We extend the univariate algorithm of Wood and Chan [1994, section 5.1.2] to simulation from a block-circulant covariance matrix. Consider the random variable, $U \sim Normal(0, I_{M \times M})$. As long as $C(\Omega)$ is positive definite,

$C(\Omega)^{1/2}U$ exists and has covariance matrix $C(\Omega)$. The subvector of $C(\Omega)^{1/2}U$ defined by the first T elements of each of the K series has covariance matrix Ω. Thus, a fast method for simulating $C(\Omega)^{1/2}U$ yields a simulation method for the original multivariate time series. This suggests the following algorithm, in which we describe each step in terms of the spectral decomposition of $C^{1/2}$ given in section 2.3.2:

Algorithm 2.10 Simulation.

- $(I \otimes F)X$*: Compute vectors* $Y_{1\cdot}, ..., Y_{K\cdot}$ *of length* M *with* $Var(Y_{k,\cdot}) = FF^*$*, using the method given in Wood and Chan (1994, section 5.1.2).*
- $P'(I \otimes F)X$*: Combine these vectors in the order* $(Y_{11}, ..., Y_{K1}, ..., Y_{1M}, ..., Y_{KM})'$.
- $B_\Omega^{1/2}P'(I \otimes F)X$*: Compute* $B_r^{1/2}Y_r$ *for each* $r = 1, ..., M$.
- $PB_\Omega^{1/2}P'(I \otimes F)X$*: Re-sort the vector to group the observations by series instead of by time.*
- $(I \otimes F^*)PB_\Omega^{1/2}P'(I \otimes F)X$*: Take the fast Fourier transform of each* $Y_{k,\cdot}$ *for* $k = 1, ..., K$.
- *Return the first* T *observations from each vector,* $Y_{k,\cdot}$.

Consider the time requirements of this method. The initialization algorithm is run once. Given a choice of M, the computation of the autocovariances and fast Fourier transforms takes $O(M \log_3 M)$ time, while the eigenvalue calculations take $O(M)$ time; as in the other algorithms we have presented, larger values of K will slow these steps down. The required M is unknown, but we found in our experiments that it needed to be increased from the initial value given in Step 1 of Algorithm 2.9 when $T = 4$ (see Tables 10, 11, and 12). In that case,

$M = O(T)$. The simulation step also uses Fast Fourier Transforms, so that it also runs in $O(M \log_3 M)$ time. In contrast, the method of Sowell [1989b] requires an initial computation of his matrix decomposition, which takes $O(T^2)$ steps; each simulation takes another $O(T^2)$ steps, since the computation of:

$$X_t = \sum_{j=1}^{t-1} \bar{A}(t-1, t-j) X_j + \bar{v}(t-1)^{1/2} u_t$$

for $t = 1, ..., T$ will require $\frac{T(T-1)}{2}$ summations.

In Table 10, we show the processing time required for initialization of the algorithm and for each simulation for both Sowell and the block circulant embedding algorithm. In this test, the processing time required to compute the covariances for Sowell's simulation method is not included in the setup, but it is included in the setup for circulant embedding, since there was the change that M would need to be increased. Despite this disadvantage, our method always faster for simulation and is faster for the initialization except for small values of T.

2.9 Maximum Likelihood Estimation and Monte Carlo

We now combine all of the computational methods we have discussed so far to run Monte Carlo experiments using the various estimation methods. We first discuss how we parameterize our models to ensure stationarity and invertibility. Second, we use Monte Carlo experiments to describe the effect of approximating the determinant using the methods discusses in section 2.7. Finally, we compare maximum likelihood estimation methods to the Whittle estimator for a variety of sample sizes.

T	Sowell Setup	Sowell Simulation	Circulant Setup	Circulant Simulation	M
4	0.003	0.005	0.017	0.001	9
8	0.007	0.009	0.047	0.003	27
16	0.021	0.021	0.132	0.008	81
32	0.078	0.052	0.130	0.008	81
64	0.273	0.142	0.375	0.023	243
128	1.039	0.414	1.091	0.070	729
256	4.074	1.365	1.090	0.070	729
512	16.248	4.764	3.257	0.205	2187
1024	63.666	17.698	3.260	0.210	2187

Table 10: Processing time needed to set up for simulation and simulate from a $FIVAR(0, \vec{d})$ with $d = (0.1, 0.4)$ and $\Sigma = (1, 0.5, 0.5, 2)$. Estimates for the setup time are based on 100 repetitions; estimates for the simulation times are based on 1000 repetitions.

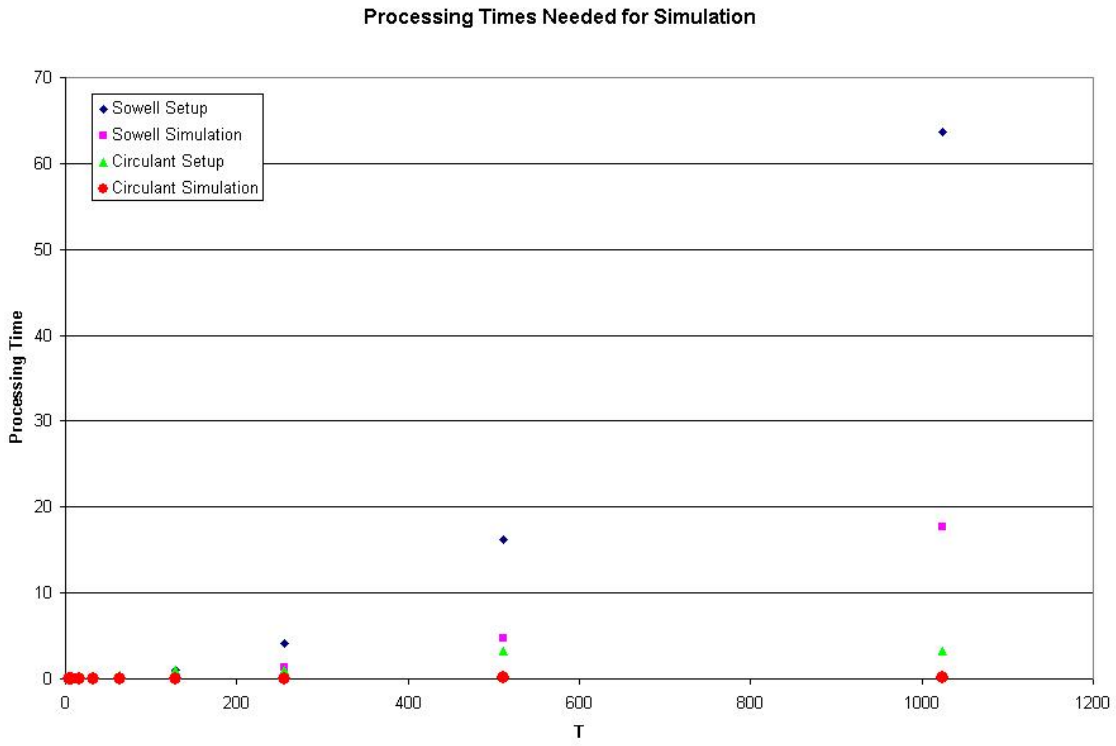

Figure 12: Processing time needed to set up for simulation and simulate from a $FIVAR(0, \vec{d})$ with $d = (0.1, 0.4)$ and $\Sigma = (1, 0.5, 0.5, 2)$. Estimates for the setup time are based on 100 repetitions; estimates for the simulation times are based on 1000 repetitions.

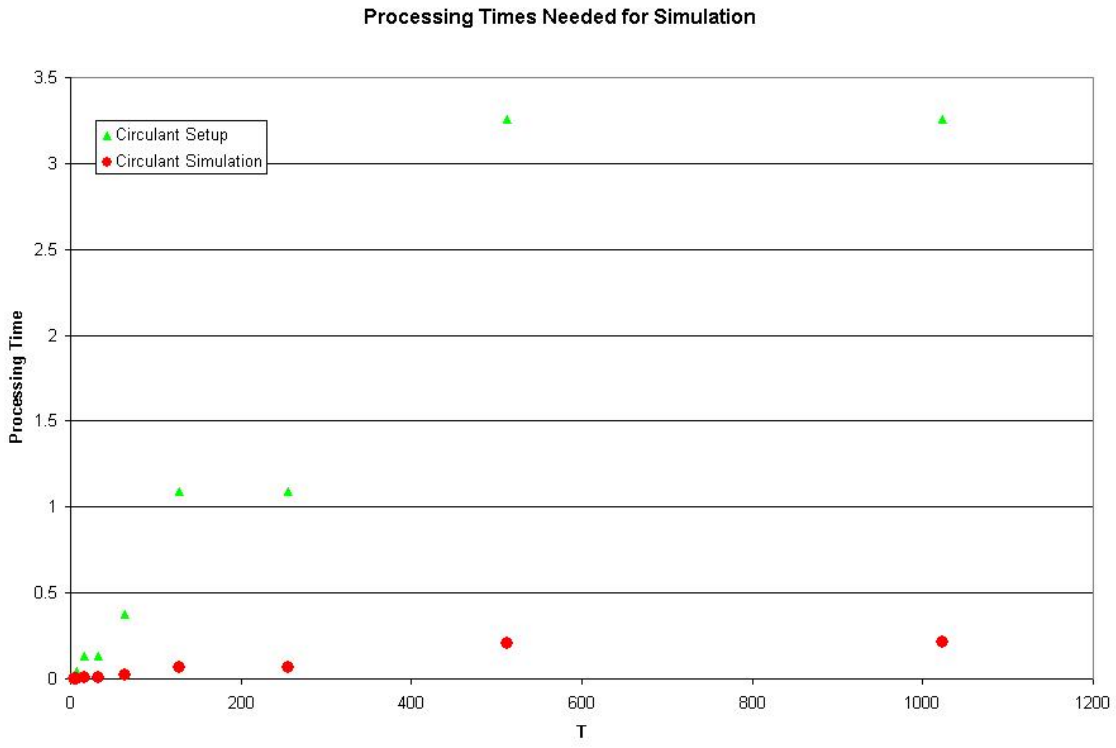

Figure 13: Processing time needed for the circulant embedding method to set up for simulation and simulate from a $FIVAR(0, \vec{d})$ with $d = (0.1, 0.4)$ and $\Sigma = (1, 0.5, 0.5, 2)$. Estimates for the setup time are based on 100 repetitions; estimates for the simulation times are based on 1000 repetitions.

T	Sowell Setup	Sowell Simulation	Circulant Setup	Circulant Simulation	M
4	0.005	0.012	0.222	0.005	27
8	0.013	0.025	0.128	0.005	27
16	0.041	0.058	0.293	0.013	81
32	0.131	0.130	0.275	0.012	81
64	0.450	0.302	0.793	0.039	243
128	1.753	0.824	2.198	0.117	729
256	6.670	2.544	2.118	0.113	729
512	27.383	9.062	6.559	0.352	2187
1024	111.766	36.267	6.741	0.397	2187

Table 11: Processing time needed to set up for simulation and simulate from a $FIVAR(1, \vec{d})$ with $d = (0.1, 0.4)$, $\Sigma = (1, 0.5, 0.5, 2)$, and $A_1 = (0.6, -0.1, 0.2, 0.8)$. Estimates for the setup time are based on 100 repetitions; estimates for the simulation times are based on 1000 repetitions.

T	Sowell Setup	Sowell Simulation	Circulant Setup	Circulant Simulation	M
4	0.005	0.010	0.230	0.006	27
8	0.013	0.023	0.153	0.006	27
16	0.052	0.053	0.414	0.016	81
32	0.168	0.107	0.367	0.013	81
64	0.470	0.285	0.762	0.040	243
128	1.599	0.713	2.125	0.109	729
256	7.424	2.425	2.529	0.107	729
512	24.251	7.763	6.096	0.310	2187

Table 12: Processing time needed to set up for simulation and simulate from a $VARFI(1, \vec{d})$ with $d = (0.1, 0.4)$, $\Sigma = (1, 0.5, 0.5, 2)$, and $A_1 = (0.6, -0.1, 0.2, 0.8)$. Estimates for the setup time are based on 100 repetitions; estimates for the simulation times are based on 1000 repetitions.

2.9.1 Useful parameterizations for maximum likelihood estimation

To ensure that our parameter estimates are associated with a stationary and invertible model, we must ensure that $|d| < 0.5$, that Σ is positive definite, and that $A(L)$ has all of its roots outside the unit circle. The constraints on d can be implemented directly with box constraints. To ensure that Σ is positive definite, we follow the standard practice of constraining the diagonal element of its Cholesky decomposition to be positive. In the case where $A(L) = I - A_1 L$, $A(L)$ has all of its roots outside the unit circle if and only if all of the singular values of A_1 are less than one. In order for the covariance computation methods described in section 2.5 to work, we must bound the singular values away from one; if they approach one too closely, M in Algorithm 2.4 or 2.5 will tend towards infinity. In order to constrain the singular values of A_1, we use a modified version of the parameterization of Ansley and Kohn [1986], in which we ensure that the singular values of A_1 never exceed a given $\sigma < 1$ (0.99 in our algorithm). This parameterization results in a matrix, P, which is unconstrained and which can be mapped one-to-one onto the space of matrices with singular values less than σ. The algorithms to reparameterize A_1 and to return it to its original form are given below:

Algorithm 2.11 Conversion of a matrix, A_1, to the Ansley-Kohn parameterization, with maximum singular value, σ.

1. *Compute* $\tilde{A} = \frac{1}{\sigma} A_1$.
2. *Set* B *equal to the Cholesky decomposition of* $I_K - \tilde{A}\tilde{A}^T$, *where* I_K *is the identity matrix of size* K.
3. *Return* $(B^{-1})^T P_1$.

Algorithm 2.12 Conversion of a matrix, P, from the Ansley-Kohn parameterization with maximum singular value σ to its original form.

1. *Set B equal to the Cholesky decomposition of $I_K + PP^T$, where I_K is the identity matrix of size K.*
2. *Set $\tilde{A} = (B^{-1})^T P$.*
3. *Return σP.*

In Ansley and Kohn's original paper, they set $\sigma = 1$; Algorithms 2.11 and 2.12 reduce to their algorithm in that case. Given these parameterizations, we may implement maximum likelihood using simple box constraints.

2.9.2 The effects of the determinant approximation

We begin by studying the effects of the determinant approximation on the computed parameter estimates. In this section, we will compare three estimation methods: exact maximum likelihood using Sowell's algorithm, maximum likelihood in which the determinant is approximated in the most naive way by $|\Sigma|^T$, and maximum likelihood using the regression approximation to the determinant presented in Algorithm 2.8. To compare the three estimation methods, we simulate datasets of length $T = 100$ and 200 from FIVAR and VARFI processes with parameters $d = (0.1, 0.4)$, $\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$, and $A_1 = \begin{pmatrix} 0.6 & -0.1 \\ 0.2 & 0.8 \end{pmatrix}$. Because of the processing time required to compute the maximum likelihood estimates using Sowell's algorithm, all of our results are based on 100 simulated datasets.

In Tables 13, 14, 15, 16, and 17, we report the mean and standard deviation of the difference between the estimated parameter values using each approximation

method and the estimated values using exact maximum likelihood. If our approximations were exact, then all of the means and standard deviations would be 0. For the regression approximation for FIVAR models, the mean difference never exceeds 0.003 in absolute value, and the standard deviation of the difference exceeds 0.01 only once. In contrast, the means and standard deviations of the differences between the parameter estimates using the naive approximation and the parameter estimates from exact maximum likelihood are quite large, especially for the estimates of the elements of Σ. For a more graphical illustration, in Figure 14, we show boxplots of the differences in the estimates of d for a FIVAR process with $T = 100$. The boxplots confirm that the regression approximation estimates deviate slightly from the estimates from exact maximum likelihood, while the naive approximation estimates often differ dramatically from the exact maximum likelihood estimates. For VARFI models, our regression approximation again does well, though some of the standard deviations of the differences are higher for the estimates of the elements of Σ. As before, the naive approximation is a much less successful approximation, though its problems in estimating Σ are less marked than for FIVAR models. These results provide further evidence that our regression approximation to the determinant works well and that the traditional approximation of $|\Omega|$ by $|\Sigma|^T$ is not a close enough approximation.

2.9.3 Comparing maximum likelihood estimation to the Whittle estimator

We now run a larger Monte Carlo in which we compare the performance of our maximum likelihood estimates with the determinant approximation to the performance of the Whittle estimator. We will test these methods on both FIVAR and VARFI processes with a variety of sample sizes and parameter configurations. Here, we

Parameter	Regression Approximation	Naive Approximation
A_{11}	-0.0018 (0.0042)	-0.1221 (0.3771)
A_{21}	-0.0001 (0.0021)	-0.0869 (0.4036)
A_{12}	0.0022 (0.0068)	0.0674 (0.3987)
A_{22}	0.0010 (0.0057)	-0.2130 (0.4312)
Σ_{11}	0.0002 (0.0015)	656.6 (2189.6)
Σ_{12}	-0.0002 (0.0078)	-334.9 (37447.6)
Σ_{22}	0.0007 (0.0078)	531204.2 (1885517)
d_1	0.0016 (0.0107)	-0.1328 (0.2876)
d_2	-0.0008 (0.0069)	0.0797 (0.0560)
Log likelihood	-0.0213 (0.1114)	38.65 (60.47)

Table 13: Mean and standard deviation of the difference between the parameter estimate using the determinant approximation and the parameter estimate from exact maximum likelihood for a FIVAR model with $T = 100$. Standard deviations are given in parentheses. Estimates based on 100 repetitions.

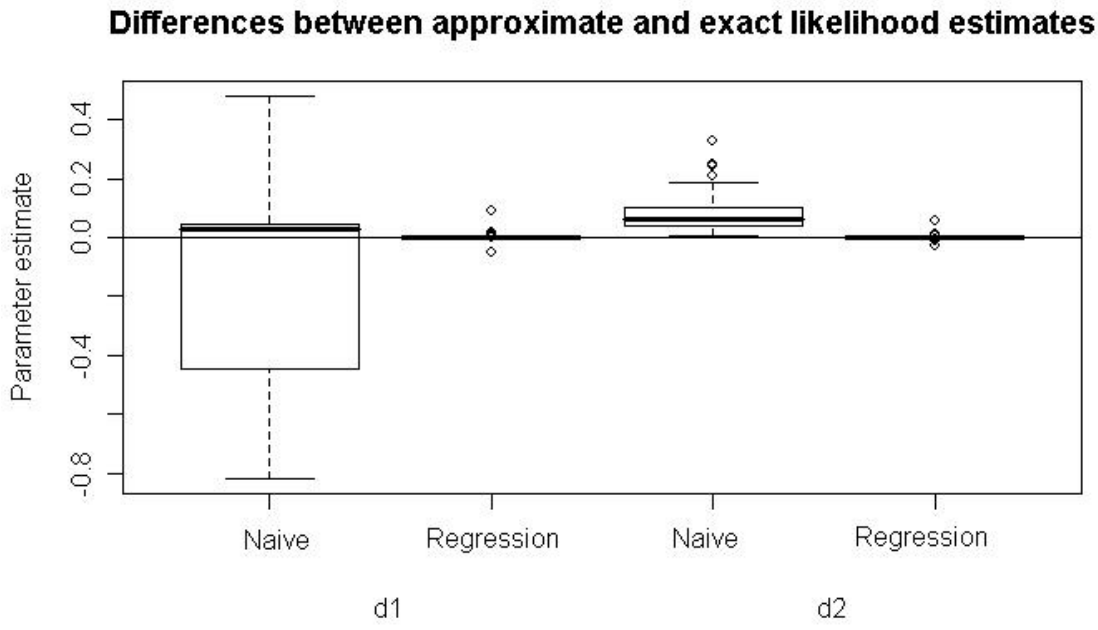

Figure 14: Boxplots of the differences between Sowell's exact maximum likelihood estimates and the two approximations in the estimates for d.

Parameter	Regression Approximation	Naive Approximation
A_{11}	-0.0016 (0.0029)	-0.1268 (0.3713)
A_{21}	0.0000 (0.0002)	-0.1202 (0.3367)
A_{12}	0.0017 (0.0020)	0.0290 (0.3319)
A_{22}	0.0012 (0.0026)	-0.1676 (0.3951)
Σ_{11}	0.0001 (0.0002)	294.635 (1306.914)
Σ_{12}	-0.0001 (0.0003)	8157.117 (129832.9)
Σ_{22}	0.0000 (0.0004)	3884703 (24760393)
d_1	0.0001 (0.0024)	-0.1213 (0.2880)
d_2	-0.0010 (0.0022)	0.0596 (0.0436)

Table 14: Mean and standard deviation of the difference between the parameter estimate using the determinant approximation and the parameter estimate from exact maximum likelihood for a FIVAR model with $T = 200$. Standard deviations are given in parentheses. Estimates based on 100 repetitions.

Parameter	Regression Approximation	Naive Approximation
A_{11}	-0.0017 (0.0122)	-0.1924 (0.1458)
A_{21}	-0.0000 (0.0055)	0.0858 (0.1220)
A_{12}	0.0001 (0.0033)	0.0934 (0.1280)
A_{22}	0.0003 (0.0038)	-0.3252 (0.1331)
Σ_{11}	0.0018 (0.0238)	-1.4827 (0.5943)
Σ_{12}	-0.00058 (0.0266)	0.8976 (0.4249)
Σ_{22}	0.0012 (0.0158)	99.05134 (1000.237)
d_1	-0.0014 (0.0257)	-0.0571 (0.1656)
d_2	-0.0004 (0.0081)	-0.0297 (0.1449)

Table 15: Mean and standard deviation of the difference between the parameter estimate using the determinant approximation and the parameter estimate from exact maximum likelihood for a VARFI model with $T = 100$. Standard deviations are given in parentheses. Estimates based on 100 repetitions.

Parameter	Regression Approximation	Naive Approximation
A_{11}	0.0004 (0.0078)	-0.1904 (0.1349)
A_{21}	-0.0003 (0.0023)	0.0912 (0.0804)
A_{12}	0.0002 (0.0025)	0.0910 (0.1099)
A_{22}	-0.0003 (0.0069)	-0.3219 (0.1300)
Σ_{11}	-0.0008 (0.0091)	1.5684 (30.0022)
Σ_{12}	0.0054 (0.1151)	-13.1160 (137.6588)
Σ_{22}	-0.0036 (0.0861)	63.4455 (629.2669)
d_1	-0.0005 (0.0090)	-0.0870 (0.1863)
d_2	0.0009 (0.0054)	-0.0768 (0.1283)

Table 16: Mean and standard deviation of the difference between the parameter estimate using the determinant approximation and the parameter estimate from exact maximum likelihood for a VARFI model with $T = 200$. Standard deviations are given in parentheses. Estimates based on 100 repetitions.

Parameter	Regression Approximation	Naive Approximation
A_{11}	-0.0016 (0.0221)	-0.1999 (0.0897)
A_{21}	0.0018 (0.0119)	0.0918 (0.0434)
A_{12}	-0.0006 (0.0120)	0.0817 (0.1209)
A_{22}	0.0003 (0.0125)	-0.3481 (0.0971)
Σ_{11}	0.0044 (0.0450)	-1.5867 (0.2511)
Σ_{12}	-0.0010 (0.0332)	0.9767 (0.2225)
Σ_{22}	0.0005 (0.0191)	4.2831 (52.8383)
d_1	0.0013 (0.0197)	-0.1048 (0.1238)
d_2	-0.0000 (0.0077)	-0.0905 (0.1100)

Table 17: Mean and standard deviation of the difference between the parameter estimate using the determinant approximation and the parameter estimate from exact maximum likelihood for a VARFI model with $T = 400$. Standard deviations are given in parentheses. Estimates based on 100 repetitions.

Model	Sowell Time	Regression Approximation Time	Naive Approximation Time
VARFI, $T = 100$	599.386	204.838	43.614
VARFI, $T = 200$	1977.928	390.172	77.807
VARFI, $T = 400$	8694.996	694.9787	131.9069

Table 18: Average processing time needed to compute the maximum likelihood estimators for each algorithm, for a variety of models.

T	MLE With Regression Approximation	Whittle
50	(0.156, 0.106)	(0.318, 0.152)
100	(0.150, 0.076)	(0.235, 0.135)
200	(0.149, 0.086)	(0.234, 0.135)

Table 19: Root mean squared errors of d estimates from a FIVAR model, based on 500 replications.

report preliminary results from simulations with $T = 50, 100$, and 200, $K = 2$, and parameters $d = (0.1, 0.4)$, $\Sigma = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}$, and $A_1 = \begin{pmatrix} 0.6 & -0.1 \\ 0.2 & 0.8 \end{pmatrix}$. All of these estimates are based on 500 simulated datasets.

In Tables 19, 20, and 21, we report the root mean squared error of each parameter estimate for each estimation method. These results show that maximum likelihood using the regression approximation performs the best in estimating both d and Σ, but the Whittle estimator does better in estimating the element of A_1, particularly the off-diagonal elements. Furthermore, we see from these results that the root mean squared error seems to be decreasing slowly as the sample size increases.

We now repeat the experiment with VARFI models. The results are given in Tables 23, 24, and 25. As in the FIVAR models, the Whittle estimator has the lower root mean squared error for some parameter estimates while our maximum likelihood estimator has lower root mean squared errors for others. For a number of parameters, such as the elements of d and A_1, the Whittle estimator performs better in the smallest sample, but the maximum likelihood estimator has a smaller RMSE for larger samples. Oddly, the Whittle estimator is dramatically better for one of the diagonal entries of Σ while the maximum likelihood estimator is dramatically better for the other. Examination of the mean estimates (not reported)

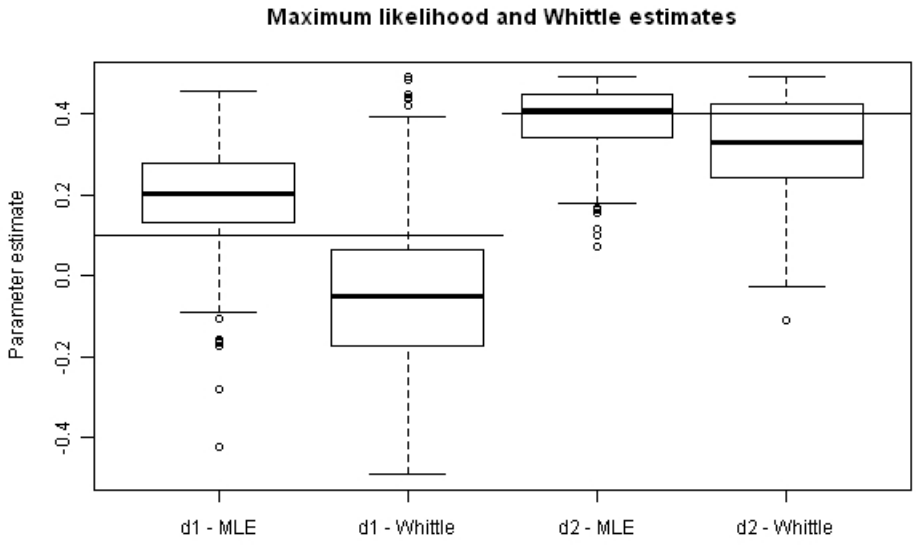

Figure 15: Boxplot of the estimated values of d, using maximum likelihood with the regression approximation and the Whittle estimator. The true values are 0.1 for d_1 and 0.4 for d_2.

T	MLE With Regression Approximation	Whittle
50	(0.213, 0.522, 0.522, 0.537)	(0.840, 0.423, 0.423, 1.494)
100	(0.158, 0.507, 0.507, 0.459)	(0.843, 0.423, 0.423, 1.584)
200	(0.158, 0.508, 0.508, 0.459)	(0.840, 0.422, 0.422, 1.580)

Table 20: Root mean squared errors of Σ estimates from a FIVAR model, based on 500 replications.

T	MLE With Regression Approximation	Whittle
50	(0.214, 0.159, 0.717, 0.127)	(0.160, 0.069, 0.199, 0.098)
100	(0.186, 0.146, 0.713, 0.093)	(0.158, 0.055, 0.137, 0.096)
200	(0.185, 0.145, 0.713, 0.092)	(0.158, 0.053, 0.138, 0.097)

Table 21: Root mean squared errors of A_1 estimates from a FIVAR model, based on 500 replications.

T	MLE with Regression Approximation	Whittle
50	137.1381	33.971
100	209.4684	73.24466
200	416.3902	163.9263

Table 22: Average processing time needed for estimation of a VARFI model over 500 repetitions.

T	MLE With Regression Approximation	Whittle
50	(0.217, 0.228)	(0.197, 0.167)
100	(0.210, 0.096)	(0.190, 0.132)
200	(0.194, 0.059)	(0.211, 0.106)

Table 23: Root mean squared errors of d estimates from a VARFI model, based on 500 replications.

shows that the Whittle estimates of the elements of Σ are biased toward zero, while the maximum likelihood estimates of the diagonal elements have an upward bias. Thus, the Whittle estimator fares better when for the smaller diagonal element, while the maximum likelihood estimator is more successful for the larger diagonal element.

Using a more extensive set of simulations, with a variety of parameter values for d and A_1, we find that the estimates of d using maximum likelihood with the regression approximation generally have smaller root mean squared errors than those from the Whittle estimator. As before, we found that the estimates of Σ from the Whittle estimator were biased toward 0, with estimates of the diagonal elements of Σ equal to 18% of their true values on average. In contrast, the estimates using maximum likelihood with the regression approximation had bias under 0.1 in most cases and root mean squared errors under 0.2. Results for

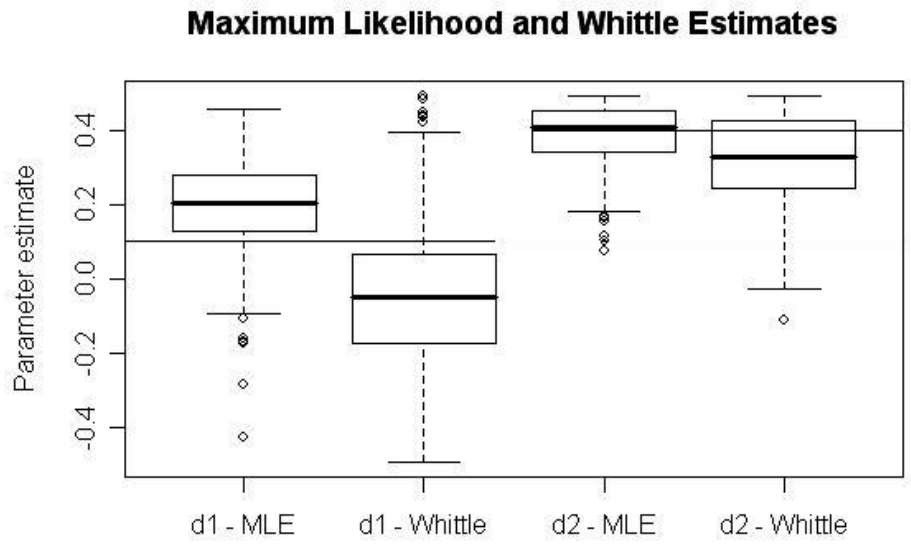

Figure 16: Boxplot of the estimated values of d, using maximum likelihood with the regression approximation and the Whittle estimator. The true values are 0.1 for d_1 and 0.4 for d_2.

T	MLE With Regression Approximation	Whittle
50	(1.467, 1.481, 1.481, 0.430)	(0.539, 0.779, 0.779, 1.505)
100	(1.521, 1.505, 1.505, 0.303)	(0.567, 0.715, 0.715, 1.588)
200	(1.520, 1.501, 1.501, 0.224)	(0.581, 0.690, 0.690, 1.626)

Table 24: Root mean squared errors of Σ estimates from a VARFI model, based on 500 replications.

T	MLE With Regression Approximation	Whittle
50	(0.162, 0.188, 0.250, 0.113)	(0.149, 0.134, 0.256, 0.105)
100	(0.134, 0.092, 0.221, 0.086)	(0.143, 0.093, 0.225, 0.097)
200	(0.099, 0.070, 0.211, 0.063)	(0.127, 0.069, 0.205, 0.080)

Table 25: Root mean squared errors of A_1 estimates from a VARFI model, based on 500 replications.

estimates of A_1 were mixed in terms of bias and RMSE. The Whittle estimator generally had lower bias and RMSE for the off-diagonal elements of A_1, while the two estimators were evenly matched on the diagonal elements. Overall, we find that maximum likelihood with the regression approximation performs better, though computing estimates from both estimators could be helpful in some applications.

2.10 Data Analysis

In this section, we apply FIVAR and VARFI models to three different datasets. First, we apply our models to the components of inflation. Second, we discuss an application to a macroeconomic model of unemployment and inflation. Finally, we discuss an application in meteorology.

2.10.1 Goods and Services Inflation

We now consider a model for inflation in the goods and services sectors. While inflation is often considered as a single number, it is actually composed of the price changes across all goods and services produced in the economy. The relationship of the inflation rates across different sectors can be helpful for predicting inflation and for understanding how price changes in one sector affect price changes in other parts of the economy. Peach et al. [2004] modeled inflation in the goods and services sectors, excluding food and energy, as cointegrated time series, without allowing for fractional differencing. In this section, we estimate FIVAR and VARFI models based on overall goods and services inflation, as measured by the Consumer Price Index, for the period February 1956 through January 2008. The data are available online from the Bureau of Labor Statistics. The data are show in Figure 17.

We first fit univariate $ARFIMA(1, d, 0)$ models to the two series using maximum likelihood. The estimates are given in Table 26. According to these esti-

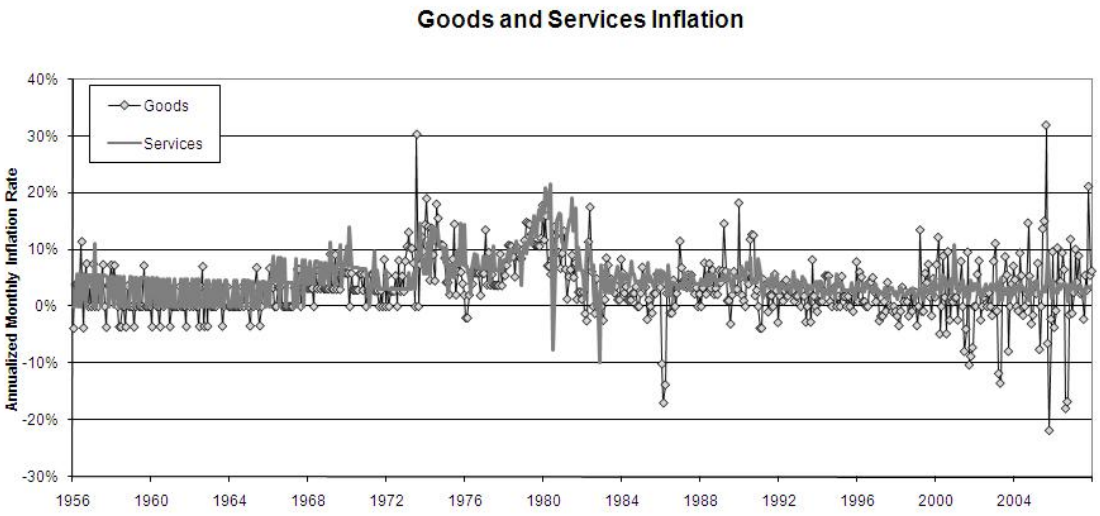

Figure 17: Annualized goods and services inflation rates, February 1956-January 2008.

mates, both series are fractionally integrated. Goods inflation is estimated to have a differencing parameter of 0.2265, while the differencing parameter of services inflation is estimated to be 0.4837, making it almost non-stationary. We use these estimates as starting values for our estimation of FIVAR models, setting all initial off-diagonal elements of A_1 and Σ to 0.

We estimate a FIVAR model based on the demeaned data, using both maximum likelihood and the Whittle estimator. Results are reported in Tables 27. As we found in the Monte Carlo simulations, the estimates of the covariance matrix based on Whittle estimator are much closer to 0 than the estimates from maximum likelihood are. Both estimators find that services inflation has a larger differencing parameter than goods inflation, with the services differencing parameter quite close to 0.5. In Figures 20 and 21, we plot the logged modulus of the cross-periodogram

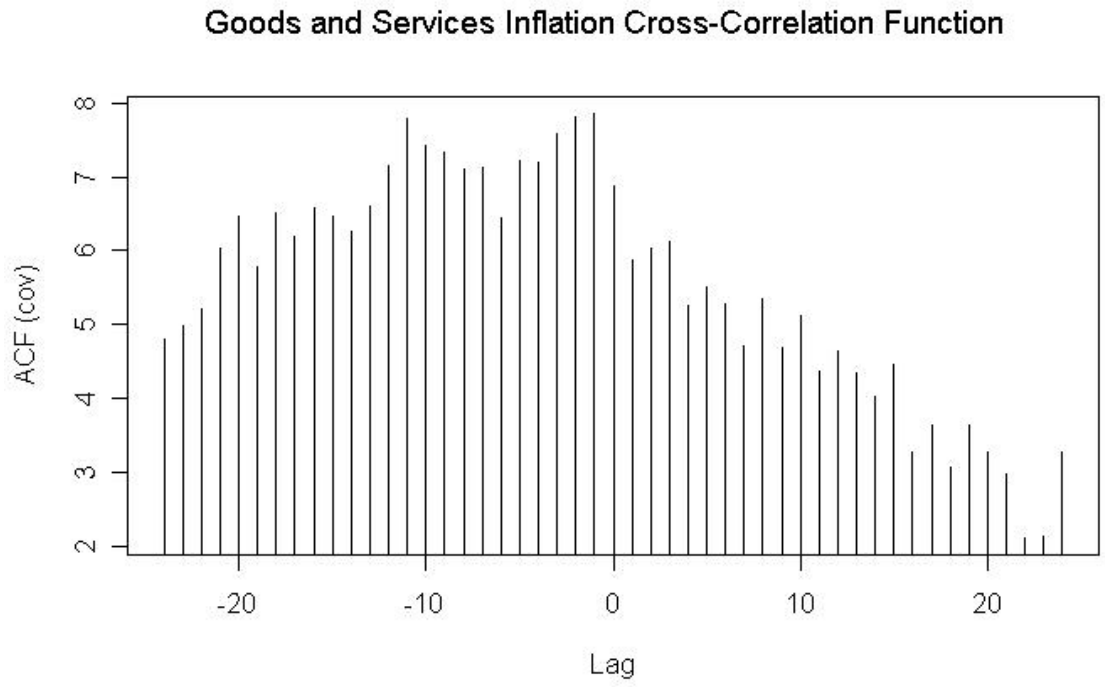

Figure 18: Empirical cross-correlation function of goods and services inflation rates.

	Goods	Services
A_1	0.1053 (0.0032)	-0.3165 (0.0013)
Σ	21.2703 (1.4529)	7.0842 (0.1523)
d	0.2265 (0.0016)	0.4837 (0.0000)
Log Likelihood	-1266.140	-924.7657

Table 26: Maximum likelihood estimates for goods and services inflation, as univariate series. Approximate asymptotic standard errors in parentheses.

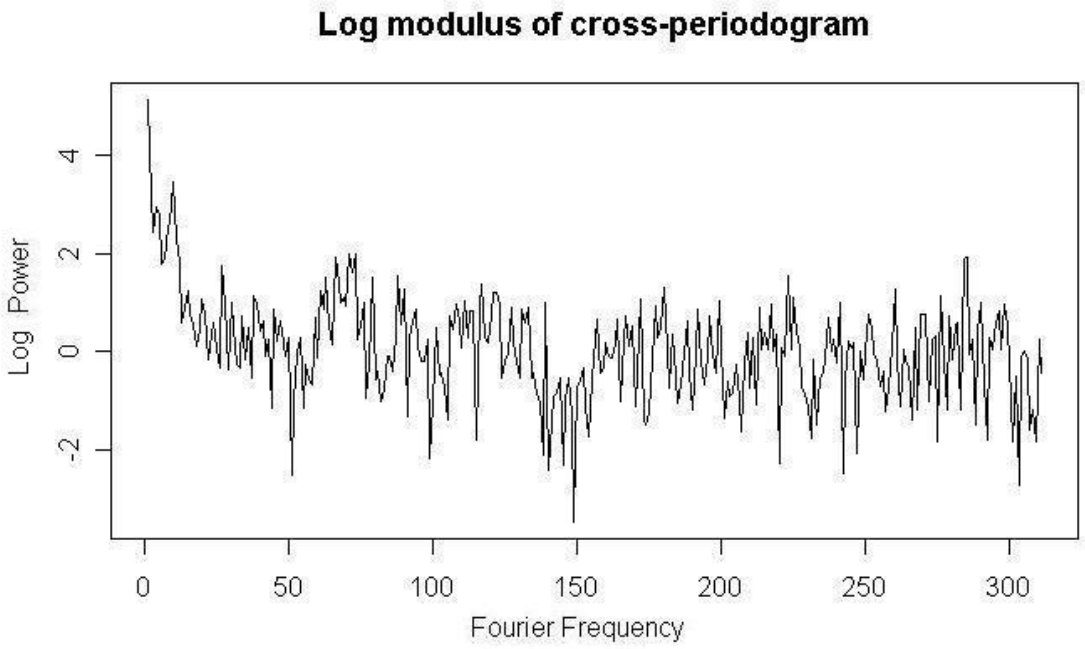

Figure 19: Log modulus of the cross-periodogram of goods and services inflation rates.

	Maximum Likelihood with Regression Approximation	Exact Maximum Likelihood	Whittle Approximation
A_{11}	0.1024 (0.0034)	0.1023	0.1865 (0.0041)
A_{21}	-0.0204 (0.0006)	-0.0204	0.0993 (0.0005)
A_{12}	0.1510 (0.0104)	0.1509	-0.0053 (0.0045)
A_{22}	-0.3101 (0.0022)	-0.3103	-0.3354 (0.0025)
Σ_{11}	21.0912 (0.0003)	21.0909	3.3864 (0.0371)
Σ_{12}	0.6260 (0.3120)	0.6257	0.1358 (0.0062)
Σ_{22}	7.0812 (0.0763)	7.0804	1.0947 (0.0039)
d_1	0.2281 (0.0017)	0.2282	0.1410 (0.0020)
d_2	0.4770 (0.0006)	0.4771	0.4875 (0.0018)
Log likelihood	-2187.109	-21887.095	-4501.032

Table 27: FIVAR estimates for goods and services inflation data. The Whittle log likelihood is the regression approximation to the likelihood at those parameter values. Approximate asymptotic standard errors are given in parentheses for all estimators except for Sowell's exact estimator.

and the logged modulus of the implied cross-spectral densities based on the two estimators. The spectral density based on the maximum likelihood estimates seems to fit the cross-periodogram more closely.

We now fit a VARFI model to this data, again using both estimators. The estimated covariance matrices are similar, but the maximum likelihood estimate of the smaller differencing parameter has dropped from 0.22 to 0. This does not mean that goods inflation is now estimated to have short memory; on the contrary, under the VARFI model the two series are estimated to have the same memory

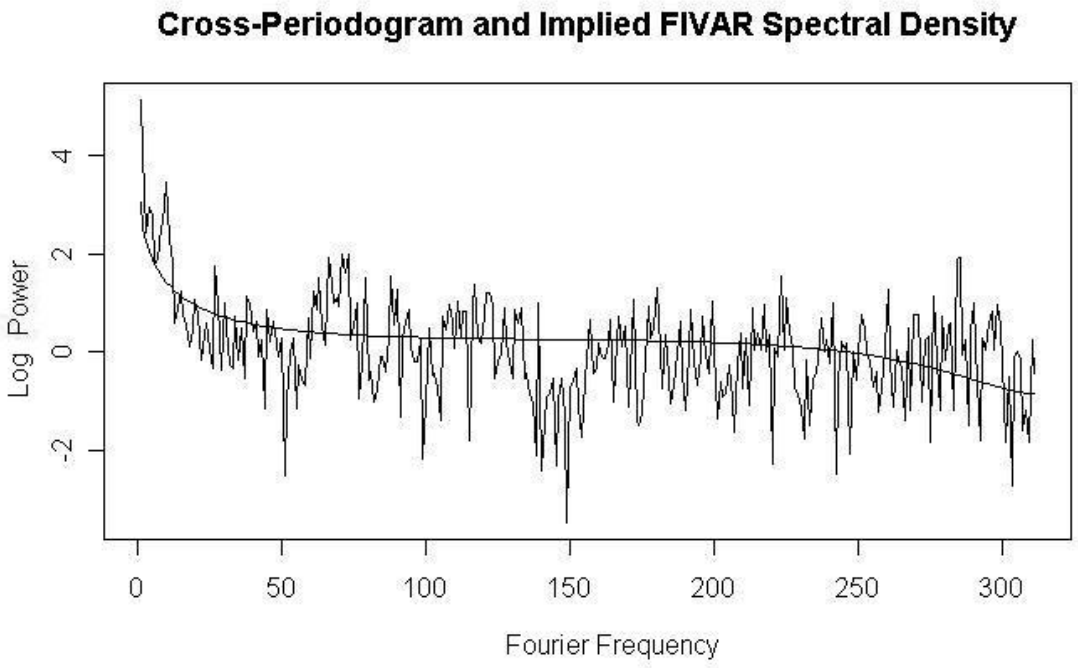

Figure 20: Log modulus of the cross-periodogram of goods and services inflation rates and of the implied cross-spectral density of the estimated FIVAR model.

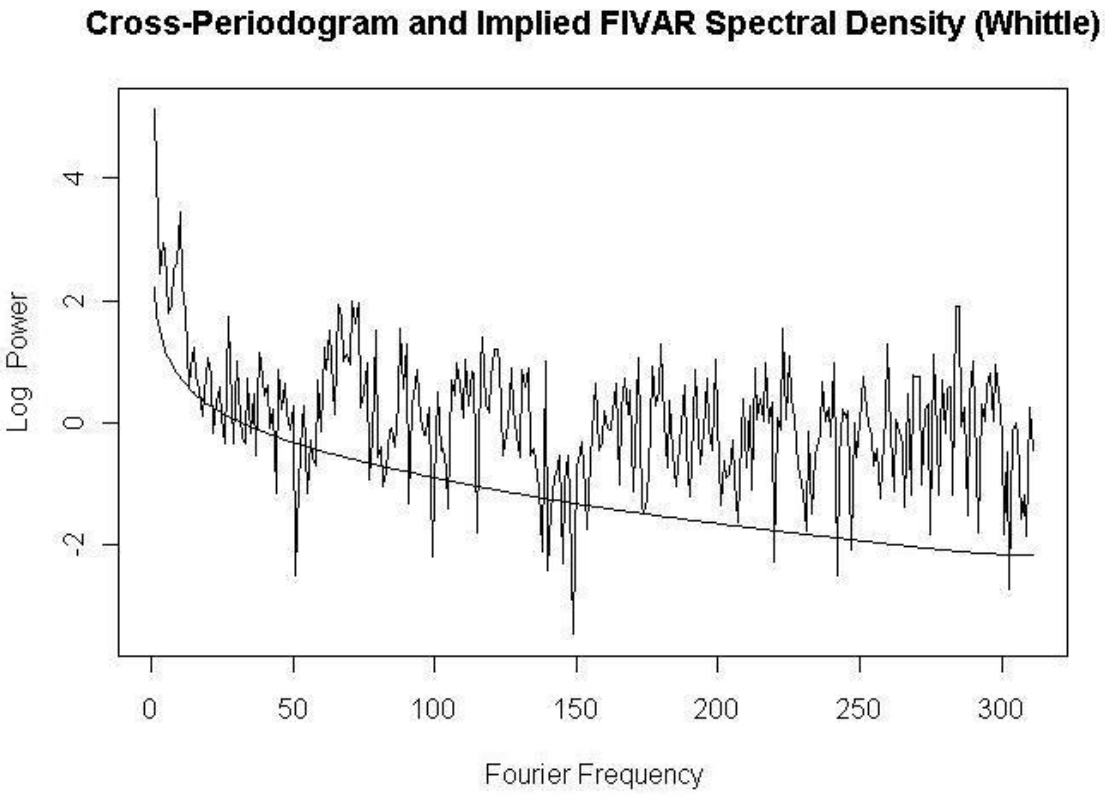

Figure 21: Log modulus of the cross-periodogram of goods and services inflation rates and of the implied cross-spectral density of the estimated FIVAR model, using the Whittle estimator.

	Maximum Likelihood with Regression Approximation	Exact Maximum Likelihood	Whittle Approximation
A_{11}	0.3027 (0.0014)	0.3027	0.1613 (0.0048)
A_{21}	-0.0237 (0.0005)	-0.0237	0.0544 (0.0005)
A_{12}	0.4245 (0.0027)	0.4245	0.0881 (0.0065)
A_{22}	-0.3085 (0.0018)	-0.3085	-0.3211 (0.0026)
Σ_{11}	20.2342 (0.8669)	20.2342	3.3736 (0.0366)
Σ_{12}	0.4605 (0.2275)	0.4605	0.1313 (0.0065)
Σ_{22}	7.0783 (0.1619)	7.0783	1.1178 (0.0040)
d_1	0.0000 (0.0000)	0.0000	0.1512 (0.0034)
d_2	0.4835 (0.0004)	0.4835	0.4890 (0.0018)
Log likelihood	-2174.263	-2174.249	-4372.655

Table 28: VARFI estimates for goods and services inflation data. The Whittle log likelihood is the regression approximation to the likelihood at those parameter values. Approximate asymptotic standard errors are given in parentheses for all estimators except for Sowell's exact estimator.

parameter. In contrast, the Whittle estimates of d are almost unchanged. As before, we compare the cross-periodogram to the implied cross-spectral densities from the two estimates in Figures 22 and 23.

Since the $FIVAR(1, \vec{d})$ and $VARFI(1, \vec{d})$ have the same number of parameters, we may compare their log likelihoods to choose between them. In this case, the VARFI model has a higher log likelihood. We may write the VARFI model in a form analogous to a VAR, where the errors driving the VAR are no longer white

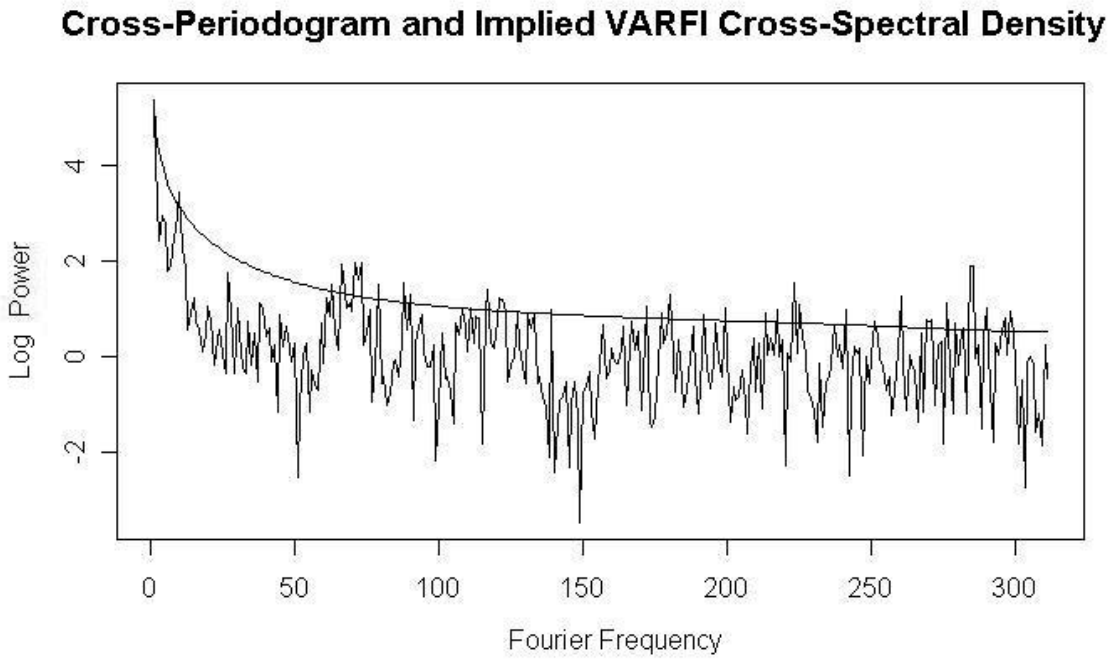

Figure 22: Log modulus of the cross-periodogram of goods and services inflation rates and of the implied cross-spectral density of the estimated VARFI model.

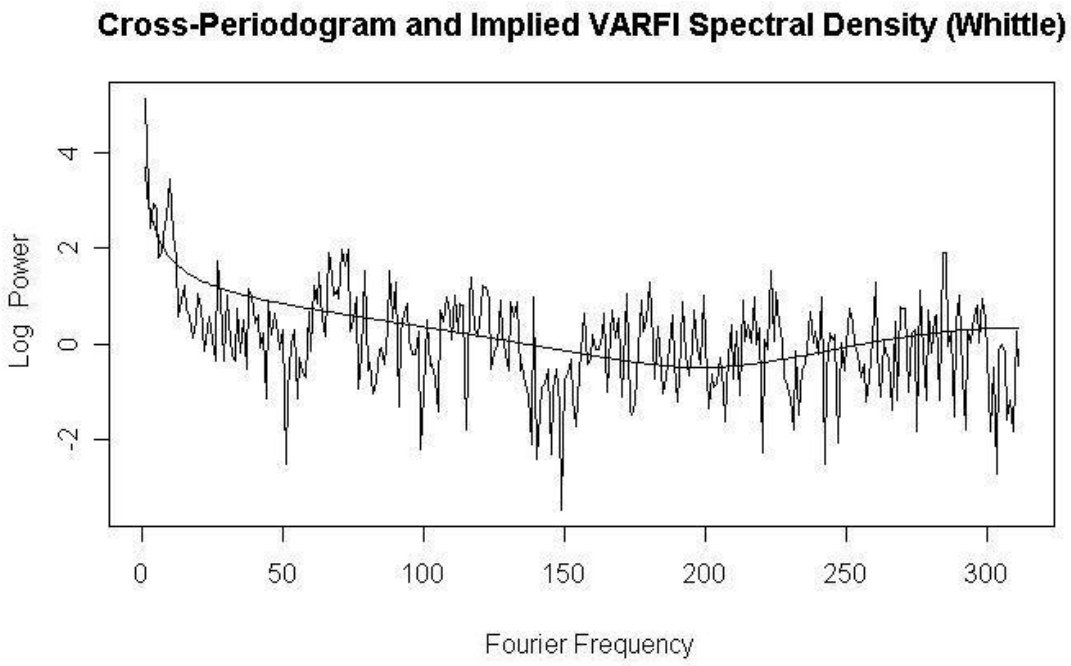

Figure 23: Log modulus of the cross-periodogram of goods and services inflation rates and of the implied cross-spectral density of the estimated VARFI model.

noise:

$$\begin{aligned} goods_t &= 0.3027 goods_{t-1} + 0.4245 services_{t-1} + u_{1t} \\ services_t &= -0.0237 goods_{t-1} - 0.3085 services_{t-1} + u_{2t} \\ \begin{pmatrix} u_{1t} \\ (1-L)^{0.4835} u_{2t} \end{pmatrix} &\sim Normal\left(0, \begin{pmatrix} 20.2342 & 0.4605 \\ 0.4605 & 7.0783 \end{pmatrix}\right) \end{aligned}$$

Though the goods equation is driven by shocks that have short memory, long memory in goods inflation is induced by the lagged services inflation. In the services equation, lagged services inflation has a negative coefficient; however, services inflation is persistent because of the persistence in the shock process. While lagged services inflation has a significant influence on goods inflation, lagged goods inflation has little effect on services inflation.

Rewriting the first equation in the VARFI model, we find that $w_t = goods_t - 0.3027 goods_{t-1} - 0.4245 services_{t-1}$ is estimated to be white noise. To confirm this, we compute w_t over the sample period and plot it in Figure 24. This series appears to be approximately white noise, though there are some periods of increased volatility, particularly near the end of the sample period. The log periodogram, shown in Figure 25, confirms that all long memory has been removed by this linear combination.

For comparison, we also fit a short memory vector autoregressive model to the data. We consider two lag lengths. First, we use a $VAR(2)$, since that model has only two more parameters than a $FIVAR(1, \vec{d})$ or $VARFI(1, \vec{d})$ model does. Second, we use the AIC to choose a lag length, and a vector autoregressive model with 10 lags is chosen. We plot the log modulus of the cross-periodogram and the log modulus of the spectral density implied by the estimates of these two models in Figures 26 and 27. When only two lags are included, the fact that the spectral density is finite at 0 is quite evident; the model cannot match the peak in the

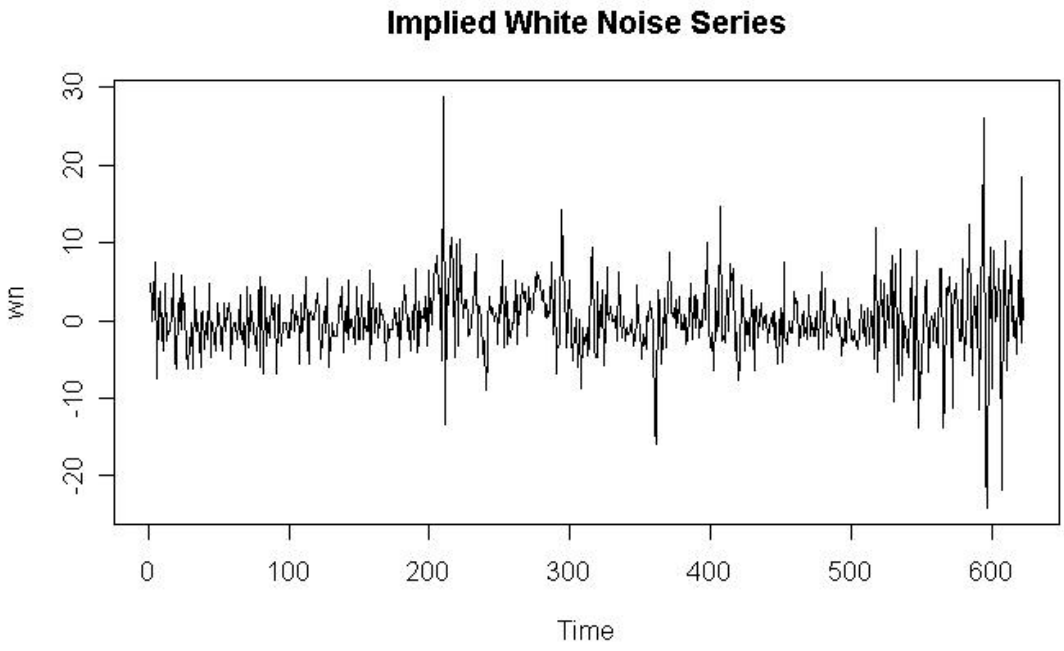

Figure 24: Time series of the linear combination of lagged goods and services inflation that the VARFI model implies is white noise.

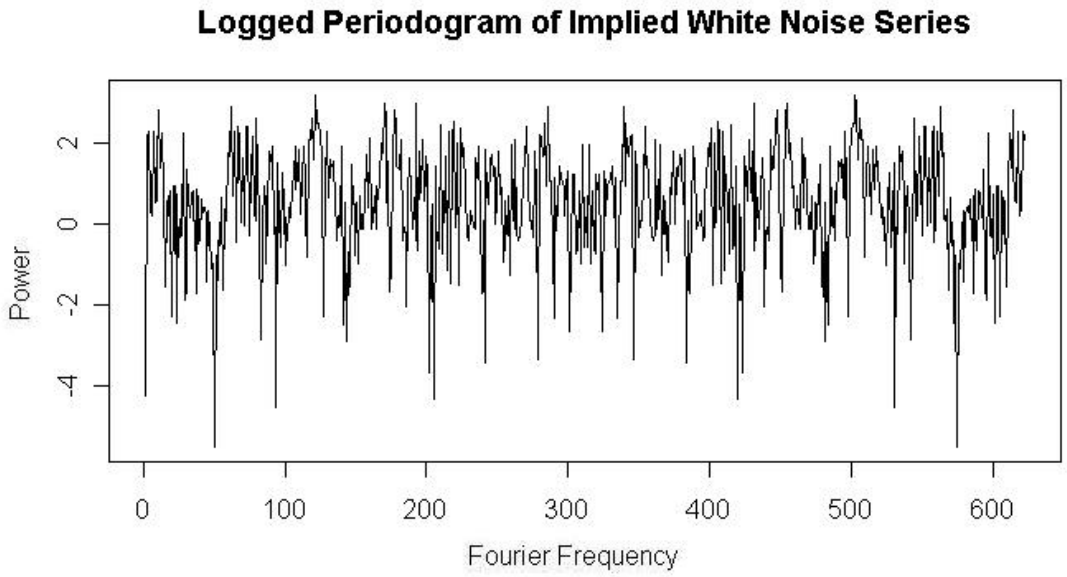

Figure 25: Log periodogram of the linear combination of lagged goods and services inflation that the VARFI model implies is white noise.

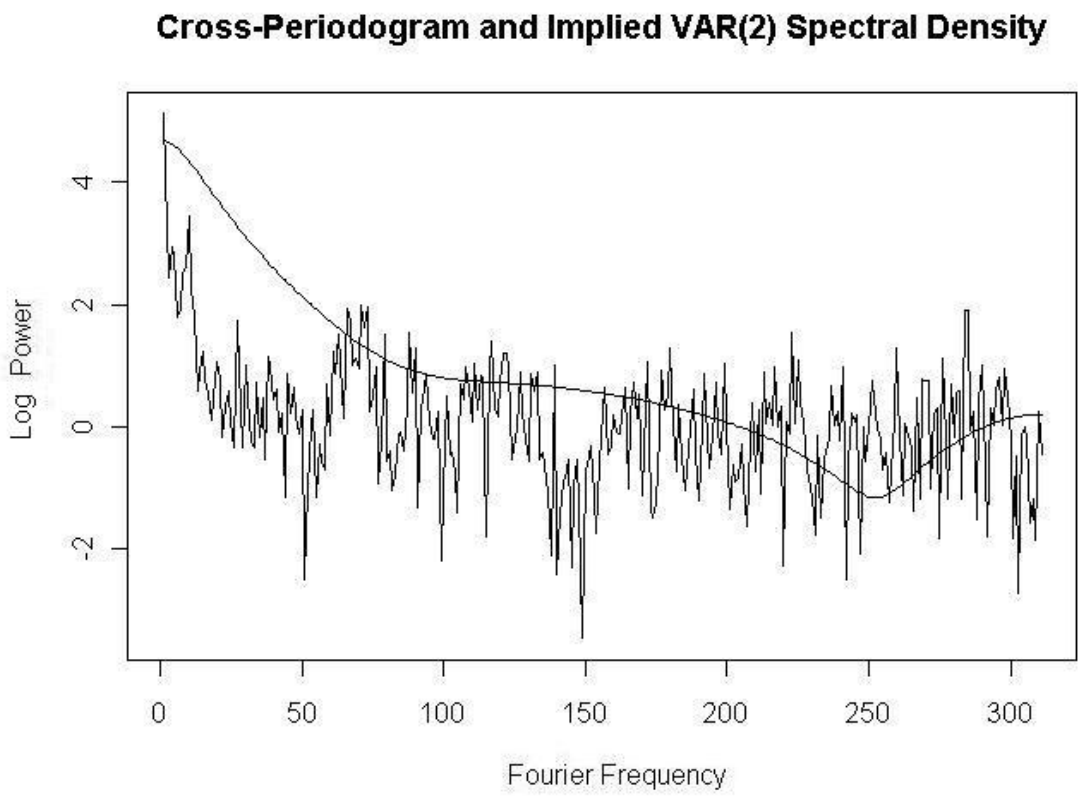

Figure 26: Log modulus of the cross-periodogram of goods and services inflation rates and of the implied cross-spectral density of the estimated VAR(2) model.

periodogram at 0. When 10 lags are included, the model fits the peak, but the spectral density is less smooth, suggesting overfitting.

As a final comparison among the models, we compute out-of-sample predictions for February through May 2008. For goods inflation, the $VAR(2)$ performed best; for services inflation, the VARFI model with the maximum likelihood estimates performs best. In both cases, the $VAR(10)$ was by far the worst performer. In Figure 28, we plot the forecasts and realization of services inflation. At the end of

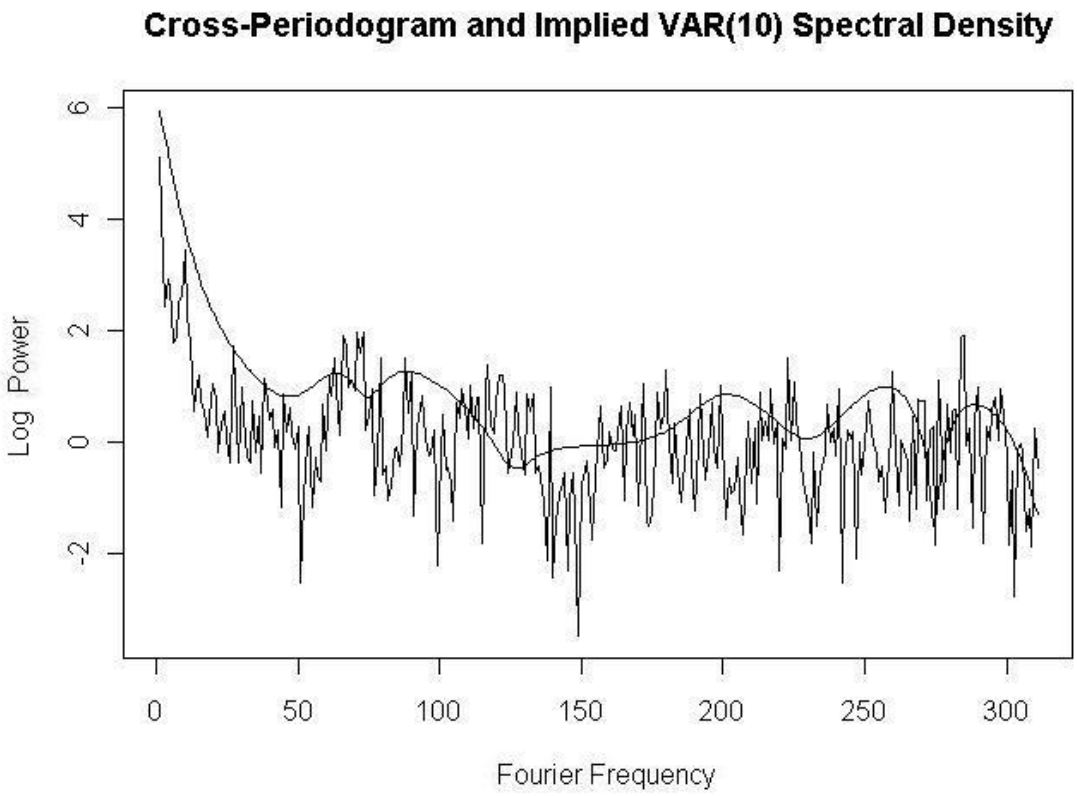

Figure 27: Log modulus of the cross-periodogram of goods and services inflation rates and of the implied cross-spectral density of the estimated VAR(10) model.

	Goods	Services
FIVAR-MLE	5.264	1.236
FIVAR-Whittle	5.189	1.256
VARFI-MLE	5.211	1.215
VARFI-Whittle	5.194	1.280
VAR(2)	5.008	1.327
VAR(10)	6.263	2.232

Table 29: Root mean squared errors for out-of-sample from February to May 2008.

the sample, services inflation had been below its mean for 26 consecutive months. The long memory structure of the VARFI and FIVAR models could model this persistence, and predicted that inflation would move very slowly toward its mean. In contrast, the predictions based on the $VAR(2)$ returned to the mean at an exponential rate. This difference accounts for the improved performance of the long memory models for services inflation.

2.10.2 Phillips Curve Data

One of the most basic models in macroeconomics is the Phillips curve, which relates the unemployment rate to inflation. (See a macroeconomics textbook, such as Hall and Taylor [1997] for more background.) The simplest form of the Phillips curve states that an increase in the slack in the economy, as measured by the unemployment rate, leads to a decrease in inflation. Empirically, we see that inflation is generally persistent (see Figure 29); this is often explained in models by assuming that people have expectations about inflation, and that the effect of the unemployment rate on inflation is relative to the expectations. The simplest form of inflation expectations sets the expectation for tomorrow equal to today's

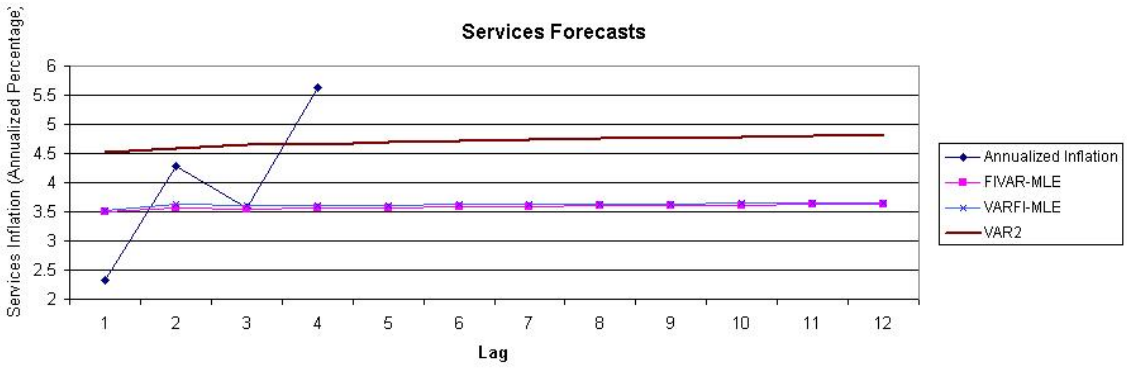

Figure 28: Realized out-of-sample services inflation and forecasts from the $VAR(2)$, VARFI and FIVAR models.

inflation (for example, Wooldridge, 2000, example 11.5). Such a model implies a relationship between the level of the unemployment rate and the first difference of the inflation rate; if the unemployment rate were constant, this would imply a unit root in inflation. However, as we see Figure 29, the unemployment rate is also persistent, while inflation is persistent but is also likely to be mean-reverting; this suggests that a multivariate long memory model might be a better description of the data. We will not justify the use of a FIVAR or VARFI model using economic theory, but only as a useful description of the data. In this estimation, we use annual data on the unemployment rate and the inflation rate from 1948 to 1996.[1] The estimated cross-correlation function for this data is given in Figure 30. This figure shows that past inflation is strongly correlated with the future unemployment rate, which runs counter to the usual understanding of the Phillips curve, in which the slack in the economy, as measured by the unemployment rate, would affect future inflation.

[1]This dataset is available from the website of Jeffrey Wooldridge at `http://www.msu.edu/ ec/faculty/wooldridge/book2.htm`, as `Phillips.RAW`.

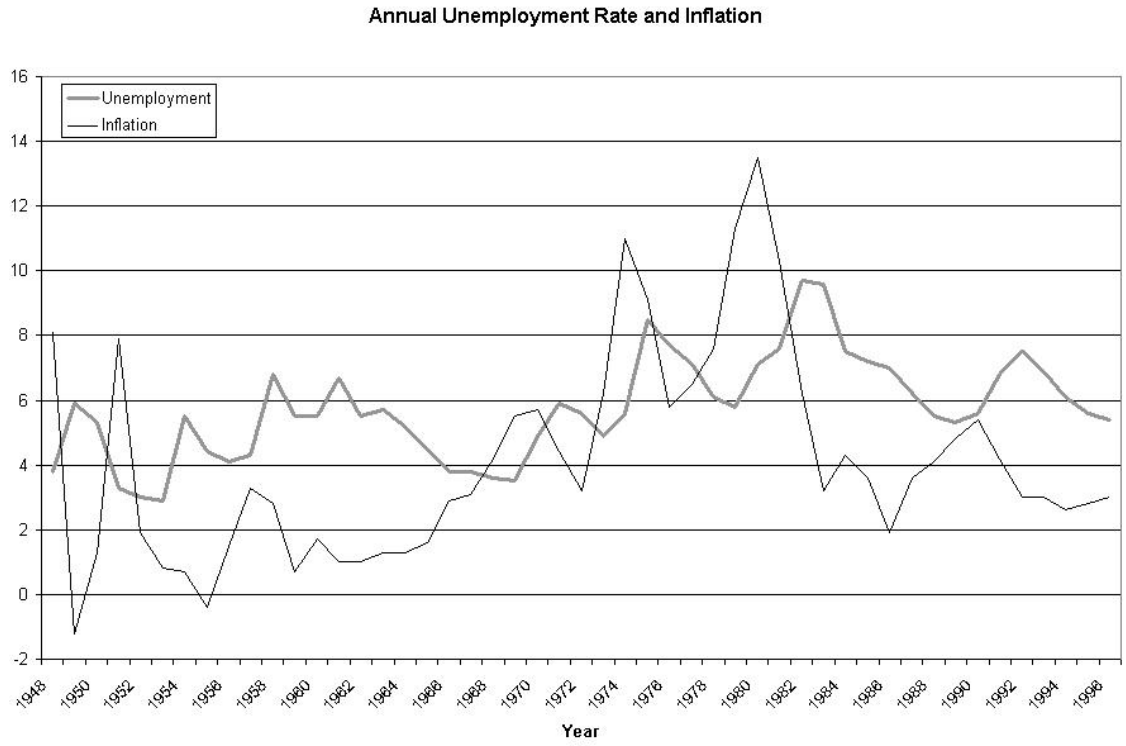

Figure 29: Annual unemployment rate and inflation rate used for estimating the Phillips curve.

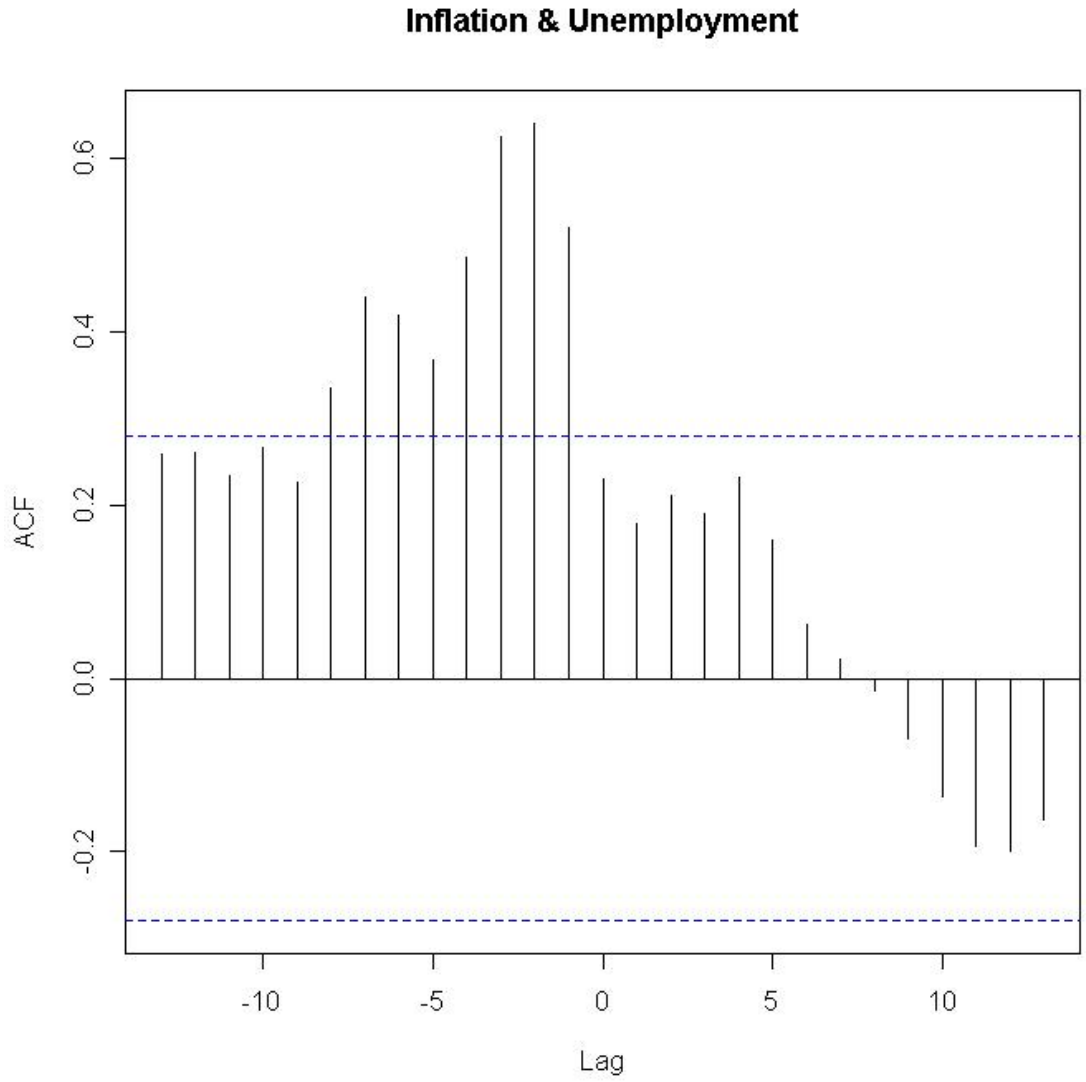

Figure 30: The empirical cross-correlation function of the unemployment rate and the inflation rate.

	Maximum Likelihood with Regression Approximation	Exact Maximum Likelihood	Whittle Approximation
A_1	(-0.1085, 0.0668, 0.9360, 0.3120)	(-0.1075, 0.0670, 0.9361, 0.3119)	(-0.1788, 0.2524, 0.3441, 0.4774)
Σ	(2.3052, -1.3910, -1.3910, 4.9402)	(2.3056, -1.3912, -1.3912, 4.9398)	(0.3000, -0.2941, -0.2941, 1.1282)
d	(0.3601, 0.3364)	(0.3595, 0.3365)	(0.4900, 0.0137)
Log likelihood	-105.3	-105.2991	-357.2535

Table 30: FIVAR estimates for Phillips curve data. The Whittle log likelihood is the regression approximation to the likelihood at those parameter values.

We first fit FIVAR models to these data using maximum likelihood with the regression approximation, exact maximum likelihood using Sowell's method, and Whittle's approximation to the likelihood. The estimated parameter values are given in Table 30. Using our default initial values, the estimates from Sowell's method failed to converge; when we used the FIVAR estimates as the initial values, the estimates converged to the values that we report. While the exact maximum likelihood estimate and the estimate from the regression approximation match quite closely, as we would expect from section 2.9.2, the Whittle estimate is very different, and the Whittle estimate for d_1 is on the boundary of the parameter space. In the maximum likelihood estimates, the estimated differencing parameters are quite close. Since the FIVAR and VARFI models are identical when the differencing parameters are equal, this suggests that the VARFI model will have similar parameter estimates.

	Maximum Likelihood with Regression Approximation	Exact Maximum Likelihood	Whittle Approximation
A_1	(-0.2228, 0.0449, 0.9020, 0.3601)	(-0.2226, 0.0449, 0.9020, 0.3601)	(-0.0110, 0.2807, 0.2600, 0.1255)
Σ	(2.2248, -1.4736, -1.4736, 4.9647)	(2.2252, -1.4741, -1.4741, 4.9643)	(0.3195, -0.3231, -0.3231, 1.1791)
d	(0.4480, 0.2402)	(0.4480, 0.2411)	(0.4677, 0.3831)
Log likelihood	-104.0927	-104.0907	-315.5599

Table 31: VARFI estimates for Phillips curve data. The Whittle log likelihood is the regression approximation to the likelihood at those parameter values.

To check this hypothesis, we estimate a VARFI model with the same data. The estimates are reported in Table 31; the estimates for both maximum likelihood methods use the FIVAR estimates as initial values. In this case, the Whittle estimates of the parameters are somewhat closer to the maximum likelihood estimates, though the estimates of the elements of Σ are still much closer to 0 than the maximum likelihood estimates of Σ.

We have estimated two distinct models based on the same data. Comparing the maximum likelihood estimates from the two models, we see that the VARFI estimates of the differencing parameters differ by more than the FIVAR estimates do, but that the averages of the estimated differencing parameters are almost identical (0.344 for the VARFI model and 0.348 for the FIVAR model). The estimates of the innovation variances match closely, while the estimates of the autoregressive parameters are of the same signs and similar magnitudes. Because

these two models have the same number of parameters, we may choose between them based on the log likelihoods. Using this criterion, we prefer the VARFI model to describe the relationship between the unemployment rate and inflation. In Figure 31, we plot the implied cross-covariances based on the VARFI model. The asymmetric pattern of slowly decaying cross-covariances is captured nicely by the VARFI model.

As we discussed in Section 2.2.2, a VARFI model is a vector autoregression driven by fractionally integrated white noise. That means that we may write this VARFI model as:

$$unemp(t) = 0.223unemp(t-1) - 0.045infl(t-1) + u_{1t} \quad (2.17)$$

$$infl(t) = -0.902unemp(t-1) - 0.360infl(t-1) + u_{2t} \quad (2.18)$$

where (u_{1t}, u_{2t}) are distributed as fractionally integrated white noise with covariance matrix $\begin{pmatrix} 2.225 & -1.473 \\ -1.473 & 4.965 \end{pmatrix}$ and differencing parameters $(0.448, 0.240)$. Equation 2.18 matches the traditional intuition about the Phillips curve: an increase in the unemployment rate is associated with a decrease in the inflation rate in the next period. We also find that an increase in inflation is associated with a decrease in the unemployment rate in the next period. In this model, though, the "shocks" are correlated across time, which leads to more persistence in both inflation and the unemployment rate, despite the negative coefficients on the AR(1) parameter in Equation 2.18. Thus, the VARFI model matches the basic economic theory of the Phillips curve but can also match the empirical persistence in the cross-covariances.

For comparison, we also fit a vector autoregressive model to this data. The Akaike Information Criterion suggests a lag length of 2 for this data. The parameter estimates for a vector autoregressive model of order 2 are given in Ta-

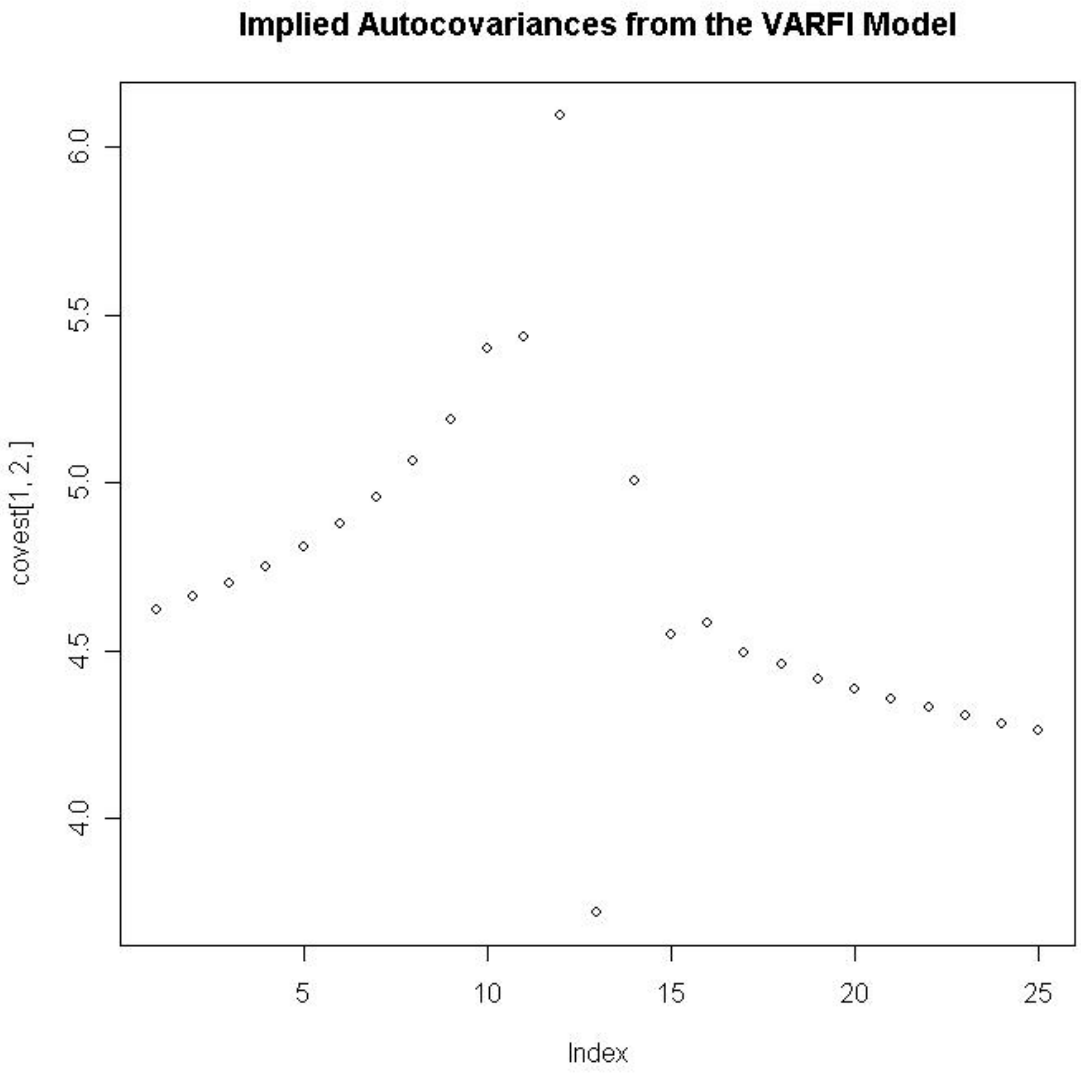

Figure 31: The cross-correlation function of the unemployment rate and the inflation rate implied by the VARFI model.

	Unemployment Rate	Inflation
Unemployment Rate - Lag 1	0.67779 (0.15544)	-0.5224 (0.3526)
Inflation - Lag 1	0.14147 (0.05766)	0.7737 (0.1308)
Unemployment Rate - Lag 2	-0.07806) (0.13205	0.5458 (0.2995)
Inflation - Lag 2	0.05758 (0.06735)	-0.0297 (0.1528)

Table 32: Parameter estimates from a VAR(2) model for the Phillips curve data. Standard errors are given in parentheses.

ble 32. The estimated covariance matrix for the innovations in this model is $\begin{pmatrix} 0.7929 & -0.3482 \\ -0.3482 & 4.0797 \end{pmatrix}$. Notice that the estimated covariance matrix entries lie between the maximum likelihood estimates and the Whittle estimates of Σ. While we cannot compare the VAR coefficients to the VARFI coefficients in the same way, we can compare the implied autocovariance functions. The cross-covariance function implied by the VAR(2) is given in Figure 32. In contrast to the covariances from the VARFI model, the covariances from the VAR model start lower and decay to 0 quite quickly. We also note that the conditional log likelihood of the VAR(2) is -154.783. Since this VAR has been estimated conditional on the first two periods, we must add on the likelihood of the initial observations in order to make the likelihood comparable to the unconditional likelihood given in Table 31. Using the unconditional covariances of a VAR(2), we find that the likelihood of the first two observations is -22.5925. Summing the two parts of the log likelihood, we find that the VAR model has a log likelihood of -177.3755, which is lower than the likelihood of the VARFI model, despite including two more estimated parameters. Thus, the VARFI model is a better fit to these data than a vector autoregression is.

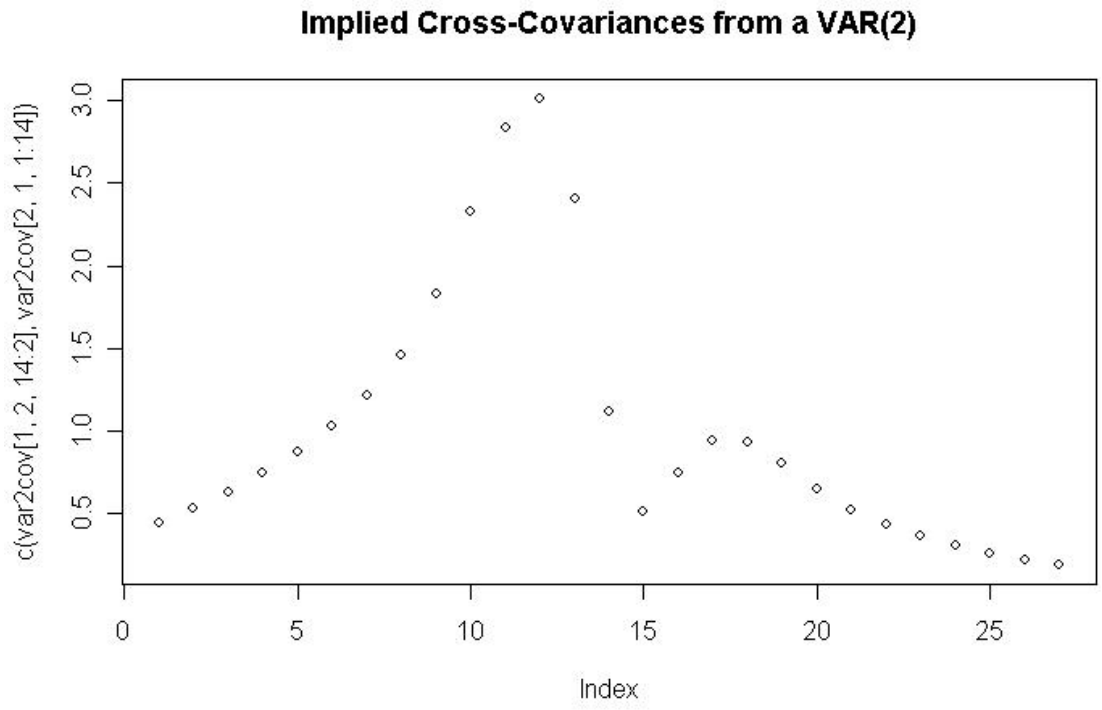

Figure 32: The cross-correlation function of the unemployment rate and the inflation rate implied by the VAR(2) model.

2.10.3 Great Lakes Precipitation

We now model data on precipitation in the Great Lakes. This data, from Hipel and McLeod[2], measures the annual precipitation, in inches, on Lakes Huron, Michigan, and Superior from 1900 to 1986. The autocorrelation functions of the three series, in Figure 33, suggest that the series for Lakes Huron and Superior have some long memory, while Lake Michigan's series has short memory or a differencing parameter very close to zero. Furthermore, the cross-correlation function of Lakes Huron and Superior, shown in Figure 33 seems to decay slowly. The two cross-correlation functions with Lake Michigan decay more quickly.

In Tables 33 and 34, we report the estimated parameter values for the FIVAR and VARFI models. Because the likelihood of the FIVAR model is dramatically higher than that of the VARFI model, we focus on the FIVAR model as the better description of the data. According to the maximum likelihood estimates of the FIVAR model, the precipitation at Lake Superior has the largest differencing parameter, while the differencing parameter of the precipitation at Lake Michigan is almost 0. The cross-covariances between Lake Huron and Lake Superior are plotted in Figure 34.

2.11 Conclusion

This paper has discussed two multivariate generalizations of fractionally integrated autoregressive models. While the two models appear similar at first glance, their implications differ dramatically. One model leads to series with different orders of integration, while the other can lead to series which have the same order of integration but a relationship like cointegration among them. We have also described

[2]These data are available online from `http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/` in the meteorology section.

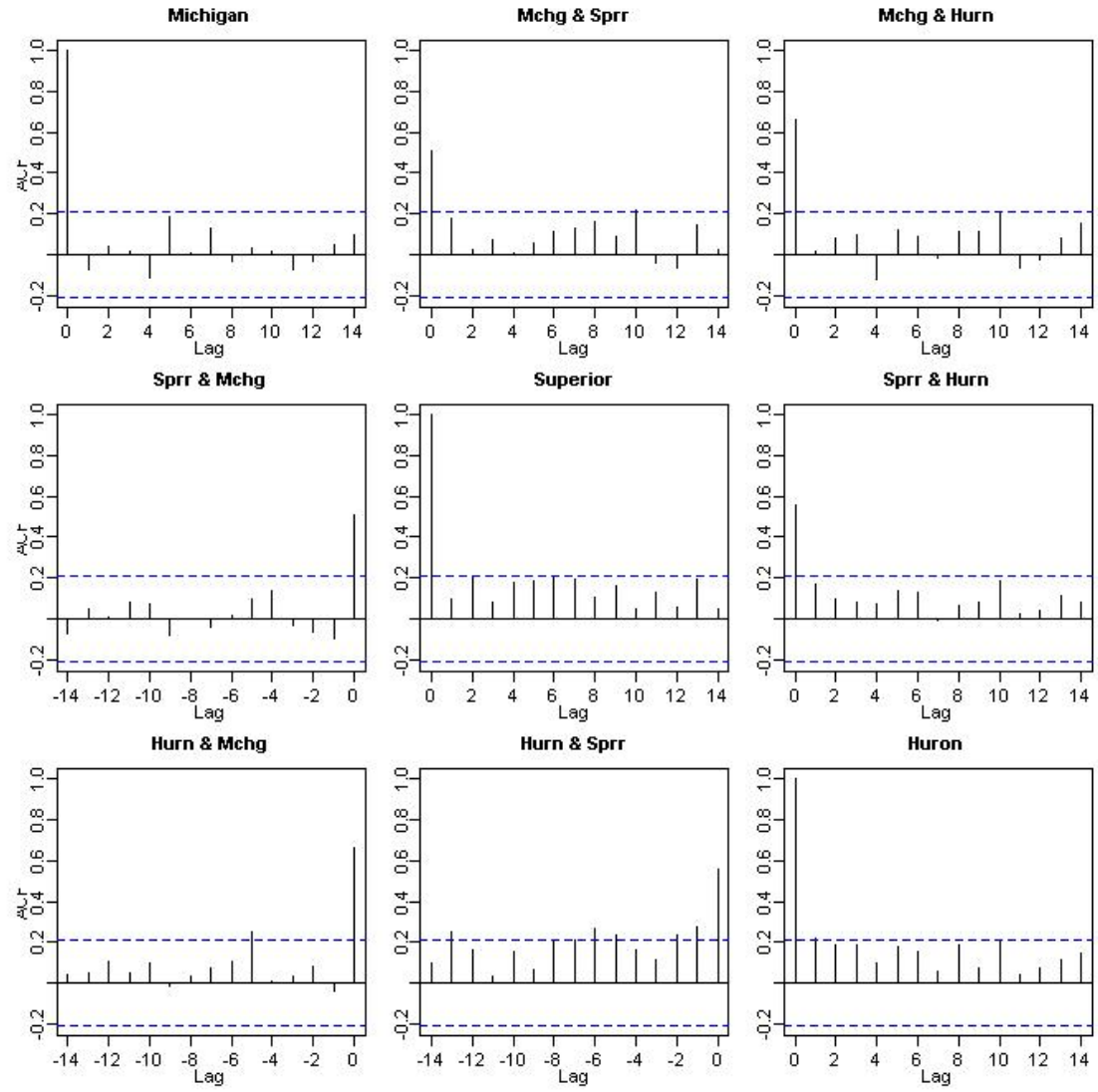

Figure 33: The empirical auto-correlation and cross-correlations function of the annual precipitation at Lakes Superior, Huron, and Michigan.

	Maximum Likelihood with Regression Approximation	Exact Maximum Likelihood	Whittle Approximation
A_1	(-0.03, -0.06, -0.01, 0.17, -0.40, 0.19, -0.25, 0.11, 0.18)	(-0.03, -0.06, -0.01, 0.17, -0.40, 0.19, -0.25, 0.11, 0.18)	(0.03, 0.28, -0.18, -0.33, 0.65, 0.26, -0.36, 0.36, 0.31)
Σ	(9.83, 5.69, 6.77, 5.69, 10.05, 5.53, 6.77, 5.53, 9.68)	(9.83, 5.69, 6.77, 5.69, 10.05, 5.53, 6.77, 5.53, 9.68)	(1.39, 0.78, 0.83, 0.78, 1.69, 0.75, 0.83, 0.75, 1.26)
d	(0.0004, 0.2464, 0.0982)	(0.0000, 0.2460, 0.0980)	(-0.1832, -0.4900, -0.2468)
Log likelihood	-380.3562	-380.3556	-1159.296

Table 33: FIVAR estimates for the precipitation data. All log likelihoods are the exact log likelihoods computed using Sowell's algorithm at the estimated parameter values.

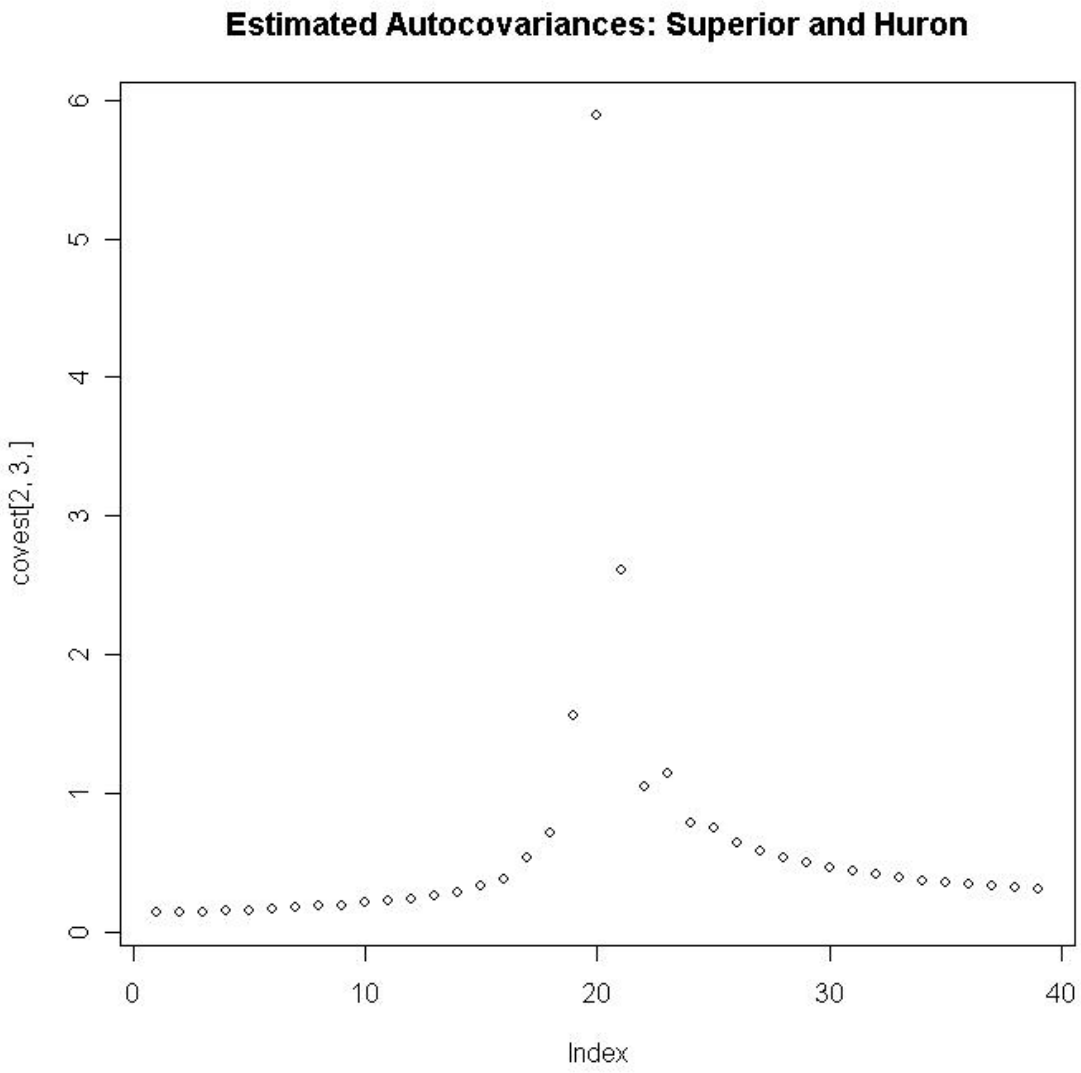

Figure 34: The cross-covariance function between precipitation at Lake Huron and Lake Superior implied by the maximum likelihood estimate of the FIVAR model.

	Maximum Likelihood with Regression Approximation	Whittle Approximation
A_1	(-0.2081, -0.3401, -0.8997, -0.7509, -0.4554, 0.3307, -0.4317, 0.4014, -0.0602)	(-0.2798, 0.3268, -0.0757, -0.4089, 0.2689, 0.3370, -0.4194 0.3236, 0.1464)
Σ	(2.8295, 3.3341, 3.4696, 3.3341, 3.9288, 4.1146, 3.4696, 4.1146, 11.0970)	(1.4674, 0.8552, 0.9218, 0.8552, 1.6725, 0.8563, 0.9218, 0.8563, 1.4106)
d	(-0.4900, -0.3196, 0.1498)	(0.0533, -0.2027, 0.0698)
Log likelihood	-583.5166	-926.7704

Table 34: VARFI estimates for the precipitation data. All log likelihoods are the exact log likelihoods computed using Sowell's algorithm at the estimated parameter values.

computationally efficient methods for using these two models. The algorithms for simulation and computing the quadratic form can be applied to any multivariate model, not just FIVAR and VARFI models. Finally, we have fit these models to data.

Much research remains to be done, because these models are relatively new. There are likely to be theoretical results on the growth of the condition number of Ω, just as there are in the univariate case. It make also be possible to prove whether it is always possible to simulate if sufficiently many covariances are used. It is also unknown whether there is a more elegant algorithm for computing the determinant. Work also remains to be done on cointegration in these models. We hope that finding algorithms which make computation with these models faster will allow them to enter wider use, so that long memory can be addressed in a multivariate context.

3 Power laws in phase and coherency for bi-variate long-memory time series

3.1 Introduction

Semiparametric models for univariate long-memory time series have been explored in detail in existing literature, but the multivariate case presents additional challenges that have not yet been fully surmounted. Of the previous work in multivariate long-memory time series, we are not aware of any that focuses on power laws in phase and coherency, but power laws or other powers of the frequency, λ, in either or both of these may affect convergence rates of estimators of other quantities, such as cointegrating parameters or memory parameters of the individual series (see Section 3.2.1 and Table 36 for details). Power laws and powers of λ in the phase and coherency have been allowed by some (but not all) authors, but their implications have not been discussed as far as we know.

After a brief review of long memory, phase and coherency, we introduce a semiparametric long-memory time series model in Section 3.2 that allows for power laws in the phase and coherency, discussing how previous authors approached phase and coherency in Section 3.2.1 and providing a number of time-domain examples in Sections 3.2.2 through 3.2.7. In Section 3.3, we discuss some of the problems that arise in estimating power laws in coherency in a long-memory context and show that the averaged periodogram estimator (APE) is consistent in these cases, under certain conditions. Unfortunately, high variability of the estimators in small samples makes power laws in the coherency hard to detect with the APE. In Section 3.4, we will show how the properties of the phase and coherency affect a number of cointegration estimators, including the narrow-band least squares estimator (NBLS) of Robinson [1994], Robinson and Marinucci, 2003] and Christensen and Nielsen

[2006] and the local Whittle estimator of Robinson [2008]. For the appropriate choice of the number of frequencies used in estimation, each of these estimators would require knowledge of the exponents in any power laws and any other powers of λ in the phase and coherency in a neighborhood of zero frequency; in light of the apparent difficulties of estimating such power laws when the sample size is not exceedingly large, requiring such knowledge seems problematic. We will prove in Section 3.4.1 that the cointegration estimator of Chen and Hurvich [2003], which we will call the very-narrow-band least-squares estimator (VNBLS), is not affected by such behavior in the phase and coherency, allowing for robust estimation of the cointegrating parameter without knowledge of the exponents in the powers of λ in the phase and coherency of the underlying series. In Section 3.5, we apply the APE to a bivariate time series of two components of the money supply and VNBLS to estimating cointegration between daily high and low stock prices.

3.1.1 Basic properties of long memory and the phase and coherency

In a univariate weakly stationary long-memory process, the spectral density obeys $f(\lambda) \sim C|1 - e^{-i\lambda}|^{-2d}$ as $\lambda \to 0^+$, for $C > 0$ and $d < \frac{1}{2}$; we say that the corresponding process is $I(d)$. One strand of the long-memory literature focuses on semiparametric methods, modeling the spectral density only in a neighborhood of zero, in order to estimate d. Estimation methods in the univariate case include the averaged periodogram estimator (APE) [Robinson, 1994, Lobato and Robinson, 1996], the log periodogram (GPH) estimator [Geweke and Porter-Hudak, 1983, Robinson, 1995a], and the Gaussian semiparametric estimator (GSE) [Kunsch, 1987, Robinson, 1995b].

In this paper, we will focus on real-valued bivariate time series, $X_t = (x_{1t}, x_{2t})'$,

with a spectral density matrix given by:

$$f(\lambda) \;=\; \begin{pmatrix} f_{11}(\lambda) & f_{12}(\lambda) \\ f_{21}(\lambda) & f_{22}(\lambda) \end{pmatrix}, \;\; \lambda \in [-\pi, \pi]$$

where $f^*(\lambda) = f(\lambda)$ and $f(-\lambda) = \overline{f(\lambda)}$, with A^* denoting the conjugate transpose of a matrix A. The cross-spectrum, $f_{12}(\lambda)$, can be decomposed into the phase, $\phi(\lambda)$, the coherency, $\rho(\lambda)$, and terms involving the auto-spectra:

$$f_{12}(\lambda) \;=\; \sqrt{f_{11}(\lambda) f_{22}(\lambda)}\rho(\lambda)e^{i\phi(\lambda)} \tag{3.1}$$

where the coherency is a real, even function with $0 \leq \rho(\lambda) \leq 1$ and the phase is an odd function that we assume lies in the interval $(-\pi, \pi]$. When $f_{11}(\lambda_0)$ or $f_{22}(\lambda_0)$ is zero or infinite for some λ_0, as may happen with $\lambda_0 = 0$ long-memory time series, $\phi(\lambda_0)$ is not uniquely defined. (Terminology regarding the coherency varies. Some authors, such as Brillinger [1981], use the term coherency for the quantity $\rho(\lambda)e^{i\phi(\lambda)}$, which Priestley [1981] calls the "complex coherency." Others, such as Koopmans [1974], Bloomfield [1976], Brockwell and Davis [1993], discuss only $\rho(\lambda)^2$, which they call either the squared coherence or the squared coherency.)

To interpret the coherency, we use the spectral representation for the coordinates, $x_{jt} = \int_{-\pi}^{\pi} e^{it\lambda} dZ_j(\lambda)$, $j = 1, 2$. The coherency is the modulus of the complex correlation of $dZ_1(\lambda)$ and $dZ_2(\lambda)$. Koopmans [1974, page 142] describes the squared coherency as "the proportion of power at frequency λ in either time series ... which can be explained by its linear regression on the other." As a very simple example, the coherency of a white noise series, $\{\epsilon_t\}$, with $Cov(\epsilon_t) = \Sigma$ is given by $\rho(\lambda) \equiv \frac{\sigma_{12}}{\sqrt{\sigma_{11}\sigma_{22}}}$, where σ_{jk} is the (j, k) element of Σ. For general bivariate time series, the strength of the relationship between the two series can vary by frequency, allowing a practitioner to identify and interpret strong or weak relationships at particular ranges of frequencies. For example, Bernanke and Powell

[1984, Table 10.7] focus on the coherency in the range of frequencies corresponding to the business cycle, usually ranging from 2 to 8 years, trying to identify which common measures of the business cycle are strongly related at those frequencies.

The phase is difficult to interpret directly, but the first derivative of the phase, called the group delay by Hannan and Thomson [1973] and others, has a straightforward interpretation. Consider the case where $x_{1t} = x_{2,t-a} + u_t$, for any real a, with $\{u_t\}$ and $\{x_{2t}\}$ uncorrelated at all leads and lags, so that $\{x_{1t}\}$ lags $\{x_{2t}\}$ by a periods. Then, the group delay is given by $\phi'(\lambda) \equiv a$. (See, for example, Priestley [1981, page 663-664] for more details.) In general, if $\phi'(\lambda)$ is not constant, we say that $\{x_{1t}\}$ lags $\{x_{2t}\}$ by $\phi'(\lambda)$ periods at frequency λ, so that the group delay varies by frequency. To compute the group delay given a cross-spectral density, $f_{12}(\lambda) = c(\lambda) - iq(\lambda)$, where $c(\lambda), q(\lambda)$ are real-valued, we take the derivative of $\arctan(-q(\lambda)/c(\lambda))$, though computing the phase itself might require the addition or subtraction of π if $c(\lambda) < 0$. As with coherency, non-constancy in the group delay can lead to hypotheses about the relationship between two time series, since one may lead the other at high frequencies but lag at low frequencies. (See, for example, Bernanke and Powell [1984, Table 10.8].)

Long-memory time series with components having different memory parameters can have phase and coherency with power laws or that depend on powers of λ. Our theoretical framework in Section 3.2 can lead to phase and coherency that satisfies the following local models:

$$\rho(\lambda) = C_\rho \lambda^{-2d_\rho} + o(\lambda^{-2d_\rho}) \tag{3.2}$$

$$\phi(\lambda) = \phi_0 + \phi_1 \lambda^\alpha + o(\lambda^\alpha) \tag{3.3}$$

where $C_\rho > 0$, $d_\rho \leq 0$, $-\pi < \phi_0 \leq \pi$, and $\alpha > 0$. We present examples in which d_ρ and α can take on a variety of possible values in Section 3.2.4 through 3.2.7.

In this paper, we will use the fact that applying real-valued linear filters of the

form $\sum_{u=-\infty}^{\infty} a_{ju} x_{j,t-u}$, for $j = 1, 2$, to the two time series individually will leave their coherency at a frequency λ unchanged (see, for example, Priestley [1981, page 661], Koopmans [1974, page 149]). Furthermore, applying identical linear filters to the two time series also leaves the phase of the time series unchanged. For example, the differences of bivariate time series will have the same phase and coherency as the original bivariate time series. This allows us to extend the concept of phase and coherency to certain non-stationary time series by identifying them with the phase and coherency of the stationary series that result from differencing them.

3.2 Some possible behaviors in the phase and coherency

We introduce a semiparametric model for a bivariate time series, $\{X_t\}$, that explicitly describes a rich variety of behavior in the phase and coherency. This model is based on requiring $\{x_{jt}\}$, for $j = 1, 2$, to be the sum of up to p component series, where each component series may have a different memory parameter. Making this representation explicit allows us to derive the phase and coherency, instead of assuming forms for the phase and coherency without reference to how they arose. In addition, the semiparametric model makes the creation of time domain examples particularly straightforward.

We assume that $\{X_t\}$ has the infinite moving average representation

$$X_t = \sum_{r=-\infty}^{\infty} \psi_r \epsilon_{t-r} \tag{3.4}$$

where the real-valued $2 \times p$ matrices, ψ_r, are specified below and $\epsilon_t = (\epsilon_{1t}, ...\epsilon_{pt})'$ is a p-variate, zero-mean series ($p \geq 2$) that satisfies the following:

Assumption 3.1 *$\{\epsilon_t\}$ is independent and identically distributed with:*

- *$Cov(\epsilon_t) = \Sigma$, where Σ is positive definite.*

- $E\left(\epsilon_{kt}^4\right) < \infty$ *for* $k = 1, ..., p$.

Allowing for more than two driving innovation series allows for straightforward descriptions of a rich variety of models (for example, those in Sections 3.2.5 and 3.2.6). Other authors, including Hannan [1970] and Robinson [2008], have also allowed $p > 2$.

Equation (3.4) implies that $\{X_t\}$ is the output of passing $\{\epsilon_t\}$ through a linear filter with transfer function $\Psi(\lambda) = \sum_{r=-\infty}^{\infty} \psi_r e^{-i\lambda r}$, a $2 \times p$ matrix with entries, $\Psi_{jk}(\lambda)$, for $j = 1, 2$ and $k = 1, ..., p$. For each (j, k), we generalize Chen and Hurvich [2003] and consider transfer functions, $\Psi_{jk}(\lambda)$ on $[-\pi, \pi]$, that can be written as:

$$\Psi_{jk}(\lambda) \;\; = \;\; (1 - e^{-i\lambda})^{-\delta_{jk}} \tau_{jk}(\lambda) e^{i\varphi_{jk}(\lambda)} \tag{3.5}$$

with $\tau_{jk}(\lambda), \delta_{jk}, \varphi_{jk}(\lambda)$ satisfying Assumptions 2-5 (3.5 appears after discussion of Assumptions 2-4):

Assumption 3.2 *For* $j = 1, 2$ *and* $k = 1, ..., p$, $\tau_{jk}(\lambda)$ *is a real, bounded, non-negative, continuous, even function on* $[-\pi, \pi]$ *that is differentiable on* $[-\pi, \pi] - \{0\}$, *with* $\tau'_{jk}(\lambda) = o(\lambda^{-1})$ *as* $\lambda \to 0^+$. *Furthermore, either* $\tau_{jk}(0) > 0$ *or* $\tau_{jk}(\lambda) = 0$ *for all* $\lambda \in [0, \pi]$; *for each* j, $\tau_{jk}(0) > 0$ *for at least one* k.

Assumption 3.3 $\delta_{jk} < 1/2$ *for all* $j = 1, 2$ *and* $k = 1, ..., p$. *When* $\tau_{jk}(\lambda) = 0$, δ_{jk} *is less than or equal to the smallest* $\delta_{jk'}$ *with* $\tau_{jk'}(\lambda) > 0$.

Assumption 3.4 $\varphi_{jk}(\lambda)$ *is an odd, differentiable function on* $[-\pi, \pi] - \{0\}$, *where* $\lim_{\lambda \to 0^+} \varphi_{jk}(\lambda)$ *exists and* $\varphi'_{jk}(\lambda)$ *is continuous at 0 or obeys* $\varphi'_{jk}(\lambda) = o(\lambda^{-1})$ *as* $\lambda \to 0^+$.

The decomposition in Equation (3.5) separates the transfer function into three different operations that transform the series $\{\epsilon_{kt}\}$ into a component of $\{x_{jt}\}$. First,

$(1-e^{-i\lambda})^{-\delta_{jk}}$ is the fractional integration operator of order δ_{jk}. Because δ_{jk} varies with k, $\{x_{jt}\}$ consists of components with potentially different orders of integration. Second, $\tau_{jk}(\lambda)$ is either a filter that changes the short-memory properties of the resulting fractionally integrated or a filter that annihilates the component near the zero frequency. Finally, $\varphi_{jk}(\lambda)$ changes the phase of $\{x_{jt}\}$ relative to the original $\{\epsilon_{kt}\}$. For example, if $\varphi_{jk}(\lambda) = -a\lambda$, for some real a, then the component of $\{x_{jt}\}$ that depends on $\{\epsilon_{kt}\}$ is lagged by a periods. More complicated phase shifts are also possible.

The spectral density of $\{X_t\}$ is simply:

$$f(\lambda) \;=\; \Psi(\lambda)\Sigma\Psi(\lambda)^*, \;\; \lambda \in [-\pi, \pi]$$

Using the representation in Equation (3.5), the autospectral densities, $f_1(\lambda)$ and $f_2(\lambda)$, are given for $j = 1, 2, \lambda \in [-\pi, \pi]$ by:

$$f_j(\lambda) \;=\; \sum_{k=1}^{p}\sum_{l=1}^{p}(1-e^{-i\lambda})^{-\delta_{jk}}(1-e^{i\lambda})^{-\delta_{jl}}\sigma_{kl}\tau_{jk}(\lambda)\tau_{jl}(\lambda)e^{i(\varphi_{jk}(\lambda)-\varphi_{jl}(\lambda))}$$

where σ_{kl} is the (k, l) element of Σ. To define the power law in the auto-spectra, it is helpful to rewrite the auto-spectrum for $j = 1, 2, \lambda \in (0, \pi]$ as

$$\begin{aligned} f_j(\lambda) \;=\; & \sum_{k=1}^{p}\sigma_{kk}|1-e^{-i\lambda}|^{-2\delta_{jk}}\tau_{jk}(\lambda)^2 \\ & +2\sum_{k=1}^{p}\sum_{l<k}|1-e^{-i\lambda}|^{-\delta_{jk}-\delta_{jl}}\sigma_{kl}\tau_{jk}(\lambda)\tau_{jl}(\lambda) \\ & \times\cos\left(\varphi_{jk}(\lambda)-\varphi_{jl}(\lambda)+\frac{(\delta_{jl}-\delta_{jk})(\pi-\lambda)}{2}\right) \end{aligned}$$

The power laws in the auto-spectra are defined by the largest δ_{jk} that have non-zero coefficients, $\tau_{jk}(\lambda)^2$, in the first equation; the sum in the second line will not change the power law in the auto-spectrum. Thus, we define

$$d_j = \max_{k:\tau_{jk}(0)>0}\delta_{jk}$$

This semiparametric model implies that, as $\lambda \to 0^+$:

$$f_j(\lambda) \sim C_j \lambda^{-2d_j} \tag{3.6}$$

where $C_j = \lim_{\lambda \to 0^+} \sum_{k=1}^{p} \sum_{l=1}^{p} \sigma_{kl} \tau_{jk}(\lambda) \tau_{jl}(\lambda) e^{i(\varphi_{jk}(\lambda) - \varphi_{jl}(\lambda))} \chi(\delta_{jk} = \delta_{jl} = d_j)$ and χ is the indicator function.

In describing the cross-spectrum, it will be convenient to separate the power law into its modulus and argument. When $\lambda \in (0, \pi]$, we use the identity $(1 - e^{-i\lambda}) = \left|2 \sin \frac{\lambda}{2}\right| e^{i(\pi - \lambda)/2}$ to rewrite Equation (3.5) as:

$$\Psi_{jk}(\lambda) = \left|2 \sin \frac{\lambda}{2}\right|^{-\delta_{jk}} \tau_{jk}(\lambda) e^{i(\varphi_{jk}(\lambda) + (\pi - \lambda)\delta_{jk}/2)} \tag{3.7}$$

Using Equations (3.5), (3.6), and (3.7), we find that for $\lambda \in (0, \pi]$, the cross-spectral density is given by:

$$\begin{aligned} f_{12}(\lambda) &= \sum_{k=1}^{p} \sum_{l=1}^{p} (1 - e^{-i\lambda})^{-\delta_{1k}} (1 - e^{i\lambda})^{-\delta_{2l}} \sigma_{kl} \tau_{1k}(\lambda) \tau_{2l}(\lambda) e^{i(\varphi_{1k}(\lambda) - \varphi_{2l}(\lambda))} \\ &= \sum_{k=1}^{p} \sum_{l=1}^{p} \left|2 \sin \frac{\lambda}{2}\right|^{-\delta_{1k} - \delta_{2l}} \sigma_{kl} \tau_{1k}(\lambda) \tau_{2l}(\lambda) e^{i(\varphi_{1k}(\lambda) - \varphi_{2l}(\lambda) + (\pi - \lambda)(\delta_{1k} - \delta_{2l})/2)} \end{aligned}$$

In order to understand the power law behavior of the cross-spectrum, we decompose the sum above into a sum of terms where the power, $\delta_{1k} + \delta_{2l}$, is constant. To do this, partition the set of $\{(k, l) : k, l \in \{1, ..., p\}$ into sets $S_1, ..., S_Q$ such that $\delta_{1k} + \delta_{2l} = \delta_{1k'} + \delta_{2l'}$ if and only if $(k, l), (k', l')$ are in the same set. Define $d_{12}(q)$ to be the value of $\frac{1}{2}(\delta_{1k} + \delta_{2l})$ for $(k, l) \in S_q$ for $q = 1, ..., Q$, with $d_{12}(q) > d_{12}(q + 1)$ for all $q = 1, ..., Q - 1$. Note that $d_{12}(1) = d_1 + d_2$. Then, for $0 < \lambda < \pi$, we may write:

$$f_{12}(\lambda) = \sum_{q=1}^{Q} \left|2 \sin \frac{\lambda}{2}\right|^{-2d_{12}(q)} s(\lambda; q)$$

where

$$s(\lambda;q) = \sum_{(k,l)\in S_q} \sigma_{kl}\tau_{1k}(\lambda)\tau_{2l}(\lambda)e^{i(\varphi_{1k}(\lambda)-\varphi_{2l}(\lambda)+(\pi-\lambda)(\delta_{1k}-\delta_{2l})/2)} \quad (3.8)$$

$$s(0;q) = \lim_{\lambda\to 0^+} s(\lambda;q) \quad (3.9)$$

$$= \sum_{(k,l)\in S_q} \sigma_{kl}\tau_{1k}(0)\tau_{2l}(0) \lim_{\lambda\to 0^+} \left(e^{i(\varphi_{1k}(\lambda)-\varphi_{2l}(\lambda)+(\pi-\lambda)(\delta_{1k}-\delta_{2l})/2)}\right) \quad (3.10)$$

To ensure that the power law behavior in the cross-spectral density is determined by the terms containing $\left|2\sin\frac{\lambda}{2}\right|^{-d_{12}(q)}$ instead of by the $s(\lambda;q)$, we make the following assumption:

Assumption 3.5 *There is at least one q such that $s(0;q) \neq 0$. Let q_0 be the smallest such q. Then, we define:*

$$d_{12} = d_{12}(q_0) \quad (3.11)$$

Whenever $s(0;q) = 0$, we require that:

$$s(\lambda;q) = o\left(\lambda^{-2(d_{12}-d_{12}(q))}\right)$$

Using this representation, we describe the power laws in the modulus of the cross-spectrum and the coherency in a neighborhood of zero frequency. As $\lambda \to 0^+$, the absolute value of the cross spectrum and the coherency obey:

$$|f_{12}(\lambda)| \sim C_{12}\lambda^{-2d_{12}} \quad (3.12)$$

$$\rho(\lambda) \sim \frac{C_{12}}{\sqrt{C_1 C_2}}\lambda^{-2\left(d_{12}-\frac{1}{2}(d_1+d_2)\right)} \quad (3.13)$$

where the power law of the cross-spectrum, d_{12}, is defined by Equation (3.11), C_1, C_2 are defined by Equation (3.6) and $C_{12} = |s(0;q_0)|$. Since $\delta_{1k}+\delta_{2l} \leq d_1+d_2$, d_{12} is bounded above by $\frac{1}{2}(d_1+d_2)$. The fact that d_{12} need not equal $\frac{1}{2}(d_1+d_2)$ was mentioned by Lobato [1997, page 139], though he did not try to estimate d_{12}.

In the case where $d_{12} < \frac{1}{2}(d_1 + d_2)$, the coherency will have power law decay in a neighborhood of zero; we will call such behavior *power law coherency*. We define:

$$d_\rho = d_{12} - \frac{1}{2}(d_1 + d_2) \tag{3.14}$$

The only decay rate of the cross-spectral density that will not lead to power law coherency is $d_{12} = \frac{1}{2}(d_1 + d_2)$. Thus, power law coherency will occur when $s(0;1) = 0$. The examples given in Sections 3.2.5 and 3.2.7 show how this can occur. In a sense, this is the opposite of cointegration (see Section 3.2.4), where there is a very strong long-run relationship between two time series because the coherency is 1 at frequency 0. Power law coherency will also affect forecasts for bivariate time series. Consider a forecast for $x_{1,T+h}$ based on on $x_{11}, x_{21}, ..., x_{1T}, x_{2T}$. The weights on $x_{21}, ...x_{2T}$ will decay more quickly with h when there is power law coherency than when there is no power law coherency. Thus, one should account for the possibility of power law coherency before computing long-range forecasts of bivariate time series.

Next, we write the phase, using the notation of Equation (3.8), for $0 < \lambda < \pi$:

$$\begin{aligned}
\phi(\lambda) &= \arg\left(\sum_{k=1}^{p}\sum_{l=1}^{p}\left|2\sin\frac{\lambda}{2}\right|^{-\delta_{1k}-\delta_{2l}}\sigma_{kl}\tau_{1k}(\lambda)\tau_{2l}(\lambda)e^{i\left(\varphi_{1k}(\lambda)-\varphi_{2l}(\lambda)+\frac{(\pi-\lambda)(\delta_{1k}-\delta_{2l})}{2}\right)}\right) \\
&= \arg\left(\sum_{q=1}^{Q}\left|2\sin\frac{\lambda}{2}\right|^{-2d_{12}(q)}s(\lambda;q)\right) \\
&= \arg\left(\sum_{q=1}^{Q}\left|2\sin\frac{\lambda}{2}\right|^{-2(d_{12}(q)-d_{12}(q_0))}s(\lambda;q)\right)
\end{aligned}$$

where q_0 is defined in Assumption 3.5. In the case of long-memory time series, the phase is not uniquely defined at frequency zero, since the auto-spectral densities are zero or infinite at that frequency. Because of this, we focus on the right-hand limit of the phase and define $\phi_0 = \arg(s(0;q_0))$. As was discussed by Shimotsu [2007], the phase need not be continuous at zero for long-memory time series, so

that ϕ_0 is not necessarily 0 or π. This is true for even the simplest bivariate long-memory time series models (FIVAR models, discussed in Section 3.2.3); further examples are discussed by Shimotsu [2007], Robinson [2008], and in Section 3.2.5. Though the phase is not continuous at zero for long-memory models, the phase is an odd function, so that $\phi'(\lambda) = \phi'(-\lambda)$. Thus, when $\lim_{\lambda \to 0^+} \phi'(\lambda) = \lim_{\lambda \to 0^-} \phi'(\lambda)$ exists, we define the group delay at frequency 0 to be the limit of the derivative as the phase approaches zero frequency.

In some long-memory models, such as fractional cointegration (Section 3.2.4) and those described in Sections 3.2.6 and 3.2.7, the phase will include powers, λ^α, where $\alpha > 0$ to ensure the existence of a right-hand limit. This means that group delay may be infinite at zero, if $\alpha < 1$. Powers of λ in the phase arise when $\left|2 \sin \frac{\lambda}{2}\right|^{-2(d_{12}(q) - d_{12}(q_0))} s(\lambda; q) \sim \lambda^\alpha$ for some q; in the case where the power depends on the term containing q_0, the power must come from powers of λ in $s(\lambda; q_0)$ itself. If we can write $\phi(\lambda) = \phi_0 + \phi_1 \lambda^\alpha + o(\lambda^\alpha)$, the group delay is given by $\phi'(\lambda) \sim \alpha \phi_1 \lambda^{\alpha - 1}$ as $\lambda \to 0^+$, as long as the terms in $o(\lambda^\alpha)$ have derivatives of a smaller order. Then, when $0 < \alpha < 1$, the group delay associated with a period of T years is proportional to $T^{1-\alpha}$ as T increases; thus, one series leads the other by increasing amounts at larger lags. When $\alpha > 1$, the group delay approaches zero at frequency 0 with power law behavior.

In some of our analysis, including our proof about the distribution of the VN-BLS estimator in Section 3.4.1, we will apply linear filters to the series in order to make the individual coordinates $I(0)$, with a phase that is continuous at frequency 0. Generalizing Chen and Hurvich [2003, Equation 7], define:

$$\Upsilon(\lambda) = \begin{pmatrix} (1 - e^{-i\lambda})^{-d_1} & 0 \\ 0 & (1 - e^{-i\lambda})^{-d_2} e^{-i(\phi_0 + \frac{\pi}{2}(d_2 - d_1)) sign(\lambda)} \end{pmatrix} \quad (3.15)$$

If $\phi_0 = \frac{\pi}{2}(d_1 - d_2)$, then $\Upsilon(\lambda)$ is the transfer function of the linear filter that

takes the d_1 difference of the first series and the d_2 difference of the second series; this matches the definition of Chen and Hurvich. Allowing ϕ_0 to vary allows for alternative right-hand limits in the phase, as Robinson [2008] allows. Then, define $\Psi^\dagger(\lambda), f^\dagger(\lambda)$ by:

$$\Psi(\lambda) = \Upsilon(\lambda)\Psi^\dagger(\lambda) \tag{3.16}$$

$$f(\lambda) = \Upsilon(\lambda)f^\dagger(\lambda)\Upsilon^*(\lambda) \tag{3.17}$$

$$f_{12}^\dagger(\lambda) = e^{-i(\phi_0+\frac{\pi}{2}(d_2-d_1))sign(\lambda)} \sum_{k=1}^{p}\sum_{l=1}^{p} |2\sin(\lambda/2)|^{d_1-\delta_{1k}+d_2-\delta_{2l}}\sigma_{kl} \tag{3.18}$$

$$\times\tau_{1k}(\lambda)\tau_{2l}(\lambda)e^{i(\varphi_{1k}(\lambda)-\varphi_{2l}(\lambda)+(\pi-\lambda)(\delta_{1k}-d_1-\delta_{2l}+d_2)/2)}, \quad \lambda > 0 \tag{3.19}$$

$f^\dagger(\lambda)$ is the spectral density of the bivariate time series obtained after fractionally differencing the two series to make them $I(0)$ with a phase that is continuous at 0. The bivariate series with spectral density $f^\dagger(\lambda)$ have the same coherency as the original series, but will have a different phase unless $d_1 = d_2$ and $\phi_0 = 0$. In all cases, $f_{12}^\dagger(\lambda)$ is continuous at $\lambda = 0$, because the discontinuity in the phase at 0 has been removed. In the case of power law coherency, the inclusion of $e^{-i(\phi_0+\frac{\pi}{2}(d_2-d_1))sign(\lambda)}$ does not affect the fact that $f_{12}^\dagger(\lambda)$ is continuous at frequency 0, since $f_{12}^\dagger(0)$ is zero.

3.2.1 Previous literature on the effects of phase and coherency on estimators

The bulk of previous literature about bivariate long-memory time series and fractional cointegration has implicitly (or explicitly) approximated the phase and coherency by constants in a neighborhood of frequency 0. Many authors [Robinson, 1995a, Lobato, 1997, 1999, Shimotsu, 2007] write semiparametric models for the cross-spectral densities of the following form:

$$f_{12}(\lambda) = C_{12}\lambda^{-d_1-d_2} + O(\lambda^{-d_1-d_2+\xi})$$

Paper	Target	Assumptions	Convergence Rate
Robinson [1995a, Assumptions 3 and 6, Theorem 3]	d_1, d_2 with GPH	$\lvert\rho(\lambda)e^{i\phi(\lambda)} - \rho(0)e^{i\phi_0}\rvert = O(\lambda^\xi)$ as $\lambda \to 0^+$, with $-\frac{1}{2} < d_1, d_2 < \frac{1}{2}$ and $\xi \in (0, 2]$	$\frac{m^{1+2\xi}}{n^{2\xi}} \to 0$ as $n \to \infty$ implies that $\hat{d}_j - d_j = O_p(m^{-1/2})$
Lobato [1997, Condition C1' and Theorem 2]	d_1, d_2 with APE	$f_{ab}(\lambda) = C_{ab}\lambda^{-d_a-d_b} + O(\lambda^{-d_a-d_b+\xi})$ as $\lambda \to 0^+$, $d_a, d_b \in (0, 1/2)$, $0 < \lvert g_{ab}\rvert < \infty$	$\hat{d}_j - d_j = O_p\left(\left(\frac{m}{n}\right)^\xi\right)$
Lobato [1999, (4), (11), and Assumptions A1 and A4]	d_1, d_2 with GSE	$\lvert f_{ab}(\lambda) - C_{ab}\lambda^{-d_a-d_b}\rvert = O(\lambda^{-d_a-d_b+\xi})$ as $\lambda \to 0^+$, with $-\frac{1}{2} < d_1, d_2 < \frac{1}{2}$, $\xi \in (0, 2]$, and r_{ab} real	$\frac{(\log m)^2 m^{1+2\xi}}{n^{2\xi}} \to 0$ as $n \to \infty$ implies that $\hat{d}_j - d_j = O_p(m^{-1/2})$
Shimotsu [2007, Assumptions 1' and 4' and Theorem 2]	d_1, d_2 with GSE	$f_{ab}(\lambda) = e^{i\pi(d_a-d_b)/2}C_{ab}\lambda^{-d_a-d_b} + O(\lambda^{-d_a-d_b+\xi})$ as $\lambda \to 0^+$, with $-\frac{1}{2} < d_1, d_2 < \frac{1}{2}$, G_{ab} real and $\xi \in (0, 2]$	$\frac{(\log m)^2 m^{1+2\xi}}{n^{2\xi}} \to 0$ as $n \to \infty$ implies that $\hat{d}_j - d_j = O_p(m^{-1/2})$

Table 35: Assumptions of previous authors regarding the cross-spectrum, phase and coherency. "GSE" is Gaussian semiparametric estimation; "APE" is averaged periodogram estimation; "GPH" is the Geweke and Porter-Hudak log periodogram regression estimator.

Paper	Target	Assumptions	Convergence Rate
Nielsen [2004, page 227]	-	$f^{\dagger}(\lambda) = \Omega(1 + O(\lambda^2))$ as $\lambda \to 0^+$, with Ω real and $\phi_0 = \frac{\pi}{2}(d_1 - d_2)$	-
Robinson and Marinucci [2003, Assumption A and Theorem 3.1]	β using NBLS	$f_{12} \sim C_{12}\lambda^{-d_1-d_2}$ as $\lambda \to 0^+$ where C_{12} is real and $0 \leq d_1 < d_2 < \frac{1}{2}$	$\hat{\beta} - \beta = O_p\left(\left(\frac{n}{m}\right)^{d_u - d_x}\right)$
Christensen and Nielsen [2006, Assumption A' and D and Theorem 2]	β using NBLS	$\lvert f_{ij}(\lambda)\rvert = O(\lambda^{\xi - d_i - d_j})$ as $\lambda \to 0^+$, with $\xi \in (0, 2]$, $d_1 + d_2 < 1/2$ and $0 \leq d_1, d_2 < 1/2$	$\frac{m^{1+2\xi}}{n^{2\xi}} \to 0$ as $n \to \infty$ implies that $\hat{\beta} - \beta = O_p\left(m^{-\frac{1}{2}+d_x-d_u} n^{d_u-d_x}\right)$
Chen and Hurvich [2003, Equation 6]	β using VNBLS	$\Psi(\lambda)$ is 2×2 of the form given in Equation (3.5) with $\delta_{jj} \geq \delta_{jk}$ for $k = 1, 2$. $\tau_{jk}(\lambda) > 0$; $\varphi_{jk}(\lambda)$ continuously differentiable at 0.	$\hat{\beta} - \beta = O_p\left(n^{d_u - d_x}\right)$
Robinson [2008, Assumptions A6, B1, and B5, Theorem 4]	β, d_1, d_2, and ϕ_0 with local Whittle	$\Phi(\lambda)\Psi(\lambda) - P = O(\lambda^{\xi})$ as $\lambda \to 0^+$ where $\Psi(\lambda)$ is an $2 \times p$ transfer function, $\Sigma = I$, and $\Phi(\lambda; \phi_0) = \mathrm{diag}(\lvert\lambda\rvert^{d_1}, \lvert\lambda\rvert^{d_2} e^{-i\mathrm{sign}(\lambda)\phi_0})$ and $\xi \in (0, 2]$. Also, $0 < \rho(0) < 1$, $0 \leq d_1, d_2 < \frac{1}{2}$.	$\frac{(\log m)^2 m^{1+2\xi}}{n^{2\xi}} \to 0$ as $n \to \infty$ implies that $\hat{\beta} - \beta = O_p\left(m^{-\frac{1}{2}+d_x-d_u} n^{d_u-d_x}\right)$

Table 36: Assumptions of previous authors regarding the cross-spectrum, phase and coherency. NBLS and VNBLS are the narrow band and very narrow band least squares estimators. β is the cointegrating parameter; see Section 3.4.

where $\xi \in (0, 2]$ and assumptions about C_{ab} vary. (See Table 36 for the precise assumptions that authors use.) As Lobato [1997] notes, this semiparametric model allows for a power law in the coherency if C_{ab} is allowed to be 0. Because the coherency and phase are not described explicitly, their properties will limit the choices of ξ. Specifically, when $d_\rho < 0$, we must have $\xi \leq -2d_\rho$. When the phase is of the form given in Equation (3.3), we must have $\xi \leq \alpha$. Because d_ρ and α can be arbitrarily close to 0, ξ may be required to be arbitrarily close to 0. All of the previously mentioned authors then use the semiparametric model given to estimate d_1, d_2, and sometimes C_{12} in the equation above, together with C_1, C_2 in the auto-spectra. For the convergence rates of the estimators of Robinson [1995a], Lobato [1999] and Shimotsu [2007], the number of frequencies, m, used in estimation must satisfy $\frac{m^{1+2\xi}}{n^{2\xi}} \to 0$ as $n \to \infty$ (in some cases, additional powers of $\log(m)$ are included in the numerator). This choice of m is required to control the bias in the estimators. In all cases, values of d_ρ and α close to 0 limit the growth rate of m and therefore the convergence rate. Nielsen [2004] makes the more restrictive assumption that $\xi = 2$ to derive results about the phase and coherency of cointegrated series; as we will show in Section 3.2.4, these assumptions are unnecessary for his main results.

In the context of cointegration estimation, Christensen and Nielsen [2006] require that the $C_{ab} = 0$ in the equation above, allowing for power law coherency but ruling out $\rho(0) > 0$ (and therefore commonly used time series models like the FIVAR). However, as in Robinson [1995a], Lobato [1997, 1999], d_ρ determines the number of frequencies that can be used in estimation, with d_ρ close to 0 leading to smaller choices of m.

Our semiparametric model generalizes that of Chen and Hurvich [2003], relaxing the following assumptions. First, they require that $p = 2$ and that $\delta_{jj} \geq \delta_{jk}$

for $j = 1, 2$. Second, Chen and Hurvich require that $\tau_{jk}(0) > 0$ for all j, k. Finally, they require that $\varphi_{jk}(\lambda)$ be continuously differentiable in an interval containing 0. These assumptions allow for power laws in the coherency and powers of λ in the phase in certain cases (such as example of Hosoya [1997] described Section 3.2.5), but limit the ways in which such power laws can occur. Because $\varphi_{jk}(\lambda)$ must be continuous at 0, the model requires that $\phi_0 = \frac{\pi}{2}(d_1 - d_2)$ when $\rho(0) > 0$. In addition, their semiparametric model cannot produce fractional cointegration at all (see Section 3.2.4) unless the resulting series are passed through a second linear filter. As we will discuss in Section 3.4.1, because they hold the number of frequencies, m, used in estimation fixed, the convergence rates of their estimators are not affected by power law behavior or powers of λ in the phase and coherency.

Robinson [2008] suggested the use of local Whittle for cointegration estimation in a context that allowed ϕ_0 to take on any value in $(-\pi, \pi] - \{-\frac{\pi}{2}, \frac{\pi}{2}\}$. However, Robinson (Assumption A6) explicitly ruled out $\rho(0) = 0$, which excludes power law coherency. In addition, he requires that $d_1, d_2 \in [0, \frac{1}{2})$. As in the other papers using local Whittle estimators, higher order terms in the phase are included in a term of the form $O\left(\lambda^{\xi}\right)$ that will affect the number of frequencies that can be used in estimation. To understand the effects of his assumptions, we compare our transfer function in Equation (3.5) to his Assumption B1. Instead of $\Upsilon(\lambda)$, Robinson [2008, page 2510] uses the operator $\Phi(\lambda; \phi_0) = \text{diag}(|\lambda|^{d_1}, |\lambda|^{d_2} e^{-i\text{sign}(\lambda)\phi_0})$; as $\lambda \to 0^+$, the two operators are identical, but they differ at higher frequencies since $\lambda \neq \left|2 \sin \frac{\lambda}{2}\right|$. In his Assumption B1, when $\delta_{jk} < d_j$ for some k with $\tau_{jk} > 0$, we must have $\xi < d_j - \delta_{jk}$. In addition, the presence of group delay or a power law in the phase leads to another upper bound on ξ. To demonstrate this, we will focus on the case in which $\varphi_{jk}(\lambda) = \phi_1 \lambda^{\alpha} + o(\lambda^{\alpha})$ as $\lambda \to 0^+$ for some $0 < \alpha \leq 1$. To ensure that any lack of smoothness comes from the phase, we require that $\delta_{jk} = d_j$ and

$\tau_{jk} = \tau_{jk}(\lambda)$ for all j, k, λ. Then, $A(\lambda) = \Phi(\lambda)\Psi(\lambda)$ is a matrix with elements that obey:

$$A_{jk}(\lambda) = \left| \frac{2\sin\left(\frac{\lambda}{2}\right)}{\lambda} \right|^{-d_j} \tau_{jk}(0) e^{i(\phi_1 \lambda^\alpha + d_j \lambda/2)} + o(\lambda^\alpha)$$

His Assumption B1 requires that there is some constant matrix, P, such that $A(\lambda) - P = O(\lambda^\xi)$. Setting $P_{jk} = A_{jk}(0)$, we find that:

$$\begin{aligned} A_{jk}(\lambda) - P &= O\left(\left|1 - e^{i(\phi_1 \lambda^\alpha + d_j \lambda/2)}\right|\right) \\ &= O\left(\sqrt{2 - 2\cos(\phi_1 \lambda^\alpha + d_j \lambda/2)}\right) \\ &= O(\phi_1 \lambda^\alpha + d_j \lambda/2) \end{aligned}$$

Even if $\alpha = 2$, we have $\xi \leq 1$, so that m can grow only as fast as $n^{2/3}$. This possibility occurs even for FIVAR processes, as will be discussed in Section 3.2.3. If $\alpha < 1$, then $\xi \leq \alpha$. Depending on the nature of the phase, ξ can be arbitrarily close to 0, so that m must be chosen to grow arbitrarily slowly, just as occurred with other estimators.

To illustrate the possible power laws and powers of λ in phase and coherency, we will describe the phase and coherency in three well-studied time series models: vector autoregressions, FIVAR models, and fractional cointegration. We will also describe new time series models that illustrate how power law coherency and powers of λ in phase can occur in the time domain.

3.2.2 Vector autoregressions

We begin by describing the phase and coherency of a simple, short-memory time series model, the vector autoregression; this is a case in which there will not be a power law in the coherency and the phase will be continuous at frequency 0. A bivariate vector autoregression (VAR) is a short-memory time series model in

which:

$$A(L)X_t \;=\; \epsilon_t$$

where L is the lag operator, $A(L)$ is a matrix polynomial of finite order, $A(0)$ is the 2×2 identity matrix, $|A(L)|$ has all of its roots outside the unit circle, and $\{\epsilon_t\}$ is bivariate white noise. The spectral density of a VAR is given by:

$$f_{VAR}(\lambda) \;=\; \frac{1}{2\pi} A\left(e^{-i\lambda}\right)^{-1} \Sigma \left(A\left(e^{-i\lambda}\right)^{-1}\right)^{*} \tag{3.20}$$

for $\lambda \in [-\pi, \pi]$. For a VAR with a non-zero cross-spectrum, $d_1 = d_2 = d_{12} = 0$. Because $|A(L)|$ has all of its roots outside the unit circle, $A(1)^{-1}$ is well-defined, and we may write:

$$f_{VAR}(0) \;=\; \frac{1}{2\pi} A(1)^{-1} \Sigma \left(A(1)^{-1}\right)^{*}$$

Thus, $f_{VAR}(\lambda)$ is continuous and non-zero at 0. When the cross-spectrum is non-zero at zero, $\phi(0)$ is 0 or π, depending on the sign of the $(1,2)$ element of $f_{VAR,12}(0)$. The coherency at zero can be calculated from the expression above and the group delay can be calculated from Equation (3.20), though neither has a simple closed form in general.

3.2.3 FIVAR models

One of the most common parametric bivariate long-memory time series models in the literature is the fractionally integrated vector autoregression (FIVAR), an extension of the univariate ARFIMA model. FIVAR models are described by Sowell [1989a], Hosoya [1996], Lobato [1997], Ravishanker and Ray [1997], Sela and Hurvich [2009], among others. We will show that FIVAR models cannot have power laws in the coherency, but that their phase is discontinuous at frequency 0 with $\phi_0 = \pi(d_1 - d_2)/2$ (agreeing with the more general result of Shimotsu [2007]).

$\{X_t\}$ is a FIVAR process if we may write:

$$A(L)D(L)X_t \;=\; \epsilon_t$$

where $D(e^{-i\lambda}) = \text{diag}\left((1-e^{-i\lambda})^{-d_1}, (1-e^{-i\lambda})^{-d_2}\right)$ and $A(L), \{\epsilon_t\}$ satisfy the assumptions given in the previous section. The spectral density of a FIVAR model is given by:

$$f(\lambda) \;=\; D\left(e^{-i\lambda}\right)^{-1} f_{VAR}(\lambda) D\left(e^{i\lambda}\right)^{-1}$$

for $\lambda \in [-\pi, \pi]$, where $f_{VAR}(\lambda)$ is the spectral density of the vector autoregressive process in Equation (3.20). Notice that $D\left(e^{-i\lambda}\right) = \Upsilon(\lambda)$ with $\phi_0 = \pi(d_1 - d_2)/2$, so that $f^{\dagger}(\lambda) = f_{VAR}(\lambda)$. In a FIVAR model, $d_\rho = 0$. (In fact, the coherency of the FIVAR equals the coherency of the original VAR at all frequencies.) The phase is given by $\phi(\lambda) = \phi_{VAR}(\lambda) + (\pi - \lambda)(d_1 - d_2)/2$. Since the spectral density of a VAR is well-defined, finite, and non-zero at 0, $\lim_{\lambda \to 0^+} = \pi(d_1 - d_2)/2$ or $\lim_{\lambda \to 0^+} = \pi(d_1 - d_2)/2 + \pi$. At all frequencies, the group delay of a FIVAR is $-\frac{d_1 - d_2}{2}$ plus the group delay of the original VAR.

As a simple example showing the difference between the group delay of a FIVAR and that of the corresponding VAR, consider the following time series:

$$x_{1t} \;=\; \epsilon_{1t} \tag{3.21}$$

$$x_{2t} \;=\; \epsilon_{2t} - \epsilon_{2,t-1} \tag{3.22}$$

where $\{\epsilon_t\} = \{(\epsilon_{1t}, \epsilon_{2t})'\}$ is white noise with covariance matrix Σ. This corresponds to a FIVAR with $A(\lambda) = I$, $d_1 = 0, d_2 = -1$. In this case, the corresponding VAR is a multivariate white noise series. If the off-diagonal element of Σ, Σ_{21}, is non-zero, then the phase of the FIVAR is defined except at frequency 0. The coherency in this example is constant and equal to that of $\{(\epsilon_{1t}, \epsilon_{2t})\}$, since $\{x_{2t}\}$ is the result of applying a linear filter to $\{\epsilon_{2t}\}$. In this case, the group delay is given by

$\phi'(\lambda) = -1/2$. This is intuitively reasonable, since x_{1t} depends only on the contemporaneous value of the innovations while $\{x_{2t}\}$ depends on the contemporaneous value and one lag with equal weights.

3.2.4 Fractional cointegration

Another well-studied model for bivariate long-memory time series is the fractional cointegration model. In this section, we will show that the coherency of fractionally cointegrated time series is 1 and that the phase is 0 or π at frequency 0; these results agree with those of Nielsen [2004]. We will also show that the group delay of fractionally cointegrated series may be infinite at frequency 0, under certain conditions.

Two univariate time series, $\{x_t\}, \{y_t\}$, are fractionally cointegrated if $\{x_t\}$ and $\{y_t\}$ are integrated of order d_x but a linear combination, $u_t = y_t - \beta x_t$, is integrated of order $d_u < d_x$. Fractionally cointegrated time series have been discussed by a number of authors, including Robinson [1994], Robinson and Marinucci, 2003], Robinson and Hualde [2003], Chen and Hurvich [2003], and Christensen and Nielsen [2006]. Here, we focus on the phase and coherency of fractionally cointegrated time series. We will discuss the estimation of β in Section 3.4.

The Granger representation theorem [Engle and Granger, 1987] noted that cointegrated series have a spectral density matrix of lower rank at frequency 0; in the bivariate case, this implies that $\rho(0) = 1$. More recently, Nielsen [2004] described the phase and coherency of fractionally cointegrated series, assuming that $1/2 < d_x < 3/2$ and that $-1/2 < d_u < 1/2$ and that $f^{\dagger}(\lambda) = \Omega(1 + O(\lambda^2))$ as $\lambda \to 0^+$, where Ω is a constant. This rules out group delay in the fractionally differenced time series at frequency zero and power law coherency with $0 > d_\rho > -1$.

To compute the phase and coherency of fractionally cointegrated time series, we assume that $d_x < 1/2$. Since the phase and coherency of two time series are unchanged if the same linear filter (in this case, differencing) is applied to both, assuming that $d_x < 1/2$ is not restrictive. To begin, we write our time series, $\{x_t\}, \{y_t\}$ in terms of $\{x_t\}, \{u_t\}$:

$$\begin{pmatrix} 1 & 0 \\ -\beta & 1 \end{pmatrix} \begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} x_t \\ u_t \end{pmatrix}$$

By assumption, $\{x_t\}$ is $I(d_x)$, $\{u_t\}$ is $I(d_u)$, and $\beta \neq 0$. Let the spectral density of $\{(x_t, u_t)'\}$ be given by $f(\lambda) = \begin{pmatrix} f_{11}(\lambda & f_{12}(\lambda) \\ f_{21}(\lambda) & f_{22}(\lambda) \end{pmatrix}$ for $\lambda \in [-\pi, \pi]$. Let $\tilde{f}, \tilde{\rho}, \tilde{\phi}$ be the spectral density, coherency, and phase of the cointegrated time series, $\{(x_t, y_t)'\}$:

$$\begin{aligned}
\tilde{f}(\lambda) &= \begin{pmatrix} 1 & 0 \\ -\beta & 1 \end{pmatrix}^{-1} \begin{pmatrix} f_{11}(\lambda) & f_{12}(\lambda) \\ f_{21}(\lambda) & f_{22}(\lambda) \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ -\beta & 1 \end{pmatrix}^{-1} \right)' \\
&= \begin{pmatrix} f_{11}(\lambda) & f_{12}(\lambda) + \beta f_{11}(\lambda) \\ f_{21}(\lambda) + \beta f_{11}(\lambda) & \beta^2 f_{11}(\lambda) + \beta f_{21}(\lambda) + \beta f_{12}(\lambda) + f_{22}(\lambda) \end{pmatrix} \\
\tilde{\rho}(\lambda) &= \frac{|f_{12}(\lambda) + \beta f_{11}(\lambda)|}{\sqrt{f_{11}(\lambda)(\beta^2 f_{11}(\lambda) + \beta f_{21}(\lambda) + \beta f_{12}(\lambda) + f_{22}(\lambda))}} \\
\tilde{\phi}(\lambda) &= \arg(f_{21}(\lambda) + \beta f_{11}(\lambda))
\end{aligned}$$

As $\lambda \to 0^+$, $f_{11}(\lambda) \sim C_1 \lambda^{-2d_x}$ and $f_{12}(\lambda) = O(\lambda^{-d_x - d_u}) = o(\lambda^{-2d_x})$, so that the terms containing $f_{11}(\lambda)$ will dominate the expressions above, so that $\tilde{\rho}(0) = 1$ and $\tilde{\phi}_0 \in \{0, \pi\}$, with the choice of $\tilde{\phi}_0$ depending on the sign of β. To compute the

group delay:

$$\begin{aligned}
\tilde{\phi}'(\lambda) &= \left((\Re(f_{21}(\lambda)) + \beta f_{11}(\lambda))^2 + \Im(f_{21}(\lambda))^2\right)^{-1} \\
&\quad \times \Bigg((\Re(f_{21}(\lambda)) + \beta f_{11}(\lambda)) \frac{d}{d\lambda} \Im(f_{21}(\lambda)) \\
&\quad - \Im(f_{21}(\lambda)) \left(\frac{d}{d\lambda} \Re(f_{21}(\lambda)) + \beta \frac{d}{d\lambda} f_{11}(\lambda) \right) \Bigg) \\
&= \frac{f_{11}(\lambda) \frac{d}{d\lambda} \Im(f_{21}(\lambda)) - \Im(f_{21}(\lambda)) \frac{d}{d\lambda} f_{11}(\lambda)}{\beta f_{11}(\lambda)^2} \\
&\quad + o\left(\frac{f_{11}(\lambda) \frac{d}{d\lambda} \Im(f_{21}(\lambda)) - \Im(f_{21}(\lambda)) \frac{d}{d\lambda} f_{11}(\lambda)}{\beta f_{11}(\lambda)^2} \right)
\end{aligned}$$

Thus, the group delay of fractionally cointegrated series depends in part on $f_{11}(\lambda) \sim C_1 \lambda^{-2d_1}$ and $\Im(f_{21}(\lambda)) = O(\lambda^{-2d_{12}})$; note that the latter bound need not be sharp. The derivatives can have a variety of power laws in a neighborhood of zero, leading to many possible values of the group delay at frequency 0.

In the commonly assumed case where $\{(x_t, u_t)'\}$ follow a FIVAR model, we have $\Im(f_{21}(\lambda)) \sim C_{Im} \lambda^{-2d_{12}}$, $\frac{d}{d\lambda} \Im(f_{21}(\lambda)) \sim C_{Im,d} \lambda^{-2d_{12}-1}$, and $\frac{d}{d\lambda} f_{11}(\lambda) \sim C_{1,d} \lambda^{-2d_1-1}$, where $C_{Im}, C_{Im,d}, C_{1,d}$ are non-zero constants. Then, $\phi'(\lambda) \sim C \lambda^{d_x - d_u - 1}$, for some non-zero C, so that the group delay at frequency 0 is 0 when $d_u > d_x + 1$, infinite when $d_u < d_x + 1$, and finite and non-zero when $d_u = d_x + 1$. Even if the derivatives had the same properties, these results would change if $\{(x_t, u_t)'\}$ had power law coherency; $d_\rho < 0$ means that smaller differences between d_x and d_u would lead to a group delay of 0 at frequency 0. In the extreme case of $d_\rho = 1$, any cointegrated series would have a group delay of 0 between $\{x_t\}$ and $\{y_t\}$. Alternative power laws in the derivatives or cases in which $\Im(f_{21}(\lambda)) = o\left(\lambda^{-2d_{12}}\right)$ could also change the group delay at frequency 0.

3.2.5 Power law coherency

Because power law coherency has not been studied in existing literature, we present two parametric models that have power law coherency. These models are particularly simple and could be extended easily to allow for a richer variety of behavior away from zero frequency. These models show how power law coherency could occur in the time domain.

For our first bivariate time series with power law coherency, assume that $d_3 < d_2 \leq d_1$ and that $\epsilon_{1t}, \epsilon_{2t}, \epsilon_{3t}$ are independent white noise series with variances $\sigma_1^2, \sigma_2^2, \sigma_3^2$, respectively. Consider the time series model below.

$$x_{1t} = (1-L)^{-d_3}\epsilon_{3t} + (1-L)^{-d_1}\epsilon_{1t} \tag{3.23}$$

$$x_{2t} = (1-L)^{-d_3}\epsilon_{3t} + (1-L)^{-d_2}\epsilon_{2t} \tag{3.24}$$

As $\lambda \to 0^+$:

$$\begin{aligned} f_1(\lambda) &\sim \frac{\sigma_1^2}{2\pi}\lambda^{-2d_1} \\ f_2(\lambda) &\sim \frac{\sigma_2^2}{2\pi}\lambda^{-2d_2} \\ f_{12}(\lambda) &\sim \frac{\sigma_3^2}{2\pi}\lambda^{-2d_3} \\ \rho(\lambda) &\sim \frac{\sigma_3^2}{\sigma_1\sigma_2}\lambda^{-2(d_3-\frac{1}{2}(d_1+d_2))} \end{aligned}$$

In this model, $\{x_{1t}\}$ and $\{x_{2t}\}$ are long-memory time series with a common component that has a smaller memory parameter than either of the individual time series. If we instead had $d_3 > \max(d_1, d_2)$, then the two time series would be cointegrated. For this reason, we will refer to this time series model as an anti-cointegration model. In anti-cointegration (as in all cases of power law coherency), the two time series are correlated in the "short run" (for frequencies away from zero), but the strength of the relationship decays to zero at frequency zero. Thus,

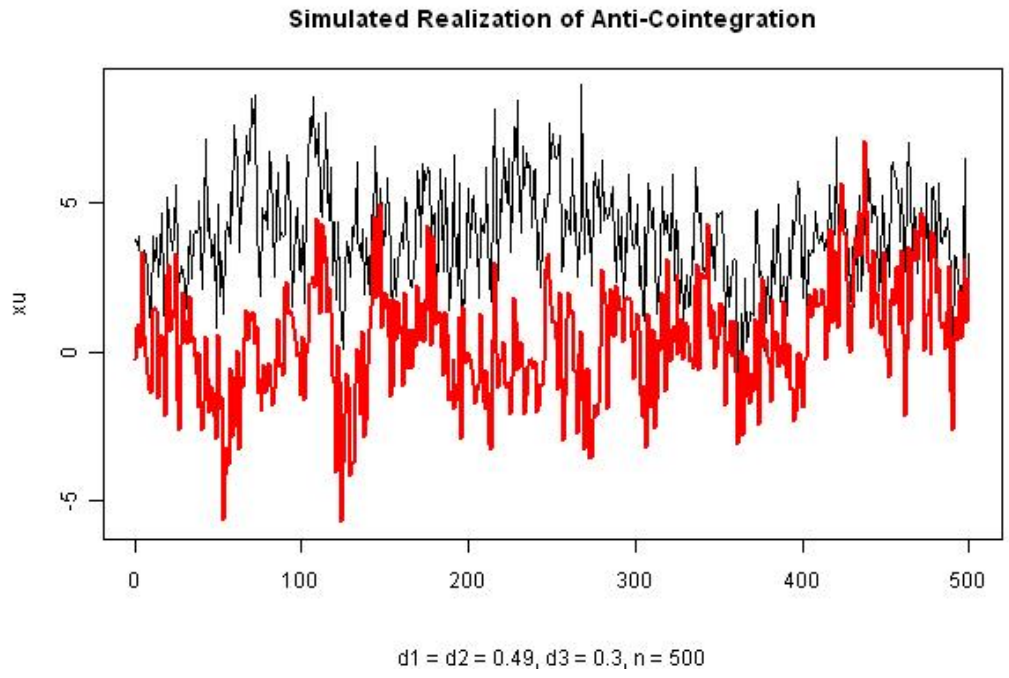

Figure 35: One simulated realization of 500 periods of the anti-cointegration model, with $d_1 = d_2 = 0.49$, and $d_3 = 0.3$.

the coherency is zero at frequency zero, instead of one at frequency zero, as occurs with cointegration. This occurs because the common component has a smaller memory parameter and is dwarfed by the more persistent idiosyncratic components at low frequencies. In this model, $\phi(\lambda) = 0$ and the group delay is zero, because the common component enters the two time series contemporaneously. A simulated realization is shown as a time series in Figure 35. The long-term movements of the time series are not strongly related, since the levels drift separately with longer memory, but the short-term movements are related.

Next, we discuss a time series model first described by Hosoya [1997, Example 2.3], in the context of quasi-log-likelihood estimation. Assume that $d_1 \neq d_2$, with

$0 < d_1 < 1/2$ and $0 < d_2 < 1/2$ and that $\{\epsilon_{1t}\}, \{\epsilon_{2t}\}$ are independent white noise series, each with variance σ^2. Consider the following two time series:

$$\begin{aligned} x_{1t} &= (1-L)^{-d_1}\epsilon_{1t} + (1-L)^{+d_2}\epsilon_{2t} \\ x_{2t} &= (1-L)^{+d_1}\epsilon_{1t} + (1-L)^{-d_2}\epsilon_{2t} \end{aligned}$$

The spectral densities and cross-spectral density of $\{X_t\}$ satisfy:

$$\begin{aligned} f_1(\lambda) &= \frac{\sigma^2}{2\pi}\left(|1-e^{-i\lambda}|^{-2d_1} + |1-e^{-i\lambda}|^{+2d_2}\right) \sim \frac{\sigma^2}{2\pi}\lambda^{-2d_1},\ \ \lambda \to 0^+ \\ f_2(\lambda) &= \frac{\sigma^2}{2\pi}\left(|1-e^{-i\lambda}|^{+2d_1} + |1-e^{-i\lambda}|^{-2d_2}\right) \sim \frac{\sigma^2}{2\pi}\lambda^{-2d_2},\ \ \lambda \to 0^+ \\ f_{12}(\lambda) &= \frac{\sigma^2}{2\pi}\left((1-e^{-i\lambda})^{+d_1}(1-e^{+i\lambda})^{-d_1} + (1-e^{-i\lambda})^{-d_2}(1-e^{+i\lambda})^{+d_2}\right) \end{aligned}$$

for $\lambda \in [-\pi, \pi]$. For this time series, $\{x_{jt}\}$ is $I(d_j)$ for $j = 1, 2$, and $\lim_{\lambda\to 0}|f_{12}(\lambda)|$ exists and is finite. Thus, the coherency has a power law with differencing parameter equal to $-\frac{d_1+d_2}{2}$. Despite the finiteness of the limit of the absolute value of the cross-spectrum, the phase is discontinuous at 0, so $f_{12}(\lambda)$ is not continuous at 0. Specifically, we have:

$$\begin{aligned} f_{12}(\lambda) &= \frac{\sigma^2}{2\pi}\left((1-e^{-i\lambda})^{d_1}(1-e^{i\lambda})^{-d_1} + (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{d_2}\right) \\ &= \frac{\sigma^2}{2\pi}\left(\exp(id_1(\pi-\lambda)) + \exp(-id_2(\pi-\lambda))\right) \\ \lim_{\lambda\to 0^+} f_{12}(\lambda) &= \frac{\sigma^2}{\pi}\exp\left(\frac{i\pi(d_1-d_2)}{2}\right) \end{aligned}$$

Thus, whenever $d_1 \neq d_2$, the phase is discontinuous at 0, even though the absolute cross-spectral density is neither zero nor infinite. However, $f_{12}^{\dagger}(0) = 0$ and therefore $f_{12}^{\dagger}(\lambda)$ is continuous at 0. The phase and group delay are given by:

$$\begin{aligned} \phi(\lambda) &= \arg\left(\exp(id_1(\pi-\lambda)) + \exp(-id_2(\pi-\lambda))\right) \\ &= \frac{(d_1-d_2)(\pi-\lambda)}{2} \\ \phi'(\lambda) &= \frac{d_2-d_1}{2} \end{aligned}$$

Thus, the group delay depends on the difference of the differencing parameters.

3.2.6 Powers of the frequency in the phase

We have already seen that powers of λ can occur in the phase in some cases of fractionally cointegrated time series. Here, we describe some other time series that are not fractionally cointegrated but do have that property.

As an example of a parametric time series model, assume that $d_1, d_2, d_3 < 1/2$, that $0 < d_2 - d_3 < 1$, that $d_1 \neq d_3$, and that $\{\epsilon_{1t}\}, \{\epsilon_{2t}\}$ are uncorrelated white noise with variances σ_1^2, σ_2^2 respectively:

$$x_{1t} = (1-L)^{-d_1}\epsilon_{1t} + (1-L)^{-d_1}\epsilon_{2t} \tag{3.25}$$

$$x_{2t} = (1-L)^{-d_2}\epsilon_{1t} + (1-L)^{-d_3}\epsilon_{1t} \tag{3.26}$$

These two time series are associated with the transfer function:

$$\Psi(\lambda) = \begin{pmatrix} (1-e^{-i\lambda})^{-d_1} & (1-e^{-i\lambda})^{-d_1} \\ (1-e^{-i\lambda})^{-d_2} + (1-e^{-i\lambda})^{-d_3} & 0 \end{pmatrix}$$

for $\lambda \in [-\pi, \pi]$. Notice that the coherency of $\{\epsilon_{1t} + \epsilon_{2t}\}$ and $\{\epsilon_{1t}\}$ is constant and that $\{x_{1t}\}$ and $\{x_{2t}\}$ can be obtained by applying the linear filters $(1-L)^{-d_1}$ and $(1-L)^{-d_1} + (1-L)^{-d_2}$, respectively, to those series. Thus, the coherency of these two time series is constant and equal to:

$$\rho(\lambda) \equiv \frac{\sigma_1^2}{\sqrt{\sigma_1^2 + \sigma_2^2}}$$

for $\lambda \in [-\pi, \pi]$. We compute the cross-spectral density to show that the group delay is infinite at zero frequency. For $\lambda \in (0, \pi]$:

$$f_{12}(\lambda) = \frac{\sigma_1^2}{2\pi}\left|2\sin\left(\frac{\lambda}{2}\right)\right|^{-d_1-d_2}\left(e^{i(\pi-\lambda)(d_1-d_2)/2} + \left|2\sin\left(\frac{\lambda}{2}\right)\right|^{d_2-d_3} e^{i(\pi-\lambda)(d_1-d_3)/2}\right)$$

$$\phi(\lambda) = \arctan\left(\frac{\sin\left(\frac{1}{2}(\pi-\lambda)(d_1-d_2)\right) + \left|2\sin\left(\frac{\lambda}{2}\right)\right|^{d_2-d_3}\sin\left(\frac{1}{2}(\pi-\lambda)(d_1-d_3)\right)}{\cos\left(\frac{1}{2}(\pi-\lambda)(d_1-d_2)\right) + \left|2\cos\left(\frac{\lambda}{2}\right)\right|^{d_2-d_3}\sin\left(\frac{1}{2}(\pi-\lambda)(d_1-d_3)\right)}\right)$$

$$\phi'(\lambda) \sim |\lambda|^{d_2-d_3-1}\tan\left(\frac{1}{2}(\pi-\lambda)(d_1-d_3)\right), \ \lambda \to 0^+$$

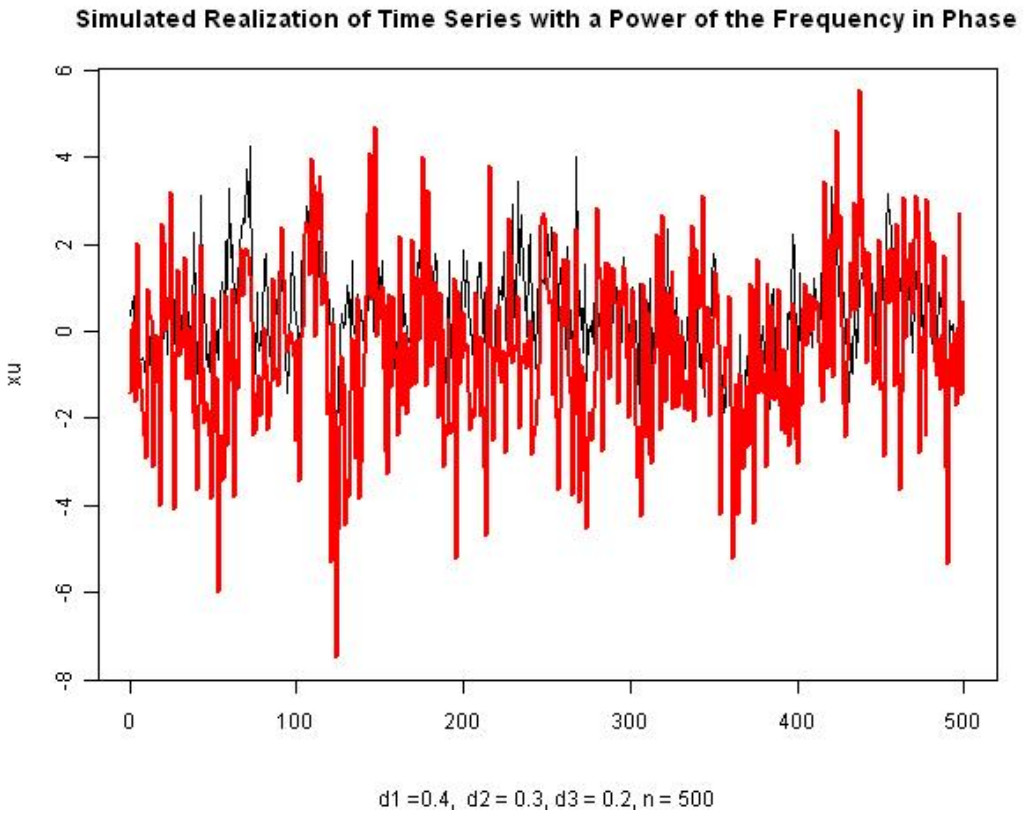

Figure 36: One simulated realization of 500 periods of the model with powers of λ in the phase, with $d_1 = 0.4$, $d_2 = 0.3$, and $d_3 = 0.2$.

Thus, $\phi(\lambda)$ obeys the condition in Equation (3.3) with $\alpha = d_2 - d_3$. One realization of this time series is shown in Figure 36.

As another example of powers of λ appearing in the phase, without an explicit time domain model, suppose that the cross-covariances of a bivariate time series have different decay rates for positive lags and for negative lags. Specifically, assume that the cross-covariances follow:

$$r_{XY}(j) \sim \begin{cases} C_l|j|^{2d_l-1} & j < 0 \\ C_r|j|^{2d_r-1} & j > 0 \end{cases}$$

with $C_l, C_r > 0$ and $0 < d_r < d_l < 1/2$. Applying Equation (2.1) of Zygmund and Fefferman [2002, Chapter V, Section 2], we find that, as $\lambda \to 0^+$:

$$\begin{aligned}
f_{XY}(\lambda) &= \frac{1}{2\pi}\sum_{j=-\infty}^{\infty} r_{XY}(j)e^{-ij\lambda} \\
&\sim \frac{1}{2\pi}\left(\sum_{j=-\infty}^{-1} C_l|j|^{2d_l-1}e^{-ij\lambda} + \sum_{j=1}^{\infty} C_r|j|^{2d_r-1}e^{-ij\lambda}\right) \\
&= \frac{1}{2\pi}\sum_{j=1}^{\infty}\Big(j^{2d_l-1}\cos(j\lambda) + j^{2d_r-1}\cos(j\lambda) + ij^{2d_l-1}\sin(j\lambda) + ij^{2d_r-1}\sin(j\lambda)\Big) \\
&\sim \frac{1}{2\pi}\Big(\lambda^{-2d_l}\Gamma(2d_l)\sin\left(\frac{\pi}{2}(1-2d_l)\right) + \lambda^{-2d_r}\Gamma(2d_r)\sin\left(\frac{\pi}{2}(1-2d_r)\right) \\
&\quad + i\lambda^{-2d_l}\Gamma(2d_l)\cos\left(\frac{\pi}{2}(1-2d_l)\right) + i\lambda^{-2d_r}\Gamma(2d_r)\cos\left(\frac{\pi}{2}(1-2d_r)\right)\Big)
\end{aligned}$$

Then, the phase and group delay are given by:

$$\begin{aligned}
\phi(\lambda) &= \arg\left(\lambda^{-2d_l}\Gamma(2d_l)e^{i\left(\frac{\pi}{2}(1-2d_l)\right)} + \lambda^{-2d_r}\Gamma(2d_r)e^{i\left(\frac{\pi}{2}(1-2d_r)\right)}\right) \\
&= \arg\left(\Gamma(2d_l)e^{i\left(\frac{\pi}{2}(1-2d_l)\right)} + \lambda^{2d_l-2d_r}\Gamma(2d_r)e^{i\left(\frac{\pi}{2}(1-2d_r)\right)}\right) \\
\phi'(\lambda) &= \Big(\Big(\Gamma(2d_l)\sin\left(\frac{\pi}{2}(1-2d_l)\right) + \lambda^{2d_l-2d_r}\Gamma(2d_r)\sin\left(\frac{\pi}{2}(1-2d_r)\right)\Big)^2 \\
&\quad + \Big(\Gamma(2d_l)\cos\left(\frac{\pi}{2}(1-2d_l)\right) + \lambda^{2d_l-2d_r}\Gamma(2d_r)\cos\left(\frac{\pi}{2}(1-2d_r)\right)\Big)^2\Big)^{-1} \\
&\quad \times\Big[\Big(\Gamma(2d_l)\sin\left(\frac{\pi}{2}(1-2d_l)\right) + \lambda^{2d_l-2d_r}\Gamma(2d_r)\sin\left(\frac{\pi}{2}(1-2d_r)\right)\Big)(2d_l-2d_r) \\
&\quad \times\lambda^{2d_l-2d_r-1}\Gamma(2d_r)\cos\left(\frac{\pi}{2}(1-2d_r)\right) \\
&\quad - \Big(\Gamma(2d_l)\cos\left(\frac{\pi}{2}(1-2d_l)\right) + \lambda^{2d_l-2d_r}\Gamma(2d_r)\cos\left(\frac{\pi}{2}(1-2d_r)\right)\Big)(2d_l-2d_r) \\
&\quad \lambda^{2d_l-2d_r-1}\Gamma(2d_r)\sin\left(\frac{\pi}{2}(1-2d_r)\right)\Big] \\
&\sim \frac{\Gamma(2d_r)(\cos\left(\frac{\pi}{2}(1-2d_r)\right) - \sin\left(\frac{\pi}{2}(1-2d_r)\right)}{\Gamma(2d_l)^2}\lambda^{2d_l-2d_r-1}
\end{aligned}$$

This provides another case of powers in the phase leading to group delay that is infinite at 0.

3.2.7 Powers in the phase and coherency

Finally, we consider a bivariate time series with powers of λ in both the phase and the coherency. Similar to the time-domain example with a power law in only the

phase, assume that $d_2 < d_1 < d_4 < 1/2$, that $d_3 < 1/2$, that $d_1 - d_2 < 1$, and that $\{\epsilon_{1t}\}, \{\epsilon_{2t}\}$ are uncorrelated white noise with variances σ_1^2, σ_2^2 respectively:

$$x_{1t} = \left((1-L)^{-d_1} + (1-L)^{-d_2}\right)\epsilon_{1t} + (1-L)^{-d_4}\epsilon_{2t} \quad (3.27)$$

$$x_{2t} = (1-L)^{-d_3}\epsilon_{1t} \quad (3.28)$$

This bivariate time series is associated with the transfer function:

$$\Psi(\lambda) = \begin{pmatrix} (1-e^{-i\lambda})^{-d_1} + (1-e^{-i\lambda})^{-d_2} & (1-e^{-i\lambda})^{-d_4} \\ (1-e^{-i\lambda})^{-d_3} & 0 \end{pmatrix}$$

for $\lambda \to 0^+$. Based on the transfer function, we find that the auto-spectra satisfy the following, as $\lambda \to 0^+$:

$$f_1(\lambda) \sim \frac{\sigma_2^2}{2\pi}\lambda^{-2d_4}$$

$$f_2(\lambda) \sim \frac{\sigma_1^2}{2\pi}\lambda^{-2d_3}$$

For $0 < \lambda < \pi$, the cross-spectrum is given by:

$$\begin{aligned} f_{12}(\lambda) &= \frac{\sigma_1^2}{2\pi}\left((1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_3} + (1-e^{-i\lambda})^{-d_2}(1-e^{i\lambda})^{-d_3}\right) \\ &= \frac{\sigma_1^2}{2\pi}(1-e^{-i\lambda})^{-d_1}(1-e^{i\lambda})^{-d_3}\left(1 + (1-e^{-i\lambda})^{d_1-d_2}\right) \end{aligned}$$

so that, as $\lambda \to 0^+$, the coherency and phase satisfy:

$$\begin{aligned} \rho(\lambda) &\sim \frac{\sigma_1}{\sigma_2}\lambda^{d_1-d_4} \\ \phi(\lambda) &= \frac{(\pi-\lambda)(d_3-d_1)}{2} + \arctan\left(\frac{\left|2\sin\frac{\lambda}{2}\right|^{d_1-d_2}\sin\left(\frac{(\pi-\lambda)(d_2-d_1)}{2}\right)}{1+\left|2\sin\frac{\lambda}{2}\right|^{d_1-d_2}\cos\left(\frac{(\pi-\lambda)(d_2-d_1)}{2}\right)}\right) \\ &= \frac{\pi(d_3-d_1)}{2} + \sin\left(\frac{\pi(d_2-d_1)}{2}\right)\lambda^{d_1-d_2} + o(\lambda^{d_1-d_2}) \end{aligned}$$

Thus, the coherency has a power law and the phase includes powers of λ. Furthermore, the power laws differ and do not depend on each other, since d_2 and d_4 can be varied separately.

3.3 Estimating the phase and coherency in a neighborhood of zero

Next, we will estimate d_{12} and d_ρ in a semiparametric framework. Semiparametric estimation methods are generally based on the periodogram matrix, defined as $I(\lambda_j) = J(\lambda_j)J(\lambda_j)^*$, where $\lambda_j = \frac{2\pi j}{n}$ is the j^{th} Fourier frequency and $J(\lambda_j) = \frac{1}{\sqrt{2\pi n}} \sum_{t=1}^{n} X_t e^{i\lambda_j t}$ is the discrete Fourier transform of the bivariate series. Most estimation methods use frequencies near 0; specifically, they use m frequencies where $m \to \infty$ and $\frac{m}{n} \to 0$.

In some cases, d_1 or d_2 may be less than $-1/2$; this may occur because the series have been differenced because of possible non-stationarity or to remove a trend. In this case, the raw periodogram or cross-periodogram is not a good estimator of the spectral density because of leakage; thus, authors such as Velasco [1999], Hurvich and Chen, and Hurvich et al. [2002] recommend tapering the series before computing the cross-periodogram. We will apply the taper of Hurvich and Chen. Using the notation of Hurvich et al. [2002], the taper is based on the sequence:

$$h_t \;=\; (1 - e^{i2\pi t/n}), t = 1, ..., n \tag{3.29}$$

If we assume that $d_1, d_2, d_{12} \in \left(-\frac{1}{2} - s, \frac{1}{2}\right)$, perhaps because the $\{X_t\}$ have been differenced s times, we must first taper the data of order s. In that case, we use the multivariate time series with (j,t) element equal to $h_t^s x_{jt}$. Then, the tapered discrete Fourier transform is $J(\lambda_j) = \frac{1}{\sqrt{2\pi n a_s}} \sum_{t=1}^{n} h_t^s X_t e^{it\lambda_j}$, with $a_s = \binom{2s}{s} = \frac{1}{n}\sum_{t=1}^{n} |h_t|^{2s}$, and the tapered periodogram is $I(\lambda_j) = J(\lambda_j)J(\lambda_j)^*$. This tapering reduces the leakage at the low frequencies and therefore reduces the bias of estimators based on the periodogram.

3.3.1 Previous estimation work for multivariate long-memory models

Extensive work has been published on estimating the spectral density, phase, and coherency at a point where the auto- and cross-spectral densities are smooth. The most common approaches first smooth the periodogram matrix and then estimate the phase and coherency based on Equation (3.1). More information and a variety of possible smoothers can be found in Bloomfield [1976, Section 10.2], Brillinger [1981, Section 7.3] or Priestley [1981, Section 9.5]. Even estimation methods that are not based directly on the smoothed periodogram, such as those of Hannan and Thomson [1973] and Granger and Hatanaka [1964, Chapter 6], require that the coherency and phase are smooth and continuous. Hidalgo [1996] shows that the phase and coherency can be consistently estimated at any frequency where $f(\lambda)$ has two continuous derivatives, even when there are singularities elsewhere in the spectrum. Thus, the coherency and phase of a bivariate long memory process can be estimated away from zero frequency if we assume that they are twice differentiable away from frequency 0.

However, the required smoothness assumptions are violated by long-memory processes in a neighborhood of zero. Even when the auto-spectra are finite and non-zero at zero, power law coherency could lead to a violation of this assumption for the cross-spectra. Furthermore, the variability of the estimated phase is quite large when the coherency is close to zero [see, for example, Brockwell and Davis, 1993, equation 11.7.9], meaning that power law coherency presents an additional challenge to estimating the phase or group delay at zero. Though the smoothed periodogram has many shortcomings, we will use it in data analysis in Section 3.5 to provide a graphical description of the phase and coherency away from frequency zero, where it is consistent.

Many methods exist for estimating the memory parameter of a univariate long-

memory time series based on the periodogram. One might consider applying similar techniques to the cross-periodogram matrix. Unfortunately, some of these methods will fail for estimating the memory parameter associated with the cross-spectral density and therefore for detecting a power law in the coherency. The most problematic method is that of Geweke and Porter-Hudak [1983], called the GPH estimator. In the univariate case, they fit the linear regression:

$$\log I_{11}(\lambda_j) \quad = \quad \beta_0 - 2d \log \lambda_j + \epsilon_j$$

In the multivariate case, it is always true that:

$$|I_{12}(\lambda_j)|^2 \quad = \quad |I_{11}(\lambda_j)| \times |I_{22}(\lambda_j)|$$

Thus, applying the GPH estimator to the modulus of the cross-periodogram would simply estimate $\frac{1}{2}(d_1+d_2)$. Robinson [1995a] applies the GPH estimator in a multivariate context, but only to estimate and test hypotheses about the memory parameters of the auto-spectra. His Assumption 3 allows for power law coherency, which leads to convergence rates that are worst when d_ρ is close to but not equal to 0, as discussed in Section 3.2.1. One could consider smoothing the cross-periodogram first and then applying the GPH estimator, as was suggested by Reisen [1994] in a univariate context; we leave the exploration of this idea for future research.

Another common semiparametric estimation technique for long-memory time series is the local Whittle estimator (also known as the Gaussian semiparametric estimator), used by Robinson [1995b] in the univariate case, by Lobato [1999] and Shimotsu [2007] in the multivariate case, and by Robinson [2008] in the multivariate case of cointegration. The local Whittle estimator requires the specification of the spectral density locally. For example, Shimotsu [2007, Assumption 1'] requires that, as $\lambda \to 0^+$:

$$f_{12}(\lambda) - e^{i(\pi-\lambda)(d_1-d_2)} \lambda^{-d_1-d_2} G_{12} = O(\lambda^{-d_1-d_2+\xi})$$

where G_{12} is a constant. When ξ is close to 0, fewer frequencies can be used in estimation, so that the convergence rate is slower. In the case of power law coherency, $G_{12} = 0$ and $\xi \leq -2d_\rho$ in the expression above. If there are powers of λ in the phase, G_{12} is not restricted, and $\xi \leq \alpha$. As Deo and Hurvich [2001] note in the context of the GPH estimator, including higher order terms in the semiparametric description of the spectral density can improve the performance of such local estimators. (Robinson [2008] does this in the context of cointegration, as we will discuss in Section 3.4.) For example, one could apply local Whittle using the parameterizations for the coherency and phase given in Equations (3.2) and (3.3). Unfortunately, this increases the number of parameters in the model, and it is generally unknown how many terms of the spectral density should be used. In addition, estimation of the phase in the context of power law coherency continues to be problematic because the coherency is close to 0 in the range of frequencies of interest.

3.3.2 The averaged cross-periodogram estimator

To estimate the power law in the coherency and the right-hand limit of the phase at zero, we adapt the averaged periodogram estimator (APE) of Robinson [1994], which he applied in the univariate case when $0 < d < \frac{1}{2}$. He assumes that, as $\lambda \to 0^+$, $f(\lambda) \sim L(1/\lambda)\lambda^{-2d}$, where $L(x)$ is a function that is slowly varying at infinity. He begins by estimating $F(\lambda) = \int_0^\lambda f(\theta)d\theta$ using the averaged periodogram:

$$\hat{F}(\lambda) \;=\; \frac{2\pi}{n} \sum_{j=1}^{[n\lambda/2\pi]} I(\lambda_j)$$

Omitting $I(0)$ removes the effects of the mean. In his Theorem 1 (page 520), he shows that, under certain conditions, $\hat{F}(\lambda_m)/F(\lambda_m) \to_p 1$ as $n \to \infty$ and

$\frac{1}{m} + \frac{m}{n} \to 0$. Second, he uses the fact that, for any q, as $\lambda \to 0^+$,

$$\begin{aligned} F(\lambda) &\sim L\left(\frac{1}{\lambda}\right)\frac{\lambda^{1-2d}}{1-2d} \\ \frac{F(q\lambda)}{F(\lambda)} &\sim q^{1-2d}\frac{L(\frac{1}{q\lambda})}{L(\frac{1}{\lambda})} \\ &\sim q^{1-2d} \end{aligned}$$

to define the averaged periodogram estimator of the memory parameter of a univariate series:

$$\hat{d} = \frac{1}{2} - \frac{\log(\hat{F}(q\lambda_m)/\hat{F}(\lambda_m))}{2\log q}$$

where $q \in (0,1)$ and m must be chosen by the user. Lobato and Robinson [1996] derived additional results about the limiting distribution of a suitably standardized version of $\hat{d}$ under various conditions, showing that it is normal when $0 < d < 1/4$ and non-normal for $1/4 < d < 1/2$. Lobato [1997] applied the averaged periodogram estimator to estimating $f^\dagger(0)$ in the multivariate case. While he did note (on page 139) the possibility of power law coherency, he was focused on estimating $f^\dagger(0)$ and the memory parameters of the auto-spectra, not a power law in the coherency in a neighborhood of 0; in his Condition C1, he required that $\rho(0) > 0$. In contrast, we are particularly interested in estimating the power law of the cross-spectral density. Thus, we extend the consistency results of Robinson [1994] and Lobato [1997] in two ways: to the case in which $d < 0$ and to the estimation of d_{12}. In order to do this, we first extend the definition of a slowly-varying function to the case where $L(z)$ may be complex-valued.

Definition 3.6 *Let $L : \mathbb{R}^+ \to \mathbb{C}$. We say that $L(z)$ is* slowly varying at infinity *if $L(z)$ is bounded away from 0 for sufficiently large z and for all $t > 0$:*

$$\lim_{z\to\infty} \frac{L(tz)}{L(z)} = 1$$

A complex-valued function, $g(z)$, is regularly varying at infinity *if $g(z) = L(z)z^a$, for some $a \in \mathbb{R}$, with $L(z)$ is slowly varying at infinity. A complex-valued function, $f(z)$, is* regularly varying at zero *if $g(z) = f(1/z)$ is regularly varying at infinity.*

Definition 3.7 *Let $f(\lambda), g(\lambda)$ be complex-valued functions for $\lambda \in (0, \pi)$. We say that*

$$f(\lambda) \sim g(\lambda)$$

as $\lambda \to 0^+$ if

$$\lim_{\lambda \to 0^+} \frac{f(\lambda)}{g(\lambda)} = 1$$

Any function that has a limit as $z \to \infty$ is slowly varying at infinity. Since the assumptions given in Section 3.2 imply that $f_{12}(\lambda)\lambda^{+2d_{12}}$ has a non-zero limit as $\lambda \to 0^+$, $f_{ab}(\lambda)$ is regularly varying at zero for $a, b = 1, 2$.

The lemma below extends Karamata's Theorem, a result on real regularly varying functions. (Vuilleumier [1976] discusses the properties of complex functions of complex variables, but those results are not relevant in this case.)

Lemma 3.8 *Suppose $f_{12}(\lambda) = L\left(\frac{1}{\lambda}\right)\lambda^{-2d_{12}}$ is a complex-valued regularly varying function at 0. Define $F_{12}(\lambda) = \int_0^\lambda f_{12}(\theta)d\theta$. Then, $F_{12}(\lambda) \sim \frac{1}{1-2d_{12}} L\left(\frac{1}{\lambda}\right)\lambda^{1-2d_{12}}$ as $\lambda \to 0^+$.*

In order to prove some results about the averaged periodogram estimator (with or without tapering), we require a mild additional assumption about the product $\tau_{jk}(\lambda)e^{i\varphi_{jk}(\lambda)}$. (This assumption is based on Definition 2 of Hurvich et al. [2002, page 316], but we use the letter γ instead of ρ.) This assumption requires that $\Psi^\dagger_{jk}(\lambda)$ is well-behaved away from frequency 0. As Chen and Hurvich [2006, page 2948] note, the fact that only frequencies in a neighborhood of zero frequency are used may allow for more general behavior away from 0.

Definition 3.9 *For some $\mu > 1, \gamma \in (1, 2]$, let $\mathcal{L}^*(\mu, \gamma)$ be the set of continuously differentiable functions, $u(\lambda)$, on $[-\pi, \pi] - \{0\}$, such that, for all $0 < |x|, |y| < \pi$,*

$$\begin{aligned}
\frac{\max_{0\le z\le\pi}|u(z)|}{\min_{0\le z\le\pi}|u(z)|} &\le \mu \\
\frac{|u(x)-u(y)|}{\min_{0\le z\le\pi}|u(z)|} &\le \mu\frac{|y-x|}{\min(|x|,|y|)} \\
\frac{|u'(x)-u'(y)|}{\min_{0\le z\le\pi}|u(z)|} &\le \mu\frac{|y-x|^{\gamma-1}}{[\min(|x|,|y|)]^{\gamma}}
\end{aligned}$$

Assumption 3.10 *For all j, k, either $\tau_{jk}(\lambda) = 0$ for all $\lambda \in [0, \pi]$, or $\tau_{jk}(\lambda)e^{i\varphi_{jk}(\lambda)} \in \mathcal{L}^*(\mu, \gamma)$ for some $\mu > 1, \gamma \in (1, 2]$.*

This assumption restricts the behavior of the phase and coherency away from zero frequency. The examples given in previous sections all have $\gamma = 2$.

Finally, we will consider consistency results under two different assumptions about the growth of the number of frequencies used in estimation as the sample size grows. The first is standard [for example, Robinson, 1994, Condition B] and is sufficient for the estimation of the memory parameters of the auto-spectra.

Assumption 3.11

$$\frac{1}{m} + \frac{m}{n} \to 0$$

as $n \to \infty$.

Assumption 3.12 *As $n \to \infty$:*

- *If $d_1 + d_2 > \frac{1}{2}$,*
$$\frac{n^{\frac{-4d_\rho}{2-4d_{12}}}}{m} \to 0$$
- *If $1 - \frac{\gamma}{2} < d_1 + d_2 \le \frac{1}{2}$,*
$$\frac{n^{\frac{-2d_\rho}{1-2d_{12}}}}{m} \to 0$$

- *If* $d_1 + d_2 < 1 - \frac{\gamma}{2}$,

$$\frac{n^{\frac{-2d_\rho}{\gamma/2-2d_\rho}}}{m} \to 0$$

Notice that the growth rates above change continuously, since $\frac{-4d_\rho}{2-2d_{12}} = \frac{-2d_\rho}{1-2d_{12}}$ when $d_1 + d_2 = \frac{1}{2}$ and $\frac{-2d_\rho}{1-2d_{12}} = \frac{2d_\rho}{2d_\rho-\gamma/2}$ when $d_1 + d_2 = 1 - \frac{\gamma}{2}$. The required growth rates for three choices of $d_1 + d_2$ are shown in Figure 37. Unlike most assumptions on the growth rate of m in the context of long memory, Assumption 3.12 requires a lower bound on the growth rate of m. (Hurvich et al. [2005, Equation (3.9)] is one other paper that requires a lower bound on the growth rate of m.) In practice, this assumption is likely to be problematic because it depends on d_{12} and d_ρ, which are what we are trying to estimate. Larger growth rates of m are generally associated with increased finite-sample bias in estimation. Furthermore, theorems establishing limiting normality of the estimated memory parameter generally require that the growth rate of m be bounded above, with a tighter bound when the spectral density is less smooth. [See, for example, Lobato and Robinson, 1996, Condition C3.] These opposing requirements are likely to cause problems for the averaged periodogram estimator for the cross-spectral density when d_ρ is very negative. The next assumption requires that the data have been tapered to a high enough order.

Assumption 3.13 *We assume that* $d_1, d_2 \in \left(-\frac{1}{2} - s, \frac{1}{2}\right)$*, for some non-negative integer,* s*, and that the data are tapered of order* s*.*

Theorem 3.14 *If* $f_{ab}(\lambda)$ *is a regularly varying complex-valued function at 0 with* $d_a, d_b < 1/2$ *for* $a, b \in \{1, 2\}$ *and Assumptions 3.10, 3.11, and 3.13 hold,*

$$\hat{F}_{ab}(\lambda_m) - F_{ab}(\lambda_m) \quad = \quad o_p(\lambda_m^{1-d_a-d_b}) \tag{3.30}$$

If $a \neq b$ *and we also assume that Assumption 3.12 holds as well, then we also have:*

$$\hat{F}_{ab}(\lambda_m) - F_{ab}(\lambda_m) \quad = \quad o_p(\lambda_m^{1-2d_{ab}}) \tag{3.31}$$

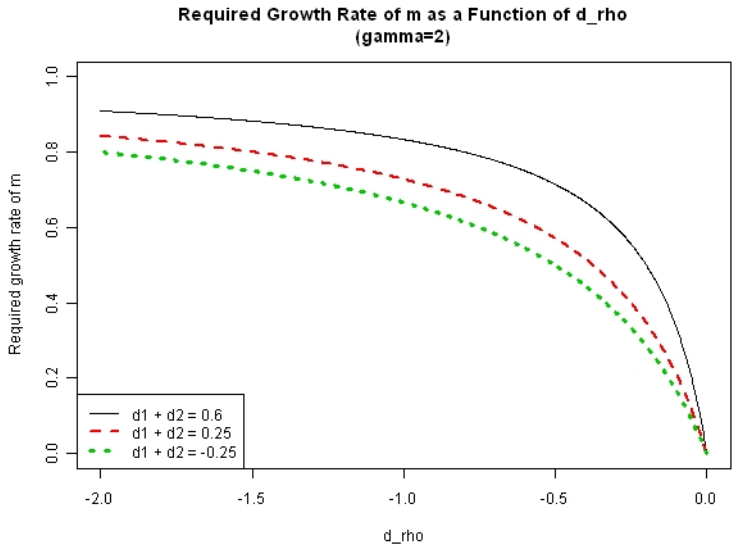

Figure 37: Minimum growth rate of m required by Assumption 3.12 as a function of d_ρ.

The theorem applies to the estimation of the integrated auto-spectrum; this extends the result of Robinson [1994] to the case where $d_a < 0$ and tapering may be used. In that case, the second part of the theorem is irrelevant and no lower bound on m is necessary. When the theorem is applied to the estimation of the cross-spectrum, power law coherency will affect the choice of m, with a more negative d_ρ placing a more stringent requirement on m.

Proof. As in Chen and Hurvich [2006], define $\tilde{j} = j + \frac{s}{2}$ to be the shifted Fourier frequency. Define $I_\epsilon(\lambda)$ to be the periodogram of ϵ_t, with (a, b) element $I_{\epsilon,ab}(\lambda)$. Let $\Psi_a(\lambda)$ be the a^{th} row of $\Psi(\lambda)$. Generalizing the proofs of Robinson [1994, Theorem 1] and Lobato [1997, Theorem 1], we decompose the difference between

the estimated averaged periodogram and the true averaged periodogram as:

$$\hat{F}_{ab}(\lambda_m) - F_{ab}(\lambda_m) = \frac{2\pi}{n}\sum_{j=1}^{m}\left(I_{ab}(\lambda_j) - \Psi_a(\lambda_{\tilde{j}})' I_\epsilon(\lambda_j)\overline{\Psi}_b(\lambda_{\tilde{j}})\right) \quad (3.32)$$

$$+\frac{2\pi}{n}\sum_{j=1}^{m}\left(\Psi_a(\lambda_{\tilde{j}})' I_\epsilon(\lambda_j)\overline{\Psi}_b(\lambda_{\tilde{j}}) - f_{ab}(\lambda_j)\right) \quad (3.33)$$

$$+\frac{2\pi}{n}\sum_{j=1}^{m} f_{ab}(\lambda_j) - F_{ab}(\lambda_m) \quad (3.34)$$

Lemma 3.22 shows that the first term is $o(\lambda_m^{-d_{aa}-d_{bb}})$ without Assumption 3.12 and $o(\lambda_m^{-2d_{ab}})$ if we do require Assumption 3.12. Lemma 3.24 shows the same for the second term. Lemma 3.25 shows that the last term is $o(\lambda_m^{-2d_{ab}})$ and therefore $o(\lambda_m^{-d_{aa}-d_{bb}})$. ■

The result in Equation (3.36) can be used directly in Theorem 3 of Robinson [1994] to show that the averaged periodogram estimator is consistent for the memory parameter of the cross-spectral density under Assumption 3.12. As before, we estimate the memory parameter of the cross-spectral density by using the fact that:

$$F_{12}(\lambda) \sim L\left(\frac{1}{\lambda}\right)\frac{\lambda^{1-2d_{12}}}{1-2d_{12}}$$

Because $L(1/\lambda)$ is complex, there are two possible ways to estimate d_{12} based on $F_{12}(\lambda)$. First, one could apply Robinson's formula to the modulus of $F_{12}(\lambda)$. Second, one could apply the formula separately for the real and imaginary parts of $F_{12}(\lambda)$ and take a weighted average based on ϕ_0. We recommend using the modulus; simulations have shown that it performs somewhat better than taking an average, especially since the optimal weights are likely to be unknown.

Theorem 3.15 *Define*

$$\hat{d}_{ab} = \frac{1}{2} - \frac{\log\left(\left|\hat{F}(q\lambda_m)\right| / \left|\hat{F}(\lambda_m)\right|\right)}{2\log q}$$

For $a = b$, if $f_{ab}(\lambda)$ is a regularly varying complex-valued function at 0 with $d_a < 1/2$ and Assumptions 3.10, 3.11, and 3.13 hold,

$$\hat{d}_{aa}(\lambda_m) - d_{aa} \;=\; o_p(1) \tag{3.35}$$

For $a \neq b$, if we also assume that Assumption 3.12 holds, then:

$$\hat{d}_{ab}(\lambda_m) - d_{ab} \;=\; o_p(1) \tag{3.36}$$

We may then estimate $\hat{d}_\rho = \hat{d}_{12} - \frac{1}{2}(\hat{d}_1 + \hat{d}_2)$, where $\hat{d}_{12}, \hat{d}_1, \hat{d}_2$ are estimated using the averaged periodogram estimator; thus $\hat{d}_\rho$ is consistent. Alternatively, $\hat{d}_1, \hat{d}_2$ could be estimated with another estimator that is consistent for univariate memory parameters. However, the convergence rate of $\hat{d}_\rho$ will generally depend on the worst convergence rate of the three estimators. To be precise, suppose that $\hat{d}_{12} = O_p(n^{-\alpha_{12}})$ and $\hat{d}_j = O_p(n^{-\alpha_j})$; then, $\hat{d}_\rho = O_p(n^{-\min(\alpha_{12},\alpha_1,\alpha_2)})$. As we will see in simulations in the next section and in Section 3.5.1, this can lead to large variability in small samples.

The averaged periodogram can be applied to estimating the jump in the phase at 0 as well as the memory parameters of the auto- and cross-spectra. If the right-hand limit of $\phi(\lambda)$ is ϕ_0 and Equation (3.36) holds, then

$$\begin{aligned}
F_{12}(\lambda) &= \int_0^\lambda f_{12}(\theta)d\theta \\
&= \int_0^\lambda |f_{12}(\theta)| \cos(\phi(\theta))d\theta + i\int_0^\lambda |f_{12}(\theta)| \sin(\phi(\theta))d\theta \\
&\approx \cos(\phi_0)\int_0^\lambda |f_{12}(\theta)|d\theta + i\sin(\phi_0)\int_0^\lambda |f_{12}(\theta)|d\theta \\
\arg(F_{12}(\lambda)) &\to \phi_0, \quad \lambda \to 0^+
\end{aligned}$$

This suggests a simple estimator for the jump in the phase:

$$\hat{\phi}_0 \;=\; \arg(\hat{F}_{12}(\lambda_m)) \tag{3.37}$$

Notice that this estimator requires the user to choose m but not q, since only one periodogram ordinate is needed for estimation. Robinson [2008, Remark 3, page 2518] suggested this estimator of the phase as well.

Corollary 3.16 *Under Assumptions 3.10, 3.12, and 3.13, if $\phi_0 \neq \pi$, then*

$$\hat{\phi}_0 - \phi_0 = o(1)$$

Proof. Since $\hat{F}_{12}(\lambda_m) \to_p F_{12}(\lambda_m)$ by Theorem 3.14 and $Arg(x)$ is continuous in x when $x \neq \pi$, we may apply the continuous mapping theorem to show that $\hat{\phi}_0 = \arg(\hat{F}_{12}(\lambda_m)) \to_p \arg(F_{12}(\lambda_m))$. As $\lambda_m \to 0^+$, since the phase is right-continuous, $\arg(F_{12}(\lambda_m)) \to \arg(F_{12}(0)) = \phi_0$. ■

3.3.3 Simulation results for APE

We now assess the performance of the APE in finite samples through simulation. We will test the APE for d_{12} and d_ρ in four cases:

- FIVAR: A $FIVAR(0, (d_1, d_2))$ model with the innovation variance of the vector autoregression equal to $\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$.
- Semilagged FIVAR: In this case, $(x_{1t}, x_{2,t-5})$ follow a $FIVAR(0, (d_1, d_2))$ model with the innovation variance of the vector autoregression equal to $\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$. This increases the group delay by 5.
- Power law coherency: The anti-cointegration model given in Equation (3.23) with $d_3 = d_2 - b$, where $b = 0.1$ or $b = 0.5$. In this case, $d_{12} = d_2 - b$ and $d_\rho = \frac{1}{2}(d_2 - d_1) - b$.
- An anti-cointegration model in which there are two common components, one of which is $I(d_u - b)$ and the other of which is $I(d_u - b - 0.5)$.

- λ^α in the phase: The model given in Equation (3.27) with $d_1 = d_2$, $d_3 = d_1 - b$ where $b = 0.1$ or $b = 0.5$. In this case, $\alpha = b$.

All simulations are based on the algorithm described by Sela and Hurvich [2009]. We allow d_1, d_2, and the number of observations to vary. We choose $m = n^g$, where $g \in \{1/6, 1/3, 1/2, 2/3, 4/5\}$. In all cases, we use $q = \frac{1}{2}$, since Lobato [1997] showed that this choice worked well for a variety of values of d_1, d_2. All results are based on 1000 replications. Here, we will focus on the case where $d_1 = 0.2, d_2 = 0$. In that case, when there is power law coherency, with $b = 0.1$ and $d_\rho = -0.2$, the lower bound on the growth rate required to ensure consistency is $\frac{2d_\rho}{2d_\rho - 1} = \frac{1}{3}$. Using $b = 0.5$ so that $d_\rho = -0.6$, the required growth rate is $\frac{2d_\rho}{2d_\rho - 1} = \frac{5}{9}$. We will focus on results based on the FIVAR model and the power law coherency models; results from other models are available from the authors.

We first describe the performance of the APE for the cross-spectral density for anti-cointegration models. The root mean squared error of the estimated values of d_{12} are shown in Tables 37, 38, and 39 when the data generating processes are the FIVAR model and the two anti-cointegration models. Table 40 shows the same when the data-generating process has a power of λ in the phase. Figure 38 presents boxplots of the estimated values of d_{12} in the anti-cointegration models when $n = 32,768$. Notice that the bias and variance of the estimators are much smaller for $m = n^{2/3}, n^{4/5}$ when $b = 0.1$ and for $m = n^{4/5}$ when $b = 0.5$, relative to smaller values of m. This occurs because the growth rates required by Assumption 3.12 differ in the two cases. The lower bound on the growth rate of m can also be seen by contrasting the two boxplots in Figure 39. The left boxplot shows a case in which m grows as $n^{1/2}$, which is less than the growth rate required for consistency; the bias and variability of $\hat{d}_{12}$ do not decrease with n. In contrast, the right boxplot shows that, when m grows more quickly (in this case, as $n^{4/5}$), the

n	$m = n^{1/6}$	$m = n^{1/3}$	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	1.237	0.972	0.781	0.626	0.514
512	1.309	0.840	0.593	0.333	0.225
2048	1.435	0.713	0.388	0.211	0.124
8192	0.957	0.582	0.247	0.126	0.072
32768	1.045	0.536	0.179	0.077	0.042

Table 37: Root mean squared error of the averaged periodogram estimator for the memory parameter of the cross-spectrum for a FIVAR model with $d_1 = 0.2, d_2 = 0$.

n	$m = n^{1/6}$	$m = n^{1/3}$	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	1.335	1.178	1.022	0.835	0.682
512	1.327	1.195	0.969	0.586	0.362
2048	1.358	1.214	0.938	0.545	0.216
8192	1.314	1.264	0.861	0.332	0.131
32768	1.222	1.271	0.857	0.236	0.081

Table 38: Root mean squared error of the averaged periodogram estimator for the memory parameter of the cross-spectrum for a power law coherency model with $d_1 = 0.2, d_2 = 0, d_\rho = -0.2$.

bias and the variance of the estimator decrease with n. Thus, the lower bound on the growth rate of m is necessary for consistency.

Figure 40 present boxplots of the estimated values of d_ρ when $n = 32,768$, setting the estimated value to 0 when the $\hat{d}_{12} - \frac{1}{2}(\hat{d}_1 + \hat{d}_2) > 0$, since d_ρ cannot be positive. Figure 41 shows how $\hat{d}_\rho$ changes as n increases for different growth rates of m. As before, estimates of d_ρ based on $m = n^{1/2}$ have approximately constant bias and variability as n grows. When $m = n^{4/5}$, the variability and bias decrease as n increases, but the estimated values of d_ρ are biased upward and remain quite

n	$m = n^{1/6}$	$m = n^{1/3}$	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	1.862	1.854	1.793	1.454	0.991
512	1.829	1.840	1.818	1.501	0.718
2048	1.666	1.878	1.897	1.533	0.602
8192	1.862	1.914	1.920	1.637	0.543
32768	1.798	1.961	1.936	1.624	0.414

Table 39: Root mean squared error of the averaged periodogram estimator for the memory parameter of the cross-spectrum for a power law coherency model with $d_1 = 0.2, d_2 = 0, d_\rho = -0.6$.

n	$m = n^{1/6}$	$m = n^{1/3}$	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	1.143	0.906	0.721	0.558	0.457
512	1.174	0.880	0.608	0.348	0.236
2048	1.345	0.714	0.395	0.224	0.138
8192	0.989	0.554	0.271	0.126	0.093
32768	1.050	0.441	0.176	0.090	0.067

Table 40: Root mean squared error of the averaged periodogram estimator for the memory parameter of the cross-spectrum with a power of λ in the phase with $d_1 = 0.2, d_2 = 0, \alpha = 0.1$.

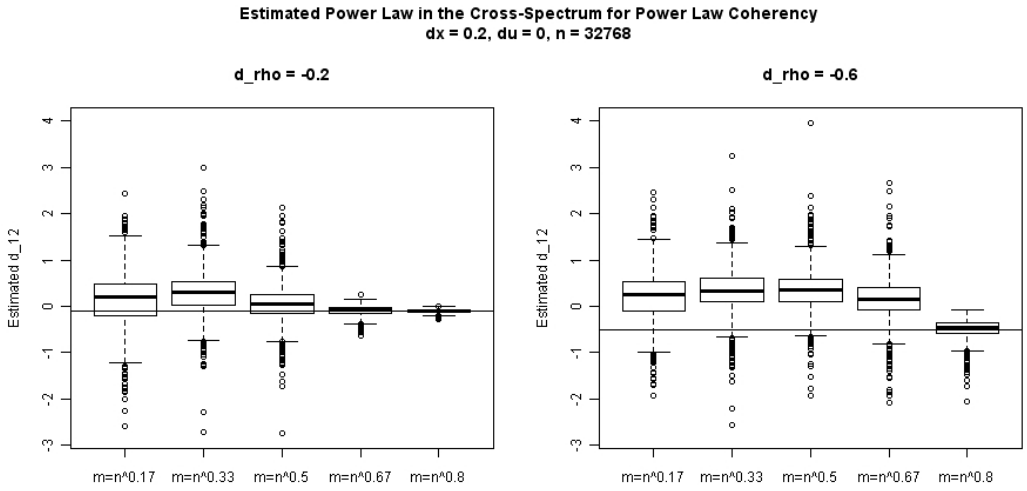

Figure 38: Estimated power law in the cross-spectral density using the averaged periodogram estimator when the true data-generating process has power law coherency, with $d_1 = 0.2, d_2 = 0, n = 32,768$ and varying choices of m. $d_\rho = -0.2$ in the left panel; $d_\rho = -0.6$ in the right panel.

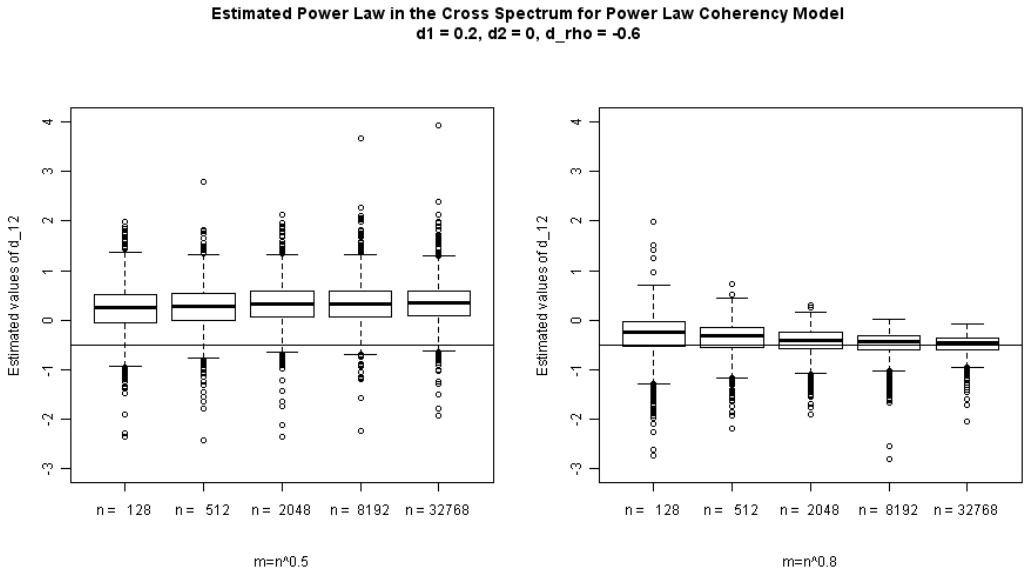

Figure 39: Estimated power law in the cross-spectrum when the true data-generating process has power law coherency with $d_1 = 0.2, d_2 = 0, d_\rho = -0.6$, and varying n. $m = n^{1/2}$ in the left panel and $m = n^{4/5}$ in the right panel.

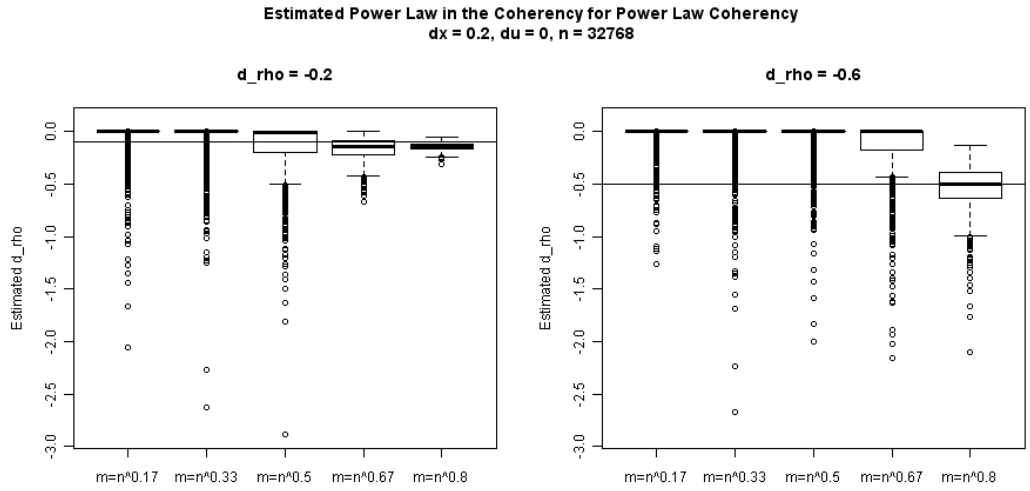

Figure 40: Estimated power law in the coherency using the averaged periodogram estimator when the true data-generating process has power law coherency with $d_1 = 0.2, d_2 = 0, n = 32,768$ and varying choices of m. $d_\rho = -0.2$ in the left panel; $d_\rho = -0.6$ in the right panel.

variable. This occurs because $\hat{d}_\rho$ depends on three different averaged periodogram estimators. Furthermore, since x_{1t} and x_{2t} contain two components with different memory parameters, the estimators of the memory parameters of the auto-spectra may be particularly badly behaved.

While the estimator is inconsistent when m grows too slowly, it is biased in finite samples when m grows too quickly. To see this more clearly, we apply the APE to data generated by an anticointegration model in which the common component is of the form $(1-L)^{d_2-b} + (1-L)^{d_2-b-0.5}$. In this case, the cross-spectrum is not well approximated by λ^{d_2-b} away from frequency 0. In Figure 42, we plot the estimated values of $\hat{d}_{12}$ for three different growth rates of m. In the left panel, m grows too slowly, so the estimator, while consistent in this case, is biased and quite

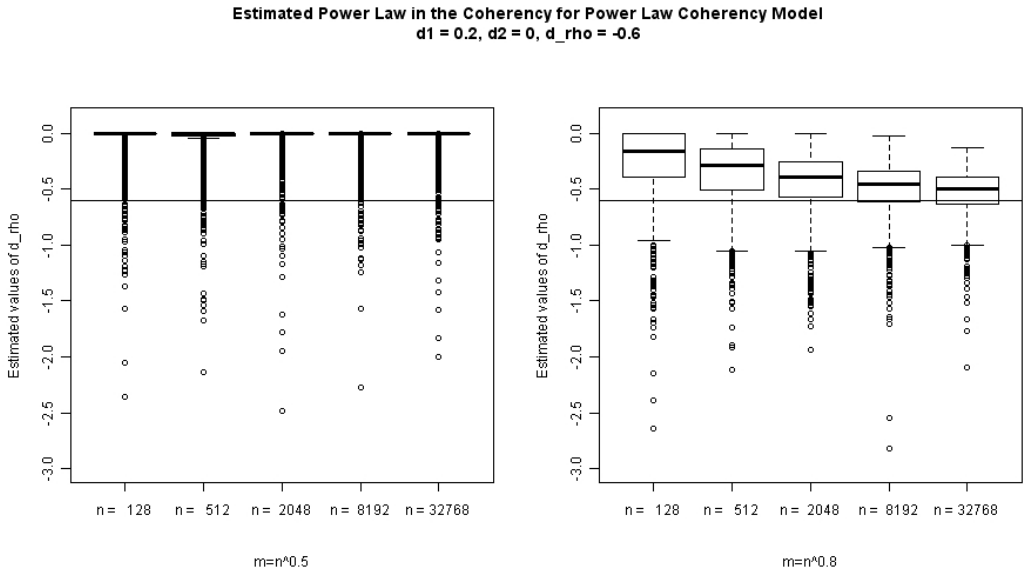

Figure 41: Estimated power law in the coherency when the true data-generating process has power law coherency with $d_1 = 0.2, d_2 = 0, d_\rho = -0.6$, and varying n. $m = n^{1/2}$ in the left panel and $m = n^{4/5}$ in the right panel.

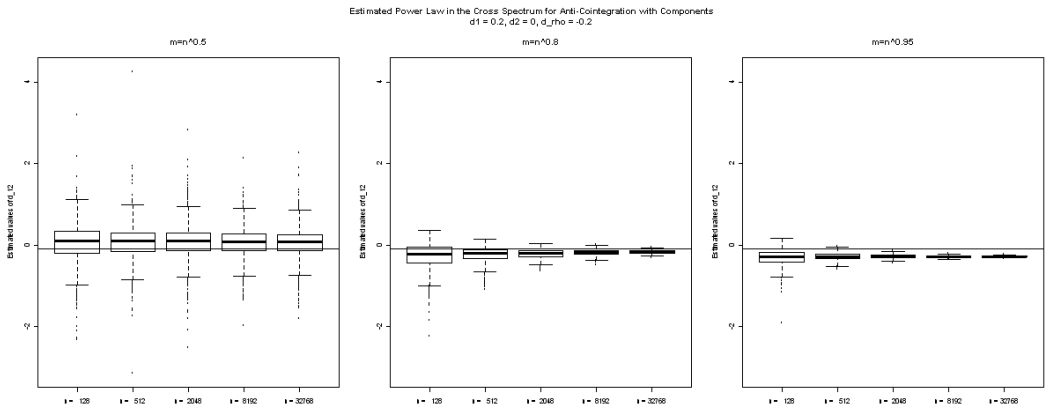

Figure 42: Estimated power law in the cross-spectrum when the true data-generating process has power law coherency with $d_1 = 0.2, d_2 = 0, d_\rho = -0.6$, based on two common components with different memory parameters, and varying n. $m = n^{1/2}$ is in the left panel, $m = n^{4/5}$ is in the middle panel, and $m = n^{0.95}$ is in the right panel.

variable. In the middle panel, m grows at a pace that leads to good performance. In the right panel, m grows so quickly that finite-sample bias is evident. In Figure 43, we plot the estimated values of the coherency in the same cases. Again, the estimator performs badly if the growth rate is chosen to be too large or too small, making the choice of the m problematic in data analysis. In Figure 44, we plot $\hat{d}_{12}$ and $\hat{d}_\rho$ for varying choices of m, holding n fixed. In this plot, we can see that the variability of the two estimators decreases with m, but that the bias first decreases and then increases with m. Furthermore, the bias in $\hat{d}_\rho$ is minimized for a different choice of m, since the biases of $\hat{d}_1$ and $\hat{d}_2$ vary with m as well, making the choice of m more difficult.

Next, we compare the distribution of the estimated power laws in the cross-spectrum and coherency in the cases where there is and is not power law coherency,

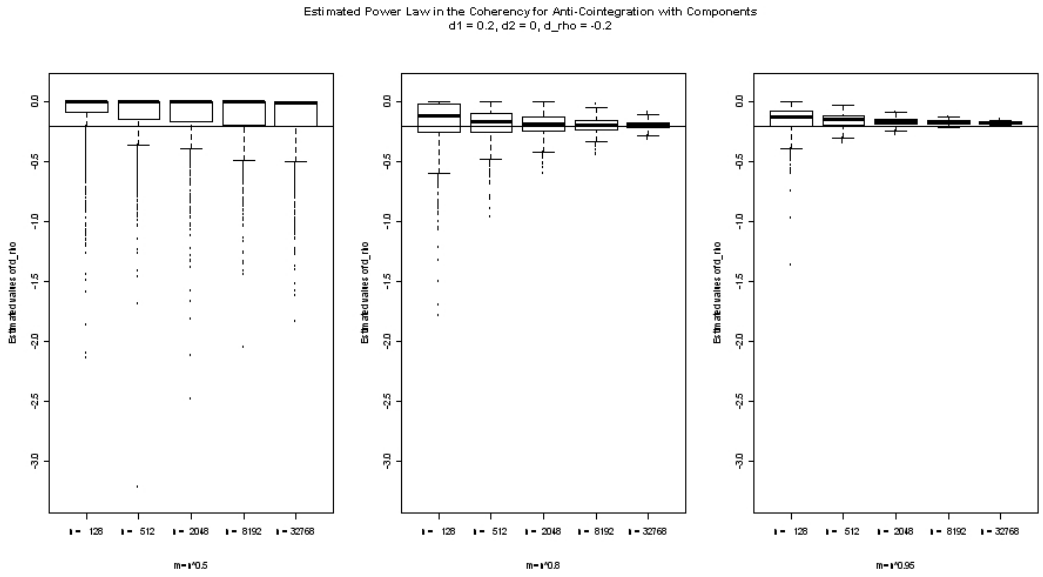

Figure 43: Estimated power law in the coherency when the true data-generating process has power law coherency with $d_1 = 0.2, d_2 = 0, d_\rho = -0.6$, based on two common components with different memory parameters, and varying n. $m = n^{1/2}$ is in the left panel, $m = n^{4/5}$ is in the middle panel, and $m = n^{0.95}$ is in the right panel.

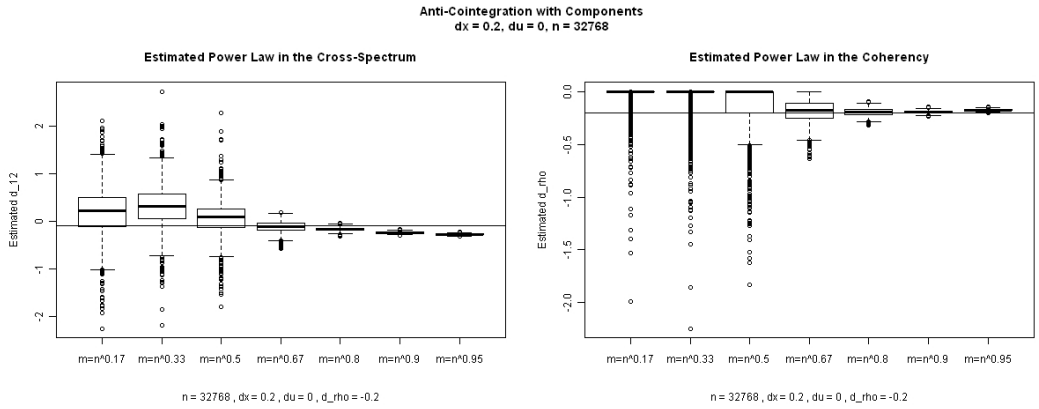

Figure 44: Estimated power laws in the cross-spectrum and coherency when the true data-generating process has power law coherency with $d_1 = 0.2, d_2 = 0, d_\rho = -0.6, n = 32768$, based on two common components with different memory parameters. $\hat{d}_{12}$ is in the left panel and $\hat{d}_\rho$ is in the right panel.

holding d_1, d_2 and n constant. In Figure 45, we show the distribution of the estimated memory parameter of the cross-spectral density, with $m = n^{4/5}$, a large enough growth rate that $\hat{d}_{12}$ will be consistent in all three cases. As d_ρ increases, the estimated values of d_{12} become more variable. In Figure 46, the estimated values of d_ρ are shown across the three models. Notice that $\hat{d}_\rho < 0$ for over 75% of the simulated realizations when $b = 0.1$, suggesting that the point estimate would lead to the correct conclusion about power law coherency in this case. For $d_\rho = -0.6$, $\hat{d}_\rho < 0$ for all realizations. However, the estimated values are quite variable. Furthermore, if m is chosen too small (as seen in the left-hand panel of Figure 41) or if n is small, detection of a power law in the coherency is more problematic. This shows that detecting power law coherency could be quite difficult in practice.

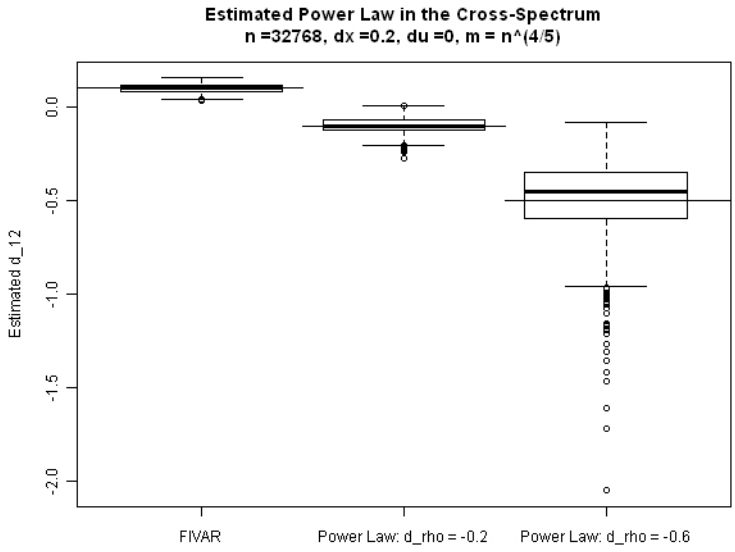

Figure 45: Estimated memory parameter of the cross-spectral density using the averaged periodogram estimator with and without power law coherency.

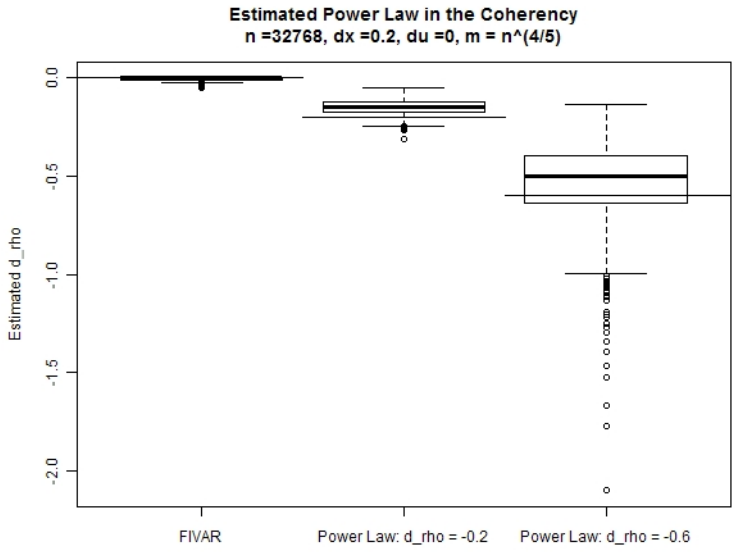

Figure 46: Estimated memory parameter of the coherency using the averaged periodogram estimator with and without power law coherency.

3.4 The effect of the phase and coherency on cointegration estimators

The cointegrating parameter between two fractionally cointegrated series can also be estimated based on the averaged periodogram. Suppose, as in Section 3.2.4, that we observe two series, $\{x_t\}, \{y_t\}$ such that both are $I(d_x)$ with a linear combination, $u_t = y_t - \beta x_t$, that is integrated of order $d_u < d_x$. One could estimate β in a variety of ways. First, one could regress y_t on x_t. However, Robinson [1994] showed that the resulting estimator is inconsistent in certain cases when $Cov(x_t, u_t) \neq 0$. As an alternative, Robinson [1994] proposed the narrow-band least squares (NBLS) estimator, given by:

$$\hat{\beta}_m = \frac{\Re(\hat{F}_{xy}(\lambda_m))}{\hat{F}_{xx}(\lambda_m)} \tag{3.38}$$

This estimator is reasonable because $F_{xy}(\lambda_m) \sim \beta \int_0^{\lambda_m} f_{11}(\lambda) d\lambda$ as $\lambda_m \to 0^+$, so that $\frac{F_{xy}(\lambda_m)}{F_{xx}(\lambda_m)} \to \beta$ as $\lambda_m \to 0^+$, as mentioned in Section 3.2.4. Since cointegration is ultimately a description of the long-run relationship between two series, estimating the cointegrating parameter in a neighborhood of frequency zero makes intuitive sense.

A number of authors have discussed the performance of this estimator under the assumption that $m \to \infty$ and $\frac{m}{n} \to 0$ as $n \to \infty$. Robinson [1994] assumed that $0 < d_u < d_x$ and showed that $\hat{\beta}_{NBLS}$ is consistent as long as $\frac{m}{n} + \frac{1}{m} \to 0$. Christensen and Nielsen [2006] described the convergence rate of the estimator in the presence of power law coherency (or, more generally, when $f_{xu}(\lambda) = O(\lambda^{-d_x - d_u + \xi})$ as $\lambda \to 0^+$ for some $\xi > 0$). Specifically, in the bivariate case, if $0 \leq d_u < d_x$ and $d_u + d_x < 1/2$, they found that $1/m + m^{1+2\xi}/n^{2\xi} \to 0$ implied that $\sqrt{m}\lambda_m^{d_u - d_x}(\hat{\beta}_m - \beta)$ tends to a normal distribution with mean 0. Suppose we choose $m = n^g$ for some $g < \frac{2\xi}{2(1+2\xi)}$. Then, we find that $\hat{\beta} - \beta = O_p\left(n^{g(-\frac{1}{2}) + (1-g)(d_u - d_x)}\right)$, so that the exponent on n is a

weighted average of $-\frac{1}{2}$ and $d_u - d_x$, where the weight on $-\frac{1}{2}$ is smaller for smaller choices of ξ. For arbitrarily small ξ, we have $\hat{\beta} - \beta = O_p(n^{d_u - d_x})$. If there is power law coherency, then we must have $\xi \leq -2d_\rho$; ξ also depends on the smoothness of the autospectra and the phase. Thus, ξ can be made arbitrarily small by choosing d_ρ close to 0.

If ξ is chosen to be too large, then the mean of the asymptotic distribution will not go to zero and may be infinite [Christensen and Nielsen, 2006, Equation 26, page 364]. Specifically, the requirement is given by (in our notation):

$$\sqrt{m}\lambda_m^{d_x+d_u-1}\frac{1}{n}\sum_{j=1}^{m}\Re(\Psi_1(\lambda_j)\Psi_2^*(\lambda_j)) = O\left(\frac{m^{1+2\xi}}{n^{2\xi}}\right)$$

In finite samples, choosing m too large is likely to lead to bias, as we will see in simulations in Section 3.4.2. This shows that the growth rate of m can be limited both by powers of λ in the phase and by power law coherency, which are likely to be unknown.

Robinson and Marinucci [2003] consider the more general case where $0 \leq d_u < d_x < 1/2$ and $\rho(0)$ need not be zero; they show in their Theorem 3.1 that $\hat{\beta}_m - \beta = O_p(\lambda_m^{d_x - d_u})$, which matches the Christensen and Nielsen result for arbitrarily small ξ. Further, they conjecture (page 341) that $\lambda_m^{d_u - d_x}(\hat{\beta} - \beta)$ will converge in probability to a non-zero constant. Robinson and Marinucci, 2003] also apply the NBLS estimator to the case where $d_x \geq 1/2$ and $d_u \geq 0$, so that the observed processes are non-stationary. A number of their results [such as, Robinson and Marinucci, Theorems 4.1-4.5 and Propositions 6.1-6.2] have limits of normalized expected values and some limiting distributions that depend on $f_{12}^{\dagger}(0)$. This suggests that their convergence rates or other results may change in the presence of power law coherency, but we will not pursue that here.

Robinson [2008] suggested a different semiparametric approach to estimate the cointegrating parameter. He applied a local Whittle estimator to the spectral

density matrix of $(x_t, y_t)'$, parameterizing it locally using β, d_x, d_u, and ϕ_0 and then estimating all four parameters. In order to ensure that ϕ_0 was identified, he required that $\rho(0) > 0$, ruling out power law coherency. He also required that $\phi_0 \neq \pm\frac{\pi}{2}$. As long as $\frac{1}{m} + \frac{m}{n} \to 0$, his estimator satisfies $\hat{\beta} - \beta = o_p\left(\left(\frac{m}{n}\right)^{d_x - d_u}\right)$ [Robinson, 2008, Theorem 3]. Robinson's Theorem 4 states that $\sqrt{m}\lambda_m^{d_u - d_x}(\hat{\beta} - \beta_0)$ converges to a normal distribution. Based on the calculations in Section 3.2.1, $n^{2\xi/(1+2\xi)}$ is an upper bound on the growth rate of m. Thus, the convergence rate is bounded above by $n^{\frac{1}{1+2\xi}(d_x - d_u) + \frac{2\xi}{2(1+2\xi)}}$, just as that of Christensen and Nielsen [2006] is; in both cases, the estimator appears to exploiting information about whether the coherency is zero or non-zero (which may not be known in practice) to gain efficiency. As before, ξ is also unlikely to be known. Furthermore, as we will see in Section 3.4.2, Robinson's estimator does not work well for small sample sizes.

3.4.1 A robust cointegration estimator

Chen and Hurvich [2003] estimating β using Equation (3.38) with m fixed. We will call the resulting estimator the very narrow-band least squares estimator (VNBLS). For the reasons discussed in Section 3.3, they recommend differencing when the series may be non-stationary or there may be polynomial trends. As in Section 3.3, the series must be tapered if they have been differenced or if they are suspected to have $d_x, d_u < -1/2$. Here, we generalize their results to allow for p driving innovation series and arbitrary ϕ_0 in the cross-spectral density of x_t, u_t. To begin, we require one mild assumption on $f^\dagger(\lambda)$, in addition to those stated in Section 3.2 (we do not require the assumptions from Section 3.3).

Assumption 3.17 *$f^\dagger(\lambda)$ is positive definite except on a set of measure 0.*

Next, we compute some quantities that will help describe the asymptotic performance of $\hat{\beta}_m$. These definitions generalize those of Chen and Hurvich [2003] to the semiparametric model given in Section 3.2. First, we compute the derivative of the transfer function:

$$\begin{aligned}
\Psi'_{jk}(\lambda) &= -i\delta_{jk}(1-e^{-i\lambda})^{-\delta_{jk}-1}e^{-i\lambda}\tau_{jk}(\lambda)e^{-\varphi_{jk}(\lambda)} \\
&\quad +(1-e^{-i\lambda})^{-\delta_{jk}}\tau'_{jk}(\lambda)e^{i\varphi_{jk}(\lambda)} + (1-e^{-i\lambda})^{-\delta_{jk}}\tau_{jk}(\lambda)i\varphi'_{jk}(\lambda)e^{i\varphi_{jk}(\lambda)} \\
&= (1-e^{-i\lambda})^{-\delta_{jk}}e^{i\varphi_{jk}(\lambda)} \\
&\quad \times \left(-i\delta_{jk}\tau_{jk}(\lambda)e^{-i\lambda}(1-e^{-i\lambda})^{-1} + \tau'_{jk}(\lambda) + i\tau_{jk}(\lambda)\varphi'_{jk}(\lambda)\right)
\end{aligned}$$

If $\tau_{jk}(\lambda) = 0$ for $\lambda \in [0, \epsilon]$, then $\Psi'_{jk}(\lambda) = 0$. As $\lambda \to 0^+$, the term including $(1-e^{-i\lambda})^{-1}$ dominates, since $\tau'_{jk}(\lambda), \varphi'_{jk}(\lambda = o(\lambda^{-1})$ by Assumptions 3.2 and 3.4. Thus, as $\lambda \to 0^+$,

$$\begin{aligned}
\Psi'_{jk}(\lambda) &\sim -i(1-e^{-i\lambda})^{-\delta_{jk}-1}e^{i\varphi_{jk}(\lambda)}\delta_{jk}\tau_{jk}(0) \\
&\sim -\delta_{jk}\tau_{jk}(0)|\lambda|^{-\delta_{jk}-1}e^{i\left(\varphi_{jk}(\lambda)+\pi\delta_{jk}/2\right)}
\end{aligned}$$

If $\delta_{jk} = 0$, then the derivative is given by:

$$\Psi'_{jk}(\lambda) = e^{i\varphi_{jk}(\lambda)}\left(\tau'_{jk}(\lambda) + i\varphi'_{jk}(\lambda)\tau_{jk}(\lambda)\right)$$

Generalizing Chen and Hurvich [2003, page 101], we define the spectral measure of $(x_t, u_t)'$ to be:

$$G(d\lambda) = f(\lambda)d\lambda$$

We then normalize the spectral measure:

$$\begin{aligned}
G_n(dx) &= n\Lambda_n G\left(\frac{dx}{n}\right)\Lambda_n \\
&= \Lambda_n \Psi\left(\frac{x}{n}\right)\Sigma\Psi^*\left(\frac{x}{n}\right)dx
\end{aligned}$$

where $\Lambda_n = \text{diag}\left(n^{d_x}, n^{d_u}\right) = \text{diag}\left(n^{d_1}, n^{d_2}\right)$. As $n \to \infty$, we find that the (j,k) element of $\Lambda_n \Psi\left(\frac{x}{n}\right)$ is:

$$\begin{aligned}
\left[\Lambda_n \Psi\left(\frac{x}{n}\right)\right]_{jk} &= n^{-d_j}(1 - e^{-i\frac{x}{n}})^{-\delta_{jk}} \tau_{jk}\left(\frac{x}{n}\right) e^{i\varphi_{jk}\left(\frac{x}{n}\right)} \\
&= n^{-d_j} \left|\frac{x}{n}\right|^{-\delta_{jk}} \left(\frac{\left|2\sin\frac{x}{2n}\right|}{\left|\frac{x}{n}\right|}\right)^{-\delta_{jk}} \tau_{jk}\left(\frac{x}{n}\right) e^{i(\varphi_{jk}(\frac{x}{n}) - \delta_{jk}(\pi - \frac{x}{n})/2)} \\
&= n^{\delta_{jk} - d_j} |x|^{-\delta_{jk}} \left(\frac{\left|2\sin\frac{x}{2n}\right|}{\left|\frac{x}{n}\right|}\right)^{\delta_{jk}} \tau_{jk}\left(\frac{x}{n}\right) e^{i(\varphi_{jk}(\frac{x}{n}) - \delta_{jk}(\pi - \frac{x}{n})/2)}
\end{aligned}$$

If $d_j = \delta_{jk}$, as $n \to \infty$ with x fixed, we have:

$$\left[\Lambda_n \Psi\left(\frac{x}{n}\right)\right]_{jk} \sim |x|^{-d_j} \tau_{jk}(0) e^{i\left(\phi_{0,jk} - \frac{\delta_{jk}\pi}{2}\right)}$$

where $\phi_{0,jk} = \lim_{\lambda \to 0^+} \varphi_{jk}(\lambda)$. If $d_j > \delta_{jk}$, as $n \to \infty$ with x fixed, $n^{\delta_{jk} - d_j} \to 0$ and $[\Lambda_n \Psi(\frac{x}{n})]_{jk} \to 0$. Therefore, $G_n(S) \to G_0(S)$, where:

$$G_0(dx) = \Pi(x) f^\dagger(0) \Pi^*(x) dx$$

where

$$\begin{aligned}
\Pi(x) &= \text{diag}\left(e^{-id_x\pi/2}|x|^{-d_x}, e^{-i(d_u\frac{\pi}{2} + \phi_0 + \frac{\pi}{2}(d_x - d_u))}|x|^{-d_u}\right) \\
&= e^{-id_x\pi/2}\text{diag}\left(|x|^{-d_x}, e^{-i\phi_0}|x|^{-d_u}\right)
\end{aligned}$$

and ϕ_0 is defined as in Section 3.2.

We also require a spectral representation for both the p-variate innovations process, which implies a spectral representation for (x_t, u_t):

$$\begin{aligned}
\epsilon_t &= \int_{-\pi}^{\pi} e^{i\lambda t} dZ_\epsilon(\lambda) \\
a_S(r) &= \frac{1}{2\pi} \int_{S/n} e^{-irx} \Psi(x) dx \\
Z_n(S) &= \sqrt{n}\Lambda_n \sum_{r=-\infty}^{\infty} a_S(r)\epsilon_r = \sqrt{n}\Lambda_n \int_{S/n} \Psi(x) dZ_\epsilon(x)
\end{aligned}$$

where $S \subset R$ is a bounded subset. The properties for $dZ_\epsilon(\lambda)$ are identical to those given in Chen and Hurvich [2003, Equations 11 and 12], except that Σ is now a $p \times p$ matrix. Given these definitions, we restate and generalize the proof of Lemma 1 of Chen and Hurvich.

Lemma 3.18 *If $S_1, ..., S_M$ are intervals in R with nonzero endpoints and $\pm S_1, ..., \pm S_M$ are disjoint, then:*

$$(Z_n(S_1), ..., Z_n(S_M)) \rightarrow^d (Z_{G_0}(S_1), ..., Z_{G_0}(S_M))$$

where for any Borel set, $\tilde{S} \subset R$, the 2×2 matrix measure, $G_0(\tilde{S})$ is defined as above, and Z_{G_0} is the bivariate, complex, Gaussian random measure satisfying:

$$\begin{aligned} E(Z_{G_0}(\tilde{S})) &= 0 \\ E(Z_{G_0}(\tilde{S})Z^*_{G_0}(\tilde{S})) &= G_0(\tilde{S}) \\ \overline{Z_{G_0}(-\tilde{S})} &= Z_{G_0}(\tilde{S}) \\ E(Z_{G_0}(\tilde{S}_1)Z^*_{G_0}(\tilde{S}_2)) &= 0 \end{aligned}$$

when $\tilde{S}_1 \cap \tilde{S}_2 = \emptyset$.

Proof. The following facts from Lemma 1 of Chen and Hurvich [2003] are unchanged by our more general assumptions:

$$\begin{aligned} Z_n(S_j) &= \overline{Z}_n(-S_j) \\ E(\Re(Z_n(S_j))\Re(Z'_n(S_k))) &= E(\Re(Z_n(S_j))\Im(Z'_n(S_k))) \\ &= E(\Im(Z_n(S_j))\Re(Z'_n(S_k))) \\ &= E(\Im(Z_n(S_j))\Im(Z'_n(S_k))) \\ &= 0 \end{aligned}$$

for $j, k = 1, ..., M$ with $j \neq k$. We now apply the Cramer-Wold device, studying an arbitrary linear combination: $a'\Re Z_n(S) + b'\Im Z_n(S)$. Since we still have

$E(Z_n(S)Z_n^*(S)) = G_n(S) \to G_0(S)$, with $G_0(S)$ positive definite by Assumption 3.17,

$$\begin{aligned}\mathrm{Var}(a'\Re Z_n(S) + b'\Im Z_n(S)) &= \mathrm{Var}(a'\Re Z_{G_0}(S) + b'\Im Z_{G_0}(S)) \\ &= \sigma_0^2 > 0\end{aligned}$$

Using the expressions for $\Upsilon(\lambda)$ and $f^\dagger(\lambda)$ given in Section 3.2, Chen and Hurvich's proof of Lemma 1 can be generalized to our formulation of $\Psi(\lambda)$ by rewriting their equations A.1 and A.2 as:

$$\begin{aligned}|a_\Delta(r,n)_{1k}| &\le Cn^{\delta_{1k}}\frac{1}{|r|} \le Cn^{d_x}\frac{1}{|r|} \\ |a_\Delta(r,n)_{2k}| &\le Cn^{\delta_{2k}}\frac{1}{|r|} \le Cn^{d_u}\frac{1}{|r|}\end{aligned}$$

Then, $\alpha\Re(Z_n(S)) + \beta'\Im(Z_n(S)) = \sum_{s=-\infty}^{\infty} W_{nr}$. The properties of W_{nr} given in their equations A.3 and A.4 can be easily generalized:

$$\begin{aligned}W_{nr} &= \sum_{k=1}^{p}\big(n^{1/2-d_x}(\alpha_1\Re(a_S(r)_{1k}) + \beta_1\Im(a_S(r)_{1k})) \\ &\quad + n^{1/2-d_u}(\alpha_2\Re(a_S(r)_{2k}) + \beta_2\Im(a_S(r)_{2k}))\big)\epsilon_{rk} \\ E(W_{nr}) &\le C\Big(n^{1-2d_x}\sum_{k=1}^{p}|a_S(r)_{1k}|^2 + n^{1-2d_u}\sum_{k=1}^{p}|a_S(r)_{2k}|^2\Big) \\ &\le Cn/r^2\end{aligned}$$

so that the bounds required in Lemma 1 continue to hold. ■

Theorem 3.19 *Suppose that $-s - 1/2 < d_u < d_x < 1/2$ and m is a fixed positive integer. Then,*

$$\{\Lambda_n w_{j,s}\}_{j=1}^m \to^d \left\{\int_{\mathbb{R}} \Delta_s(x + 2\pi j)dZ_{G_0}(x)\right\}_{j=1}^m$$

as $n \to \infty$*, where*

$$\begin{aligned}
\Lambda_n &= diag(n^{-d_x}, n^{-d_u}) \\
\Delta_s(x) &= \binom{2s}{s}^{-1/2} \sum_{k=0}^{s} \binom{s}{k} (-1)^k \Delta(x + 2\pi k) \\
\Delta(x) &= \frac{e^{ix} - 1}{\sqrt{2\pi} i x}
\end{aligned}$$

and Z_{G_0} *is the bivariate complex Gaussian measure defined in Lemma 3.18.*

Proof. The proof in the more general case is identical to that of Chen and Hurvich [2003, pages 115-120], because Y_n and the related quantities in the proof depend only on the autospectra and therefore are not affected by the phase and coherency.

■

Corollary 1 of Chen and Hurvich [2003] continues to hold when we use our more general definition of $G_0(dx)$:

Corollary 3.20

$$n^{d_x - d_u} \left(\hat{\beta}_m - \beta \right) \quad \to^d \quad \frac{\sum_{j=1}^{m} \left(A_{xj} A_{uj} + B_{xj} B_{uj} \right)}{\sum_{j=1}^{m} \left(A_{xj}^2 + B_{xj}^2 \right)}$$

where $\{A_{xj}, A_{uj}, B_{xj}, B_{uj}\}_{j=1}^{m}$ *are jointly normal random variables with zero mean and covariances determined by:*

$$\begin{aligned}
E(A_j A_k') &= \frac{1}{2} \Re(L_1(j,k) + L_2(j,k)) \\
E(B_j B_k') &= \frac{1}{2} \Re(L_2(j,k) - L_1(j,k)) \\
E(A_j B_k') &= \frac{1}{2} \Im(L_1(j,k) - L_2(j,k)) \\
L_1(j,k) &= \int_R \Delta_s(x + 2\pi j) \Delta_s(-x + 2\pi k) G_0(dx) \\
L_2(j,k) &= \int_R \Delta_s(x + 2\pi j) \overline{\Delta_s(x + 2\pi k)} G_0(dx)
\end{aligned}$$

with $A_j = (A_{xj}, A_{uj})'$ *and* $B_j = (B_{xj}, B_{uj})'$ *for* $j, k = 1, .., m$.

Remark 3.21 *Using the definition of $G_0(dx)$, we find that:*

$$\begin{aligned} L_1(j,k) &= \int_R \Delta_s(x+2\pi j)\Delta_s(-x+2\pi k)\Pi(x)f^\dagger(0)\Pi^*(x)dx \\ L_2(j,k) &= \int_R \Delta_s(x+2\pi j)\overline{\Delta_s(x+2\pi k)}\Pi(x)f^\dagger(0)\Pi^*(x)dx \end{aligned}$$

When $[f^\dagger(0)]_{12} = 0$ (or, equivalently, $\rho(0) = 0$, as would happen with power law coherency), we have $E(A_{xj}A_{uk}) = E(A_{xj}B_{uk}) = E(B_{xj}B_{uk}) = 0$ for all $j,k = 1,...,m$. Since $A_{xk}, A_{uk}, B_{xk}, B_{uk}$ are multivariate Gaussian, zero covariances imply independence, so that:

$$\begin{aligned} E\left(\frac{\sum_{j=1}^m (A_{xj}A_{uj} + B_{xj}B_{uj})}{\sum_{j=1}^m \left(A_{xj}^2 + B_{xj}^2\right)}\right) &= \sum_{k=1}^m \left[E(A_{uk})E\left(\frac{A_{xk}}{\sum_{j=1}^m \left(A_{xj}^2 + B_{xj}^2\right)}\right)\right. \\ &\quad \left. + E(B_{uk})E\left(\frac{B_{xk}}{\sum_{j=1}^m \left(A_{xj}^2 + B_{xj}^2\right)}\right)\right] \\ &= 0 \end{aligned}$$

so that the mean of the asymptotic distribution of VNBLS is 0 when $[f^\dagger(0)]_{12} = 0$, just as the mean of the distribution of NBLS is 0 under the conditions of Christensen and Nielsen [2006].

When $\rho(0) > 0$, the asymptotic distribution may have a non-zero mean, just as Robinson and Marinucci [2003] conjecture will occur with NBLS. However, the presence of power laws in the coherency or in the phase does not appear here, demonstrating that VNBLS estimator are not affected by the lack of smoothness, unlike the NBLS estimator.

The result in Corollary 3.20 can be used in inference. In order to perform inference, one must estimate $f^\dagger(0), \phi_0, d_x$, and d_u; local Whittle estimates like those of Robinson [2008] could be adapted for this purpose. Given these estimates, one can then estimate the variances and covariances of A_j and B_k using numerical integration. Then, one could simulate from the distribution given in the corollary

to find cutoffs for inference. We leave the details of these calculations and an exploration of its performance to future research.

3.4.2 Simulation results for cointegration estimators

We compare the performance of the VNBLS and NBLS estimators to the performance of Robinson's local Whittle estimator in finite samples using simulation. As in Section 3.3.3, we will consider four data-generating processes for $(x_t, u_t)'$: a FIVAR model, a semilagged FIVAR model, with (x_t, u_{t-5}) following a FIVAR model, anti-cointegration to create power law coherency, and the model in Section 3.2.6 that leads to powers of λ in the phase. We then create $y_t = x_t + u_t$, so that $\beta = 1$. We apply the VNBLS and NBLS estimators and the local Whittle (LW) estimator of Robinson [2008] to $(x_t, y_t)'$ to estimate β. All results are based on 1000 replications.

In this section, we will report results when $d_x = 0.4, d_u = 0.2$. In that case, the results of of the previous section and of Robinson and Marinucci [2003] imply that $\hat{\beta} - \beta = O_p(n^{-0.2})$ for VNBLS and NBLS. Local Whittle when $\rho(0) > 0$ and NBLS when $\rho(0) = 0$ will be $O_p(n^{g(-0.5)+(1-g)(-0.2)})$, where $m = n^g$, with g bounded above by $\frac{2\xi}{1+2\xi}$. Notice that choosing d_x, d_u such that $d_x - d_u$ is larger will reduce the difference between the two convergence rates and improves the performance of all of the estimators.

In all of these cases, $\text{Cov}(x_t, u_t) \neq 0$, so that OLS is asymptotically biased; this is quite evident in simulations (not shown). The bias is smaller in the case of the semilagged FIVAR model because the contemporaneous covariance is smaller.

In Tables 41, 42, and 43, we report the root mean squared errors for the three estimators when $(x_t, u_t)'$ are a FIVAR process with $d_x = 0.4, d_u = 0.2$. In this case, the NBLS and VNBLS estimators are $O_p(n^{-0.2})$. Because of the group delay

n	$m = 4$	$m = 20$	$m = 36$
128	0.277	0.263	0.279
512	0.264	0.214	0.226
2048	0.166	0.170	0.180
8192	0.133	0.136	0.144

Table 41: Root mean squared error of the VNBLS, when $(x_t, u_t)'$ are generated by a FIVAR process with $d_x = 0.4, d_u = 0.2$.

in the FIVAR model, we must have $g < \frac{2}{3}$, which means that the local Whittle convergence rate is at best $O_p(n^{-0.4})$; smaller growth rates of m will lead to smaller convergence rates. Even with this simple data generating process, the VNBLS and NBLS estimators performs best in finite samples. Furthermore, in larger samples, the root mean square errors are lower for small growth rates of m and for small fixed m. Local Whittle performs very badly in small samples, and even with $n = 8192$ has a much larger root mean squared error than the VNBLS and NBLS estimators. As shown in Figure 47, the VNBLS and NBLS estimators are biased downward, with VNBLS with $m = 4$ having the least bias (though slightly more variability). In contrast, the local Whittle estimator, shown in the bottom row of Figure 47, is biased upward and quite variable, especially in smaller samples. In Figure 48, we plot the same estimators on the same scale in the case where $n = 8192$. The left-hand boxplot shows that the local Whittle estimator is biased upward and quite variable relative to the NBLS and VNBLS estimators. The right boxplot shows the same VNBLS and NBLS estimators on a smaller scale. Again, we see that VNBLS with $m = 4$ has the least bias and more variability.

Tables 44, 45, and 46 show the root mean squared errors of the cointegration estimators for semilagged FIVAR processes, in which (x_t, u_{t-5}) follow a FIVAR

n	$m = n^{1/6}$	$m = n^{1/3}$	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	0.748	0.274	0.251	0.269	0.288
512	0.626	0.203	0.215	0.24	0.264
2048	0.176	0.164	0.184	0.213	0.242
8192	0.133	0.136	0.159	0.192	0.225

Table 42: Root mean squared error of NBLS, when $(x_t, u_t)'$ are generated by a FIVAR process with $d_x = 0.4, d_u = 0.2$.

n	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	30.214	27.595	24.205
512	22.076	19.111	18.071
2048	13.969	10.117	8.771
8192	7.273	2.135	2.212

Table 43: Root mean squared error of the local Whittle cointegration estimator with m growing with n, when $(x_t, u_t)'$ are generated by a FIVAR process with $d_x = 0.4, d_u = 0.2$.

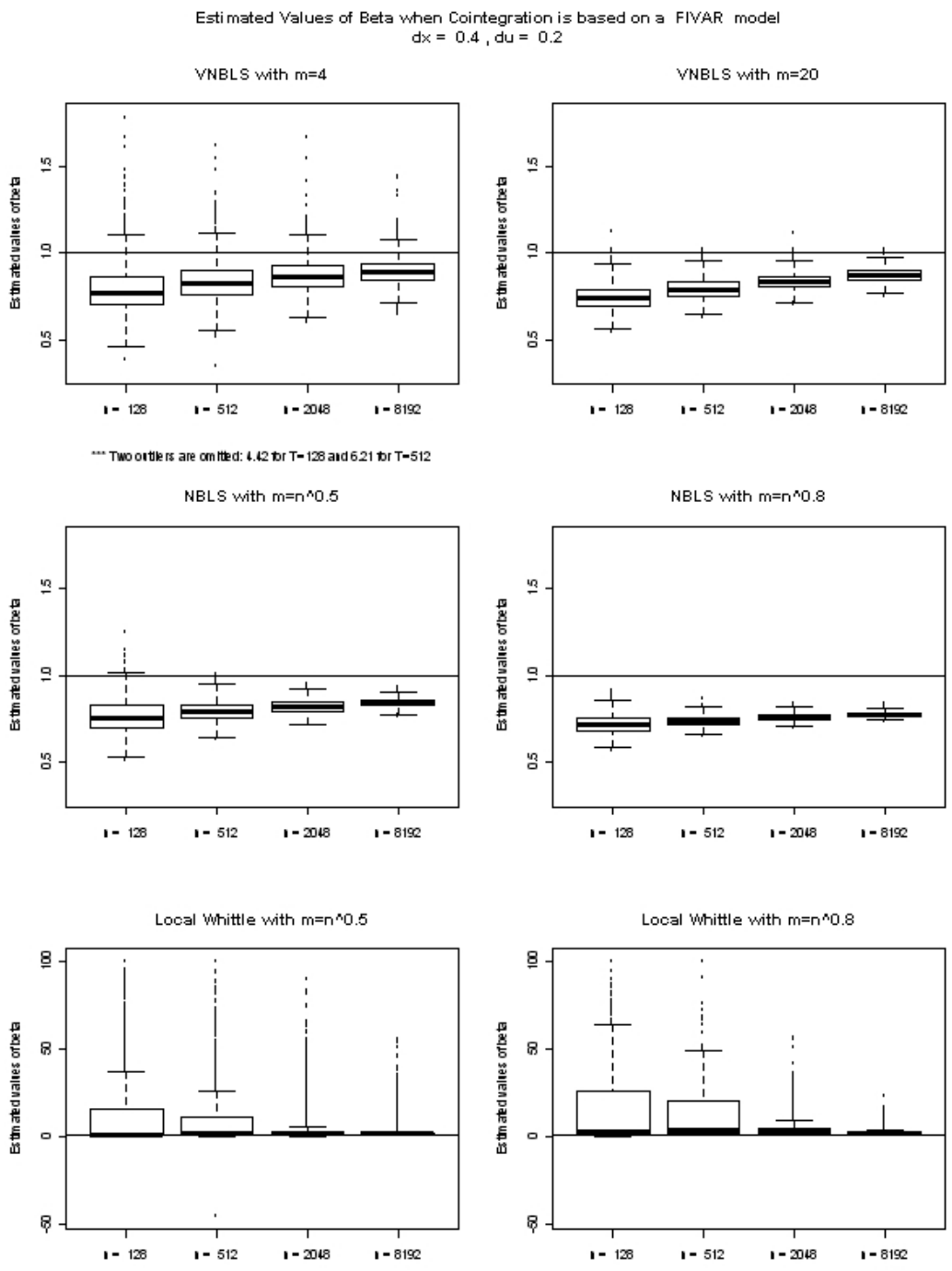

Figure 47: Estimated values of β when x_t, u_t follow a FIVAR model with $d_x = 0.4$ and $d_u = 0.2$.

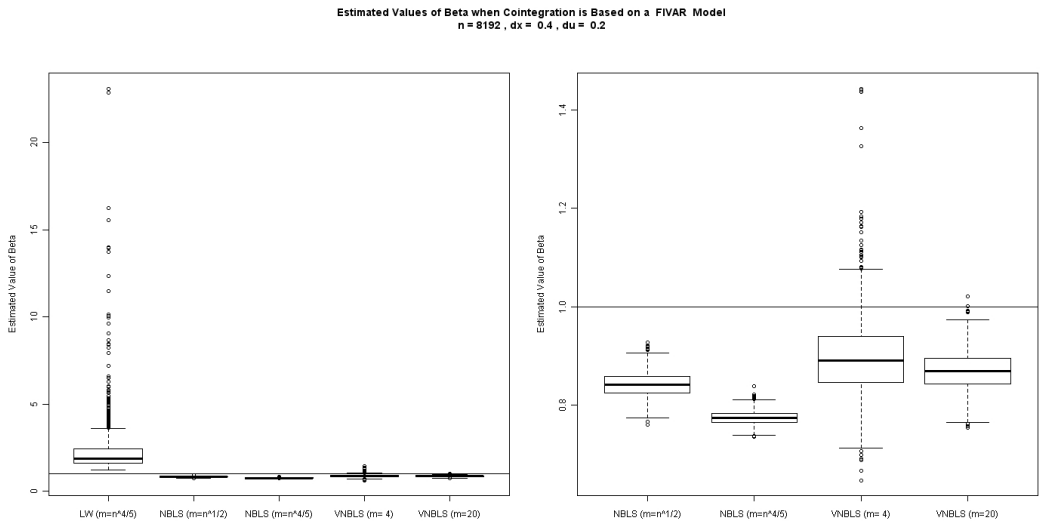

Figure 48: Estimated values of β for the FIVAR model with $d_x = 0.4$ and $d_u = 0.2$, $n = 8192$. The right panel excludes the local Whittle estimator so that the distribution of the VNBLS and NBLS estimators is visible.

n	$m = 4$	$m = 20$	$m = 36$
128	0.252	0.144	0.125
512	0.239	0.21	0.183
2048	0.168	0.175	0.186
8192	0.133	0.138	0.147

Table 44: Root mean squared error of VNBLS, when $(x_t, u_t)'$ are generated by a semilagged FIVAR process with $d_x = 0.4, d_u = 0.2$.

process. As before, VNBLS and NBLS are $O_p(n^{-0.2})$ while the local Whittle estimator is $O_p(n^{-0.4})$ in the best case. The performance of VNBLS for a semilagged FIVAR process is almost identical to its performance for a FIVAR process. NBLS also performs similarly for FIVAR and semilagged FIVAR processes, except that it performs much better for semilagged FIVAR processes with $m = n^{4/5}$. The local Whittle estimator performs much better for semilagged FIVAR processes than for FIVAR processes when $m = n^{2/3}, n^{4/5}$. This result is unexpected, since one might expect that increased group delay would hurt the performance of estimators that used more frequencies. However, even with the improvement in performance, the NBLS and local Whittle estimators is only somewhat smaller than that of VNBLS. Figure 49 shows boxplots of the three estimators for varying choices of m. As before, VNBLS has the least bias of any estimator. As before, the local Whittle estimator is quite variable for smaller samples. Also, the bias of the local Whittle estimator is not strictly decreasing with the sample size.

Figure 50 compares the performance of the estimators when (x_t, u_t) follow a FIVAR model to when (x_t, u_t) follow a semilagged FIVAR model. Both the local Whittle estimator with $m = n^{2/3}$ and NBLS with $m = n^{4/5}$ perform better for the semilagged FIVAR model than for the FIVAR model. However, this local

n	$m = n^{1/6}$	$m = n^{1/3}$	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	1.613	0.237	0.18	0.135	0.111
512	0.325	0.211	0.208	0.115	0.108
2048	0.232	0.168	0.189	0.165	0.11
8192	0.133	0.138	0.164	0.182	0.092

Table 45: Root mean squared error of NBLS, when $(x_t, u_t)'$ are generated by a semilagged FIVAR process with $d_x = 0.4, d_u = 0.2$.

n	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	19.386	15.455	17.117
512	15.324	0.255	5.271
2048	14.348	0.169	0.12
8192	7.232	0.109	0.177

Table 46: Root mean squared error of the local Whittle cointegration estimator with m growing with n, when $(x_t, u_t)'$ are generated by a semilagged FIVAR process with $d_x = 0.4, d_u = 0.2$.

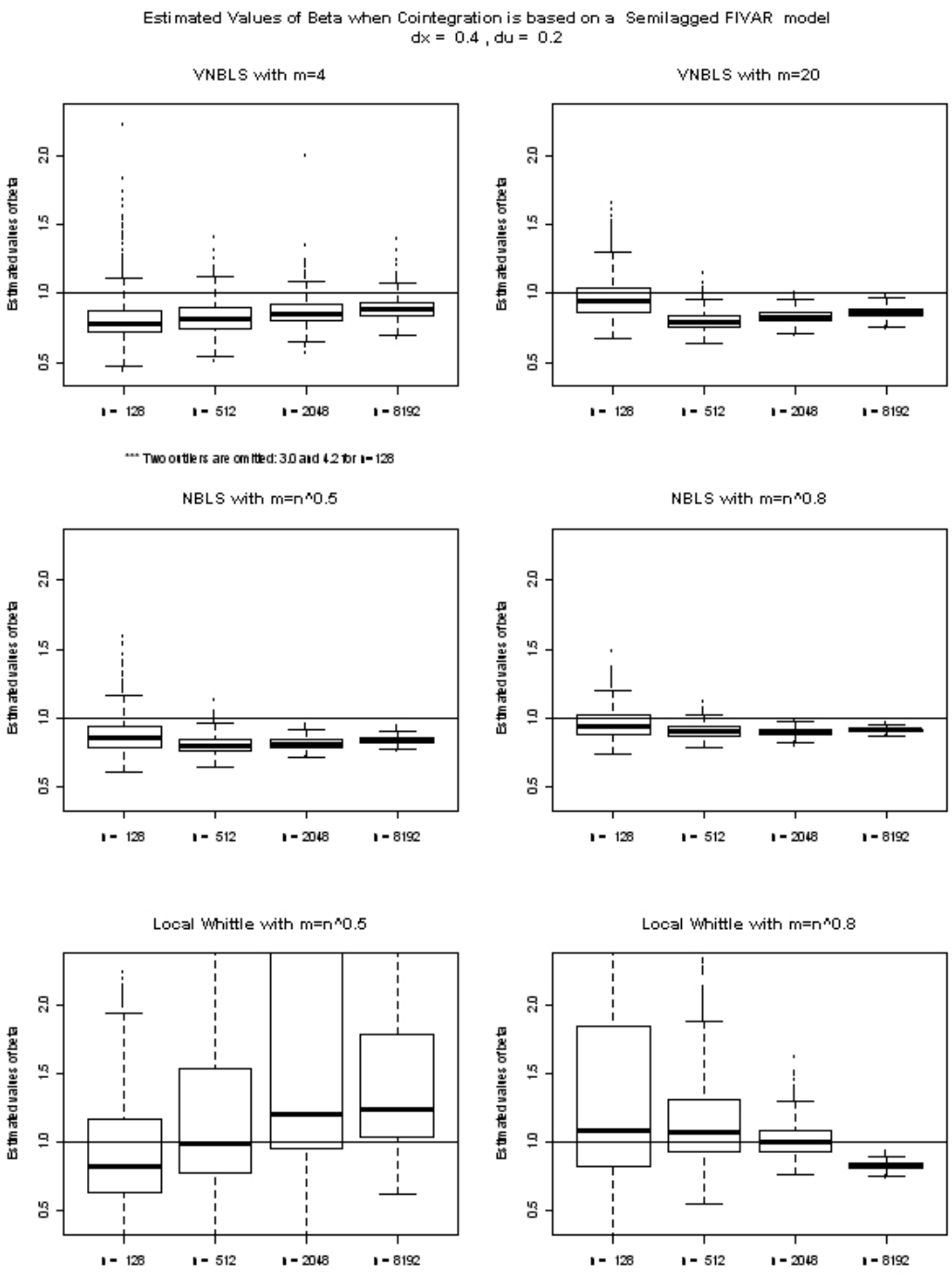

Figure 49: Estimated values of β when (x_t, u_t) follow a semilagged FIVAR model with $d_x = 0.4$ and $d_u = 0.2$.

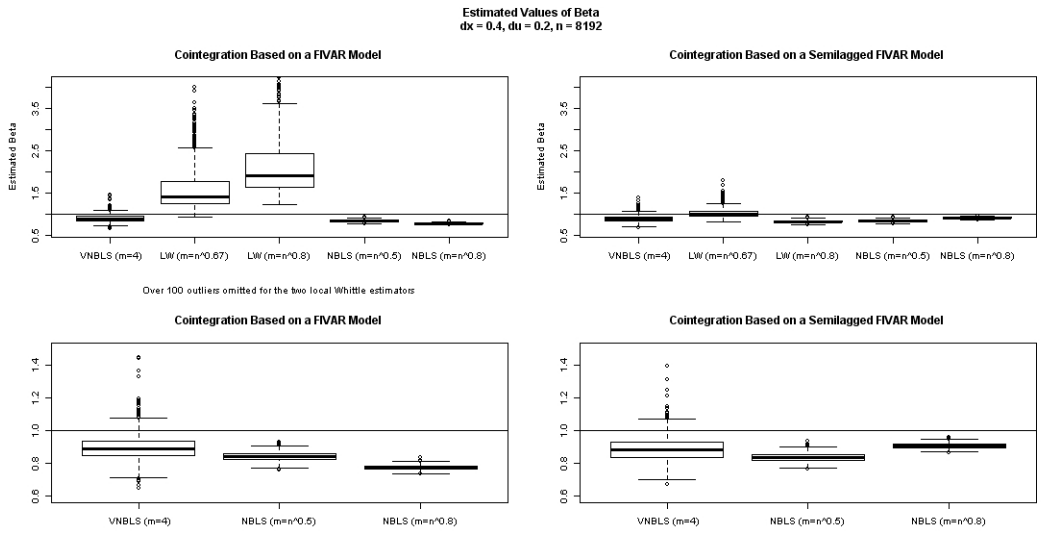

Figure 50: Estimated values of β when (x_t, u_t) follow a FIVAR model or a semi-lagged FIVAR model with $d_x = 0.4$, $d_u = 0.2$, and $n = 8192$.

Whittle estimator was chosen specifically because it had the best performance. In addition, the boxplots shown in Figure 49 suggest that the bias of the local Whittle and NBLS estimators is not monotonically descreasing, leaving the possibility that their performance could degrade for some larger n. In contrast, the performance of VNBLS and NBLS with a smaller growth rate for m are robust to the addition of the lag in the data generating process.

Next, we report the RMSE of the VNBLS and NBLS estimators when $(x_t, u_t)'$ have power law coherency. We will focus on the results when $d_{12} = 0.1$, so that $d_\rho = -0.2$. While VNBLS continues to be $O_p(n^{-0.2})$ in this case, the results of Christensen and Nielsen [2006] require that $\xi \leq -2d_\rho = 0.4$, so that the estimator is $O_p(n^{-0.32})$. Power law coherency is not allowed by Robinson's assumptions and the local Whittle estimator performs badly, so we report only limited results. Tables

47 and 48 show the root mean squared errors for VNBLS and NBLS, respectively. In this case, slightly larger values of fixed m perform better. Figure 51 presents boxplots of the various estimators, including that of the local Whittle estimator. Even with $m = 4$, the estimator generally performs quite well, with very little bias in the larger samples; this likely occurs because the limiting distribution given in Corollary 3.20 has mean zero. However, there are two large outliers in the smaller samples. Since $d_\rho = -0.2$, allowing $m = n^{4/5}$ would yield a growth rate that is too fast to satisfy the conditions of Christensen and Nielsen [2006]; it also leads to larger root mean squared errors because the bias decays quite slowly, as seen in the second row and column of Figure 51. The first column in the second row instead uses $m = n^{1/6}$, which is a growth rate that would be allowed by the smoothness of the coherency according to Christensen and Nielsen [2006]. This choice of m leads to a large number of outliers for smaller n but very little bias. The local Whittle estimator continues to be quite variable when $m = n^{1/6}$ and has a very large upward bias when $m = n^{4/5}$.

In Figure 52, we compare the performance of VNBLS and NBLS for two different values of d_ρ. The bias of NBLS estimators is smaller when $d_\rho = -0.6$ than when $d_\rho = -0.6$, with the bias almost 0 for $m = n^{1/2}$. This occurs because the spectral density is smoother when $d_\rho = -0.6$ than when $d_\rho = -0.2$. The performance of VNBLS does not change substantially between the two values of d_ρ, providing another example of the robustness of VNBLS to the behavior of the cross-spectral density near 0.

Now, we consider the performance of the VNBLS, NBLS and local Whittle estimators in the case where $(x_t, u_t)'$ have a power of λ in the phase, with $\alpha = 0.5$. (The results when $\alpha = 0.1$ have somewhat larger root mean squared errors across all the estimators, but the same patterns occur.) As before, VNBLS and NBLS

n	$m = 4$	$m = 20$	$m = 36$
128	1.63	0.241	0.269
512	0.833	0.156	0.176
2048	0.188	0.101	0.106
8192	0.133	0.073	0.068

Table 47: Root mean squared error of VNBLS, when $(x_t, u_t)'$ have power law coherency with $d_x = 0.4, d_u = 0.2, d_\rho = -0.2$.

n	$m = n^{1/6}$	$m = n^{1/3}$	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	2.408	0.317	0.223	0.25	0.284
512	2.445	0.165	0.158	0.205	0.253
2048	0.267	0.108	0.11	0.159	0.221
8192	0.133	0.073	0.076	0.128	0.196

Table 48: Root mean squared error of NBLS, when $(x_t, u_t)'$ have power law coherency with $d_x = 0.4, d_u = 0.2, d_\rho = -0.2$.

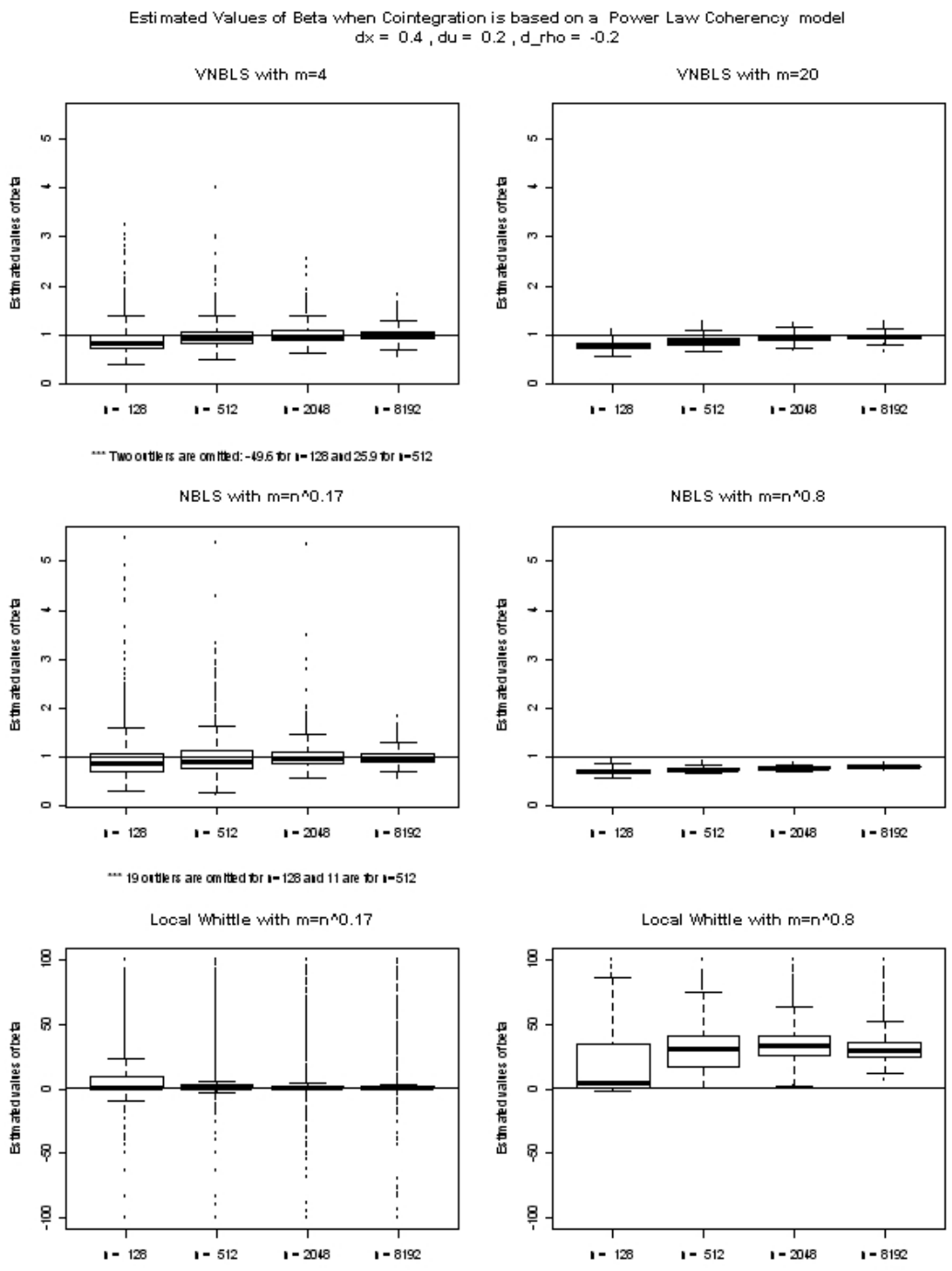

Figure 51: Estimated values of β when (x_t, u_t) have power law coherency with $d_x = 0.4, d_u = 0.2, d_\rho = -0.2$.

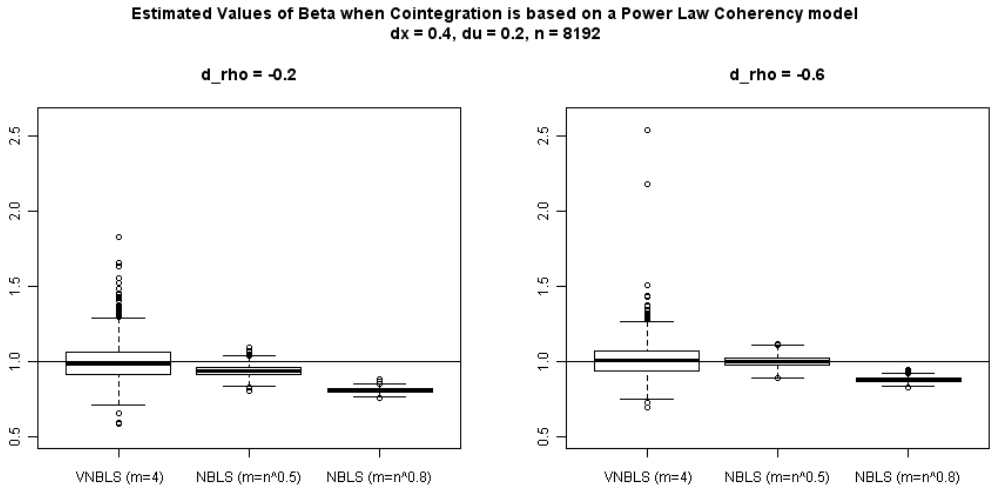

Figure 52: Estimated values of β when (x_t, u_t) have power law coherency with $d_x = 0.4, d_u = 0.2, n = 8192$, for different values of d_ρ.

are $O_p(n^{-0.2})$. In this case, the local Whittle estimator requires that $\xi < \alpha = 0.5$, so his estimator must be $O_p(n^{-0.35})$ at best. In Table 49, we find that smaller fixed values of m are preferable in both the smallest and largest samples. Figure 53 shows boxplots of the various estimators. As before, VNBLS and NBLS are biased downward; the bias is greatest when $m = n^{4/5}$. Table 50 shows that smaller growth rates of m have lower root mean squared errors as the sample size grows, as we have seen before. The local Whittle estimator continues to perform badly, as shown in Table 51, but its performance improves with the sample size, as can be seen in the bottom row of Figure 53.

Figure 54 plots the estimated values from VNBLS and NBLS for $\alpha = 0.1$ and $\alpha = 0.5$. The performance is similar across the two values of α, but the bias of NBLS is larger when $\alpha = 0.1$. In the same cases, the local Whittle estimator (not shown) is biased upward, more variable than VNBLS and NBLS, with the bias and

n	$m = 4$	$m = 20$	$m = 36$
128	0.283	0.324	0.366
512	0.23	0.241	0.26
2048	0.205	0.182	0.195
8192	0.133	0.14	0.149

Table 49: Root mean squared error of VNBLS, when $(x_t, u_t)'$ are generated by a process with a power of λ in the phase with $d_x = 0.4, d_u = 0.2, \alpha = 0.5$.

n	$m = n^{1/6}$	$m = n^{1/3}$	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	0.6	0.282	0.296	0.338	0.392
512	6.812	0.22	0.244	0.287	0.345
2048	0.225	0.173	0.2	0.246	0.306
8192	0.133	0.14	0.168	0.213	0.273

Table 50: Root mean squared error of NBLS, when $(x_t, u_t)'$ are generated by a process with a power of λ in the phase with $d_x = 0.4, d_u = 0.2, \alpha = 0.5$.

variability larger when $\alpha = 0.1$ than when $\alpha = 0.5$. As before, the performance of VNBLS is robust, in this case to changes in α.

In order to compare the performance of the estimators when the cross-spectral density is unknown, we compute the maximum root mean squared error of each estimator over all of the cases described above. Then, for each n, we determine which estimator has the smallest maximum (minimax) RMSE. When $n = 128$, the estimator with the minimax RMSE is NBLS with $m = n^{1/2}$. When $n = 512$, the estimator with the minimax RMSE is NBLS with $m = n^{1/3}$. When $n = 2048$ and when $n = 8192$, the minimax estimator is VNBLS with $m = 4$. In all cases, the maximum occurs when $\lambda^{0.1}$ appears in the phase. Thus, in small samples, NBLS with relatively small powers is most robust to a variety of behaviors of the

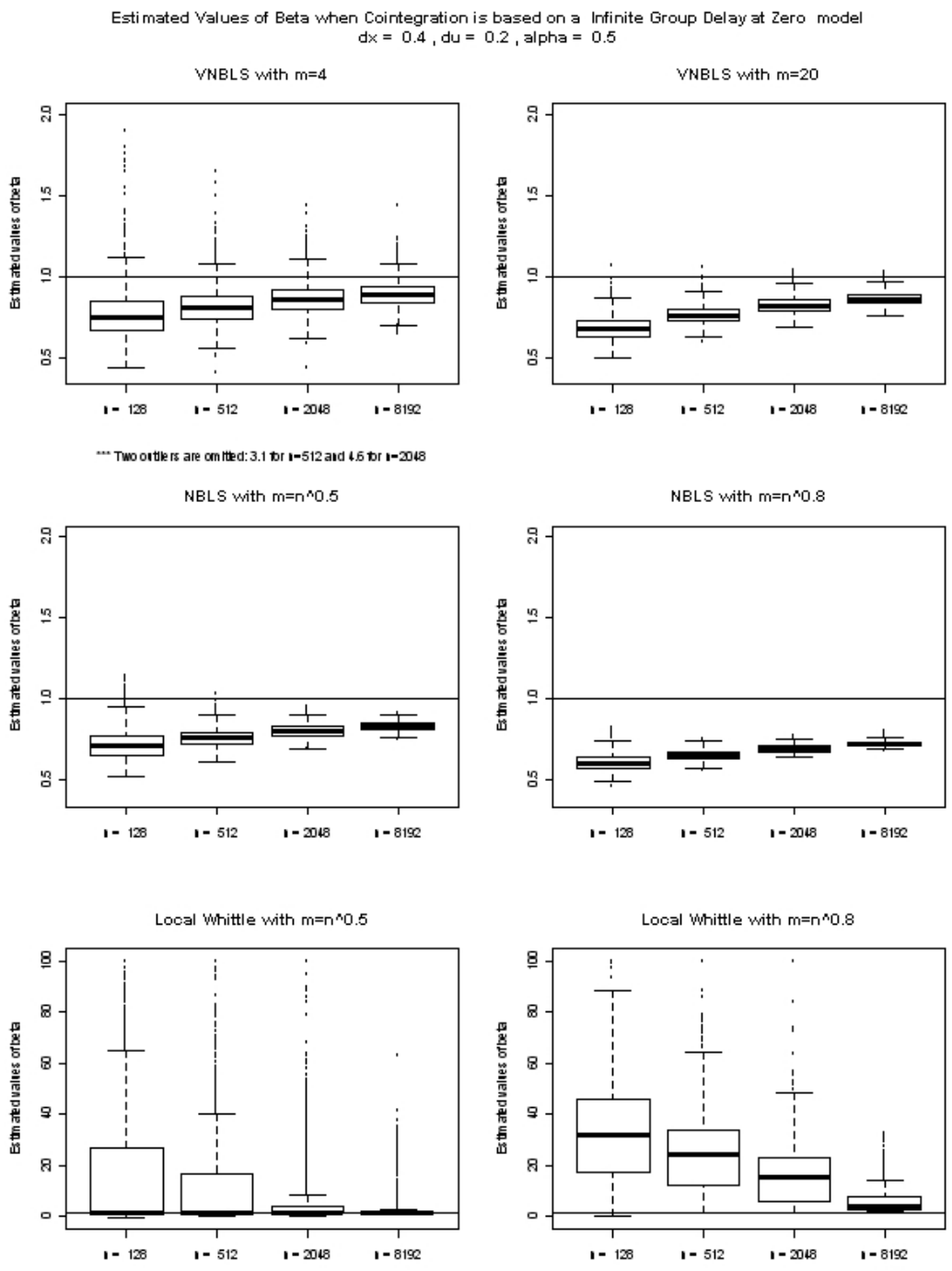

Figure 53: Estimated values of β for cointegration when (x_t, u_t) follow a process with a power of λ in the phase with $d_x = 0.4, d_u = 0.2, \alpha = 0.5$.

n	$m = n^{1/2}$	$m = n^{2/3}$	$m = n^{4/5}$
128	33.274	36.904	38.419
512	23.253	21.523	28.66
2048	16.116	10.826	19.335
8192	6.182	2.129	7.324

Table 51: Root mean squared error of the local Whittle cointegration estimator with m growing with n, when $(x_t, u_t)'$ are generated by a process with a power of λ in the phase with $d_x = 0.4, d_u = 0.2, \alpha = 0.5$.

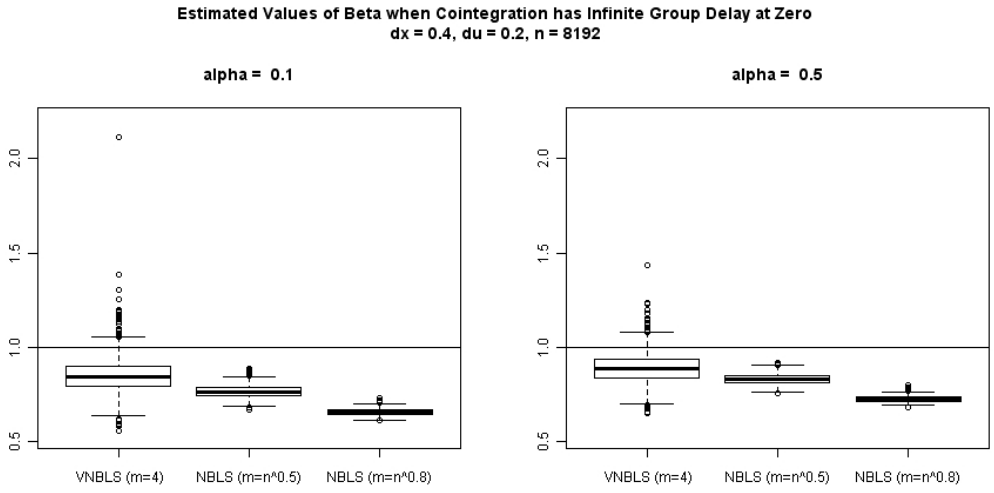

Figure 54: Estimated values of β for cointegration when (x_t, u_t) follow a process with a power of λ in the phase with $d_x = 0.4, d_u = 0.2, n = 8192$, for varying values of α.

cross-spectral density. For larger samples, VNBLS with a small choice of m is most robust.

3.5 Data analysis

3.5.1 Phase and coherency in practice: Money supply growth

We examine monthly estimates of money stock from January 1959 to September 2009[3] (608 observations). We focus on two different supplies of money. M1 consists of easily accessible money, such as currency and demand deposits, while M2 consists of M1 together with forms of money that require more time to access, such as savings deposits and money market accounts. In order to remove the component common to M1 and M2, we focus on describing the relationship between M1 and M2 less M1. Both series have clear upward trends in their levels; we will work with the difference in logs, shown in Figure 55. The plot shows some common movements, such as a period of comovement in the late 1960's and mid-1970's. However, the long run movements are less related, with M1 growing faster in the mid-1980's and mid-1990's but M2 less M1 growing faster in other periods. The autocorrelation functions (shown in Figure 56) decay slowly, with the cross-correlation function appearing to decay more quickly. Also, the peak cross-covariance occurs at a lag of about 9, suggesting that group delay might occur. The logarithms of the auto-periodograms (Figure 57) have an approximately linear relationship with the log frequency near frequency zero, suggesting that the individual series have long memory; it is unclear from this figure which series has a larger memory parameter. GPH estimates of the memory parameters of the individual series are quite sensitive to the number of frequencies used; a variety of choices of m are shown in Table 52. The GPH estimates suggest the presence of long memory in

[3]Source: Federal Reserve, `http://www.federalreserve.gov/releases/h6/hist/h6hist1.txt`

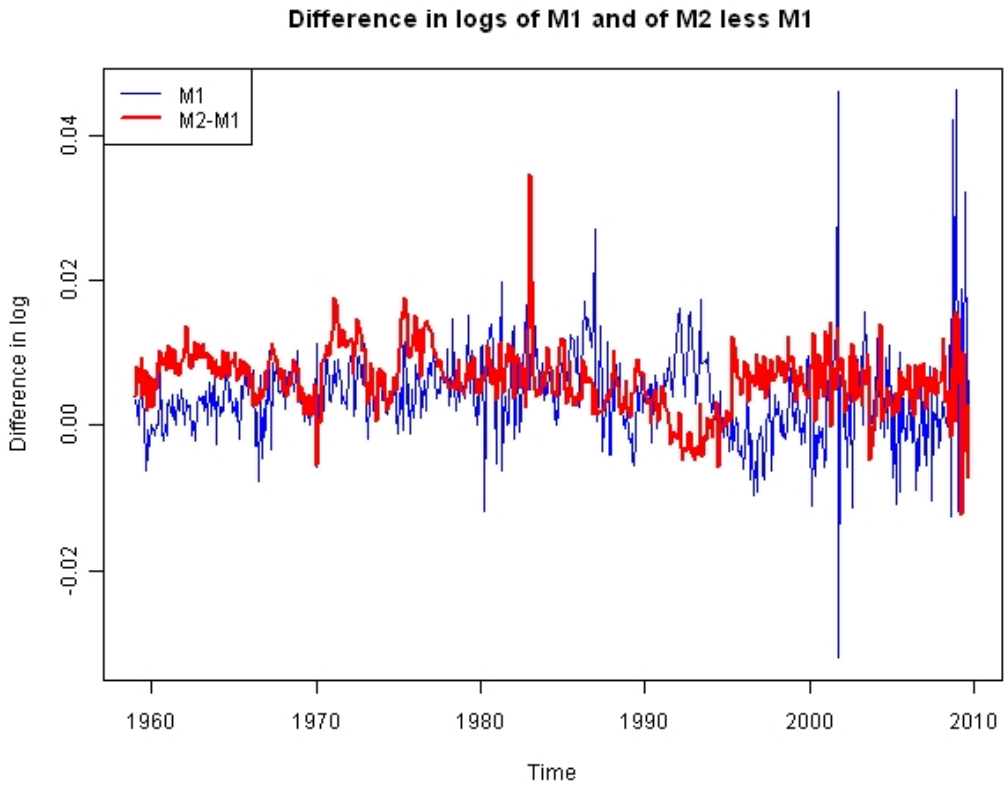

Figure 55: Time series of differences in logs of M1 and M2-M1.

the individual series. We could test for equality of the memory parameters using the results of Robinson [1995a], but those results would be limited in the presence of power law coherency or powers of λ in the phase, as discussed in Section 3.2.1.

To estimate the coherency and phase, we first smooth the periodogram, using the `spgram` function in R [R Development Core Team, 2008] with modified Daniell smoothers of widths $(21, 21)$. Though the smoothing is likely to be problematic close to the zero frequency because of the long memory, it gives us a general idea of the shape [Hidalgo, 1996]. The coherency of the two series is not significantly different from zero except at frequencies ranging from approximately $0.01(2\pi)$ to

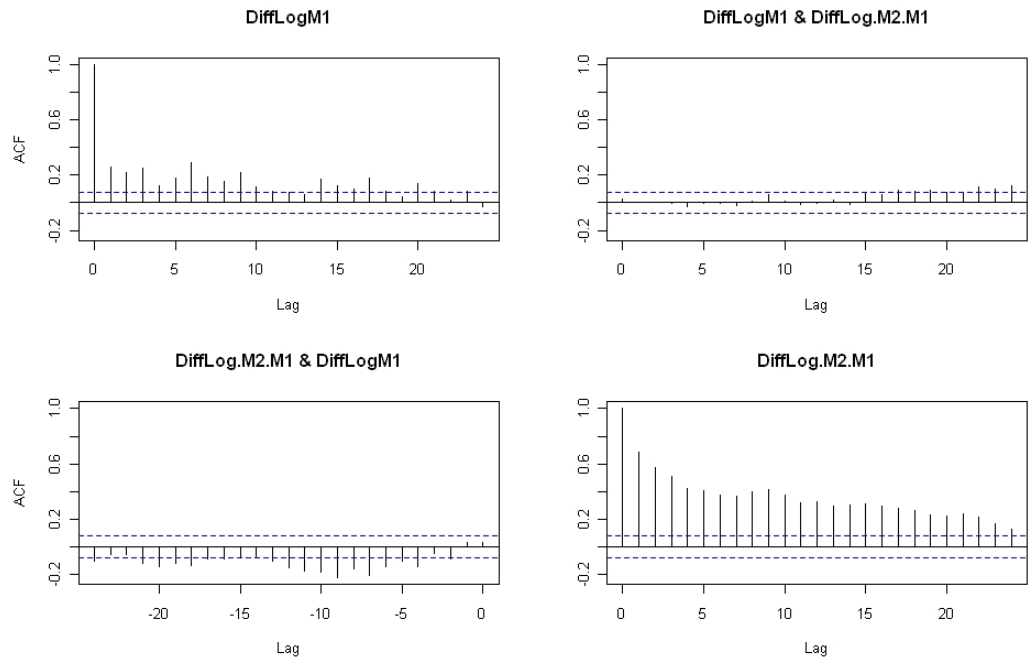

Figure 56:]

Auto- and cross-correlation functions of the differences in logs of M1 and M2-M1.

	d_{M1}	d_{M2-M1}
$n^{1/2} = 24$	0.115 (0.241)	0.426 (0.142)
$n^{3/5} = 46$	0.081 (0.126)	0.323 (0.099)
$n^{2/3} = 71$	0.157 (0.089)	0.342 (0.083)
$n^{3/4} = 122$	0.291 (0.060)	0.237 (0.060)
$n^{4/5} = 168$	0.457 (0.061)	0.174 (0.045)

Table 52: GPH estimates of d for M1 and M2-M1 for varying powers of n. Standard errors from regression given in parentheses.

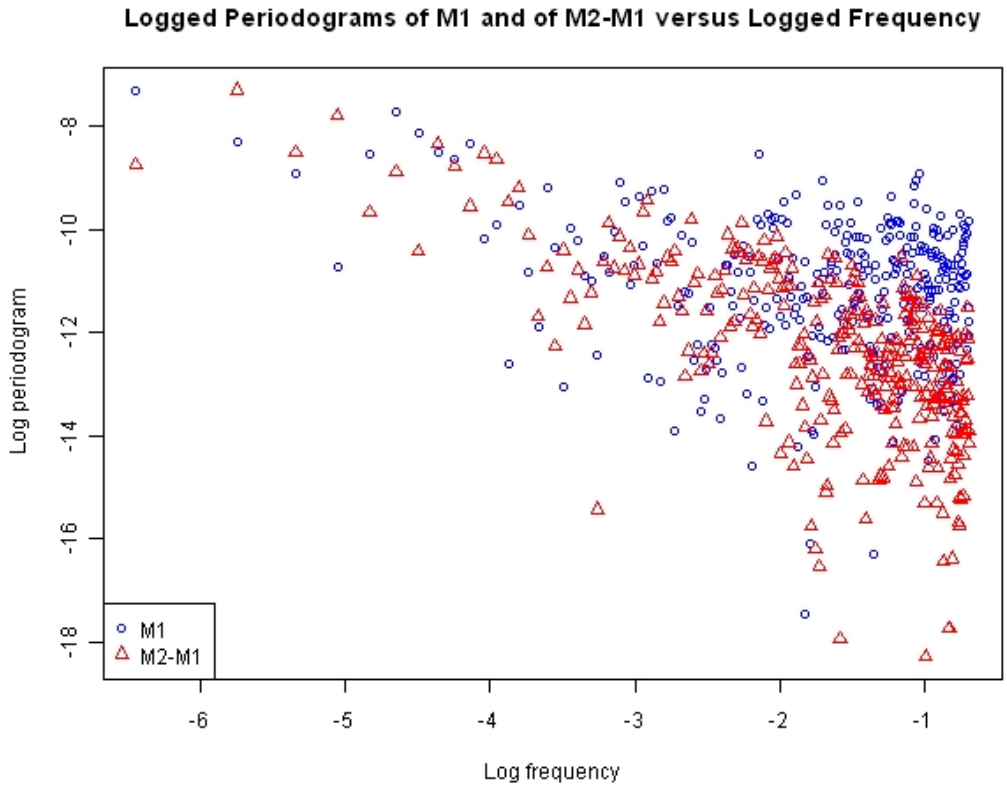

Figure 57: Log auto-periodograms of the differences in logs of M1 and M2-M1 versus the log frequency.

approximately $0.09(2\pi)$ (periods ranging from just under 1 year to just over 8 years). The coherency peaks just above 0.35 around frequency $0.06(2\pi)$, which corresponds to a period of just over 16 months. This suggests that only the longer run movements of M1 and M2-M1 (with periods greater than 1 year) are related. However, the coherency decreases toward zero at the zero frequency, suggesting power law coherency, so that very long-run movements are not related. The estimated phase is shown in Figure 59. Because the coherency is close to zero over most of the range, the phase estimates are quite noisy. In the range of frequencies where the coherency is larger, the phase shows a clear upward slope, suggesting that M1 growth leads growth in M2-M1.

The phase and coherency have straightforward economic interpretations. M1 and M2-M1 move together at business cycle frequencies, with M1 leading M2-M1. M1 may lead M2-M1 because changes in the money stock might appear in short-term deposits first. In the long-run, the relationship between the two declines; this could occur because people's preferences for holding M1 versus M2-M1 versus other assets might change over long periods.

With this evidence for power law coherency, we can apply the averaged periodogram estimator for varying choices of m, holding q fixed at 0.5. The estimated values of the memory parameters of the individual series are in the range of the GPH estimates and vary less as m changes. However, the estimated power law in the cross-spectrum is always larger than the mean of the two auto-memory parameters, which appears to rule out power law coherency. As we saw in Section 3.3.3, identification of power law coherency is quite challenging even when $n = 8192$; in this case, $n = 608$. Thus, it is not clear that we can rule out power law coherency, even though $\hat{d}_\rho = 0$.

To provide further evidence that the averaged periodogram estimator may not

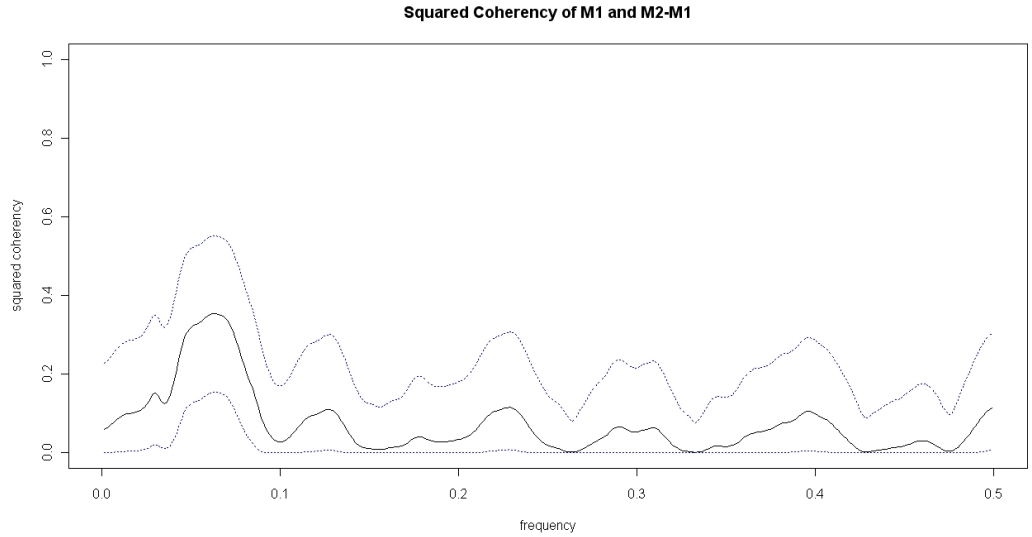

Figure 58: Estimated coherency of the differences in logs of M1 and M2-M1 (smoothed using $spans = (21, 21)$).

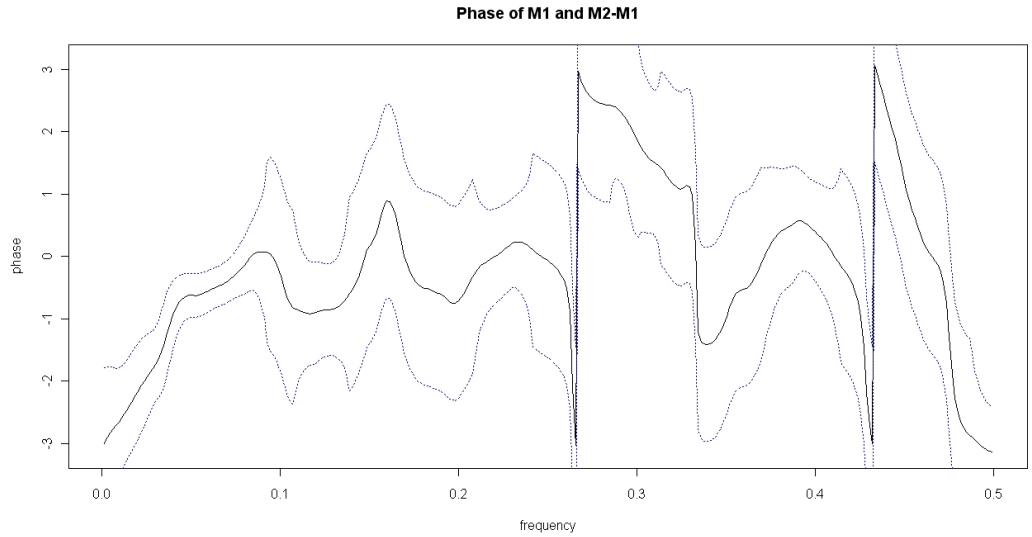

Figure 59: Estimated phase of the differences in logs of M1 and M2-M1 (smoothed using $spans = (21, 21)$).

m	$\hat{d}_{M1}$	$\hat{d}_{M2-M1}$	$\hat{d}_{M1,M2-M1}$
$n^{1/2} = 24$	0.347	0.375	0.409
$n^{3/5} = 46$	0.326	0.285	0.335
$n^{2/3} = 71$	0.335	0.300	0.425
$n^{3/4} = 122$	0.206	0.336	0.329
$n^{4/5} = 168$	0.269	0.379	0.358

Table 53: APE estimates of d_{M1}, d_{M2-M1}, and $d_{M1,M2-M1}$ for varying choices of m.

be able to find evidence of a power law in the coherency in this dataset, we simulate 1000 datasets with $n = 608, d_1 = 0.15, d_2 = 0.3$, and varying values of d_ρ, using the anti-cointegration model. For each dataset, we estimated d_ρ for the values of m used in Table 53. In Table 54, we report the proportion of times that $\hat{d}_\rho < 0$ in the sample; the probability is highest for $n^{4/5}$ when $d_\rho = -0.675$, and that probability is only 0.8. In many cases, the probability that the point estimate is non-zero is under 0.5. In Figure 60, we plot the estimated values of d_ρ for varying true values of d_ρ. The estimated values become more spread out for more negative values of d_ρ. In all cases, the estimates are biased upward, with the bias particularly pronounced for more negative values of d_ρ. This shows that the performance of the APE is quite poor in a sample of this size.

This dataset provides graphical evidence based on the smoothed periodogram that power law coherency may exist. However, the averaged periodogram estimator is not able to identify power law coherency in a sample of this size, as shown in simulation.

d_ρ	$n^{1/2} = 24$	$n^{3/5} = 46$	$n^{2/3} = 71$	$n^{3/4} = 122$	$n^{4/5} = 168$
0	0.239	0.186	0.183	0.135	0.095
0	0.181	0.173	0.146	0.086	0.058
-0.025	0.271	0.228	0.210	0.159	0.133
-0.125	0.293	0.324	0.402	0.394	0.387
-0.225	0.278	0.368	0.447	0.520	0.586
-0.325	(0.218)	0.298	0.426	0.626	0.745
-0.425	(0.208)	(0.271)	0.371	0.653	0.799
-0.675	(0.191)	(0.192)	(0.221)	0.514	0.804
-0.925	(0.148)	(0.161)	(0.209)	(0.437)	0.731

Table 54: Proportion of simulations in which $\hat{d}_\rho < 0$. Numbers in parentheses indicate that the power of n used is too small for the APE to be consistent for the particular d_ρ.

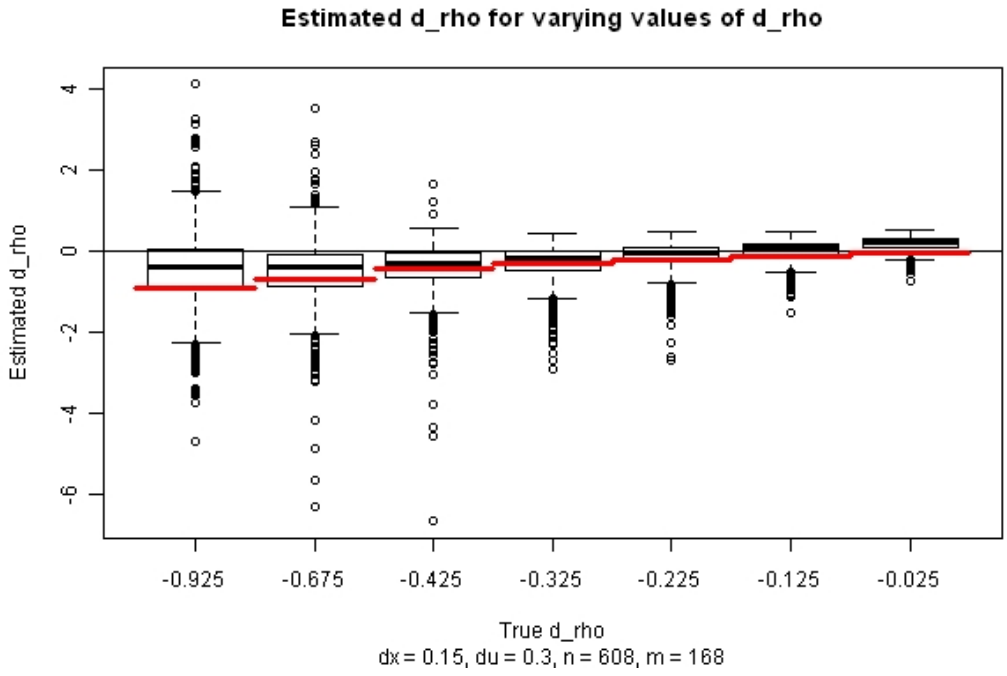

Figure 60: Estimated values of d_ρ in simulations for varying values of d_ρ.

3.5.2 Cointegration: High and low stock prices

Now, we test for cointegration between daily high and low stock prices for the S&P 500 index from January 1962 through October 2009 (12025 observations after differencing). Cheung [2007] considered the possibility of cointegration between high and low stock prices, assuming that prices were $I(1)$ and looking for cointegrating relationships that were $I(0)$. However, it is possible that fractional cointegration occurs instead. In particular, the log range, defined as the difference between the log of the high stock price and the log of the low stock price over an interval, is an estimator of stock volatility [Alizadeh et al., 2002], which is often thought to have long memory [Breidt et al., 1998, Hurvich and Soulier, 2009, among many others].

Figure 61 shows the unsmoothed periodograms of the difference in the logs of the high and low stock prices. The two series appear to be $I(0)$ and close to white noise, as would be expected for stock prices. Figure 62 plots the log periodograms versus the log frequencies; this plot also suggests that the series are $I(0)$. At the higher frequencies, the periodograms are not quite flat, suggesting that the log differences in high and low prices are not quite white noise (they need not be, since they are not stock returns). In Figure 63, we plot the estimated coherency of the two series. The estimated coherency is one at the zero frequency, providing graphical evidence of cointegration. The coherency then declines until it is zero at the high frequencies, suggesting that the high and low stock prices are almost unrelated at high frequencies. We plot the phase in Figure 64. The phase is flat at low frequencies, suggesting that the high and low stock prices move together, as one would expect. At higher frequencies, the phase is quite variable; as before, the low coherency leads to variable estimates of the phase.

We now estimate the cointegrating relationship between the two series. Estimated values of β for a variety of choices of m are given in Table 55 for the NBLS

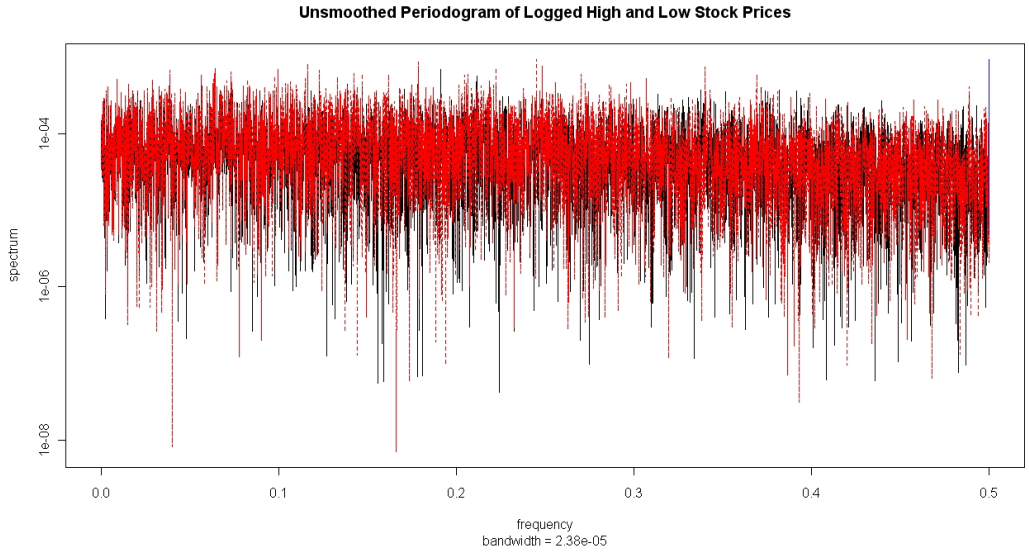

Figure 61: Auto-periodograms of the log differences of daily high and low stock prices.

and local Whittle estimator. The estimated values are all close to one, as one might expect. In contrast, the OLS estimate is $\hat{\beta} = 0.800$ with a standard error of 0.007, demonstrating the bias of OLS in this case.

Next, we consider the log range, defined to be the log high less the log low; this is the cointegrating relationship if $\beta = 1$. We now study the relationship between the log high and the log range. Figure 65 shows the unsmoothed logged periodogram of the log range after differencing and tapering; the log periodogram has an approximately linear relationship with the log frequency for smaller ordinates, suggesting that the memory parameter of the log range is negative. This provides graphical evidence that cointegration does exist. Table 56 shows the estimated memory parameters of the auto-spectra and cross-spectrum for various choices of m; we taper the data because we have differenced it once. The estimated memory parameter of the log high is around 0, while the estimated memory parameter of

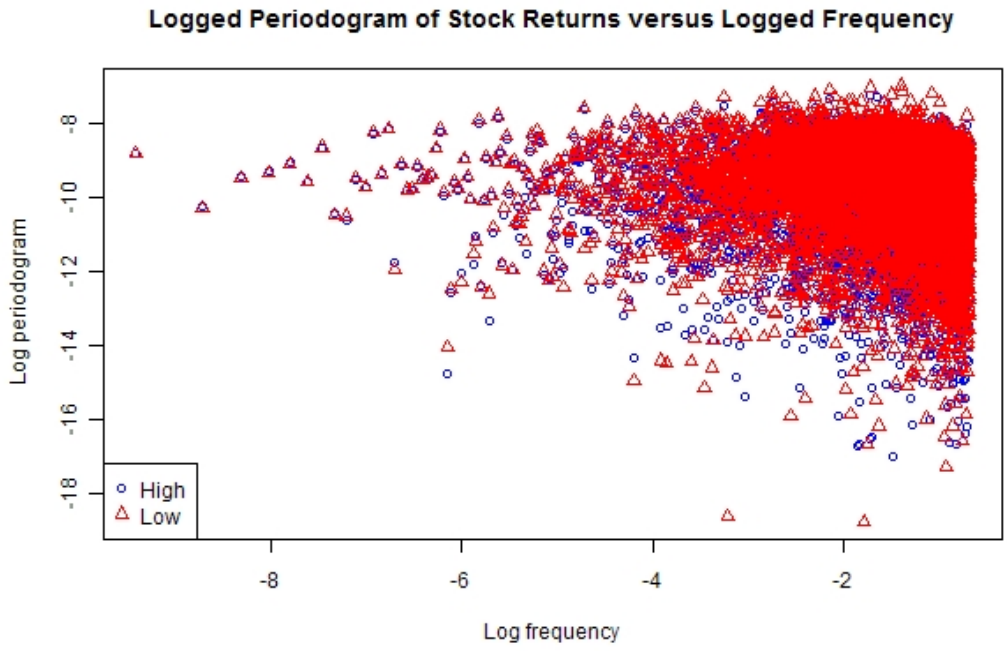

Figure 62: Log periodograms of the log differences of daily high and low stock prices versus the log frequency.

m	$\hat{\beta}_{NBLS}$	$\hat{\beta}_{NBLS,taper}$	$\hat{\beta}_{LW}$
4	1.004	0.999	-
10	0.997	1.002	-
$n^{1/3} = 23$	0.991	0.997	1.015
$n^{1/2} = 110$	0.968	0.980	1.011
$n^{2/3} = 525$	0.938	0.953	1.012
$n^{4/5} = 1837$	0.936	0.963	1.010

Table 55: NBLS and local Whittle estimates of the cointegrating relationship between high and low stock prices for varying values of m.

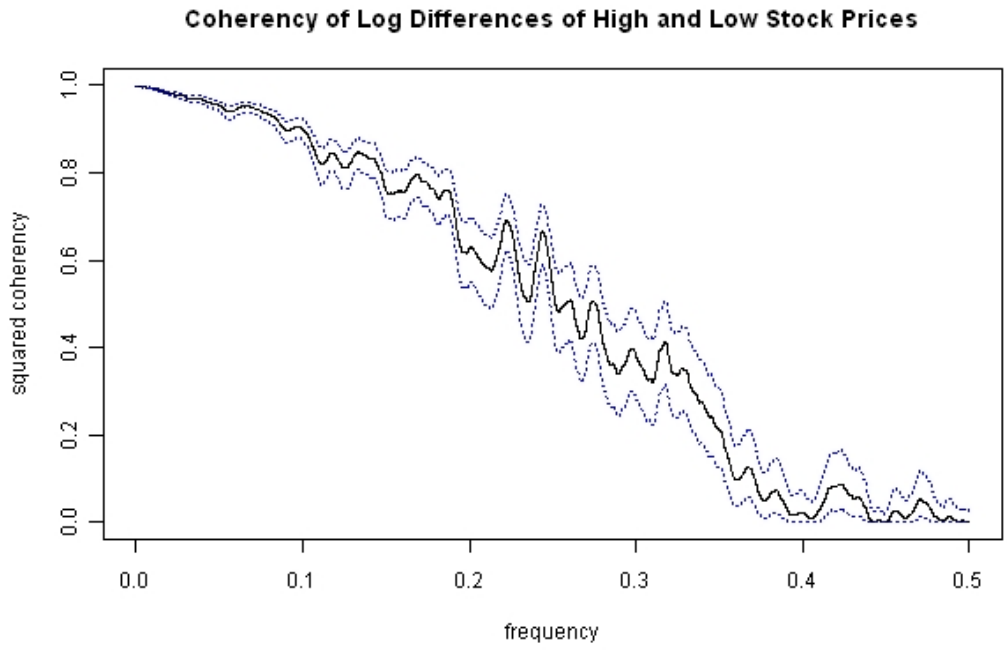

Figure 63: Estimated coherency of the log differences of daily high and low stock prices; smoothing with $spans = (91, 91)$.

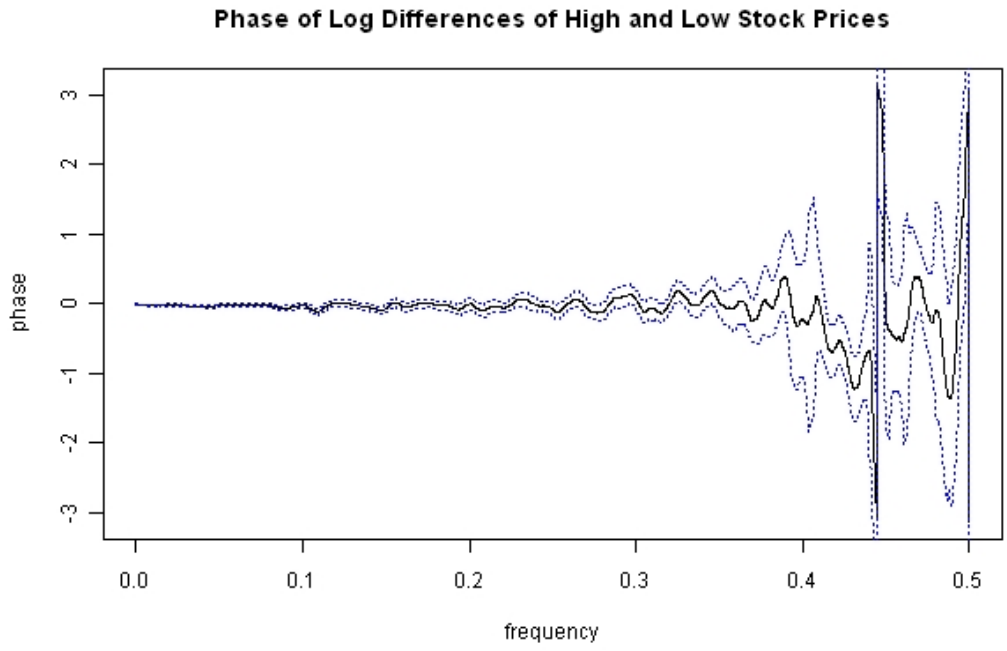

Figure 64: Estimated phase of the log differences of daily high and low stock prices.

	APE			Local Whittle	
	$\hat{d}_{High}$	$\hat{d}_{Range}$	$\hat{d}_{High,Range}$	$\hat{d}_{High}$	$\hat{d}_{Range}$
$n^{1/2} = 109$	0.021	-0.748	-0.232	0.034	-0.460
$n^{3/5} = 280$	0.064	-0.552	0.021	-0.016	-0.425
$n^{2/3} = 524$	0.013	-0.606	-0.172	-0.018	0.450
$n^{3/4} = 1148$	-0.038	-0.548	0.032	-0.040	-0.482
$n^{4/5} = 1836$	0.018	-0.668	-0.021	-0.032	-0.490

Table 56: APE estimates of d_{High}, d_{Range}, and $d_{High,Range}$ for varying powers of m. Local Whittle estimates for d_{High} and d_{Range} using the estimator of Robinson [2008]. Data are tapered with the Hurvich and Chen taper of order 1 for the APE.

the log range ranges from -0.75 to -0.55, dramatically lower than 0 but higher than -1, indicating fractional cointegration. Figure 66 shows the coherency between the log range and the log high. The coherency is highest at high frequencies, drops to zero, and then increases to about 0.2 near zero frequency. Figure 67 shows the phase between the log range and the log high. The phase is approximately zero for the high frequencies and then approaches π at frequency zero. Thus, in the short run, the range and the high are positively related, while in the long run the range and the high are negatively related. In addition, Figure 67 suggests the possibility of a power of λ in the phase at zero. Figure 68 plots the log phase, adjusted to center it at zero and make the slope positive, versus the log frequency. There does appear to be a straight line relationship; $\hat{\alpha} = 0.975$ (with a standard error of 0.002), based on a linear regression on the first 30 Fourier frequencies. Since $\hat{\alpha}$ is approximately one, there may be finite, non-zero group delay at frequency 0, instead of a power of λ that would lead to infinite group delay at frequency 0. Some of the behavior may be an artifact of smoothing.

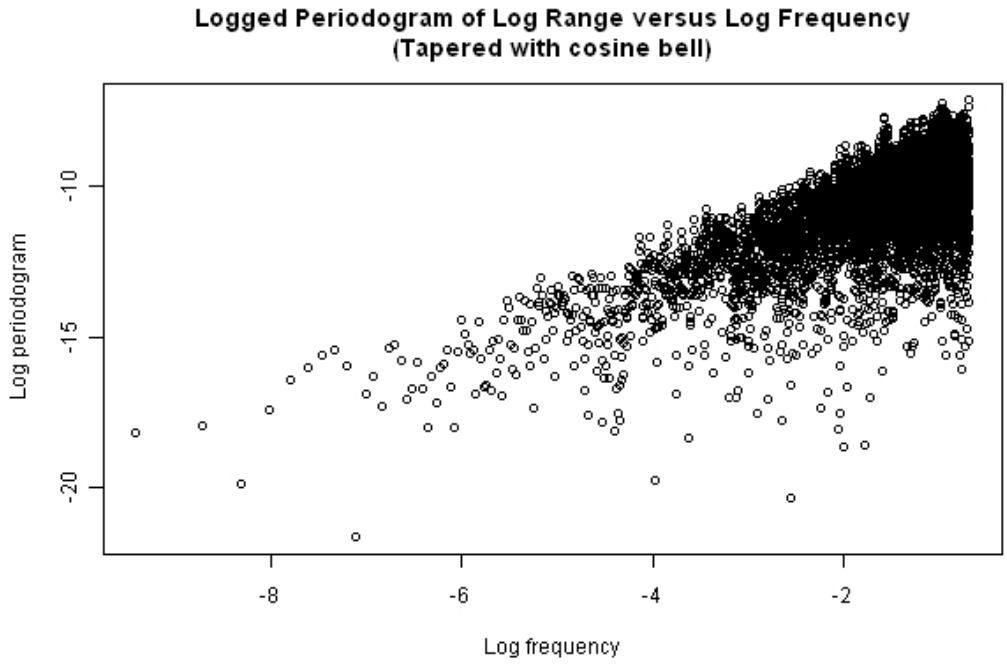

Figure 65: Log periodogram versus log frequency for the differenced log range.

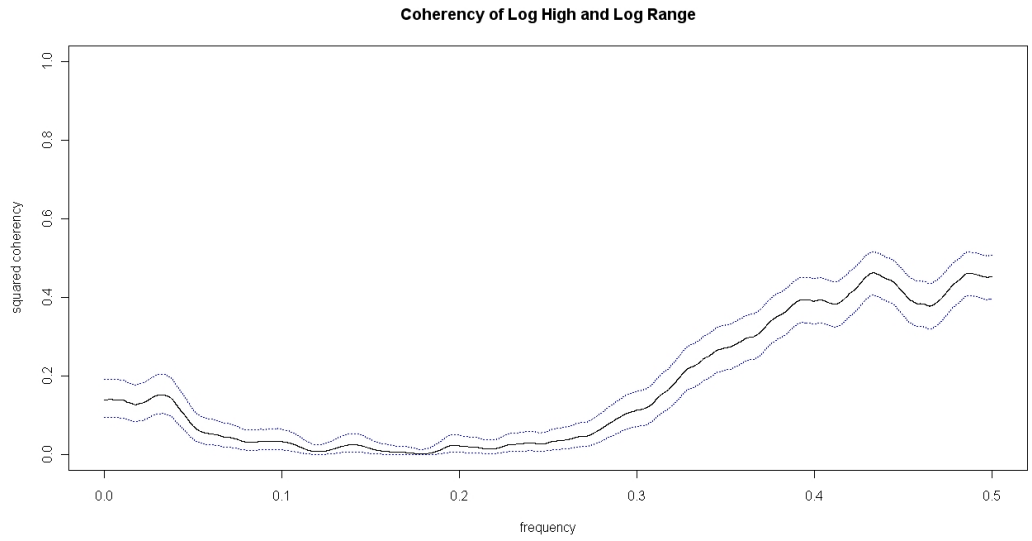

Figure 66: Coherency of the differenced log range and differenced log high with $spans = (251, 251)$.

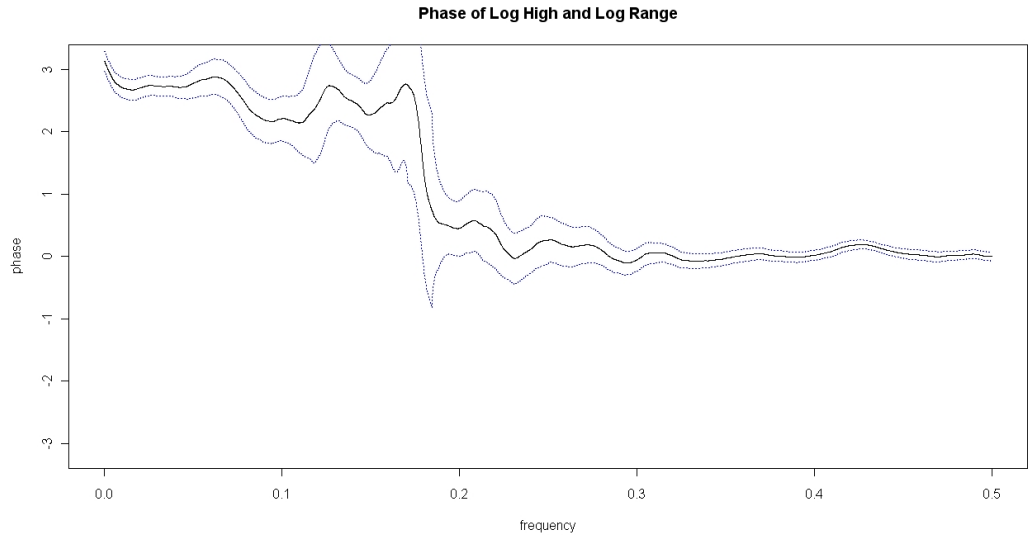

Figure 67: Phase of the differenced log range and differenced log high with $spans = (251, 251)$.

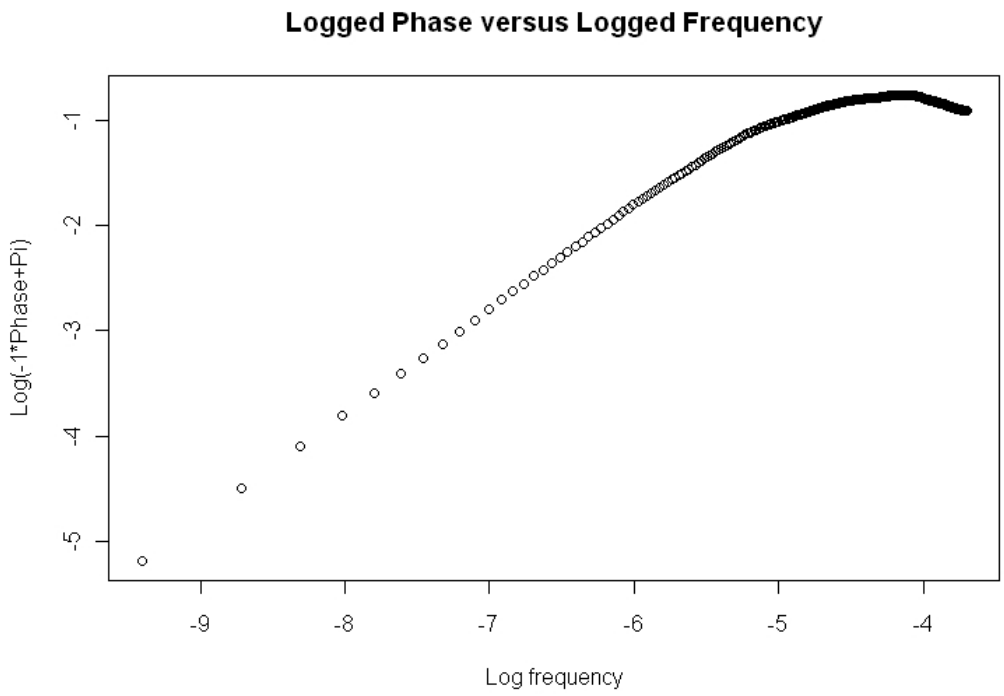

Figure 68: Log of $\pi - \hat{\phi}(\lambda)$ versus log frequency for frequencies up to 0.05π.

3.6 Conclusion

In this paper, we have discussed the possibility of a power law in coherency and powers of λ in the phase for bivariate long-memory time series. We have described the implications for the interpretation of the time series behavior and provided time-domain examples. The average periodogram estimator provides a possible estimator for the power law in the coherency, but can be quite variable in small samples. We have also discussed the challenges of estimating the cointegrating relationship between two time series when there is possible power of λ in the phase or power law coherency; the very narrow-band least squares estimator provides solves the problem of needing to restrict the growth rate of m to accommodate unknown powers in the phase and coherency. Finally, we have applied our estimators to two bivariate time series: money stock measures and high and low stock prices.

Because the phase and coherency have not received much attention in the recent literature, many possibilities for future research remain. First, better estimation techniques may exist; either GPH based on the smoothed cross-periodogram or local Whittle estimators that explicitly describe the local-to-zero behavior of the phase and coherency may be useful in some cases. Second, this paper has been limited to the study of bivariate time series; more challenges may appear when there are three or more time series.

The phase and coherency can provide important insights into the relationships of two time series, which can help to understand the underlying mechanisms that generate them. Powers of λ and power laws in the phase and coherency arise naturally in the context of long-memory time series, allowing for particularly interesting relationships. Plots of the phase and coherency should be included among those used by practitioners and their behavior should be considered before choosing an estimator whenever possible.

3.7 Technical Lemmas

Proof of Lemma 3.8. Given any $\zeta \in (-\pi, \pi]$, define $L_R\left(\frac{1}{\lambda};\zeta\right) = \Re\left(L\left(\frac{1}{\lambda}\right)e^{i\zeta}\right)$ and $L_I\left(\frac{1}{\lambda};\zeta\right) = \Im\left(L\left(\frac{1}{\lambda}\right)e^{i\zeta}\right)$, so that $f_{12}(\lambda) = e^{-i\zeta}\left(L_R\left(\frac{1}{\lambda};\zeta\right) + iL_I\left(\frac{1}{\lambda};\zeta\right)\right)\lambda^{-2d_{12}}$. Then,

$$F_{12}(\lambda) \;=\; e^{-i\zeta}\left[\int_0^\lambda L_R\left(\frac{1}{\lambda};\zeta\right)\theta^{-2d_{12}}d\theta + i\int_0^\lambda L_I\left(\frac{1}{\lambda};\zeta\right)\theta^{-2d_{12}}d\theta\right] \quad (3.39)$$

Choose ζ such that $\lim_{\lambda\to 0^+} L_R\left(\frac{1}{\lambda};\zeta\right) \neq 0$ and $\lim_{\lambda\to 0^+} L_I\left(\frac{1}{\lambda};\zeta\right) \neq 0$. Since $L(z)$ is bounded away from 0, such a ζ always exists. Then, $L_R\left(\frac{1}{\lambda};\zeta\right), L_I\left(\frac{1}{\lambda};\zeta\right)$ are both slowly varying functions. Applying Karamata's theorem [see, for example, Bingham et al., 1989],

$$\begin{aligned}\int_0^\lambda L_R\left(\frac{1}{\lambda};\zeta\right)\theta^{-2d_{12}}d\theta &\sim \frac{L_R\left(\frac{1}{\lambda};\zeta\right)}{1-2d_{12}}\lambda^{1-2d_{12}} \\ \int_0^\lambda L_I\left(\frac{1}{\lambda};\zeta\right)\theta^{-2d_{12}}d\theta &\sim \frac{L_I\left(\frac{1}{\lambda};\zeta\right)}{1-2d_{12}}\lambda^{1-2d_{12}}\end{aligned}$$

Substituting these integrals into the right-hand-side of Equation (3.39), we find that:

$$\begin{aligned}F_{12}(\lambda) &\sim e^{-i\zeta}\frac{\lambda^{1-2d_{12}}}{1-2d_{12}}\left(L_R\left(\frac{1}{\lambda};\zeta\right) + iL_I\left(\frac{1}{\lambda};\zeta\right)\right) \\ &= \frac{\lambda^{1-2d_{12}}}{1-2d_{12}}L\left(\frac{1}{\lambda}\right)\end{aligned}$$

■

Lemma 3.22 *Under the conditions of Theorem 3.14, excluding Assumption 3.12,*

$$E\left(\frac{2\pi}{n}\sum_{j=1}^{m}\left|I(\lambda_j) - \Psi(\lambda_{\tilde{j}})I_\epsilon(\lambda_j)\Psi^*(\lambda_{\tilde{j}})\right|\right) \;=\; o\left(\lambda_m^{1-d_{aa}-d_{bb}}\right)$$

If we further assume that Assumption 3.12 holds and $a \neq b$,

$$\left|E\left(\frac{2\pi}{n}\sum_{j=1}^{m}\left(I_{ab}(\lambda_j) - \Psi_a(\lambda_{\tilde{j}})I_\epsilon(\lambda_j)\Psi_b^*(\lambda_{\tilde{j}})\right)\right)\right| \;=\; o\left(\lambda_m^{1-2d_{ab}}\right)$$

Proof. A proof very similar to that of Lemma 18 of Chen and Hurvich [2006] shows that:

$$E\left(\left|I_{ab}(\lambda_j) - \Psi_a(\lambda_{\tilde{j}})I_\epsilon(\lambda_j)\Psi_b^*(\lambda_{\tilde{j}})\right|\right) \;\leq\; C\lambda_j^{-d_{aa}-d_{bb}}j^{-\gamma/2}$$

(Their Assumption 2 is similar to our Assumption 3.10, but uses $\Psi_{jk}^\dagger(\lambda)$ instead of $\tau_{jk}(\lambda)e^{i\varphi_{jk}(\lambda)}$. Because the results are proved for each (j,k) separately, we can allow δ_{jk} to vary with j,k when we apply the univariate results from Hurvich et al. [2002] in the proof of that lemma.)

Then, with C being an arbitrary non-zero constant that may change from one line to the next, we find the expected modulus of the sum in (3.32):

$$E\left(\frac{2\pi}{n}\sum_{j=1}^{m}\left|I_{ab}(\lambda_j) - \Psi_a(\lambda_{\tilde{j}})I_\epsilon(\lambda_j)\Psi_b^*(\lambda_{\tilde{j}})\right|\right) \tag{3.40}$$

$$\leq\; \frac{C}{n}\sum_{j=1}^{m}\lambda_j^{-d_{aa}-d_{bb}}j^{-\gamma/2} \tag{3.41}$$

$$=\; Cn^{-1+d_{aa}+d_{bb}}\sum_{j=1}^{m}j^{-d_{aa}-d_{bb}-\gamma/2} \tag{3.42}$$

Based on the value of $-d_{aa}-d_{bb}-\gamma/2$, we have three cases:

Case 1: $-d_{aa}-d_{bb}-\gamma/2 < -1$. In this case, $\sum_{j=1}^{m}j^{-d_{aa}-d_{bb}-\gamma/2} = O(1)$. Because $d_{aa}+d_{bb}<1$,

$$\begin{aligned} n^{-1+d_{aa}+d_{bb}}\sum_{j=1}^{m}j^{-d_{aa}-d_{bb}-\gamma/2} &= O\left(n^{-1+d_{aa}+d_{bb}}\right) \\ &= O\left(\lambda_m^{1-d_{aa}-d_{bb}}m^{d_{aa}+d_{bb}-1}\right) \\ &= O\left(\lambda_m^{1-2d_{ab}}n^{-2d_\rho}m^{2d_{ab}-1}\right) \end{aligned}$$

Since $d_{aa}+d_{bb}<1$ by stationarity, Equation (3.43) is $o\left(\lambda_m^{1-d_{aa}-d_{bb}}\right)$. If Assumption 3.12 holds, then Equation (3.43) is $o\left(\lambda_m^{1-2d_{ab}}\right)$.

Case 2: $-d_{aa}-d_{bb}-\gamma/2 = -1$. Then, $\sum_{j=1}^{m}j^{-d_{aa}-d_{bb}-\gamma/2} = \sum_{j=1}^{m}\frac{1}{j} =$

$O(\log(m))$, and we have:

$$n^{-1+d_{aa}+d_{bb}} \sum_{j=1}^{m} j^{-d_{aa}-d_{bb}-\gamma/2} = O\left(n^{-1+d_{aa}+d_{bb}} \log(m)\right) \quad (3.43)$$

$$= O\left(\lambda_m^{1-d_{aa}-d_{bb}} m^{d_{aa}+d_{bb}-1} \log(m)\right) \quad (3.44)$$

$$= O\left(\lambda_m^{1-2d_{ab}} n^{-2d_\rho} m^{2d_{ab}-1} \log(m)\right) \quad (3.45)$$

As before, $d_{aa} + d_{bb} < 1$ by stationarity, so $m^{d_{aa}+d_{bb}-1} \log(m) = o(1)$ and Equation (3.44) is $o\left(\lambda_m^{1-d_{aa}-d_{bb}}\right)$. If Assumption 3.12 holds, then Equation (3.45) is $o\left(\lambda_m^{1-2d_{ab}}\right)$.

Case 3: $-d_{aa} - d_{bb} - \gamma/2 > -1$. In this case, we rewrite:

$$Cn^{-1+d_{aa}+d_{bb}} \sum_{j=1}^{m} j^{-d_{aa}-d_{bb}-\gamma/2} = O\left(\frac{n^{-\gamma/2}}{n} \sum_{j=1}^{m} \lambda_j^{-d_{aa}-d_{bb}-\gamma/2}\right) \quad (3.46)$$

$$= O\left(n^{-\gamma/2} \lambda_m^{1-d_{aa}-d_{bb}-\gamma/2}\right) \quad (3.47)$$

$$= O\left(\frac{1}{m^{\gamma/2}} \lambda_m^{1-d_{aa}-d_{bb}}\right) \quad (3.48)$$

$$= O\left(\frac{\lambda_m^{2d_\rho}}{m^{\gamma/2}} \lambda_m^{1-2d_{ab}}\right) \quad (3.49)$$

Since $m \to \infty$, (3.48) is $o(\lambda_m^{1-d_{aa}-d_{bb}})$. If $\frac{n^{\frac{2d_\rho}{2d_\rho-\gamma/2}}}{m} \to 0$, then $m^{-\gamma/2}\lambda_m^{2d_\rho} = o(1)$ and (3.49) is $o(\lambda_m^{1-2d_{ab}})$.

■

The next lemma is closely related to Lemma 19 of Chen and Hurvich [2006], generalizing the result to the case where ϵ_t is non-Gaussian. In certain cases, the bound on $|E\left(S_{ab}(\lambda_j)S_{ab}(\lambda_k)\right)|$ could replace $-d_{aa} - d_{bb}$ by $-2d_{ab}$; however, this will happen only for values of the fourth cumulants and $E\left(J_{\epsilon,u_1}(\lambda_j)\overline{J_{\epsilon,v_2}(\lambda_j)}\right)$ $\times E\left(J_{\epsilon,u_2}(\lambda_k)\overline{J_{\epsilon,v_1}(\lambda_k)}\right)$ that preserve the power law coherency properties.

Lemma 3.23 *Let $I_{\epsilon,u,v}(\lambda_j)$ be the (u,v) element of the cross-periodogram of p-variate white noise. Let $\lambda_{\tilde{j}}$ be the shifted Fourier frequency. Let $1 \le j,k \le n/2$.*

Define

$$S(\lambda_j) \;=\; \Psi(\lambda_{\tilde{j}})I_\epsilon(\lambda_j)\Psi^*(\lambda_{\tilde{j}}) - f(\lambda_j)$$

Let $S_{ab}(\lambda_j)$ *be the* (a,b) *element of* $S(\lambda_j)$*. Then,*

$$E\left|S_{ab}(\lambda_j)S_{ab}(\lambda_k)\right| \;\leq\; \begin{cases} C|\lambda_{\tilde{j}}\lambda_{\tilde{k}}|^{-(d_{aa}+d_{bb})} + O\left(\frac{1}{n}|\lambda_{\tilde{j}}\lambda_{\tilde{k}}|^{-(d_{aa}+d_{bb})}\right) & |j-k| \leq s \\ O\left(\frac{1}{n}|\lambda_{\tilde{j}}\lambda_{\tilde{k}}|^{-(d_{aa}+d_{bb})}\right) & |j-k| > s \end{cases}$$

Proof. Following Chen and Hurvich [2006], we write:

$$\begin{aligned} & E\left(S_{ab}(\lambda_j)S_{ab}(\lambda_k)\right) && (3.50) \\ = & \sum_{u_1=1}^{p}\sum_{u_2=1}^{p}\sum_{v_1=1}^{p}\sum_{v_2=1}^{p} \Psi_{au_1}(\lambda_{\tilde{j}})\overline{\Psi_{au_2}(\lambda_{\tilde{k}})}\,\overline{\Psi_{bv_1}(\lambda_{\tilde{j}})}\Psi_{bv_2}(\lambda_{\tilde{k}}) && (3.51) \\ & \times E\left(\left(I_{\epsilon,u_1v_1}(\lambda_j) - \sigma_{u_1v_1}\right)\left(I_{\epsilon,u_2v_2}(\lambda_k) - \sigma_{u_2v_2}\right)\right) && (3.52) \end{aligned}$$

with

$$\begin{aligned} & E\left(\left(I_{\epsilon,u_1v_1}(\lambda_j) - \sigma_{u_1v_1}\right)\left(I_{\epsilon,u_2v_2}(\lambda_k) - \sigma_{u_2v_2}\right)\right) \\ = \; & cum\left(J_{\epsilon,u_1}(\lambda_j), J_{\epsilon,u_2}(\lambda_k), \overline{J_{\epsilon,v_1}(\lambda_j)}, \overline{J_{\epsilon,v_2}(\lambda_k)}\right) \\ & + E\left(J_{\epsilon,u_1}(\lambda_j)\overline{J_{\epsilon,v_2}(\lambda_j)}\right) E\left(J_{\epsilon,u_2}(\lambda_k)\overline{J_{\epsilon,v_1}(\lambda_k)}\right) \end{aligned}$$

and

$$E\left(J_{\epsilon,u_1}(\lambda_j)\overline{J_{\epsilon,v_2}(\lambda_j)}\right) E\left(J_{\epsilon,u_2}(\lambda_k)\overline{J_{\epsilon,v_1}(\lambda_k)}\right) \;=\; C\chi(|j-k| \leq s)$$

Next, we compute the cumulant:

$$\begin{aligned}
& cum\left(J_{\epsilon,u_1}(\lambda_j), J_{\epsilon,u_2}(\lambda_j), \overline{J_{\epsilon,v_1}(\lambda_j)}, \overline{J_{\epsilon,v_2}(\lambda_j)}\right) \\
= \; & cum\left(\frac{1}{\sqrt{2\pi n}}\sum_{t=1}^{n}\epsilon_{u_1,t}e^{it\lambda_j}, \frac{1}{\sqrt{2\pi n}}\sum_{t=1}^{n}\epsilon_{u_2,t}e^{it\lambda_k}, \frac{1}{\sqrt{2\pi n}}\sum_{t=1}^{n}\epsilon_{v_1,t}e^{-it\lambda_j}, \right. \\
& \left. \frac{1}{\sqrt{2\pi n}}\sum_{t=1}^{n}\epsilon_{v_2,t}e^{-it\lambda_k}\right) \\
= \; & \frac{1}{(2\pi n)^2}cum\left(\sum_{t=1}^{n}\epsilon_{u_1,t}e^{it\lambda_j}, \sum_{t=1}^{n}\epsilon_{u_2,t}e^{it\lambda_k}, \sum_{t=1}^{n}\epsilon_{v_1,t}e^{-it\lambda_j}, \sum_{t=1}^{n}\epsilon_{v_2,t}e^{-it\lambda_k}\right) \\
= \; & \frac{1}{(2\pi n)^2}\sum_{t=1}^{n}cum(\epsilon_{u_1,t}e^{it\lambda_j}, \epsilon_{u_2,t}e^{it\lambda_k}, \epsilon_{v_1,t}e^{-it\lambda_j}, \epsilon_{v_2,t}e^{-it\lambda_k}) \\
= \; & \frac{1}{(2\pi n)^2}\sum_{t=1}^{n}cum(\epsilon_{u_1,t}, \epsilon_{u_2,t}, \epsilon_{v_1,t}, \epsilon_{v_2,t}) \\
= \; & \frac{1}{(2\pi)^2 n}cum(\epsilon_{u_1,1}, \epsilon_{u_2,1}, \epsilon_{v_1,1}, \epsilon_{v_2,1})
\end{aligned}$$

which is $O\left(\frac{1}{n}\right)$ by Assumption 3.1.

Substituting these results into Equation (3.52), we find that:

$$\begin{aligned}
& E|S_{ab}(\lambda_j)S_{ab}(\lambda_k)| \\
= \; & O\left(\left(\chi(|j-k|\leq s)+\frac{1}{n}\right)\sum_{u_1=1}^{p}\sum_{u_2=1}^{p}\sum_{v_1=1}^{p}\sum_{v_2=1}^{p}\left|\Psi_{au_1}(\lambda_{\tilde{j}})\overline{\Psi}_{au_2}(\lambda_{\tilde{k}})\overline{\Psi}_{bv_1}(\lambda_{\tilde{j}})\Psi_{bv_2}(\lambda_{\tilde{k}})\right|\right) \\
= \; & O\left(\left(\chi(|j-k|\leq s)+\frac{1}{n}\right)|\lambda_{\tilde{j}}\lambda_{\tilde{k}}|^{-(d_{aa}+d_{bb})}\right)
\end{aligned}$$

■

Lemma 3.24 *Let $S(\lambda_j) = \Psi(\lambda_{\tilde{j}})I_\epsilon(\lambda_j)\Psi^*(\lambda_{\tilde{j}}) - f(\lambda_j)$. Under the conditions of Theorem 3.14, excluding Assumption 3.12,*

$$E\left(\frac{4\pi^2}{n^2}\sum_{j=1}^{m}\sum_{k=1}^{m}S_{ab}(\lambda_j)\overline{S_{ab}(\lambda_k)}\right) \;=\; o\left(\lambda_m^{2-2d_{aa}-2d_{bb}}\right)$$

If $a \neq b$ and Assumption 3.12 holds,

$$E\left(\frac{4\pi^2}{n^2}\sum_{j=1}^{m}\sum_{k=1}^{m}S_{ab}(\lambda_j)\overline{S_{ab}(\lambda_k)}\right) \;=\; o\left(\lambda_m^{2-4d_{ab}}\right)$$

Proof. Applying Lemma 3.23, the expected square of the sum in 3.33:

$$E\left(\frac{4\pi^2}{n^2}\sum_{j=1}^{m}\sum_{k=1}^{m}S_{ab}(\lambda_j)\overline{S_{ab}(\lambda_k)}\right) \tag{3.53}$$

$$\leq \frac{4\pi^2}{n^2}\sum_{j=1}^{m}\sum_{k=1}^{m}E\left(S_{ab}(\lambda_j)\overline{S_{ab}(\lambda_k)}\right) \tag{3.54}$$

$$= O\Bigg(\frac{1}{n^3}\sum_{j=1}^{m}\sum_{k=1}^{m}\lambda_{\tilde{j}}^{-d_{aa}-d_{bb}}\lambda_{\tilde{k}}^{-d_{aa}-d_{bb}} \tag{3.55}$$

$$+\frac{1}{n^2}\sum_{j=1}^{m}\sum_{k=1}^{m}|\lambda_{\tilde{j}}\lambda_{\tilde{k}}|^{-d_{aa}-d_{bb}}\chi(|j-k|\leq s)\Bigg) \tag{3.56}$$

Because $d_{aa}+d_{bb}<1$:

$$\begin{aligned}\frac{1}{n^3}\sum_{j=1}^{m}\sum_{k=1}^{m}\lambda_{\tilde{j}}^{-d_{aa}-d_{bb}}\lambda_{\tilde{k}}^{-d_{aa}-d_{bb}} &= \frac{1}{n}\left(\frac{1}{n}\sum_{j=1}^{m}\lambda_{\tilde{j}}^{-d_{aa}-d_{bb}}\right)^2\\ &= O\left(\frac{1}{n}\lambda_m^{2-2d_{aa}-d_{bb}}\right)\end{aligned}$$

We rewrite Equation (3.56) as:

$$\begin{aligned}&\frac{1}{n^2}\sum_{j=1}^{m}\sum_{k=1}^{m}|\lambda_{\tilde{j}}\lambda_{\tilde{k}}|^{-d_{aa}-d_{bb}}\chi(|j-k|\leq s)\\ =\ & O\left(\frac{1}{n^2}\sum_{j=1}^{m}\lambda_{\tilde{j}}^{-2d_{aa}-2d_{bb}}\right)\\ =\ & O\left(n^{-2+d_{aa}+d_{bb}}\sum_{j=1}^{m}j^{-2d_{aa}-2d_{bb}}\right)\end{aligned}$$

As in Lemma 3.22, there are three cases, now based on the value of $-2d_{aa}-2d_{bb}$.

Case 1: $-2d_{aa}-2d_{bb}<-1$. In this case, $\sum_{j=1}^{m}j^{-2d_{aa}-2d_{bb}}=O(1)$ as $m\to\infty$. Thus,

$$n^{-2+d_{aa}+d_{bb}}\sum_{j=1}^{m}j^{-2d_{aa}-2d_{bb}} = O\left(n^{-2+d_{aa}+d_{bb}}\right) \tag{3.57}$$

$$= O\left(\lambda_m^{2-2d_{aa}-2d_{bb}}m^{2d_{aa}+2d_{bb}-2}\right) \tag{3.58}$$

$$= O\left(\lambda_m^{2-4d_{ab}}n^{-4d_\rho}m^{4d_{ab}-2}\right) \tag{3.59}$$

Under Assumption 3.11, the expression in Equation (3.58) is $o(\lambda_m^{2-2d_{aa}-2d_{bb}})$ because $d_{aa} + d_{bb} < 1$ and $m \to \infty$. Under Assumption 3.12, the expression in Equation (3.59) is $o(\lambda_m^{2-2d_{ab}})$.

Case 2: $-2d_{aa} - 2d_{bb} = -1$. In this case, $\sum_{j=1}^{m} j^{-2d_{aa}-2d_{bb}} = O(\log(m))$ as $m \to \infty$. Then,

$$n^{-2+d_{aa}+d_{bb}} \sum_{j=1}^{m} j^{-2d_{aa}-2d_{bb}} = O\left(n^{-2+d_{aa}+d_{bb}} \log(m)\right) \quad (3.60)$$

$$= O\left(\lambda_m^{2-2d_{aa}-2d_{bb}} m^{2d_{aa}+2d_{bb}-2} \log(m)\right) \quad (3.61)$$

$$= O\left(\lambda_m^{2-4d_{ab}} n^{-4d_\rho} m^{4d_{ab}-2} \log(m)\right) \quad (3.62)$$

Under Assumption 3.11, the expression in Equation (3.61) is $o(\lambda_m^{2-2d_{aa}-2d_{bb}})$ because $d_{aa} + d_{bb} < 1$ and $m \to \infty$. Under Assumption 3.12, the expression in Equation (3.62) is $o(\lambda_m^{2-2d_{ab}})$.

Case 3: $-2d_{aa} - 2d_{bb} > -1$. Then, we have

$$n^{-2+d_{aa}+d_{bb}} \sum_{j=1} j^{-2d_{aa}-2d_{bb}} = O\left(\frac{1}{n}\lambda_m^{2-2d_{aa}-2d_{bb}} + \lambda_m^{1-2d_{aa}-2d_{bb}}\right) \quad (3.63)$$

$$= O\left(\frac{1}{m}\lambda_m^{2-2d_{aa}-2d_{bb}}\right) \quad (3.64)$$

$$= O\left(\frac{\lambda_m^{2d_\rho}}{m}\lambda_m^{2-2d_{ab}}\right) \quad (3.65)$$

Since $m \to \infty$, (3.64) is $o(\lambda_m^{2(1-d_{aa}-d_{bb})})$. If $\frac{n^{\frac{2d_\rho}{2d_\rho-1}}}{m} \to 0$, then $\frac{\lambda_m^{2d_\rho}}{m} = o(1)$ and (3.65) is $o\left(\lambda_m^{2(1-2d_{ab})}\right)$.

■

The following lemma applies whenever $f_{ab}(\lambda)$ is a regularly-varying complex function with $d_{ab} < 1/2$; it does not require that $L(1/\lambda)$ has a limit as $\lambda \to 0^+$ or that it is in $\mathcal{L}^*$.

Lemma 3.25 *Let $f_{ab}(\lambda)$ be regularly varying at 0 with $d_{ab} < 1/2$.*

$$\frac{2\pi}{n}\sum_{j=1}^{m} f_{ab}(\lambda_j) - F_{ab}(\lambda_m) \quad = \quad o(F_{ab}(\lambda_m))$$

Proof. Robinson [1994, Proposition 1, page 525] has already proven this result in the case that $0 \leq d_{ab} < 1/2$.

Following the proof of Robinson [1994, Proposition 1, page 525], for large enough n,

$$\begin{aligned}\frac{2\pi}{n}\sum_{j=1}^{m} f_{ab}(\lambda_j) - F_{ab}(\lambda_m) \quad &\leq \quad \left|\sum_{j=1}^{m}\int_{\lambda_{j-1}}^{\lambda_j}\left(L\left(\frac{1}{\lambda_j}\right)\lambda_j^{-2d_{ab}} - L\left(\frac{1}{\lambda}\right)\lambda^{-2d_{ab}}\right)d\lambda\right| \\ &\quad + o\left(\frac{1}{n}\sum_{j=1}^{m} L\left(\frac{1}{\lambda_j}\right)\lambda_j^{-2d_{ab}} + \int_0^{\lambda_m} L\left(\frac{1}{\lambda}\right)\lambda^{-2d_{ab}}\right)\end{aligned}$$

We decompose the first term into two parts:

$$\begin{aligned}&\left|\sum_{j=1}^{m}\int_{\lambda_{j-1}}^{\lambda_j}\left(L\left(\frac{1}{\lambda_j}\right)\lambda_j^{-2d_{ab}} - L\left(\frac{1}{\lambda}\right)\lambda^{-2d_{ab}}\right)d\lambda\right| \\ \leq \quad &\sum_{j=1}^{m}\left|L\left(\frac{1}{\lambda_j}\right)\int_{\lambda_{j-1}}^{\lambda_j}\left(\lambda_j^{-2d_{ab}} - \lambda^{-2d_{ab}}\right)d\lambda\right| \\ &+\sum_{j=1}^{m}\left|\int_{\lambda_{j-1}}^{\lambda_j}\lambda^{-2d_{ab}}\left(L\left(\frac{1}{\lambda_j}\right) - L\left(\frac{1}{\lambda}\right)\right)d\lambda\right|\end{aligned}$$

For the first term, we use a Taylor series expansion to find:

$$\begin{aligned}\int_{\lambda_{j-1}}^{\lambda_j}\left(\lambda_j^{-2d_{ab}} - \lambda^{-2d_{ab}}\right)d\lambda \quad &= \quad \left(\frac{2\pi}{n}\right)^{1-2d_{ab}} j^{-2d_{ab}} - \frac{1}{1-2d}\left(\lambda_j^{1-2d_{ab}} - \lambda_{j-1}^{1-2d_{ab}}\right) \\ &= \quad \left(\frac{2\pi}{n}\right)^{1-2d_{ab}}\left[j^{-2d_{ab}} - \frac{1}{1-2d_{ab}}\big(j^{1-2d_{ab}} - j^{1-2d_{ab}}\right. \\ &\quad \left. +(1-2d_{ab})j^{-2d_{ab}} + O(j^{-1-2d_{ab}})\big)\right] \\ &= \quad O\left(\frac{1}{j^2}\lambda_j^{1-2d_{ab}}\right)\end{aligned}$$

Then,

$$\left|L\left(\frac{1}{\lambda_j}\right)\int_{\lambda_{j-1}}^{\lambda_j}\left(\lambda_j^{-2d_{ab}} - \lambda^{-2d_{ab}}\right)d\lambda\right| \quad = \quad O\left(\frac{1}{j^2}L\left(\frac{1}{\lambda_j}\right)\lambda_j^{1-2d_{ab}}\right)$$

Summing over j,

$$\sum_{j=1}^{m}\left|L\left(\frac{1}{\lambda_j}\right)\int_{\lambda_{j-1}}^{\lambda_j}\left(\lambda_j^{-2d_{ab}}-\lambda^{-2d_{ab}}\right)d\lambda\right| \;=\; O\left(\frac{1}{m}L\left(\frac{1}{\lambda_m}\right)\lambda_m^{1-2d_{ab}}\right)$$

For the second term, Seneta [1970, page 7] notes that we can choose the slowly varying function, $L(z)$, to be differentiable with $\frac{zL'(z)}{L(z)}\to 0$. Using integration by parts, we may write:

$$\begin{aligned}\int_{\lambda_{j-1}}^{\lambda_j}\lambda^{-2d_{ab}}L\left(\frac{1}{\lambda}\right)d\lambda &= \frac{1}{1-2d_{ab}}\left(L\left(\frac{1}{\lambda_j}\right)\lambda_j^{1-2d_{ab}}-L\left(\frac{1}{\lambda_{j-1}}\right)\lambda_{j-1}^{1-2d_{ab}}\right)\\ &\quad+\frac{1}{1-2d_{ab}}\int_{\lambda_{j-1}}^{\lambda_j}\left(\frac{1}{\lambda}L'\left(\frac{1}{\lambda}\right)\right)\lambda^{-2d_{ab}}d\lambda\end{aligned}$$

The last term is of a smaller order as $n\to\infty$ because $\frac{\frac{1}{\lambda}L'(\frac{1}{\lambda})}{L(\frac{1}{\lambda})}\to 0$. Thus,

$$\begin{aligned}\int_{\lambda_{j-1}}^{\lambda_j}\lambda^{-2d_{ab}}\left(L\left(\frac{1}{\lambda_j}\right)-L\left(\frac{1}{\lambda}\right)\right)d\lambda &= \frac{L\left(\frac{1}{\lambda_j}\right)}{1-2d_{ab}}\left(\lambda_j^{1-2d_{ab}}-\lambda_{j-1}^{1-2d_{ab}}\right)\\ &\quad-\frac{1}{1-2d_{ab}}\left(L\left(\frac{1}{\lambda_j}\right)\lambda_j^{1-2d_{ab}}\right.\\ &\quad\left.-L\left(\frac{1}{\lambda_{j-1}}\right)\lambda_{j-1}^{1-2d_{ab}}\right)+o\left(\lambda_j^{1-2d_{ab}}\right)\\ &= \frac{\lambda_{j-1}^{1-2d_{ab}}}{1-2d_{ab}}\left(L\left(\frac{1}{\lambda_j}\right)-L\left(\frac{1}{\lambda_{j-1}}\right)\right)\\ &\quad+o\left(\lambda_j^{1-2d_{ab}}\right)\\ &= o\left(\lambda_j^{1-2d_{ab}}\right)\end{aligned}$$

since $L\left(\frac{1}{\lambda_j}\right)-L\left(\frac{1}{\lambda_{j-1}}\right)=o(1)$. Thus, summing over j,

$$\sum_{j=1}^{m}\left|\int_{\lambda_{j-1}}^{\lambda_j}\lambda^{-2d_{ab}}\left(L\left(\frac{1}{\lambda_j}\right)-L\left(\frac{1}{\lambda}\right)\right)d\lambda\right| \;=\; o\left(F_{ab}(\lambda_m)\right)$$

■

4 RE-EM Trees: A New Data Mining Approach for Longitudinal Data

4.1 Introduction

Some response data are one dimensional: observations over time or across individuals. However, panel or longitudinal data, in which we observe many individuals over multiple periods, offers a particular opportunity for understanding, as we observe the different paths that a variable might take across individuals. Such opportunities are especially attractive with large amounts of data, as this allows us to fit complex or highly structured models to the data. In this paper, we present a data mining approach that is specialized for longitudinal data. This method combines the flexibility of data mining methods with the specific nature of a longitudinal dataset.

Suppose we observe a panel of individuals $i = 1, ..., I$ at times $t = 1, ..., T_i$. Throughout this paper, we will refer to a member of the panel, i, as an individual, and a single observation period for an individual, (i, t), as an observation. That is, one individual is associated with multiple observations. The covariates may be constant over time, constant across individuals, or varying across time and individuals. For each observation, we observe a vector of covariates, $\mathbf{x}_{it} = (x_{it1}, ..., x_{itK})'$, and a response, y_{it}. Our model also includes a known design matrix, Z_{it}, which may vary each period and depend on the covariates, and a vector of unknown time-constant, individual-specific effects, $\mathbf{b}_i$. In the case where only the intercept varies across individuals, Z_i is a matrix of ones and b_i is the individual-specific intercept. The inclusion of a general design matrix allows for the individual effects to be more complicated, since the effects may depend on the observed characteristics of

individuals. Consider a general effects model with additive errors:

$$y_{it} = Z_{it}b_i + f(x_{it1}, ..., x_{itK}) + \varepsilon_{it} \tag{4.1}$$

$$\begin{pmatrix} \varepsilon_{i1} \\ \vdots \\ \varepsilon_{iT_i} \end{pmatrix} \sim Normal(0, R_i) \tag{4.2}$$

Throughout this paper, we assume that the errors, ε_{it}, are independent across individuals. Depending on our assumptions about $\mathbf{b}_i$ and f, the general model may reduce to different well-known models. Consider the case where $Z_{it} = 1$ so that b_i is an individual-specific intercept. If f is a known function that is linear in the parameters and the b_i are taken as fixed or potentially correlated with the predictors, then this is a fixed effects model. Under the same assumptions on f, if we instead assume that b_i are random variables that are uncorrelated with the predictors, then the model is a random effects model. If Z_i includes one or more covariates and $\mathbf{b}_i$ is again taken to be uncorrelated with the predictors, this becomes a random parameters model.

Random effects models, when appropriate, are more efficient than fixed effects models, because the number of parameters estimated in a fixed effects model increases with the addition of more individuals. This is especially important when T_i is small relative to I. Furthermore, fixed effects models with individual-specific intercepts (by far the most common kind) do not allow the inclusion of predictors that are always constant for individuals, such as gender. Finally, because the distribution of the fixed effects, b_i, is not estimated, we have no basis for estimating the individual-specific effects in predictions for individuals not in the sample. Wooldridge [2002, Section 10.2.1] discussed the differences between these two models in more detail.

Fixed and random effects models typically assume a parametric form for f,

which might be too restrictive an assumption. The functional form of f is frequently unknown, and assuming a linear model may not be the best option. Furthermore, K may be very large, so that including all of the predictors directly may lead to overfitting and therefore poor predictions. A variety of nonparametric and data mining methods exist to estimate f in the case when b_i is constant across individuals, including ridge regression, splines, and myriad others. We focus on regression trees, as described by Breiman et al. [1984]. One could fit a regression tree to a longitudinal data set, ignoring the longitudinal data structure and assuming that $\mathbf{b}_i = \mathbf{0}$ for all i. However, just as fitting a linear model without fixed or random effects to a longitudinal dataset can lead to incorrect estimates and inference, applying a nonparametric method designed for cross-sectional data directly to longitudinal data may be misleading. Instead, we propose a method that accounts for the additional longitudinal structure in the data.

We continue in section 4.2 with a review of random effects models, regression trees, and the existing literature on data mining methods for longitudinal data. In section 4.3, we present our model and estimation method. In sections 4.4 and 4.5, we apply this method to datasets on traffic fatality rates and on Amazon third party transactions, respectively. In section 4.6, we use Monte Carlo simulations to explore the efficacy of the method. Section 4.7 concludes with a discussion of potential future work.

4.2 Previous Work

4.2.1 Random Effects Models

The parametric random effects model is given by:

$$\begin{aligned}
\begin{pmatrix} y_{i1} \\ \vdots \\ y_{iT} \end{pmatrix} &= Z_i \mathbf{b}_i + \begin{pmatrix} f(x_{i11}, ..., x_{i1K}) \\ \vdots \\ f(x_{iT1}, ..., x_{iTK}) \end{pmatrix} + \begin{pmatrix} \varepsilon_{i1} \\ \vdots \\ \varepsilon_{it_i} \end{pmatrix} \\
\begin{pmatrix} \varepsilon_{i1} \\ \vdots \\ \varepsilon_{it_i} \end{pmatrix} &\sim Normal(0, R_i) \\
\mathbf{b}_i &\sim Normal(0, D)
\end{aligned}$$

where $f(x_{it1}, ..., x_{itK}) = \beta_1 x_{it1} + ... + \beta_K x_{itK}$ is a parametric linear function. This model assumes that the random effects and the errors are independent of each other and of the covariates.

The parameters, $\beta = (\beta_1, ..., \beta_K)$, can be estimated using standard regression techniques. These techniques treat both the individual-specific random effects and the observation-specific errors as part of the errors term of the regression. A linear regression of y_{it} on $x_{it\cdot}$ will yield consistent and unbiased estimates of the parameters of f, since we assume that neither the effects nor the errors are correlated with the covariance. However, the estimated β from linear regression will not have the minimum variance because of the covariance structure in $Z_i\mathbf{b}_i+\varepsilon_i$. Using generalized least squares accounts for the correlation in the error terms. For more information on the GLS approach, see Wooldridge [2002, section 10.4].

The two-stage approach to fitting random effects models, described by Harville [1977a], yields estimates of the random effects, $\mathbf{b}_i$, instead of including them in the error terms. While maximum likelihood estimation can be applied directly to the

two-stage approach, we focus instead on the EM algorithm for two-stage random effects models given by Laird and Ware [1982]. To apply the EM algorithm, they note that, given estimates for the parameters defining R_i and D, one can estimate the $\mathbf{b}_i$ and therefore expected values of sufficient statistics for R_i and D. At the same time, one can estimate R_i and D given their sufficient statistics. This suggests a specialization of the EM algorithm to random effects estimation [Laird and Ware, 1982]:

1. Initialize the random effects, $\hat{\mathbf{b}}_i$, to zero and the covariance matrices, $\hat{D}, \hat{R}_i$, to identity matrices of the correct sizes.

2. Iterate through the following steps until the estimated random effects, $\hat{\mathbf{b}}_i$, converge:

   (a) Estimate a linear regression to fit β, based on the target variable, $y_{it} - Z_{it}\hat{\mathbf{b}}_i$, and predictors, $\mathbf{x}_{it\cdot} = (x_{it1}, ..., x_{itK})$, for $i = 1, ..., I$ and $t = 1, ..., T_i$.

   (b) Estimate the new random effects, $\hat{\mathbf{b}}_i$, and errors, $\hat{\varepsilon}_i$, given the covariance matrices, $\hat{D}, \hat{R}_i$, and β.

   (c) Estimate the covariance matrices, $\hat{D}, \hat{R}_i$, using the new estimates of the random effects and errors.

The resulting estimated random effects are the empirical Bayes estimates.

This estimation method for a random effects linear model can be modified in a number of different ways. Laird and Ware [1982, section 5] describe how the EM algorithm can be used for restricted maximum likelihood (REML) estimation as well as maximum likelihood estimation of the random effects model. REML accounts for the degrees of freedom lost in estimating I random effects by using maximum

likelihood on linear combinations of the original data, called error contrasts, that are chosen to be linearly independent. This leads to an unbiased estimate of the variance of the random effects, which is generally preferable. [See Patterson and Thompson, 1971, Harville, 1977a, Laird and Ware, 1982, for more information.]

A generalization of the linear random effects model allows for autocorrelation in the error terms, with the most common structure an autoregressive model of order one. However, Verbeke and Molenberghs [2000, pages 28-29] note that estimating the autocorrelation parameter can be challenging for linear random effects models, even for simple autocorrelation models. Despite this limitation, models including autocorrelation may be helpful in some cases. To test for autocorrelation in a linear random effects model, we can use a likelihood ratio test. Let l_0 be the log likelihood of a linear random effects model with no autocorrelation and l_{AR} be the log likelihood of a linear random effects model with autocorrelation modeled using p parameters. Then, $-2(l_0 - l_{AR}) \sim \chi^2_p$, as in any likelihood ratio test. We can also test whether including autocorrelation in a linear random effects model is useful by comparing the predictive power of the model with and without autocorrelation. Score tests for autocorrelation and heteroskedasticity for linear models with random effects have been suggested by Chi and Reinsel [1989], Verbeke and Molenberghs [2003], and Lin and Wei, among others.

Afshartous and de Leeuw [2005] discuss prediction for linear random effects models. They describe three different models: an "OLS model" in which a model is fitted separately for each individual, a "prior model" which is a Bayesian version of fitting a linear model that ignores the random effects, and a "multilevel model" that is a Bayesian random effects model. In simulations, they find that the multilevel model performs best, and that the performance of the OLS model approaches that of the multilevel model as the number of observations per individual increases from

5 to 100. They find that the prior model performs quite badly.

4.2.2 The Regression Tree Framework

Regression trees were originally popularized by Breiman et al. [1984], though they dated the use of tree structures in regression to the Automatic Interaction Detection Program of Morgan and Sonquist [1963]. We use the implementation of regression trees in the `rpart` package [Therneau and Atkinson, 2006] of the statistical software package `R` [R Development Core Team, 2008]. A regression tree is a binary tree, where each non-terminal node is split into two nodes based on the values of a single predictor. To find the predicted value for a response, one finds the correct terminal node, g, based on the predictors and then takes the mean of all the response values in that node, $\mu(g)$. To construct such a tree, we measure the "impurity" of responses at a node, g, by:

$$SS(g) = \sum_{i \in g} (y_i - \mu(g))^2$$

Given any node, g, and any possible split, s, of the node into two daughter nodes, g_L and g_R, we define the split function as

$$\phi(s, g) = SS(g) - SS(g_L) - SS(g_R)$$

At each step, the split is chosen to maximize $SS(s, g)$ over all possible splits at all existing terminal nodes.

This splitting continues until all of the elements of each node have the same response value or the number of observations in each node reaches a given minimum value. The resulting tree is overly complex. The next step is to "prune" the tree by removing some of the branches. To quantify the desired amount of pruning,

Breiman et al. [1984] defined the error complexity of a tree as:

$$R_\alpha(T) = \sum_{g \in |\tilde{T}|} SS(g) + \alpha |\tilde{T}|$$

where $\tilde{T}$ is the set of terminal nodes. This expression combines a measure of the in-sample accuracy of the tree with a penalty for the number of nodes. The value of the complexity parameter, α, helps to determine the size of the resulting tree by weighting the penalty for the size of the tree. Varying the complexity parameter leads to a sequence of nested trees, ranging from a tree that has only a root to the unpruned, overly complex tree. The complexity parameter could be chosen by cross-validation; for simplicity, we use the default value given in the `rpart` package.

Unlike linear models, regression trees are capable of handling missing predictor values using surrogate split. When predictors are missing, the best split, s^*, is found based on all cases where the predictor is not missing. A second "surrogate" split, $\tilde{s}$, can also be chosen such that the probability that s^* sends an observation to the same node as $\tilde{s}$ is maximized. This surrogate split can be used for observations where a particular predictor value is missing. For more information about the performance of surrogate split and other missing data methods in the case of classification trees, see Ding and Simonoff [2010].

4.2.3 Previous applications of trees to longitudinal data

Segal [1992] and De'Ath [2002], apparently independently, proposed the first application of regression trees to longitudinal data, in the case where $T_i = T$ for all i. Both created trees in which the response variable was the vector $\mathbf{y}_i = (y_{i1}, ..., y_{iT})$. At each node, a vector of means, $\mu(g)$, was produced, where $\mu_t(g)$ is the estimated value for y_{it} at node g. Changing the response variable in this way required that the split function be modified. Segal suggested two alternatives. Only the first is

applicable to the model we have presented[4]. At each node, define

$$SS(g) = \sum_{i \in g} (\mathbf{y}_i - \mu(g))' V(\theta, g)^{-1} (\mathbf{y}_i - \mu(g))$$

where θ is a vector of parameters describing the covariance matrix, V, of the observations within a group. Note that $V(\theta, g)$ can be any covariance matrix that depends on a small number of parameters, such as the covariance matrix for $AR(1)$ errors or for the exchangeable (compound symmetry) model that implies constant correlation in the errors. As with traditional regression trees, the split function splits node g into daughter nodes g_L and g_R to maximize $\phi(s, g) = SS(g) - SS(g_L) - SS(g_R)$. Notice that the same V must be used in computing $SS(g_L)$ and $SS(g_R)$ to ensure that $\phi(s, g) \geq 0$, but V can be updated in each node after a split has been chosen. De'Ath [2002] also estimated a mean function at each node. Since his work was based on observations of multiple species in a single location, instead of the same individual across time, he assumed that $V(\theta, g)$ was the identity matrix. His version is available as the `R` package `mvpart` [De'ath, 2006]. Larsen and Speckman [2004] proposed a similar approach in which they estimated V as the sample covariance matrix over the full dataset. Abdolell et al. [2002] discussed the use of trees to find clusters based on a single predictor and a longitudinal outcome variable. Hsiao and Shih [2007] also estimated a vector of means using a tree structure; they use an extension of GUIDE [Loh, 2002].

Notice that the approach of Segal and De'Ath and others depends on a single set of predictors for all of the observation periods. This requires that the values of time-varying predictors observed after the first period cannot be used to predict

[4]Segal's second method was based on identifying heterogeneity in the covariances, and required that the means be fitted through another method. Since the function of interest to us, f, describes the relationship of the covariates to the mean value, fitting the means through another method would not make sense in our model.

any observations. This could lead to a loss of information and therefore poorer predictions. Alternatively, all of the periods of time-varying predictors could be used for predicting every observation; this would likely not make sense in practice, since that would allow for covariate values from future time periods to be used in predicting response values from earlier time periods. Furthermore, these trees cannot be used for the prediction of future periods for the same individuals. That is, if we observe only $y_{i1}, ..., y_{i,T-1}$ for each i, this model will not be able to predict y_{iT}, since the means for period T must be constructed based on observations for that period. These two limitations are quite serious in the applications we will present.

More recent work by Galimberti and Montanari [2002] developed a way to create trees that include both time-varying covariates and a longitudinal data structure. While their underlying model is similar to ours, their implementation was much more complex. They first assumed that the covariances of the errors and the random effects were estimated outside their procedure. They then modified the split function to account for the correlation structure. Because they allowed for time-varying covariates, different observations for the same group could appear in different nodes; this made the split function particularly complicated. Their algorithm is not generally available in software. Furthermore, there is no way to handle observations with missing predictors. Finally, because the group-specific effects are never estimated, one could not predict future observations for individuals already included in the sample. This paper will present an algorithm that accomplishes their goal in a more direct way, while also overcoming these weaknesses.

Other papers have also applied the tools of data mining to longitudinal data. Some followed the approach of Segal [1992], applying his method to other types of responses. Zhang [1998] considered the case of binary response variables; these

are classification trees instead of regression trees. Lee [2005, 2006] and Lee et al. [2005] used generalized estimating equations to fit trees for general types of response variables; their trees were not the traditional regression trees. Instead, they estimated a model using maximum likelihood at each node and then split based on the residuals from estimation. These models also depend on a single set of predictors for all periods and cannot predict future observations for individuals in the sample. Other papers have considered data mining methods other than trees for longitudinal data. Zhang [1997] used adaptive splines to fit longitudinal data models, while Evgeniou et al. [2006] used ridge regression to fit models of consumer heterogeneity. We do not pursue either of these methods further.

4.3 The RE-EM Tree Estimation Method

Consider a version of the model with additive individual effects given above, in which we impose additional structure on the effects:

$$\begin{aligned}
\begin{pmatrix} y_{i1} \\ \vdots \\ y_{iT} \end{pmatrix} &= Z_i \mathbf{b}_i + \begin{pmatrix} f(x_{i11}, ..., x_{i1K}) \\ \vdots \\ f(x_{iT1}, ..., x_{iTK}) \end{pmatrix} + \begin{pmatrix} \varepsilon_{i1} \\ \vdots \\ \varepsilon_{it_i} \end{pmatrix} \\
\varepsilon_i = \begin{pmatrix} \varepsilon_{i1} \\ \vdots \\ \varepsilon_{it_i} \end{pmatrix} &\sim Normal(0, R_i) \\
\mathbf{b}_i &\sim Normal(0, D)
\end{aligned}$$

We propose an estimation method that uses a tree structure to estimate f, but also incorporates individual-specific random effects, $\mathbf{b}_i$. In this method, the nodes may split based on any covariate, so that different observations for the same individual may be placed in different nodes. However, our method ensures that the longitudinal structure in the errors is preserved.

To estimate this model, we must estimate f and D, as well as R_i and $\mathbf{b}_i$ for each i. Our estimation method is based on the ideas of the Expectation-Maximization (EM) algorithm of Laird and Ware [1982], where the M-step is based on using a regression tree instead of traditional parametric maximum likelihood methods. This method is analogous to fitting a parametric random effects using the EM algorithm, and for that reason we call it a Random Effects-Expectation Maximization (RE-EM) Tree.

Algorithm 4.1 Estimation of a RE-EM Tree.

1. *Initialize the random effects, $\hat{\boldsymbol{b}}_i$, to zero and the covariance matrices, $\hat{D}, \hat{R}_i$, to identity matrices of the correct sizes.*

2. *Iterate through the following steps until the estimated random effects, $\hat{\boldsymbol{b}}_i$, converge:*

   (a) *Estimate a regression tree approximating f, based on the target variable, $y_{it} - Z_{it}\hat{\boldsymbol{b}}_i$, and predictors, $\boldsymbol{x}_{it\cdot} = (x_{it1}, ..., x_{itK})$, for $i = 1, ..., I$ and $t = 1, ..., T_i$.*

   (b) *Estimate the new random effects, $\hat{\boldsymbol{b}}_i$, and errors, $\hat{\varepsilon}_i$, given the covariance matrices, $\hat{D}, \hat{R}_i$, and the tree.*

   (c) *Estimate the covariance matrices, $\hat{D}, \hat{R}_i$, using the new estimates of the random effects and errors.*

This method differs from Laird and Ware only in step $2a$; they fit a parametric model rather than a regression tree. Like fixed effects estimation methods, this method estimates the individual effects. Like the random effects model, the individual effects are assumed to be independent of the predictors; the design matrix,

Z_i, can include predictors, which would relate the predictors to the individual-specific intercept in a known way.

The relationship between the covariance matrices and the random effects and errors will depend on the specification of the covariance matrices. Laird and Ware allow D to be a general covariance matrix and assume that $R_i = \sigma^2 I_{n_i \times n_i}$, where $I_{n_i \times n_i}$ is the identity matrix. If the random effects and the errors were observed, the sufficient statistics for R_i and D, respectively, would be:

$$\begin{aligned} t_1 &= \sum_{i=1}^{m} \varepsilon_i^T \varepsilon_i \\ t_2 &= \sum_{i=1}^{m} \mathbf{b}_i \mathbf{b}_i^T \end{aligned}$$

To use the EM algorithm, we compute the expectations of these sufficient statistics, given $\hat{f}$, the predictor values, and the response values from:

$$W_i(\hat{\theta}) = (R_i(\hat{\theta}) + Z_i D(\hat{\theta}) Z_i^T)^{-1} \tag{4.3}$$

$$\hat{\mathbf{b}}_i = D(\hat{\theta}) Z_i^T W_i(\hat{\theta})(\mathbf{y}_i - \hat{f}(x_i)) \tag{4.4}$$

$$\hat{\varepsilon}_i = \mathbf{y}_i - \hat{f}(x_{i\cdot}) - Z_i \hat{\mathbf{b}}_i \tag{4.5}$$

$$\hat{t}_1 = \sum_{i=1}^{m} \left(\hat{\varepsilon}_i^T \hat{\varepsilon}_i + tr(Var(\hat{\varepsilon}_i | y_i, \hat{f}, \hat{\theta})) \right) \tag{4.6}$$

$$\hat{t}_2 = \sum_{i=1}^{m} \left(\hat{\mathbf{b}}_i \hat{\mathbf{b}}_i^T + Var(\hat{\mathbf{b}}_i | y_i, \hat{f}, \hat{\theta}) \right) \tag{4.7}$$

These formulas match those of Laird and Ware, except that our $\hat{f}(x_i)$ is a regression tree instead of a linear function. While this method is similar in spirit to the EM algorithm, it does not maximize a likelihood and therefore is not a true EM algorithm and does not necessarily have the same properties.

As an alternative approach, we may estimate the random effects associated with a regression tree using a traditional random effects linear model with fixed effects corresponding to the fitted values, $\hat{f}(x_i)$. Estimation methods for such models are

included in most statistical packages. Then, we can subtract the estimated random effects from the target variable and estimate a new regression tree, as before. This yields an alternative estimation method.

Algorithm 4.2 Estimation of a RE-EM Tree.

1. *Initialize the random effects,* $\hat{\boldsymbol{b}}_i$*, to zero.*

2. *Iterate through the following steps until the estimated random effects,* $\hat{\boldsymbol{b}}_i$*, converge:*

   (a) *Estimate a regression tree approximating* f*, based on the target variable,* $y_{it} - Z_{it}\hat{\boldsymbol{b}}_i$*, and predictors,* $\boldsymbol{x}_{it\cdot} = (x_{it1}, ..., x_{itK})$*, for* $i = 1, ..., I$ *and* $t = 1, ..., T_i$*. Use this regression tree to create a set of indicator variables,* $I(\boldsymbol{x}_{it\cdot} \in g_p)$*, where* g_p *ranges over all of the terminal nodes in the tree.*

   (b) *Fit the linear random effects model,* $y_{it} = Z_{it}\boldsymbol{b}_i + I(\boldsymbol{x}_{it\cdot} \in g_p)\mu_p + \epsilon_{it}$*. Extract* $\hat{\boldsymbol{b}}_i$ *from the estimated model.*

In this version of the algorithm, Step 2*b* contains its own optimization, in order to estimate the random effects from the linear model. Including the optimization within this step often leads to fewer iterations for the algorithm, as the estimated random effects converge more quickly.

The linear model with random effects in Step 2*b* can be estimated using maximum likelihood or using restricted maximum likelihood (REML). In most of the results we present, we estimate the linear model with REML, because it yields unbiased estimates for the variance. As we show in Section 4.6.5, using maximum likelihood instead of REML has a very small effect on the resulting estimates.

Using a linear model with random effects directly also allows us to account for autocorrelation using existing estimation methods for linear models. Allowing

for autocorrelation can lead to different estimated effects and therefore different trees. Therefore, testing for autocorrelation is not as straightforward as for parametric models because different trees have different implied linear models. In this paper, we test for autocorrelation using two different likelihood ratio tests, one in which the linear model being estimated corresponds to the RE-EM tree where autocorrelation is not allowed and one corresponding to the RE-EM tree where autocorrelation is allowed. In the examples we consider in Sections 4.4 and 4.5, the two tests lead to the same conclusions.

Given a RE-EM tree, the associated random effects, and the estimated covariance matrices, we can predict out-of-sample observations. Suppose the tree is estimated on data for individuals $i = 1, ..., N_1$ for periods $t = 1, ..., T_1$; for notational simplicity, we are assuming that all individuals have the same number of observations, though this is not required. The first type of out-of-sample prediction is predicting future observations for individuals in the sample; that is, $t > T_1$ for $1 \leq i \leq N_1$. For this sort of prediction, we predict $f(x_{it1}, ..., x_{itK})$ using the estimated tree and then add on $Z_i\hat{\mathbf{b}}_i$, which is known from the estimation process. The second sort of prediction is for individuals for whom there are no observations of the response; that is, $i > N_1$. Then, we have no basis for estimating $\mathbf{b}_i$, so we set it to $\mathbf{0}$. Therefore, our best predictor is $f(x_{it1}, ..., x_{itK})$. Finally, we might wish to predict future observations for new individuals; that is, $i > N_1$ but the target is observed for $t = 1, ..., T_1$ and we wish to predict for $t > T_1$. Then, we can use the observations in the first T_1 periods to estimate $\hat{\mathbf{b}}_i$. Estimating the new random effect uses equations (4.3) and (4.4), with Z_i equal to the design matrix for the new individual and R_i equal to the covariance matrix for the individual based on the estimated parameters from the original model. We then proceed with prediction as before.

To illustrate our method, consider the artificial data given in Figure 69. This is a panel of six individuals observed over three periods each. These data are generated according to the model:

$$\begin{aligned} y_{it} &= b_i + 2x_i + 3I(t > 2 \cap x_i = 1) + \varepsilon_{it} \\ b_i &\sim Normal(0, 1) \\ \varepsilon_{it} &\sim Normal(0, 1) \end{aligned}$$

where $I(\cdot)$ is the indicator function which is 1 if the statement is true and 0 otherwise. We define x_i as a time-invariant covariate which is 1 for $i = 1, 2, 3$ and zero for $i = 4, 5, 6$. This model corresponds to a tree structure with two splits: the first is based on x_i, and the second is in the branch where $x_i = 1$ and based on whether $t > 2$.

To estimate the RE-EM tree for this data using Method 1, we set $Z_i = \mathbf{1}_4$ to be a vector of length 4 consisting of ones for each i, and we set $\hat{D} = I_{1\times1}$, $\hat{b}_i = 0$, $\hat{R}_i = I_{4\times4}$, where $I_{n\times n}$ is an $n \times n$ identity matrix. Since $\hat{b}_i = 0$, in the first step we apply `rpart` to the dataset with predictors t and x_i and observed values y_{it}, for all i and t. In this simple case, the tree in the initial estimation matches the true tree structure, though the means at the nodes are not identical to the true means because of the random effects and random errors. Because t takes on only integer values, the split $t < 2.5$ is equivalent to splitting on $t \leq 2$ and $t > 2$. Given this tree, we first compute the weighting matrix:

$$\begin{aligned} W_i &= (I_{4\times4} + \mathbf{1}_4 I_{1\times1} \mathbf{1}_4^T)^{-1} \\ &= \begin{pmatrix} 0.8 & -0.2 & -0.2 & -0.2 \\ -0.2 & 0.8 & -0.2 & -0.2 \\ -0.2 & -0.2 & 0.8 & -0.2 \\ -0.2 & -0.2 & -0.2 & 0.8 \end{pmatrix} \end{aligned}$$

Using this weighting matrix, we compute the estimated random effects, $\hat{b}_i$, and errors, $\hat{\varepsilon}_{it}$ for all observations and periods. The errors are shown in Figure 70. The estimated random effects are

$$(-1.2431335, 1.446752, -0.2036185, -1.5737146, 0.4378226, 1.135892)$$

Given the estimated errors and random effects, we then estimate the covariance matrices that they imply:

$$\begin{aligned} D &= [2.273075] \\ R_i &= 1.567379 I_{4\times 4} \end{aligned}$$

A new tree is then estimated based on the target variable, $y_{it} - \hat{b}_i$. The second estimate of the tree is identical to the first. The new values of D, R_i, and the tree lead to a new weighting matrix and new estimates for the errors and random effects. The estimates for the errors are shown in Figure 71. The algorithm continues through additional iterations until the estimates of the random effects change by only a small amount from one iteration to the next. In this case, three iterations are required, the final estimated random effects are $(-1.41, 1.64, -.023, -1.79, 0.50, 1.29)$, and the final estimated errors are given in Figure 72. The estimated variance of the random effects is $D = [2.780]$, which is lower than the true variance of 4. The estimated error covariance matrix is $R_i = 0.903 I_{4\times 4}$, which is close to the true covariance matrix of $I_{4\times 4}$. These errors are no longer separated by group and no longer show a pattern with respect to time. In this particular case, the tree structure, given in Figure 73 does not change at each iteration, though the random effects estimates do. In fact, because the random effects are constrained to sum to 0, the estimated means at the nodes do not change either in this simple case. In more complicated cases, the tree structure and

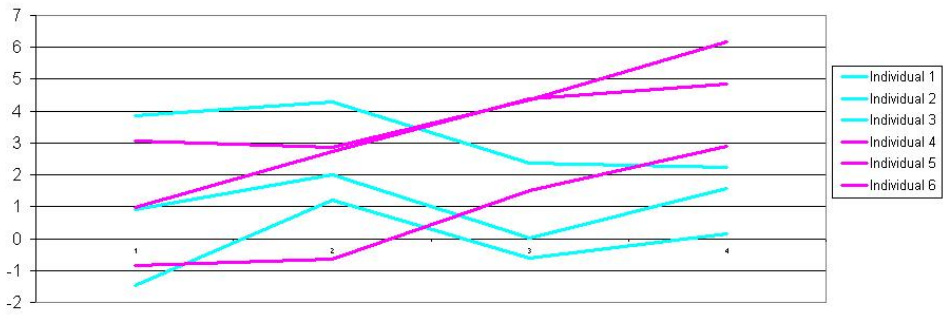

Figure 69: Data for the simple example of RE-EM trees.

the means at the nodes may change many times as the estimated random effects evolve.

We repeat this exercise using Method 2. As before, we initialize the random effects to 0, so that we fit our first regression tree to the original data, ignoring the panel data structure. Next, instead of computing $\hat{D}$ and $\hat{R}_i$ directly, we fit a linear random effects model to the data. This model has three predictors, which are indicator variables for the three terminal nodes of the fitted tree. That is, the predictors for the linear model are $I(x_i = 0)$, $I(x_i = 1, t < 2.5)$, and $I(x_i = 1, t > 2.5)$. The fitted linear model estimates $\hat{D} = 2.7801$ and $\hat{R}_i = 0.9025965 I_{4\times 4}$. The estimated random effects after the first round are

$$(-1.4478475, 1.6849971, -0.2371496, -1.8328674, 0.5099214, 1.3229460)$$

and the estimated errors are shown in Figure 74. For the second iteration of the algorithm, we fit the tree again, using the target variable less the estimated random effects. Because the fitted tree is identical in this simple case, the estimated random effects are identical, and the algorithm converges in two steps instead of three. Notice that the estimated trees are identical and the estimated random

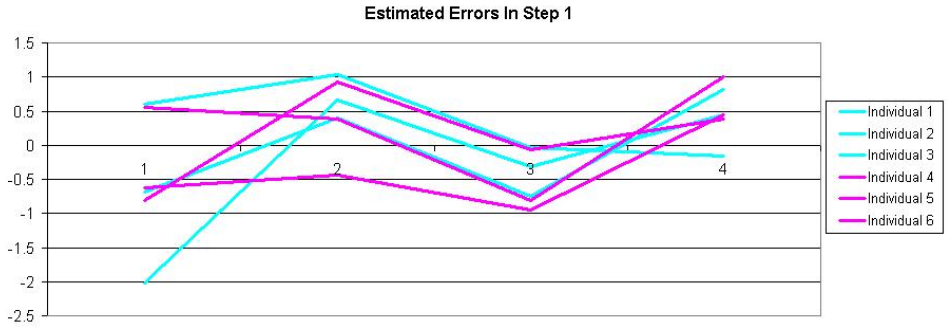

Figure 70: Estimated errors after the first iteration of the RE-EM procedure for the simple example.

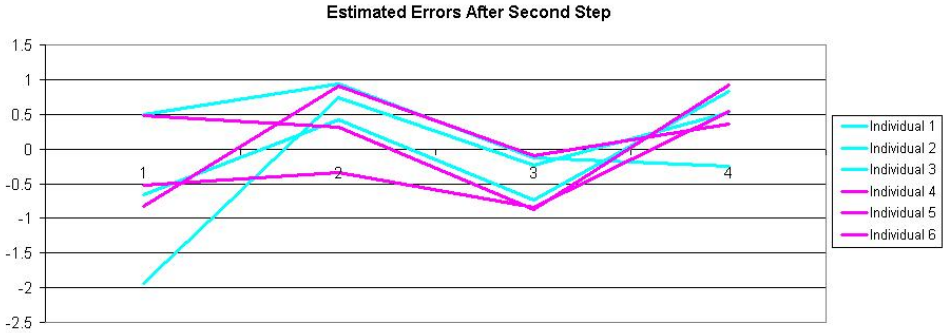

Figure 71: Estimated errors after the second iteration of the RE-EM procedure for the simple example.

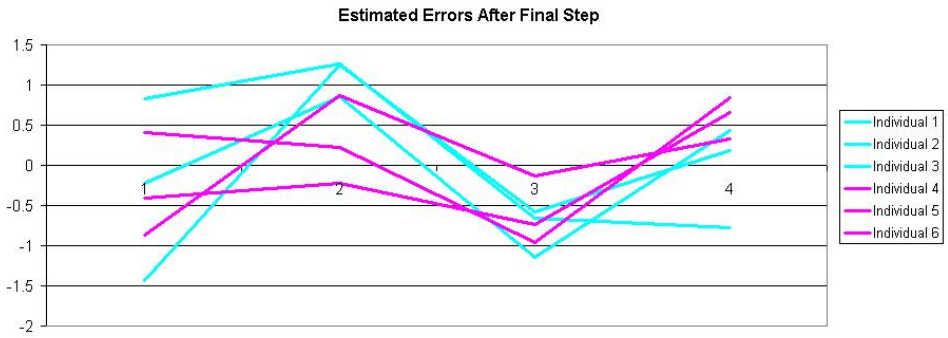

Figure 72: Estimated errors after the final iteration of the RE-EM procedure for the simple example.

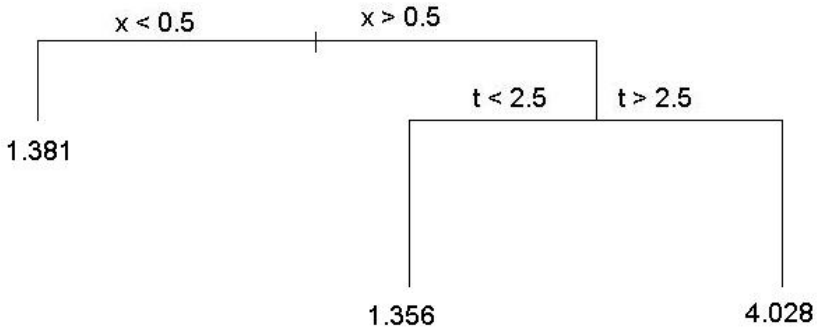

Figure 73: Estimated RE-EM tree for the simple example.

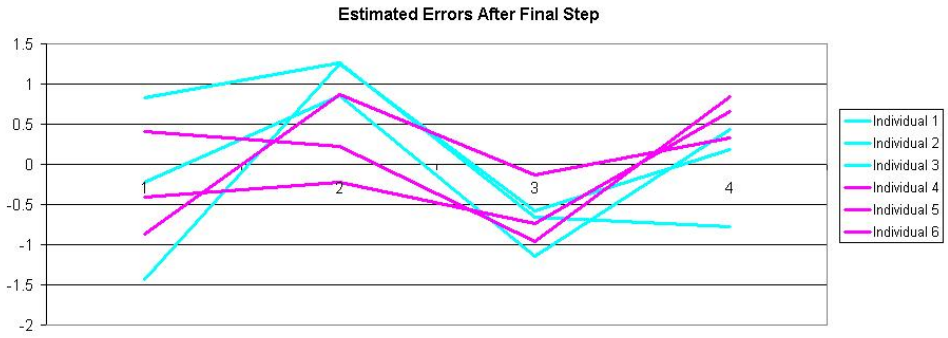

Figure 74: Estimated errors after the final iteration of the RE-EM procedure for the simple example, using Method 2.

effects are similar across the two methods.

In more complicated cases, the tree structure may change from one iteration to the next, so that the estimated random effects change each iteration. Even though this occurs, we have found that Method 2 often converges in fewer iterations than Method 1 does, with similar results, as we will show in Section 4.6.5. Therefore, unless otherwise stated, we will use Method 2 in the remainder of this paper.

4.4 Application to State Traffic Fatality Rates

As our first application, we consider data on traffic fatality rates for the forty-eight contiguous states from 1982-1999. These data were first studied by Dee and Sela [2003] using a variety of parametric longitudinal data models to find the effect of increased highway speed limits on the traffic fatality rates for different demographic groups. In this paper, we focus on three traffic fatality rates: the overall traffic fatality rate per 100,000 in population, the traffic fatality rate for drivers aged 16 to 24, and the traffic fatality rate for drivers aged 65 and older. These two age

groups are usually of the most interest to policy-makers, since they have higher traffic fatality rates than the rest of the population. Most of the predictors that Dee and Sela used were traffic law variables, such as seat belt regulations, drunk driving laws, and the state maximum speed limit; the state unemployment rate was also a predictor, to proxy for the effect of the economy. (See Dee and Sela [2003, Section 2] for more information about the predictors used.) In addition to those predictors, we include the proportion of the population that falls into the different demographic groups, by both age and gender, the number of vehicle miles traveled, and the total population of the state.

In addition to the variables included, we might expect time-constant, state-level effects for a variety of reasons. The geographic location may affect traffic fatalities, because of the dangers of colder or wetter weather. The amount of urbanization may affect the amount that people drive and therefore the chances of traffic fatalities. These two factors are likely to affect the estimated effects for all three groups. States that have a large number of visitors or part-year residents, such as Florida, may have higher traffic fatality rates, because these drivers are not counted as part of the population. This factor might affect the random effects for the different groups by different amounts, depending on the types of people who visit these areas.

In the original work using these data, Dee and Sela showed that the inclusion of state and year fixed effects has a dramatic effect on the conclusions drawn from the data. With neither state nor year effects, raising the speed limit from 55 to 65 miles per hour or to a higher speed limit was associated with a large, significant increase in the overall traffic fatality rate. However, a model with state effects, year effects, state-specific time trends, and the previously mentioned control variables showed a small but significant decline in the traffic fatality rate associated with increasing

the speed limit from 55 to 65 miles per hour and a small and insignificant increase in the traffic fatality rate associated with raising the speed limit from 55 miles per hour to 70 or above. In regressions with the same predictors and effect structure, they found statistically insignificant speed limit effects for the 16-24 age group, but a statistically significant increase in the traffic fatality rate for people 65 and over associated with the change from a 55 mile per hour speed limit to a speed limit of 70 or more.

We first estimate two linear random effects models and one linear model that does not include state-level fixed or random effects with the data. One linear random effects model allows for autocorrelation in the form of an autoregression of order one in the errors, while the other does not. For the estimation of the linear models, we include only those variables used in Dee and Sela, since the demographic variables are highly collinear. The parameter estimates are given in Tables 58, 60 and 62. When there are no random effects, the speed limit effects are estimated to be large, positive, and significant, as Dee and Sela found. The results when random effects are included are similar to those of Dee and Sela, who included fixed effects instead of random effects. In the case of the overall traffic fatality rate, all of the estimates have the same signs as the results from the fixed effects model, but the negative effect of the 65 mile per hour speed limit is no longer statistically significant, while the positive effect of the 70 mile per hour speed limit is now marginally significant in the case of no autocorrelation. As Dee and Sela found, the inclusion of state-level effects reverses our interpretation of the safety impact of increasing the speed limit to 65 MPH. Inclusion of an autoregressive component in the errors inflates the standard errors but does not change the estimated coefficients appreciably. Likelihood ratio tests comparing the models with and without autocorrelation strongly favor models with autocorrelation ($p < 10^{-12}$

for all three demographic groups).

The traditional regression tree for the overall traffic fatality rate is given in Figure 75, while the RE-EM tree for the overall traffic fatality rate is given in Figure 76. The regression tree is dramatically more complicated than the RE-EM tree, because it must account for state effects within the tree structure. The RE-EM tree splits primarily on demographic variables, such as the percentage of the population that is young and female (16-19 or 20-24) or that over 65 and female; this makes sense since those are the two age groups that have the highest traffic fatality rates and since the fatality rates can vary by gender as well. The number of vehicle miles traveled also has an influence in one branch. Interestingly, no traffic law variables are included in this tree.

In Figure 77, we plot the estimated autocorrelation function for the residuals from the RE-EM tree. This plot suggests that there is autocorrelation in the residuals, with a pattern of decay that suggests an autoregressive model of order 1. Therefore, we re-fit the RE-EM tree allowing for $AR(1)$ autocorrelation in the model for the effects. The resulting RE-EM tree is given in Figure 78. The fitted tree is similar to the tree that does not allow for autocorrelation; the initial splits are identical, but the later splits differ. In this tree, two traffic law variables, both related to drunk driving, appear in later splits in the tree. The estimated autocorrelation parameter is 0.682. Likelihood ratio tests reject the model that does not allow for autocorrelation ($p < 10^{-20}$). Figure 79 shows that including autocorrelation in the effects model has removed most of the autocorrelation from the residuals, although the fitted model has apparently induced a small negative autocorrelation at lag 1.

We compare a variety of diagnostic measures for the linear model with random effects and the RE-EM tree, based on the residuals when we allow for $AR(1)$

	Overall	Youth	Elderly
(Intercept)	98.66507***	190.8942***	57.11093***
	(8.97076)	(14.7963)	(8.65138)
Year	-0.69219***	-1.3406***	-0.35718***
	(0.08948)	(0.1476)	(0.08629)
Speed Limit = 65	7.15477***	11.9351***	5.66071***
	(0.67118)	(1.107)	(0.64729)
Speed Limit = 70	12.90605***	19.0259***	10.41544***
	(1.21727)	(2.0078)	(1.17394)
Speed Limit = 75	15.80194***	20.6055***	12.9224***
	(1.29147)	(2.1301)	(1.24549)
No Speed Limit	20.79805***	31.0482***	16.92129***
	(3.27849)	(5.4075)	(3.16177)
Drinking Age	-1.42971	-0.7947	-1.75325
	(1.44402)	(2.3818)	(1.39261)

Table 57: Estimates for a linear model without random effects on traffic fatality data. * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

	Overall	Youth	Elderly
Effective Drinking Age	0.46987	-1.0881	1.39135
	(1.44639)	(2.3857)	(1.39489)
Seatbelt Law	-0.46292	-0.6311	-0.13931
	(0.32517)	(0.5363)	(0.31359)
Zero Tolerance	-0.60897	-1.5543	0.1275
	(0.62563)	(1.0319)	(0.60335)
Illegal at 0.10 or higher BAC	-1.60223**	-3.4322***	-0.75947
	(0.64641)	(1.0662)	(0.62339)
Illegal at 0.08 or higher BAC	-2.83248***	-5.0305***	0.08456
	(0.92259)	(1.5217)	(0.88974)
Administrative License Revocation	0.99472*	1.9489**	0.49321
	(0.50959)	(0.8405)	(0.49145)
State Unemployment Rate	47.2114***	30.7516	21.14483*
	(12.04553)	(19.8677)	(11.61668)

Table 58: Estimates for a linear model without random effects on traffic fatality data (continued). * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

	Overall	Youth	Elderly
(Intercept)	62.42159***	131.82208***	14.30351**
	(4.146223)	(9.868981)	(7.10903)
Year	-0.43682***	-0.94767***	0.00889
	(0.037717)	(0.091593)	(0.066439)
Speed Limit = 65	-0.44886	0.64592	-0.3143
	(0.296813)	(0.720802)	(0.522835)
Speed Limit = 70	1.00679**	2.67542**	0.92691
	(0.519861)	(1.262892)	(0.916517)
Speed Limit = 75	0.32998	-0.82559	-0.26072
	(0.545555)	(1.325681)	(0.962492)
No Speed Limit	1.05358	-3.15316	5.60697**
	(1.283375)	(3.122235)	(2.270924)
Drinking Age	0.13975	0.94384	-0.61686
	(0.509252)	(1.240274)	(0.903617)

Table 59: Estimates for a Linear Model With Random Effects on Traffic Fatality Data. * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

	Overall	Youth	Elderly
Effective Drinking Age	0.15512 (0.513689)	-0.74626 (1.250999)	1.12263 (0.911339)
Seatbelt Law	-0.10496 (0.144242)	-0.24742 (0.350382)	-0.04974 (0.25425)
Zero Tolerance	0.67259** (0.27076)	-0.02206 (0.657761)	0.92016* (0.477361)
Illegal at 0.10 or higher BAC	-1.19929*** (0.32447)	-3.25404*** (0.786506)	-0.45141 (0.568881)
Illegal at 0.08 or higher BAC	-1.95985*** (0.490386)	-4.5206*** (1.187307)	-0.5522 (0.857278)
Administrative License Revocation	-1.19046*** (0.283271)	-1.53646** (0.685298)	-1.12318** (0.494211)
State Unemployment Rate	-72.84035*** (6.024733)	-143.87058*** (14.60634)	-36.37834*** (10.567461)

Table 60: Estimates for a Linear Model With Random Effects on Traffic Fatality Data (continued). * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

	Overall	Youth	Elderly
(Intercept)	62.01909***	128.06005***	17.914821**
	(5.266569)	(11.720584)	(8.341541)
Year	-0.40661***	-0.92632***	0.002843
	(0.049783)	(0.110499)	(0.078992)
Speed Limit = 65	-0.22177	0.32622	0.121079
	(0.329678)	(0.815068)	(0.587671)
Speed Limit = 70	0.78943	2.48954*	1.472234
	(0.594690)	(1.440762)	(1.037760)
Speed Limit = 75	0.38165	-0.81324	0.586569
	(0.663700)	(1.551812)	(1.114612)
No Speed Limit	-0.50025	-1.89094	3.851167
	(1.774208)	(3.862005)	(2.760424)
Drinking Age	0.18101	0.62095	-0.857127
	(0.420657)	(1.193983)	(0.879430)
Effective Drinking Age	-0.06150	-0.36043	1.176236
	(0.448331)	(1.231629)	(0.904367)

Table 61: Estimates for a Linear Model With Random Effects that allows for autocorrelation in the errors. * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

	Overall	Youth	Elderly
Seatbelt Law	0.00366	-0.17236	0.005488
	(0.167905)	(0.405913)	(0.291941)
Zero Tolerance	0.54450*	0.04536	0.775221
	(0.314691)	(0.760825)	(0.547556)
Illegal at 0.10 or higher BAC	-3.17865***	-3.25404***	-0.430758
	(0.862268)	(0.786506)	(0.621078)
Illegal at 0.08 or higher BAC	-3.96598***	-4.5206***	-0.583449
	(1.353713)	(1.187307)	(0.967941)
Administrative License Revocation	-1.73787**	-1.53646**	-0.958228*
	(0.803572)	(0.685298)	(0.571800)
State Unemployment Rate	-58.49532***	-134.14615***	-30.436739**
	(7.408274)	(17.219815)	(12.295101)
Autoregressive Parameter	0.5833092	0.3185156	0.2893601

Table 62: Estimates for a Linear Model With Random Effects that allows for autocorrelation in the errors (continued). * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

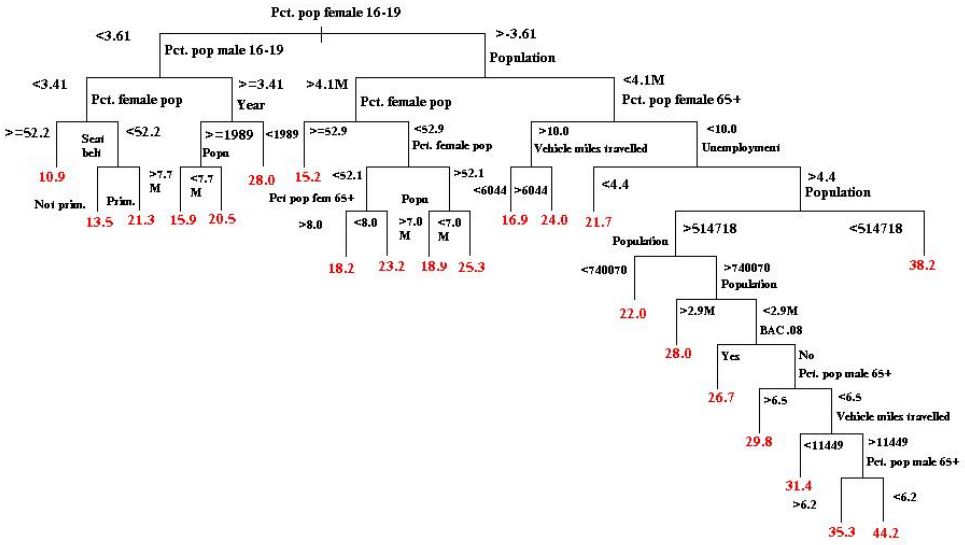

Figure 75: Estimated tree without random effects for the overall traffic fatality rate.

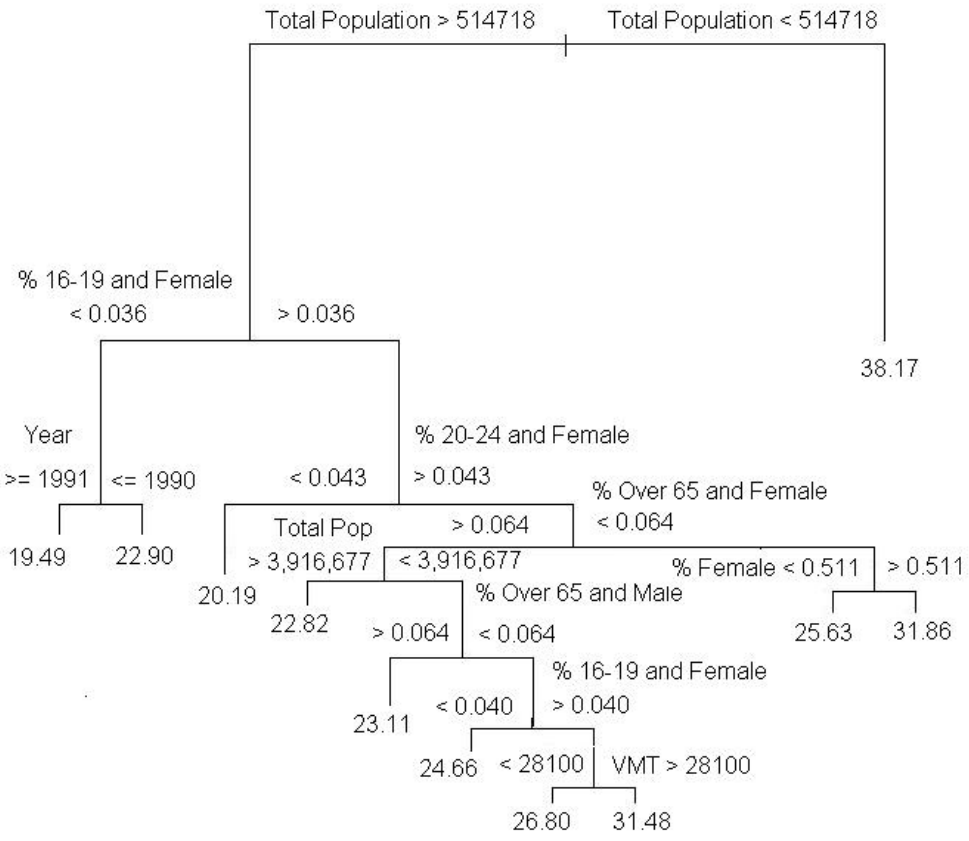

Figure 76: Estimated RE-EM tree for the overall traffic fatality rate.

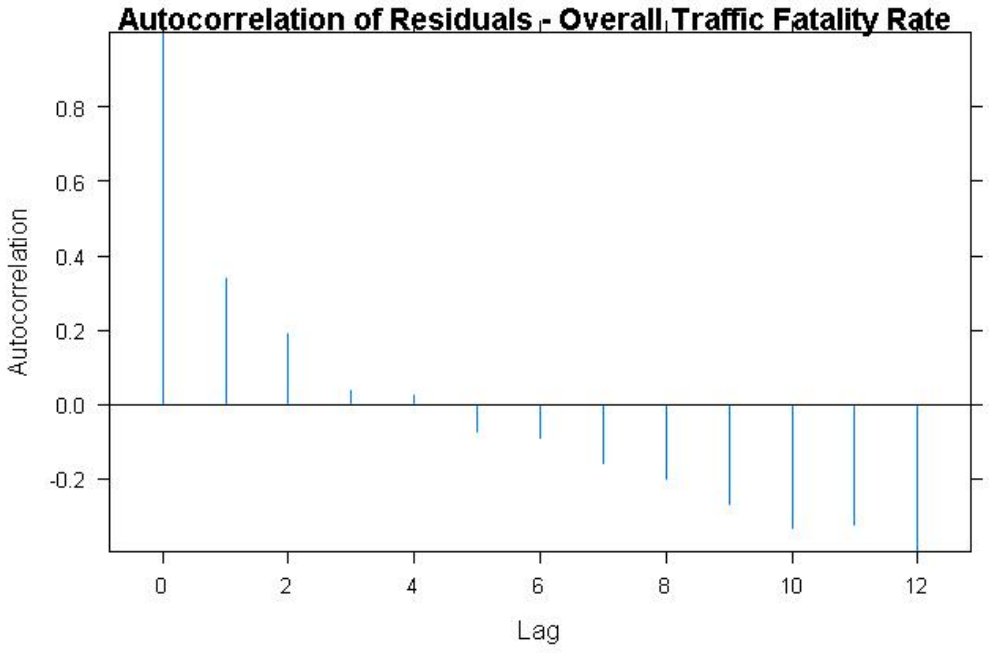

Figure 77: Estimated autocorrelation function of the estimated residuals for each state for the overall traffic fatality rate, when there is no autoregressive component in the RE-EM tree.

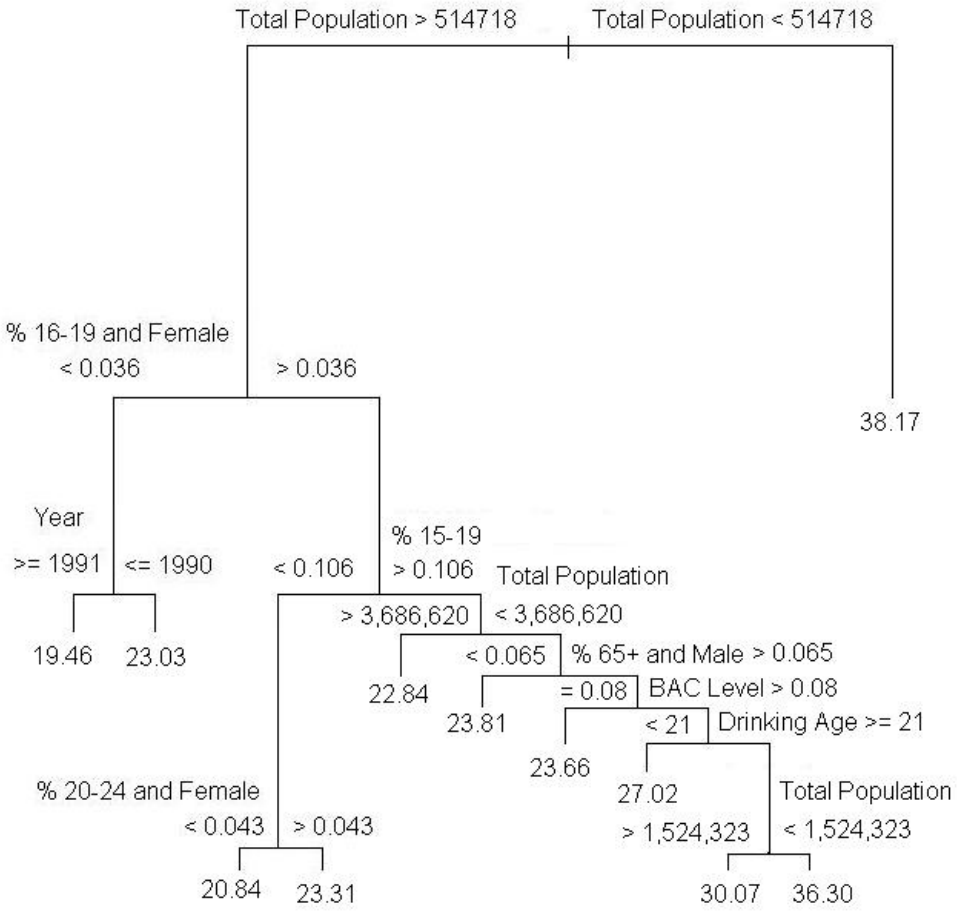

Figure 78: Estimated RE-EM tree with autocorrelation for the overall traffic fatality rate.

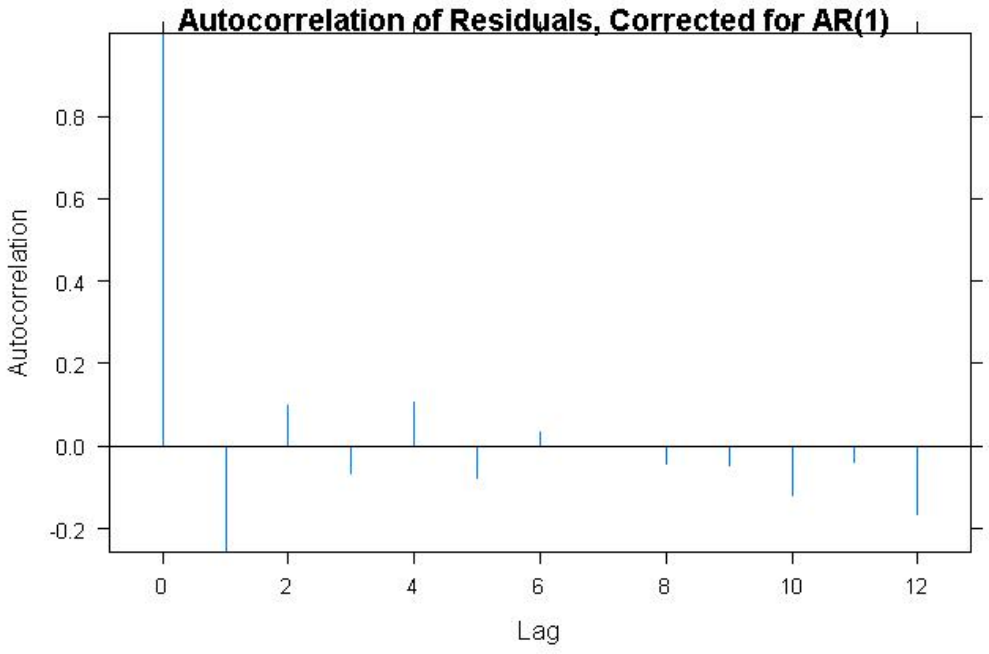

Figure 79: Estimated autocorrelation function of the estimated residuals for each state for the overall traffic fatality rate, when there is an autoregressive component in the RE-EM tree.

autocorrelation in the errors. First, we plot the fitted values versus the residuals for each model in Figures 80 and 82. Both show evidence of heteroskedasticity, with a larger variance associated with larger fitted values. The difference is more pronounced for the linear model than for the RE-EM tree. In Figures 81 and 83, we plot the residuals for each state. The variability of residuals differs across states in both models, with New Mexico and Wyoming having the most variable residuals in both models. Montana's residuals from the linear model have more variability than its residuals from the RE-EM tree. These three states have three of the four highest average fatality rates (Mississippi is third), which suggests that the differences in variability across states can be attributed to the increased variability when the fitted values are higher. To check for normality, Figures 84 and 85 show quantile-quantile plots by state. In both models, there are some deviations from normality in the upper tails of the distributions of residuals for Wyoming, New Mexico and Montana. The heteroskedasticity and the deviations from normality in the tails suggest that taking logs might be helpful in modeling. We will do that later in this section.

The RE-EM trees without autocorrelation for the youth and for the elderly are given in Figures 86 and 88, respectively. As with the trees for the overall traffic fatality rate, the RE-EM trees are dramatically more parsimonious than the trees without random effects, which we do not present here. The RE-EM trees that allow for autocorrelation, given in Figures 87 and 89, have initial splits that are identical to the RE-EM tree without autocorrelation, but later splits differ. This again matches what we found in the case of the overall tree. As before, likelihood ratio tests strongly reject the model without autocorrelation, though the estimated autocorrelations are smaller (0.276 for youth and 0.211 for the elderly). In the youth RE-EM tree, only year and demographics matter. Most of the demographic

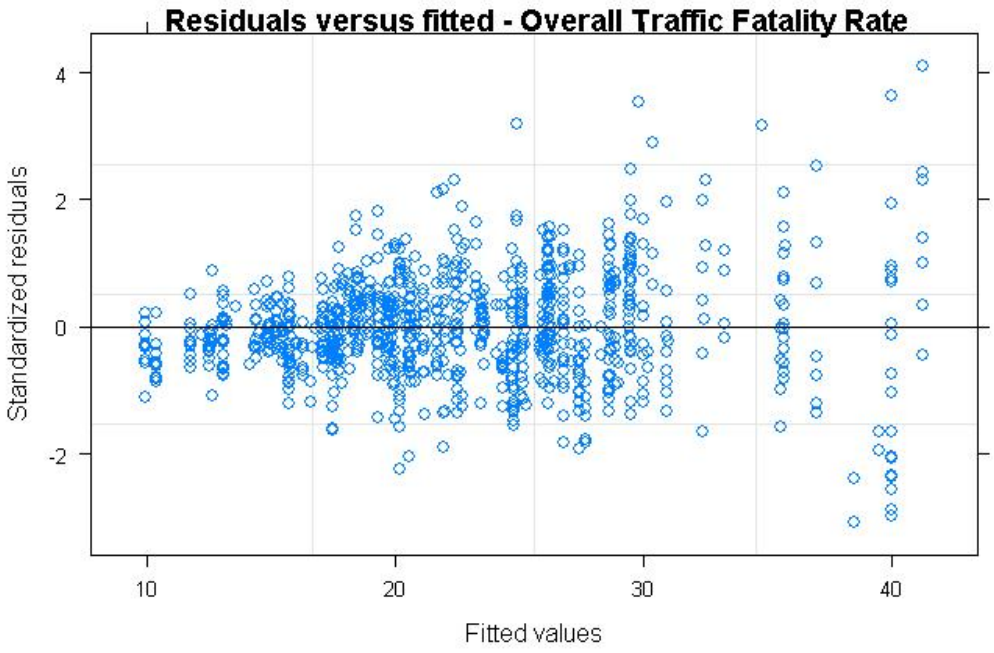

Figure 80: Plots of the residuals versus the fitted values from the RE-EM tree for the overall traffic fatality rate.

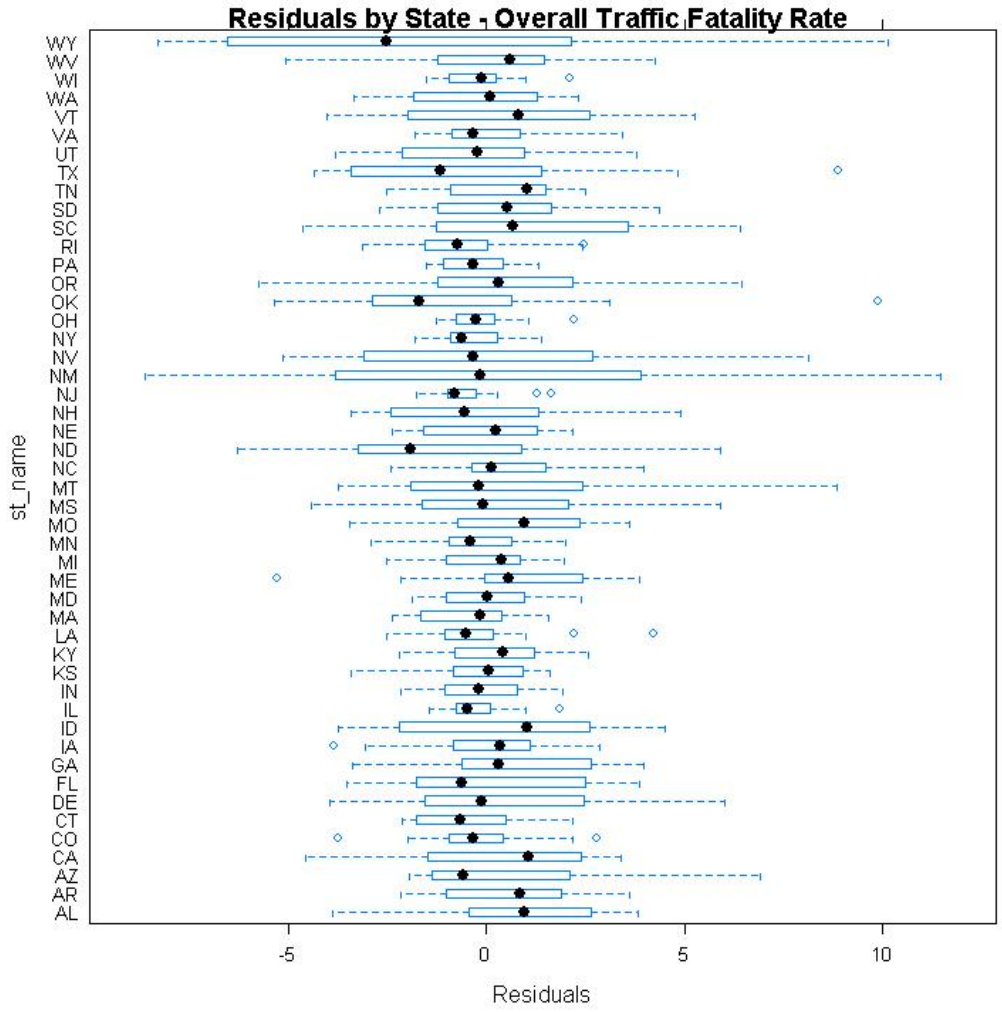

Figure 81: Boxplots of the estimated residuals from the RE-EM tree for each state for the overall traffic fatality rate.

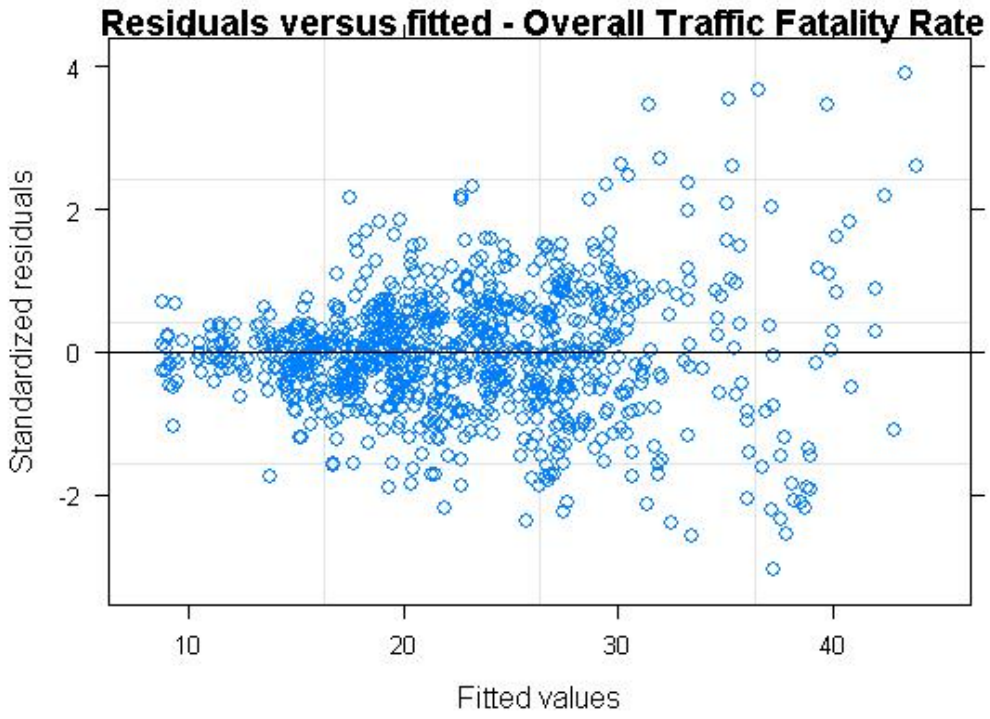

Figure 82: Plots of the residuals versus the fitted values from the linear model with random effects for the overall traffic fatality rate.

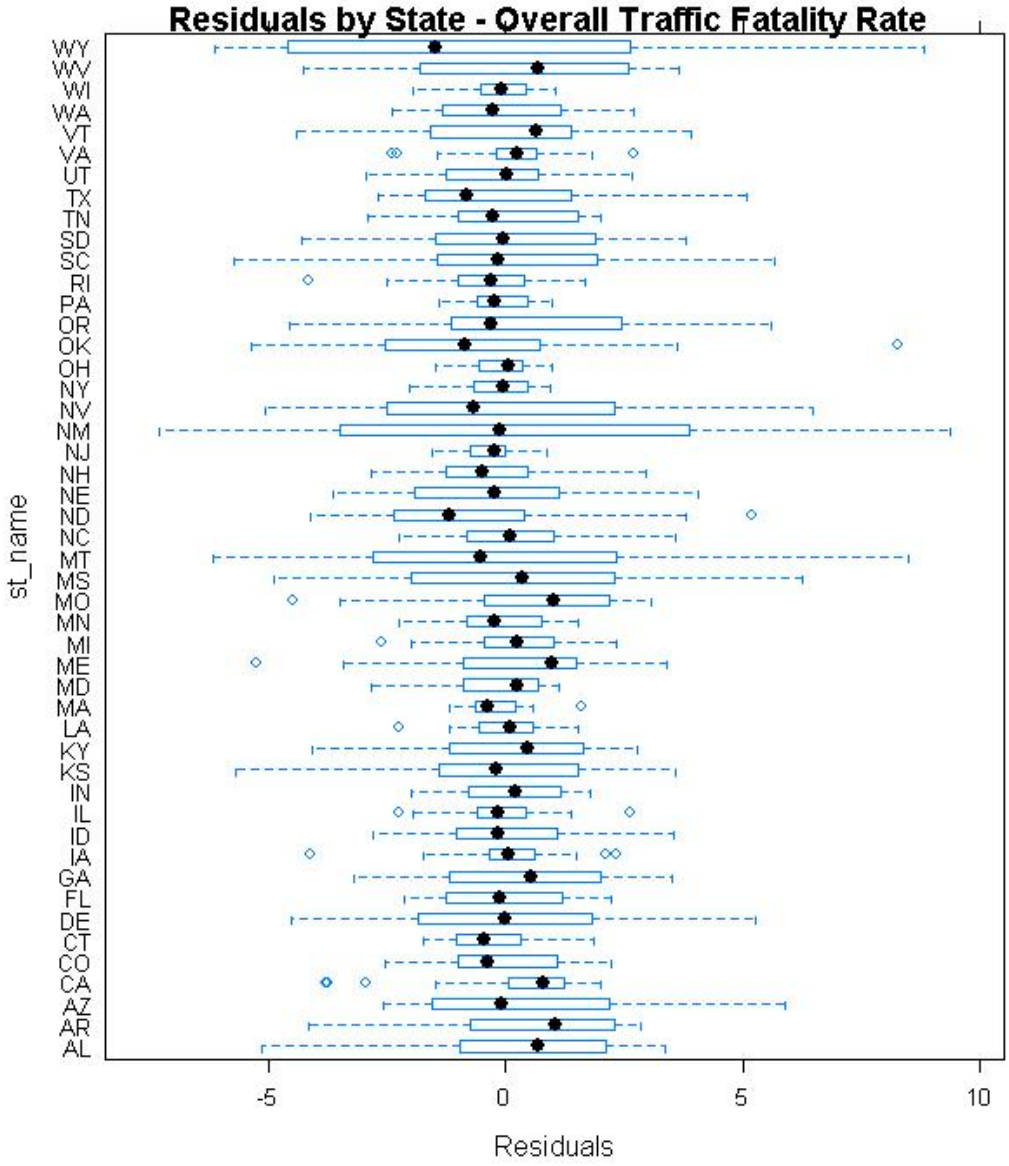

Figure 83: Boxplots of the estimated residuals from the linear random effects model for each state for the overall traffic fatality rate.

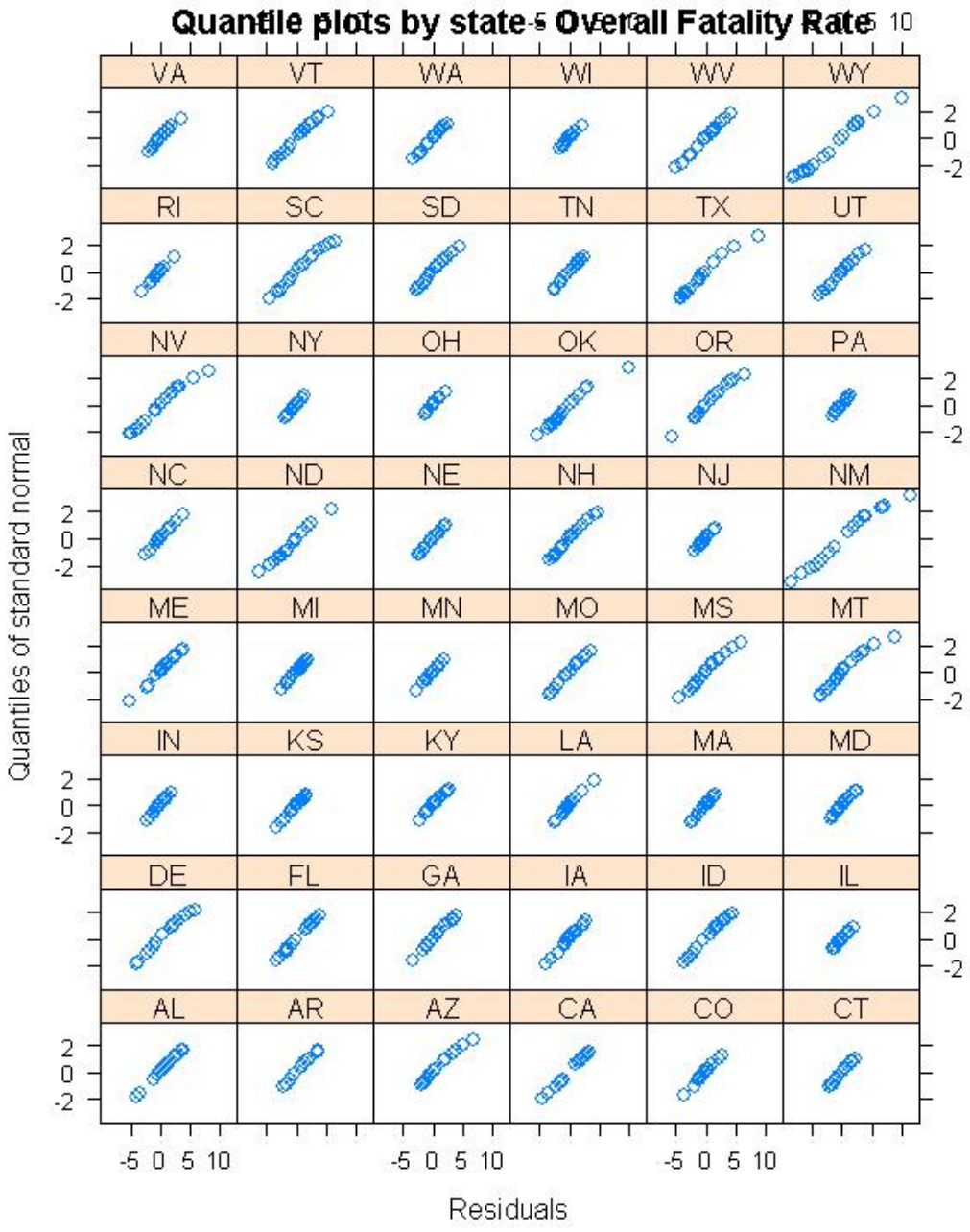

Figure 84: Quantile-quantile plots of the estimated residuals from the RE-EM tree for each state for the overall traffic fatality rate.

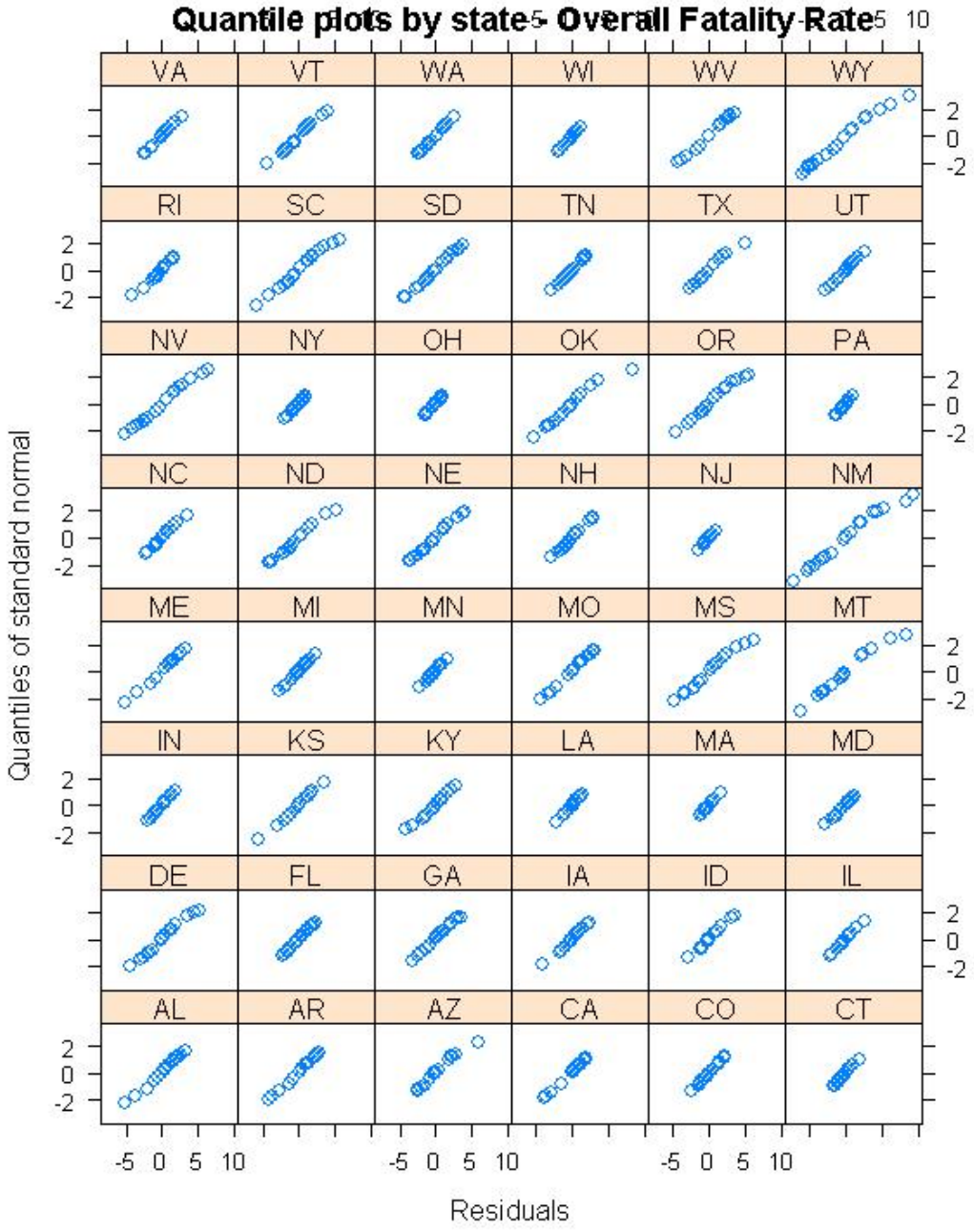

Figure 85: Quantile-quantile plots of the estimated residuals from the linear random effects model for each state for the overall traffic fatality rate.

splits seem to be accounting for the fact that age matters in traffic fatalities for this group; the tree splits on the percentage of the population that is 15-19 or 16-19 and either male or female, while the percentage of the population that is over 65 appears in one split in the tree for the youth traffic fatality rate. The RE-EM tree for older drivers depends on a wide range of variables, including demographic variables, the unemployment rate, and speed limits. This suggests that the other types of drivers on the road, or perhaps the other type of people in the community, influence the traffic fatality rates of specific demographic groups. Speed limits appear in the branch of the tree corresponding to states with a higher percentage of women; this suggests that the traffic fatality rate of older women is most affected by increased speed limits, a finding that agrees with Dee and Sela. As with the other demographic groups, a RE-EM tree that allows for autocorrelation is quite similar. In this case, the split on the number of vehicle miles traveled is removed, but the structure of the tree is identical otherwise.

We can compare the six models that we have fit by computing the root mean squared error of the in-sample fits. We report these in Table 63. These in-sample results show that estimating random effects, whether in a linear model or in a tree, reduces the in-sample RMSE. This is not surprising, since the errors include the random effects if those effects are not estimated. The RE-EM tree has a lower in-sample RMSE than the linear effects model for both the youth and elderly traffic fatality rates, but a slightly higher RMSE for the overall rate. Despite the likelihood ratio tests' rejection of models without autocorrelation, the in-sample RMSE is similar for the two linear models, and the in-sample RMSE of the RE-EM tree without autocorrelation is noticeably smaller in the case of the overall traffic fatality rate, even though the estimated autocorrelation is highest in this model. This may be related to the observation of Verbeke and Molenberghs [2000] that esti-

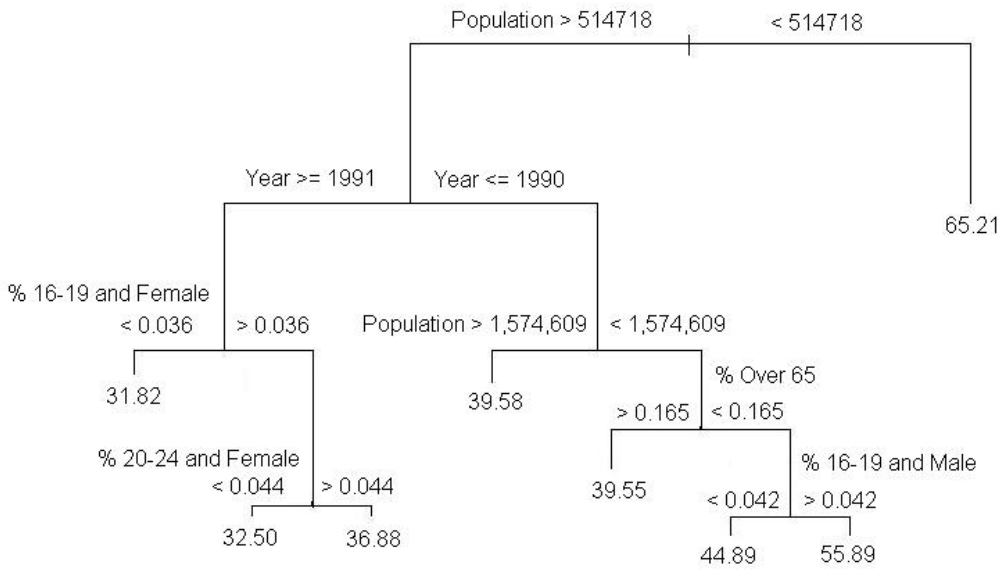

Figure 86: Estimated RE-EM tree for the traffic fatality rate for people aged 16 to 24.

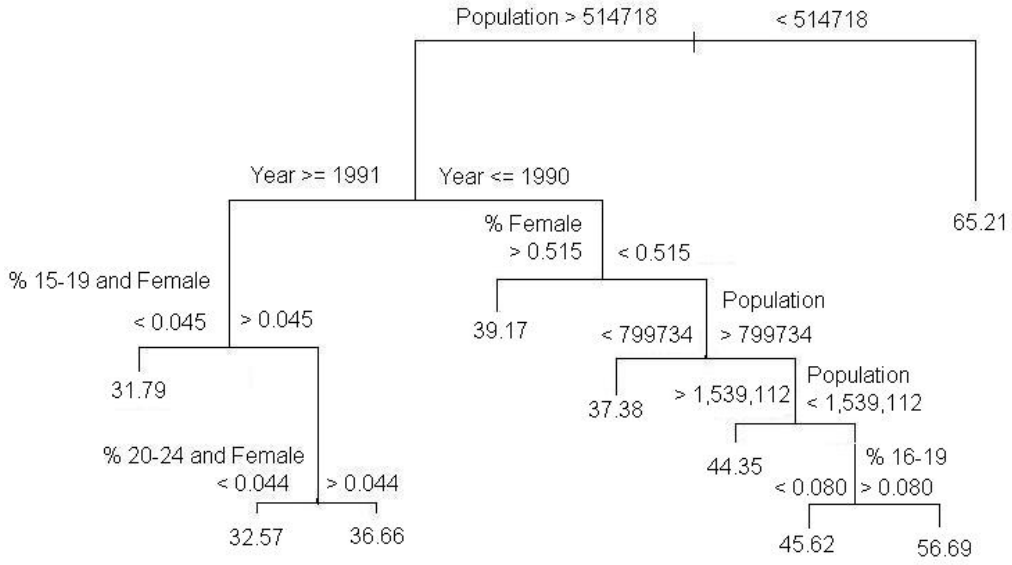

Figure 87: Estimated RE-EM tree for the traffic fatality rate for people aged 16 to 24, allowing for autocorrelation in the error terms.

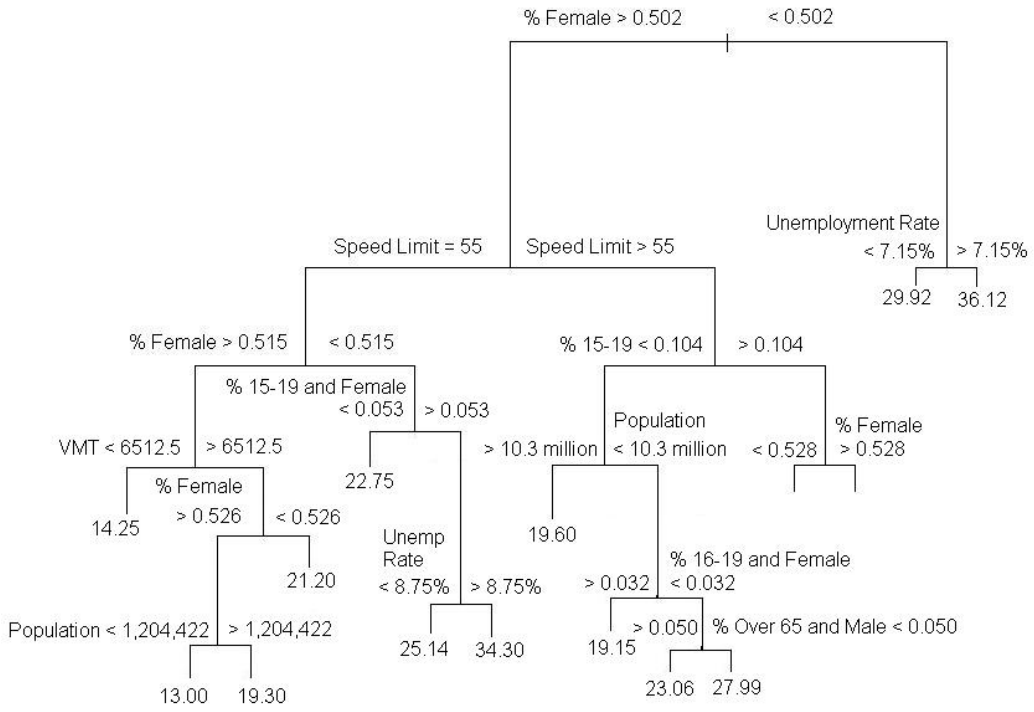

Figure 88: Estimated RE-EM tree for the traffic fatality rate for people 65 and older.

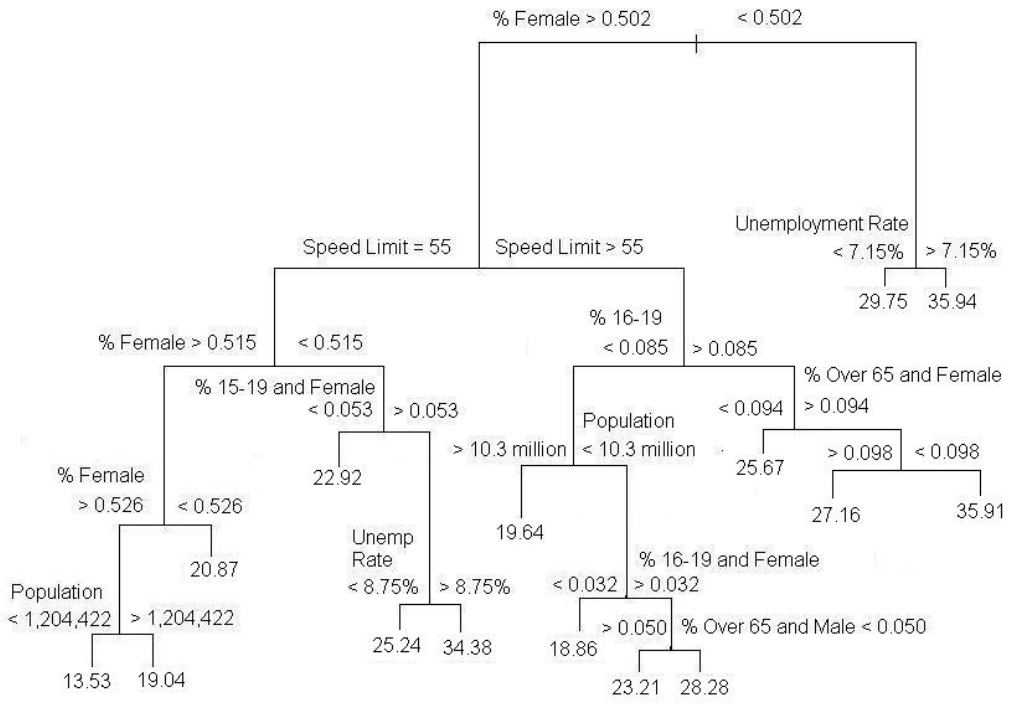

Figure 89: Estimated RE-EM tree for the traffic fatality rate for people 65 and older, allowing for autocorrelation in the error terms.

	Overall	Youth	Elderly
Linear Model	6.234	10.283	6.012
Linear Model with Random Effects	2.063	5.027	3.666
Linear Model with Random Effects and Autocorrelation	2.103	5.043	3.683
Regression Tree	3.880	7.104	4.123
RE-EM Tree	2.070	4.949	3.460
RE-EM Tree with Autocorrelation	2.339	4.937	3.415

Table 63: In-sample root mean squared error for traffic fatality data.

mating the level of autocorrelation is difficult in random effects models. As we can see from the autocorrelation function in Figure 79, the estimated autocorrelation function of the normalized residuals has negative autocorrelation at the first lag. It is possible that the linear model overestimated the amount of autocorrelation, leading to worse in-sample fit.

In Figures 90, 91, and 92, we map the estimated random effects from the RE-EM trees without autocorrelation. Darker shading corresponds to larger estimated random effects. Note that the estimated random effects for the three different fatality rates are highly correlated; the effects for the overall traffic fatality rate have a correlation of 0.954 with the youth fatality rate and 0.831 with the elderly fatality rate. This suggests that the effects largely measure a characteristic that is common to the state, not to the particular demographic group within the state. In particular, we notice that the random effects are generally highest in the Southeastern states, in Arizona, and in New York. The correlation between the random effects from the linear model and the random effects from the corresponding RE-EM tree is also high, as reported in Table 64, while scatter plots of the effects are given in

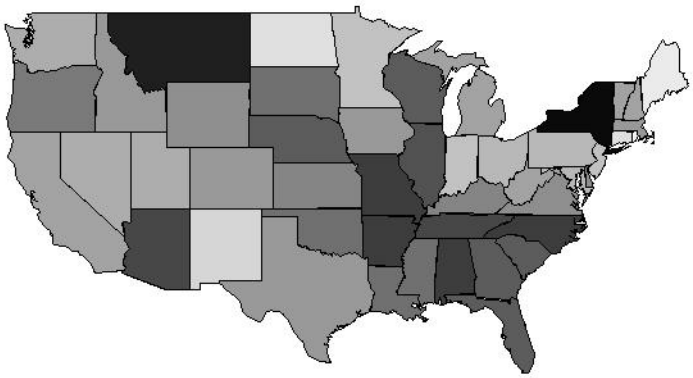

Figure 90: Estimated random effects from the RE-EM tree without autocorrelation for the overall traffic fatality rate.

Figures 93 through 98. The relationship between the estimated effects from the linear model and the estimated effects from the RE-EM tree is quite strong; the clear outlier in the plots is Wyoming, which is estimated to have a large effect by the linear model but which is instead split off into its own branch in the RE-EM tree. This indicates that the linear and tree models estimate the underlying relationship in similar ways for most states. We also compare the estimated random effects from each model to the state-specific means, ignoring the covariates. While there is some positive linear relationship between the state mean fatality rates and the estimated random effects from the RE-EM tree, the relationship is quite weak. The RE-EM tree random effects are less correlated with the state means, suggesting that the RE-EM tree is fitting more than simply the state means.

We also measure the out of sample performance of the various models. We re-estimate the models excluding the last two periods of data (1998 and 1999) for each state and then use the model to predict the traffic fatality rates in 1998 and 1999. The root mean squared errors of prediction are given in Table 65. The RE-EM

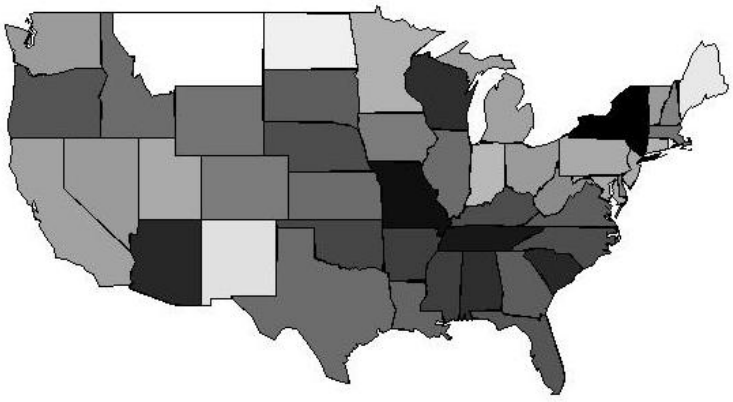

Figure 91: Estimated random effects from the RE-EM tree without autocorrelation for the traffic fatality rate for people aged 16 to 24.

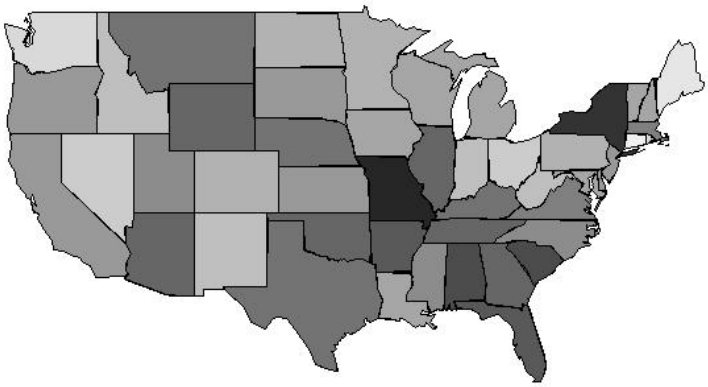

Figure 92: Estimated random effects from the RE-EM tree without autocorrelation for the traffic fatality rate for people aged 65 and older.

	Overall	Youth	Elderly
State Means and RE-EM Tree Effects	0.356	0.341	0.338
State Means and Linear Model Effects	0.387	0.351	0.403
RE-EM Tree Effects and Linear Model Effects	0.907	0.886	0.885

Table 64: Correlations of state means and state-level effects for the RE-EM tree without autocorrelation and linear model without autocorrelation.

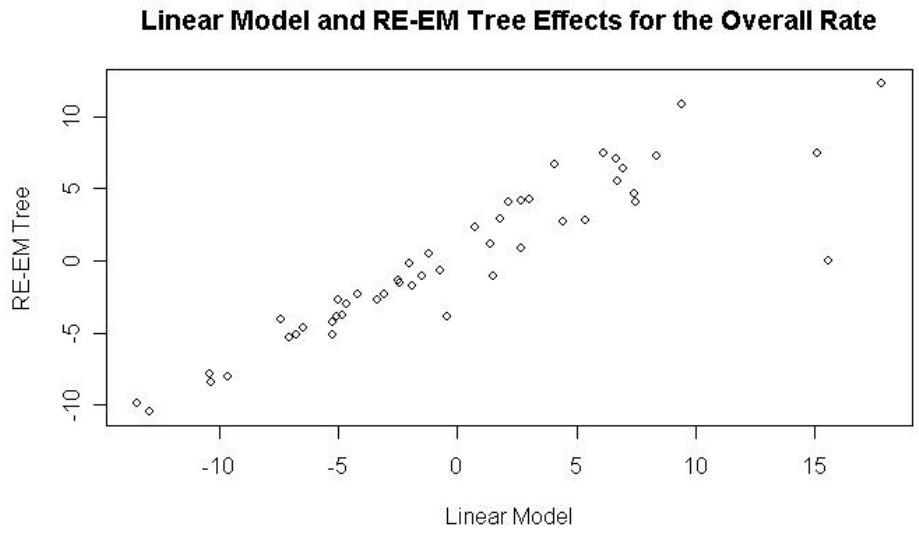

Figure 93: Scatterplot of random effects from the RE-EM tree without autocorrelation versus those from the linear random effects model for the overall traffic fatality rate.

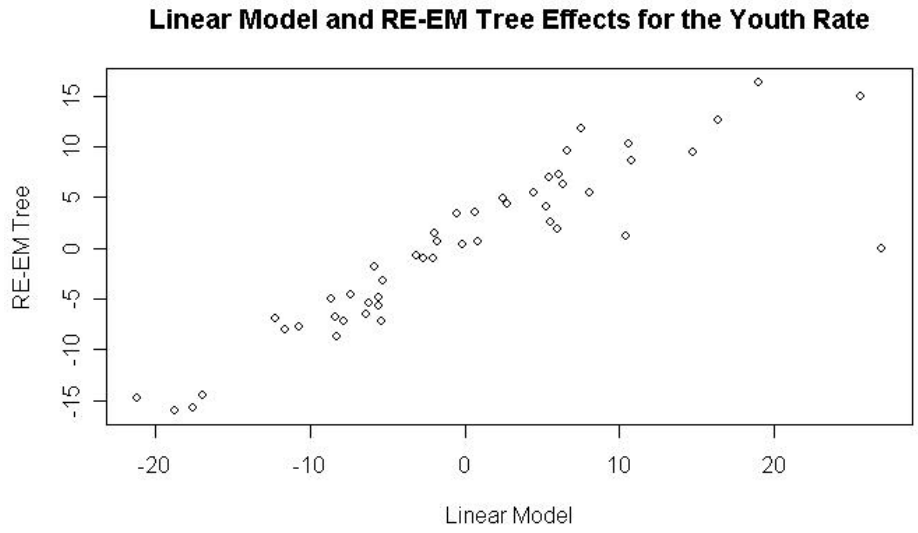

Figure 94: Scatterplot of random effects from the RE-EM tree without autocorrelation versus those from the linear random effects model for the youth traffic fatality rate.

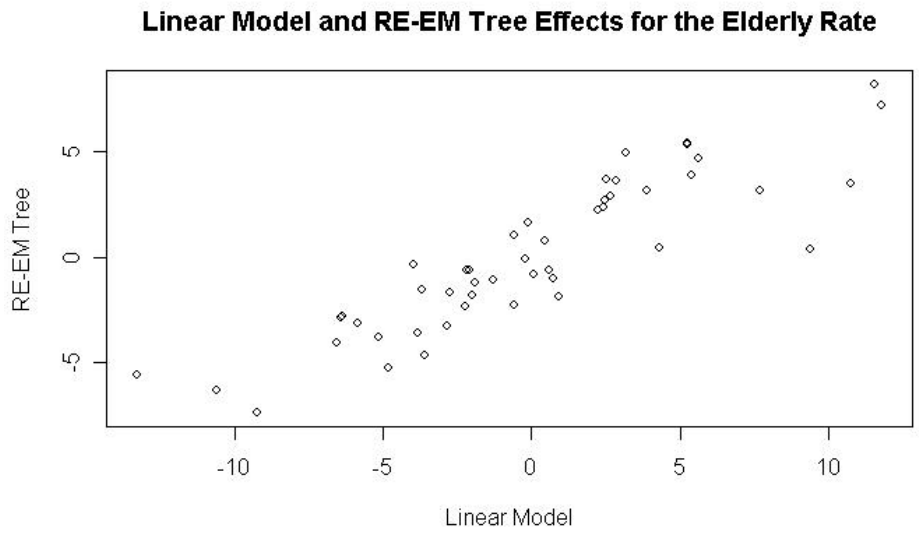

Figure 95: Scatterplot of random effects from the RE-EM tree without autocorrelation versus those from the linear random effects model for the traffic fatality rate for people aged 65 and older.

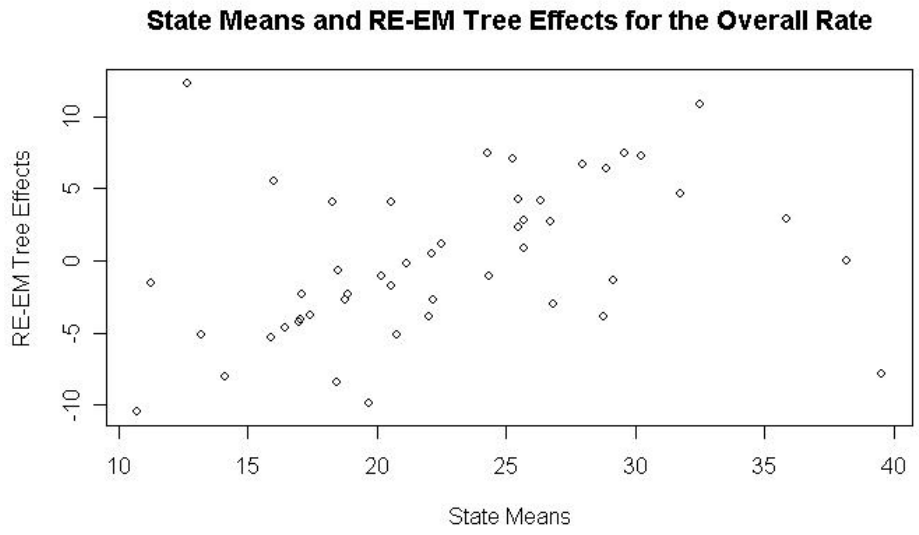

Figure 96: Scatterplot of state-specific mean overall fatality rates and random effects from the RE-EM tree without autocorrelation for the overall traffic fatality rate.

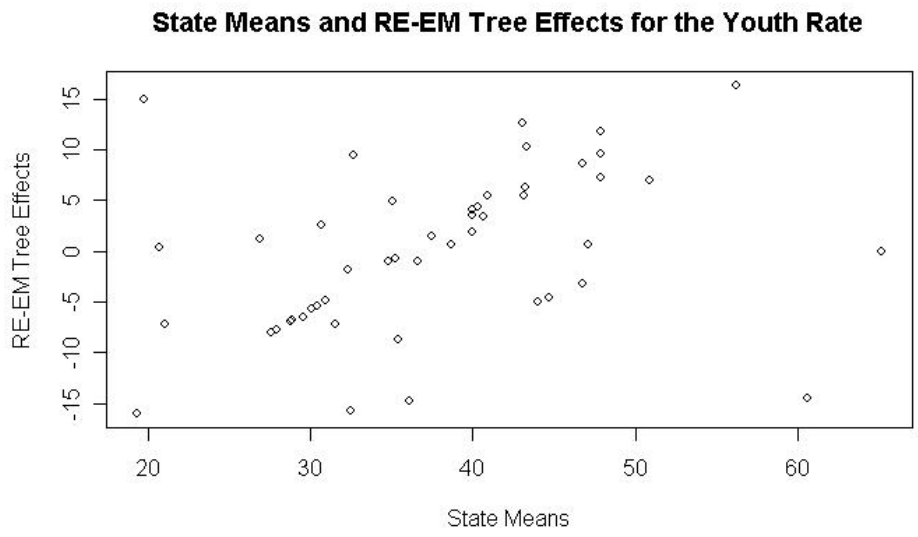

Figure 97: Scatterplot of state-specific mean youth fatality rates and random effects from the RE-EM tree without autocorrelation for the youth traffic fatality rate.

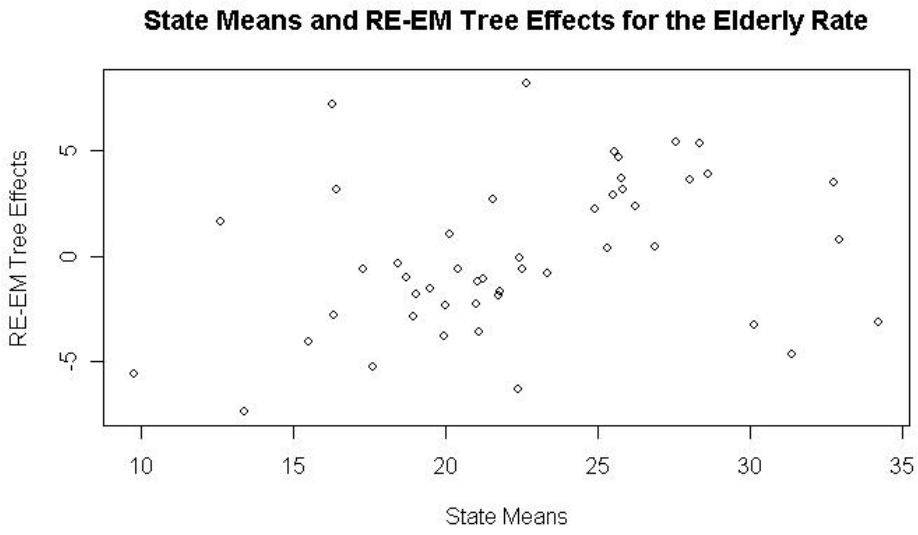

Figure 98: Scatterplot of state-specific mean elderly fatality rates and random effects from the RE-EM tree without autocorrelation for the elderly traffic fatality rate.

	Overall	Youth	Elderly
Linear Model	5.692	8.564	5.855
Linear Model with Random Effects	2.326	5.654	3.746
Linear Model with Random Effects and Autocorrelation	2.626	5.561	3.793
Regression Tree	5.425	8.971	5.186
RE-EM Tree	2.767	7.109	4.037
RE-EM Tree with Autocorrelation	2.630	7.005	3.990

Table 65: Root Mean Squared Error of Traffic Fatality Predictions. The root mean squared error is based on estimating a model through 1997 and using the estimated model to predict the observations for 1998 and 1999 for each state.

tree outperforms the tree without random effects for all three traffic fatality rates; paired signed rank tests show that the squared prediction errors are statistically significant in all three cases, though the differences for the youth and the elderly traffic fatality rates are only marginally significant ($p = 0.012$ and $p = 0.036$ respectively). The linear random effects model does slightly better than the RE-EM tree, with the difference in squared prediction errors statistically significant only for the youth traffic fatality rate. The linear model without random effects has significantly higher squared prediction error in all cases. This suggests that a linear model is reasonable in this case, as long as that model includes state-specific random effects. These results show the importance of accounting for state-specific effects in the traffic fatality rate and demonstrate that the nonparametric approach of a RE-EM tree can be as useful as a linear model, without the need to choose a structure beforehand.

In their original paper, Dee and Sela estimate the parametric models using the

logarithm of the traffic fatality rate instead of the level. The heteroskedasticity that we have observed also suggests that the logarithm of the traffic fatality rate will be more useful. Therefore, we repeat the out-of-sample experiment using logged traffic fatality rates. The resulting plots of residuals versus fitted values, given in Figures 99 and 100 for the two models, show that this operation has removed the heteroskedasticity. The root mean squared errors for the logged models are shown in Table 67. When logged traffic fatality rates are used, the linear model with random effects outperforms the RE-EM tree for the overall traffic fatality rate, and the difference is marginally statistically significant for youth traffic fatality rates ($p = 0.039$); the two models are statistically indistinguishable for the elderly traffic fatality rate. These results suggest that the models based on levels instead of logarithms were misspecified. When a model is correctly specified, we expect that the parametric model will perform at least as well as the RE-EM tree. However, the relative performances when we do not take logarithms shows that the RE-EM tree can perform well in a wide variety of situations, such as when the model is not correctly specified, and still generally outperforms a tree that does not incorporate random effects.

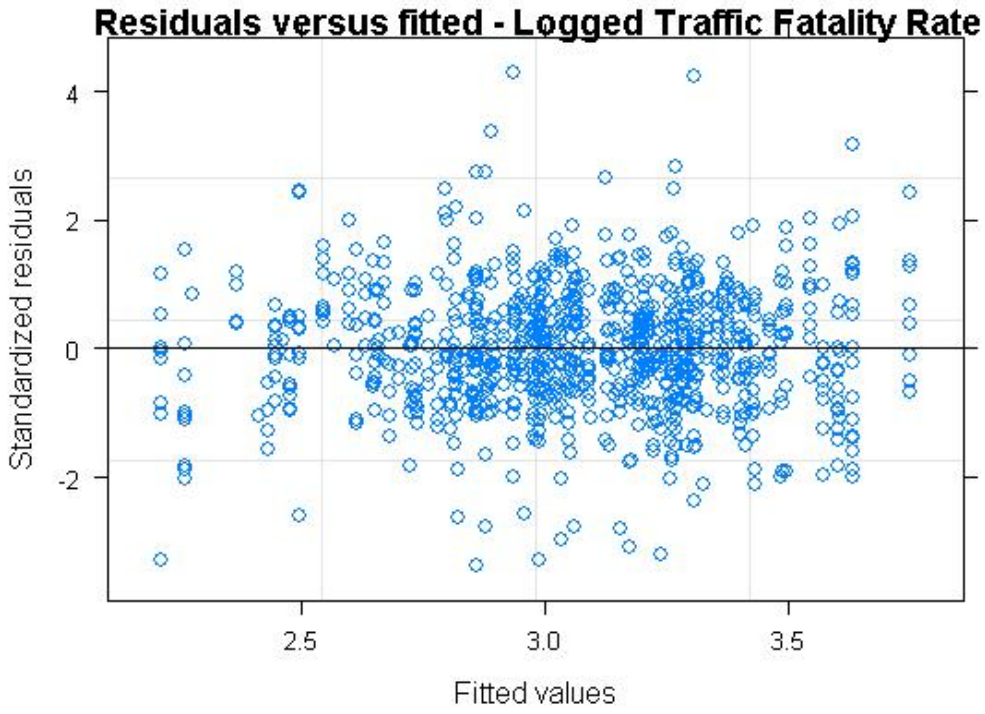

Figure 99: Plots of the residuals versus the fitted values from the RE-EM tree for the overall traffic fatality rate.

	Overall	Youth	Elderly
Linear Model	0.2755	0.2721	0.2773
Linear Model with Random Effects	0.0838	0.1242	0.1633
Linear Model with Random Effects and Autocorrelation	0.0858	0.1246	0.1640
Regression Tree	0.1895	0.1887	0.1964
RE-EM Tree	0.0863	0.1206	0.1543
RE-EM Tree with Autocorrelation	0.0872	0.1194	0.1581

Table 66: In-sample root mean squared error for traffic fatality data.

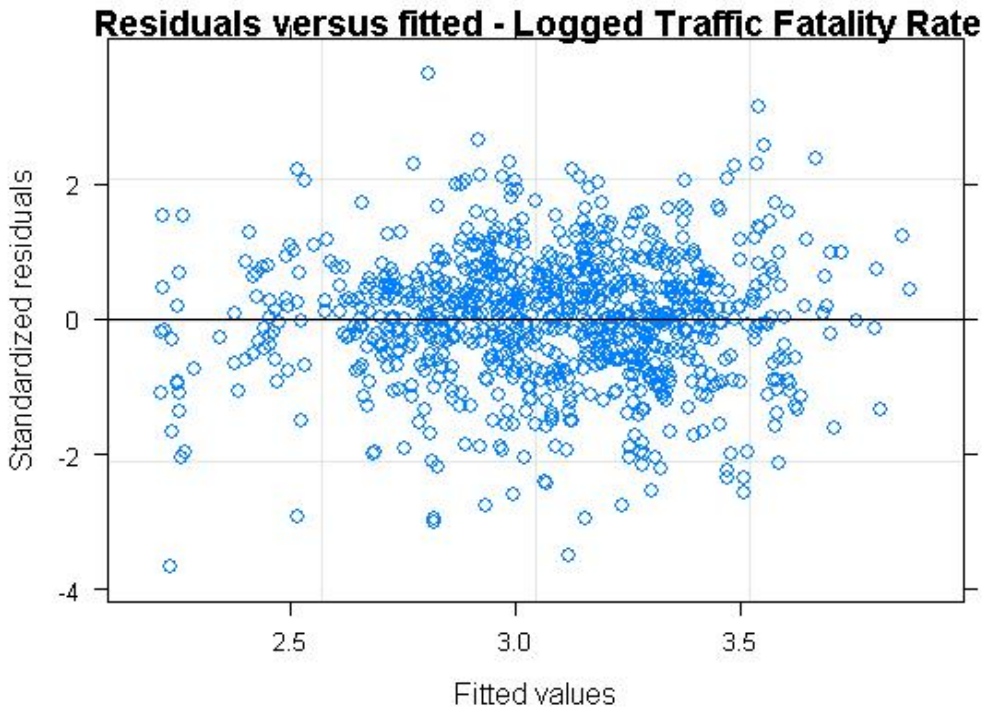

Figure 100: Plots of the residuals versus the fitted values from the linear model with random effects for the overall traffic fatality rate.

	Overall	Youth	Elderly
Linear Model	0.2771	0.2760	0.2482
Linear Model with Random Effects	0.1017	0.1573	0.1509
Linear Model with Random Effects and Autocorrelation	0.1151	0.1547	0.1520
Regression Tree	0.2512	0.2515	0.2051
RE-EM Tree	0.1407	0.2131	0.1775
RE-EM Tree with Autocorrelation	0.1267	0.2260	0.1669

Table 67: Root Mean Squared Error of Traffic Fatality Predictions. The root mean squared error is based on estimating a model through 1997 and using the estimated model to predict the observations for 1998 and 1999 for each state.

4.5 Application to Transactions Data

We now apply this method to a much larger dataset on third-party sellers on Amazon Web Services. (See Ghose et al. [2005] for background on this dataset and its first use.) Our data consist of 9484 transactions for 250 distinct software titles; thus, there are 250 individuals in the panel with a varying number of observations per individual. (While there are also some sellers who are included more than once, our longitudinal structure is based only on the products.) In this exercise, our target variable is the price premium that a seller can command; this is the difference between the price at which the good is sold and the average price of all of the competing goods in the marketplace. We also model the relative price premium, which is the ratio of those two quantities, and the logarithm of the relative price premium. Predictors include both the seller's own reputation and the characteristics of its competitors. The seller's reputation is measured by the total number of comments and the number of positive and negative comments received from buyers over different time periods. The length of time that the seller has been in the marketplace is also a predictor. Other predictors include the number of competitors, the quality of competing products, and the average reputation of the competitors, and the average prices of the competing products.

We first fit a tree without random effects and a RE-EM tree to the data. The estimated regression tree without random effects is shown in Figure 101, while the RE-EM tree is shown in Figure 102. The trees split on a variety of variables, and the structures of the two trees are quite different. Unlike in the previous example, the complexity of the two trees is similar; the RE-EM tree has 11 terminal nodes while the tree without random effects has 15. For this data, a RE-EM tree that allows for autocorrelation, shown in Figure 103, has the same structure as a RE-EM tree that does not allow for autocorrelation. Only one split differs; both trees split

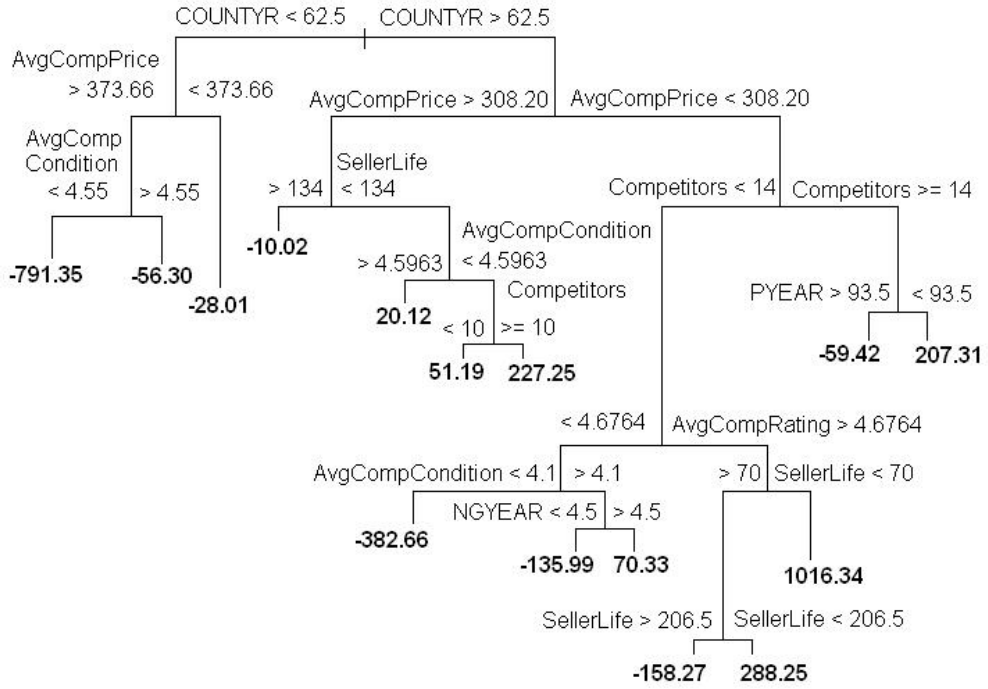

Figure 101: Estimated tree without random effects for the price premium in the transactions data.

on `SellerLife` but they split at different values. Because the effects models differ, the estimated values at the nodes differ slightly. The autocorrelation parameter is estimated to be 0.313 and the model without autocorrelation is strongly rejected ($p < 10^{-100}$ when we use either tree to compute the random effects model).

For comparison, we fit linear models with and without random effects. Because some of the predictors have missing values, we cannot fit linear models that include all of the possible predictors, even though both RE-EM trees and regression trees without random effects can handle predictors with missing values using surrogate split as discussed in section 4.2.2. In addition, some of the predictors are strongly collinear. Instead, we fit linear models that include all of the predictors that ap-

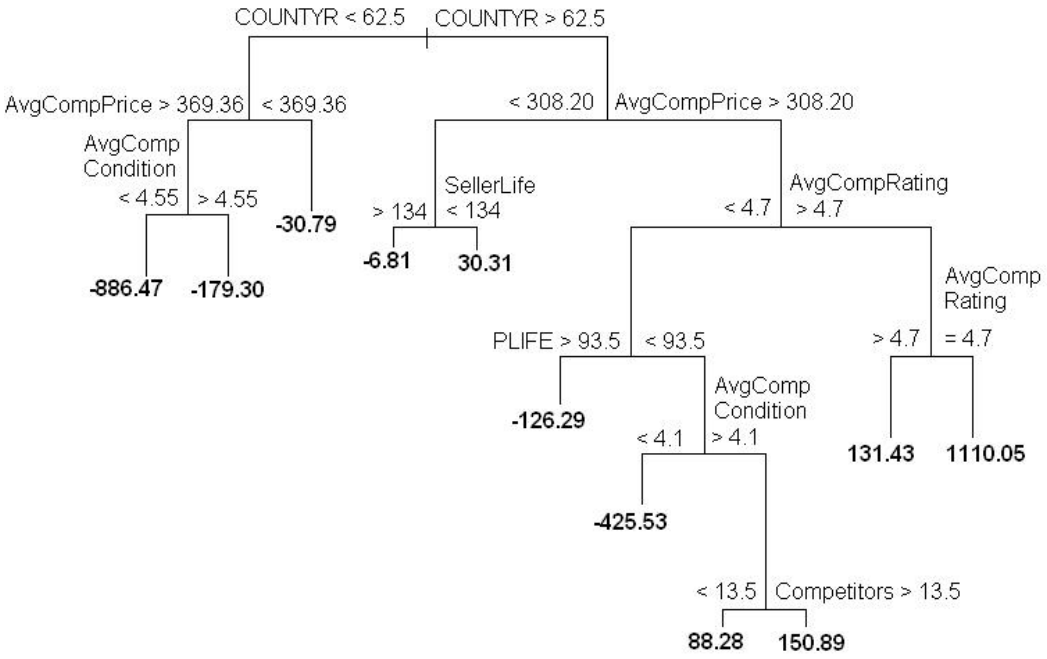

Figure 102: Estimated RE-EM tree for the price premium in the transactions data.

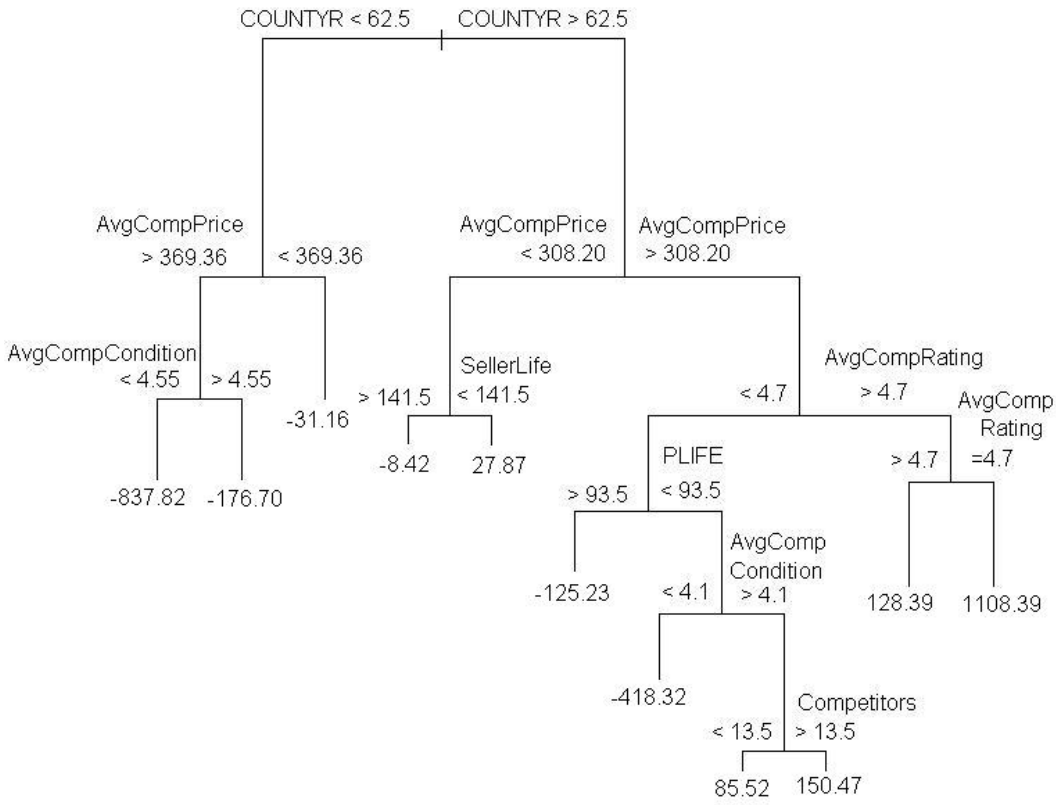

Figure 103: Estimated RE-EM tree with autocorrelation for the price premium in the transactions data.

pear in the RE-EM tree, since it happened that none of the predictors chosen for the RE-EM tree had missing values. The parameter estimates from these models are given in Table 68. Few variables are statistically significant in the linear model without random effects, while all of the variables are at least marginally statistically significant when random effects are included. Two of the variables that are statistically significant in the model without random effects, the average competitor price and the number of competitors, are statistically significant with the opposite signs when random effects are included. This underscores the importance of including random effects in the estimation of parameters. The average competitor price appears in the RE-EM tree twice; in one branch, lower competitor prices are associated with higher premiums, while in the other branch lower prices are associated with lower premiums. This ambiguous effect is impossible for a linear model without interactions to pick up and may explain why the coefficient changed sign from the linear model without random effects to the linear model with random effects.

We look at some diagnostic plots of the linear model with random effects and the RE-EM tree, to check whether our assumptions hold. Figure 104 plots the residuals against the fitted values from the RE-EM tree. This plot highlights some observations with particularly low fitted values; there seems to be more variability about that fitted value. There is also a single observation with a large fitted value and a zero residual. This observation corresponds to the right-most branch of the RE-EM tree. The same pattern appears in the corresponding plot for the linear model with random effects, in Figure 105. Certain titles seem to have larger variance than others, as shown in Figures 106 and 107. Quantile-quantile plots for the two models, shown in Figures 108 and 109, show that the distribution of residuals is highly non-normal, with very fat tails. Thus, the usual parametric

Variable	Linear Model	Random Effects Model	Random Effects - AR(1)
(Intercept)	88.800**	501.756***	330.62***
	(34.895)	(52.742)	(44.60)
Average Competitor Price	0.064***	-1.654***	-1.367***
	(0.004)	(0.031)	(0.027)
Average Condition of Competing Goods	-0.218	12.231*	14.760**
	(4.943)	(7.323)	(6.292)
Average Rating of Competitors	7.168	-22.043***	-17.078***
	(4.764)	(6.044)	(4.985)
Life of the Seller	0.001	0.002**	0.001*
	(0.001)	(0.001)	(0.0006)
Number of Competitors	2.115***	-1.099***	-0.864**
	(0.160)	(0.418)	(0.345)
Lifetime Positive Comments	-1.659***	-1.615***	-0.661***
	(0.099)	(0.084)	(0.084)
Number of Comments in the Last Year	-0.001	-0.002*	-0.0015
	(0.001)	(0.001)	(0.001)

Table 68: Parameter estimates for the linear models for the price premium with and without random effects. Standard errors are reported in parentheses. * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

Figure 104: Plot of residuals versus fitted values from the estimated RE-EM tree for the price premium in the transactions data.

assumptions required for the linear model fail. Because of this, we conclude that we must consider alternative functional forms of the target variable for the linear model to be useful (as we will do later in this section) or that we must abandon the linear model altogether.

The autocorrelation functions for the two models without autocorrelation are given in Figures 110 and 111. These plots show unmodeled autocorrelation in the residuals; the autocorrelation is slightly larger in the linear model. The same plots when the models include autocorrelation are given in Figures 112 and 113; these plots show that the autocorrelation has been removed, but this has led to some negative autocorrelation at the first lag.

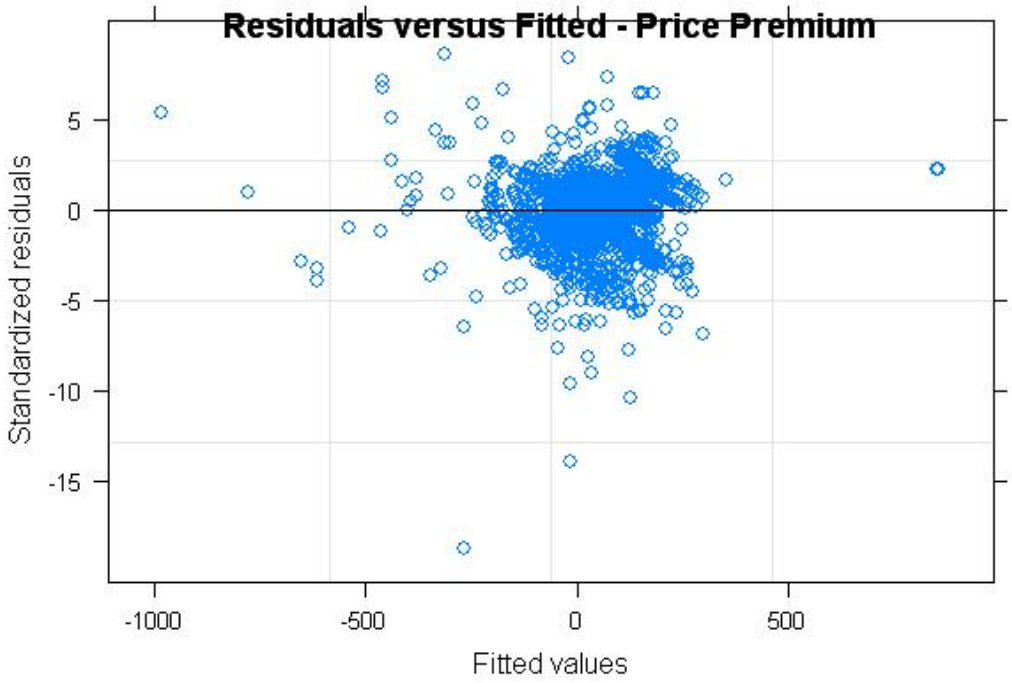

Figure 105: Plot of residuals versus fitted values from the estimated linear random effects model for the price premium in the transactions data.

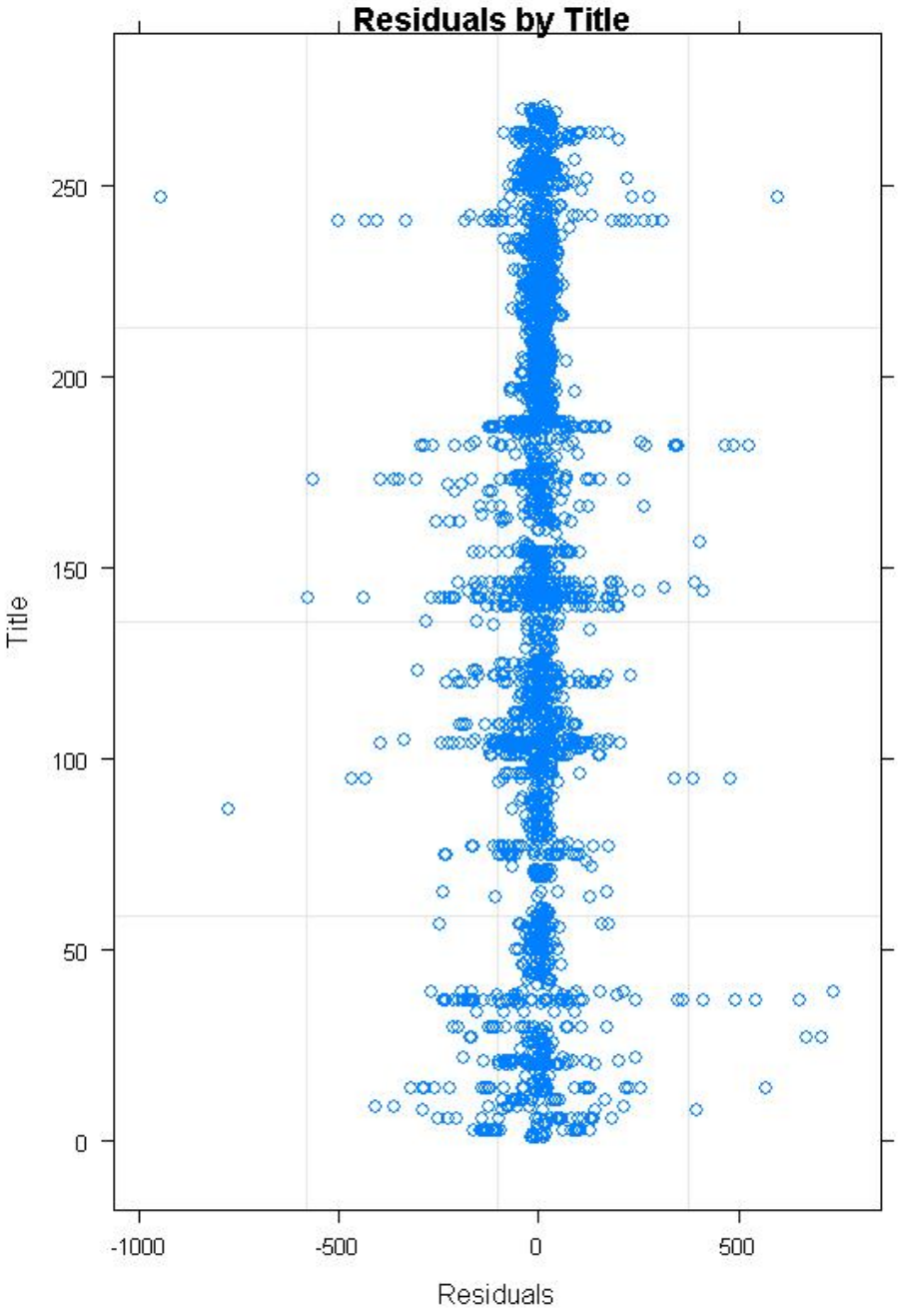

Figure 106: Plot of residuals for each software title from the estimated RE-EM tree for the price premium in the transactions data.

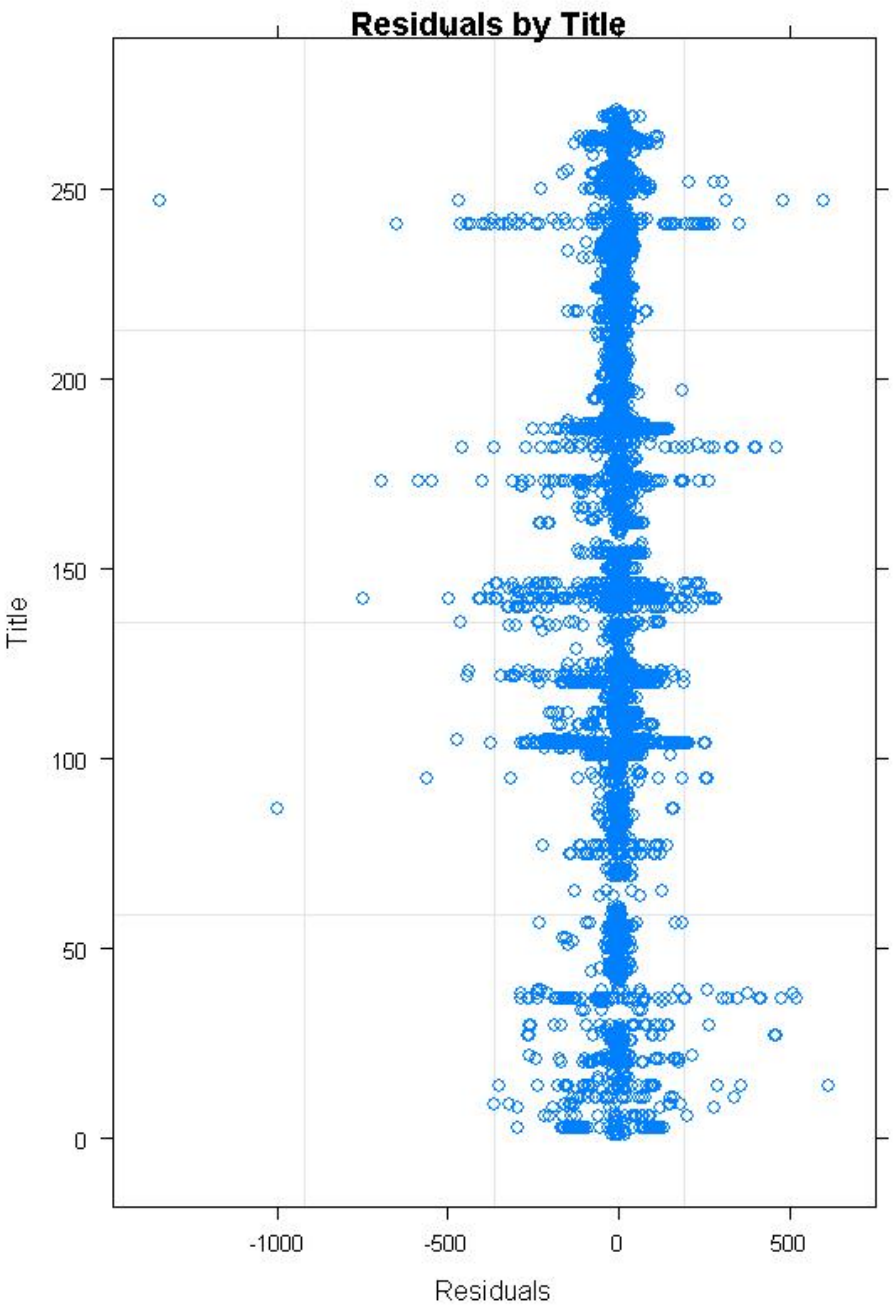

Figure 107: Plot of residuals for each software title from the estimated linear random effects model for the price premium in the transactions data.

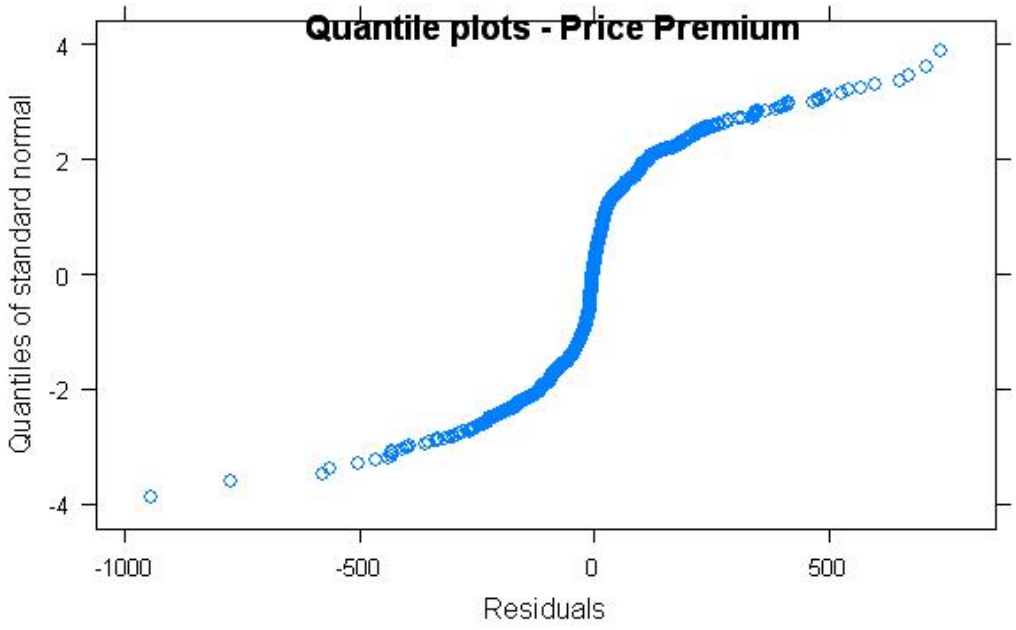

Figure 108: Quantile-quantile plot of residuals from the estimated RE-EM tree for the price premium in the transactions data.

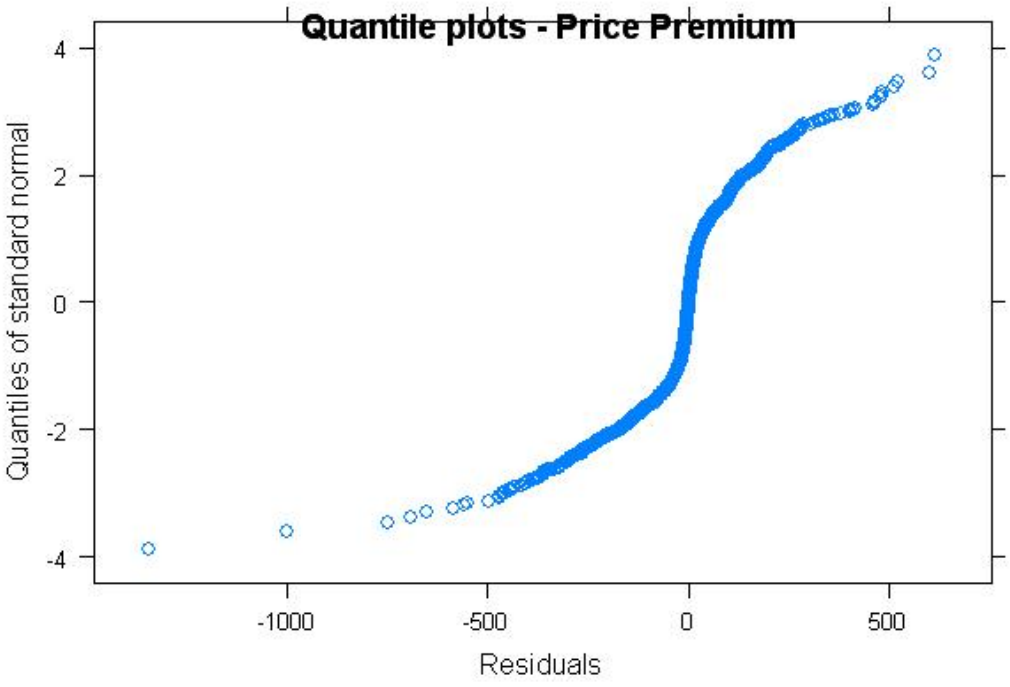

Figure 109: Quantile-quantile plot of residuals from the estimated linear random effects model for the price premium in the transactions data.

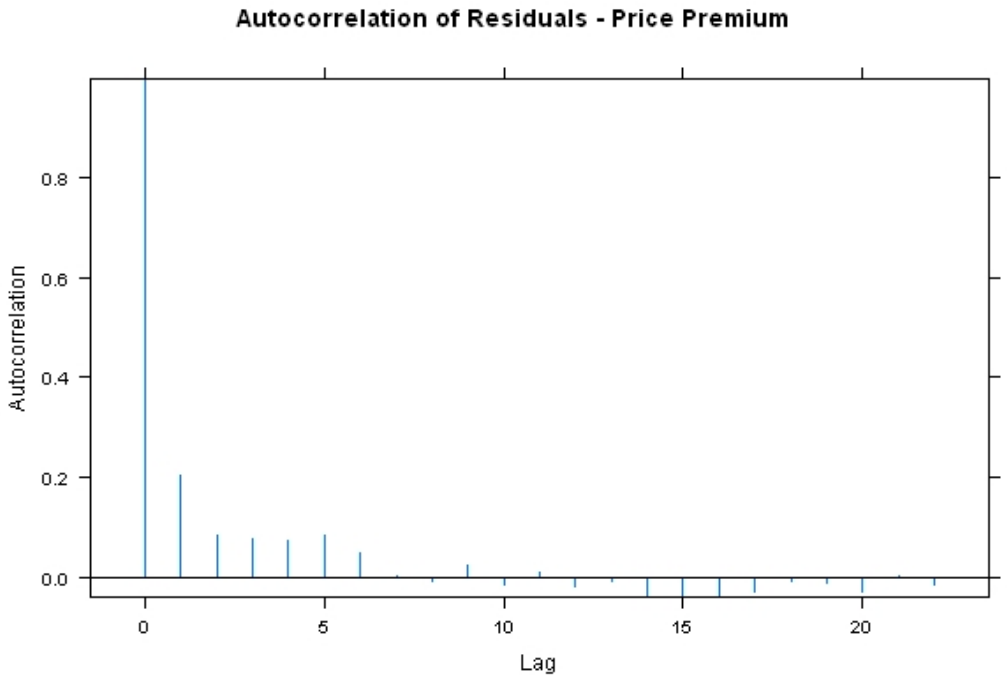

Figure 110: Autocorrelation function of the residuals from the estimated RE-EM tree without autocorrelation for the price premium in the transactions data.

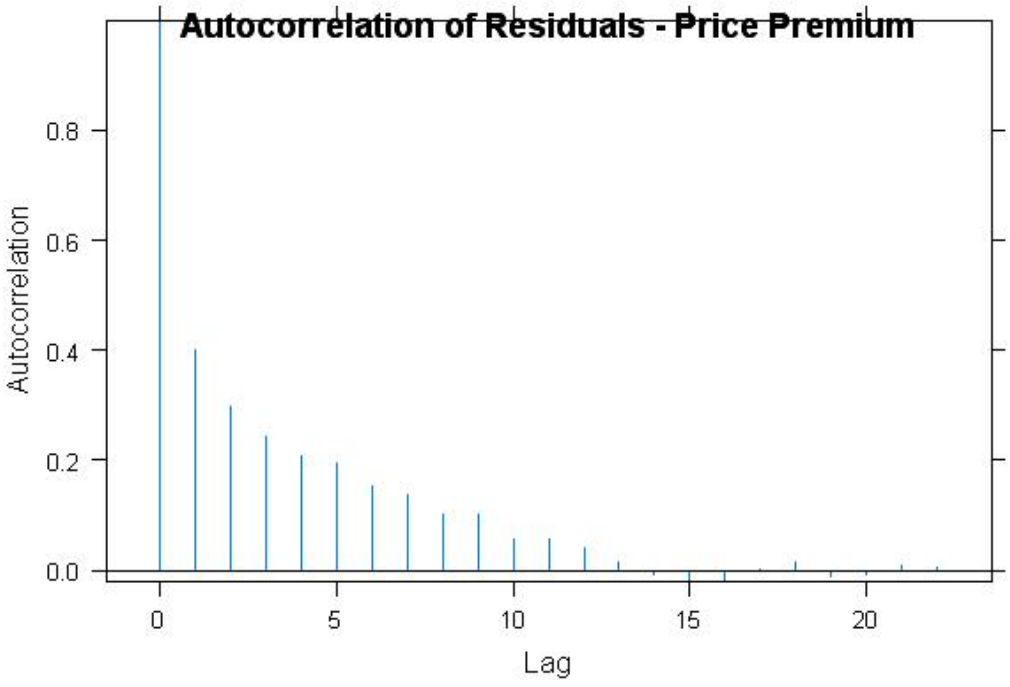

Figure 111: Autocorrelation function of the residuals from the estimated linear random effects model without autocorrelation for the price premium in the transactions data.

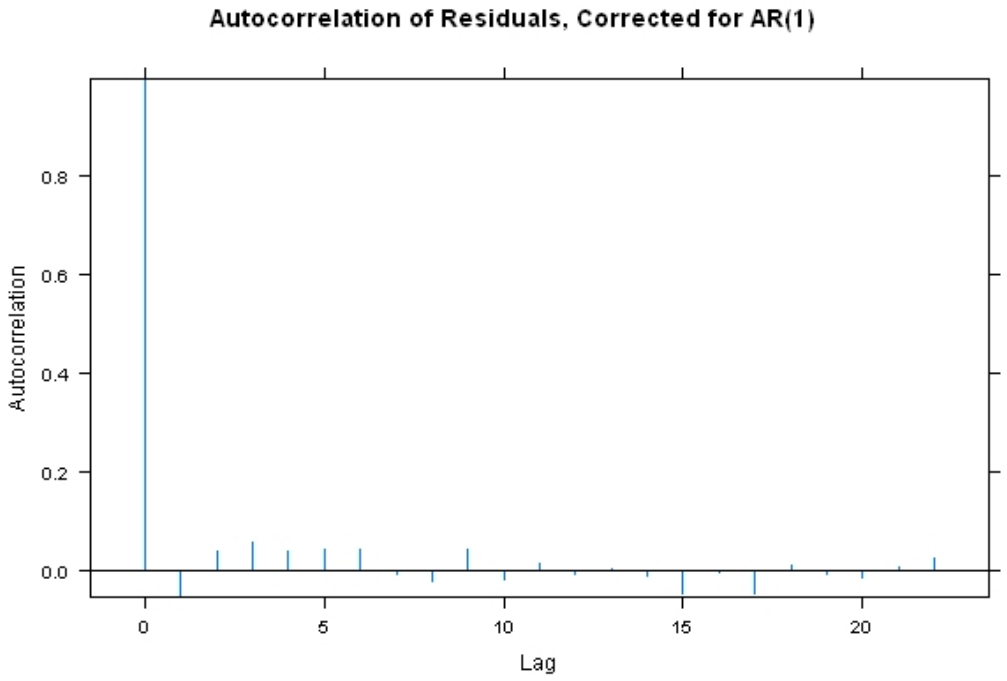

Figure 112: Autocorrelation function of the residuals from the estimated RE-EM tree with autocorrelation for the price premium in the transactions data.

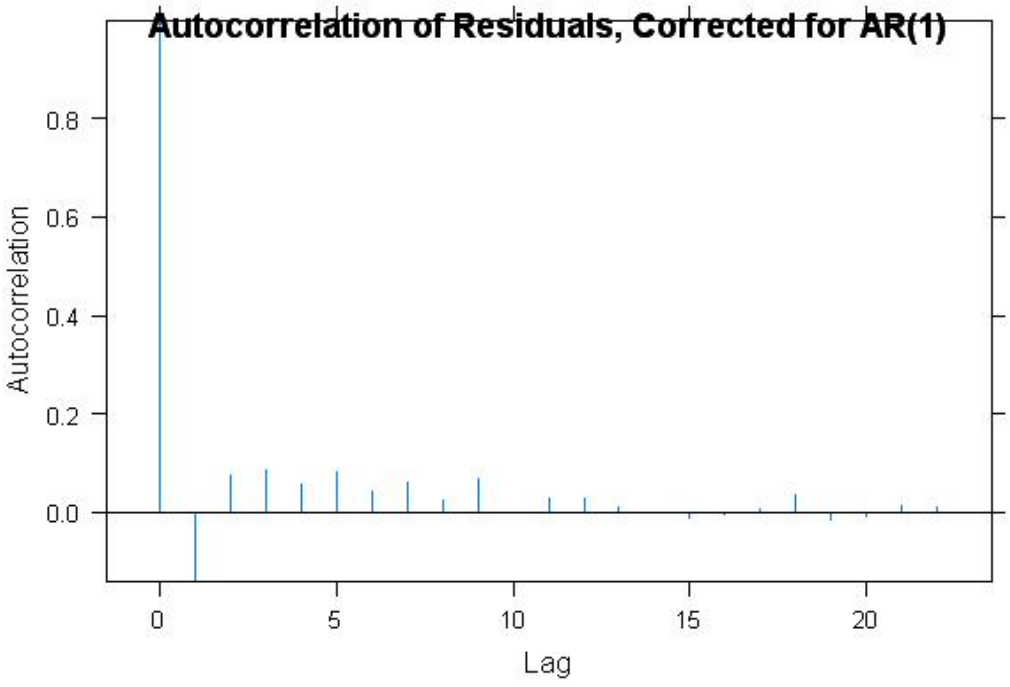

Figure 113: Autocorrelation function of the residuals from the estimated linear random effects model with autocorrelation for the price premium in the transactions data.

Change	RE-EM Tree	Linear Effects Model
Omit	33.53	76.13
Change to 750	7.14	58.92
Change to 1300	2.71	60.57

Table 69: Root mean squared difference between the fitted values using the original data and the fitted values with the influential observation modified.

In the linear model, the observations with the largest fitted value are likely to be influential. In this dataset, nine observations have the target value equal to 1016.34; the next largest target is 647.1. In the RE-EM tree, these observations are partitioned off by themselves, which means that changing their value will have a smaller effect on the estimated values for the other observations, as long as they stay in a separate node. Because the coefficients from the linear random effects model depend on all of the observations, the estimated values for other observations will be more strongly influenced by the target values for influential observations. We can quantify the effect of the influential observations by omitting them or by changing their target value, re-estimating each model, and measuring the changes in the fitted values of the other observations. The results are shown in Table 69. In every case, the fitted values for the other observations change less when we use a RE-EM tree than when we use a linear random effects model. The change for the RE-EM tree is particularly small when we increase the target value of the influential observations. Thus, using a regression tree instead of a linear model helps to mitigate the effect of influential observations.

We use the tree models and the linear models to compute three different types of root mean squared errors; all are reported in Table 70. First, we compute the RMSE of the fitted values when the tree is fit to the complete sample. Because of

the complex nature of the data, the trees have lower in-sample root mean squared errors. Second, we use leave-one-out cross-validation to measure out-of-sample prediction performance. To measure the performance when a random effect can be estimated, we exclude one transaction (observation) at a time, allowing the tree to estimate a random effect corresponding to an observation based on the other observations for that individual. To measure the performance for new individuals, we repeat the leave-one-out cross-validation by excluding all of the observations for a single individual at each iteration. For each type of cross-validation, we measure performance by the root mean square error of prediction for the omitted observation(s). In-sample and when single observations are excluded, the linear model not including random effects has the largest root mean squared error, while the RE-EM tree has the smallest RMSE. When all of the observations for an individual are excluded, the linear model with random effects performs much worse. We will see this behavior again in our Monte Carlo experiments in Section 4.6 and will discuss it there. Again, the RE-EM tree performs best, though its RMSE is not very different from the RMSE of a regression tree without random effects.

Given the observed non-normality and heteroskedasticity, we repeat this analysis using the relative price premium, which is the sale price divided by the average price of the competing products. The fitted trees with and without random effects are given in Figures 114, 115, and 116. The RE-EM tree without autocorrelation splits primarily on variables describing the amount of feedback. The tree without random effects and the RE-EM tree with autocorrelation look similar but differ greatly from the RE-EM tree without autocorrelation. These two trees split less frequently on the feedback variables and more on the characteristics of the competing products, such as the number of competitors and the average price and condition of competing products.

Model	In-sample	Excluding Observations	Excluding Titles
Linear Model	95.71	95.88	96.92
Linear Model with Random Effects	70.90	73.62	461.48
Linear Model with Random Effects - AR(1)	72.21	74.75	387.18
Tree without Random Effects	66.08	69.66	89.38
RE-EM Tree	58.48	64.54	88.53
RE-EM Tree - AR(1)	58.90	63.88	87.90
FE-EM Tree	58.64	65.67	91.10

Table 70: In-sample root mean squared errors and root mean squared errors from cross-validation leaving out one observation or one software title at a time, using the transactions data, using the price premium.

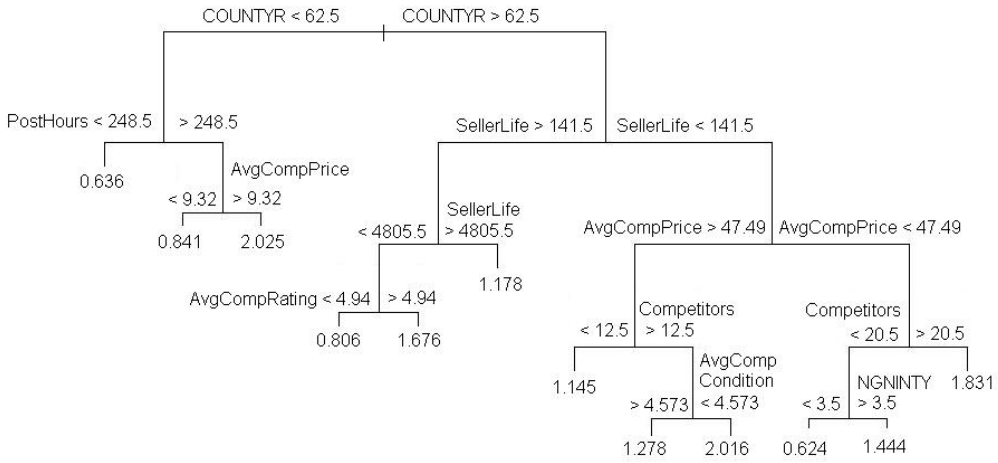

Figure 114: Estimated tree without random effects for the relative price premium in the transactions data.

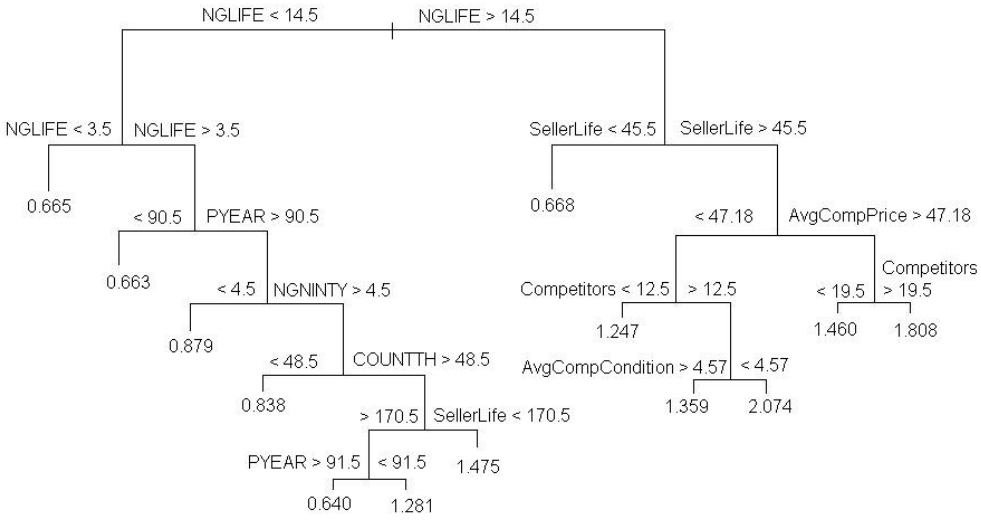

Figure 115: Estimated RE-EM tree for the relative price premium in the transactions data.

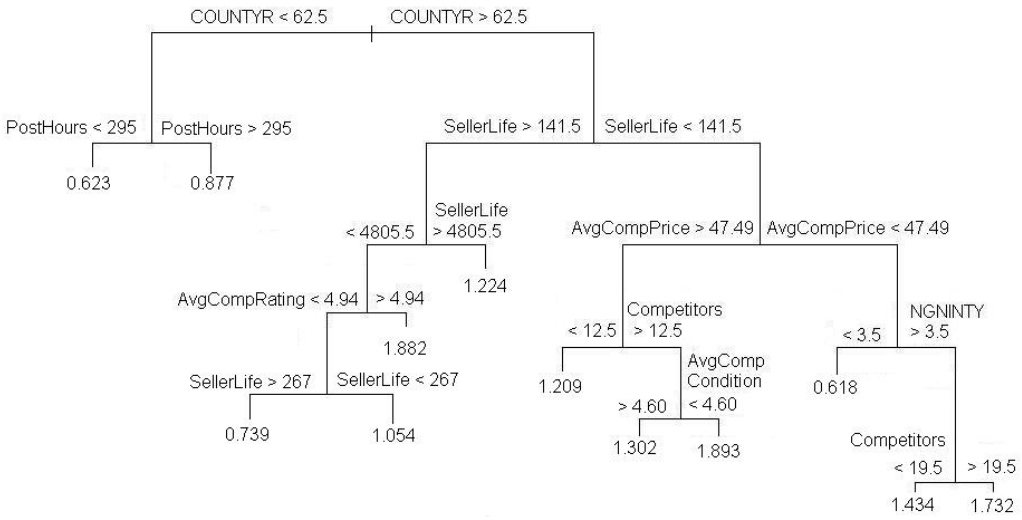

Figure 116: Estimated RE-EM tree with autocorrelation for the relative price premium in the transactions data.

For this target variable, some of the variables used in the trees have missing values. Therefore, we cannot use those variables in the linear models. Instead, we consider one model that uses all of the variables with no missing values chosen by any of the trees for the price premium, relative price premium and another that uses only the variables with no missing values chosen by the trees based on the relative price premium. Based on leave-one-out cross-validation, we choose to use the larger model. The coefficients for the estimated linear models are given in Table 72. These linear models have a number of variables which are estimated to be statistically significant. The coefficients on the number of lifetime positive comments, the number of negative comments in the last year, and the average competitor price are significantly less than zero in all three models. The sign on the number of lifetime positive comments is unexpected; it seems more likely that positive comments would lead to the ability to command higher price premiums. The average competitor price appears in all three estimated trees as well; however, it has a positive relationship with the relative price premium in the RE-EM tree without autocorrelation, a negative relationship in the RE-EM tree with autocorrelation, and positive and negative relationships in different branches in the tree without random effects. Other predictors, such as the average rating of competitors and the number of hours posted, are statistically significant in more than one model but take on different signs in the different models. These results show the difficulty of fitting a linear model to a complicated dataset.

The root mean squared errors of in-sample fits and leave-one-out cross-validation experiments are given in Table 73. As before, the RE-EM tree has the lowest root mean squared error in all three cases while the linear model with random effects has a very high RMSE in the case where titles are excluded. As before, the tree without random effects outperforms all three linear models, again suggesting that

Variable	Linear Model	Random Effects Model	Random Effects - AR(1)
(Intercept)	2.431***	3.813***	2.479***
	(0.168)	(0.211)	(0.144)
Average Rating of Competitors	0.086***	-0.070***	4.024E-3
	(0.018)	(0.026)	(0.016)
Number of Lifetime Positive Comments	-0.020***	-0.016***	-3.676E-3***
	(0.0001)	(0.001)	(7.427E-4)
Number of Comments in the Last Year	9.080E-6	1.8E-5**	5.1E-6
	(9.132E-6)	(8.58E-6)	(6.49E-6)
Hours Posted	-1.941E-4***	-1.51E-4***	1.017E-4***
	(2.177E-5)	(2.069E-5)	(1.391E-5)
Number of Negative Comments in the Last Year	-7.550E-5***	-8.4E-5***	-3.99E-5***
	(2.254E-3)	(2.073E-5)	(1.392E-5)
Number of Lifetime Negative Comments	-8.184E-3***	-5.023E-3***	-1.110E-3
	(1.367E-3)	(1.251E-3)	(8.394E-4)

Table 71: Parameter estimates for the linear models for the relative price premium with and without random effects. Standard errors are reported in parentheses. * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

Variable	Linear Model	Random Effects Model	Random Effects - AR(1)
Number of Comments in the Last Thirty Days	8.349E-5**	7.2E-5*	7.05E-5***
	(4.046E-5)	(3.707E-5)	(2.433E-5)
Seller Life	6.569E-6	4E-6	1E-7
	(4.029E-6)	(3.72E-6)	(2.60E-6)
Average Competitor Price	-1.204E-4***	-1.326E-3***	-1.605E-3***
	(1.440E-5)	(8.821E-5)	(7.818E-5)
Number of Competitors	4.390E-3***	3.347E-3*	-3.390E-4
	(5.878E-4)	(1.719E-3)	(1.115E-3)
Average Condition of Competing Goods	-3.953E-4	-0.159***	-0.176***
	(0.018)	(0.030)	(0.021)

Table 72: Parameter estimates for the linear models for the relative price premium with and without random effects (continued). Standard errors are reported in parentheses. * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

Model	In-sample	Excluding Observations	Excluding Titles
Linear Model	0.3517	0.3533	0.3560
Linear Model with Random Effects	0.3066	0.3219	0.4874
Linear Model with Random Effects - AR(1)	0.3250	0.3361	0.5692
Tree without Random Effects	0.2551	0.2676	0.2987
RE-EM Tree	0.2109	0.2270	0.2968
RE-EM Tree - AR(1)	0.2208	0.2390	0.2927
FE-EM Tree	0.2102	0.2267	0.3016

Table 73: In-sample root mean squared errors and root mean squared errors from cross-validation leaving out one observation or one software title at a time, using the transactions data, using the relative price premium.

a linear model is not appropriate for this dataset.

We also consider diagnostic plots. Plots of the fitted values versus the residuals (Figures 117 and 118) show that using the relative price premium removes the group of observations that are well below the bulk of the observations. However, a few observations with fitted values well above the others remain. Also, because the relative price premium cannot be negative, the residuals cannot go below the negative of the fitted value, leading to the lower corner of the plot of the residuals versus fitted values having no values. This pattern is particularly evident for the linear random effects model. Figures 119 and 120 plot the residuals by title for the RE-EM tree and the linear random effects model respectively. The variability of residuals by title has been reduced by using the relative price premium, par-

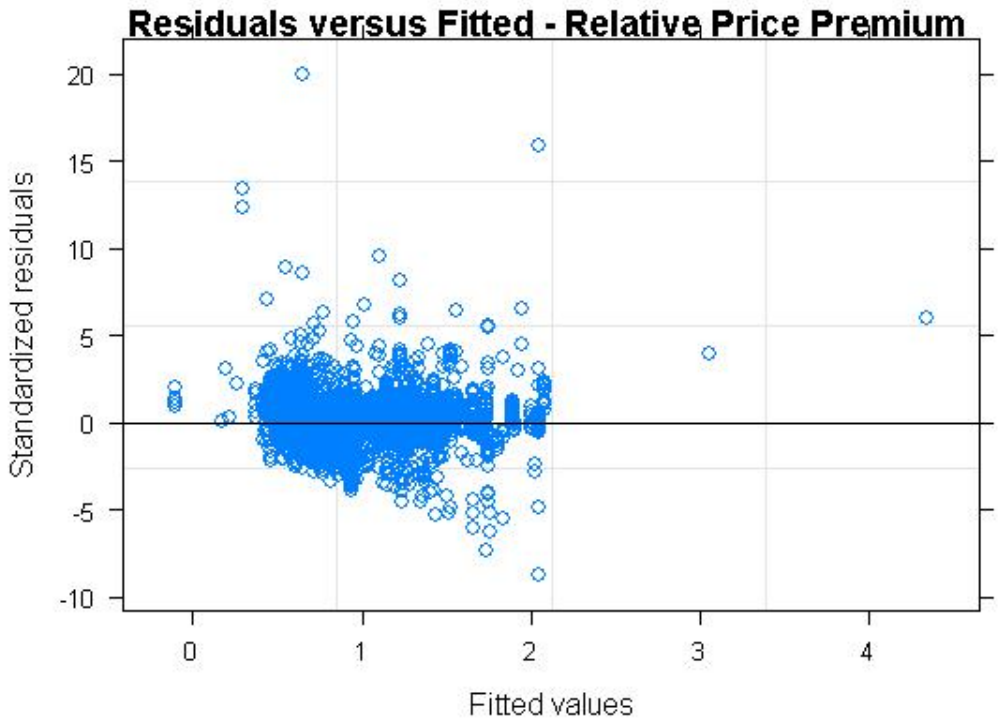

Figure 117: Plot of residuals versus fitted values from the estimated RE-EM tree for the transactions data fit to the relative price premium.

ticularly for the RE-EM tree. Quantile-quantile plots (Figures 121 and 122) show that the distribution of residuals continues to be non-normal, particularly because of outliers. The autocorrelation functions of the two models that do not model autocorrelation (Figures 123 and 124) show that there is autocorrelation in the residuals. Fitting the two models and allowing for autocorrelation removes the autocorrelation, but again induces a slight negative autocorrelation, as we see in Figures 125 and 126.

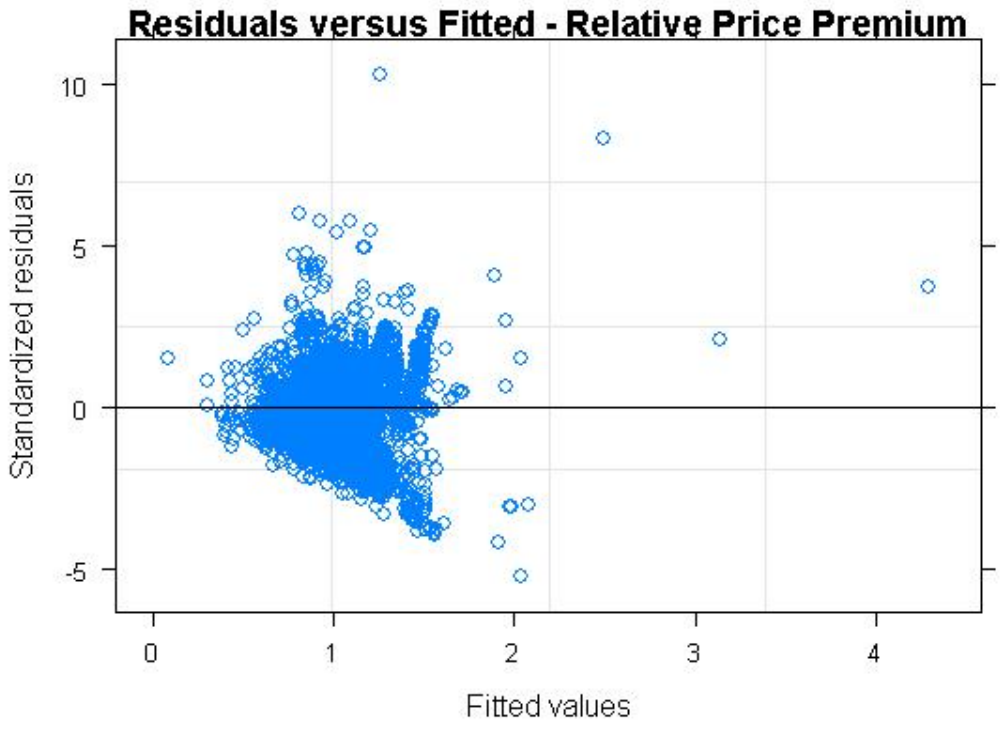

Figure 118: Plot of residuals versus fitted values from the estimated linear random effects model for the transactions data fit to the relative price premium.

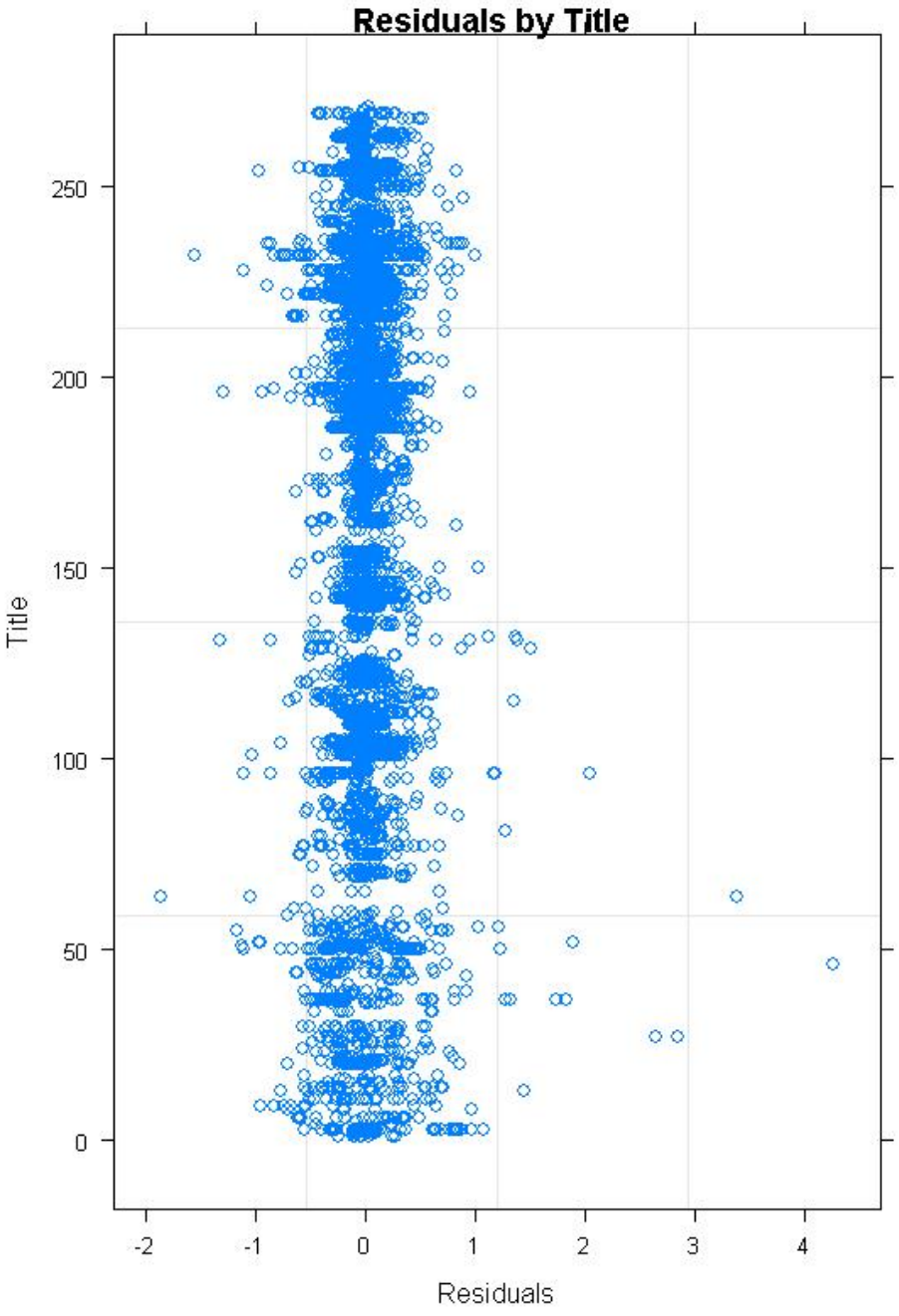

Figure 119: Plot of residuals for each software title from the estimated RE-EM tree for the transactions data fit to the relative price premium.

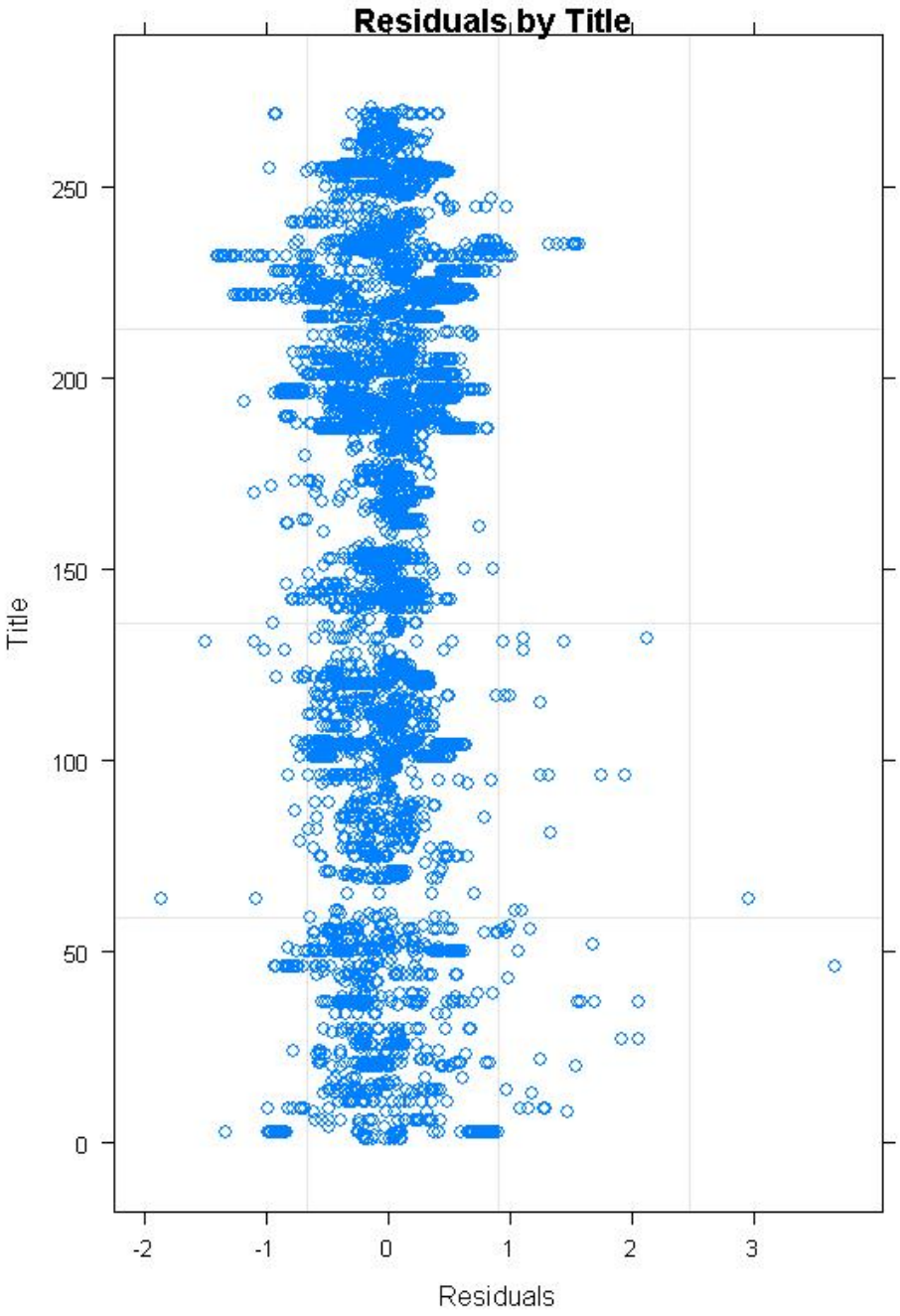

Figure 120: Plot of residuals for each software title from the estimated linear random effects model for the transactions data fit to the relative price premium.

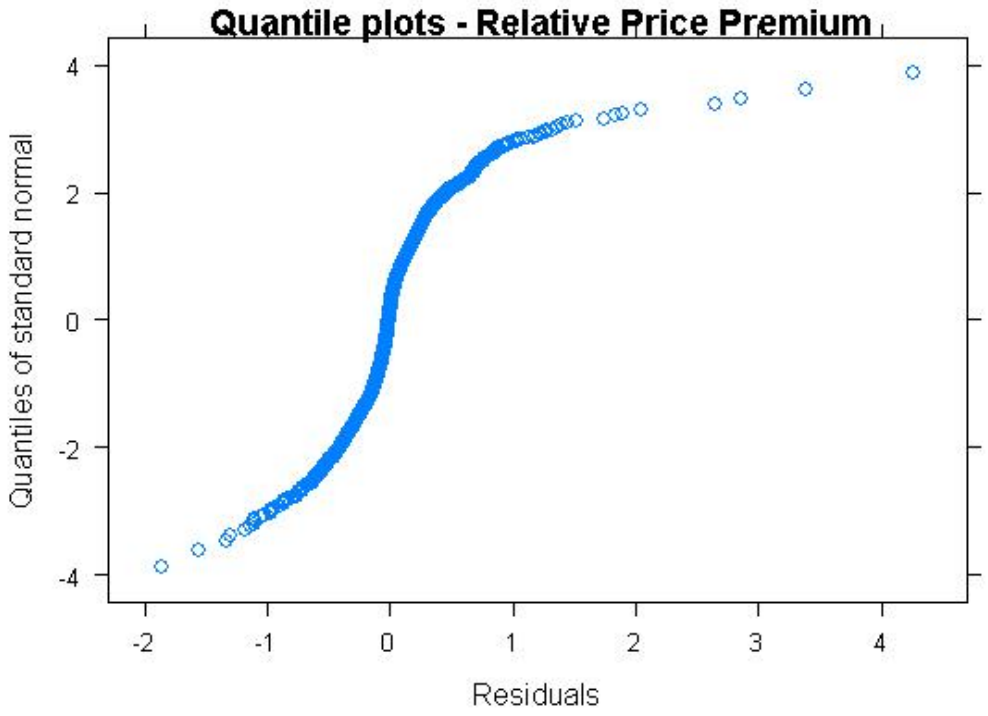

Figure 121: Quantile-quantile plot of residuals from the estimated RE-EM tree for the transactions data fit to the relative price premium.

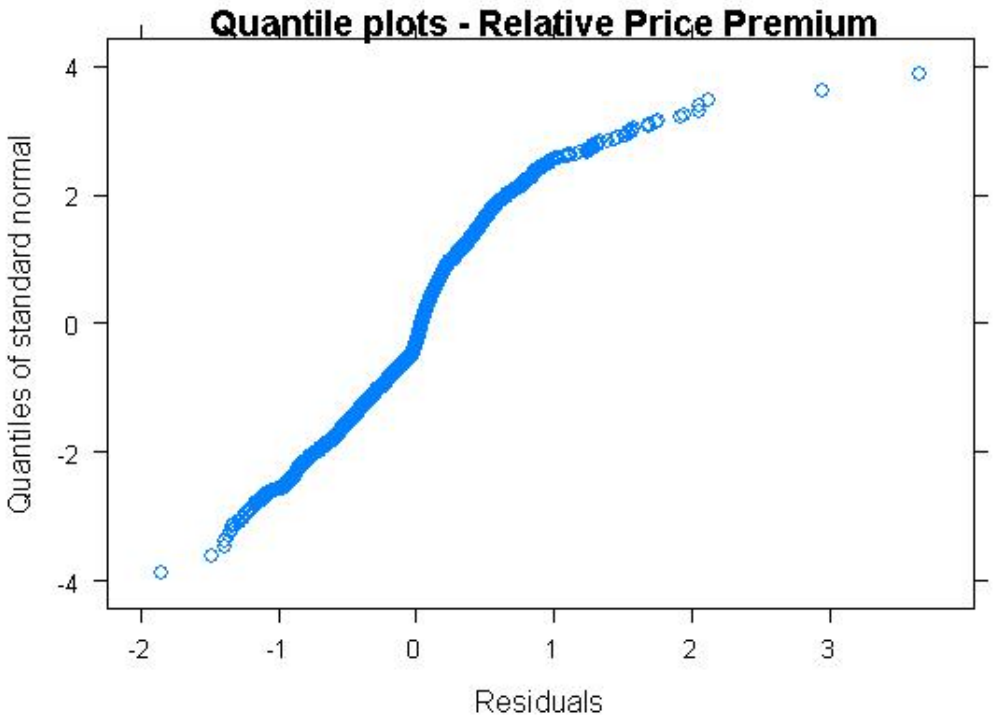

Figure 122: Quantile-quantile plot of residuals from the estimated linear random effects model for the transactions data fit to the relative price premium.

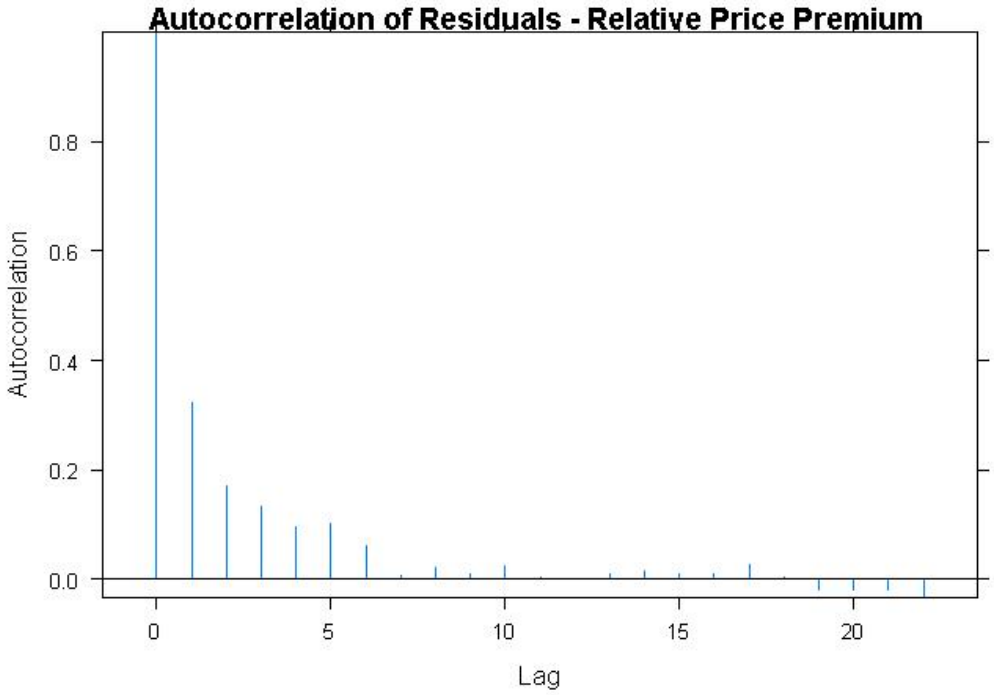

Figure 123: Autocorrelation function of the residuals from the estimated RE-EM tree without autocorrelation for the transactions data fit to the relative price premium.

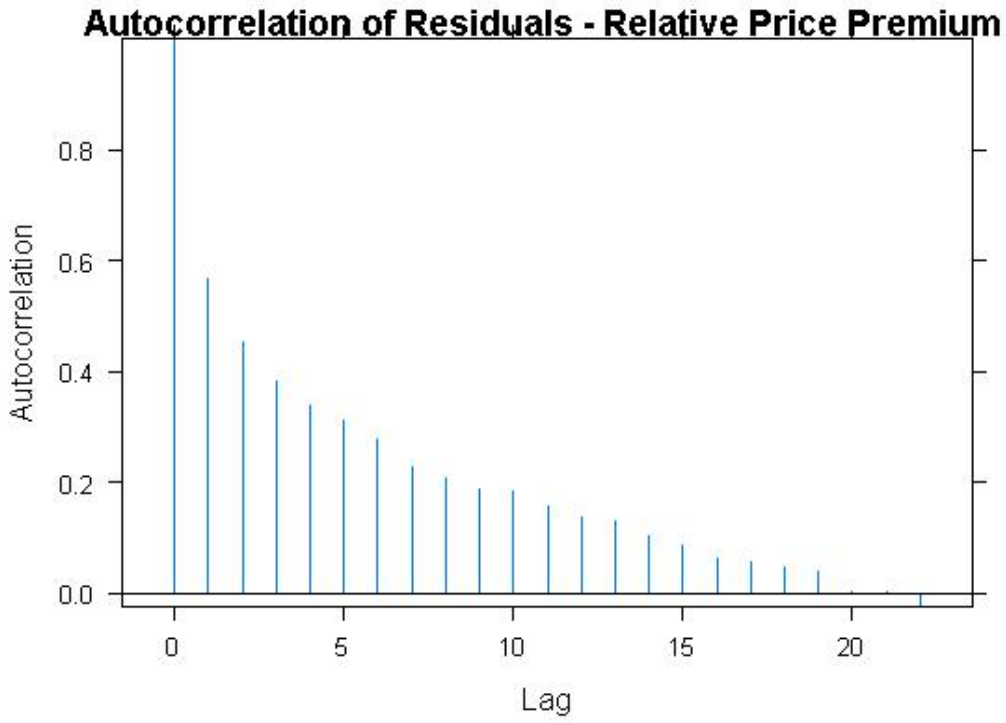

Figure 124: Autocorrelation function of the residuals from the estimated linear random effects model without autocorrelation for the transactions data fit to the relative price premium.

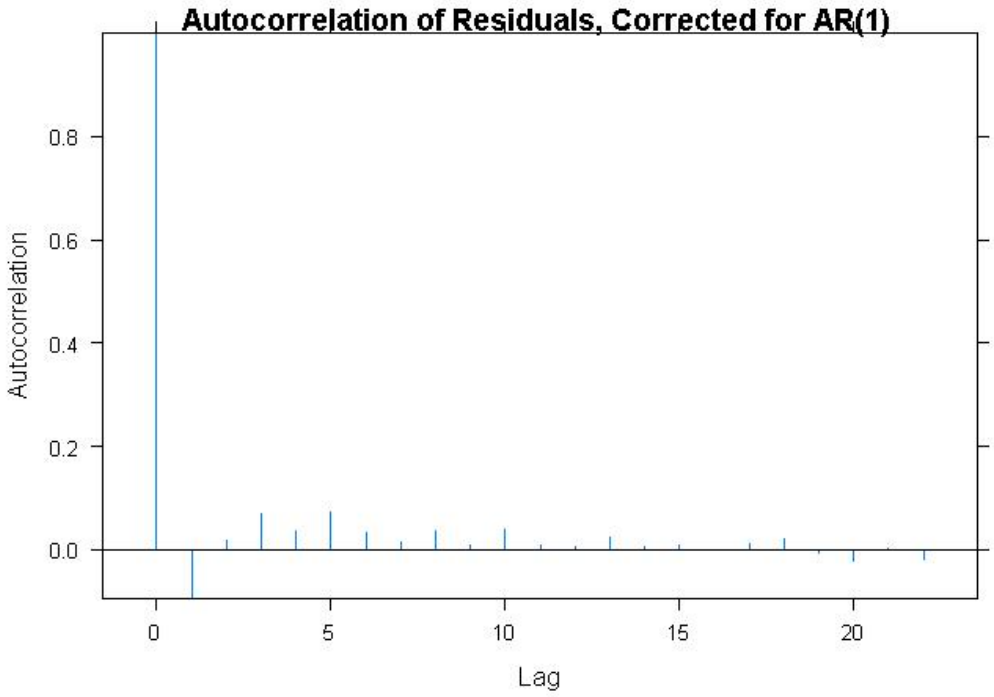

Figure 125: Autocorrelation function of the residuals from the estimated RE-EM tree with autocorrelation for the transactions data fit to the relative price premium.

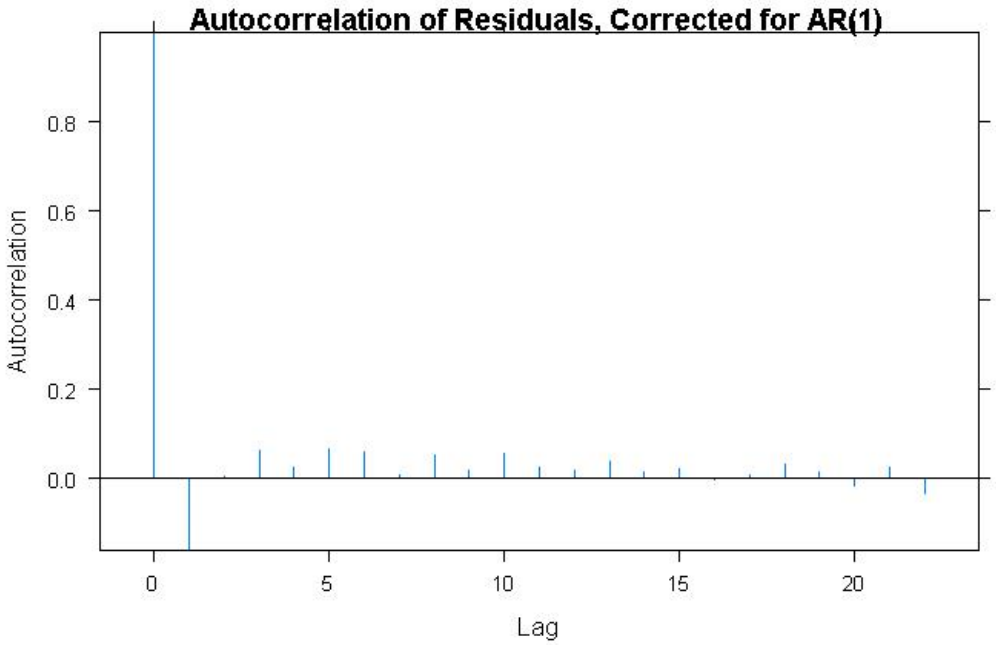

Figure 126: Autocorrelation function of the residuals from the estimated linear random effects model with autocorrelation for the transactions data fit to the relative price premium.

Because of the continued heteroskedasticity in the residuals, we now model the logarithm of the relative price premium. The fitted trees without random effects, with random effects, and with random effects and autocorrelation are plotted in Figures 127, 128, and 129, respectively. In this case, the trees differ very little. The one difference in the tree structure is that the tree without random effects splits at $SellerLife = 4805.5$ while the RE-EM trees with and without autocorrelation split at $SellerLife = 4653.5$. The estimated means at the nodes differ slightly across the trees, since subtracting the estimated random effects changes the values slightly. In these trees, the amount of feedback in the last year and the life of the seller are the main predictors, while characteristics of the competitors (the number, the average rating, and their average price) and the number of hours that the listing has been posted appear in other splits. The likelihood ratio test for autocorrelation rejects the hypothesis of no autocorrelation ($p < 10^{-200}$), and the autocorrelation functions associated with the RE-EM trees with and without autocorrelation, shown in Figures 132 and 133, respectively, demonstrate that there is autocorrelation in the residuals that is removed when the autocorrelation is included in the model. As before, allowing for autocorrelation leads to a slightly negative autocorrelation at the first lag.

As before, we fit linear models with and without random effects to these data, using the predictors chosen by the RE-EM tree. (As with the price premium, none of the chosen predictors have missing values.) Many of the predictors chosen by the RE-EM tree have coefficients that are not significantly different from zero in the linear models. The number of hours posted and the average competitor price are the only predictors that are statistically significant in all three models, and coefficient on the number of hours posted switches sign when the model allows for autocorrelation. The estimated autocorrelation function of the residuals, given in

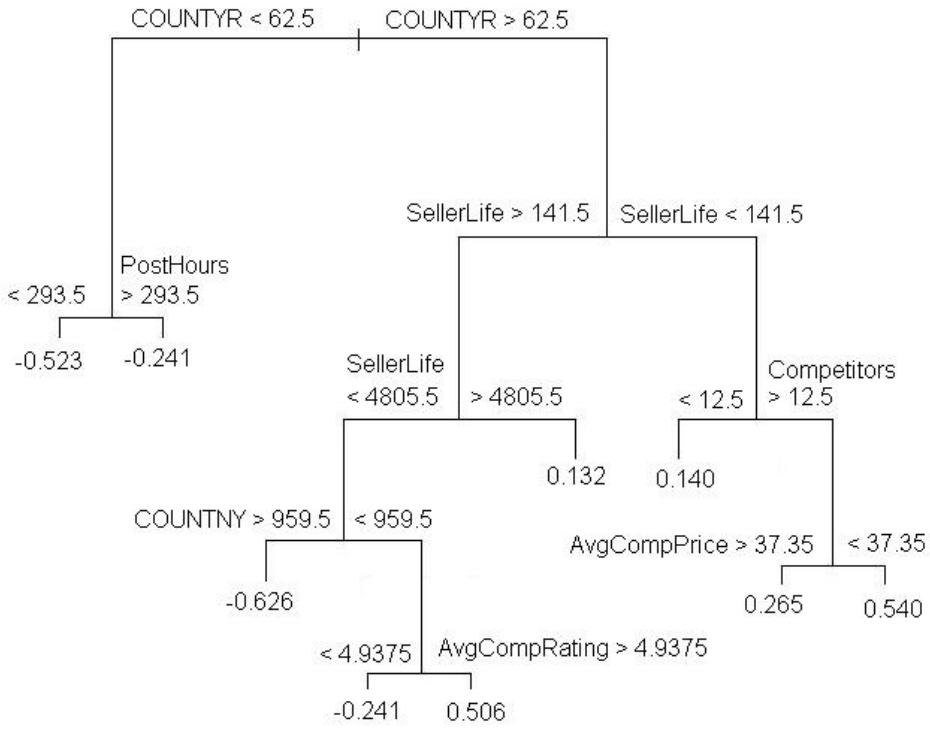

Figure 127: Estimated tree without random effects for the logged relative price premium in the transactions data.

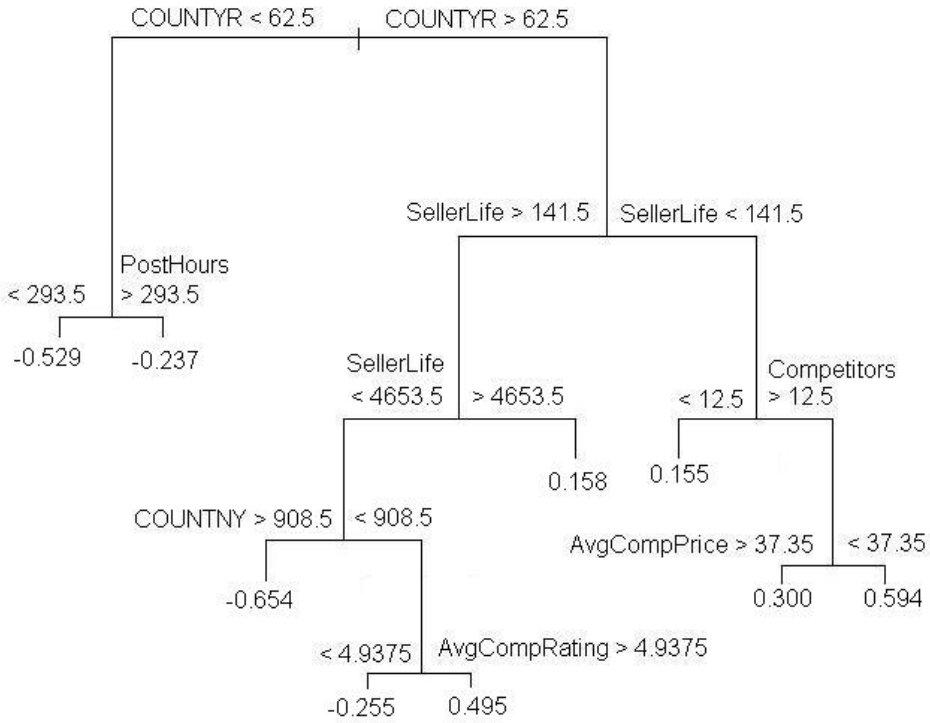

Figure 128: Estimated RE-EM tree for the logged relative price premium in the transactions data.

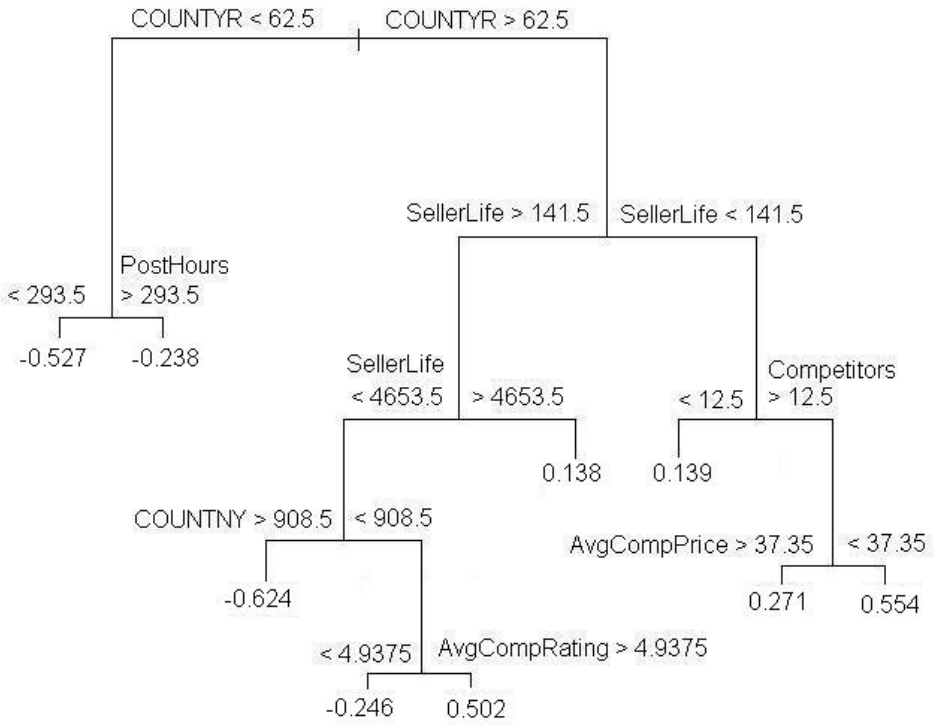

Figure 129: Estimated RE-EM tree with autocorrelation for the logged relative price premium in the transactions data.

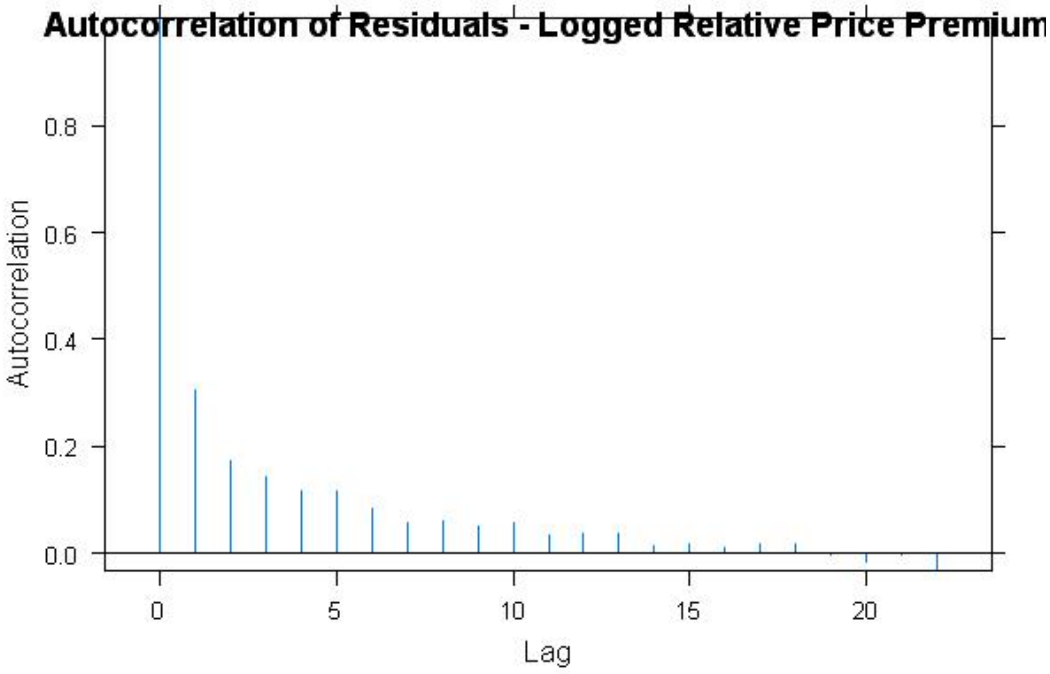

Figure 130: Autocorrelation function of the residuals from the estimated RE-EM tree without autocorrelation for the transactions data fit to the relative price premium.

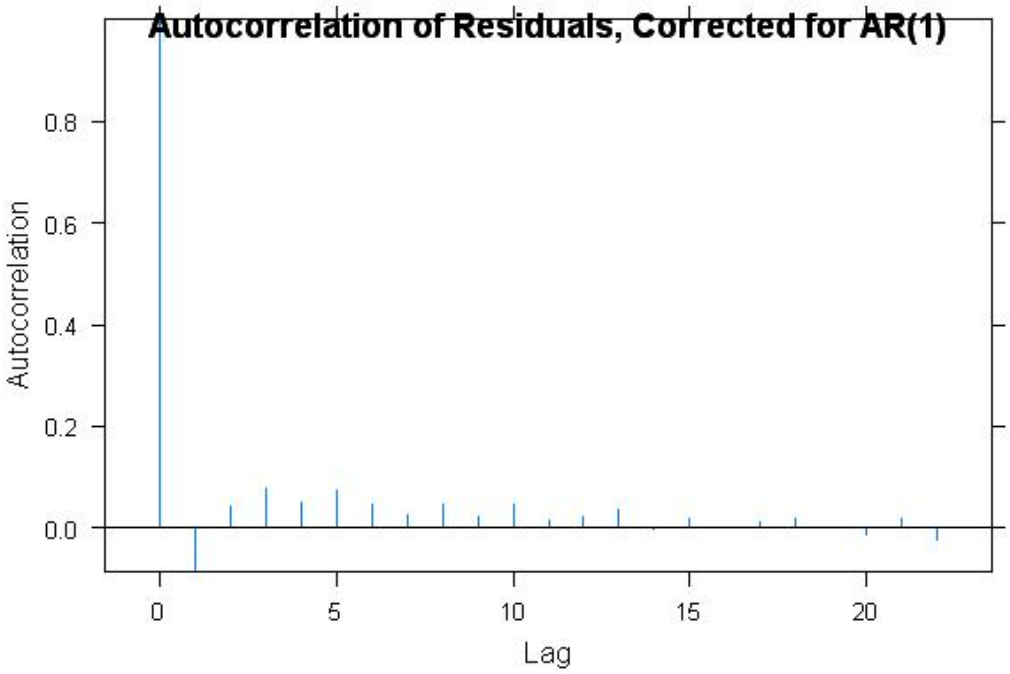

Figure 131: Autocorrelation function of the residuals from the estimated RE-EM tree with autocorrelation for the transactions data fit to the relative price premium.

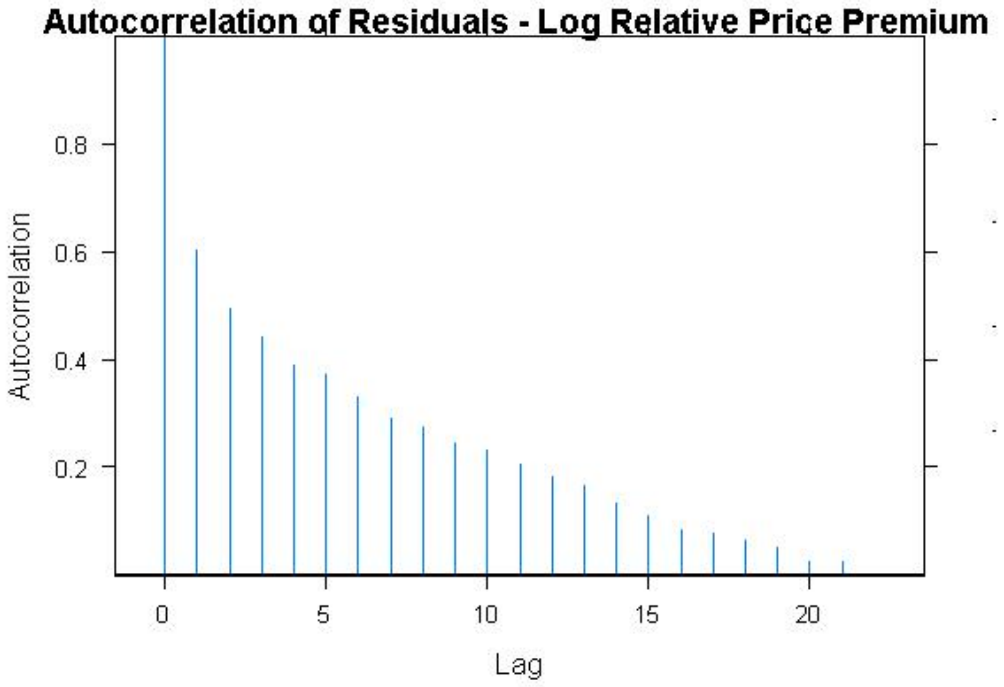

Figure 132: Autocorrelation function of the residuals from the estimated random effects linear model without autocorrelation for the transactions data fit to the logged relative price premium.

Figure 132, shows that autocorrelation is present in the residuals; the amount of autocorrelation in the linear model is slightly less than the autocorrelation in the estimated RE-EM tree. A plot of the autocorrelation function when autocorrelation of autoregressive order one is included in the model is given in Figure 133. As before, accounting for autocorrelation greatly reduces the amount of autocorrelation in the residuals but induces a slight negative autocorrelation at the first lag.

Diagnostic plots for the models for the logged relative price premium show

Variable	Linear Model	Random Effects Model	Random Effects - AR(1)
(Intercept)	0.277***	0.034	0.129 (0.092)
	(0.089)	(0.132)	
Number of Comments in the Last Year	-4.275E-6	1.289E-5	9.55E-6
	(1.035E-5)	(9.95E-6)	(7.60E-6)
Number of Hours Posted	-4.033E-4***	-2.516E-4***	1.171E-4***
	(2.455E-5)	(2.416E-5)	(1.686E-5)
Seller Life	6.404E-6	2.7E-7	-2.70E-6
	(4.386E0-6)	(4.16E-6)	(2.91E-6)
Number of Competitors	-1.047E-3	-1.267E-4	-1.541E-3
	(6.661E-4)	(1.789E-3)	(1.288E-3)
Number of Negative Comments in the Last Years	-7.401E-6	-3.005E-5*	-1.256E-5
	(1.659E-5)	(1.590E-5)	(1.253E-5)
Average Competitor Price	-3.327E-5**	-4.510E-4***	-6.545E-4***
	(1.634E-5)	(6.181E-5)	(6.405E-5)
Average Rating of Competitors	-0.054***	7.124E-3	-0.014
	(0.020)	(0.029)	(0.019)

Table 74: Parameter estimates for the linear models for the logged relative price premium with and without random effects. Standard errors are reported in parentheses. * - Significantly different from zero at the 10% level. ** - Significantly different from zero at the 5% level *** - Significantly different from zero at the 1% level.

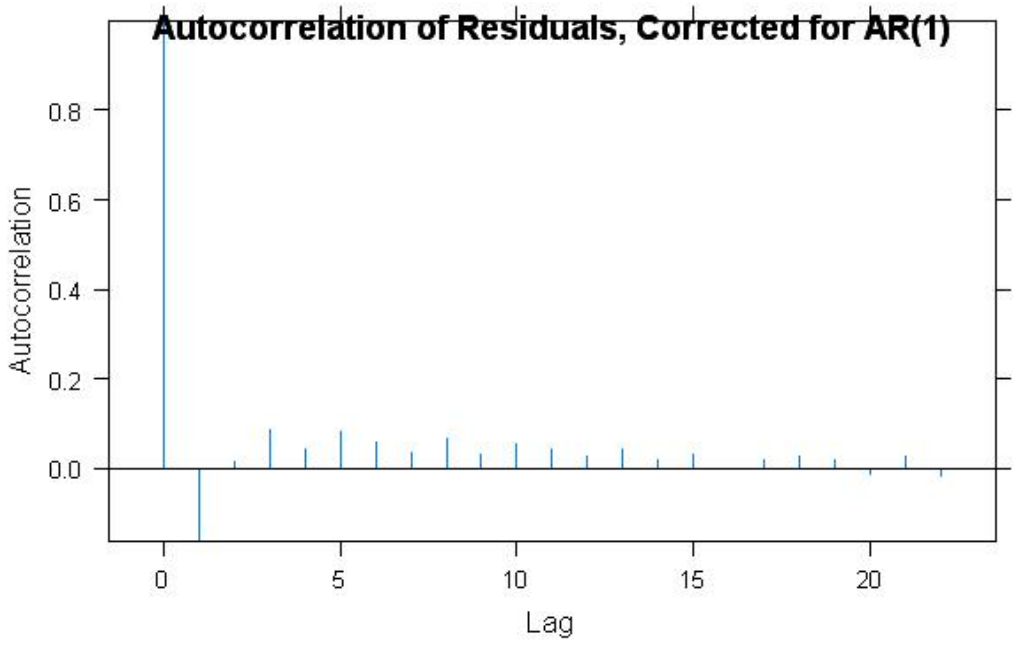

Figure 133: Autocorrelation function of the residuals from the estimated random effects linear model with autocorrelation for the transactions data fit to the logged relative price premium.

that taking the logarithm has reduced the heteroskedasticity somewhat, but that non-normality and outliers remain. Plots of the residuals versus the fitted values are given in Figures 134 and 135 for the RE-EM tree and linear random effects model respectively. Both show a large negative outlier, but little evidence of heteroskedasticity. Plots of the residuals by title, given in Figures 136 and 137 for the two models, show some heteroskedasticity and outliers, but less than we saw before taking logs. Omitting the outlier and re-estimating the models has little effect on the estimates. Quantile-quantile plots, shown in Figures 138 and 139, show that the residuals are not normally distributed in this model either. The residuals for the linear random effects model have a slightly more normal distribution than those of the RE-EM tree.

Again, we compute the root mean squared error for predictions using leave-one-out cross validation in which we omit one observation at a time and then one title at a time. The results are given in Table 75. Again, the three tree models outperform the three linear models. In this case, the tree without random effects has a slightly lower root mean squared error than the trees with random effects in the case in which we exclude all the observations for the title. This is a case in which we might expect their performance to be similar, since excluding all the observations for a title removes the possibility of estimating a title-specific random effect. When we exclude single observations and therefore can estimate the random effects, the RE-EM trees perform better than the tree without random effects.

In this section, we have modeled three different functions of the price premium. We have found that the underlying model cannot be well approximated by a linear model with random effects with either functional form of the target variable. Trees without random effects perform better than linear models, but they are not able to use the additional title-specific information when it is available for predictions.

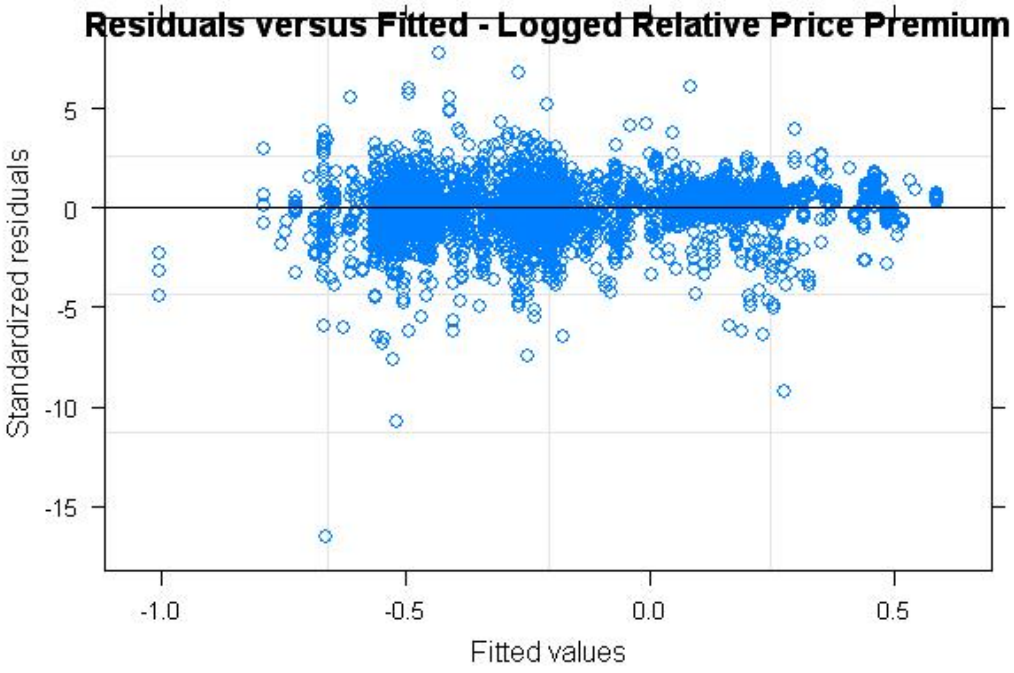

Figure 134: Plot of residuals versus fitted values from the estimated RE-EM tree for the transactions data fit to the logged relative price premium.

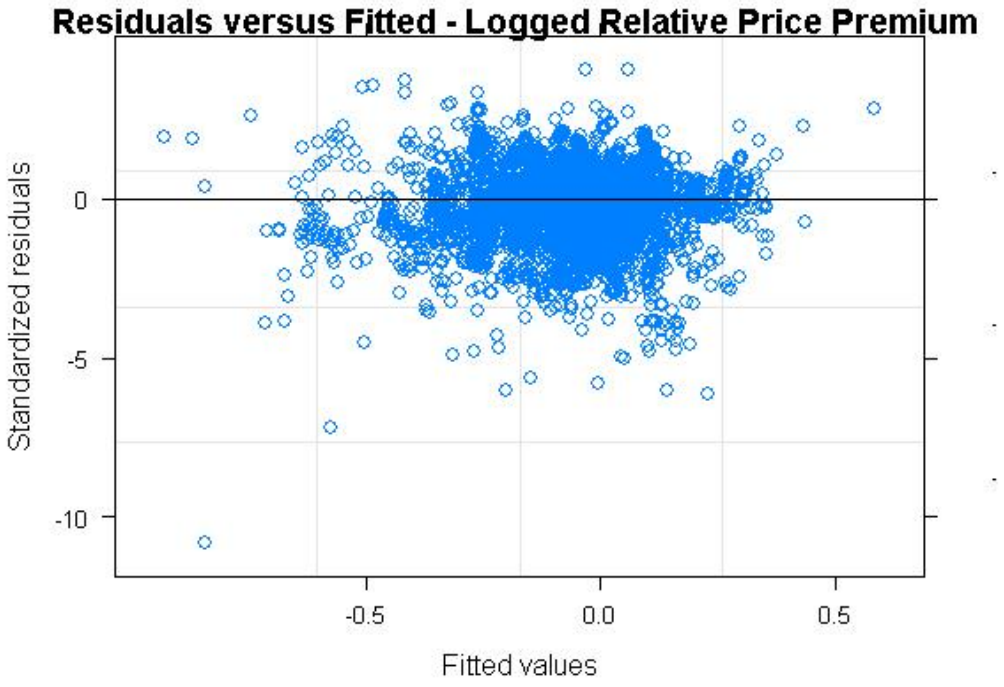

Figure 135: Plot of residuals versus fitted values from the estimated linear random effects model for the transactions data fit to the logged relative price premium.

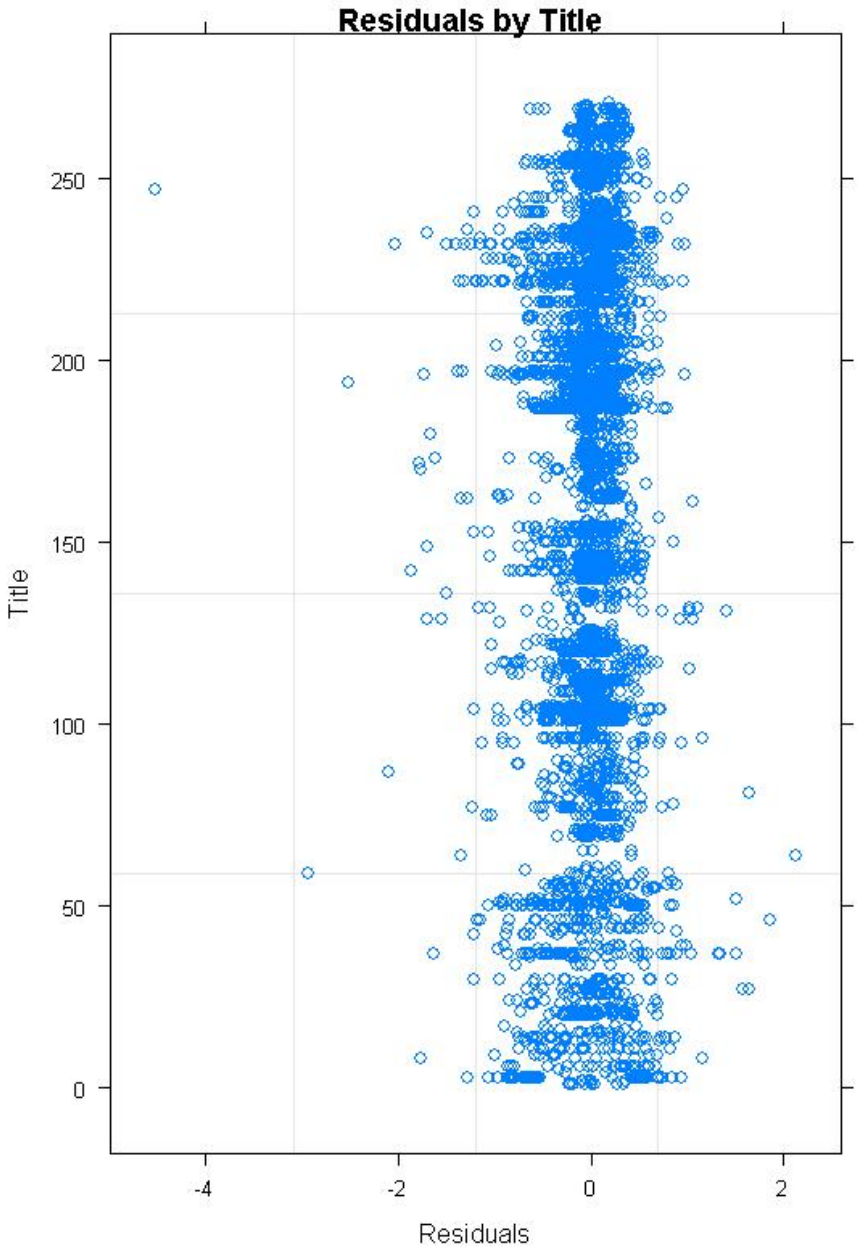

Figure 136: Plot of residuals for each software title from the estimated RE-EM tree for the transactions data fit to the logged relative price premium.

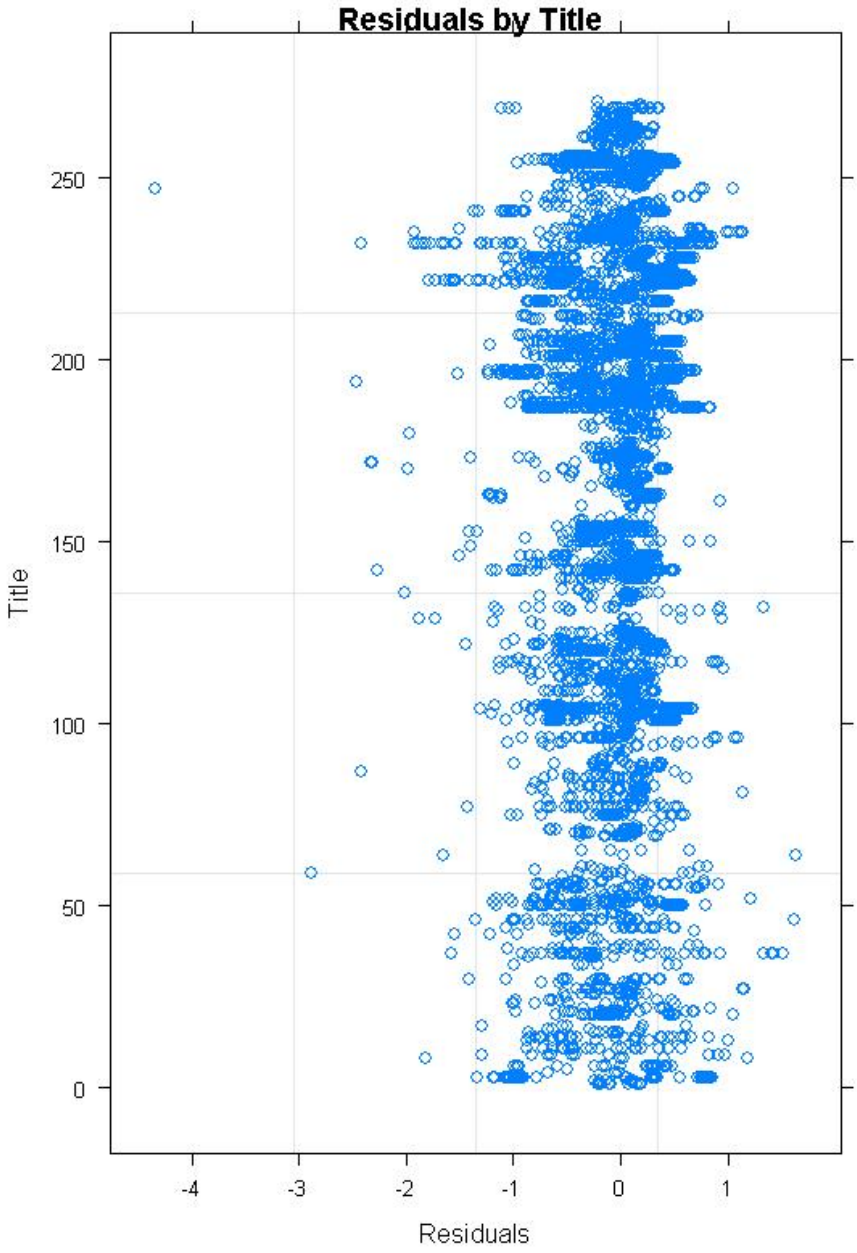

Figure 137: Plot of residuals for each software title from the estimated linear random effects model for the transactions data fit to the logged relative price premium.

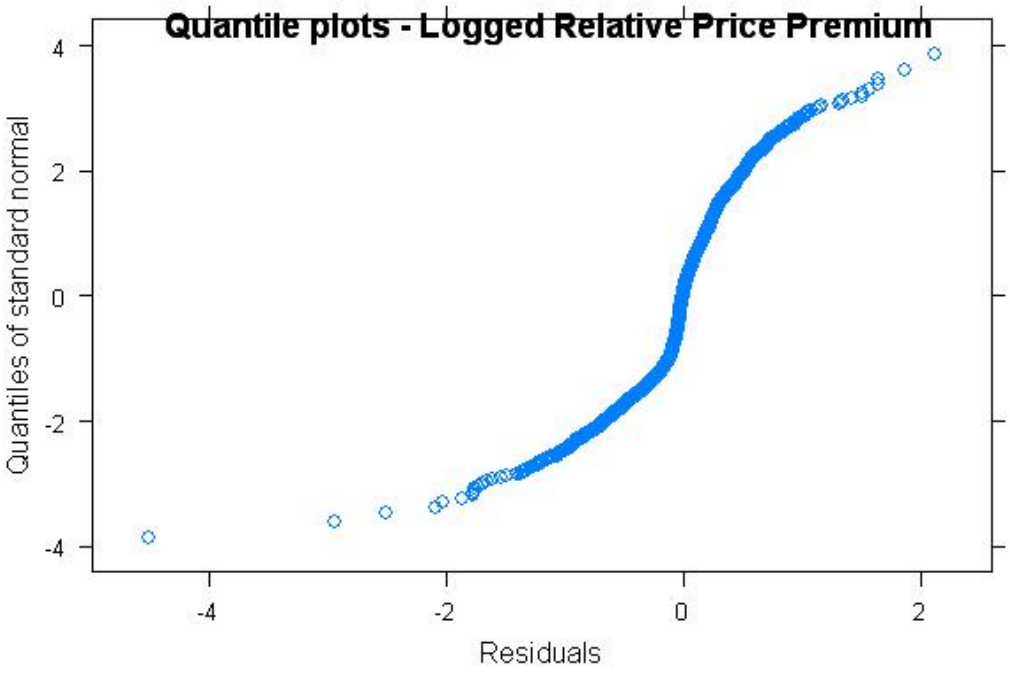

Figure 138: Quantile-quantile plot of residuals from the estimated RE-EM tree for the transactions data fit to the logged relative price premium.

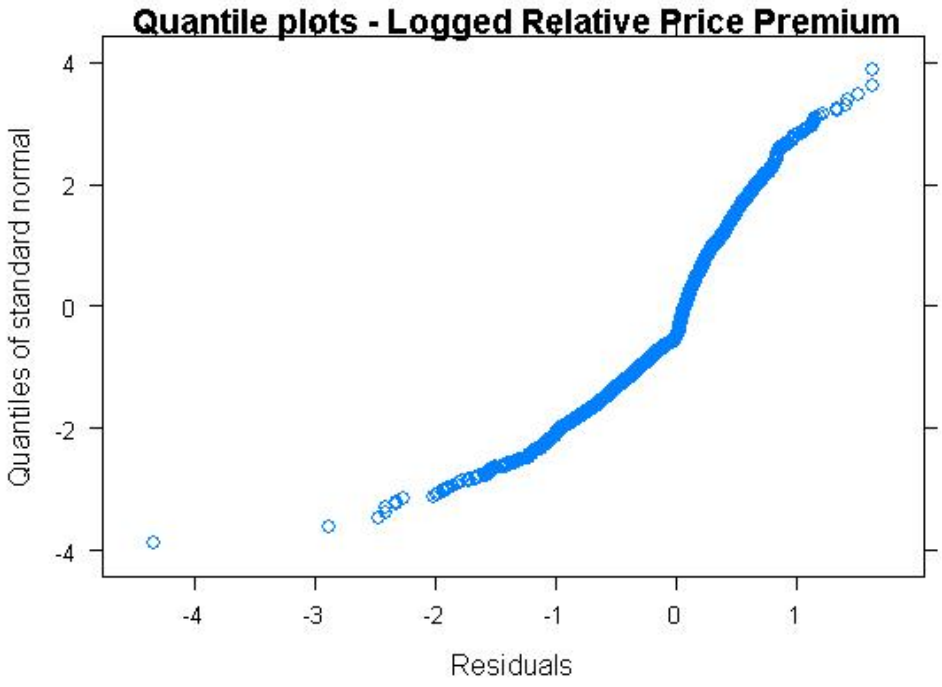

Figure 139: Quantile-quantile plot of residuals from the estimated linear random effects model for the transactions data fit to the logged relative price premium.

Model	In-sample	Excluding Observations	Excluding Titles
Linear Model	0.3643	0.4054	0.4069
Linear Model with Random Effects	0.3379	0.3758	0.4282
Linear Model with Random Effects - AR(1)	0.3592	0.3846	0.4607
Tree without Random Effects	0.2824	0.2845	0.2949
RE-EM Tree	0.2576	0.2708	0.2987
RE-EM Tree - AR(1)	0.2626	0.2713	0.2963
FE-EM Tree	0.2438	0.2686	0.3022

Table 75: In-sample root mean squared errors and root mean squared errors from cross-validation leaving out one observation or one software title at a time, using the transactions data, using the logged relative price premium.

The flexibility of the RE-EM tree allows us to model the price premium without worrying about the effect of the choice of functional form.

4.6 Simulations

4.6.1 Experimental design

We now use Monte Carlo simulations to assess the usefulness and effectiveness of the RE-EM tree method. These simulations consider datasets that contain $I = 50, 100, 200$ or 400 individuals, with the number of observations varying across individuals, with averages of approximately 10, 25, or 38 observations per individual[5]. We consider three data generating processes, to assess cases in which the tree model is only an approximation to reality. In each experiment, we compare the performance of the RE-EM tree with a tree that does not account for random effects and with parametric linear models that do and do not include random effects.

Our data generation procedure is based on estimated models for the price premium fit to the full transactions dataset discussed in Section 4.5. This simulates complex yet realistic data patterns. Specifically, the "true" models are a RE-EM tree in the first round of experiments, a linear model with scalar random effects in the second round, and a more complicated model in the third. In the third case, we define f by estimating a linear model including all possible interactions of the eight continuous variables that appeared in the trees, listed in Table 68, together with the squares of `AvgCompPrice`, `AvgCompLife`, `AvgCompCondition`, and `AvgCompRating`. All but the last of the squared variables has a statistically significant coefficient, and some of the interactions terms have statistically signifi-

[5]The average number of observations per individual in the underlying dataset on which the simulations are based is 38.

cant coefficients as well. Each model is estimated based on the full dataset. This estimation yields a prediction for any set of covariates as well as a list of estimated random effects, $\hat{b}_i$, and estimated observation errors, $\hat{\varepsilon}_{it}$, for each individual. For each sample size, I, we use the covariates from a random sample (with replacement) of I individuals to compute the expected value, $E(y_{it})$ of the target variable given the true model. We use a random effect, $\hat{b}_i$, from one randomly chosen individual, and choose the $\hat{\varepsilon}_{it}$ for $t = 1, ..., T$ from the set of all estimated errors. Then, the new observed data consist of $y_{it} = E(y_{it}) + \hat{b}_i + \hat{\varepsilon}_{it}$ together with the covariates. Data are created in the same way for an additional 50 individuals who are used as the hold-out sample. For each group of $I + 50$ individuals, we resample 50 times in this way, which allows us to check for any effects of the covariates on predictive performance. We then move on to a new sample of size $I + 50$ and repeat the experiment for 50 different samples of individuals.

To measure performance both in- and out-of-sample, we fit each model to the first 75% of observations for I individuals. We compute the in-sample RMSE based on the observations that were included in the sample and then predict the future observations for those individuals to estimate the out-of-sample performance of the models for future observations for individuals used in estimation. For the additional sample of 50 individuals, we predict the first 75% of their observations using just $\hat{f}$; this allows us to measure the prediction performance for new individuals. Finally, we use the original fitted model and the first 75% of observations for the new individuals to predict the last 25% of observations for those individuals. This allows us to measure the prediction performance for future observations of new individuals. These results are discussed in Section 4.6.2. In that section, we also test whether the root mean squared errors from RE-EM trees differ significantly from those of other methods, using the Wilcoxon signed-rank test. Furthermore,

since we know the true values of b_i and $f(x_{it\cdot})$, we find the root mean squared error of the estimates of these quantities in sample. We discuss these results in Section 4.6.3. We also explore the effect of increased variability in the errors and of autocorrelation in the errors in Section 4.6.4. We compute the root mean squared differences between in-sample fits for different RE-EM estimation methods; we discuss these results in Section 4.6.5.

Finally, we look at the performance of the estimators in balanced panels in Section 4.6.6, which allows us to describe the performance of individual-specific regressions and regression trees as well as the trees of Segal [1992] and De'Ath [2002] (MVPART), using the `MVPART` package of De'ath [2006] in `R`.

4.6.2 Predictive Performance

In Tables 76, 77, and 78, we present the in-sample root mean squared errors for each model when the true data generating processes are a RE-EM tree, a linear model with random effects, and the more complicated model, respectively. In each table, we present the results of estimating six different models: a linear model without random effects (LM), a linear model with random effects and without autocorrelation (LME), a linear model with random effects and autocorrelation of order 1 (LME - AR(1)), a regression tree without random effects (RPART), a RE-EM tree without autocorrelation (REEM), and a RE-EM tree with autocorrelation of order 1 (REEM - AR(1)). We also considered models fit to each individual separately, as Afshartous and de Leeuw [2005] did. We found that these models performed quite badly. While the linear model sometimes fit well in-sample (because the sample size was small), RPART did not fit well in-sample and both models performed very badly in the prediction of future values. Individual models would not be able to be used at all for predictions for new individuals.

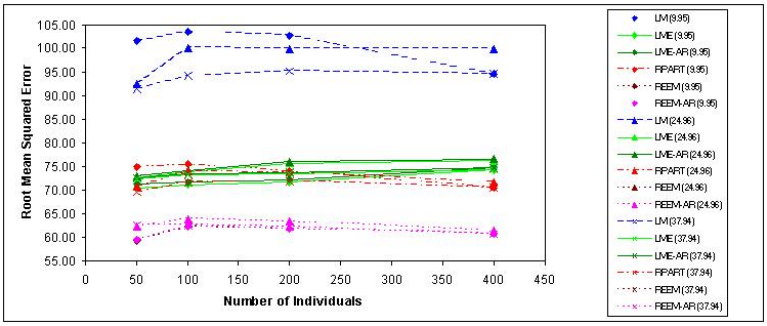

Figure 140: In-sample root mean squared errors when the true data generating process is a RE-EM tree.

When the true process is a RE-EM tree, given in Table 76, the RE-EM tree without autocorrelation has the lowest RMSE in-sample. In the other two cases, given in Tables 77 and 78, the linear model with random effects has the lower in-sample root mean squared error. Figures 140 and 141 present the same results graphically when the true data generating processes are the RE-EM tree and the linear model with random effects. All of the differences between models for the same sample sizes are statistically significant. The linear model without random effects has the highest RMSE in all cases. The RMSE is generally constant or decreasing as a function of the number of observations per group. This occurs in part because of the larger total number of observations across all individuals.

Next, we consider the prediction error for future observations for individuals that are already in the sample, reported in Tables 79, 80, and 81 for the three data generating processes. As with the in-sample fits, the RE-EM tree has the lowest RMSE when it is the true data generating process (Table 79), while the linear model with random effects has the lowest RMSE in the other two cases (Tables 80 and 81). The difference between the linear model with random effects and the RE-EM tree is not very large in a practical sense in the latter two cases, though

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	101.66	70.57	71.33	74.95	59.42	59.63
50	24.96	92.45	72.68	73.06	70.71	62.20	62.20
50	37.94	91.33	72.37	72.64	69.60	62.66	62.65
100	9.95	103.56	71.06	71.74	75.51	62.19	62.42
100	24.96	100.08	73.92	74.15	73.93	63.95	63.94
100	37.94	94.20	73.16	73.41	71.63	62.58	62.58
200	9.95	102.64	71.63	72.32	73.85	61.73	61.90
200	24.96	99.84	75.68	75.97	73.72	63.37	63.37
200	37.94	95.15	73.50	73.75	71.92	62.12	62.12
400	9.95	104.14	71.48	72.10	72.66	60.62	60.73
400	24.96	99.95	76.29	76.61	71.67	61.38	61.38
400	37.94	94.63	74.29	74.61	70.62	60.67	60.67

Table 76: In-sample root mean squared errors when the true data generating process is a RE-EM tree.

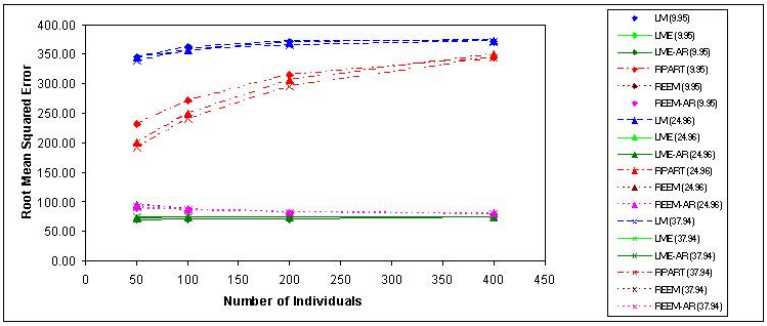

Figure 141: In-sample root mean squared errors when the true data generating process is a RE-EM tree.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	345.03	69.67	69.70	232.35	88.76	88.79
50	24.96	344.13	73.25	73.25	201.36	94.79	94.60
50	37.94	339.47	74.25	74.25	192.27	91.60	91.60
100	9.95	362.10	70.54	70.56	271.01	85.29	85.26
100	24.96	358.22	73.75	73.75	249.65	87.43	87.52
100	37.94	356.53	74.47	74.47	241.20	86.38	86.39
200	9.95	370.64	71.22	71.23	315.89	81.19	81.19
200	24.96	370.31	74.02	74.02	305.87	82.42	82.46
200	37.94	365.81	74.78	74.78	295.44	81.96	81.95
400	9.95	372.90	71.68	71.69	353.27	77.66	77.70
400	24.96	373.27	74.07	74.07	349.56	79.56	79.60
400	37.94	371.12	74.71	74.71	343.59	79.20	79.18

Table 77: In-sample root mean squared errors when the true data generating process is a linear model with random effects.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	337.94	70.63	70.66	229.48	84.09	84.43
50	24.96	335.07	74.15	74.15	196.41	91.25	90.96
50	37.94	325.75	74.74	74.74	181.51	89.38	89.39
100	9.95	354.40	71.33	71.35	264.03	84.32	84.35
100	24.96	348.77	74.17	74.18	236.78	87.16	87.27
100	37.94	349.04	74.83	74.83	228.77	85.33	85.33
200	9.95	366.58	72.29	72.30	307.45	81.61	81.57
200	24.96	361.47	74.62	74.62	288.63	84.26	84.10
200	37.94	360.58	75.13	75.13	281.53	82.22	82.19
400	9.95	371.50	72.63	72.63	343.41	78.79	78.81
400	24.96	370.81	74.78	74.78	338.73	80.22	80.28
400	37.94	370.53	75.08	75.08	333.18	79.44	79.45

Table 78: In-sample root mean squared errors when the true data generating process is a more complicated model.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	96.03	92.16	91.60	83.37	80.99	80.90
50	24.96	88.70	77.82	78.04	74.63	70.88	70.87
50	37.94	98.09	87.80	88.06	78.94	74.07	74.04
100	9.95	94.57	88.84	88.37	79.72	75.29	75.27
100	24.96	93.55	79.80	79.82	75.23	68.41	68.41
100	37.94	98.80	86.95	87.24	76.94	69.77	69.77
200	9.95	94.53	88.95	88.45	75.93	70.37	70.33
200	24.96	94.13	80.33	80.47	75.83	68.03	68.02
200	37.94	98.30	86.47	86.77	74.32	66.39	66.39
400	9.95	96.37	89.82	89.30	73.49	67.45	67.45
400	24.96	93.44	80.83	80.99	72.80	64.92	64.95
400	37.94	98.29	87.22	87.61	71.35	63.08	63.08

Table 79: Out-of-sample root mean squared prediction errors for future observations when the true data generating process is a RE-EM tree.

it is statistically significant. In a few cases (in particular, when $I = 200, 400$ and $E(T_i) = 9.95$ for a linear model and when $I = 100$ and $E(T_i) = 9.95$ for the complicated linear model), the difference between the RMSE between the linear model with random effects and the RE-EM tree is not statistically significant. When the RE-EM tree is the true data generating process, a regression tree without random effects outperforms the linear model without random effects in some cases, presumably because the tree is closer to the true data generating process.

For new observations, prediction using RE-EM trees generally has the lowest mean squared errors, though the linear model without random effects performs about as well, particularly when the true data generating process is a linear model

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	348.41	80.14	80.20	281.39	107.95	108.17
50	24.96	344.50	77.10	77.10	232.31	101.87	101.69
50	37.94	339.79	76.84	76.84	217.09	97.49	97.54
100	9.95	362.98	80.57	80.61	314.46	99.28	99.26
100	24.96	358.44	77.34	77.34	273.78	92.24	92.34
100	37.94	356.58	76.91	76.92	261.91	90.69	90.73
200	9.95	370.14	80.47	80.48	342.01	92.44	92.47
200	24.96	370.09	77.91	77.91	319.81	87.07	87.09
200	37.94	365.47	77.06	77.06	306.45	85.40	85.41
400	9.95	371.62	81.21	81.22	360.97	88.15	88.19
400	24.96	372.76	77.69	77.69	353.33	83.45	83.48
400	37.94	370.42	77.19	77.19	346.49	82.21	82.17

Table 80: Out-of-sample root mean squared prediction errors for future observations when the true data generating process is a linear model with random effects.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	341.65	81.75	81.81	275.44	106.28	106.50
50	24.96	335.32	76.84	76.85	225.39	98.09	97.83
50	37.94	325.51	77.04	77.04	208.19	95.81	95.82
100	9.95	354.80	81.22	81.25	304.76	99.31	99.51
100	24.96	348.21	77.79	77.79	263.50	92.42	92.56
100	37.94	348.82	77.50	77.50	252.51	90.01	89.98
200	9.95	365.84	81.77	81.79	337.30	93.39	93.44
200	24.96	361.02	77.83	77.83	305.59	88.72	88.57
200	37.94	359.65	77.59	77.59	295.91	86.07	86.05
400	9.95	369.87	81.91	81.93	356.56	89.57	89.59
400	24.96	370.27	78.07	78.07	345.12	83.93	83.99
400	37.94	369.96	77.80	77.80	338.76	83.14	83.15

Table 81: Out-of-sample root mean squared prediction errors for future observations when the true data generating process is a more complicated model.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	114.35	114.26	114.00	104.40	97.64	97.65
50	24.96	101.02	155.49	151.22	98.37	92.39	92.47
50	37.94	97.81	169.15	165.39	94.24	89.36	89.36
100	9.95	108.86	110.06	109.74	92.38	90.26	90.22
100	24.96	102.00	190.25	188.16	86.83	85.40	85.39
100	37.94	91.28	180.68	175.52	81.29	79.48	79.48
200	9.95	111.54	113.22	112.97	88.91	89.44	89.35
200	24.96	101.22	202.51	197.00	85.33	85.00	85.04
200	37.94	95.16	214.96	204.20	76.74	76.32	76.32
400	9.95	103.03	103.79	103.71	76.72	77.06	77.04
400	24.96	99.36	183.10	177.68	75.09	74.92	74.92
400	37.94	94.41	182.10	171.50	73.85	74.06	74.06

Table 82: Out-of-sample root mean squared prediction errors for new observations when the true data generating process is a RE-EM tree.

with random effects. Full results are given in Tables 82, 83, and 84. The difference between the root mean squared errors for the linear model and the RE-EM tree are not statistically significant (at the $p = 0.01$ level) for five parameter configurations when the true data generating process is a linear model with random effects and for another five when the true process is the more complicated linear model. The difference between the root mean squared errors of prediction for the regression trees with and without random effects is not statistically significant for one parameter configuration.

Finally, we compare the predictions for individuals who were not in the original sample, using some of their observations to estimate random effects. As before, the

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	389.97	427.45	427.30	462.81	390.54	390.48
50	24.96	390.77	482.34	482.34	472.75	391.76	391.57
50	37.94	388.99	486.33	486.35	466.38	387.46	387.44
100	9.95	379.54	425.21	425.05	449.48	382.34	382.26
100	24.96	378.83	473.14	473.14	455.95	383.75	383.85
100	37.94	374.03	480.44	480.42	453.98	377.03	377.05
200	9.95	369.20	420.81	420.65	418.96	371.29	371.26
200	24.96	370.13	479.10	479.10	426.25	374.86	374.86
200	37.94	366.47	501.59	501.59	426.06	371.75	371.67
400	9.95	371.02	417.28	417.17	391.71	372.92	372.94
400	24.96	363.72	496.29	496.28	390.61	368.75	368.72
400	37.94	361.90	501.69	501.69	393.39	367.12	367.05

Table 83: Out-of-sample root mean squared prediction errors for new observations when the true data generating process is a linear model with random effects.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	388.79	426.10	425.92	460.22	388.54	388.36
50	24.96	375.90	471.31	471.27	457.93	382.63	382.46
50	37.94	376.90	462.49	462.47	457.94	378.06	377.94
100	9.95	370.44	404.33	404.12	442.36	373.53	373.53
100	24.96	362.82	474.06	473.94	442.85	368.70	368.57
100	37.94	363.39	456.98	456.93	445.31	365.42	365.37
200	9.95	366.48	403.98	403.80	419.61	368.24	368.28
200	24.96	358.28	484.34	484.28	420.68	364.40	364.39
200	37.94	354.47	484.45	484.38	422.96	358.49	358.49
400	9.95	360.94	407.53	407.32	392.17	363.72	363.70
400	24.96	355.79	478.48	478.43	391.79	359.79	359.74
400	37.94	350.17	478.39	478.33	390.44	353.36	353.34

Table 84: Out-of-sample root mean squared prediction errors for new observations when the true data generating process is a more complicated model.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	100.29	91.52	91.45	93.40	80.79	80.88
50	24.96	93.19	80.59	80.80	93.87	75.15	75.17
50	37.94	99.87	90.86	91.21	93.31	77.35	77.36
100	9.95	94.41	89.89	89.87	82.27	75.03	75.09
100	24.96	95.92	83.24	83.42	85.75	75.74	75.71
100	37.94	96.87	86.54	86.91	81.87	71.53	71.53
200	9.95	97.28	92.95	92.93	78.14	74.07	73.92
200	24.96	93.17	82.61	82.80	85.74	74.01	74.01
200	37.94	96.87	86.83	87.18	76.49	67.36	67.36
400	9.95	95.25	88.49	88.49	74.07	68.56	68.53
400	24.96	93.32	82.39	82.67	76.20	67.54	67.54
400	37.94	95.49	86.23	86.74	72.11	63.84	63.84

Table 85: Out-of-sample root mean squared prediction errors for future observations for new individuals when the true data generating process is a RE-EM tree.

RE-EM tree performs best when it is the true data generating process, as shown in Table 85. The linear model performs best in the other two cases, given in Tables 86 and 87. However, the difference between the linear model and the RE-EM tree (which consistently performs second best) narrows considerably as the sample size grows, and the RE-EM tree outperforms the linear model with random effects in one case.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	385.91	80.28	80.30	461.29	101.15	101.00
50	24.96	389.05	77.48	77.48	471.97	96.81	96.69
50	37.94	387.42	76.81	76.81	465.31	96.15	96.13
100	9.95	375.42	79.70	79.71	447.52	93.09	93.04
100	24.96	377.09	77.13	77.13	454.41	92.08	92.07
100	37.94	373.47	76.19	76.19	453.25	88.61	88.66
200	9.95	364.47	79.51	79.52	414.69	88.59	88.49
200	24.96	368.33	77.10	77.10	422.94	86.03	86.06
200	37.94	365.28	76.36	76.36	424.33	84.33	84.38
400	9.95	366.84	79.26	79.27	387.96	84.31	84.45
400	24.96	362.30	76.05	76.05	388.37	81.93	81.94
400	37.94	360.50	76.55	76.55	391.76	82.74	82.72

Table 86: Out-of-sample root mean squared prediction errors for future observations for new individuals when the true data generating process is a linear model with random effects.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	381.86	81.94	81.95	453.62	101.77	101.98
50	24.96	375.39	77.83	77.84	456.08	97.64	97.67
50	37.94	376.39	78.06	78.06	455.80	96.70	96.68
100	9.95	365.10	80.91	80.92	439.32	94.52	94.36
100	24.96	361.18	78.02	78.02	440.59	90.71	90.84
100	37.94	361.67	77.80	77.80	443.32	90.64	90.66
200	9.95	361.67	79.92	79.93	416.34	88.36	88.26
200	24.96	355.93	77.48	77.48	418.84	85.94	85.99
200	37.94	353.05	77.10	77.10	421.63	84.64	84.60
400	9.95	356.87	80.75	80.76	388.12	87.02	87.01
400	24.96	353.88	76.98	76.98	390.12	82.55	82.55
400	37.94	348.45	76.67	76.67	388.52	81.73	81.73

Table 87: Out-of-sample root mean squared prediction errors for future observations for new individuals when the true data generating process is a more complicated model.

4.6.3 Estimation of the Underlying Function and Random Effects

In addition to assessing the predictive power of the models, we measure how well the linear model with random effects and the RE-EM tree are able to estimate the random effects, and how well each of the estimation methods are able to estimate the fitted values. For all three data generating processes, the RE-EM tree has a lower root mean squared error in estimating the random effects (Tables 88, 89, and 90). The RE-EM tree and the linear model without random effects generally have the lowest root mean squared errors in estimated the values of $f(x_{it})$ at each point (Tables 91, 92, and 93). Furthermore, the performance of the linear model with random effects generally deteriorates as the number of observations per individual grows, while the performance of the RE-EM tree stays the same or improves.

The performance of the RE-EM tree seems to contradict the results of the previous section, in which the linear model with random effects often performs better for predicting future observations for individuals in the sample and outside the sample. To better understand this phenomenon, we consider a decomposition of the root mean squared prediction error. Consider a single new observation, $y_{it} = \alpha_i + f(x_{i\cdot t}) + \epsilon_{it}$. The predicted value for that observation is $\hat{y}_{it} = \hat{\alpha}_{it} + \hat{f}(x_{i\cdot t})$, where $\hat{f}$ is the estimated linear model or regression tree. Then, we may write the squared error and its expected value as:

$$\begin{aligned}
y_{it} - \hat{y}_{it} &= (\alpha_i - \hat{\alpha}_i) + (f(x_{i\cdot t}) - \hat{f}(x_{i\cdot t})) + \epsilon_{it} \\
(y_{it} - \hat{y}_{it})^2 &= (\alpha_i - \hat{\alpha}_i)^2 + (f(x_{i\cdot t}) - \hat{f}(x_{i\cdot t}))^2 + \epsilon_{it}^2 \\
&\quad +2(\alpha_i - \hat{\alpha}_i)(f(x_{i\cdot t}) - \hat{f}(x_{i\cdot t})) + 2(\alpha_i - \hat{\alpha}_i)\epsilon_{it} \\
&\quad +2(f(x_{i\cdot t}) - \hat{f}(x_{i\cdot t}))\epsilon_{it} \\
E((y_{it} - \hat{y}_{it})^2) &= E((\alpha_i - \hat{\alpha}_i)^2) + E((f(x_{i\cdot t}) - \hat{f}(x_{i\cdot t}))^2) + E(\epsilon_{it}^2) \\
&\quad +2E((\alpha_i - \hat{\alpha}_i)(f(x_{i\cdot t}) - \hat{f}(x_{i\cdot t})))
\end{aligned}$$

I	$E(T_i)$	LME	LME- AR(1)	REEM	REEM - AR(1)
50	9.95	63.13	60.35	43.86	43.57
50	24.96	100.54	96.70	38.98	39.04
50	37.94	124.62	120.29	38.49	38.47
100	9.95	65.98	62.99	33.63	33.17
100	24.96	165.17	161.56	28.81	28.80
100	37.94	172.96	165.92	29.39	29.31
200	9.95	62.18	58.95	24.61	24.37
200	24.96	155.28	148.74	21.53	21.54
200	37.94	197.12	184.95	21.01	20.97
400	9.95	65.29	62.03	21.99	21.92
400	24.96	145.07	137.50	18.35	18.36
400	37.94	174.75	159.64	17.99	17.99

Table 88: Root mean squared errors of estimated random effects when the true data generating process is a RE-EM tree.

I	$E(T_i)$	LME	LME - AR(1)	REEM	REEM - AR(1)
50	9.95	196.65	196.34	130.19	130.01
50	24.96	288.35	288.32	131.95	131.73
50	37.94	289.51	289.51	125.32	125.28
100	9.95	214.70	214.33	106.48	106.56
100	24.96	278.21	278.21	108.21	108.28
100	37.94	298.11	298.08	106.27	106.27
200	9.95	208.55	208.21	91.04	90.99
200	24.96	300.53	300.52	97.37	97.33
200	37.94	305.72	305.71	95.51	95.50
400	9.95	212.71	212.39	85.19	85.18
400	24.96	309.96	309.94	89.55	89.61
400	37.94	320.59	320.58	91.29	91.23

Table 89: Root mean squared errors of estimated random effects when the true data generating process is a linear model.

I	$E(T_i)$	LME	LME - AR(1)	REEM	REEM - AR(1)
50	9.95	183.28	182.98	127.25	126.75
50	24.96	280.39	280.34	134.04	134.21
50	37.94	281.09	281.03	122.66	122.53
100	9.95	177.47	177.08	103.70	103.58
100	24.96	279.38	279.21	109.76	109.80
100	37.94	275.62	275.55	98.76	98.65
200	9.95	184.66	184.26	90.96	90.98
200	24.96	308.49	308.41	101.07	101.07
200	37.94	316.99	316.89	97.17	97.00
400	9.95	191.94	191.57	85.17	85.27
400	24.96	316.94	316.85	91.99	91.99
400	37.94	330.06	329.97	88.62	88.49

Table 90: Root mean squared errors of estimated random effects when the true data generating process is the complicated model.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	75.43	79.61	79.46	50.04	51.41	51.32
50	24.96	61.31	117.00	114.12	36.78	36.16	36.20
50	37.94	60.02	137.93	134.85	34.31	30.66	30.68
100	9.95	76.58	80.60	80.37	38.75	38.93	38.74
100	24.96	71.19	181.30	179.04	30.08	29.41	29.40
100	37.94	63.03	180.89	174.95	25.97	23.79	23.76
200	9.95	75.13	77.77	77.59	27.52	28.10	28.01
200	24.96	70.94	176.30	170.92	24.04	24.15	24.17
200	37.94	64.34	209.72	197.99	18.59	17.96	17.94
400	9.95	76.90	79.52	79.38	20.88	21.57	21.52
400	24.96	71.13	165.42	159.12	14.84	14.82	14.82
400	37.94	63.44	188.57	174.23	9.57	9.18	9.18

Table 91: Root mean squared errors of estimated values of $f(x)$ when the true data generating process is a RE-EM tree.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	129.68	200.67	200.34	275.97	137.20	136.94
50	24.96	129.74	313.29	313.27	300.28	141.18	140.85
50	37.94	131.79	297.52	297.53	301.51	130.85	130.78
100	9.95	103.10	219.90	219.53	253.12	109.73	109.70
100	24.96	101.80	303.07	303.07	269.54	114.18	114.20
100	37.94	103.40	312.10	312.08	274.28	110.45	110.36
200	9.95	83.60	213.94	213.60	203.32	90.90	90.85
200	24.96	83.84	320.56	320.55	218.49	98.60	98.59
200	37.94	83.91	313.03	313.02	224.09	96.37	96.31
400	9.95	76.35	217.35	217.04	131.49	83.24	83.24
400	24.96	72.94	327.21	327.19	141.53	88.56	88.55
400	37.94	74.02	333.73	333.73	149.71	91.08	90.98

Table 92: Root mean squared errors of estimated values of $f(x)$ when the true data generating process is a linear model.

I	$E(T_i)$	LM	LME	LME - AR(1)	RPART	REEM	REEM - AR(1)
50	9.95	129.33	187.74	187.44	270.28	134.32	134.12
50	24.96	127.18	296.64	296.59	290.81	141.71	141.57
50	37.94	126.29	294.48	294.42	290.91	126.39	126.32
100	9.95	101.46	180.51	180.11	249.46	108.59	108.74
100	24.96	101.31	303.43	303.25	268.18	115.40	115.27
100	37.94	99.98	288.85	288.78	274.86	103.46	103.37
200	9.95	86.00	188.84	188.45	209.45	92.11	92.24
200	24.96	85.91	334.19	334.10	226.46	105.20	105.15
200	37.94	85.22	328.98	328.88	232.96	97.35	97.31
400	9.95	77.22	197.19	196.83	151.64	85.01	85.00
400	24.96	75.36	341.62	341.53	160.63	91.62	91.60
400	37.94	74.81	341.17	341.08	169.58	88.79	88.70

Table 93: Root mean squared errors of estimated values of $f(x)$ when the true data generating process is the complicated model.

Notice that $E(\epsilon_{it}^2)$ is constant across the two models, while the simulations have shown that both $E((\alpha_i - \hat{\alpha}_i)^2)$ and $E((f(x_{i\cdot t}) - \hat{f}(x_{i\cdot t}))^2)$ are larger for linear models with random effects than for RE-EM trees. In order for $E((y_{it} - \hat{y}_{it})^2)$ to be smaller for linear models, we must have $2E((\alpha_i - \hat{\alpha}_i)(f(x_{i\cdot t}) - \hat{f}(x_{i\cdot t})))$ smaller for linear models as well. This term is the covariance between the errors in estimating the random effects and the errors in estimating the values of f for the corresponding individuals. In Tables 94, 95, and 96, we report the average correlation between these two quantities for the models that do not include autoregressive components; the correlations when autoregressive components are estimated are almost identical. The correlation is negative for both linear models with random effects and RE-EM trees, but it is much closer to -1 for linear models with random effects. It seems that the linear model has difficulty distinguishing between the functional variation and the variation in the random effects. While this is not a problem when the random effect can be estimated for an individual using previous observations, it causes problems for prediction for new individuals, as we saw in Tables 82, 83 and 84. This also causes problems when the values of the underlying function, f, are of interest.

I	$E(T_i)$	LME	REEM
50	9.95	-0.7182	-0.5824
50	24.96	-0.6910	-0.5301
50	37.94	-0.7081	-0.5259
100	9.95	-0.7500	-0.4600
100	24.96	-0.8021	-0.4211
100	37.94	-0.7986	-0.4568
200	9.95	-0.7586	-0.3177
200	24.96	-0.8485	-0.3105
200	37.94	-0.9049	-0.3401
400	9.95	-0.7835	-0.2346
400	24.96	-0.8642	-0.2020
400	37.94	-0.9004	-0.2146

Table 94: Average correlation between the errors in estimating the random effects and the errors in estimating the fitted values when the true data generating process is a RE-EM tree.

I	$E(T_i)$	LME	REEM
50	9.95	-0.9687	-0.8057
50	24.96	-0.9961	-0.8519
50	37.94	-0.9976	-0.8632
100	9.95	-0.9844	-0.8131
100	24.96	-0.9975	-0.8684
100	37.94	-0.9985	-0.8923
200	9.95	-0.9875	-0.8183
200	24.96	-0.9981	-0.8942
200	37.94	-0.9987	-0.9102
400	9.95	-0.9891	-0.8507
400	24.96	-0.9984	-0.9062
400	37.94	-0.9990	-0.9292

Table 95: Average correlation between the errors in estimating the random effects and the errors in estimating the fitted values when the true data generating process is a linear model with random effects.

I	$E(T_i)$	LME	REEM
50	9.95	-0.9619	-0.8182
50	24.96	-0.9939	-0.8641
50	37.94	-0.9954	-0.8630
100	9.95	-0.9715	-0.8067
100	24.96	-0.9959	-0.8722
100	37.94	-0.9972	-0.8757
200	9.95	-0.9805	-0.8172
200	24.96	-0.9973	-0.8866
200	37.94	-0.9982	-0.9042
400	9.95	-0.9838	-0.8376
400	24.96	-0.9978	-0.9018
400	37.94	-0.9985	-0.9194

Table 96: Average correlation between the errors in estimating the random effects and the errors in estimating the fitted values when the true data generating process is a non-linear model.

4.6.4 Varying Model Parameters

We now consider the effect of varying the properties of the model in two ways. In our first experiment, we double the standard deviation of the errors, $\hat{\varepsilon}_{it}$. In our second experiment, we add an autoregressive component to the errors, by setting:

$$\begin{aligned} e_{i1} &= \varepsilon_{i1} \\ e_{it} &= \rho e_{i,t-1} + \sqrt{1-\rho^2}\varepsilon_{it} \end{aligned}$$

All of the results in Sections 4.6.2 and 4.6.3 use $\rho = 0$. Here, we use $\rho = 0.5$ and $\rho = 0.9$ to explore the effects of autocorrelation on predictive performance.

When we double the standard deviation of the errors, we find similar results, with the root mean squared errors being higher. (The root mean squared errors are not necessarily doubled, since the effect variance is unchanged.) The success of the models relative to each other is the same, so we do not present the results here.

In Tables 97, 98 and 99, we present the in-sample root mean squared error for the various estimators and data generating processes, holding the average number of observations per individual fixed at 38. For almost all of the estimators, the RMSE declines as ρ increases. This decrease is larger when the RE-EM tree is the true data generating process and quite small when the linear model or the more complicated linear model is the true data generating process. The difference between the RMSE of the estimators that do and do not include the autoregressive component is largest when $\alpha = 0.9$, but it is small even then.

In Tables 100, 101 and 102, we present the in-sample root mean squared error for the various estimators and data generating processes, holding the number of individuals fixed at 100 and allowing ρ and $E(T_i)$ to vary. Again, most of the in-sample RMSE's decline as the autoregressive parameter increases. In the case of

		I			
α	Model Type	50	100	200	400
0.0	LM	91.33	94.20	95.15	94.63
0.0	LME	72.37	73.16	73.50	74.29
0.0	LME-AR	72.64	73.41	73.75	74.61
0.0	RPART	69.60	71.63	71.92	70.62
0.0	RE-EM Tree	62.66	62.58	62.12	60.67
0.0	RE-EM-AR	62.65	62.58	62.12	60.67
0.5	LM	89.15	93.00	95.30	94.96
0.5	LME	71.30	71.46	72.35	73.24
0.5	LME-AR	72.93	72.62	73.31	74.10
0.5	RPART	67.64	71.11	71.41	70.68
0.5	RE-EM Tree	59.93	60.58	60.03	59.09
0.5	RE-EM-AR	60.63	61.23	60.68	59.80
0.9	LM	86.38	91.07	94.55	95.52
0.9	LME	61.96	64.35	64.04	65.12
0.9	LME-AR	67.17	67.73	66.40	66.78
0.9	RPART	62.14	67.75	71.44	70.81
0.9	RE-EM Tree	50.74	50.98	51.45	49.84
0.9	RE-EM-AR	54.13	54.99	55.17	55.49

Table 97: In-sample root mean squared error when the true data generating process is a RE-EM tree as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

α	Model Type	I = 50	100	200	400
0.0	LM	339.47	356.53	365.81	371.12
0.0	LME	74.25	74.47	74.78	74.71
0.0	LME-AR	74.25	74.47	74.78	74.71
0.0	RPART	192.27	241.20	295.44	343.59
0.0	RE-EM Tree	91.60	86.38	81.96	79.20
0.0	RE-EM-AR	91.60	86.39	81.95	79.18
0.5	LM	340.16	357.92	366.03	371.28
0.5	LME	71.88	72.34	72.59	72.58
0.5	LME-AR	72.03	72.45	72.67	72.65
0.5	RPART	187.51	240.38	295.27	343.73
0.5	RE-EM Tree	91.50	84.73	79.53	77.34
0.5	RE-EM-AR	91.18	84.19	79.47	77.41
0.9	LM	338.23	356.40	367.39	372.17
0.9	LME	60.00	60.32	60.68	60.82
0.9	LME-AR	64.39	63.87	62.46	62.49
0.9	RPART	185.60	238.57	294.78	343.17
0.9	RE-EM Tree	78.50	72.48	69.31	66.45
0.9	RE-EM-AR	81.41	73.16	69.63	66.88

Table 98: In-sample root mean squared error when the true data generating process is a RE-EM tree as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

		I			
α	Model Type	50	100	200	400
0.0	LM	325.75	349.04	360.58	370.53
0.0	LME	74.74	74.83	75.13	75.08
0.0	LME-AR	74.74	74.83	75.13	75.08
0.0	RPART	181.51	228.77	281.53	333.18
0.0	RE-EM Tree	89.38	85.33	82.22	79.44
0.0	RE-EM-AR	89.39	85.33	82.19	79.45
0.5	LM	325.02	348.34	361.95	369.05
0.5	LME	72.30	72.61	72.94	72.95
0.5	LME-AR	72.43	72.72	73.03	73.03
0.5	RPART	184.48	227.54	284.90	332.71
0.5	RE-EM Tree	88.36	83.41	80.37	77.62
0.5	RE-EM-AR	88.04	82.94	80.36	77.55
0.9	LM	323.20	351.60	359.52	367.29
0.9	LME	60.65	61.20	61.01	61.44
0.9	LME-AR	63.28	63.64	62.46	62.77
0.9	RPART	175.57	226.27	280.81	330.38
0.9	RE-EM Tree	75.78	73.99	68.73	66.43
0.9	RE-EM-AR	79.69	76.30	68.25	67.05

Table 99: In-sample root mean squared error when the true data generating process is the more complicated model as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

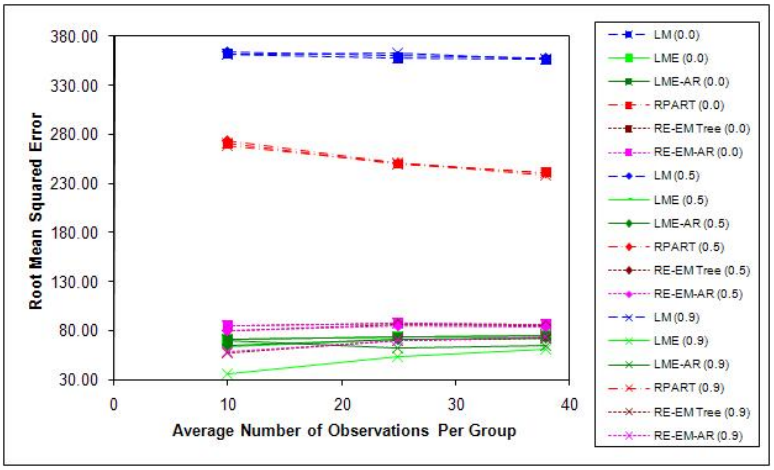

Figure 142: In-sample root mean squared error when the true data generating process is a linear model with random effects. The number of individuals, I, is fixed at 100. The autocorrelation of the errors is given in parentheses next to the estimation method.

the linear model with random effects and an autoregressive component, the reverse holds again. In this case, we see that the LME-AR model performs particularly badly when the number of observations per individual is small, but its performance improves rapidly as $E(T_i)$ increases. The case in which this is most extreme is given in Figure 142, when the true data generating process is a linear model.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	103.56	100.08	94.20
0.0	LME	71.06	73.92	73.16
0.0	LME-AR	71.74	74.15	73.41
0.0	RPART	75.51	73.93	71.63
0.0	RE-EM Tree	62.19	63.95	62.58
0.0	RE-EM-AR	62.42	63.94	62.58
0.5	LM	98.76	95.69	93.00
0.5	LME	65.18	71.41	71.46
0.5	LME-AR	70.40	72.76	72.62
0.5	RPART	71.94	72.80	71.11
0.5	RE-EM Tree	54.38	60.92	60.58
0.5	RE-EM-AR	58.80	61.84	61.23
0.9	LM	101.06	96.45	91.07
0.9	LME	51.00	60.64	64.35
0.9	LME-AR	56.21	64.70	67.73
0.9	RPART	68.60	69.13	67.75
0.9	RE-EM Tree	37.00	47.62	50.98
0.9	RE-EM-AR	44.63	52.35	54.99

Table 100: In-sample root mean squared error when the true data generating process is a RE-EM tree as α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	362.10	358.22	356.53
0.0	LME	70.54	73.75	74.47
0.0	LME-AR	70.56	73.75	74.47
0.0	RPART	271.01	249.65	241.20
0.0	RE-EM Tree	85.29	87.43	86.38
0.0	RE-EM-AR	85.26	87.52	86.39
0.5	LM	364.28	359.86	357.92
0.5	LME	63.04	70.68	72.34
0.5	LME-AR	64.85	70.85	72.45
0.5	RPART	273.27	250.61	240.38
0.5	RE-EM Tree	79.91	85.85	84.73
0.5	RE-EM-AR	80.30	85.21	84.19
0.9	LM	361.62	363.48	356.40
0.9	LME	36.21	53.23	60.32
0.9	LME-AR	70.11	61.71	63.87
0.9	RPART	268.06	250.28	238.57
0.9	RE-EM Tree	57.52	70.39	72.48
0.9	RE-EM-AR	58.59	70.10	73.16

Table 101: In-sample root mean squared error when the true data generating process is a linear model with random effects where α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	354.40	348.77	349.04
0.0	LME	71.33	74.17	74.83
0.0	LME-AR	71.35	74.18	74.83
0.0	RPART	264.03	236.78	228.77
0.0	RE-EM Tree	84.32	87.16	85.33
0.0	RE-EM-AR	84.35	87.27	85.33
0.5	LM	354.62	352.66	348.34
0.5	LME	63.88	71.09	72.61
0.5	LME-AR	64.94	71.26	72.72
0.5	RPART	265.97	237.95	227.54
0.5	RE-EM Tree	77.06	84.52	83.41
0.5	RE-EM-AR	77.54	84.50	82.94
0.9	LM	356.74	354.42	351.60
0.9	LME	38.65	54.55	61.20
0.9	LME-AR	52.28	59.73	63.64
0.9	RPART	264.38	240.90	226.27
0.9	RE-EM Tree	58.04	70.14	73.99
0.9	RE-EM-AR	60.09	70.91	76.30

Table 102: In-sample root mean squared error when the true data generating process is the more complicated model where α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

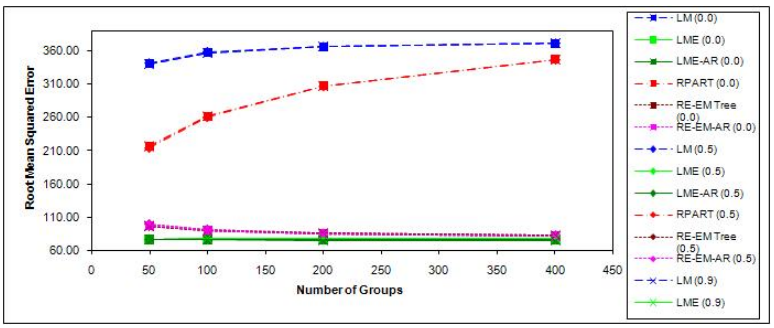

Figure 143: Root mean squared error of prediction for future observations of the individuals included in the sample when the true data generating process is a linear model with random effects. The average number of observations per individual is fixed at 38. The autocorrelation of the errors is given in parentheses next to the estimation method.

In Tables 103, 104 and 105, we present the root mean squared error of the predictions for future observations of individuals in the sample, holding the average number of observations per individual fixed. Figure 143 presents the root mean squared errors graphically when the data generating process is a linear model. Tables 106, 107 and 108 give the root mean squared error holding the number of individuals fixed instead; Figure 144 presents the same information graphically when the data generating process is a linear model. The patterns that we saw for the in-sample root mean squared errors generally continue to hold. When the data generating process is a RE-EM tree, the effect of changing ρ is most pronounced. The linear model with random effects and an autoregressive component again performs badly when $\rho = 0.9$ and the number of observations per individual is small.

Next, we consider the root mean squared error of prediction for individuals not in the sample. The resulting RMSE's when the average number of observations is

		I			
α	Model Type	50	100	200	400
0.0	LM	98.09	98.80	98.30	98.29
0.0	LME	87.80	86.95	86.47	87.22
0.0	LME-AR	88.06	87.24	86.77	87.61
0.0	RPART	78.94	76.94	74.32	71.35
0.0	RE-EM Tree	74.07	69.77	66.39	63.08
0.0	RE-EM-AR	74.04	69.77	66.39	63.08
0.5	LM	96.65	98.32	98.77	98.48
0.5	LME	88.34	87.04	89.37	88.91
0.5	LME-AR	88.81	87.36	89.36	89.23
0.5	RPART	77.85	76.59	74.86	71.64
0.5	RE-EM Tree	73.29	70.18	67.51	64.08
0.5	RE-EM-AR	72.93	69.70	67.24	63.81
0.9	LM	93.99	97.16	97.78	99.38
0.9	LME	86.56	87.25	86.24	88.02
0.9	LME-AR	85.06	85.75	85.22	86.87
0.9	RPART	78.26	76.20	75.78	72.14
0.9	RE-EM Tree	74.06	69.64	68.63	64.48
0.9	RE-EM-AR	72.64	67.59	66.18	62.41

Table 103: Root mean squared error of prediction for future observations of individuals in the sample when the true data generating process is a RE-EM tree as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

		I			
α	Model Type	50	100	200	400
0.0	LM	339.79	356.58	365.47	370.42
0.0	LME	76.84	76.91	77.06	77.19
0.0	LME-AR	76.84	76.92	77.06	77.19
0.0	RPART	217.09	261.91	306.45	346.49
0.0	RE-EM Tree	97.49	90.69	85.40	82.21
0.0	RE-EM-AR	97.54	90.73	85.41	82.17
0.5	LM	341.15	357.94	365.64	370.69
0.5	LME	77.70	78.23	78.20	78.29
0.5	LME-AR	77.30	77.90	77.87	78.00
0.5	RPART	214.90	260.73	307.00	346.90
0.5	RE-EM Tree	100.75	92.56	86.29	83.39
0.5	RE-EM-AR	100.10	91.61	85.92	83.15
0.9	LM	339.53	356.77	367.17	371.54
0.9	LME	77.27	78.13	78.40	78.40
0.9	LME-AR	75.98	76.22	74.98	74.88
0.9	RPART	215.18	260.96	307.11	346.59
0.9	RE-EM Tree	97.42	90.92	86.69	83.68
0.9	RE-EM-AR	97.67	88.74	84.35	81.36

Table 104: Root mean squared error of prediction for future observations of individuals in the sample when the true data generating process is a RE-EM tree as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

		I			
α	Model Type	50	100	200	400
0.0	LM	325.51	348.82	359.65	369.96
0.0	LME	77.04	77.50	77.59	77.80
0.0	LME-AR	77.04	77.50	77.59	77.80
0.0	RPART	208.19	252.51	295.91	338.76
0.0	RE-EM Tree	95.81	90.01	86.07	83.14
0.0	RE-EM-AR	95.82	89.98	86.05	83.15
0.5	LM	325.34	347.94	361.28	368.30
0.5	LME	78.57	78.47	78.76	78.71
0.5	LME-AR	78.18	78.13	78.46	78.43
0.5	RPART	212.16	249.50	299.65	338.28
0.5	RE-EM Tree	98.53	90.99	87.10	83.87
0.5	RE-EM-AR	97.75	90.19	86.80	83.51
0.9	LM	323.65	351.38	358.76	366.56
0.9	LME	77.78	78.48	78.63	78.68
0.9	LME-AR	75.40	76.21	75.50	75.48
0.9	RPART	204.70	251.72	296.19	335.83
0.9	RE-EM Tree	96.07	91.92	86.56	83.88
0.9	RE-EM-AR	97.06	91.49	83.42	81.68

Table 105: Root mean squared error of prediction for future observations of individuals in the sample when the true data generating process is the more complicated model as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	94.57	93.55	98.80
0.0	LME	88.84	79.80	86.95
0.0	LME-AR	88.37	79.82	87.24
0.0	RPART	79.72	75.23	76.94
0.0	RE-EM Tree	75.29	68.41	69.77
0.0	RE-EM-AR	75.27	68.41	69.77
0.5	LM	94.21	92.47	98.32
0.5	LME	88.17	78.83	87.04
0.5	LME-AR	86.35	78.78	87.36
0.5	RPART	78.96	76.47	76.59
0.5	RE-EM Tree	74.69	70.29	70.18
0.5	RE-EM-AR	73.44	69.63	69.70
0.9	LM	93.02	89.90	97.16
0.9	LME	78.31	77.28	87.25
0.9	LME-AR	76.87	75.95	85.75
0.9	RPART	76.90	74.71	76.20
0.9	RE-EM Tree	63.64	67.26	69.64
0.9	RE-EM-AR	62.58	64.39	67.59

Table 106: Root mean squared error of prediction for future observations of individuals in the sample when the true data generating process is a RE-EM tree as α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	362.98	358.44	356.58
0.0	LME	80.57	77.34	76.91
0.0	LME-AR	80.61	77.34	76.92
0.0	RPART	314.46	273.78	261.91
0.0	RE-EM Tree	99.28	92.24	90.69
0.0	RE-EM-AR	99.26	92.34	90.73
0.5	LM	365.45	359.61	357.94
0.5	LME	81.35	78.90	78.23
0.5	LME-AR	80.91	78.37	77.90
0.5	RPART	316.54	274.86	260.73
0.5	RE-EM Tree	101.20	94.97	92.56
0.5	RE-EM-AR	100.11	93.75	91.61
0.9	LM	363.23	363.83	356.77
0.9	LME	58.88	75.49	78.13
0.9	LME-AR	83.30	75.63	76.22
0.9	RPART	311.91	276.61	260.96
0.9	RE-EM Tree	81.45	91.39	90.92
0.9	RE-EM-AR	79.69	87.71	88.74

Table 107: Root mean squared error of prediction for future observations of individuals in the sample when the true data generating process is a linear model with random effects where α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	354.80	348.21	348.82
0.0	LME	81.22	77.79	77.50
0.0	LME-AR	81.25	77.79	77.50
0.0	RPART	304.76	263.50	252.51
0.0	RE-EM Tree	99.31	92.42	90.01
0.0	RE-EM-AR	99.51	92.56	89.98
0.5	LM	354.47	352.30	347.94
0.5	LME	82.06	79.53	78.47
0.5	LME-AR	80.78	79.01	78.13
0.5	RPART	306.79	264.02	249.50
0.5	RE-EM Tree	98.94	93.73	90.99
0.5	RE-EM-AR	97.67	93.06	90.19
0.9	LM	356.93	354.62	351.38
0.9	LME	61.11	76.19	78.48
0.9	LME-AR	67.60	74.49	76.21
0.9	RPART	307.59	267.39	251.72
0.9	RE-EM Tree	82.91	91.38	91.92
0.9	RE-EM-AR	81.78	88.34	91.49

Table 108: Root mean squared error of prediction for future observations of individuals in the sample when the true data generating process is the more complicated model where α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

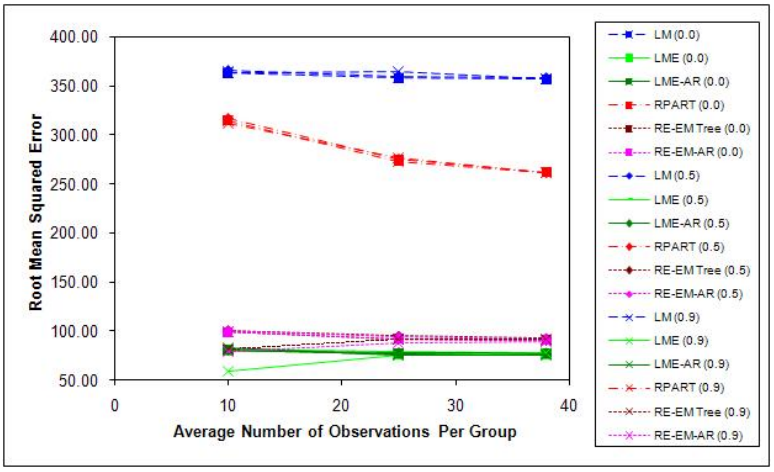

Figure 144: Root mean squared error of prediction for future observations of the individuals included in the sample when the true data generating process is a linear model with random effects. The number of individuals, I, is fixed at 100. The autocorrelation of the errors is given in parentheses next to the estimation method.

fixed are given in Tables 109, 110 and 111 and in Figure 145. Tables 112, 113 and 114, as well as Figure 146, give the root mean squared errors when the number of individuals is fixed. The linear model with random effects continues to have the largest root mean squared errors of all the predictors, and its root mean squared error increases as ρ increases. This suggests that autocorrelation exacerbates the problems that the linear model has in distinguishing between the random effects and the fixed function. The linear model with random effects that allows for autocorrelation has root mean squared errors that are similar to or slightly worse than the linear model with random effects that does not allow for autocorrelation. The performance of the linear model without random effects and of all of the tree models are not greatly affected by the autoregressive process in the errors.

		I			
α	Model Type	50	100	200	400
0.0	LM	97.81	91.28	95.16	94.41
0.0	LME	169.15	180.68	214.96	182.10
0.0	LME-AR	165.39	175.52	204.20	171.50
0.0	RPART	94.24	81.29	76.74	73.85
0.0	RE-EM Tree	89.36	79.48	76.32	74.06
0.0	RE-EM-AR	89.36	79.48	76.32	74.06
0.5	LM	96.19	96.35	95.68	94.05
0.5	LME	163.30	192.73	193.75	178.56
0.5	LME-AR	160.94	189.08	188.13	173.37
0.5	RPART	89.94	85.47	78.47	72.79
0.5	RE-EM Tree	85.50	84.41	77.91	72.78
0.5	RE-EM-AR	85.48	84.16	77.83	72.72
0.9	LM	94.57	97.79	91.71	94.60
0.9	LME	152.60	165.58	216.60	225.73
0.9	LME-AR	153.80	176.00	221.40	231.62
0.9	RPART	95.41	89.28	78.03	75.24
0.9	RE-EM Tree	87.63	86.31	77.60	75.17
0.9	RE-EM-AR	88.08	86.58	77.28	75.01

Table 109: Root mean squared error of prediction for new individuals when the true data generating process is a RE-EM tree as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

α	Model Type	I = 50	100	200	400
0.0	LM	388.99	374.03	366.47	361.90
0.0	LME	486.33	480.44	501.59	501.69
0.0	LME-AR	486.35	480.42	501.59	501.69
0.0	RPART	466.38	453.98	426.06	393.39
0.0	RE-EM Tree	387.46	377.03	371.75	367.12
0.0	RE-EM-AR	387.44	377.05	371.67	367.05
0.5	LM	385.09	374.83	364.69	366.76
0.5	LME	467.94	491.80	476.45	493.21
0.5	LME-AR	475.44	499.55	483.11	499.66
0.5	RPART	468.40	453.76	423.49	397.45
0.5	RE-EM Tree	385.08	378.73	368.43	370.08
0.5	RE-EM-AR	384.38	378.33	368.08	369.76
0.9	LM	385.15	372.09	365.53	365.91
0.9	LME	491.36	498.25	498.49	499.75
0.9	LME-AR	513.85	517.04	512.91	512.89
0.9	RPART	467.43	450.34	425.26	399.38
0.9	RE-EM Tree	385.95	374.86	371.17	370.69
0.9	RE-EM-AR	387.08	374.55	370.16	369.93

Table 110: Root mean squared error of prediction for new individuals when the true data generating process is a linear random effects model as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

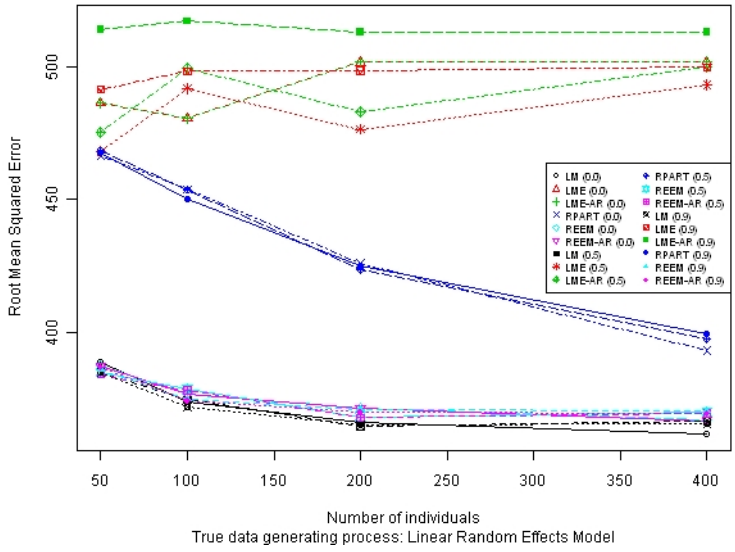

Figure 145: Root mean squared error of prediction for new individuals when the true data generating process is a linear model with random effects. The average number of observations per individual is fixed at 38. The autocorrelation of the errors is given in parentheses next to the estimation method.

		I			
α	Model Type	50	100	200	400
0.0	LM	376.90	363.39	354.47	350.17
0.0	LME	462.49	456.98	484.45	478.39
0.0	LME-AR	462.47	456.93	484.38	478.33
0.0	RPART	457.94	445.31	422.96	390.44
0.0	RE-EM Tree	378.06	365.42	358.49	353.36
0.0	RE-EM-AR	377.94	365.37	358.49	353.34
0.5	LM	374.74	358.51	354.98	350.64
0.5	LME	469.61	483.51	477.02	480.54
0.5	LME-AR	477.17	489.52	481.57	483.88
0.5	RPART	453.04	437.49	419.97	393.96
0.5	RE-EM Tree	373.90	361.02	358.47	354.67
0.5	RE-EM-AR	373.08	360.82	358.12	354.26
0.9	LM	380.67	361.42	352.68	355.77
0.9	LME	478.61	469.38	476.02	507.43
0.9	LME-AR	498.23	484.66	488.88	517.44
0.9	RPART	457.38	444.45	419.14	393.10
0.9	RE-EM Tree	381.79	363.29	356.78	361.14
0.9	RE-EM-AR	382.42	363.58	356.35	360.81

Table 111: Root mean squared error of prediction for new individuals when the true data generating process is the more complicated model as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	108.86	102.00	91.28
0.0	LME	110.06	190.25	180.68
0.0	LME-AR	109.74	188.16	175.52
0.0	RPART	92.38	86.83	81.29
0.0	RE-EM Tree	90.26	85.40	79.48
0.0	RE-EM-AR	90.22	85.39	79.48
0.5	LM	107.23	105.84	96.35
0.5	LME	107.31	225.40	192.73
0.5	LME-AR	107.06	225.91	189.08
0.5	RPART	89.20	94.93	85.47
0.5	RE-EM Tree	87.11	93.21	84.41
0.5	RE-EM-AR	87.05	93.70	84.16
0.9	LM	110.92	102.27	97.79
0.9	LME	113.37	204.48	165.58
0.9	LME-AR	112.01	206.14	176.00
0.9	RPART	97.27	94.13	89.28
0.9	RE-EM Tree	91.89	90.58	86.31
0.9	RE-EM-AR	91.92	90.69	86.58

Table 112: Root mean squared error of prediction for new individuals when the true data generating process is a RE-EM tree as α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	379.54	378.83	374.03
0.0	LME	425.21	473.14	480.44
0.0	LME-AR	425.05	473.14	480.42
0.0	RPART	449.48	455.95	453.98
0.0	RE-EM Tree	382.34	383.75	377.03
0.0	RE-EM-AR	382.26	383.85	377.05
0.5	LM	376.06	375.18	374.83
0.5	LME	437.34	479.80	491.80
0.5	LME-AR	445.66	488.41	499.55
0.5	RPART	447.36	452.67	453.76
0.5	RE-EM Tree	378.81	379.17	378.73
0.5	RE-EM-AR	378.87	378.93	378.33
0.9	LM	377.23	376.46	372.09
0.9	LME	468.48	504.57	498.25
0.9	LME-AR	486.12	519.14	517.04
0.9	RPART	449.51	456.65	450.34
0.9	RE-EM Tree	378.42	384.62	374.86
0.9	RE-EM-AR	379.31	384.55	374.55

Table 113: Root mean squared error of prediction for new individuals when the true data generating process is a linear model with random effects where α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	370.44	362.82	363.39
0.0	LME	404.33	474.06	456.98
0.0	LME-AR	404.12	473.94	456.93
0.0	RPART	442.36	442.85	445.31
0.0	RE-EM Tree	373.53	368.70	365.42
0.0	RE-EM-AR	373.53	368.57	365.37
0.5	LM	372.89	362.43	358.51
0.5	LME	427.07	480.44	483.51
0.5	LME-AR	432.01	488.62	489.52
0.5	RPART	441.66	444.88	437.49
0.5	RE-EM Tree	375.98	367.55	361.02
0.5	RE-EM-AR	375.44	367.53	360.82
0.9	LM	372.38	363.97	361.42
0.9	LME	465.84	475.49	469.38
0.9	LME-AR	481.35	493.68	484.66
0.9	RPART	449.21	445.06	444.45
0.9	RE-EM Tree	377.06	367.81	363.29
0.9	RE-EM-AR	377.61	368.07	363.58

Table 114: Root mean squared error of prediction for new individuals when the true data generating process is the more complicated model where α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

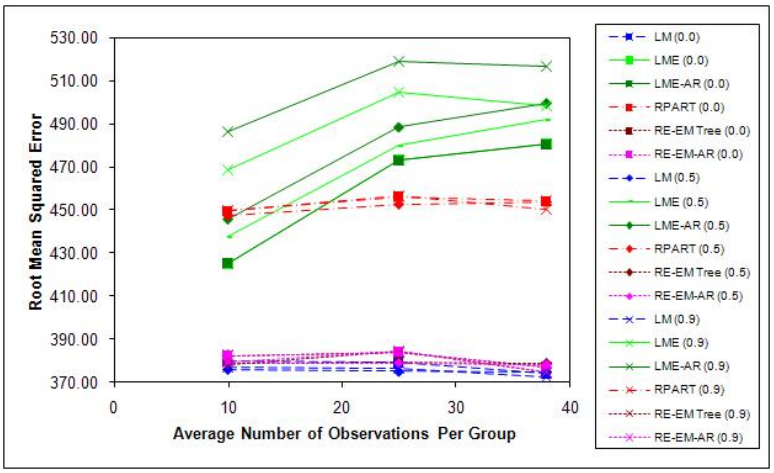

Figure 146: Root mean squared error of prediction for new individuals when the true data generating process is a linear model with random effects. The number of individuals, I, is fixed at 100. The autocorrelation of the errors is given in parentheses next to the estimation method.

Finally, we present the root mean squared errors for future observations for individuals not in the sample when we fix the average number of observations (Tables 115, 116 and 117) or the number of individuals (Tables 118, 119 and 120). The resulting patterns are similar to those of future observations for individuals in the sample.

Overall, we find that including autocorrelation or changing the error variance does not affect our previous conclusions that the RE-EM tree performs better than or almost as well as the best estimator in a wide variety of cases.

α	Model Type	I 50	100	200	400
0.0	LM	99.87	96.87	96.87	95.49
0.0	LME	90.86	86.54	86.83	86.23
0.0	LME-AR	91.21	86.91	87.18	86.74
0.0	RPART	93.31	81.87	76.49	72.11
0.0	RE-EM Tree	77.35	71.53	67.36	63.84
0.0	RE-EM-AR	77.36	71.53	67.36	63.84
0.5	LM	101.90	97.67	99.87	96.05
0.5	LME	91.86	88.08	90.80	87.85
0.5	LME-AR	93.18	89.09	91.90	89.04
0.5	RPART	91.64	83.25	80.49	72.46
0.5	RE-EM Tree	78.57	74.03	71.45	64.80
0.5	RE-EM-AR	78.59	73.71	71.34	64.70
0.9	LM	97.59	98.19	95.38	98.03
0.9	LME	88.03	90.65	86.26	87.54
0.9	LME-AR	89.83	91.46	87.53	88.59
0.9	RPART	95.26	86.86	78.12	74.38
0.9	RE-EM Tree	77.16	75.95	68.93	66.55
0.9	RE-EM-AR	77.58	75.66	68.53	66.09

Table 115: Root mean squared error of prediction for future observations for individuals not in the original sample. The true data generating process is a RE-EM tree as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

		I			
α	Model Type	50	100	200	400
0.0	LM	387.42	373.47	365.28	360.50
0.0	LME	76.81	76.19	76.36	76.55
0.0	LME-AR	76.81	76.19	76.36	76.55
0.0	RPART	465.31	453.25	424.33	391.76
0.0	RE-EM Tree	96.15	88.61	84.33	82.74
0.0	RE-EM-AR	96.13	88.66	84.38	82.72
0.5	LM	383.93	373.29	362.96	365.49
0.5	LME	77.87	77.88	77.17	77.71
0.5	LME-AR	77.60	77.75	77.07	77.66
0.5	RPART	467.11	451.80	421.55	395.86
0.5	RE-EM Tree	97.43	90.47	85.02	83.33
0.5	RE-EM-AR	97.27	90.35	85.04	83.36
0.9	LM	384.18	371.64	363.94	365.21
0.9	LME	78.18	77.56	76.96	77.42
0.9	LME-AR	79.65	78.50	76.85	77.36
0.9	RPART	466.53	448.87	423.77	397.88
0.9	RE-EM Tree	96.27	88.70	83.88	82.70
0.9	RE-EM-AR	104.17	90.38	83.46	82.65

Table 116: Root mean squared error of prediction for future observations for individuals not in the original sample. The true data generating process is a RE-EM tree as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

		I			
α	Model Type	50	100	200	400
0.0	LM	376.39	361.67	353.05	348.45
0.0	LME	78.06	77.80	77.10	76.67
0.0	LME-AR	78.06	77.80	77.10	76.67
0.0	RPART	455.80	443.32	421.63	388.52
0.0	RE-EM Tree	96.70	90.64	84.64	81.73
0.0	RE-EM-AR	96.68	90.66	84.60	81.73
0.5	LM	372.42	357.43	353.85	348.67
0.5	LME	78.50	78.86	78.24	78.56
0.5	LME-AR	78.24	78.72	78.14	78.50
0.5	RPART	451.79	435.66	418.80	391.14
0.5	RE-EM Tree	95.20	90.94	86.07	84.22
0.5	RE-EM-AR	94.72	90.95	85.87	84.07
0.9	LM	378.87	360.18	351.28	354.36
0.9	LME	78.83	78.68	78.03	78.07
0.9	LME-AR	79.22	78.83	77.91	78.02
0.9	RPART	455.07	442.45	416.37	391.87
0.9	RE-EM Tree	100.56	91.81	84.32	82.81
0.9	RE-EM-AR	113.23	96.61	84.47	82.71

Table 117: Root mean squared error of prediction for future observations for individuals not in the original sample. The true data generating process is the more complicated model as α and I vary. The expected number of observations per individual, $E(T_i)$, is fixed at 38. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	94.41	95.92	96.87
0.0	LME	89.89	83.24	86.54
0.0	LME-AR	89.87	83.42	86.91
0.0	RPART	82.27	85.75	81.87
0.0	RE-EM Tree	75.03	75.74	71.53
0.0	RE-EM-AR	75.09	75.71	71.53
0.5	LM	95.77	99.09	97.67
0.5	LME	91.16	88.37	88.08
0.5	LME-AR	91.04	88.79	89.09
0.5	RPART	82.32	92.85	83.25
0.5	RE-EM Tree	76.11	82.19	74.03
0.5	RE-EM-AR	74.71	82.00	73.71
0.9	LM	102.08	97.90	98.19
0.9	LME	83.18	82.79	90.65
0.9	LME-AR	83.11	83.34	91.46
0.9	RPART	91.40	93.10	86.86
0.9	RE-EM Tree	68.67	76.57	75.95
0.9	RE-EM-AR	68.68	76.77	75.66

Table 118: Root mean squared error of prediction for future observations for individuals not in the original sample. The true data generating process is a RE-EM tree where α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	375.42	377.09	373.47
0.0	LME	79.70	77.13	76.19
0.0	LME-AR	79.71	77.13	76.19
0.0	RPART	447.52	454.41	453.25
0.0	RE-EM Tree	93.09	92.08	88.61
0.0	RE-EM-AR	93.04	92.07	88.66
0.5	LM	371.74	374.21	373.29
0.5	LME	80.71	78.71	77.88
0.5	LME-AR	81.23	78.54	77.75
0.5	RPART	444.12	451.11	451.80
0.5	RE-EM Tree	94.31	91.44	90.47
0.5	RE-EM-AR	94.67	91.06	90.35
0.9	LM	372.90	374.70	371.64
0.9	LME	58.91	74.93	77.56
0.9	LME-AR	69.00	78.55	78.50
0.9	RPART	446.66	454.61	448.87
0.9	RE-EM Tree	73.60	88.49	88.70
0.9	RE-EM-AR	75.15	89.47	90.38

Table 119: Root mean squared error of prediction for future observations for individuals not in the original sample. The true data generating process is a linear model with random effects where α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

		$E(T_i)$		
α	Model Type	9.95	24.96	37.94
0.0	LM	365.10	361.18	361.67
0.0	LME	80.91	78.02	77.80
0.0	LME-AR	80.92	78.02	77.80
0.0	RPART	439.32	440.59	443.32
0.0	RE-EM Tree	94.52	90.71	90.64
0.0	RE-EM-AR	94.36	90.84	90.66
0.5	LM	367.67	361.18	357.43
0.5	LME	80.77	80.05	78.86
0.5	LME-AR	80.71	79.78	78.72
0.5	RPART	437.56	441.91	435.66
0.5	RE-EM Tree	93.23	94.42	90.94
0.5	RE-EM-AR	93.19	94.42	90.95
0.9	LM	366.72	362.11	360.18
0.9	LME	60.05	75.27	78.68
0.9	LME-AR	65.07	77.21	78.83
0.9	RPART	441.84	443.67	442.45
0.9	RE-EM Tree	76.27	87.14	91.81
0.9	RE-EM-AR	77.55	88.69	96.61

Table 120: Root mean squared error of prediction for future observations for individuals not in the original sample. The true data generating process is the more complicated model where α and $E(T_i)$ vary. The number of individuals, I, is fixed at 100. Model type is the type of model fitted to the data.

4.6.5 Stability of Tree Estimates

We assess the stability of our tree estimates by starting our estimation with alternative initial values for the random effects. The results we have presented so far fit RE-EM trees with initial values of 0 for all of the random effects. Here, we fit trees in which the initial values for the random effects are the estimated random effects from the first tree in random order. We also fit RE-EM trees where the initial values are the estimated random effects in reverse order, so that the group that had the largest (most positive) estimated random effect has an initial random effect value equal to the smallest (most negative) random effect. As additional comparisons, we fit trees using maximum likelihood instead of restricted maximum likelihood when we estimate the linear model and using an alternative optimization method in fitting the maximum likelihood. In this section, we also fit trees using Method 1. We report the root mean squared error between the fitted values of each tree estimated with these alternative methods and the fitted values of the original tree for the three data generating processes, relative to the in-sample RMSE of the baseline method in Tables 121, 122, and 123.

These tables show that the estimated fitted values are generally similar across the different estimation possibilities. Changing the estimation method has the largest impact on the estimated values, with the relative root mean squared errors exceeding 1 in some cases when Method 1 is used and the true data generating process is a linear model. This effect generally declines with sample size. Changing the initial values of the random effects has a smaller impact, and the relative RMSE between fitted values with different initial values declines steadily as the sample size grows. The change in estimates based on using maximum likelihood instead of REML to estimate the random effects is even smaller. There was almost no change in estimates when an alternative optimization method is used for estimating

I	$E(T_i)$	Random Initial Values	Reverse Initial Values	ML	Optim	Method 1
50	9.95	0.2879	0.3115	0.0829	0.0007	0.2484
50	24.96	0.2363	0.2706	0.0419	0.0000	0.1802
50	37.94	0.1858	0.2162	0.0306	0.0000	0.2635
100	9.95	0.2083	0.2551	0.0235	0.0001	0.1522
100	24.96	0.1523	0.2003	0.0103	0.0000	0.1077
100	37.94	0.1553	0.1985	0.0110	0.0000	0.2176
200	9.95	0.1239	0.1791	0.0059	0.0000	0.0872
200	24.96	0.0850	0.1321	0.0020	0.0000	0.0502
200	37.94	0.0744	0.1156	0.0018	0.0000	0.1080
400	9.95	0.0861	0.1542	0.0022	0.0000	0.0554
400	24.96	0.0519	0.1102	0.0006	0.0000	0.0228
400	37.94	0.0417	0.0879	0.0005	0.0000	0.0777

Table 121: RMSE between fitted values of original RE-EM tree and fitted values of RE-EM trees with alternative initial values or linear model estimation methods, relative to the in-sample RMSE of the baseline process, when the true data generating process is a RE-EM tree.

the linear model; the relative RMSE is under 0.0001 for all of the different data generating processes and parameter configurations. This shows that the initial values or estimation method chosen to fit a RE-EM tree have a limited effect on the fitted values, especially for larger sample sizes. Also, the choices in fitting the linear model, such as whether to use REML or maximum likelihood have very small effects.

I	$E(T_i)$	Random Initial Values	Reverse Initial Values	ML	Optim	Method 1
50	9.95	0.8344	0.8578	0.2315	0.0000	0.7507
50	24.96	0.7072	0.7053	0.1854	0.0000	0.7234
50	37.94	0.6465	0.6052	0.1722	0.0000	1.1293
100	9.95	0.6515	0.6577	0.2042	0.0000	0.6418
100	24.96	0.5426	0.5273	0.1397	0.0000	0.6112
100	37.94	0.4619	0.4422	0.0999	0.0000	1.2142
200	9.95	0.4956	0.4945	0.1484	0.0000	0.5207
200	24.96	0.3943	0.3621	0.0606	0.0000	0.4736
200	37.94	0.3274	0.2915	0.0435	0.0000	1.1737
400	9.95	0.2602	0.2846	0.0462	0.0000	0.3359
400	24.96	0.2303	0.2277	0.0149	0.0000	0.3223
400	37.94	0.2064	0.1880	0.0158	0.0000	0.8628

Table 122: RMSE between fitted values of original RE-EM tree and fitted values of RE-EM trees with alternative initial values or linear model estimation methods, relative to the in-sample RMSE of the baseline process, when the true data generating process is a linear model.

I	$E(T_i)$	Random Initial Values	Reverse Initial Values	ML	Optim	Method 1
50	9.95	0.7743	0.7882	0.1924	0.0000	0.7227
50	24.96	0.6984	0.6971	0.1708	0.0000	0.7084
50	37.94	0.5687	0.5895	0.1369	0.0000	1.0863
100	9.95	0.6330	0.6298	0.1565	0.0000	0.6204
100	24.96	0.5219	0.5056	0.1096	0.0000	0.6138
100	37.94	0.4496	0.4356	0.0892	0.0000	1.1574
200	9.95	0.4784	0.4882	0.1157	0.0000	0.5233
200	24.96	0.3795	0.3617	0.0628	0.0000	0.4996
200	37.94	0.3214	0.3038	0.0424	0.0000	1.1637
400	9.95	0.2689	0.3061	0.0481	0.0000	0.3673
400	24.96	0.2406	0.2397	0.0205	0.0000	0.3524
400	37.94	0.2106	0.2025	0.0132	0.0000	0.9425

Table 123: RMSE between fitted values of original RE-EM tree and fitted values of RE-EM trees with alternative initial values or linear model estimation methods, relative to the in-sample RMSE of the baseline process, when the true data generating process is a non-linear model.

4.6.6 Performance in balanced panels

We now describe the performance of trees with and without random effects, linear models with and without random effects, MVPART, and individual-specific regressions and regression trees in balanced panels. As in Section 4.6.2, we will compare the root mean squared errors in-sample and for different types of predictions.

We first discuss the in-sample root mean squared errors for each method when the true data generating processes are a RE-EM tree, a linear model with random effects, and the more complicated model, respectively, omitting detailed results to save space. In all cases, fitting individual linear regressions (for $T = 10$) or individual regression trees (for $T = 25, 50, 100$) has the lowest RMSE in-sample. Of those estimation methods that fit a single model to the full sample, the RE-EM tree has the lowest in-sample RMSE when the true data generating process is a RE-EM tree. When the true process is a linear model or a linear model with quadratic terms, the linear model with random effects has the lowest in-sample root mean squared error for $T = 10$ and the RE-EM tree has the lowest in-sample RMSE for $T = 25, 50, 100$. The linear model without random effects has the highest RMSE in all cases. The RMSE is generally constant or decreasing as a function of the number of observations per group, except for individual linear regressions. This occurs in part because of the larger total number of observations across all individuals.

Next, we consider the prediction error for future observations for individuals that are already in the sample, reported in Table 124 when the true data generating process is a RE-EM tree and in Table 125 when the true data generating process is a linear random effects model. Because MVPART cannot be used for predicting future observations of individuals in the sample, it does not appear in the tables. The results for the more complicated model are similar to those for the linear

model, so they are omitted. The RE-EM tree has the lowest RMSE when it is the true data generating process. The linear model with random effects has the lowest RMSE for the other two data generating processes when $T = 10, 25$, or 50, while the RE-EM tree performs better for $T = 100$. The difference between the root mean squared errors is insignificant when $T = 10$ and $I = 400, 2000$ and significant otherwise. When the RE-EM tree is the true data generating process, a regression tree without random effects outperforms the linear models, presumably because the tree is closer to the true data generating process. In all cases, fitting separate linear models leads to very poor predictive performance, despite its sometimes good in-sample performance. Fitting separate regression trees leads to the second-worst predictive performance when the true data generating process is a RE-EM tree (again despite good in-sample performance), but performs well when the true process is a linear random effects model when T and I are both small. In the latter case, the individual regression trees split very few times, using little information about the covariates. Furthermore, individual regressions or regression trees do not benefit from increases in I because they do not use the additional information provided by observations from other individuals.

Next, we consider predictions for observations of new individuals. Because individual-specific linear regressions and regression trees cannot be used for predictions for new individuals, no results for them can be included in these tables. When the true data generating process is a RE-EM tree, prediction using RE-EM trees generally has the lowest mean squared errors, though the regression tree without random effects performs about as well. When the true process is a linear model with random effects, linear regression and LME perform best for small values of I and T, while the RE-EM tree performs best in most cases where I is at least 400. Full results are given in Table 126 for the RE-EM tree and Table 127 for the linear

I	T	LM	LM-Ind	LME	RPART	RPART-Ind	REEM
50	10	90.22	1.29E+06	82.64	76.80	95.42	72.36
50	25	100.04	1.49E+05	94.84	80.42	97.47	74.99
50	50	105.88	1.90E+04	99.63	76.08	93.56	69.61
50	100	105.57	5.15E+03	102.88	76.71	118.84	69.60
100	10	90.98	2.07E+06	83.94	73.93	94.88	68.47
100	25	97.44	2.84E+05	91.46	76.10	98.75	70.47
100	50	106.29	2.79E+04	100.41	72.21	92.12	65.52
100	100	99.01	5.82E+03	97.23	71.70	112.99	64.24
200	10	90.82	2.34E+06	84.49	70.46	96.82	64.99
200	25	95.13	2.77E+05	90.11	70.47	98.37	64.04
200	50	104.99	4.10E+04	99.27	69.69	93.09	62.35
200	100	107.44	6.21E+03	104.40	69.03	117.15	61.16
400	10	89.89	2.42E+06	82.58	70.63	99.78	64.89
400	25	97.29	3.07E+05	92.44	73.64	101.47	66.97
400	50	103.59	4.30E+04	98.15	68.51	92.35	61.30
400	100	103.77	6.81E+03	100.88	68.56	116.99	60.83
1000	10	89.72	2.70E+06	83.46	70.32	97.37	64.56
1000	25	96.27	3.16E+05	91.32	73.15	101.23	66.68
1000	50	102.51	5.27E+04	97.29	68.68	93.08	61.39
1000	100	107.28	6.51E+03	104.11	68.33	115.12	60.62
2000	10	89.77	2.72E+06	83.61	70.46	96.36	64.79
2000	25	96.93	3.43E+05	91.62	72.27	102.43	66.13
2000	50	105.09	5.50E+04	99.08	68.52	93.17	61.07

Table 124: Out-of-sample root mean squared prediction errors for future observations when the true data generating process is a RE-EM tree with a balanced panel.

I	T	LM	LM-Ind	LME	RPART	RPART-Ind	REEM
50	10	354.69	1.13E+06	93.06	305.60	100.69	133.99
50	25	362.45	1.87E+05	87.96	305.61	102.91	109.08
50	50	362.16	3.52E+04	95.06	314.19	104.26	99.17
50	100	365.89	4.00E+03	102.88	366.87	124.78	96.77
100	10	365.10	1.91E+06	92.85	336.96	102.23	120.48
100	25	368.32	2.80E+05	87.23	331.12	101.92	98.41
100	50	371.46	4.47E+04	94.38	343.33	105.81	93.50
100	100	374.75	5.24E+03	103.20	377.77	124.91	93.06
200	10	369.47	2.26E+06	93.76	354.78	102.75	102.55
200	25	374.41	3.45E+05	88.92	356.57	104.01	90.46
200	50	373.54	3.75E+04	95.43	360.88	106.00	90.00
200	100	372.23	5.35E+03	102.05	375.20	123.28	92.45
400	10	373.97	3.00E+06	93.64	369.02	103.44	95.77
400	25	375.56	3.68E+05	89.67	371.06	104.53	88.90
400	50	378.20	4.62E+04	93.83	375.67	105.39	89.63
400	100	378.71	5.75E+03	102.85	379.46	124.91	91.43
1000	10	376.12	3.65E+06	95.19	376.18	103.64	94.00
1000	25	378.38	3.94E+05	88.95	378.59	103.74	88.07
1000	50	377.30	6.68E+04	95.09	377.58	105.84	88.75
1000	100	380.23	5.93E+03	103.56	380.50	125.77	90.33
2000	10	375.58	3.45E+06	94.56	375.85	103.17	94.06
2000	25	373.89	4.06E+05	88.87	374.33	103.81	88.27
2000	50	380.91	5.54E+04	94.91	381.29	105.85	88.48

Table 125: Out-of-sample root mean squared prediction errors for future observations when the true data generating process is a linear model with random effects with a balanced panel.

model (again, the more complicated model results are similar to the those of the linear model). The difference between the root mean squared errors for the linear model without random effects and the RE-EM tree are not statistically significant for some cases of larger I and $T = 10, 25$.

Finally, we examine the predictions for individuals who were not in the original sample, using some of their observations to estimate random effects. As before, the RE-EM tree performs best when it is the true data generating process, as shown in Table 128. The linear model performs best for small samples and the RE-EM tree performs best for larger samples in the other two cases, as shown in Table 129 for the linear model.

In all of the different types of prediction, the RE-EM tree estimation has the best predictive performance when it is the true model and good performance otherwise, especially for larger sample sizes. The success of the RE-EM tree when it is not the correct model allows us to apply it to situations when the model is unknown and is likely to be complicated, such as the transactions data. This agrees with the leave-one-out cross-validation transactions study that found that the RE-EM tree performed best for that dataset.

4.6.7 Summary of Monte Carlo Results

These simulations have found that including random effects dramatically improves the accuracy of in-sample fits and forecasts of future observations for individuals in the sample. The linear model with random effects performs best in certain tests; it has the lowest root mean squared error in-sample and in predictions when a random effect can be estimated, except when the true data generating process is a RE-EM tree. However, the linear model with random effects performs badly when predicting the target variable for new individuals when a random effect cannot be

I	T	LM	LME	RPART	REEM	MVPART
50	10	110.14	109.79	96.69	95.61	113.29
50	25	101.18	103.88	85.11	83.34	104.25
50	50	107.87	114.07	89.35	88.05	114.93
50	100	114.29	115.80	87.69	85.81	120.99
100	10	108.92	108.54	85.59	84.97	110.85
100	25	101.54	104.61	79.74	78.78	106.63
100	50	104.14	108.34	77.35	77.13	106.54
100	100	103.82	105.60	73.02	72.37	107.55
200	10	100.83	101.87	74.19	74.23	102.19
200	25	99.13	101.23	70.99	70.94	96.75
200	50	103.05	108.06	71.39	71.32	101.34
200	100	100.54	101.85	68.95	68.80	104.10
400	10	101.02	101.63	71.71	71.89	97.44
400	25	96.90	99.45	70.63	70.46	92.53
400	50	96.41	98.58	71.02	70.67	94.30
400	100	106.57	108.75	68.97	68.93	105.89
1000	10	104.79	104.97	71.44	71.50	88.44
1000	25	97.30	99.23	69.55	69.48	84.76
1000	50	101.49	105.29	68.71	68.69	91.28
1000	100	102.43	104.24	68.28	68.26	93.97
2000	10	111.35	112.23	68.21	68.28	81.62
2000	25	96.60	99.44	68.87	68.89	81.59
2000	50	96.99	100.99	67.64	67.64	86.60

Table 126: Out-of-sample root mean squared prediction errors for new observations when the true data generating process is a RE-EM tree.

I	T	LM	LME	RPART	REEM	MVPART
50	10	391.35	385.68	482.42	404.66	433.33
50	25	387.76	377.63	475.71	391.28	432.39
50	50	379.24	371.52	457.46	373.94	427.74
50	100	384.02	377.45	443.65	376.77	435.58
100	10	378.55	375.65	460.33	389.34	418.70
100	25	374.05	370.12	448.79	375.75	417.39
100	50	365.32	362.58	418.20	364.36	406.13
100	100	368.65	365.44	405.17	362.15	415.64
200	10	373.57	372.40	423.93	376.45	401.48
200	25	374.82	373.45	410.65	372.87	403.78
200	50	377.37	376.78	405.64	374.83	405.67
200	100	366.91	365.76	382.46	362.31	399.06
400	10	373.15	373.48	387.69	373.00	390.53
400	25	369.18	369.80	379.80	368.62	388.74
400	50	367.50	368.88	372.25	365.95	382.94
400	100	373.38	373.16	375.24	369.25	391.25
1000	10	370.60	371.22	370.36	369.68	373.67
1000	25	368.02	368.60	367.77	367.27	373.56
1000	50	385.80	386.49	385.38	383.59	389.33
1000	100	383.33	384.12	383.54	380.47	385.82
2000	10	373.75	375.62	373.70	373.15	373.77
2000	25	353.82	355.53	353.71	353.96	353.72
2000	50	359.68	361.19	359.73	357.94	359.97

Table 127: Out-of-sample root mean squared prediction errors for new observations when the true data generating process is a linear model with random effects.

I	T	LM	LME	RPART	REEM
50	10	94.18	88.34	88.66	80.94
50	25	89.82	83.75	82.48	73.82
50	50	106.98	100.09	88.47	79.04
50	100	117.40	115.03	85.15	77.55
100	10	91.97	89.70	77.82	72.20
100	25	96.48	92.78	78.42	71.23
100	50	107.04	101.22	76.25	68.82
100	100	103.71	102.34	71.71	63.91
200	10	91.90	82.73	68.38	63.19
200	25	95.00	91.64	73.80	67.52
200	50	107.45	100.98	77.22	69.36
200	100	107.05	103.03	69.31	61.65
400	10	88.68	82.01	68.72	63.92
400	25	93.09	88.84	72.06	65.24
400	50	104.77	100.05	71.03	62.98
400	100	106.47	104.11	68.53	60.61
1000	10	88.54	85.11	70.26	64.56
1000	25	92.95	89.87	71.80	65.56
1000	50	103.70	97.98	68.38	60.70
1000	100	108.97	106.06	68.58	61.35
2000	10	89.78	89.88	68.77	62.71
2000	25	98.87	94.96	71.53	65.28
2000	50	96.47	92.33	66.96	59.82

Table 128: Out-of-sample root mean squared prediction errors for future observations for new individuals when the true data generating process is a RE-EM tree.

I	T	LM	LME	RPART	REEM
50	10	385.56	102.76	478.79	136.73
50	25	385.95	93.56	473.63	113.03
50	50	378.89	92.97	458.34	100.20
50	100	381.64	103.72	447.67	102.54
100	10	374.06	93.70	458.43	118.81
100	25	374.77	89.51	448.43	100.28
100	50	365.97	96.12	418.71	99.30
100	100	367.07	103.93	403.63	93.47
200	10	369.09	97.61	418.07	106.74
200	25	376.02	92.35	411.57	93.69
200	50	377.51	94.56	404.80	91.82
200	100	365.20	102.95	383.91	92.74
400	10	369.07	92.66	382.36	93.35
400	25	368.02	87.63	379.84	86.96
400	50	369.68	97.78	374.99	92.76
400	100	372.99	103.77	375.44	89.93
1000	10	367.10	95.01	367.46	94.74
1000	25	367.62	87.00	367.40	86.73
1000	50	387.21	97.55	386.94	91.12
1000	100	382.21	101.52	382.27	90.52
2000	10	369.90	94.28	369.73	93.14
2000	25	355.38	89.76	355.41	89.64
2000	50	361.88	95.49	362.18	89.43

Table 129: Out-of-sample root mean squared prediction errors for future observations for new individuals when the true data generating process is a linear model with random effects.

estimated. In Section 4.6.3, we found that the problem occurs because the linear model with random effects estimates the random effects and fixed effects badly, but errs in a way that keeps the total more accurate. The RE-EM tree does not have this problem and therefore performs well in all of the different forecasting tasks. The problems with the linear model are exacerbated when ρ moves toward 1, but reduced when the panel is balanced. Furthermore, we find that the difference between the RE-EM tree and the linear model with random effects decreases as the sample size grows for those tasks in which the linear random effects model is best. Overall, the RE-EM tree is a successful estimation method in a variety of situations and is the clear method of choice when the true relationship in the population takes the form of a tree.

4.7 Conclusion and Future Work

In this paper, we have presented a new tool for data mining with longitudinal data. The RE-EM tree preserves the structure of longitudinal data while providing the flexibility to use time-varying covariates. Using datasets on traffic fatalities and on web transactions, we have shown that RE-EM trees can improve predictive performance and allow us to model our target variables without assuming that linear models hold. In our Monte Carlo experiments, we have found that RE-EM trees outperform trees that do not allow for random effects, are more effective than other methods when the true relationship takes the form of a tree, and are comparable to or better than linear models that include random effects, even when a tree model is not true.

This paper has explored the basics of the RE-EM tree method. First, we have used the default parameters for `rpart` throughout our estimation and simulation; alternative values may be preferable for RE-EM trees. Second, methods such as

bagging and boosting build on a tree structure as a way to improve predictive performance [see for example, Hastie et al., 2001, Section 8.7 and Chapter 10]. We expect that the improvements from these methods would carry over when they are applied to RE-EM trees as well. Further, these methods might generalize to classification trees, which would extend their use to another class of models. Finally, one could extend the existing consistency results for regression trees to RE-EM trees, checking whether f or the random effects are estimated consistently.

Since these trees extend the use of data mining methods into the area of longitudinal data, a wide variety of potential applications exist. This tool will allow researchers to move beyond fitting linear models to panel data and can uncover interactions and non-linear relationships that could not be found before.

References

M. Abdolell, M. LeBlanc, D. Stephens, and R. V. Harrison. Binary partitioning for continuous longitudinal data: categorizing a prognostic variable. *Statistics in Medicine*, 21:3395–3409, 2002.

D. Afshartous and Jan de Leeuw. Prediction in multilevel models. *Journal of Educational and Behavioral Statistics*, 30:109–139, 2005.

Sassan Alizadeh, Michael W. Brandt, and Francis X. Diebold. Range-based estimation of stochastic volatility models. *Journal of Finance*, 57:1047–1091, 2002.

Craig F. Ansley and Robert Kohn. A note on reparameterizing a vector autoregressive moving average model to enforce stationarity. *Journal of Statistical Computation and Simulation*, 24:99–106, 1986.

Richard T. Baillie. Long memory processes and fractional integration in econometrics. *Journal of Econometrics*, 73:5–59, 1996.

Ben S. Bernanke and James L. Powell. The cyclical behavior of industral labor markets: a comparison of the prewar and postwar eras. NBER Working Paper, Number 1376, 1984.

Stefano Bertelli and Massimiliano Caporin. A note on calculating autocovariances of long-memory processes. *Journal of Time Series Analysis*, 23, 2002.

N. H. Bingham, C. M. Goldie, and J. L. Teugels. *Regular Variation*. Cambridge University Press, 1989.

Peter Bloomfield. *Fourier Analysis of Time Series: an Introduction*. John Wiley & Sons, 1976.

Albrecht Bottcher and Bernd Silbermann. *Introduction to large truncated Toeplitz matrices*. Springer, 1999.

F. J. Breidt, N. Crato, and P. de Lima. The detection and estimation of long memory in stochastic volatility. *Journal of Econometrics*, 83:325–348, 1998.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

David R. Brillinger. *Time Series: Data Analysis and Theory*. Holden-Day, Inc., 1981.

Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods*. Springer-Verlag, 1993.

Raymond H. Chan and Michael K. Ng. Conjugate gradient methods for Toeplitz systems. *SIAM Review*, 38(3):427–482, 1996.

Tony F. Chan. An optimal circulant preconditioner for Toeplitz systems. *SIAM Journal of Statistical Computation*, 9:766–771, 1988.

Tony F. Chan and Julia A. Olkin. Circulant preconditioners for Toeplitz-block matrices. *Numerical Algorithms*, 6:89–101, 1994.

Willa Chen, Clifford M. Hurvich, and Yi Lu. On the correlation matrix of the discrete fourier transform and the fast solution of large Toeplitz systems for long-memory time series. *Journal of the American Statistical Association*, 101, 2006.

Willa W. Chen and Clifford M. Hurvich. Estimating fractional cointegration in the presence of polynomial trends. *Journal of Econometrics*, 117:95–121, 2003.

Willa W. Chen and Clifford M. Hurvich. Semiparametric estimation of fractional cointegrating subspaces. *Annals of Statistics*, 34, 2006.

Yin-Wong Cheung. An empirical model of daily highs and lows. *International Journal of Finance and Economics*, 12:1–20, 2007.

E. M. Chi and G. C. Reinsel. Models for longitudinal data with random effects and AR(1) errors. *Journal of the American Statistical Association*, 84:452–459, 1989.

Bent Jesper Christensen and Morten Ørregaard Nielsen. Asymptotic normality of narrow-band least squares in the stationary fractional cointegration model and volatility forecasting. *Journal of Econometrics*, 133:343–371, 2006.

Ching-Fan Chung. Calculating and analyzing impulse responses for the vector arfima model. *Economics Letters*, 71:17–25, 2001.

R.B. Davies and D.S. Harte. Tests for the hurst effect. *Biometrika*, 74:95–101, 1987.

G. De'Ath. Multivariate regression trees: a new technique for modeling species-environment relationships. *Ecology*, 83:1105–1117, 2002.

G. De'ath. *mvpart: Multivariate partitioning*, 2006. R package version 1.2-4.

T. S. Dee and R. J. Sela. The fatality effects of highway speed limits by gender and age. *Economics Letters*, 79:401–408, 2003.

A. P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977.

Eugene D. Denman and Alex N. Beavers. The matrix sign function and computations in systems. *Applied Mathematics and Computation*, 2:63–94, 1976.

Rohit Deo, Clifford Hurvich, and Yi Lu. Forecasting realized volatility using a long-memory stochastic volatility model: estimation, prediction and seasonal adjustment. *Journal of Econometrics*, 131, 2006.

Rohit S. Deo and Clifford M. Hurvich. On the log periodogram regression estimator of the memory parameter in long memory stochastic volatility models. *Econometric Theory*, 17:686–710, 2001.

Y. Ding and J S. Simonoff. An investigation of missing data methods for classification trees applied to binary response data. *Journal of Machine Learning Research*, 11:131–170, 2010.

Jurgen A. Doornik and Marius Ooms. Computational aspects of maximum likelihood estimation of autoregressive fractionally integrated moving average models. *Computational Statistics and Data Analysis*, 42, 2003.

W. Dunsmuir and E.J. Hannan. Vector linear time series models. *Advances in Applied Probability*, 8, 1976.

Robert F. Engle and C. W. J. Granger. Co-integration and error correction: representation, estimation, and testing. *Econometrica*, 55, 1987.

T. Evgeniou, M. Pontil, and O. Toubia. A convex optimization approach to modeling consumer heterogeneity in conjoint estimation. 2006. URL `http://www.cs.ucl.ac.uk/staff/M.Pontil/reading/rrhet.pdf`.

G. Galimberti and A. Montanari. Regression trees for longitudinal data with time-

dependent covariates. In K. Jajuga, A. Sokolowski, and H.-H. Bock, editors, *Classification, Clustering and Data Analysis*, pages 391–398. Springer, 2002.

J. Geweke and Susan Porter-Hudak. The estimation and application of long memory time series models. *Journal of Time Series Analysis*, 4:221–238, 1983.

A. Ghose, P. Ipeirotis, and A. Sundararajan. The dimensions of reputation in electronic markets. Technical Report 06-02, NYU CeDER Working Paper, 2005.

C. W. J. Granger and M. Hatanaka. *Spectral analysis of economic time series*. Princeton University Press, 1964.

C. W. J. Granger and R. Joyeux. An introduction to long memory time series models and fractional differencing. *Journal of Time Series Analysis*, 1:15–39, 1980.

Ulf Grenander and Gabor Szego. *Toeplitz Forms and Their Applications*. University of California Press, 1958.

Robert E. Hall and John B. Taylor. *Macroeconomics*. W. W. Norton and Company, 1997.

James D. Hamilton. *Time Series Analysis*. Princeton University Press, 1994.

E. J. Hannan and P. J. Thomson. Estimating group delay. *Biometrika*, 60, 1973.

E.J. Hannan. *Multiple Time Series*. John Wiley and Sons, Inc., 1970.

D. A. Harville. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, 72: 320–340, 1977a.

David A. Harville. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, 72:320–338, 1977b.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2001.

Michael T. Heath. *Scientific Computing: an Introductory Survey*. McGraw-Hill, 2002.

J. Hidalgo. Spectral analysis for bivariate time series with long memory. *Econometric Theory*, 12:773–792, 1996.

J. R. M. Hosking. Fractional differencing. *Biometrika*, 68:165–176, 1981.

Yuzo Hosoya. The quasi-likelihood approach to statistical inference on multiple time-series with long-range dependence. *Journal of Econometrics*, 73:217–236, 1996.

Yuzo Hosoya. A limit theory for long-range dependence and staistical inference on related models. *Annals of Statistics*, 25:105–137, 1997.

W.-C. Hsiao and Y.-S. Shih. Splitting variable selection for multivariate regression trees. *Statistics and Probability Letters*, 77:265–271, 2007.

J. Hualde and P.M. Robinson. Root-n-consistent estimation of weak fractional cointegrations. *Journal of Econometrics*, 140:450–484, 2007.

Clifford M. Hurvich and Willa W. Chen. An efficient taper for potentially overdifferenced long-memory time series. *Journal of Time Series Analysis*, 21.

Clifford M. Hurvich and Philippe Soulier. Stochastic volatility models with long memory. In *Handbook of Financial Time Series*, pages 345–354. Springer, 2009.

Clifford M. Hurvich, Eric Moulines, and Philippe Soulier. The FEXP estimator for potentially non-stationary linear time series. *Stochastic Processes and their Applications*, 97:307–340, 2002.

Clifford M. Hurvich, Eric Moulines, and Philippe Soulier. Estimating long memory in volatility. *Econometrica*, 73:1283–1328, 2005.

L. H. Koopmans. *The Spectral Analysis of Time Series*. 1974.

H. R. Kunsch. Statistical aspects of self-similar processes. In Yu Prohorov and V. V. Sazanov, editors, *Proceedings of the First World Congress of the Bernoulli Society 1*, pages 67–74, Utrecht, 1987. VNU Science Press.

N. M. Laird and J. H. Ware. Random-effects models for longitudinal data. *Biometrics*, 38:963–974, 1982.

D. R. Larsen and P. L. Speckman. Multivariate regression trees for analysis of abundance data. *Biometrics*, 60:543–549, 2004.

S. K. Lee. On generalized multivariate decision tree by using gee. *Computational Statistics and Data Analysis*, 49:1105–1119, 2005.

S. K. Lee. On classification and regression trees for multiple responses and its application. *Journal of Classification*, 23:123–141, 2006.

S. K. Lee, H.-C. Kang, S.-T. Han, and K.-H. Kim. Using generalized estimating equations to learn decision trees with multivariate responses. *Data Mining and Knowledge Discovery*, 11:273–293, 2005.

J. Lin and B. Wei. Testing for heteroscedasticity and/or autocorrelation in longitudinal mixed effect nonlinear models with AR(1) errors. *Communications in Statistics - Theory and Methods*, 36.

I. N. Lobato. A semiparametric two-step estimator in a multivariate long memory model. *Journal of Econometrics*, 90:129–153, 1999.

I. N. Lobato and P.M. Robinson. Averaged periodogram estimation of long memory. *Journal of Econometrics*, 73:303–324, 1996.

Ignacio N. Lobato. Consistency of the averaged cross-periodogram in long memory time series. *Journal of Time Series Analysis*, 18, 1997.

W.-Y. Loh. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386, 2002.

Albert Luceno. A fast likelihood approximation for vector general linear processes with long series: Application to fractional differencing. *Biometrika*, 83, 1996.

Vance L. Martin and Nigel P. Wilkins. Indirect estimation of arfima and varfima models. *Journal of Econometrics*, 93:149–175, 1999.

J. N. Morgan and J. A. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58:415–434, 1963.

Morten Orregaard Nielsen. Spectral analysis of fractionally cointegrated systems. *Economics Letters*, 83:225–231, 2004.

H. D. Patterson and R. Thompson. Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58:545–554, 1971.

Richard W. Peach, Robert W. Rich, and Alexis Anoniades. The historical and recent behavior of goods and services inflation. *Economic Policy Review*, 10: 19–31, 2004.

Maurice B. Priestley. *Spectral analysis and time series*. Academic Press, 1981.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL `http://www.R-project.org`.

Nalini Ravishanker and Bonnie K. Ray. Bayesian analysis of vector ARFIMA processes. *Australian Journal of Statistics*, 39:295–312, 1997.

Nalini Ravishanker and Bonnie K. Ray. Bayesian prediction for vector ARFIMA processes. *International Journal of Forecasting*, 18, 2002.

Valderio A. Reisen. Estimation of the fractional difference parameter in the ARIMA(p, d, q) model using the smoothed periodogram. *Journal of Time Series Analysis*, 15, 1994.

P. M. Robinson and D. Marinucci. Narrow-band analysis of nonstationary processes. *Annals of Statistics*, 29.

P. M. Robinson and D. Marinucci. *Semiparametric frequency domain analysis of fractional cointegration*, chapter 14, pages 334–374. Oxford University Press, 2003.

Peter M. Robinson. Semiparametric analysis of long-memory time series. *Annals of Statistics*, 22:515–539, 1994.

Peter M. Robinson. Log-periodogram regression of time series with long range dependence. *Annals of Statistics*, 23, 1995a.

Peter M. Robinson. Gaussian semiparametric estimation of long range dependence. *Annals of Statistics*, 23:1630–1661, 1995b.

Peter M. Robinson. Multiple local Whittle estimation in stationary systems. *Annals of Statistics*, 36, 2008.

P.M. Robinson and J. Hualde. Cointegration in fractional systems with unknown integration orders. *Econometrica*, 71, 2003.

M. R. Segal. Tree-structured models for longitudinal data. *Journal of the American Statistical Association*, 87:407–418, 1992.

Rebecca J. Sela and Clifford M. Hurvich. Computationally efficient gaussian maximum likelihood methods for vector ARFIMA models. *Journal of Time Series Analysis*, 30, 2009.

Eugene Seneta. *Regularly Varying Functions*. Springer-Verlag Lecture Notes in Mathematics, 1970.

Jonathan R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Unpublished manuscript, 1994.

Katsumi Shimotsu. Gaussian semiparametric estimation of multivariate fractionally integrated processes. *Journal of Econometrics*, 137:277–310, 2007.

Fallaw Sowell. Maximum likelihood estimation of fractionally integrated time series models. Unpublished manuscript, 1989a.

Fallaw Sowell. A decomposition of block Toeplitz matrices with applications to vector time series. Unpublished manuscript, 1989b.

T. M Therneau and B. Atkinson. *rpart: Recursive Partitioning*, 2006. R port by Brian Ripley.

Wen-Jen Tsay. Maximum likelihood estimation of stationary multivariate ARFIMA processes. Unpublished manuscript, 2007.

Carlos Velasco. Non-stationary log-periodogram regression. *Journal of Econometrics*, 91:325–371, 1999.

G. Verbeke and G. Molenberghs. *Linear mixed models for longitudinal data*. Springer, 2000.

G. Verbeke and G. Molenberghs. The use of score tests for inference on variance components. *Biometrics*, 59:254–262, 2003.

Monique Vuilleumier. Slowly varying functions in the complex plane. *Transactions of the American Mathematical Society*, 218:343–348, 1976.

Eric Weisstein. Hypergeometric function, 2008. `http://mathworld.wolfram.com/HypergeometricFunction.html`.

Peter Whittle. On the fitting of multivariate autoregressions, and the approximate canonical factorization of a spectral density matrix. *Biometrika*, 50, 1963.

Andrew T.A. Wood and Grace Chan. Simulation of stationary gaussian processes in $[0, 1]^d$. *Journal of Computational and Graphical Statistics*, 3, 1994.

J. M. Wooldridge. *Econometric Analysis of Cross Section and Panel Data*. MIT Press, 2002.

H. Zhang. Multivariate adaptive splines for analysis of longitudinal data. *Journal of Computational and Graphical Statistics*, 6:74–91, 1997.

H. Zhang. Classification trees for multiple binary responses. *Journal of the American Statistical Association*, 93:180–193, 1998.

Antoni Zygmund and Robert Fefferman. *Trigonometric Series*. Cambridge University Press, 2002.