Generating Longitudinally Correlated Binary Data

by

Katrina Rogers-Stewart, B. Math (Carleton)

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfilment of
the requirements for the degree of
Master of Science

School of Mathematics and Statistics
Ottawa-Carleton Institute for Mathematics and Statistics
Carleton University
Ottawa, Ontario
April, 2006

© Copyright
2006, Katrina Rogers-Stewart

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-16502-7

Our file Notre référence

ISBN: 978-0-494-16502-7

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

In this thesis we investigate methods of generating longitudinally correlated binary data. Several approaches to generate such data have been proposed in the literature. We review a selection of these and consider those suggested by Qaqish (2003) [15], Bahadur (1961) [1] and Kanter (1975) [10] further.

To evaluate these methods we consider three correlation structures; autoregressive of order 1, moving average of order 1, and exchangeable. The approach suggested by Kanter (1975) [10] only allows stationary autoregressive processes to be generated. However, McKenzie (1981) [12] explains how to expand Kanter (1975) [10] to incorporate stationary autoregressive moving average correlations structures. Here we propose an extension to Kanter (1975) [10] which allows a stationary exchangeable correlation structure to be generated. We also suggest a model, based upon Kanter (1975) [10], which allows for nonstationary autoregressive processes.

For the three generating techniques we determine, from a theoretically perspective, their capability to generate according to a specific correlation structure and parameter. A direct consequence of generating binary data is that, for a specific correlation structure, all these methods impose range restrictions for the unconditional mean and correlation parameter. We establish these range restrictions for the three correlation structures we consider. Additionally, we propose an extension to widen the allowable ranges for the Kanter (1975) [10] model and related extensions.

Subsequently, we perform a simulation study to analyze the properties of the

generated data. In particular, we wish to examine the ability to adequately estimate correlation parameters, along with parameters associated with covariates in a model. In this respect it was found that all the techniques had very comparable results, with the exception of the extension to Kanter (1975) [10] for nonstationary autoregressive sequences. This extension resulted in unsatisfactory estimation when compared to the other methods. This simulation study implies that the usefulness of the generation methods is determined by the range restrictions, and the correlation structures and parameters that they allow.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Patrick Farrell, for his continued support. He has introduced and further exposed me to research and has guided me through my academic path. I am indebted to him for the wealth of knowledge and experiences he has relayed to me. I have learnt an immense amount, on an assortment of topics, through our many conversations.

To Mathieu Lemire, thank you for accompanying me on this journey. It has been wonderful having you by my side. A special thank you to my parents, Faith and Jim, and sister Lydia for their support and encouragement throughout my university education.

Table of Contents

Abstract	ii
Acknowledgments	iv
Table of Contents	v
List of Tables	viii
List of Figures	x
1 Introduction	1
2 Literature Review	4
3 Description of Generating Methods and Related Extensions	10
3.1 QAQ	10
3.1.1 AR(2) with QAQ	11
3.2 BAH	12
3.3 KAN	13
3.3.1 Extension to Nonstationary	15
4 Range Restrictions	16
4.1 QAQ	17
4.2 BAH	17

4.3	KAN	18
4.3.1	Extension to KAN model	19
4.4	Range Restrictions for Correlation Parameters	19
4.4.1	Stationary Processes	20
4.4.2	Nonstationary Range Restrictions	21
5	Model and Simulation Study	31
5.1	The Model	32
5.2	Designs	32
5.3	Estimation	33
5.4	Generation	35
5.5	Technical Circumstances To Consider	35
6	Simulation Results	39
6.1	Stationary Designs	39
6.2	Nonstationary Designs	43
7	Conclusion and Discussion	46
	List of References	49
	Appendix A Derivation of Qaqish (2003) Results	51
	Appendix B Derivation of Kanter (1975) Results and Extensions	57
B.1	Derivation of Kanter (1975) Results for a stationary AR(1) process .	57
B.2	MA(1)	60
B.3	EX	64
B.4	Derivation of KAN* for AR(1), MA(1) and Exchangeable structures .	66
B.5	AR(2)	67

Appendix C Extension of Kanter (1975) To Nonstationary AR(p)	71
C.1 Details	72
Appendix D Programming Code	77
D.1 Maple code	77
D.2 FORTRAN code	79

List of Tables

3.1	Formulae for $b_{tt'}$	11
3.2	Models for generating KAN.	13
3.3	Formulae for ρ , u and w in the KAN model.	15
4.1	Models for generating KAN*.	19
4.2	AR(1) stationary range restrictions.	22
4.3	MA(1) stationary range restrictions.	23
4.4	EX stationary range restrictions.	24
4.5	AR(1) Transformation 1 nonstationary range restrictions.	25
4.6	AR(1) Transformation 2 nonstationary range restrictions.	26
4.7	MA(1) Transformation 1 nonstationary range restrictions.	27
4.8	MA(1) Transformation 2 nonstationary range restrictions.	28
4.9	EX Transformation 1 nonstationary range restrictions.	29
4.10	EX Transformation 2 nonstationary range restrictions.	30
5.1	Results for designs demonstrating the effect of ill-conditioned X_g matrices.	37
5.2	Results for designs demonstrating the effect of cluster size.	38
6.1	Results for stationary AR(1) designs.	40
6.2	Results for stationary MA(1) designs.	42
6.3	Results for stationary EX designs.	42
6.4	Results for stationary AR(1) designs with constant ρ_g across groups. .	44

6.5	Results for stationary MA(1) designs with constant ρ_g across groups.	44
6.6	Results for stationary EX designs with constant ρ_g across groups. . .	44
6.7	Results for nonstationary AR(1) designs.	45
6.8	Results for nonstationary MA(1) designs.	45
6.9	Results for nonstationary EX designs.	45

List of Figures

4.1	AR(1) stationary range restrictions.	22
4.2	MA(1) stationary range restrictions.	23
4.3	EX stationary range restrictions.	24
4.4	AR(1) Transformation 1 nonstationary range restrictions.	25
4.5	AR(1) Transformation 2 nonstationary range restrictions.	26
4.6	MA(1) Transformation 1 nonstationary range restrictions.	27
4.7	MA(1) Transformation 2 nonstationary range restrictions.	28
4.8	EX Transformation 1 nonstationary range restrictions.	29
4.9	EX Transformation 2 nonstationary range restrictions.	30

Chapter 1

Introduction

Longitudinally correlated binary data commonly arises in statistical and medical studies. It can occur in clinical trials where independent patients are treated over time to assess the effectiveness of a certain treatment for a particular disease. A portion of the study conducted by Ware et al. (1984) [20] is analyzed in Fitzmaurice and Laird (1993) [7], with wheezing status in children investigated. Specifically Fitzmaurice and Laird (1993) [7] examine the relationship between the covariate of a mother's smoking status and her child's wheezing status. Data on children in six sites, as to whether a child wheezed and if the child's mother was a smoker, was collected over a period of four years at one year intervals. All the children are of different mothers, accordingly each child is assumed independent of one another. Longitudinally correlated binary data also frequently arises in ophthalmologic and dental applications. In these situations the correlation appears from taking multiple measurements on the same individual.

In the analysis of such data we typically wish to estimate some parameter or parameters in a model that describes the binary responses as a function of covariates. A variety of models and estimation methods have been proposed in the literature. A likelihood-based method is introduced by Fitzmaurice and Laird (1993) [7], a quasi-least squares method by Shults and Chaganty (1998) [16], generalized quasi-likelihood

(GQL) approaches are discussed in Sutradhar and Das (1999) [18] and Sutradhar (2003) [17], while models using generalized estimating equations (GEE) can be found in Chaganty and Joe (2004) [3] and Sutradhar (2003) [17].

In any model the properties of the estimators of model parameters are of utmost importance. When developing a model one is obliged to assess these properties, which often requires a simulation study where data must be generated. To generate longitudinally correlated binary data, the presence of the longitudinal correlations needs to be acknowledged. In any generating method the correlation structure must be incorporated into the generation mechanism, which typically imposes range restrictions on the correlation parameters. The foundation for these range restrictions is the Bernoulli parameter associated with each response being confined to the range $(0, 1)$. The restrictions realized for each method depend highly on the approach used to generate the data, and their extent varies greatly between methods.

A number of articles have suggested methods to generate longitudinally correlated binary data. In this study we shall investigate three fundamentally different approaches to generating this type of data. We investigate the techniques proposed by Qaqish (2003) [15], Bahadur (1961) [1] and Kanter (1975) [10]. In particular we investigate for each technique the effects on model parameter estimators and the limitations imposed by the range restrictions.

The model proposed in Kanter (1975) [10] is designed for stationary autoregressive processes. McKenzie (1981) [12] expands this concept to incorporate autoregressive - moving average models. We introduce here three separate extensions to Kanter (1975) [10]; the inclusion of exchangeable correlations, an alternative model to enlarge the state space established by the range restrictions, and the generation of nonstationary data.

In Chapter 2 we present a literature review, with the focus on the three generating methods to be investigated here. Extensions to the generating methods are proposed

and developed in Chapter 3, while Chapter 4 discusses the imposed range restrictions associated with each generation method. In order to assess the impact of these range restrictions, we carried out a simulation study where the correlated binary response data are modeled as a function of covariates according to a logit link. Properties of estimators of regression and correlation parameters are of interest in the study. Chapter 5 provides the details of the simulation study, including the model, estimation procedures, and technical considerations, with the results provided in Chapter 6. A conclusion and discussion is given in Chapter 7.

Chapter 2

Literature Review

Suppose that the binary responses of a cluster (individual) are correlated over T outcomes. If we additionally suppose a number of clusters have identical means and correlation structures, then we can refer to these similarly specified clusters as a group. We let $\mathbf{Y}_{gi} = (Y_{gi1}, \dots, Y_{giT})^T$ represent a random vector corresponding to the i^{th} cluster in the g^{th} group, and $\mathbf{y}_{gi} = (y_{gi1}, \dots, y_{giT})^T$ the analogous vector of binary observations. In Chapters 5 and 6 we consider a simulation study of such a situation. For simplicity we will suppress the subscripts indicating the group and cluster in what follows. Hence, for any cluster in general, we will use the notation $\mathbf{Y} = (Y_1, \dots, Y_T)^T$ and $\mathbf{y} = (y_1, \dots, y_T)^T$.

We let $\boldsymbol{\mu} = E[\mathbf{Y}]$ be the unconditional expectation of $\mathbf{Y}$ and $\rho_{tt'} = Corr(Y_t, Y_{t'})$ be the correlation between Y_t and $Y_{t'}$, $1 \leq t, t' \leq T$. We let ρ denote the single correlation parameter associated with the following correlation structures; autoregressive of order 1 (AR(1)), moving average of order 1 (MA(1)) and exchangeable (EX).

Various generation methods for producing correlated binary data have been introduced by Gange (1995) [8], Park, Park and Shin (1996) [14], Emrich and Piedmonte (1991) [5], Lunn and Davies (1998) [11], Oman and Zucker (2001) [13], Qaqish (2003) [15], Bahadur (1961) [1] and Kanter (1975) [10], are discussed in this chapter. The Oman and Zucker (2001) [13] model utilizes a continuous distribution. The

Qaqish (2003) [15] and Bahadur (1961) [1] techniques employ differing representations of a multivariate binary distribution. Kanter (1975) [10] models an autoregressive process using sequences of Bernoulli random variables and the addition modulo 2 operation. All of these methods will involve constraints resulting from range restrictions, and will result in various theoretical and practical difficulties for differing correlations.

Discussions on range restrictions and correlation problems are provided in Bondesson (2003) [2], Chaganty and Joe (2004) [3] and Joe (1997) [9].

Gange (1995) [8] suggests using an iterative proportional fitting algorithm to generate categorical data. In the context of the binary problem discussed here, one would specify the unconditional means and the correlations, hence defining the marginal and pairwise probabilities. By fitting a log-linear model of a 2^T contingency table with first and second order interaction effects and using the iterative proportional fitting algorithm, the joint distribution of $\mathbf{Y}$ can be completely determined. To generate the data we select an element with T components proportional to the joint probability. The number of probabilities which need to be calculated increases quickly with the size of the sequence. Hence this algorithm is most appropriate for relatively small processes.

Park, Park and Shin (1996) [14] consider Poisson random variables $V_t, 1 \leq t \leq T$, defined as linear combination of τ independent Poisson random variables. The binary response is defined as $Y_t = I\{V_t = 0\}$, where $I\{\cdot\}$ is the indicator function. The value of τ and the means of the corresponding Poisson variables are determined by the means and correlation of the binary sequence to be generated. The partial sums of the independent Poisson variables are capable of generating a wide variety of correlation structures, including the three discussed here. However, the use of Poisson variables permits only non-negative correlations to be generated with model.

Emrich and Piedmonte (1991) [5] propose a method which first generates a multivariate normal random vector and then performs a transformation to obtain a binary sequence. Let $\Phi[x_1, x_2, \rho]$ represent the cumulative distribution function for a standard bivariate normal random variable with correlation coefficient ρ . First, the equations

$$\Phi[z(\mu_t), z(\mu_{t'}), \rho_{tt'}] = \delta_{tt'}[\mu_t(1 - \mu_t)\mu_{t'}(1 - \mu_{t'})]^{1/2} + \mu_t\mu_{t'},$$

where $z(\mu)$ denotes the μ th quantile of the standard normal distribution, need to be solved using numerical integration for $\rho_{tt'}$ for all combinations of $1 \leq t \leq T - 1$ and $2 \leq t' \leq T$. We must then ensure the matrix $[\Sigma]_{tt'} = \rho_{tt'}$ is positive-definite. If it is not, we need to provide a different correlation structure. Next generate a multivariate normal random vector, $Z = (Z_1, Z_2, \dots, Z_T)$, with mean $\mathbf{0}$ and correlation matrix Σ . Finally set $Y_t = I\{Z_t \leq z(\mu_t)\}$. This method has the flexibility to generate from many correlation structures, and in particular those with negative correlations.

Lunn and Davies (1998) [11] introduce a model for efficient generation of stationary EX and AR(1) correlations. However, the method can only generate data with positive correlations. Let Y_0 and B_t both be distributed Bernoulli with parameter μ and A_t Bernoulli with parameter $\sqrt{\rho}$. Then $Y_t = A_t Y_0 + (1 - A_t)B_t$ generates a stationary EX process. For AR(1) define A_t as having mean ρ and B_t with expected value μ . The model

$$Y_t = \begin{cases} B_1, & t = 1 \\ A_t Y_{t-1} + (1 - A_t)B_t, & t \geq 2 \end{cases}$$

produces a stationary AR(1) sequence.

Omar and Zucker (2001) [13] expand upon the results of Lunn and Davies (1998) [11]. In particular nonstationary AR(1), MA(1) and EX sequences can be generated.

The method entails choosing any continuous distribution and setting $\theta_t = F^{-1}(\mu_t)$, where F is the cumulative distribution function. Two sequences of independent random variables are then generated B_t by an F distribution and A_t by a Bernoulli distribution. A third variable, Z_t , is also calculated, its definition depending on A , B and the correlation structure. For example, the values of Z_t for AR(1), MA(1), and EX structures are as follows.

AR(1)	$Z_t = \begin{cases} B_1, & t = 1 \\ A_t Z_{t-1} + (1 - A_t) B_t, & t \geq 2 \end{cases}$
MA(1)	$Z_t = A_t B_{t-1} + (1 - A_t) B_t$
EX	$Z_t = A_t B_0 + (1 - A_t) B_t$

The Y_t are subsequently defined by comparing Z_t to θ , specifically $Y_t = I\{Z_t \leq \theta_t\}$.

Qaqish (2003) [15] gives a data generation method which explicitly employs the conditional distribution of Y_t given $(Y_1, \dots, Y_{t-1})^T$. The representation of the conditional distribution presented in Qaqish (2003) [15] incorporates an additive adjustment applied to the mean. The general form is reduced to the specific case,

$$\begin{aligned}
 \lambda_t &= P(Y_t = 1 | Y_1 = y_1, \dots, Y_{t-1} = y_{t-1}) \\
 &= E(Y_t | Y_1 = y_1, \dots, Y_{t-1} = y_{t-1}) \\
 &= \mu_t + \sum_{t'=1}^{t-1} b_{tt'}(y_{t'} - \mu_{t'}).
 \end{aligned} \tag{2.1}$$

The correlation structure is reflected through the $b_{tt'}$, and any structure is permissible. The $b_{tt'}$ for some common correlation structures are presented in 3.1.

The following generation method is based upon the Bahadur (1961) [1] representation of a multivariate binary distribution. We define $Z_t = [Y_t - \mu_t]/[\mu_t(1 - \mu_t)]^{1/2}$,

then the density presented in Bahadur (1961) [1] is,

$$f(y_1, \dots, y_T) = \left\{ \prod_{t=1}^T \mu_t^{y_t} (1 - \mu_t)^{1-y_t} \right\} \left\{ 1 + \sum_{1 \leq t < t' \leq T} E(Z_t Z_{t'}) z_t z_{t'} + \sum_{1 \leq t < t' < t'' \leq T} E(Z_t Z_{t'} Z_{t''}) z_t z_{t'} z_{t''} + \dots + E(Z_1 Z_2 \dots Z_T) z_1 z_2 \dots z_T \right\}.$$

Assuming correlations higher than second order are negligible, we approximate this by,

$$f(y_1, \dots, y_T) = \left\{ \prod_{t=1}^T \mu_t^{y_t} (1 - \mu_t)^{1-y_t} \right\} \left\{ 1 + \sum_{1 \leq t < t' \leq T} \rho_{tt'} z_t z_{t'} \right\}. \quad (2.2)$$

Using this joint density, the conditional probability $P(Y_t = 1 | Y_1 = y_1, \dots, Y_{t-1} = y_{t-1})$ can be found, and y_t can subsequently be generated. An interesting and advantageous property of (2.2) is that the form for the joint density for any subset of $\{y_1, \dots, y_T\}$ remains unchanged.

The technique for creating correlated binary data given by Kanter (1975) [10] is only designed for generating stationary autoregressive processes. However, McKenzie (1981) [12] extends it to stationary autoregressive - moving average (ARMA) models, and in Chapter 3 we propose an extension to allow nonstationary autoregressive processes to be generated.

To generate from an AR(p) process, Y_1 is generated from a Bernoulli random variable with mean μ . The subsequent Y_t are generated according to an autoregressive process of order $\min(t-1, p)$. Kanter (1975) [10] suggests the model

$$Y_t = \sum_{t'=1}^p U_t^{(t')} (Y_{t-t'} \oplus W_t) + \left(1 - \sum_{t'=1}^p U_t^{(t')} \right) W_t, \quad t \geq p+1$$

for generating Y_t according to an AR(p) process, where $U_t^{(t')}$ and W_t are Bernoulli random variables. Here we let the symbol $\oplus$ represents addition modulo 2. We define the probabilities $P(U_t^{(t')} = 1) = u^{(t')}$ and $P(W_t = 1) = w$ for some fixed $u^{(t')}$ and w

determined by the values of μ and the correlation parameters. We discuss this further in Chapter 3.

The methods reviewed above will all generate longitudinally correlated binary data, however certain obstacles arise with each. Gange (1995) [8] employs an iterative procedure, while Emrich and Piedmonte (1991) [5] and Omar and Zucker (2001) [13] require numerical integration. The techniques proposed by Emrich and Piedmonte (1991) [5], Lunn and Davies (1998) [11], Omar and Zucker (2001) [13] and Kanter (1975) [10] are limited to generating from relatively few correlation structures. With Qaqish (2003) [15] the need to determine values of $b_{tt'}$ for a particular correlation structure arises. The method proposed by Bahadur (1961) [1] is restricted to small magnitude correlations, while Park, Park and Shin (1996) [14] is applicable for non-negative correlations.

Qaqish (2003) [15] and Bahadur (1961) [1] are appealing for their relative simplicity and wide range of correlations. Although Kanter (1975) [10] is firmly defined for a particular correlation structure, we wish to explore the potential of such a method. Accordingly, the data generating methods proposed in Qaqish (2003) [15], Bahadur (1961) [1] and Kanter (1975) [10] are investigated further and hereon in will be referred to as QAQ, BAH and KAN respectively.

Chapter 3

Description of Generating Methods and Related Extensions

McKenzie (1981) [12] extends Kanter (1975) [10] to ARMA models, and we show here an adaptation which produces exchangeable correlation structures. Additionally, we extend Kanter (1975) [10] to produce nonstationary autoregressive processes.

In this work, we concentrate on the applicability of these methods to AR(1), MA(1) and EX correlations. However, we also demonstrate how to apply these methods to AR(2), which is not dealt with explicitly in any of the original papers.

3.1 QAQ

To obtain y_t under QAQ we use the following algorithm.

Step 1. Let $t = 1$. Generate a Bernoulli random variable with mean μ_1 . Set Y_1 to this value.

Step 2. Let $t = t + 1$ and compute $b_{tt'}$ for t and $t' = 1, \dots, t - 1$

Step 3. Evaluate λ_t in equation (2.1).

Step 4. Generate a Bernoulli random variable with mean λ_t . Set Y_t to this value.

If $t \leq T$, then go to Step 2.

Though any correlation structure is feasible with QAA, this flexibility is constrained by the need for the form of $b_{tt'}$ to be resolved for a specified R . The formulae of the $b_{tt'}$ for some common correlation structures are established in Appendix A, and are summarized below.

AR(1)	$b_{tt'} = \frac{\rho}{1+(t-2)\rho} \left(\frac{v_{tt}}{v_{t't'}} \right)^{\frac{1}{2}}, \quad t' = 1, \dots, t-1$
MA(1)	$b_{tt'} = \begin{cases} 0, & t' = 1, \dots, t-2 \\ \rho \left(\frac{v_{tt}}{v_{t't'}} \right)^{\frac{1}{2}}, & t' = t-1 \end{cases}$
EX	$b_{tt'} = \left(\frac{v_{tt}}{v_{t't'}} \right)^{\frac{1}{2}} \left[\frac{a^{t'} - a^{-t'}}{a^{-t} - a^t} \right], \quad a = \frac{\sqrt{1-4\rho^2}-1}{2\rho}$

Table 3.1: Formulae for $b_{tt'}$.

Equation (2.1) is also subject to another interpretation; it is reminiscent of a regression estimator in survey sampling, with the $b_{tt'}$ acting as regression coefficients.

3.1.1 AR(2) with QAA

Although we focus on the three structures given in Table 3.1, the QAA method is theoretically applicable to any correlation structure. However, before data for a certain correlation structure can be generated, the form of the $b_{tt'}$ must be derived. Specifically, QAA requires the computation of the inverse of the correlation matrix. If we consider an AR(2) correlation, this is an onerous task for $t > 3$. Using the Maple code in Appendix D, we calculate $b_{tt'}$ for $t = 10$,

$$b_{10,t'} = \begin{cases} \rho_1^9 + 8\rho_2\rho_1^7 + 21\rho_1^5\rho_2^2 + 20\rho_1^3\rho_2^3 + 5\rho_1\rho_2^4, & t' = 1 \\ 0, & t' \neq 1 \end{cases},$$

where ρ_1 and ρ_2 represent the two correlation parameters associated with an AR(2) process. Note that $Cov(Y_1, Y_{10}) = \rho_1^9 + 8\rho_2\rho_1^7 + 21\rho_1^5\rho_2^2 + 20\rho_1^3\rho_2^3 + 5\rho_1\rho_2^4$. Thus, for

AR(2) we conjecture that

$$b_{tt'} = \begin{cases} \text{Cov}(Y_1, Y_t) & , \quad t' = 1 \\ 0 & , \quad t' \neq 1 \end{cases}$$

for any t .

3.2 BAH

To generate data according to BAH we let $\mathbf{y} = (c_1, \dots, c_T)$ and compute equation (2.2) for all 2^T combinations of $c_1, \dots, c_T$ in $\{0, 1\}$. A random cluster is then generated by following the algorithm given by Devroye (1986) [4].

Step 1. Define $z_0, \dots, z_{2^T}$ by

$$z_0 = 0$$

$$z_j = z_{j-1} + f_j, \quad j = 1 \dots, 2^T - 1$$

$$z_{2^T} = 1$$

Step 2. Generate a random uniform deviate U on $[0, 1]$.

Step 3. Return the j th sequence where $z_j \leq U < z_{j+1}$.

For example, to generate data with BAH having an AR(2) correlation, we simply need to know the form of the correlation matrix. The values within the correlation matrix combined with (2.2) immediately provide the ability to generate the desired data.

3.3 KAN

It is apparent that KAN is much more restricted in terms of the variety of correlation structures that it can accommodate, as well as not being able to produce nonstationary data. McKenzie (1981) [12] provides an analogous model for ARMA, and below we introduce a model for generating EX data. The form of the extensions is the same, with the value of Y_t involving an addition modulo 2 operation on a combination of U_t , W_t and the previous values in the binary sequence. Table 3.2 shows the specific data generation models for each AR(1), MA(1) and EX, all of which will be denoted by KAN.

AR(1)	$Y_t = U_t\{Y_{t-1} \oplus W_t\} + (1 - U_t)W_t \quad t \geq 2$
MA(1)	$Y_t = U_t\{W_{t-1} \oplus W_t\} + (1 - U_t)W_t \quad t \geq 1$
EX	$Y_t = U_t\{Y_0 \oplus W_t\} + (1 - U_t)W_t \quad t \geq 1$

Table 3.2: Models for generating KAN.

Here, U_t and W_t are Bernoulli random variables where $P(U_t = 1) = u$ and $P(W_t = 1) = w$. Below are the generating algorithms for each correlation structure. The varying configurations of the models necessitate the slight differences among the algorithms.

For a stationary AR(1) process:

Step 1. Let $t = 1$. Generate a Bernoulli random variable with mean μ . Set Y_1 to this value.

Step 2. Let $t = t + 1$. Generate two Bernoulli random variables with means u and w . Set these values to U_t and W_t respectively.

Step 3. Evaluate Y_t for the AR(1) model in Table 3.2. If $t \leq T$, then go to Step 2.

For a stationary MA(1) process:

Step 1. Let $t = 1$. Generate a Bernoulli random variable with mean w . Set W_0 to this value.

Step 2. Let $t = t + 1$ and generate two Bernoulli random variables with means u and w . Set these values to U_t and W_t respectively.

Step 3. Evaluate Y_t for the MA(1) model in Table 3.2. If $t \leq T$, then go to Step 2.

For a stationary EX process:

Step 1. Let $t = 1$. Generate a Bernoulli random variable with mean μ . Set Y_0 to this value.

Step 2. Let $t = t + 1$. Generate two Bernoulli random variables with means u and w . Set these values to U_t and W_t respectively.

Step 3. Evaluate Y_t for the EX model in Table 3.2. If $t \leq T$, then go to Step 2.

The values of u and w are governed by μ , ρ and the correlation structure. In the AR(1) case the requirement of $E(Y_t) = \mu$ compels $w = \mu(1 - u)/(1 - 2u\mu) = \mu(1 - \rho)$, and we can also show that $\rho = u(1 - 2\mu)/(1 - 2u\mu)$. These derivations, as well as the corresponding results for MA(1) and EX, are presented in Appendix B. They are summarized in Table 3.3 below.

Correlation	ρ	u	w
AR(1)	$\frac{u(1-2\mu)}{1-2u\mu}$	$\frac{\rho}{1-2\mu(1-\rho)}$	$\mu(1-\rho)$
MA(1)	$\frac{(w-\mu)(1-2\mu)(1-w)}{(2w-1)\mu(1-\mu)}$	$\frac{\mu-w}{w(1-2w)}$	—
EX	$\left[\frac{w-\mu}{\mu}\right]^2$	$\frac{\sqrt{\rho}}{2\mu(1+\sqrt{\rho})-1}$	$\mu(1+\sqrt{\rho})$

 Table 3.3: Formulae for ρ , u and w in the KAN model.

3.3.1 Extension to Nonstationary

The original KAN method is seemingly restrictive, because it is confined to generating stationary data. We consider the analogous model to KAN for generating nonstationary AR(p) data, but retain the time subscript on the Bernoulli parameters defining W_t and $U_t^{(t')}$. The model itself,

$$Y_t = \sum_{t'=1}^p U_t^{(t')} (Y_{t-t'} \oplus W_t) + \left(1 - \sum_{t'=1}^p U_t^{(t')}\right) W_t, \quad t \geq p+1,$$

remains unchanged. However, the following probabilities are redefined as

$$P(U_t^{(t')} = 1) = u_t^{(t')} = \frac{\rho^{(t')} \sqrt{\mu_t(1-\mu_t)}}{\sqrt{\mu_{t-t'}(1-\mu_{t-t'})} \left(1 - 2\mu_t + 2\sqrt{\mu_t(1-\mu_t)} \sum_{j=1}^p \rho^{(j)} \sqrt{\frac{\mu_{t-j}}{1-\mu_{t-j}}}\right)}$$

$$P(W_t = 1) = w_t = \frac{\mu_t - \sum_{t'=1}^p u_t^{(t')} \mu_{t-t'}}{1 - 2 \sum_{t'=1}^p u_t^{(t')} \mu_{t-t'}},$$

where $1 \leq t \leq T$ and $1 \leq t' \leq p$. This alteration allows a nonstationary AR(p) process to be generated. The details of this extension are presented Appendix C.

Chapter 4

Range Restrictions

The fact that longitudinally correlated binary data are generated based on Bernoulli parameters that must lie between zero and one impose natural range restrictions on the associated correlation parameters, regardless of the correlation structures and the data generation mechanism. Firstly, $\text{corr}(Y_t, Y_{t'}) = \rho_{tt'}$ must be positive definite, while the second stems from restrictions imposed by μ_t and $\mu_{t'}$. Specifically, the binary nature of the variables implies that $\max(0, \mu_t + \mu_{t'} - 1) \leq E(Y_t Y_{t'}) \leq \min(\mu_t, \mu_{t'})$ $t \neq t'$. If we define $\psi_t = (\frac{\mu_t}{1-\mu_t})^{\frac{1}{2}}$, then

$$\max(-\psi_t \psi_{t'}, -\frac{1}{\psi_t \psi_{t'}}) \leq \rho_{tt'} \leq \min(\frac{\psi_t}{\psi_{t'}}, \frac{\psi_{t'}}{\psi_t}) t \neq t'$$

follows. (Qaqish, 2003)

Further range restrictions are imposed by the individual generating mechanisms. In certain instances these additional restrictions can become quite restrictive when compared to the natural restrictions. In the following discussion we consider stationary data only, and thus define $\mu = \mu_1 = \dots = \mu_T$ and $\psi = \psi_1 = \dots = \psi_T$. The principles are readily extendible to nonstationarity.

4.1 QAQ

In the case of QAQ, it is necessary that

$$0 \leq \lambda_t = \mu_t + \sum_{t'=1}^{t-1} b_{tt'}(Y_{t'} - \mu_t) \leq 1, t = 1, \dots, T \quad (4.1)$$

hold for λ_t to be a valid probability. Note that the $b_{tt'}$ contain the information about the correlation structure, and are in fact functions of the correlation parameters. With a specified correlation structure and a given μ , the restrictions on these parameters are obtained directly from (4.1).

It is noteworthy that for an AR(1) correlation no additional restrictions are imposed by the QAQ mechanism; the natural restrictions suffice in this instance. If we consider an exchangeable structure we notice that no new limitations are placed upon positive correlations, however negative correlations are restrained by

$$-[(n-1) \left\{ \max(\psi^2, \frac{1}{\psi^2}) + 1 \right\} - 1]^{-1} \leq \rho.$$

Bounds for ρ in the MA(1) structure can also be found explicitly .

4.2 BAH

The BAH technique also readily applies itself to acquiring range restrictions. In a similar manner to QAQ, we require

$$0 \leq \left[\prod_{t=1}^T \mu^{y_t} (1-\mu)^{1-y_t} \right] \left[1 + \frac{1}{\mu(1-\mu)} \sum_{1 \leq t < t' \leq T} \rho_{tt'} (y_t - \mu)(y_{t'} - \mu) \right] \leq 1,$$

or equivalently

$$-\mu(1-\mu) \leq \sum_{1 \leq t < t' \leq T} \rho_{tt'}(y_t - \mu)(y_{t'} - \mu) \leq \frac{\mu(1-\mu)}{\prod_{t=1}^T \mu^{y_t}(1-\mu)^{1-y_t}}, \quad (4.2)$$

be true so that $f(y_1, \dots, y_T)$ is a joint density. The range restrictions are given implicitly by (4.2). However, some correlation structures allow for a further reduction, such as MA(1) and exchangeable. For an MA(1) correlation the range restrictions simplify to

$$\frac{-\mu(1-\mu)}{(T-1)[\max\{\mu, 1-\mu\}]^2} \leq \rho \leq \frac{1}{T-1}.$$

Other correlations, for instance AR(1), do not reduce well. The AR(1) structure produces an inequality involving a $T-1$ polynomial in ρ which has no closed form solution for $T \geq 4$.

4.3 KAN

Finding range restrictions for KAN is not as simple as the two methods previously examined. As discussed in the previous chapter, to generate binary responses KAN requires a unique equation for each correlation structure. Another fundamental difference is the additional parameter u used in the KAN equations. The restrictions are initially imposed on u , and these indirectly control the range of ρ .

For example, in the AR(1) case we require that $0 \leq \mu(1-u)/(1-2u\mu) \leq 1$ and $0 \leq u \leq 1$ as both μ and u are Bernoulli probabilities. These combined restrictions imply that $0 \leq u \leq \min\left\{1, \frac{1-\mu}{\mu}\right\}$. Transforming this restriction into terms of ρ results in

$$\begin{aligned}
0 \leq \rho \leq 1 & \quad , \mu \leq \frac{1}{2} \\
\rho = 0 & \quad , \mu = \frac{1}{2} \cdot \\
-\frac{1-\mu}{\mu} \leq \rho \leq 0 & \quad , \mu > \frac{1}{2}
\end{aligned}$$

A similar process can be applied for MA(1) and exchangeable correlation structures with the initial Bernoulli probabilities changed accordingly.

4.3.1 Extension to KAN model

In this section we present an extension to the KAN model that widens the restrictions on the μ_t and ρ parameters. By defining $Y_t^* = 1 - Y_t$ we note that $E(Y_t^*) = E(1 - Y_t) = 1 - \mu_t$ and $Corr(Y_t^*, Y_{t'}^*) = Corr(Y_t, Y_{t'})$. The resulting models for the three correlation structures are given in Table 4.1. This new model will be referred to as KAN*. The ranges for the extended KAN model are the same ranges for the KAN model; however not surprisingly they are reflected about $\mu = 0.5$.

AR(1)	$Y_t^* = U_t(Y_{t-1} \oplus W_t) + (1 - U_t)(1 - W_t), t \geq 2$
MA(1)	$Y_t^* = 1 - U_t(W_{t-1} \oplus W_t) - (1 - U_t)W_t, t \geq 1$
EX	$Y_t^* = U_t(Y_0 \oplus W_t) + (1 - U_t)(1 - W_t), t \geq 1$

Table 4.1: Models for generating KAN*.

4.4 Range Restrictions for Correlation Parameters

In order to compare the three binary data generation methods in terms of range restrictions and correlation parameters we initially consider stationary AR(1), MA(1) and exchangeable structures over $T = 4$ time periods. For each of these three correlation structures we compute the appropriate range restrictions for the associated

correlation parameter, ρ , at each of $\mu = 0.05(0.05)0.95$. The numerical values of the range restrictions are given in this chapter in various tables, while graphical representations are provided in figures. Note that the tables of range restrictions do not provide those of KAN*; however these can be seen in the graphical displays.

4.4.1 Stationary Processes

The QAA method provides the broadest range for an AR(1) correlation, allowing any feasible combination of μ and ρ permitted by the natural restrictions. The BAH method mimics the QAA range in shape, but with a greatly reduced permissible range. KAN has the most unusual range of the three, being exactly half of that of QAA, and thus of the natural restrictions also. This final generating process does not tolerate positive ρ for $\mu > 1/2$ or negative ρ for $\mu < 1/2$. However, if we additionally consider the ranges allowed by KAN*, then the entire range permitted by the natural restrictions are obtained.

With an MA(1) correlation, unlike AR(1), the range for QAA does not encompass the ranges of the other two methods. In general QAA has the broadest range of the three, with its foremost advantage being a large domain containing negative ρ . BAH has a range similar in extent to QAA, but diminished in regard to negative ρ . Also, when μ is at the extremes, it allows for a greater expanse of positive ρ . KAN has the smallest range overall, but when grouped with KAN*, it provides the greatest possibilities for the combination of small or large μ with small ρ . It should be noted that the capacity of KAN and KAN* for negative correlations is minimal.

For the EX correlation structure QAA has the most extensive range, covering all positive correlations and a small set of negative correlations. BAH has a similar range to QAA for $\rho < 0.5$, but its ability to generate for $\rho > 0.5$ is almost absent. The KAN method can generate all positive correlation with any μ . It has greater permissible

ranges than BAH and is close to those ranges allowed by QAQ.

In general, over the three correlation structures examined, QAQ provides the widest ranges. The range supplied by BAH is fairly constant, with a band across small positive correlations for all μ . The combination of KAN and KAN* have irregular ranges, but are typically wide.

4.4.2 Nonstationary Range Restrictions

In this section the range restrictions for two nonstationary processes are presented. The nonstationarity is characterized by two transformations which are defined in Section 5.2.

Similar to the stationary case, QAQ has the greatest ranges while BAH is applicable for most correlations of small magnitude. The KAN and KAN* methods are only applicable to an AR(1) nonstationary process, but for this correlation permit large ranges.

Figure 4.1: AR(1) stationary range restrictions.

μ	<i>QAQ</i>	<i>BAH</i>	<i>KAN</i>
0.05	(-0.053 , 1.000)	(-0.018 , 0.427)	(0.000 , 1.000)
0.10	(-0.111 , 1.000)	(-0.038 , 0.442)	(0.000 , 1.000)
0.15	(-0.176 , 1.000)	(-0.061 , 0.462)	(0.000 , 1.000)
0.20	(-0.250 , 1.000)	(-0.088 , 0.486)	(0.000 , 1.000)
0.25	(-0.333 , 1.000)	(-0.120 , 0.520)	(0.000 , 1.000)
0.30	(-0.429 , 1.000)	(-0.158 , 0.503)	(0.000 , 1.000)
0.35	(-0.538 , 1.000)	(-0.205 , 0.481)	(0.000 , 1.000)
0.40	(-0.667 , 1.000)	(-0.262 , 0.449)	(0.000 , 1.000)
0.45	(-0.818 , 1.000)	(-0.335 , 0.434)	(0.000 , 1.000)
0.50	(-1.000 , 1.000)	(-0.430 , 0.430)	(0.000 , 0.000)
0.55	(-0.818 , 1.000)	(-0.335 , 0.434)	(-0.818 , 0.000)
0.60	(-0.667 , 1.000)	(-0.262 , 0.449)	(-0.667 , 0.000)
0.65	(-0.538 , 1.000)	(-0.205 , 0.481)	(-0.538 , 0.000)
0.70	(-0.429 , 1.000)	(-0.158 , 0.503)	(-0.429 , 0.000)
0.75	(-0.333 , 1.000)	(-0.120 , 0.520)	(-0.333 , 0.000)
0.80	(-0.250 , 1.000)	(-0.088 , 0.486)	(-0.250 , 0.000)
0.85	(-0.176 , 1.000)	(-0.061 , 0.462)	(-0.176 , 0.000)
0.90	(-0.111 , 1.000)	(-0.038 , 0.442)	(-0.111 , 0.000)
0.95	(-0.053 , 1.000)	(-0.018 , 0.427)	(-0.053 , 0.000)

Table 4.2: AR(1) stationary range restrictions.

Figure 4.2: MA(1) stationary range restrictions.

μ	QAQ	BAH	KAN
0.05	(-0.050 , 0.196)	(-0.018 , 0.333)	(0.000 , 0.474)
0.10	(-0.099 , 0.259)	(-0.037 , 0.333)	(0.000 , 0.444)
0.15	(-0.147 , 0.302)	(-0.059 , 0.333)	(0.000 , 0.412)
0.20	(-0.194 , 0.333)	(-0.083 , 0.333)	(0.000 , 0.375)
0.25	(-0.239 , 0.358)	(-0.111 , 0.333)	(0.000 , 0.333)
0.30	(-0.281 , 0.379)	(-0.143 , 0.333)	(0.000 , 0.286)
0.35	(-0.322 , 0.396)	(-0.179 , 0.333)	(0.000 , 0.231)
0.40	(-0.362 , 0.410)	(-0.222 , 0.333)	(0.000 , 0.167)
0.45	(-0.399 , 0.423)	(-0.273 , 0.333)	(0.000 , 0.091)
0.50	(-0.434 , 0.434)	(-0.333 , 0.333)	(0.000 , 0.000)
0.55	(-0.399 , 0.423)	(-0.273 , 0.333)	(-0.047 , 0.000)
0.60	(-0.362 , 0.410)	(-0.222 , 0.333)	(-0.064 , 0.000)
0.65	(-0.322 , 0.396)	(-0.179 , 0.333)	(-0.067 , 0.000)
0.70	(-0.281 , 0.379)	(-0.143 , 0.333)	(-0.064 , 0.000)
0.75	(-0.239 , 0.358)	(-0.111 , 0.333)	(-0.057 , 0.000)
0.80	(-0.194 , 0.333)	(-0.083 , 0.333)	(-0.048 , 0.000)
0.85	(-0.147 , 0.302)	(-0.059 , 0.333)	(-0.037 , 0.000)
0.90	(-0.099 , 0.259)	(-0.037 , 0.333)	(-0.025 , 0.000)
0.95	(-0.050 , 0.196)	(-0.018 , 0.333)	(-0.012 , 0.000)

Table 4.3: MA(1) stationary range restrictions.

Figure 4.3: EX stationary range restrictions.

μ	QAQ	BAH	KAN
0.05	(-0.017 , 1.000)	(-0.009 , 0.352)	(0.000 , 1.000)
0.10	(-0.034 , 1.000)	(-0.019 , 0.375)	(0.000 , 1.000)
0.15	(-0.053 , 1.000)	(-0.029 , 0.405)	(0.000 , 1.000)
0.20	(-0.071 , 1.000)	(-0.042 , 0.444)	(0.000 , 1.000)
0.25	(-0.091 , 1.000)	(-0.056 , 0.500)	(0.000 , 1.000)
0.30	(-0.111 , 1.000)	(-0.071 , 0.583)	(0.000 , 1.000)
0.35	(-0.132 , 1.000)	(-0.090 , 0.623)	(0.000 , 1.000)
0.40	(-0.154 , 1.000)	(-0.111 , 0.545)	(0.000 , 1.000)
0.45	(-0.176 , 1.000)	(-0.136 , 0.510)	(0.000 , 1.000)
0.50	(-0.200 , 1.000)	(-0.167 , 0.500)	(0.000 , 1.000)
0.55	(-0.176 , 1.000)	(-0.136 , 0.510)	(0.000 , 1.000)
0.60	(-0.154 , 1.000)	(-0.111 , 0.545)	(0.000 , 1.000)
0.65	(-0.132 , 1.000)	(-0.090 , 0.623)	(0.000 , 1.000)
0.70	(-0.111 , 1.000)	(-0.071 , 0.583)	(0.000 , 1.000)
0.75	(-0.091 , 1.000)	(-0.056 , 0.500)	(0.000 , 1.000)
0.80	(-0.071 , 1.000)	(-0.042 , 0.444)	(0.000 , 1.000)
0.85	(-0.053 , 1.000)	(-0.029 , 0.405)	(0.000 , 1.000)
0.90	(-0.034 , 1.000)	(-0.019 , 0.375)	(0.000 , 1.000)
0.95	(-0.017 , 1.000)	(-0.009 , 0.352)	(0.000 , 1.000)

Table 4.4: EX stationary range restrictions.

Figure 4.4: AR(1) Transformation 1 nonstationary range restrictions.

μ	<i>QAQ</i>	<i>BAH</i>	<i>KAN</i>
0.05	(-0.025 , 0.415)	(-0.067 , 0.978)	(0.000 , 0.985)
0.10	(-0.055 , 0.439)	(-0.139 , 0.958)	(0.000 , 0.963)
0.15	(-0.089 , 0.470)	(-0.217 , 0.940)	(0.000 , 0.932)
0.20	(-0.129 , 0.501)	(-0.304 , 0.923)	(0.000 , 0.879)
0.25	(-0.177 , 0.475)	(-0.400 , 0.908)	(0.000 , 0.774)
0.30	(-0.236 , 0.428)	(-0.507 , 0.895)	(0.000 , 0.457)
0.35	(-0.309 , 0.411)	(-0.629 , 0.882)	—
0.40	(-0.402 , 0.405)	(-0.769 , 0.871)	—
0.45	(-0.336 , 0.408)	(-0.637 , 0.860)	(-0.660 , 0.000)
0.50	(-0.264 , 0.419)	(-0.515 , 0.850)	(-1.000 , 0.000)
0.55	(-0.209 , 0.442)	(-0.417 , 0.840)	(-1.000 , 0.000)
0.60	(-0.166 , 0.489)	(-0.336 , 0.832)	(-1.000 , 0.000)
0.65	(-0.131 , 0.500)	(-0.269 , 0.824)	(-1.000 , 0.000)
0.70	(-0.103 , 0.484)	(-0.212 , 0.816)	(-1.000 , 0.000)
0.75	(-0.079 , 0.461)	(-0.164 , 0.809)	(-1.000 , 0.000)
0.80	(-0.058 , 0.442)	(-0.122 , 0.802)	(-1.000 , 0.000)
0.85	(-0.041 , 0.427)	(-0.085 , 0.796)	(-1.000 , 0.000)
0.90	(-0.025 , 0.415)	(-0.053 , 0.790)	(-1.000 , 0.000)
0.95	(-0.012 , 0.405)	(-0.025 , 0.784)	(-1.000 , 0.000)

Table 4.5: AR(1) Transformation 1 nonstationary range restrictions.

Figure 4.5: AR(1) Transformation 2 nonstationary range restrictions.

μ	<i>QAQ</i>	<i>BAH</i>	<i>KAN</i>
0.05	(-0.012 , 0.405)	(-0.025 , 0.784)	(0.000 , 0.776)
0.10	(-0.025 , 0.415)	(-0.053 , 0.790)	(0.000 , 0.772)
0.15	(-0.041 , 0.427)	(-0.085 , 0.796)	(0.000 , 0.767)
0.20	(-0.058 , 0.442)	(-0.122 , 0.802)	(0.000 , 0.761)
0.25	(-0.079 , 0.461)	(-0.164 , 0.809)	(0.000 , 0.753)
0.30	(-0.103 , 0.484)	(-0.212 , 0.816)	(0.000 , 0.742)
0.35	(-0.131 , 0.500)	(-0.269 , 0.824)	(0.000 , 0.726)
0.40	(-0.166 , 0.489)	(-0.336 , 0.832)	(0.000 , 0.701)
0.45	(-0.209 , 0.442)	(-0.417 , 0.840)	(0.000 , 0.659)
0.50	(-0.264 , 0.419)	(-0.515 , 0.850)	(0.000 , 0.569)
0.55	(-0.336 , 0.408)	(-0.637 , 0.860)	(0.000 , 0.251)
0.60	(-0.381 , 0.405)	(-0.769 , 0.871)	—
0.65	(-0.293 , 0.413)	(-0.629 , 0.882)	—
0.70	(-0.211 , 0.441)	(-0.507 , 0.895)	(-0.368 , 1.000)
0.75	(-0.157 , 0.490)	(-0.400 , 0.908)	(-0.347 , 0.000)
0.80	(-0.112 , 0.494)	(-0.304 , 0.923)	(-0.269 , 0.000)
0.85	(-0.075 , 0.457)	(-0.217 , 0.940)	(-0.193 , 0.000)
0.90	(-0.042 , 0.429)	(-0.139 , 0.958)	(-0.124 , 0.000)
0.95	(-0.015 , 0.407)	(-0.067 , 0.978)	(-0.059 , 0.000)

Table 4.6: AR(1) Transformation 2 nonstationary range restrictions.

Figure 4.6: MA(1) Transformation 1 nonstationary range restrictions.

μ	QAQ	BAH
0.01	(-0.005 , 0.318)	(-0.013 , 0.121)
0.05	(-0.025 , 0.318)	(-0.067 , 0.249)
0.10	(-0.053 , 0.318)	(-0.139 , 0.328)
0.15	(-0.084 , 0.318)	(-0.215 , 0.375)
0.20	(-0.119 , 0.318)	(-0.276 , 0.402)
0.25	(-0.158 , 0.318)	(-0.332 , 0.422)
0.30	(-0.204 , 0.318)	(-0.385 , 0.432)
0.35	(-0.256 , 0.318)	(-0.425 , 0.419)
0.40	(-0.317 , 0.318)	(-0.404 , 0.406)
0.45	(-0.274 , 0.318)	(-0.363 , 0.392)
0.50	(-0.224 , 0.318)	(-0.322 , 0.378)
0.55	(-0.184 , 0.318)	(-0.282 , 0.362)
0.60	(-0.150 , 0.318)	(-0.244 , 0.345)
0.65	(-0.121 , 0.318)	(-0.207 , 0.327)
0.70	(-0.096 , 0.318)	(-0.172 , 0.307)
0.75	(-0.075 , 0.318)	(-0.139 , 0.285)
0.80	(-0.056 , 0.318)	(-0.108 , 0.259)
0.85	(-0.040 , 0.318)	(-0.078 , 0.229)
0.90	(-0.025 , 0.318)	(-0.050 , 0.192)
0.95	(-0.012 , 0.318)	(-0.024 , 0.140)

Table 4.7: MA(1) Transformation 1 nonstationary range restrictions.

Figure 4.7: MA(1) Transformation 2 nonstationary range restrictions.

μ	QAQ	BAH
0.01	(-0.002 , 0.318)	(-0.005 , 0.066)
0.05	(-0.012 , 0.318)	(-0.024 , 0.140)
0.10	(-0.025 , 0.318)	(-0.050 , 0.192)
0.15	(-0.040 , 0.318)	(-0.078 , 0.229)
0.20	(-0.056 , 0.318)	(-0.108 , 0.259)
0.25	(-0.075 , 0.318)	(-0.139 , 0.285)
0.30	(-0.096 , 0.318)	(-0.172 , 0.307)
0.35	(-0.121 , 0.318)	(-0.207 , 0.327)
0.40	(-0.150 , 0.318)	(-0.244 , 0.345)
0.45	(-0.184 , 0.318)	(-0.282 , 0.362)
0.50	(-0.224 , 0.318)	(-0.322 , 0.378)
0.55	(-0.274 , 0.318)	(-0.363 , 0.392)
0.60	(-0.317 , 0.318)	(-0.404 , 0.406)
0.65	(-0.256 , 0.318)	(-0.425 , 0.419)
0.70	(-0.204 , 0.318)	(-0.385 , 0.432)
0.75	(-0.158 , 0.318)	(-0.332 , 0.422)
0.80	(-0.119 , 0.318)	(-0.276 , 0.402)
0.85	(-0.084 , 0.318)	(-0.215 , 0.375)
0.90	(-0.053 , 0.318)	(-0.139 , 0.328)
0.95	(-0.025 , 0.318)	(-0.067 , 0.249)

Table 4.8: MA(1) Transformation 2 nonstationary range restrictions.

Figure 4.8: EX Transformation 1 nonstationary range restrictions.

μ	QAQ	BAH
0.01	(-0.002 , 0.262)	(-0.007 , 0.976)
0.05	(-0.013 , 0.277)	(-0.033 , 0.893)
0.10	(-0.027 , 0.301)	(-0.063 , 0.809)
0.15	(-0.042 , 0.334)	(-0.091 , 0.740)
0.20	(-0.060 , 0.380)	(-0.117 , 0.683)
0.25	(-0.080 , 0.451)	(-0.142 , 0.635)
0.30	(-0.103 , 0.435)	(-0.165 , 0.594)
0.35	(-0.129 , 0.398)	(-0.188 , 0.559)
0.40	(-0.160 , 0.387)	(-0.187 , 0.528)
0.45	(-0.138 , 0.392)	(-0.161 , 0.500)
0.50	(-0.113 , 0.416)	(-0.138 , 0.476)
0.55	(-0.093 , 0.465)	(-0.117 , 0.454)
0.60	(-0.075 , 0.434)	(-0.098 , 0.435)
0.65	(-0.061 , 0.384)	(-0.081 , 0.417)
0.70	(-0.048 , 0.349)	(-0.065 , 0.401)
0.75	(-0.038 , 0.324)	(-0.051 , 0.386)
0.80	(-0.028 , 0.304)	(-0.039 , 0.373)
0.85	(-0.020 , 0.289)	(-0.028 , 0.360)
0.90	(-0.013 , 0.277)	(-0.017 , 0.349)
0.95	(-0.006 , 0.267)	(-0.008 , 0.338)

Table 4.9: EX Transformation 1 nonstationary range restrictions.

Figure 4.9: EX Transformation 2 nonstationary range restrictions.

μ	<i>QAQ</i>	<i>BAH</i>
0.01	(-0.001 , 0.260)	(-0.002 , 0.330)
0.05	(-0.006 , 0.267)	(-0.008 , 0.338)
0.10	(-0.013 , 0.277)	(-0.017 , 0.349)
0.15	(-0.020 , 0.289)	(-0.028 , 0.360)
0.20	(-0.028 , 0.304)	(-0.039 , 0.373)
0.25	(-0.038 , 0.324)	(-0.051 , 0.386)
0.30	(-0.048 , 0.349)	(-0.065 , 0.401)
0.35	(-0.061 , 0.384)	(-0.081 , 0.417)
0.40	(-0.075 , 0.434)	(-0.098 , 0.435)
0.45	(-0.093 , 0.465)	(-0.117 , 0.454)
0.50	(-0.113 , 0.416)	(-0.138 , 0.476)
0.55	(-0.138 , 0.392)	(-0.161 , 0.500)
0.60	(-0.160 , 0.387)	(-0.187 , 0.528)
0.65	(-0.129 , 0.398)	(-0.188 , 0.559)
0.70	(-0.103 , 0.435)	(-0.165 , 0.594)
0.75	(-0.080 , 0.451)	(-0.142 , 0.635)
0.80	(-0.060 , 0.380)	(-0.117 , 0.683)
0.85	(-0.042 , 0.334)	(-0.091 , 0.740)
0.90	(-0.027 , 0.301)	(-0.063 , 0.809)
0.95	(-0.013 , 0.277)	(-0.033 , 0.893)

Table 4.10: EX Transformation 2 nonstationary range restrictions.

Chapter 5

Model and Simulation Study

In any application we will typically develop a model to describe the data, and later will estimate the model parameters. We naturally would like to analyze and evaluate the properties of these estimators. An examination of this type can be difficult to perform analytically; as such asymptotic approximations are common. In addition, in longitudinal data the finite sample properties can be more of interest. An investigation of these properties often necessitates Monte Carlo simulation and consequently a choice of data generation mechanisms is required. To evaluate the characteristics of the data generated by each of the three generating methods examined here (QAQ, BAH and KAN) we performed a simulation study.

Each method was used to generate 1000 data sets consisting of longitudinal binary sequences of four observations for each of 100 independent clusters. We consider both stationary and nonstationary correlation structures. For stationary processes we investigate situations where groups of clusters have both differing means and correlation parameters, as well as cases in which only the means differ (same correlation parameter). In the nonstationary scenario we only consider the latter. The choice of designs where all clusters regardless of group possess the same correlation parameter is motivated by what typically is assumed in practice.

5.1 The Model

We refer to a collection of clusters which shares the same correlation structure and correlation parameter as a group, and we let G be the number of such groups within a design, which consists of I independent clusters. Also let I_g , $g = 1, \dots, G$, represent the number of clusters in group g , and $I = \sum_{g=1}^G I_g$ the total number of clusters.

We choose to investigate a logistic regression model with no intercept,

$$P(Y_{git} = 1) = \frac{\exp(\mathbf{X}_{gt}\boldsymbol{\beta})}{1 + \exp(\mathbf{X}_{gt}\boldsymbol{\beta})}$$

where $\boldsymbol{\beta}$ is a vector of p parameters of interest and $\mathbf{X}_{gt} = (X_{gt1}, \dots, X_{gtp})^T$ is a vector of covariates for time period t , of any cluster in group g . We also define the $T \times p$ matrix $\mathbf{X}_g = (\mathbf{X}_{g1}^T, \dots, \mathbf{X}_{gT}^T)^T$.

5.2 Designs

We let ρ_g and $\boldsymbol{\mu}_g = E(Y_{git}) = (\mu_{g1}, \dots, \mu_{gT})^T$ represent the correlation parameter and the vector of means for all clusters in the g^{th} group respectively, $g = 1, \dots, G$. Several designs involving various sets of $\boldsymbol{\mu}_g$, ρ_g , and correlation structures were chosen. In all instances of the model, $p = 2$ and $\boldsymbol{\beta} = (1, 1)^T$. The tables summarizing the simulation results in Chapter 6 indicate the choice of the other parameters for each design, along with the value specified for $\mathbf{X}_g$ and I_g . In the stationary designs, the values of X_{gt} remain constant over time. The nonstationarity of some of the designs is introduced by describing the $\mathbf{X}_g$'s as a function of time. Specifically, two transformations are used, both set $X_{gt1} = \log(\mu_{gt}) - X_{gt2}$ where $X_{gt2} = 1/t$ for the first transformation and $X_{gt2} = -1/t$ for the second, where $t = 1, \dots, T$. The choice of values for $\mathbf{X}_g$ is an extremely important consideration, as certain selections can lead to an ill-conditioned

design matrix. We discuss this further in Section 5.5.

The designs chosen contained clusters classified into between one to three groups. Most of the designs have $I = 100$; we discuss this choice for the total number of clusters as a technical consideration in Section 5.5. Depending on the range restrictions, certain designs prohibit one or two of the generating methods from being applicable. The choice of designs displays a wide variety of combinations of μ_g , ρ , and correlation structures allowed by at least one of the methods. For any given design and valid generating technique we accordingly generate I binary response sequences of $T = 4$, and repeat $n = 1000$ times. Note that we verified that each data generation mechanism was producing binary responses with the specified correlation structure by computing lag correlations for pairs of time periods from one through four.

5.3 Estimation

To better understand the attributes of the different generating methods we estimate ρ_g and β . We use the intuitive moment estimator given by Sutradhar (2003) [17],

$$\hat{\rho}_g = \frac{\sum_{i=1}^{I_g} \sum_{t=1}^{T-1} \tilde{y}_{git} \tilde{y}_{g,i,t+1} / I_g(T-1)}{\sum_{i=1}^{I_g} \sum_{t=1}^T \tilde{y}_{git}^2 / I_g T},$$

where $\tilde{y}_{git} = (y_{git} - \mu_{gt}) / \sqrt{\mu_{gt}(1 - \mu_{gt})}$ is the standardized response corresponding to time period $t = 1, \dots, T$ of cluster i in group g . The estimator $\hat{\rho}_g$ provides a single estimate of ρ_g for the entire group g .

In practical applications, it is typical to assume for modelling purposes that clusters have different means, but a constant correlation across all clusters. For designs in which such a situation was specified we use an estimator pooled over groups:

$$\hat{\rho}_{pl} = \frac{\sum_{g=1}^G \sum_{i=1}^{I_g} \sum_{t=1}^{T-1} \tilde{y}_{git} \tilde{y}_{g,i,t+1} / I_g (T-1)}{\sum_{g=1}^G \sum_{i=1}^{I_g} \sum_{t=1}^T \tilde{y}_{git}^2 / I_g T}.$$

Of particular interest is the proportion of estimates of ρ which are outside of the theoretical range for a given generating method. An unduly high percentage could be an indication that the method is inadequate for data generation in those circumstances. Should an estimate of ρ be beyond its range we continue use of that estimate, but note a range restriction violation. The possibility of observing such violations is greatest when ρ is near the range boundary. As such, a number of the designs focus on these situations.

The estimation procedure that we use for β is the GQL approach presented in Sutradhar (2003) [17]. We let $\mathbf{y}_{gi} = (y_{gi1}, y_{gi2}, \dots, y_{giT})$ represent the observed binary outcomes. We also define $A_g = \text{diag}[\mu_{g1}(1 - \mu_{g1}), \dots, \mu_{gT}(1 - \mu_{gT})]$ as a diagonal matrix with the variances of Y_{git} , for any particular cluster i in group g , on the diagonal. Finally, $\Sigma_g(\hat{\rho}_g) = A_g^{1/2} C(\hat{\rho}_g) A_g^{1/2}$ is the true covariance of $\mathbf{y}_{gi}$ and $C(\rho_g)$ is the true correlation structure.

We first estimate ρ_g for each $g = 1, \dots, G$, and then use the $\hat{\rho}_g$ to estimate β , which is obtained by solving,

$$\sum_{g=1}^G \sum_{i=1}^{I_g} \mathbf{X}_g^T \mathbf{A}_g \Sigma_g^{-1}(\hat{\rho}_g) (\mathbf{y}_{gi} - \boldsymbol{\mu}_g) = \mathbf{0}.$$

An iterative procedure is then commenced, where the ρ_g are estimated again based upon the estimate of $\boldsymbol{\mu}_g$ prompted by the estimate of β . This process continues until a stopping criteria is met, typically when the relative error is beyond a designated threshold.

5.4 Generation

The generating process varies greatly for each method, however the initial step is the same for all. Every process requires y_{gi1} to be set as a randomly generated Bernoulli with probability of success μ_{g1} . The uniqueness and creativity in each method is apparent by the differences in forming $y_{gi2}, \dots, y_{giT}$.

Generating from the QAA method entails setting y_{git} as a Bernoulli random variable with probability λ_g , an adjusted value for μ_g . The variable λ_g is adjusted by a linear combination of $(b_{g1}, \dots, b_{g,t-1})$ and $(y_{gi1}, \dots, y_{g,i,t-1})$. The $b_g \equiv b$ are defined in Section 3.1 for each of the three correlation structures discussed.

The BAH method provides a formula for the joint distribution of $(y_{gi1}, \dots, y_{git})$, $t = 1, \dots, T$. We can thus easily find the conditional distribution of y_{git} by using the joint distributions of $(y_{gi1}, \dots, y_{git})$ and $(y_{gi1}, \dots, y_{g,i,t-1})$. Employing this conditional distribution we can then generate y_{git} .

To generate y_{git} from KAN we rely on two independent random Bernoulli variables U_{git} and W_{git} , and depending on the correlation structure, some previous value of Y_{git} . We set $P(U_{git} = 1) = u_{gt}$ where u_{gt} is some fixed constant and $P(W_{git} = 1) = w_{gt}$ where w_{gt} is dependent on u_{gt} and μ_{gt} . Each correlation structure has differing formulae for w_{gt} and Y_{git} , which are available in Section 3.3. The randomness of U_{git} and W_{git} provides entirely for the randomness in Y_{git} . The use of modulo base 2 in generating Y_{git} is fundamental to the KAN method.

5.5 Technical Circumstances To Consider

It is possible to construct designs for which plausible results are technically unattainable, but are theoretically feasible. A design is prone to technical problems if the estimation encounters invertibility issues with a design matrix for the covariates. If

the $\mathbf{X}_g$ matrix is ill-conditioned, estimation may be poor. To illustrate this we considered a stationary AR(1) design consisting of two groups of 50 clusters with means of 0.40 and 0.75 and correlation parameters of -0.15 and 0.40 respectively. Keeping I_g , μ_g and ρ_g constant as specified above, we considered a simulation study involving four different X_g design matrices as summarized in Table 5.1. Correlated binary responses were generated for $T = 4$ using QAQ for $n = 1000$ simulated data sets. The results obtained for estimating β and ρ using a GQL procedure are given in the right-hand panel of the table. Note that the range restriction violations have been omitted from this table and subsequent ones in the section as no violations occurred in the designs.

The mean estimates of the correlation parameters and the estimated standard deviations of ρ and $\hat{\rho}$ remain stable as the design matrix becomes more ill-conditioned; however the counterparts for β and $\hat{\beta}$ are greatly affected.

Another troublesome situation occurs when the outcome across all time periods is the same for each cluster within a group (but not necessarily the same over the whole group). The estimate for ρ_g corresponding to such a group is 1, and the estimated correlation matrix is exactly singular. The GQL procedure will be incapable of producing an estimate as it relies upon inverting this matrix. This situation can occur with simultaneously large ρ_g and extreme values for μ_g . The random aspect of the simulation determines whether a specific instance of the design will fail.

A final consideration to note is the total number of clusters necessary for sufficiently accurate estimation of β and the correlation parameter. Table 5.2 illustrates that the mean estimates for β and the correlation parameter are closer to their true values as I increases. Also, the estimated standard errors for both estimators decrease with more total number of clusters. It was also verified that each generating method simulated the appropriate correlation structure. Note that we use $I = 100$ in what

AR(1) Design				Estimates	
I_g	μ_g	ρ_g	X'_{g1}	QAQ	
50	.40	-.15	(-.20, -.20)	$\hat{\rho}_1$	-.1558 (.080)
50	.75	.40	(.55, .55)	$\hat{\rho}_2$	.3928 (.090)
				$\hat{\beta}_1$	.6202 (25.548)
				$\hat{\beta}_2$	1.4098 (25.388)
50	.40	-.15	(-.20, -.20)	$\hat{\rho}_1$	-.1558 (.080)
50	.75	.40	(.50, .60)	$\hat{\rho}_2$	.3928 (.090)
				$\hat{\beta}_1$	1.0977 (5.254)
				$\hat{\beta}_2$	.9451 (4.595)
50	.40	-.15	(-.20, -.20)	$\hat{\rho}_1$	-.1558 (.080)
50	.75	.40	(.25, .85)	$\hat{\rho}_2$	.3928 (.090)
				$\hat{\beta}_1$	1.0314 (.994)
				$\hat{\beta}_2$	1.0095 (.466)
50	.40	-.15	(-.20, -.20)	$\hat{\rho}_1$	-.1558 (.080)
50	.75	.40	(.00, 1.10)	$\hat{\rho}_2$	.3928 (.090)
				$\hat{\beta}_1$	1.0263 (.671)
				$\hat{\beta}_2$	1.0145 (.204)

Table 5.1: Results for designs demonstrating the effect of ill-conditioned X_g matrices.

follows in Chapter 6.

AR(1) Nonstationary Design				Estimates			
I_g	μ_{g1}	ρ_g	X'_{g1}		QAQ	BAH	KAN
10	.65	.35	(-.87, -.25)	$\hat{\rho}_{pl}$	.3460 (.095)	.3381 (.086)	—
20	.35	.35	(-.37, -.25)		0	20	—
10	.50	.35	(.25, -.25)	$\hat{\beta}_1$	1.0722 (.451)	1.0714 (.441)	—
Transformation 2				$\hat{\beta}_2$	1.0216 (.341)	1.0168 (.339)	—
25	.65	.35	(-.87, -.25)	$\hat{\rho}_{pl}$	.3517 (.059)	.3462 (.054)	—
50	.35	.35	(-.37, -.25)		0	—	—
25	.50	.35	(.25, -.25)	$\hat{\beta}_1$	1.0380 (.270)	1.0255 (.276)	—
Transformation 2				$\hat{\beta}_2$	1.0235 (.207)	1.0094 (.206)	—
250	.65	.35	(-.87, -.25)	$\hat{\rho}_{pl}$	.3613 (.018)	.3500 (.018)	—
500	.35	.35	(-.37, -.25)		0	— 0	—
250	.50	.35	(.25, -.25)	$\hat{\beta}_1$	1.0027 (.083)	1.0040 (.083)	—
Transformation 2				$\hat{\beta}_2$	1.0004 (.066)	1.0014 (.064)	—
10	.65	.35	(-.87, -.25)	$\hat{\rho}_{pl}$	.3460 (.095)	.3381 (.086)	—
20	.35	.35	(-.37, -.25)	$\hat{\beta}_1$	1.0722 (.451)	1.0714 (.441)	—
10	.50	.35	(.25, -.25)	$\hat{\beta}_2$	1.0216 (.341)	1.0168 (.339)	—
Transformation 2							
25	.65	.35	(-.87, -.25)	$\hat{\rho}_{pl}$	.3517 (.059)	.3462 (.054)	—
50	.35	.35	(-.37, -.25)	$\hat{\beta}_1$	1.0380 (.270)	1.0255 (.276)	—
25	.50	.35	(.25, -.25)	$\hat{\beta}_2$	1.0235 (.207)	1.0094 (.206)	—
Transformation 2							
250	.65	.35	(-.87, -.25)	$\hat{\rho}_{pl}$	.3613 (.018)	.3500 (.018)	—
500	.35	.35	(-.37, -.25)	$\hat{\beta}_1$	1.0027 (.083)	1.0040 (.083)	—
250	.50	.35	(.25, -.25)	$\hat{\beta}_2$	1.0004 (.066)	1.0014 (.064)	—
Transformation 2							

Table 5.2: Results for designs demonstrating the effect of cluster size.

Chapter 6

Simulation Results

The results of the simulation study are presented in this Chapter, and summarized in Tables 6.1 to 6.9. The values for I_g , μ_g , ρ_g and X_{gt} for each design are given in the left-hand side panel of the tables. The right-hand side provides the averages of the estimates for β and the correlation parameters with estimated standard deviations in brackets. We assume that any differences in these values across the methods are attributable to the generating mechanism. The values below the average estimates for the correlation parameters reflect the percentage of range restriction violations over the 1000 simulated data sets.

6.1 Stationary Designs

Tables 6.1 through 6.3 contain the simulation results for a variety of stationary designs with AR(1), MA(1) and EX correlation structures respectively. These initial designs all investigate scenarios where the correlation parameter ρ_g is not the same for all I clusters. Regardless of the data generation mechanism used, the means of the estimates for both β and ρ_g are close to the true values for these parameters and there is little difference in the estimates and standard errors obtained under

QAQ, BAH and KAN. Generally speaking however, the standard deviations of $\hat{\rho}_g$ appear to be lowest when data are generated according to BAH. The mean estimates of ρ_g under the three data generation methods seem to be consistently below ρ_g , suggesting that the estimators likely possess a slight negative bias. On the other hand, the estimators of the coefficient vector β appear to be positively biased, with those based on KAN being the least so. The estimated standard deviations of the estimators for β are mostly similar, but the methods do display slight variations over different correlations structures.

AR(1) Design				Estimates			
I_g	μ_g	ρ_g	X'_{g1}		QAQ	BAH	KAN
25	.50	.00	(1.00, -1.00)	$\hat{\rho}_1$	-.0079 (.115) 0	-.0079 (.115) 0	-.0079 (.115) 100
50	.10	.25	(-1.20, -1.00)	$\hat{\rho}_2$	.2305 (.127) 0	.2340 (.111) 3	.2339 (.127) 3
25	.30	.38	(.00, -.85)	$\hat{\rho}_3$	.3734 (.118) 0	.3814 (.112) 14	.3765 (.116) 0
				$\hat{\beta}_1$	1.0041 (.158)	1.0026 (.158)	1.0070 (.161)
				$\hat{\beta}_2$	1.0234 (.158)	1.0212 (.156)	1.0195 (.154)
50	.30	.38	(-2.00, 1.15)	$\hat{\rho}_1$	.3756 (.083) 0	.3761 (.077) 5	.3726 (.084) 0
50	.60	-.20	(-1.60, 2.01)	$\hat{\rho}_2$	-.2095 (.077) 0	-.2085 (.075) 23	-.2052 (.078) 0
				$\hat{\beta}_1$	1.0130 (.207)	1.0156 (.201)	1.0102 (.206)
				$\hat{\beta}_2$	1.0099 (.192)	1.0122 (.187)	1.0063 (.196)
50	.40	-.15	(-2.00, 1.60)	$\hat{\rho}_1$	-.1558 (.080) 0	—	—
50	.75	.40	(-.90, 2.00)	$\hat{\rho}_2$	.3928 (.090) 0	—	—
				$\hat{\beta}_1$	1.0164 (.170)	—	—
				$\hat{\beta}_2$	1.0154 (.180)	—	—

Table 6.1: Results for stationary AR(1) designs.

MA(1) Design				Estimates			
I_g	μ_g	ρ_g	X'_{g1}	QAQ	BAH	KAN	
50	.20	.22	(-2.00, .61)	$\hat{\rho}_1$	.2169 (.093) 10	.2160 (.089) 9	.2156 (.084) 3
50	.40	.10	(1.60, -2.01)	$\hat{\rho}_2$	.0850 (.085) 0	.0845 (.085) 0	.0917 (.081) 29
				$\hat{\beta}_1$	1.0166 (.139)	1.0173 (.139)	1.0118 (.147)
				$\hat{\beta}_2$	1.0153 (.145)	1.0160 (.145)	1.0105 (.156)
25	.50	.00	(1.00, -1.00)	$\hat{\rho}_1$	-.0078 (.115) 0	-.0078 (.115) 0	-.0064 (.113) 100
50	.10	.20	(-1.20, -1.00)	$\hat{\rho}_2$	.1846 (.116) 26	.1857 (.110) 11	.1951 (.110) 5
25	.30	.25	(.00, -.85)	$\hat{\rho}_3$	.2443 (.113) 11	.2443 (.108) 19	.2472 (.111) 38
				$\hat{\beta}_1$	1.0019 (.154)	1.0016 (.154)	1.0137 (.152)
				$\hat{\beta}_2$	1.0180 (.148)	1.0182 (.149)	1.0133 (.153)
50	.80	.25	(-.60, 1.99)	$\hat{\rho}_1$	.2406 (.093) 16	—	—
50	.40	-.30	(1.60, -2.01)	$\hat{\rho}_2$	-.3052 (.069) 19	—	—
				$\hat{\beta}_1$	1.0291 (.235)	—	—
				$\hat{\beta}_2$	1.0218 (.173)	—	—
50	.10	.37	(-2.00, -.20)	$\hat{\rho}_1$	—	—	-.0039 (.080) 0
50	.30	.20	(1.15, -2.00)	$\hat{\rho}_2$	—	—	-.0074 (.082) 0
				$\hat{\beta}_1$	—	—	1.0088 (.122)
				$\hat{\beta}_2$	—	—	1.0075 (.103)
50	.40	-.15	(-2.00, 1.60)	$\hat{\rho}_1$	—	-.1560 (.078) 20	—
50	.90	.28	(2.00, .20)	$\hat{\rho}_2$	—	.2625 (.106) 24	—
				$\hat{\beta}_1$	—	1.0164 (.125)	—
				$\hat{\beta}_2$	—	1.0155 (.172)	—
50	.10	.30	(-2.00, -.20)	$\hat{\rho}_1$	—	.2880 (.110) 35	.2942 (.104) 7
50	.30	.20	(1.15, -2.00)	$\hat{\rho}_2$	—	.1872 (.081) 3	.1921 (.078) 13
				$\hat{\beta}_1$	—	1.0178 (.149)	1.0104 (.138)
				$\hat{\beta}_2$	—	1.0143 (.118)	1.0095 (.111)

Table 6.2 Results for stationary MA(1) designs — continued on next page.

MA(1) Design				Estimates		
I_g	μ_g	ρ_g	X'_{g1}	QAQ	BAH	KAN
50	.80	.25	(-.60, 1.99)	$\hat{\rho}_1$.2406 (.093) 16	.2404 (.088) 14	—
50	.40	-.15	(1.60, -2.01)	$\hat{\rho}_2$ -.1612 (.079) 0	-.1607 (.078) 20	—
				$\hat{\beta}_1$ 1.0242 (.249)	1.0254 (.247)	—
				$\hat{\beta}_2$ 1.0203 (.176)	1.0213 (.175)	—

Table 6.2: Results for stationary MA(1) designs.

EX Design				Estimates		
I_g	μ_g	ρ_g	X'_{g1}	QAQ	BAH	KAN
50	.20	.30	(-2.00, .61)	$\hat{\rho}_1$.2858 (.119) 0	.2937 (.104) 6	.2855 (.103) 0
50	.40	.45	(1.60, -2.01)	$\hat{\rho}_2$.4432 (.092) 0	.4440 (.082) 11	.4454 (.089) 0
				$\hat{\beta}_1$ 1.0234 (.170)	1.0225 (.169)	1.0205 (.169)
				$\hat{\beta}_2$ 1.0235 (.194)	1.0209 (.191)	1.0211 (.199)
25	.40	.45	(-2.00, 1.60)	$\hat{\rho}_1$.4388 (.127) 0	.4390 (.122) 19	.4426 (.132) 0
50	.10	.20	(-.20, -2.00)	$\hat{\rho}_2$.1845 (.133) 4	.1873 (.120) 10	.1785 (.123) 5
25	.30	.50	(1.15, -2.00)	$\hat{\rho}_3$.4900 (.142) 0	.5000 (.119) 23	.5005 (.127) 0
				$\hat{\beta}_1$ 1.0273 (.188)	1.0283 (.192)	1.0212 (.184)
				$\hat{\beta}_2$ 1.0147 (.134)	1.0101 (.134)	1.0201 (.128)
50	.45	.80	(1.50, -1.70)	$\hat{\rho}_1$.7965 (.068) 0	— —	.7970 (.065) 0
50	.65	.20	(.20, .42)	$\hat{\rho}_2$.1905 (.098) 0	— —	.1953 (.089) 1
				$\hat{\beta}_1$.9933 (.344)	—	1.0008 (.340)
				$\hat{\beta}_2$ 1.0187 (.294)	—	1.0060 (.282)
50	.80	.30	(-.60, 1.99)	$\hat{\rho}_1$.2858 (.119) 0	.2937 (.104) 6	— —
50	.40	.50	(1.60, -2.01)	$\hat{\rho}_2$.4944 (.089) 0	.4952 (.078) 27	— —
				$\hat{\beta}_1$ 1.0226 (.348)	1.0239 (.350)	—
				$\hat{\beta}_2$ 1.0231 (.216)	1.0234 (.217)	—

Table 6.3: Results for stationary EX designs.

During the simulation study we noted any time a value of $\hat{\rho}_g$ or $\hat{\rho}_{pl}$ was outside its theoretical range. The percentages of such range restriction violations are reported in the tables in this chapter. We might be concerned when we estimate the correlation to be a value that is theoretically impossible to generate. It is perhaps an indication of something irregular within the generated data. Of particular importance would be the effect of range restriction violations on the estimation of β , as this parameter describes the relationship between the response and the covariates. However, it would appear that the percentage of range restriction violations seems to have little impact upon the estimation of this parameter vector.

In practice, a more common assumption is that of a constant correlation parameter in the assumed structure for all clusters. In this situation we use $\hat{\rho}_{pl}$ to obtain a single estimate of the correlation parameter for all clusters. Results for designs of this type are given in Tables 6.4 to 6.6. All three generating methods perform similarly in this type of stationary design.

6.2 Nonstationary Designs

Another important assessment from a practical standpoint is the comparison of the properties of the estimators for β and ρ_{pl} when the data generation methods are used to simulate nonstationary binary sequences. Results for these nonstationary designs for QAQ and BAH are similar to those obtained under stationary designs. However, the nonstationary extension to KAN provides very poor estimation of β . Also, the underestimation of ρ is not as prevalent in the nonstationary designs.

AR(1) Stationary Design				Estimates			
I_g	μ_g	ρ_g	X'_{g1}	QAQ	BAH	KAN	
25	.25	.30	(-.40, -.70)	$\hat{\rho}_{pl}$	.2869 (.128)	.2911 (.116)	.2860 (.127)
50	.35	.30	(.80 , -1.42)		0	2	1
25	.45	.30	(.20 , -.40)	$\hat{\beta}_1$	1.0396 (.387)	1.0384 (.390)	1.0448 (.381)
				$\hat{\beta}_2$	1.0205 (.216)	1.0193 (.215)	1.0196 (.212)

Table 6.4: Results for stationary AR(1) designs with constant ρ_g across groups.

MA(1) Stationary Design				Estimates			
I_g	μ_g	ρ_g	X'_{g1}	QAQ	BAH	KAN	
25	.10	.20	(.00, -2.20)	$\hat{\rho}_{pl}$	.1823 (.169)	.1859 (.158)	.1794 (.152)
50	.20	.20	(-.50 , -1.89)		34	27	18
25	.30	.20	(-.80 , -.05)	$\hat{\beta}_1$	1.0170 (.279)	1.0146 (.282)	1.0092 (.258)
				$\hat{\beta}_2$	1.0163 (.107)	1.0155 (.107)	1.0065 (.099)

Table 6.5: Results for stationary MA(1) designs with constant ρ_g across groups.

EX Stationary Design				Estimates			
I_g	μ_g	ρ_g	X'_{g1}	QAQ	BAH	KAN	
25	.80	.30	(1.10, .29)	$\hat{\rho}_{pl}$	.2790 (.164)	.2892 (.147)	.2892 (.148)
50	.70	.30	(.20 , .65)	1	15	2	
25	.90	.30	(-.20 , 2.40)	$\hat{\beta}_1$	1.0292 (.306)	1.0262 (.305)	1.0241 (.311)
				$\hat{\beta}_2$	1.0224 (.167)	1.0228 (.175)	1.0250 (.172)

Table 6.6: Results for stationary EX designs with constant ρ_g across groups.

AR(1) Nonstationary Design				Estimates		
I_g	μ_{g1}	ρ_g	X'_{g1}	QAQ	BAH	KAN
25	.35	.75	(-.87, .25)	$\hat{\rho}_{pl}$.7679 (.037)	—	—
50	.63	.75	(.28, .25)	0	—	—
25	.50	.75	(-.25, .25)	$\hat{\beta}_1$ 1.0578 (.378)	—	—
Transformation 1				$\hat{\beta}_2$ 1.0174 (.237)	—	—
25	.35	.35	(-.37, -.25)	$\hat{\rho}_{pl}$.3613 (.072)	.3457 (.061)	.3913 (.068)
50	.20	.35	(-1.14, -.25)	0	0	0
25	.50	.35	(.25, -.25)	$\hat{\beta}_1$ 1.0185 (.261)	1.0120 (.257)	1.0468 (.245)
Transformation 2				$\hat{\beta}_2$ 1.0335 (.261)	1.0267 (.259)	0.7194 (.263)

Table 6.7: Results for nonstationary AR(1) designs.

MA(1) Nonstationary Design				Estimates		
I_g	μ_{g1}	ρ_g	X'_{g1}	QAQ	BAH	KAN
25	.40	-.25	(-.66, .25)	$\hat{\rho}_{pl}$ -.2477 (.048)	—	—
50	.20	-.25	(-1.64, .25)	0	—	—
25	.30	-.25	(-1.10, .25)	$\hat{\beta}_1$ 1.0172 (.151)	—	—
Transformation 1				$\hat{\beta}_2$ 1.0359 (.282)	—	—
25	.65	.30	(.87, -.25)	$\hat{\rho}_{pl}$.2860 (.057)	.2950 (.054)	—
50	.75	.30	(1.35, -.25)	1	32	—
25	.85	.30	(1.99, -.25)	$\hat{\beta}_1$ 1.0191 (.191)	1.0228 (.192)	—
Transformation 2				$\hat{\beta}_2$ 1.0274 (.369)	1.0366 (.368)	—

Table 6.8: Results for nonstationary MA(1) designs.

EX Nonstationary Design				Estimates		
I_g	μ_{g1}	ρ_g	X'_{g1}	QAQ	BAH	KAN
25	.40	.35	(-.66, .25)	$\hat{\rho}_{pl}$.3479 (.071)	.3468 (.064)	—
50	.60	.35	(.16, .25)	0	26	—
25	.20	.35	(-1.64, .25)	$\hat{\beta}_1$ 1.0298 (.209)	1.0160 (.211)	—
Transformation 1				$\hat{\beta}_2$ 1.0086 (.223)	.9894 (.232)	—
25	.60	.34	(.66, -.25)	$\hat{\rho}_{pl}$.3543 (.071)	.3318 (.065)	—
50	.30	.34	(-.60, -.25)	0	19	—
25	.45	.34	(.05, -.25)	$\hat{\beta}_1$ 1.0502 (.300)	1.0487 (.299)	—
Transformation 2				$\hat{\beta}_2$ 1.0199 (.211)	1.0130 (.218)	—

Table 6.9: Results for nonstationary EX designs.

Chapter 7

Conclusion and Discussion

In the literature, several methods for generating longitudinally correlated binary data have been proposed. We discussed a selection of these approaches in Chapter 2, each of which imposes restraints on the allowable range for the unconditional means, correlation structure and parameters. In this study we focus on the three techniques introduced by Qaqish (2003) [15], Bahadur (1961) [1] and Kanter [10] referred to as QAQ, BAH and KAN respectively.

The QAQ method permits generation from any nonstationary structure, although Qaqish (2003) [15] only considered AR(1), MA(1) and EX. A general correlation structure can be considered, however the $b_{tt'}$ reflecting the structure must first be determined. The second method, BAH, allows for any nonstationary structure in a very straight forward manner. By contrast, KAN and its proposed extensions are only defined for stationary correlation structures in the AR and MA families.

This study has extended QAQ to higher order AR structures and KAN to nonstationary structures. However, the nonstationary extension to the KAN model, while theoretically possible, yields poor results when generating data. The three methods have been compared in terms of range restrictions for the correlation parameters associated with stationary AR(1), MA(1) and exchangeable structures at differing values

of μ . Generally speaking, Qaqish provides the greatest flexibility, in particular for AR(1) and exchangeable structures. In the MA(1) case, although QAQ again yields the widest allowable range for the correlation parameter for a majority of values for μ , there are benefits to the use of Kanter (1975) [10] and Bahadur (1961) [1] when μ is either very small (< 0.15) or very large (> 0.85) respectively. Note that this study has suggested an improvement to the approach of Kanter (1975) [10] that results in more favorable range restrictions. To assess the impact of the range restrictions, we carried out a simulation study.

We initially considered stationary designs, some of which had the correlation structure and parameter fixed across all clusters. Of interest were the properties of estimators of regression parameters based on a GQL approach. The proportion of range restriction violations was noted, and while there was a noticeable difference in the proportion of violations among the methods for some designs, there were no large differences in the estimation, particularly of β . Typically it is β that we wish to estimate, while the correlation parameter is a nuisance parameter. The most relevant concern arose when the design matrix of covariates was ill-conditioned, and the degree to which it was ill-conditioned. We next considered nonstationary designs, with similar results.

The research conducted in this study leads to additional further work that might be considered. For example, we examined model estimation with a GQL procedure; one could investigate estimation properties under an alternative approach. Link functions other than logit could be investigated. In addition, we found that the imposed range restrictions on the model primarily dictated the usefulness of that model. Recently, Farrell and Sutradhar (2006) [6] propose a generation model which allows full ranges of the correlation parameters. The approach is similar to Qaqish (2003) [15],

except that the conditional probability that $Y_t = 1$ is modeled as a logit link of previous responses instead of linearly. However, the construction of the model makes it difficult to specify a correlation structure. Finally, in this study, we altered KAN to KAN* and thus provided a wider applicable range for the mean and correlation parameter. Additional adjustments, to KAN or the other methods, could possibly allow for an improvement in range restriction breath. Another interesting consideration would be to incorporate a missing data mechanism into the data generation model.

List of References

- [1] R. Bahadur. A representation of the joint distribution of responses to n dichotomous items. In *Studies in Item Analysis and Prediction*. Stanford Mathematical Studies in the Social Sciences, 1961.
- [2] L. Bondesson. On a minimum correlation problem. *Statistics and Probability Letters*, 62:361–370, 200.
- [3] N.R. Chaganty and H. Joe. Efficiency of generalized estimating equations for binary responses. *J. R. Statist. Soc. B*, 66(4):851–860, 2004.
- [4] L. Devroy. *Non-Uniform Random Variate Generation*. New York: Springer-Verlag, 1986.
- [5] L.J. Emrich and M.R. Piedmonte. A method for generating high-dimensional multivariate binary variates. *The American Statistician*, 45(4):302–304, 1991.
- [6] P. Farrell and B. Sutradhar. A non-linear conditional probability model for generating correlated binary data. *Statistics and Probability Letters*, 76(4):353–361, 2006.
- [7] G. Fitzmaurice and N. Laird. A likelihood-based method for analysing longitudinal binary responses. *Biometrika*, 80(1):141–151, 1993.
- [8] S. Gange. Generating multivariate categorical variates using the iterative proportional fitting algorithm. *The American Statistician*, 45(2):134–138, 1995.
- [9] H. Joe. *Multivariate models and dependence concepts*. Chapman and Hall, 1997.
- [10] M. Kanter. Autoregression for discrete processes mod 2. *Journal of Applied Probability*, 12:371–375, 1975.
- [11] A. D. Lunn and S. J. Davies. A note on generating correlated binary variables. *Biometrika*, 85(2):487–490, 1998.

- [12] E. McKenzie. Extending the correlation structure of exponential autoregressive moving average processes. *Journal of Applied Probability*, 18:1–9, 1981.
- [13] S.D. Oman and D.M. Zucker. Modelling and generating correlated binary variables. *Biometrika*, 88(1):287–290, 2001.
- [14] C. G. Park, T. Park, and D. W. Shin. A simple method for generating correlated binary variates. *The American Statistician*, 50(4):306–310, 1996.
- [15] B. Qaqish. A family of multivariate binary distributions for simulating correlated binary variables with specified marginal means and correlations. *Biometrika*, 90(2):455–463, 2003.
- [16] J. Shults and N.R. Chaganty. The analysis of serially correlated data using quasi-least squares. *Biometrics*, 54:1622–1630, 1998.
- [17] B.C. Sutradhar. An overview on regression models for discrete longitudinal responses. *Statistical Science*, 18(3):377–393, 2003.
- [18] B.C. Sutradhar and K. Das. On the efficiency of regression estimators in generalised linear models for longitudinal data. *Biometrika*, 86:459–465, 1999.
- [19] B.C. Sutradhar and P. Kumar. The inversion of correlation matrix for ma(1) process. *Applied Mathematics Letters*, 16(3):317–321, 2003.
- [20] J.H. Ware, D.W. Dockery, A. III Spiro, F.E. Speizer, and B.F. Ferris Jr. Passive smoking, gas cooking and respiratory health in children living in six cities. *Am. Ren. Respir. Dis.*, 129:366–374, 1984.

Appendix A

Derivation of Qaqish (2003) Results

In this appendix we derive the $b_{it'}$ for Qaqish (2003) [15] for AR(1), MA(1) and exchangeable correlation structures.

Note that for the Qaqish (2003) data generation method, if we set $X_t = (Y_1, \dots, Y_{t-1})^T$ we have that

$$P(Y_t = 1|X_t) = E(Y_t|X_t) = E(Y_t|Y_1, \dots, Y_{t-1}) = \mu_t + \sum_{j=1}^{t-1} b_{tj}(y_j - \mu_j)$$

where

$$b_t = (b_{t1}, \dots, b_{t,t-1})' = G_t^{-1} S_t = [Cov(X_t)]^{-1} Cov(X_t, Y_t)$$

in which $Cov(X_t)$ is the covariance matrix of $(Y_1, \dots, Y_{t-1})$, and $Cov(X_t, Y_t)$ is a $t-1$ vector of the covariance of Y_t with

$$G_t^{-1} = \begin{pmatrix} V_{11}^{-1/2} & & 0 \\ & \ddots & \\ 0 & & V_{t-1,t-1}^{-1/2} \end{pmatrix} C_X^{-1}$$

where $v_{jj} = \mu_j(1 - \mu_j)$, $j = 1, \dots, t-1$ and C_X defines the correlation matrix of X_t , and

$$S_t = \begin{pmatrix} V_{11}^{-1/2} & & 0 \\ & \ddots & \\ 0 & & V_{t-1,t-1}^{-1/2} \end{pmatrix} Cov(X_t, Y_t).$$

For example, if an AR(1) correlation structure is considered, then

$$C_X = \begin{pmatrix} 1 & \rho & \dots & \rho^{t-2} \\ \rho & 1 & & \\ \vdots & & \ddots & \\ \rho^{t-2} & & & 1 \end{pmatrix}$$

so that

$$[C_X^{-1}]_{jk} = \begin{cases} \frac{\rho}{\rho^2-1}, & |j-k|=1 \\ -\frac{\rho^2+1}{\rho^2-1}, & j=k=2, \dots, t-2 \\ -\frac{1}{\rho^2-1}, & j=k=2 \text{ or } j=k=t-1 \\ 0, & \text{otherwise} \end{cases}.$$

In addition, since

$$Cov(X_t, Y_t) = \begin{pmatrix} \rho^{t-1} \sqrt{v_{11} v_{tt}} \\ \vdots \\ \rho \sqrt{v_{t-1,t-1} v_{tt}} \end{pmatrix}$$

we have

$$[G_t^{-1}]_{jk} = \begin{cases} \frac{\rho}{\sqrt{v_{jj}(\rho^2-1)}}, & |j-k|=1 \\ -\frac{\rho^2+1}{\sqrt{v_{jj}(\rho^2-1)}}, & j=k=2, \dots, t-2 \\ -\frac{1}{\sqrt{v_{jj}(\rho^2-1)}}, & j=k=2 \text{ or } j=k=t-1 \\ 0, & \text{otherwise} \end{cases}$$

and

$$[S_t]_j = \rho^{t-j} \sqrt{v_{tt}}, \quad j = 1, \dots, t-1$$

so that

$$\begin{aligned}
b_{tj} &= [G_t^{-1} S_t]_j = \sum_{k=1}^{t-1} [G_t^{-1}]_{jk} [S_t]_k \\
b_{t1} &= \frac{\sqrt{v_{tt}}}{\sqrt{v_{11}(\rho^2-1)}} \{-\rho^{t-1} + \rho \cdot \rho^{t-2}\} = 0 \\
b_{t2} &= \frac{\sqrt{v_{tt}}}{\sqrt{v_{jj}(\rho^2-1)}} \{\rho \cdot \rho^{t-(j-1)} - (\rho^2 + 1) \cdot \rho^{t-j} + \rho \cdot \rho^{t-(j+1)}\} \\
&= \frac{\sqrt{v_{tt}}}{\sqrt{v_{jj}(\rho^2-1)}} \{\rho^{t-j+2} - \rho^{t-j+2} - \rho^{t-j} + \rho^{t-j}\} \\
&= 0 \\
b_{t,t-1} &= \frac{\sqrt{v_{tt}}}{\sqrt{v_{t-1,t-1}(\rho^2-1)}} \{\rho \cdot \rho^2 - \rho\} = \rho \left(\frac{v_{tt}}{v_{t-1,t-1}} \right)^{\frac{1}{2}}.
\end{aligned}$$

Thus,

$$b_{tj} = \begin{cases} 0, & j = 1, \dots, t-2 \\ \rho \left(\frac{v_{tt}}{v_{t-1,t-1}} \right)^{\frac{1}{2}}, & j = t-1 \end{cases}.$$

Similarly for an MA(1) process,

$$C_X = \begin{pmatrix} 1 & \rho & 0 & \dots \\ \rho & 1 & & \\ 0 & & \ddots & \\ \vdots & & & 1 \end{pmatrix}$$

Sutradhar and Kumar (2003) [19] give the form of C_X^{-1} as

$$\begin{aligned}
[C_X^{-1}]_{jk} &= \frac{1+a^2}{1-a^2} \left[(a^{|j-k|} - a^{2(t+1)-j-k-2}) \right. \\
&\quad \left. - \frac{a^{j+k}}{1-a^{2(t+1)-2}} (1 - a^{2(t+1)-2j-2}) (1 - a^{2(t+1)-2k-2}) \right]
\end{aligned}$$

where, since $\rho = -\frac{a}{1+a^2}$, generally speaking $a = \frac{-1 \pm \sqrt{1-4\rho^2}}{2\rho}$. However, since

$-1 < \rho < 1$, then $-1 < a < 1$, which implies $a = \frac{\sqrt{1-4\rho^2}-1}{2\rho}$. In addition, since

$$\text{Cov}(X_t, Y_t) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \rho\sqrt{v_{t-1,t-1}v_{tt}} \end{pmatrix}$$

we have

$$\begin{aligned} [G_t^{-1}]_{jk} &= \frac{1+a^2}{\sqrt{v_{jj}(1-a^2)}} \left[(a^{|j-k|} - a^{2(t+1)-j-k-2}) \right. \\ &\quad \left. - \frac{a^{j+k}}{1-a^{2(t+1)-2}} (1 - a^{2(t+1)-2j-2}) (1 - a^{2(t+1)-2k-2}) \right] \end{aligned}$$

and

$$[S_t]_j = \begin{cases} 0, & j = 1, \dots, t-2 \\ \rho\sqrt{v_{tt}}, & j = t-1 \end{cases}$$

so that

$$\begin{aligned} b_{tj} &= [G_t^{-1}S_t]_j \\ &= \sum_{k=1}^{t-1} [G_t^{-1}]_{jk} [S_t]_k \\ &= \rho\sqrt{v_{tt}} [G_t^{-1}]_{j,t-1} \\ &= \rho\sqrt{v_{tt}} \frac{1+a^2}{\sqrt{v_{jj}(1-a^2)}} \left[\{a^{|j-t+1|} - a^{2(t+1)-j-(t-1)-2}\} - \right. \\ &\quad \left. \frac{a^{j+t-1}}{1-a^{2(t+1)-2}} \{ (1 - a^{2(t+1)-2j-2}) (1 - a^{2(t+1)-2(t-1)-2}) \} \right] \\ &= \rho \left(\frac{v_{tt}}{v_{jj}} \right)^{\frac{1}{2}} \frac{1+a^2}{(1-a^2)} \left[\{a^{-j+t-1} - a^{t-j+1}\} - \frac{a^{j+t-1}}{1-a^{2t}} \{ (1 - a^{2(t-j)}) (1 - a^2) \} \right] \\ &= \rho \left(\frac{v_{tt}}{v_{jj}} \right)^{\frac{1}{2}} \frac{1+a^2}{(1-a^2)} \left[a^{-j+t-1} (1-a^2) - \frac{a^{j-1}}{a^{-t}-a^t} \{ (1 - a^{2(t-j)}) (1 - a^2) \} \right] \end{aligned}$$

$$\begin{aligned}
b_{tj} &= \rho \left(\frac{v_{tt}}{v_{jj}} \right)^{\frac{1}{2}} (1 + a^2) \left[\frac{a^{-j+t-1} (a^{-t} - a^t)}{a^{-t} - a^t} - \frac{a^{j-1} (1 - a^{2(t-j)})}{a^{-t} - a^t} \right] \\
&= \rho \left(\frac{v_{tt}}{v_{jj}} \right)^{\frac{1}{2}} (1 + a^2) \left[\frac{a^{-j-1} - a^{2t-j-1} - a^{j-1} + a^{2t-j-1}}{a^{-t} - a^t} \right] \\
&= \rho \left(\frac{v_{tt}}{v_{jj}} \right)^{\frac{1}{2}} \left(\frac{1 + a^2}{a} \right) \left[\frac{a^{-j} - a^j}{a^{-t} - a^t} \right] \\
&= \left(\frac{v_{tt}}{v_{jj}} \right)^{\frac{1}{2}} \left[\frac{a^j - a^{-j}}{a^{-t} - a^t} \right].
\end{aligned}$$

Finally, for an exchangeable process,

$$C_X = \begin{pmatrix} 1 & & \rho \\ & \ddots & \\ \rho & & 1 \end{pmatrix}$$

so that

$$[C_X^{-1}]_{jk} = \begin{cases} \frac{\rho}{[(t-2)\rho+1][\rho-1]}, & j \neq k \\ -\frac{(t-3)\rho+1}{[(t-2)\rho+1][\rho-1]}, & j = k \end{cases}.$$

In addition, since

$$Cov(X_t, Y_t) = \begin{pmatrix} \rho \sqrt{v_{11} v_{tt}} \\ \vdots \\ \rho \sqrt{v_{t-1, t-1} v_{tt}} \end{pmatrix}$$

we have

$$[G_t^{-1}]_{jk} = \begin{cases} \frac{\rho}{\sqrt{v_{jj}}[(t-2)\rho+1][\rho-1]}, & j \neq k \\ -\frac{(t-3)\rho+1}{\sqrt{v_{jj}}[(t-2)\rho+1][\rho-1]}, & j = k \end{cases}$$

and

$$[S_t]_j = \rho\sqrt{v_{tt}}, \quad j = 1, \dots, t-1$$

so that

$$\begin{aligned}
 b_{tj} &= [G_t^{-1}S_t]_j \\
 &= \sum_{k=1}^{t-1} [G_t^{-1}]_{jk} [S_t]_k \\
 &= \rho\sqrt{v_{tt}} \left\{ \frac{1}{\sqrt{v_{jj}}[(t-2)\rho+1][\rho-1]} [\rho(t-2) - (t-3)\rho - 1] \right\} \\
 &= \frac{\rho\sqrt{v_{tt}}}{\sqrt{v_{jj}}[(t-2)\rho+1]} \\
 &= \frac{\rho}{1+(t-2)\rho} \left(\frac{v_{tt}}{v_{jj}} \right)^{\frac{1}{2}}, \quad j = 1, \dots, t-1.
 \end{aligned}$$

Appendix B

Derivation of Kanter (1975) Results and Extensions

B.1 Derivation of Kanter (1975) Results for a stationary AR(1) process

Setting $E[Y_1] = \mu$, Kanter (1975) 3.3 suggests generating Y_t according to

$$Y_t = U_t (Y_{t-1} \oplus W_t) + (1 - U_t)W_t, \quad t \geq 2.$$

Since

$$\begin{aligned} \mu &= E[Y_t] \\ &= E[U_t (Y_{t-1} \oplus W_t) + (1 - U_t)W_t] \\ &= u [\mu(1 - w) + (1 - \mu)w] + (1 - u)w \\ &= -2u\mu w + u\mu + w \\ &= w(1 - 2u\mu) + u\mu \end{aligned}$$

we have that $P(W = 1) = w = \frac{\mu(1-u)}{1-2u\mu}$ and $P(U = 1) = u = \frac{\mu(\mu-w)}{\mu(1-2w)}$.

To demonstrate that the above expression generates Y_t according to a stationary

AR(1) process, first note that

$$\begin{aligned}
& E[Y_{t+1}Y_t], \quad t \geq 1 \\
&= E\left\{ \left[U_{t+1}(Y_t \oplus W_{t+1}) + (1 - U_{t+1})W_{t+1} \right] Y_t \right\} \\
&= \mu E\left\{ U_{t+1}(1 - W_{t+1}) + (1 - U_{t+1})W_{t+1} \right\} \\
&= \mu \left[u(1 - w) + (1 - u)w \right] \\
&= \mu \left[u + w(1 - 2u) \right] \\
&= \mu \left[u + \frac{\mu(1 - u)}{1 - 2u\mu}(1 - 2u) \right] \\
&= \mu \left[\frac{u - 2u^2\mu + \mu - 3u\mu + 2u^2\mu}{1 - 2u\mu} \right] \\
&= \frac{u\mu(1 - 3\mu + 2\mu^2) + \mu^2(1 - 2u\mu)}{1 - 2u\mu} \\
&= \frac{u\mu(1 - \mu)(1 - 2\mu)}{1 - 2u\mu} + \mu^2.
\end{aligned}$$

Similary,

$$\begin{aligned}
& E[Y_{t+j}Y_t], \quad j \geq 2 \\
&= E\left\{ \left[U_{t+j}(Y_{t+j-1} \oplus W_{t+j}) + (1 - U_{t+j})W_{t+j} \right] Y_t \right\} \\
&= uE\left\{ Y_t(Y_{t+j-1} \oplus W_{t+j}) \right\} + (1 - u)w\mu \\
&= u(1 - w)E\{Y_tY_{t+j-1}\} + uwE\left\{ (1 - Y_{t+j-1})Y_t \right\} + (1 - u)w\mu \\
&= E\{Y_{t+j-1}Y_t\} \left[u(1 - w) - uw \right] + uw\mu + w\mu - uw\mu \\
&= E\{Y_{t+j-1}Y_t\} u(1 - 2w) + w\mu \\
&= E\{Y_{t+j-1}Y_t\} u \left[1 - \frac{2\mu(1 - u)}{1 - 2u\mu} \right] + \frac{\mu^2(1 - u)}{1 - 2u\mu} \\
&= E\{Y_{t+j-1}Y_t\} u \left[\frac{1 - 2u\mu}{1 - 2u\mu} - \frac{2\mu - 2\mu u}{1 - 2u\mu} \right] + \mu^2 \left[\frac{1 - 2u\mu}{1 - 2u\mu} - \frac{u - 2\mu u}{1 - 2u\mu} \right] \\
&= E\{Y_{t+j-1}Y_t\} \left[\frac{u(1 - 2\mu)}{1 - 2u\mu} \right] + \mu^2 \left[1 - \frac{u(1 - 2\mu)}{1 - 2u\mu} \right] \\
&= E\{Y_{t+j-1}Y_t\} \rho + \mu^2(1 - \rho)
\end{aligned}$$

Equivalently,

$$E\{Y_{t+j-1}Y_t\} = \frac{E\{Y_{t+j}Y_t\} + \mu^2(1 - \rho)}{\rho}$$

Setting $\text{Corr}(Y_{t+j}, Y_t) = \rho^j$ since we consider an AR(1) process we have,

$$\begin{aligned} \text{Corr}(Y_{t+j}, Y_t) &= \frac{E[Y_{t+j}Y_t] - E[Y_{t+j}]E[Y_t]}{\sqrt{E\left\{\left(Y_{t+j} - E[Y_{t+j}]\right)^2\right\}E\left\{\left(Y_t - E[Y_t]\right)^2\right\}}} \\ &= \frac{E[Y_{t+j}Y_t] - \mu^2}{\mu(1 - \mu)} \\ &= \rho^j \end{aligned}$$

so that

$$E[Y_{t+j}Y_t] = \rho^j\mu(1 - \mu) + \mu^2. \quad (\text{B.1})$$

By definition B.1 holds for $j = 1$. Assume it is true for $j = k - 1$, then

$$\begin{aligned} E\{Y_{t+k-1}Y_t\} &= \rho^{k-1}\mu(1 - \mu) + \mu^2 \\ \Rightarrow \frac{E\{Y_{t+k}Y_t\} + \mu^2(1 - \rho)}{\rho} &= \rho^{k-1}\mu(1 - \mu) + \mu^2 \\ \Rightarrow E\{Y_{t+k}Y_t\} + \mu^2 - \rho\mu^2 &= \rho^k\mu(1 - \mu) + \rho\mu^2 \\ \Rightarrow E[Y_{t+k}Y_t] &= \rho^k\mu(1 - \mu) + \mu^2. \end{aligned}$$

Thus an AR(1) structure is generated. Note that the correlation parameter for this

structure can be expressed as follows

$$\begin{aligned}
 \rho &= \text{Corr}(Y_{t+1}, Y_t) \\
 &= \frac{E[Y_{t+1}Y_t] - E[Y_{t+1}]E[Y_t]}{\sqrt{E\left\{\left(Y_{t+1} - E[Y_{t+1}]\right)^2\right\}E\left\{\left(Y_t - E[Y_t]\right)^2\right\}}} \\
 &= \left[\frac{u\mu(1-\mu)(1-2\mu)}{1-2u\mu} + \mu^2 - \mu^2 \right] / \left[\mu(1-\mu) \right] \\
 &= \frac{u(1-2\mu)}{1-2u\mu}.
 \end{aligned}$$

This implies that $u = \frac{\mu(\mu-w)}{\mu(1-2w)} = \frac{\rho}{1-2\mu(1-\rho)}$ and $w = \frac{\mu(1-u)}{1-2u\mu} = \mu(1-\rho)$.

B.2 MA(1)

McKenzie (1981) [12] proposes to use

$$Y_t = U_t (W_{t-1} \oplus W_t) + (1 - U_t)W_t, \quad t \geq 1$$

for generating stationary MA(1) binary responses. Recall that W_0 is defined as a Bernoulli random variable with parameter w . Here

$$\begin{aligned}
 \mu &= E[Y_t] \\
 &= E\left[U_t (W_{t-1} \oplus W_t) + (1 - U_t)W_t\right] \\
 &= u\left[0 + w(1-w) + (1-w)w + 0\right] + (1-u)w \\
 &= -2uw^2 + w(1+u) \\
 &= w(u+1-2uw).
 \end{aligned}$$

To verify that the above model generates a stationary MA(1) binary sequence, Y_t , we note that

$$\begin{aligned}
& E[Y_{t+1}Y_t], \quad t \geq 1 \\
&= E\left\{\left[U_{t+1}(W_t \oplus W_{t+1}) + (1 - U_{t+1})W_{t+1}\right]\left[U_t(W_{t-1} \oplus W_t) + (1 - U_t)W_t\right]\right\} \\
&= E\left\{U_{t+1}(W_t \oplus W_{t+1})U_t(W_{t-1} \oplus W_t) + U_{t+1}(W_t \oplus W_{t+1})(1 - U_t)W_t \right. \\
&\quad \left. + (1 - U_{t+1})W_{t+1}U_t(W_{t-1} \oplus W_t) + (1 - U_{t+1})W_{t+1}(1 - U_t)W_t\right\} \\
&= E\left\{U^2(W_t \oplus W_{t+1})(W_{t-1} \oplus W_t) + U(1 - U)(W_t \oplus W_{t+1})W_t \right. \\
&\quad \left. + U(1 - U)W_{t+1}(W_{t-1} \oplus W_t) + (1 - U)^2W_{t+1}W_t\right\} \\
&= u^2\left[w^2(1 - w) + w(1 - w)^2\right] + u(1 - u)w(1 - w) \\
&\quad + 2u(1 - u)w^2(1 - w) + (1 - u)^2w^2 \\
&= u^2w^2 - u^2w^3 + u^2w - 2u^2w^2 + u^2w^3 + uw - u^2w - uw^2 + u^2w^2 \\
&\quad + 2uw^2 - 2u^2w^2 - 2uw^3 + 2u^2w^3 + w^2 - 2uw^2 + u^2w^2 \\
&= w(-u^2w + u - uw - 2uw^2 + 2u^2w^2 + w) \\
&= uw(2uw^2 - uw - 2w^2 + w - 2w + 1) + w^2 \\
&= uw(2w - 1)\left[\frac{w - \mu}{2w - 1} - w - 1\right] + w^2 \\
&= \frac{(w - \mu)^2 + (2w - 1)(\mu w + \mu - w)}{(2w - 1)},
\end{aligned}$$

so that

$$\begin{aligned}
Corr(Y_{t+1}, Y_t) &= \frac{E[Y_{t+1}Y_t] - E[Y_{t+1}]E[Y_t]}{\sqrt{E\left\{\left(Y_{t+1} - E[Y_{t+1}]\right)^2\right\}E\left\{\left(Y_t - E[Y_t]\right)^2\right\}}} \\
&= \left[\frac{(w - \mu)^2 + (2w - 1)(\mu w + \mu - w)}{(2w - 1)} - \mu^2 \right] / \left[\mu(1 - \mu) \right] \\
&= \frac{(w - \mu)^2 + (2w - 1)(\mu w + \mu - w - \mu^2)}{(2w - 1)\mu(1 - \mu)} \\
&= \frac{(w - \mu) \left[w - \mu + (2w - 1)(\mu - 1) \right]}{(2w - 1)\mu(1 - \mu)} \\
&= \frac{(w - \mu)(-w - 2\mu + 2w\mu + 1)}{(2w - 1)\mu(1 - \mu)} \\
&= \frac{(w - \mu)(1 - 2\mu)(1 - w)}{(2w - 1)\mu(1 - \mu)} \\
&= \rho.
\end{aligned}$$

This gives a quadratic in terms of u , and subsequently we have $w = (u + 1 - \sqrt{(u + 1)^2 - 8u\mu})/(4u)$.

In addition,

$$\begin{aligned}
& E[Y_{t+j}Y_t], \quad j \geq 1 \\
&= E\left\{\left[U_{t+j}(W_{t+j-1} \oplus W_{t+j}) + (1 - U_{t+j})W_{t+j}\right] \cdot \right. \\
&\quad \left. \left[U_t(W_{t-1} \oplus W_t) + (1 - U_t)W_t\right]\right\} \\
&= E\left\{U_{t+j}(W_{t+j-1} \oplus W_{t+j})U_t(W_{t-1} \oplus W_t) + U_{t+j}(W_{t+j-1} \oplus W_{t+j})(1 - U_t)W_t \right. \\
&\quad \left. + (1 - U_{t+j})W_{t+j}U_t(W_{t-1} \oplus W_t) + (1 - U_{t+j})W_{t+j}(1 - U_t)W_t\right\} \\
&= E\left\{U^2(W_{t+j-1} \oplus W_{t+j})(W_{t-1} \oplus W_t) + U(1 - U)(W_{t+j-1} \oplus W_{t+j})W_t \right. \\
&\quad \left. + U(1 - U)W_{t+j}(W_{t-1} \oplus W_t) + (1 - U)^2W_{t+j}W_t\right\} \\
&= 4u^2w^2(1 - w)^2 + 4u(1 - u)w^2(1 - w) + (1 - u)^2w^2 \\
&= w^2\left[4u^2(1 - w)^2 + 4u(1 - u)(1 - w) + (1 - u)^2\right] \\
&= w^2\left[2u(1 - w) + 1 - u\right]^2 \\
&= w^2\left[u + 1 - 2uw\right]^2 \\
&= \mu^2,
\end{aligned}$$

so that

$$\begin{aligned}
& Corr(Y_{t+j}, Y_t), \quad j \geq 1 \\
&= \frac{E[Y_{t+j}Y_t] - E[Y_{t+j}]E[Y_t]}{\sqrt{E\left\{\left(Y_{t+j} - E[Y_{t+j}]\right)^2\right\}E\left\{\left(Y_t - E[Y_t]\right)^2\right\}}} \\
&= \frac{\mu^2 - \mu^2}{\mu(1 - \mu)} \\
&= 0.
\end{aligned}$$

Hence the model proposed by McKenzie (1981) [12] generates an MA(1) correlation structure.

B.3 EX

With regards to a stationary exchangeable correlation structure, we propose to generate binary data according to

$$Y_t = U_t (Y_0 \oplus W_t) + (1 - U_t)W_t, \quad t \geq 1$$

where $E[Y_0] = \mu$.

Therefore

$$\begin{aligned} \mu &= E[Y_t] \\ &= E[U_t (Y_0 \oplus W_t) + (1 - U_t)W_t] \\ &= u[0 + \mu(1 - w) + (1 - \mu)w + 0] + (1 - u)w \\ &= u\mu - u\mu w + uw - u\mu w + w - uw \\ &= w(1 - 2u\mu) + u\mu \end{aligned}$$

so that $P(W_t = 1) = w = \frac{\mu(1-u)}{1-2u\mu}$.

To establish that the model proposed does indeed generate Y_t according to an EX correlation structure, we note that

$$\begin{aligned}
& E[Y_t Y_{t'}], \quad t \neq t' \\
&= E \left\{ \left[U_t (Y_0 \oplus W_t) + (1 - U_t) W_t \right] \left[U_{t'} (Y_0 \oplus W_{t'}) + (1 - U_{t'}) W_{t'} \right] \right\} \\
&= E \left\{ U_t (Y_0 \oplus W_t) U_{t'} (Y_0 \oplus W_{t'}) + U_t (Y_0 \oplus W_t) (1 - U_{t'}) W_{t'} \right. \\
&\quad \left. + (1 - U_t) W_t U_{t'} (Y_0 \oplus W_{t'}) + (1 - U_t) W_t (1 - U_{t'}) W_{t'} \right\} \\
&= E \left\{ U (Y_0 \oplus W_t) U (Y_0 \oplus W_{t'}) + U (Y_0 \oplus W_t) (1 - U) W_{t'} \right. \\
&\quad \left. + (1 - U) W_t U (Y_0 \oplus W_{t'}) + (1 - U) W_t (1 - U) W_{t'} \right\} \\
&= E \left\{ U^2 \left[\mu(1 - W)^2 + (1 - \mu) W^2 \right] \right. \\
&\quad \left. + 2U(1 - U)W \left[\mu(1 - W) + (1 - \mu)W \right] + (1 - U)^2 W^2 \right\} \\
&= E \left\{ U^2 \left[\mu - 2\mu w + W^2 \right] + 2U(1 - U)W \left[\mu - 2\mu w + W \right] + (1 - U)^2 W^2 \right\} \\
&= E \left\{ U^2 \mu - 2U^2 \mu w + U^2 W^2 \right. \\
&\quad \left. + 2U \mu w - 2U^2 \mu w - 4U \mu w^2 + 4U^2 \mu w^2 + 2U W^2 - 2U^2 W^2 \right. \\
&\quad \left. + W^2 - 2U W^2 + U^2 W^2 \right\} \\
&= E \left\{ U^2 \mu - 4U^2 \mu w + 2U \mu w - 4U \mu w^2 + 4U^2 \mu w^2 + W^2 \right\} \\
&= u^2 \mu - 4u^2 \mu w + 2u \mu w - 4u \mu w^2 + 4u^2 \mu w^2 + w^2 \\
&= u^2 \mu (1 - 4w + 4w^2) + 2u \mu w (1 - 2w) + w^2 \\
&= \left[\frac{w - \mu}{\mu(2w - 1)} \right]^2 \mu (1 - 2w)^2 - 2 \left[\frac{w - \mu}{\mu(2w - 1)} \right] \mu w (2w - 1) + w^2 \\
&= \frac{(w - \mu)^2}{\mu} - 2w(w - \mu) + w^2 \\
&= \frac{(w - \mu)^2}{\mu} + 2w\mu - w^2,
\end{aligned}$$

so that

$$\begin{aligned}
& Corr(Y_t, Y_{t'}) \\
&= \frac{E[Y_t Y_{t'}] - E[Y_t]E[Y_{t'}]}{\sqrt{E[(Y_t - E[Y_t])^2]E[(Y_{t'} - E[Y_{t'}])^2]}} \\
&= \frac{\frac{(w-\mu)^2}{\mu} + 2w\mu - w^2 - \mu^2}{\mu(1-\mu)} \\
&= \frac{(w-\mu)^2}{\mu^2(1-\mu)} - \frac{(w-\mu)^2}{\mu(1-\mu)} \\
&= \frac{(w-\mu)^2}{\mu^2(1-\mu)}[1-\mu] \\
&= \left[\frac{w-\mu}{\mu} \right]^2 \\
&= \rho,
\end{aligned}$$

which suggests a constant correlation at all lags.

Thus,

$$\begin{cases} w = \mu(1 - \sqrt{\rho}), & u = \frac{\sqrt{\rho}}{1 - 2\mu(1 - \sqrt{\rho})}, & \mu \leq 1/2 \\ w = \mu(1 + \sqrt{\rho}), & u = \frac{\sqrt{\rho}}{2\mu(1 + \sqrt{\rho}) - 1}, & \mu > 1/2 \end{cases}$$

B.4 Derivation of KAN* for AR(1), MA(1) and Exchangeable structures

In Section 4.3.1 we propose an extension to KAN, KAN*, which provides a widened range restrictions. This extension is based upon two simple facts; $E(1 - Y_t) = 1 - \mu_t$ and $Corr(1 - Y_t, 1 - Y_{t'}) = Corr(Y_t, Y_{t'})$. Bellow, the models and corresponding parameters are stated for the three correlation structures discussed.

To generate an AR(1) correlation structure we define Y_1 to be a Bernoulli random variable with $E(Y_1) = \mu$, and have

$$Y_t = U_t (Y_{t-1} \oplus W_t) + (1 - U_t)(1 - W_t), \quad t \geq 2,$$

where $u = \frac{\rho}{1-2(1-\mu)(1-\rho)}$ and $w = (1 - \mu)(1 - \rho)$.

For an MA(1) process we use the model

$$Y_t = 1 - U_t (W_{t-1} \oplus W_t) - (1 - U_t)W_t, \quad t \geq 1,$$

with $\rho = \frac{(w+\mu-1)(2\mu-1)(1-w)}{(2w-1)\mu(1-\mu)}$ leading to the values of u and w .

An EX binary sequence can be generated with

$$Y_t = U_t (Y_0 \oplus W_t) + (1 - U_t)(1 - W_t), \quad t \geq 1,$$

where

$$\begin{cases} w = (1 - \mu)(1 - \sqrt{\rho}), & u = \frac{\sqrt{\rho}}{1-2(1-\mu)(1-\sqrt{\rho})}, & 1/2 \leq \mu \\ w = (1 - \mu)(1 + \sqrt{\rho}), & u = \frac{\sqrt{\rho}}{2(1-\mu)(1+\sqrt{\rho})-1}, & 1/2 > \mu \end{cases}.$$

B.5 AR(2)

Y_1 and Y_2 are generated as in a stationary AR(1). The following model is used to generate subsequent terms,

$$Y_t = U_t^{(1)} (y_{t-1} \oplus W_t) + U_t^{(2)} (y_{t-2} \oplus W_t) + (1 - U_t^{(1)} - U_t^{(2)})W_t, \quad t \geq 3.$$

We can see that this model generates an AR(2) process by noting that

$$\begin{aligned}
\mu &= E[Y_t] \\
&= E\left[U_t^{(1)} (y_{t-1} \oplus W_t) + U_t^{(2)} (y_{t-2} \oplus W_t) + (1 - U_t^{(1)} - U_t^{(2)})W_t\right] \\
&= u^{(1)}\left[\mu(1 - w) + (1 - \mu)w\right] + u^{(2)}\left[\mu(1 - w) + (1 - \mu)w\right] + (1 - u^{(1)} - u^{(2)})w \\
&= w\left[1 - 2\mu(u^{(1)} + u^{(2)})\right] + \mu(u^{(1)} + u^{(2)}),
\end{aligned}$$

so that,

$$\begin{aligned}
&E[Y_{t+1}Y_t], \quad t \geq 1 \\
&= E\left\{\left[U_{t+1}^{(1)} (y_t \oplus W_{t+1}) + U_{t+1}^{(2)} (y_{t-1} \oplus W_{t+1}) + (1 - U_{t+1}^{(1)} - U_{t+1}^{(2)})W_{t+1}\right]Y_t\right\} \\
&= \mu E\left\{U_{t+1}^{(1)} (1 - W_{t+1}) + U_{t+1}^{(2)} (y_{t-1} \oplus W_{t+1}) + (1 - U_{t+1}^{(1)} - U_{t+1}^{(2)})W_{t+1}\right\} \\
&= \mu\left\{u^{(1)} (1 - w) + u^{(2)}\left[\mu(1 - w) + (1 - \mu)w\right] + (1 - u^{(1)} - u^{(2)})w\right\} \\
&= \mu\left\{w\left[1 - 2u^{(1)} - 2u^{(2)}\mu\right] + u^{(1)} + u^{(2)}\mu\right\} \\
&= \mu\left\{\frac{\mu(1 - u^{(1)} - u^{(2)})}{1 - 2\mu(u^{(1)} + u^{(2)})}\left[1 - 2u^{(1)} - 2u^{(2)}\mu\right] + \frac{u^{(1)} + u^{(2)}\mu}{1 - 2\mu(u^{(1)} + u^{(2)})}\right\} \\
&= \mu\left\{\frac{\mu - 3\mu u^{(1)} - 2\mu^2 u^{(2)} + u^{(1)}}{1 - 2\mu(u^{(1)} + u^{(2)})}\right\} \\
&= \frac{\mu u^{(1)}(1 - \mu)(1 - 2\mu)}{1 - 2\mu(u^{(1)} + u^{(2)})} + \mu^2
\end{aligned}$$

and,

$$\begin{aligned}
& E[Y_{t+j}Y_t], \quad j \geq 2 \\
&= E\left\{\left[U_{t+j}^{(1)}(Y_{t+j-1} \oplus W_{t+j}) + U_{t+j}^{(2)}(Y_{t+j-2} \oplus W_{t+j}) + (1 - U_{t+j}^{(1)} - U_{t+j}^{(2)})W_{t+j}\right]Y_t\right\} \\
&= u^{(1)}E\left\{Y_t(Y_{t+j-1} \oplus W_{t+j})\right\} + u^{(2)}E\left\{Y_t(Y_{t+j-2} \oplus W_{t+j})\right\} + (1 - u^{(1)} - u^{(2)})w\mu \\
&= u^{(1)}(1 - w)E\{Y_tY_{t+j-1}\} + u^{(1)}wE\left\{Y_t(1 - Y_{t+j-1})\right\} + u^{(2)}(1 - w)E\{Y_tY_{t+j-2}\} \\
&\quad + u^{(2)}wE\left\{Y_t(1 - Y_{t+j-2})\right\} + (1 - u^{(1)} - u^{(2)})w\mu \\
&= E\{Y_tY_{t+j-1}\}\left[u^{(1)}(1 - w) - u^{(1)}w\right] + E\{Y_tY_{t+j-2}\}\left[u^{(2)}(1 - w) - u^{(2)}w\right] + w\mu \\
&= E\{Y_tY_{t+j-1}\}u^{(1)}(1 - 2w) + E\{Y_tY_{t+j-2}\}u^{(2)}(1 - 2w) + w\mu \\
&= (1 - 2w)\left[E\{Y_tY_{t+j-1}\}u^{(1)} + E\{Y_tY_{t+j-2}\}u^{(2)}\right] + w\mu \\
&= \left[1 - 2\left(\frac{\mu(1 - u^{(1)} - u^{(2)})}{1 - 2\mu(u^{(1)} + u^{(2)})}\right)\right]\left[E\{Y_tY_{t+j-1}\}u^{(1)} + E\{Y_tY_{t+j-2}\}u^{(2)}\right] \\
&\quad + \left(\frac{\mu(1 - u^{(1)} - u^{(2)})}{1 - 2\mu(u^{(1)} + u^{(2)})}\right)\mu \\
&= \frac{E\{Y_tY_{t+j-1}\}u^{(1)}(1 - 2\mu)}{1 - 2\mu(u^{(1)} + u^{(2)})} + \frac{E\{Y_tY_{t+j-2}\}u^{(2)}(1 - 2\mu)}{1 - 2\mu(u^{(1)} + u^{(2)})} \\
&\quad + \mu^2\left[\frac{1 - 2\mu(u^{(1)} + u^{(2)})}{1 - 2\mu(u^{(1)} + u^{(2)})} - \frac{u^{(1)}(1 - 2\mu)}{1 - 2\mu(u^{(1)} + u^{(2)})} - \frac{u^{(2)}(1 - 2\mu)}{1 - 2\mu(u^{(1)} + u^{(2)})}\right].
\end{aligned}$$

Now recall that in general, for an AR(2) process,

$$\text{Corr}(Y_{t+1}, Y_t) = \rho^{(1)}\text{Corr}(Y_t, Y_t),$$

and

$$\begin{aligned}
& \text{Corr}(Y_{t+j}, Y_t) = \rho^{(1)}\text{Corr}(Y_{t+j-1}, Y_t) + \rho^{(2)}\text{Corr}(Y_{t+j-2}, Y_t), \quad j \geq 2 \\
& \Rightarrow \frac{E[Y_{t+j}Y_t] - \mu^2}{\mu(1 - \mu)} = \rho^{(1)}\frac{E[Y_{t+j-1}Y_t] - \mu^2}{\mu(1 - \mu)} + \rho^{(2)}\frac{E[Y_{t+j-2}Y_t] - \mu^2}{\mu(1 - \mu)} \\
& \Rightarrow E[Y_{t+j}Y_t] = E[Y_{t+j-1}Y_t]\rho^{(1)} + E[Y_{t+j-2}Y_t]\rho^{(2)} + \mu^2(1 - \rho^{(1)} - \rho^{(2)}).
\end{aligned}$$

Thus, if we define $\rho^{(1)} = \frac{u^{(1)}(1 - 2\mu)}{1 - 2\mu(u^{(1)} + u^{(2)})}$ and $\rho^{(2)} = \frac{u^{(2)}(1 - 2\mu)}{1 - 2\mu(u^{(1)} + u^{(2)})}$, then

$$\begin{aligned}
 & E[Y_{t+j}Y_t], \quad j \geq 2 \\
 &= \frac{E\{Y_t Y_{t+j-1}\} u^{(1)} (1-2\mu)}{1-2\mu(u^{(1)}+u^{(2)})} + \frac{E\{Y_t Y_{t+j-2}\} u^{(2)} (1-2\mu)}{1-2\mu(u^{(1)}+u^{(2)})} \\
 &\quad + \mu^2 \left[\frac{1-2\mu(u^{(1)}+u^{(2)})}{1-2\mu(u^{(1)}+u^{(2)})} - \frac{u^{(1)}(1-2\mu)}{1-2\mu(u^{(1)}+u^{(2)})} - \frac{u^{(2)}(1-2\mu)}{1-2\mu(u^{(1)}+u^{(2)})} \right] \\
 &= E\{Y_t Y_{t+j-1}\} \rho^{(1)} + E\{Y_t Y_{t+j-2}\} \rho^{(2)} + \mu^2 [1 - \rho^{(1)} - \rho^{(2)}],
 \end{aligned}$$

and a stationary AR(2) process is generated.

The following formulae can also be directly derived from above: $w = \frac{\mu(1-u^{(1)}-u^{(2)})}{1-2\mu(u^{(1)}+u^{(2)})}$,
 $u^{(1)} = \frac{\rho^{(1)}}{1-2\mu(1-\rho^{(1)}-\rho^{(2)})}$ and $u^{(2)} = \frac{\rho^{(2)}}{1-2\mu(1-\rho^{(1)}-\rho^{(2)})}$.

Appendix C

Extension of Kanter (1975) To Nonstationary AR(p)

In order to generate binary data according to a nonstationary AR(p) process, we begin by setting $Y_1 = W_1$, and $P(W_1 = 1) = w_1 = \mu_1$. Subsequently, we generate Y_t , $2 \leq t \leq p$ according to an AR($t - 1$) process, and finally Y_t , $t \geq p + 1$ according to an AR(p) process.

Specifically, we generate using

$$Y_t = \sum_{j=1}^p U_t^{(j)} (Y_{t-j} \oplus W_t) + \left(1 - \sum_{j=1}^p U_t^{(j)}\right) W_t, \quad t \geq p + 1,$$

where

$$P(W_t = 1) = w_t = \frac{\mu_t - \sum_{j=1}^p u_t^{(j)} \mu_{t-j}}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}},$$

$$\begin{aligned} P(U_t^{(j)} = 1) &= u_t^{(j)} \\ &= \frac{\rho^{(j)} \sqrt{\mu_t(1 - \mu_t)}}{\sqrt{\mu_{t-j}(1 - \mu_{t-j})}} \left(1 - 2\mu_t + 2\sqrt{\mu_t(1 - \mu_t)} \sum_{k=1}^p \rho^{(k)} \sqrt{\frac{\mu_{t-k}}{1 - \mu_{t-k}}}\right), \quad 1 \leq j \leq k, \end{aligned}$$

and $P(U_t^{(j)} = 1, U_t^{(k)} = 1) = 0, j \neq k, 1 \leq j, k \leq p$. In addition, W_t is independent of $U_{t'}^{(j)}, 1 \leq j \leq p$, and $U_t^{(j)}$ is independent of $U_{t'}^{(k)}, t \neq t', 1 \leq j, k \leq p$.

C.1 Details

First note that,

$$\begin{aligned}
 \mu_t &= E\{Y_t\} \\
 &= E\left\{\sum_{j=1}^p U_t^{(j)} (Y_{t-j} \oplus W_t) + \left(1 - \sum_{j=1}^p U_t^{(j)}\right) W_t\right\} \\
 &= \sum_{j=1}^p u_t^{(j)} [\mu_{t-j}(1 - w_t) + (1 - \mu_{t-j})w_t] + \left(1 - \sum_{j=1}^p u_t^{(j)}\right) w_t \\
 &= \sum_{j=1}^p u_t^{(j)} (\mu_{t-j} + w_t - 2\mu_{t-j}w_t) + \left(1 - \sum_{j=1}^p u_t^{(j)}\right) w_t \\
 &= w_t \left(1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}\right) + \sum_{j=1}^p u_t^{(j)} \mu_{t-j}.
 \end{aligned}$$

Therefore,

$$w_t = \frac{\mu_t - \sum_{j=1}^p u_t^{(j)} \mu_{t-j}}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}}$$

and

$$\begin{aligned}
& E[Y_t Y_{t-1}], \quad t \geq 2 \\
&= E \left\{ \left[\sum_{j=1}^p U_t^{(j)} (Y_{t-j} \oplus W_t) + \left(1 - \sum_{j=1}^p U_t^{(j)} \right) W_t \right] Y_{t-1} \right\} \\
&= \mu_{t-1} E \left\{ U_t^{(1)} (1 - W_t) + \sum_{j=2}^p U_t^{(j)} (Y_{t-j} \oplus W_t) + \left(1 - U_t^{(1)} - \sum_{j=2}^p U_t^{(j)} \right) W_t \right\} \\
&= \mu_{t-1} \left\{ u_t^{(1)} (1 - w_t) + \sum_{j=2}^p u_t^{(j)} [\mu_{t-j} (1 - w_t) + (1 - \mu_{t-j}) w_t] \right. \\
&\quad \left. + \left(1 - u_t^{(1)} - \sum_{j=2}^p u_t^{(j)} \right) w_t \right\} \\
&= \mu_{t-1} \left\{ u_t^{(1)} (1 - w_t) + \sum_{j=2}^p u_t^{(j)} [\mu_{t-j} + w_t - 2\mu_{t-j} w_t] + \left(1 - u_t^{(1)} - \sum_{j=2}^p u_t^{(j)} \right) w_t \right\} \\
&= \mu_{t-1} \left\{ w_t \left(1 - 2u_t^{(1)} - 2 \sum_{j=2}^p u_t^{(j)} \mu_{t-j} \right) + u_t^{(1)} + \sum_{j=2}^p u_t^{(j)} \mu_{t-j} \right\} \\
&= \mu_{t-1} \left\{ \frac{(\mu_t - \sum_{j=1}^p u_t^{(j)} \mu_{t-j}) (1 - 2u_t^{(1)} - 2 \sum_{j=2}^p u_t^{(j)} \mu_{t-j})}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} \right. \\
&\quad \left. + \frac{(1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}) (u_t^{(1)} + \sum_{j=2}^p u_t^{(j)} \mu_{t-j})}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} \right\} \\
&= \mu_{t-1} \left\{ \frac{\mu_t (1 - 2u_t^{(1)} - 2 \sum_{j=2}^p u_t^{(j)} \mu_{t-j}) - u_t^{(1)} \mu_{t-1} + u_t^{(1)}}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} \right\} \\
&= \mu_{t-1} \left\{ \frac{\mu_t (1 - 2u_t^{(1)} - 2 \sum_{j=2}^p u_t^{(j)} \mu_{t-j}) - u_t^{(1)} \mu_{t-1} + u_t^{(1)}}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} + \frac{\mu_{t-1} \mu_t (1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j})}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} - \mu_{t-1} \mu_t \right\} \\
&= \mu_{t-1} \left\{ \frac{\mu_t (1 - 2u_t^{(1)} - 2 \sum_{j=2}^p u_t^{(j)} \mu_{t-j}) - u_t^{(1)} \mu_{t-1} + u_t^{(1)}}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} \right\} + \frac{\mu_{t-1} \mu_t (1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j})}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} - \mu_{t-1} \mu_t \\
&= \mu_{t-1} \left\{ \frac{\mu_t (2u_t^{(1)} \mu_{t-1} - 2u_t^{(1)}) + u_t^{(1)} (1 - \mu_{t-1})}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} \right\} - \mu_{t-1} \mu_t \\
&= \frac{\mu_{t-1} (1 - \mu_{t-1}) u_t^{(1)} (1 - 2\mu_t)}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} - \mu_{t-1} \mu_t.
\end{aligned}$$

Thus

$$\begin{aligned}
& \text{Corr}(Y_t Y_{t-1}) \\
&= \frac{E[Y_t Y_{t-1}] - E[Y_t]E[Y_{t-1}]}{\sqrt{E\left\{\left(Y_t - E[Y_t]\right)^2\right\}E\left\{\left(Y_{t-1} - E[Y_{t-1}]\right)^2\right\}}} \\
&= \frac{\frac{\mu_{t-1}(1-\mu_{t-1})u_t^{(1)}(1-2\mu_t)}{1-2\sum_{j=1}^p u_t^{(j)}\mu_{t-j}} - \mu_{t-1}\mu_t + \mu_{t-1}\mu_t}{\sqrt{\mu_t(1-\mu_t)}\sqrt{\mu_{t-1}(1-\mu_{t-1})}} \\
&= \frac{\sqrt{\mu_{t-1}(1-\mu_{t-1})}u_t^{(1)}(1-2\mu_t)}{\sqrt{\mu_t(1-\mu_t)}\left(1-2\sum_{j=1}^p u_t^{(j)}\mu_{t-j}\right)} \\
&= \frac{\sqrt{\mu_{t-1}(1-\mu_{t-1})}}{\sqrt{\mu_t(1-\mu_t)}}\left(\frac{u_t^{(1)}(1-2\mu_t)}{1-2\sum_{j=1}^p u_t^{(j)}\mu_{t-j}}\right) \\
&= \rho^{(1)}.
\end{aligned}$$

Similarly, define $\rho^{(j)} = \frac{\sqrt{\mu_{t-j}(1-\mu_{t-j})}}{\sqrt{\mu_t(1-\mu_t)}}\left(\frac{u_t^{(j)}(1-2\mu_t)}{1-2\sum_{k=1}^p u_t^{(k)}\mu_{t-k}}\right)$, $1 \leq j \leq p$.

Now,

$$\begin{aligned}
& E[Y_t Y_{t-u}], \quad u \geq 2 \\
&= E \left\{ \left[\sum_{j=1}^p U_t^{(j)} (Y_{t-j} \oplus W_t) + \left(1 - \sum_{j=1}^p U_t^{(j)} \right) W_t \right] Y_{t-u} \right\} \\
&= E \left\{ Y_{t-u} \left[\sum_{j=1}^p U_t^{(j)} (Y_{t-j} \oplus W_t) \right] \right\} + E \left\{ \left(1 - \sum_{j=1}^p U_t^{(j)} \right) W_t \mu_{t-u} \right\} \\
&= \sum_{j=1}^p E \left\{ U_t^{(j)} Y_{t-u} Y_{t-j} (1 - W_t) \right\} + \sum_{j=1}^p E \left\{ U_t^{(j)} Y_{t-u} (1 - Y_{t-j}) W_t \right\} \\
&\quad + \left(1 - \sum_{j=1}^p u_t^{(j)} \right) w_t \mu_{t-u} \\
&= \sum_{j=1}^p u_t^{(j)} (1 - w_t) E \{ Y_{t-u} Y_{t-j} \} + \mu_{t-u} \sum_{j=1}^p u_t^{(j)} w_t - \sum_{j=1}^p u_t^{(j)} w_t E \{ Y_{t-u} Y_{t-j} \} \\
&\quad + \left(1 - \sum_{j=1}^p u_t^{(j)} \right) w_t \mu_{t-u} \\
&= w_t \mu_{t-u} + \sum_{j=1}^p u_t^{(j)} (1 - 2w_t) E \{ Y_{t-u} Y_{t-j} \} \\
&= \left[\frac{\mu_t - \sum_{j=1}^p u_t^{(j)} \mu_{t-j}}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} \right] \mu_{t-u} \\
&\quad + \sum_{j=1}^p u_t^{(j)} \left[1 - 2 \left(\frac{\mu_t - \sum_{j=1}^p u_t^{(j)} \mu_{t-j}}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} \right) \right] E \{ Y_{t-u} Y_{t-j} \} \\
&= \left\{ \mu_t + \frac{2 \mu_t \sum_{j=1}^p u_t^{(j)} \mu_{t-j} - \sum_{j=1}^p u_t^{(j)} \mu_{t-j}}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} \right\} \mu_{t-u} \\
&\quad + \sum_{j=1}^p \left[\frac{u_t^{(j)} (1 - 2 \mu_t)}{1 - 2 \sum_{j=1}^p u_t^{(j)} \mu_{t-j}} \right] E \{ Y_{t-u} Y_{t-j} \} \\
&= \left\{ \mu_t - \sum_{j=1}^p \mu_{t-j} \left[\frac{u_t^{(j)} (1 - 2 \mu_t)}{1 - 2 \sum_{k=1}^p u_t^{(k)} \mu_{t-k}} \right] \right\} \mu_{t-u} \\
&\quad + \sum_{j=1}^p \rho^{(j)} \left[\frac{\sqrt{\mu_t (1 - \mu_t)}}{\sqrt{\mu_{t-j} (1 - \mu_{t-j})}} \right] E \{ Y_{t-u} Y_{t-j} \} \\
&= \mu_t \mu_{t-u} - \sum_{j=1}^p \rho^{(j)} \left[\frac{\sqrt{\mu_t (1 - \mu_t)}}{\sqrt{\mu_{t-j} (1 - \mu_{t-j})}} \right] \mu_{t-j} \mu_{t-u} \\
&\quad + \sum_{j=1}^p \rho^{(j)} \left[\frac{\sqrt{\mu_t (1 - \mu_t)}}{\sqrt{\mu_{t-j} (1 - \mu_{t-j})}} \right] E \{ Y_{t-u} Y_{t-j} \} \\
&= \mu_t \mu_{t-u} + \sum_{j=1}^p \rho^{(j)} \left[\frac{\sqrt{\mu_t (1 - \mu_t)}}{\sqrt{\mu_{t-j} (1 - \mu_{t-j})}} \right] \left[E \{ Y_{t-u} Y_{t-j} \} - \mu_{t-j} \mu_{t-u} \right].
\end{aligned}$$

Hence an AR(p) process is generated.

Finally, in order to express $u_t^{(j)}$ in terms of $(\rho^{(1)}, \rho^{(2)}, \dots, \rho^{(p)})$ and $(\mu_{t-p}, \mu_{t-p+1}, \dots, \mu_t)$, $1 \leq j \leq p$, we proceed as follows

$$\begin{aligned}
\rho^{(j)} &= \frac{\sqrt{\mu_{t-j}(1-\mu_{t-j})}}{\sqrt{\mu_t(1-\mu_t)}} \left(\frac{u_t^{(j)}(1-2\mu_t)}{1-2\sum_{k=1}^p u_t^{(k)}\mu_{t-k}} \right) \\
\Rightarrow u_t^{(j)} &= \rho^{(j)} \frac{\sqrt{\mu_t(1-\mu_t)}}{\sqrt{\mu_{t-j}(1-\mu_{t-j})}} \left(\frac{1-2\sum_{k=1}^p u_t^{(k)}\mu_{t-k}}{1-2\mu_t} \right) \\
\Rightarrow \sum_{j=1}^p u_t^{(j)}\mu_{t-j} &= \frac{\sqrt{\mu_t(1-\mu_t)}}{1-2\mu_t} \left(1-2\sum_{k=1}^p u_t^{(k)}\mu_{t-k} \right) \sum_{j=1}^p \frac{\rho^{(j)}\mu_{t-j}}{\sqrt{\mu_{t-j}(1-\mu_{t-j})}} \\
\Rightarrow \sum_{k=1}^p u_t^{(k)}\mu_{t-k} &= \frac{\sqrt{\mu_t(1-\mu_t)}}{1-2\mu_t} \left(1-2\sum_{k=1}^p u_t^{(k)}\mu_{t-k} \right) \sum_{k=1}^p \frac{\rho^{(k)}\mu_{t-k}}{\sqrt{\mu_{t-k}(1-\mu_{t-k})}} \\
\Rightarrow \left(\sum_{k=1}^p u_t^{(k)}\mu_{t-k} \right) &\left(1-2\mu_t + 2\sqrt{\mu_t(1-\mu_t)} \sum_{k=1}^p \rho^{(k)} \sqrt{\frac{\mu_{t-k}}{1-\mu_{t-k}}} \right) \\
&= \sqrt{\mu_t(1-\mu_t)} \sum_{k=1}^p \rho^{(k)} \sqrt{\frac{\mu_{t-k}}{1-\mu_{t-k}}} \\
\Rightarrow \sum_{k=1}^p u_t^{(k)}\mu_{t-k} &= \frac{\sqrt{\mu_t(1-\mu_t)} \sum_{k=1}^p \rho^{(k)} \sqrt{\frac{\mu_{t-k}}{1-\mu_{t-k}}}}{\left(1-2\mu_t + 2\sqrt{\mu_t(1-\mu_t)} \sum_{k=1}^p \rho^{(k)} \sqrt{\frac{\mu_{t-k}}{1-\mu_{t-k}}} \right)} \\
\Rightarrow 1-2\sum_{k=1}^p u_t^{(k)}\mu_{t-k} &= \frac{1-2\mu_t}{\left(1-2\mu_t + 2\sqrt{\mu_t(1-\mu_t)} \sum_{k=1}^p \rho^{(k)} \sqrt{\frac{\mu_{t-k}}{1-\mu_{t-k}}} \right)} \\
\Rightarrow \rho^{(j)} &= \frac{\sqrt{\mu_{t-j}(1-\mu_{t-j})}}{\sqrt{\mu_t(1-\mu_t)}} \left(1-2\mu_t + 2\sqrt{\mu_t(1-\mu_t)} \sum_{k=1}^p \rho^{(k)} \sqrt{\frac{\mu_{t-k}}{1-\mu_{t-k}}} \right) u_t^{(j)} \\
\Rightarrow u_t^{(j)} &= \frac{\rho^{(j)} \sqrt{\mu_t(1-\mu_t)}}{\sqrt{\mu_{t-j}(1-\mu_{t-j})} \left(1-2\mu_t + 2\sqrt{\mu_t(1-\mu_t)} \sum_{k=1}^p \rho^{(k)} \sqrt{\frac{\mu_{t-k}}{1-\mu_{t-k}}} \right)}.
\end{aligned}$$

Appendix D

Programming Code

D.1 Maple code

```
with(linalg):
C1_1 := 1:
C1_2 := rho[1]:
C1_3 := rho[1]*C1_2 + rho[2]*C1_1:
C1_4 := rho[1]*C1_3 + rho[2]*C1_2:
C1_5 := rho[1]*C1_4 + rho[2]*C1_3:
C1_6 := rho[1]*C1_5 + rho[2]*C1_4:
C1_7 := rho[1]*C1_6 + rho[2]*C1_5:
C1_8 := rho[1]*C1_7 + rho[2]*C1_6:
C1_9 := rho[1]*C1_8 + rho[2]*C1_7:
C1_10 := rho[1]*C1_9 + rho[2]*C1_8:
C2_3 := rho[1]*C1_3:
C2_4 := rho[1]*C1_4:
C2_5:= rho[1]*C1_5:
C2_6 := rho[1]*C1_6:
C2_7 := rho[1]*C1_7:
C2_8 := rho[1]*C1_8:
C2_9 := rho[1]*C1_9:
C2_10 := rho[1]*C1_10:
C3_4 := rho[1]*C2_4 + rho[2]*C1_4:
C3_5 := rho[1]*C2_5 + rho[2]*C1_5:
C3_6 := rho[1]*C2_6 + rho[2]*C1_6:
C3_7 := rho[1]*C2_7 + rho[2]*C1_7:
C3_8 := rho[1]*C2_8 + rho[2]*C1_8:
C3_9 := rho[1]*C2_9 + rho[2]*C1_9:
```



```

C3_10 := rho[1]*C2_10 + rho[2]*C1_10:
C4_5 := rho[1]*C3_5 + rho[2]*C2_5:
C4_6 := rho[1]*C3_6 + rho[2]*C2_6:
C4_7 := rho[1]*C3_7 + rho[2]*C2_7:
C4_8 := rho[1]*C3_8 + rho[2]*C2_8:
C4_9 := rho[1]*C3_9 + rho[2]*C2_9:
C4_10 := rho[1]*C3_10 + rho[2]*C2_10:
C5_6 := rho[1]*C4_6 + rho[2]*C3_6:
C5_7 := rho[1]*C4_7 + rho[2]*C3_7:
C5_8 := rho[1]*C4_8 + rho[2]*C3_8:
C5_9 := rho[1]*C4_9 + rho[2]*C3_9:
C5_10 := rho[1]*C4_10 + rho[2]*C3_10:
C6_7 := rho[1]*C5_7 + rho[2]*C4_7:
C6_8 := rho[1]*C5_8 + rho[2]*C4_8:
C6_9 := rho[1]*C5_9 + rho[2]*C4_9:
C6_10 := rho[1]*C5_10 + rho[2]*C4_10:
C7_8 := rho[1]*C6_8 + rho[2]*C5_8:
C7_9 := rho[1]*C6_9 + rho[2]*C5_9:
C7_10 := rho[1]*C6_10 + rho[2]*C5_10:
C8_9 := rho[1]*C7_9 + rho[2]*C6_9:
C8_10 := rho[1]*C7_10 + rho[2]*C6_10:
C9_10 := rho[1]*C8_10 + rho[2]*C7_10:
# t=10
C_x := Matrix([
  [1, C1_2, C1_3, C1_4, C1_5, C1_6, C1_7, C1_8, C1_9 ],
  [C1_2, 1, C2_3, C2_4, C2_5, C2_6, C2_7, C2_8, C2_9 ],
  [C1_3, C2_3, 1, C3_4, C3_5, C3_6, C3_7, C3_8, C3_9 ],
  [C1_4, C2_4, C3_4, 1, C4_5, C4_6, C4_7, C4_8, C4_9 ],
  [C1_5, C2_5, C3_5, C4_5, 1, C5_6, C5_7, C5_8, C5_9 ],
  [C1_6, C2_6, C3_6, C4_6, C5_6, 1, C6_7, C6_8, C6_9 ],
  [C1_7, C2_7, C3_7, C4_7, C5_7, C6_7, 1, C7_8, C7_9 ],
  [C1_8, C2_8, C3_8, C4_8, C5_8, C6_8, C7_8, 1, C8_9 ],
  [C1_9, C2_9, C3_9, C4_9, C5_9, C6_9, C7_9, C8_9, 1 ]
]):
VInvdiag := Matrix([diag(1/sigma,1/sigma,1/sigma,1/sigma,1/sigma,
  1/sigma,1/sigma,1/sigma,1/sigma)]):
Cinv_x := 1/C_x:
cov_xy := sigma^2*Matrix([ [C1_10],[C2_10],[C3_10],
  [C4_10],[C5_10],[C6_10],[C7_10],[C8_10],[C9_10]]):
Ginv := VInvdiag . Cinv_x:
S := VInvdiag . cov_xy:
b := simplify(Ginv . S);
simplify(C1_10);

```

D.2 FORTRAN code

```
PROGRAM Binary
```

```
USE MSFLIB
USE MSIMSL
USE commonData
USE IO
USE rhoRelated
USE estimate
USE generate
USE PORTLIB
```

```
USE corrEst
```

```
IMPLICIT NONE
REAL(4) s
```

```
CALL readData()
CALL setRestrictions()
202 FORMAT(10(F9.5))
```

```
OPEN (UNIT=4, FILE="NonStationary Rho Ranges.dat")
DO method = 1, numMethods
DO likeCluster = 1, numLikeClusters
SELECT CASE (usedMethods(method))
CASE ('B')
print 203, "Bahadur      mui =",
mu(clusterBoundaries(likeCluster,1),1), "",
rhoRestriction(method,clusterBoundaries(likeCluster,1),1), "<rho<",
rhoRestriction(method,clusterBoundaries(likeCluster,1),2),
" ", corrNames(corrType)
rhoRestriction(method,clusterBoundaries(likeCluster,1),1),
rhoRestriction(method,clusterBoundaries(likeCluster,1),2)
CASE ('Q')
print 203, "Qaqish      mui =",
mu(clusterBoundaries(likeCluster,1),1), "",
rhoRestriction(method,clusterBoundaries(likeCluster,1),1), "<rho<",
rhoRestriction(method,clusterBoundaries(likeCluster,1),2),
" ", corrNames(corrType)
rhoRestriction(method,clusterBoundaries(likeCluster,1),1),
rhoRestriction(method,clusterBoundaries(likeCluster,1),2)
```

```

CASE ('K')
print 203, "Kanter      mui =",
mu(clusterBoundaries(likeCluster,1),1), "",
rhoRestriction(method,clusterBoundaries(likeCluster,1),1), "<rho<",
rhoRestriction(method,clusterBoundaries(likeCluster,1),2),
" ", corrNames(corrType)
END SELECT
ENDDO
ENDDO
100 FORMAT(A12, F5.2, 2(F6.3))
CLOSE(4)

203 FORMAT(A20, F10.3, A8, F10.3, A8, F10.3, a3, A5)

corrEstimate(:, :, :, :, :) = 1

s = SECNDS(0.0)
DO rep = 1, REPS
print *, "generating rep", rep
  CALL generateData()
print *, "estimating rep", rep
  CALL estimationCORR()
  CALL estimationNEWTON()

ENDDO
s = SECNDS(s)
PRINT *, 'Procedure took ',s, 'seconds to run.'

CALL writeData()

END PROGRAM Binary

MODULE commonData

IMPLICIT NONE

INTEGER, parameter :: T=4, p=2, REPS=1000, ITMAX=100, seedStart=1320
INTEGER, parameter :: AR1 = 1, MA=2, EX =3, IND = 4

```

```

REAL, parameter :: ERRREL = 0.0001
CHARACTER (5), parameter :: corrNames(4)= ('AR1', 'MA', 'EX', 'IND'/)

! paramters obtained from info.dat
INTEGER :: K, numLikeClusters, count, corrType, numMethods, transType
! counters for DO loops
INTEGER :: rep, method, likeCluster, i, j
! combine estimate over clusters or not
LOGICAL :: combined, stationary
! which of B, K and Q are used
CHARACTER*1 :: usedMethods(3)

! numLikeClusters, 2 (start/end)
INTEGER, ALLOCATABLE, DIMENSION (:,:) :: clusterBoundaries
! method, i, t
INTEGER, ALLOCATABLE, DIMENSION (:,::,:) :: y
! i, t, p
REAL, ALLOCATABLE, DIMENSION (:,::,:) :: X
! rep, method, p
REAL, ALLOCATABLE, DIMENSION (:,::,:) :: betaEstimate
! i, t
REAL, ALLOCATABLE, DIMENSION (:) :: gama
! i, t
REAL, ALLOCATABLE, DIMENSION (:,:) :: mu
! i
REAL, ALLOCATABLE, DIMENSION (:) :: rho
! i, t, t
REAL, ALLOCATABLE, DIMENSION (:,::,:) :: C
! rep, method, i
REAL, ALLOCATABLE, DIMENSION (:,::,:) :: rhoEstimate
! method, i, 2 (min/max)
REAL, ALLOCATABLE, DIMENSION (:,::,:) :: rhoRestriction
! rep, method, i
LOGICAL, ALLOCATABLE, DIMENSION (:,::,:) :: rangeViolated
! likeCluster
CHARACTER (5), ALLOCATABLE, DIMENSION (:) :: corrStructure
! numLikeClusters, reps, method, t, t
REAL, ALLOCATABLE, DIMENSION (:,:::,:::,:::) :: corrEstimate

CONTAINS

SUBROUTINE Allocate

```

```

ALLOCATE (clusterBoundaries(numLikeClusters,2))
ALLOCATE (y(numMethods,K,T))
ALLOCATE (x(K,T,P))
ALLOCATE (betaEstimate(REPS,numMethods,p))
ALLOCATE (mu(K,T))
ALLOCATE (rho(K))
ALLOCATE (C(K,T,T))
ALLOCATE (rhoEstimate(REPS,numMethods,K))
ALLOCATE (rhoRestriction(numMethods,K,2))
ALLOCATE (gama(K))
ALLOCATE (rangeViolated(REPS,numMethods,K))
ALLOCATE (corrStructure(numLikeClusters))
rangeViolated = .FALSE.

ALLOCATE (corrEstimate(numLikeClusters,REPS,numMethods,T,T))

END SUBROUTINE Allocate

LOGICAL FUNCTION validProbability(probability)
REAL, INTENT(IN) :: probability
validProbability = (probability<1 .AND. probability>0)
ENDFUNCTION validProbability

SUBROUTINE evaluateMu(beta, mu)
REAL :: power, beta(p), mu(K,T)

DO j=1,K
DO i=1,T
power = sum(X(j,i,:) * beta)
mu(j,i) = exp(power)/(1+exp(power))
ENDDO
ENDDO
ENDSUBROUTINE evaluateMu

END MODULE commonData

MODULE IO

USE commonData
USE rhoRelated
USE correlationRelated
IMPLICIT NONE

```

CONTAINS

```

SUBROUTINE readLine(startCluster)
  INTEGER, INTENT(INOUT) :: startCluster
  INTEGER :: numGroup, endCluster, j
  REAL :: muInput, rhoInput, xInput(p), Xt2(T)
  CHARACTER (5) :: corrStructureInput

  READ(2, *) numGroup, muInput, rhoInput, xInput, corrStructureInput
  PRINT 100, numGroup, corrStructureInput, muInput, rhoInput, xInput
  100 FORMAT(I5, A6, 10(F6.3))
  endCluster = startCluster + numGroup - 1
  clusterBoundaries(likeCluster,1) = startCluster
  clusterBoundaries(likeCluster,2) = endCluster
  mu(startCluster:endCluster,:) = muInput
  X(startCluster:endCluster,:,1) = xInput(1)
  X(startCluster:endCluster,:,2) = calcX(xInput(1), muInput)

  IF(.NOT. stationary) THEN
    DO i=1,T
      SELECT CASE (transType)
        CASE (1)
          Xt2(i) = i/4.0
        CASE (2)
          Xt2(i) = -i/4.0
        CASE DEFAULT
          print *, "Transformation Type ", transType, " is not valid."
          CALL EXIT(-1)
        ENDSELECT
      ENDDO
      X(startCluster:endCluster,:,1) = calcX(Xt2(1), muInput)
      X(startCluster:endCluster,1,2) = Xt2(1)
      X(startCluster:endCluster,2,2) = Xt2(2)
      X(startCluster:endCluster,3,2) = Xt2(3)
      X(startCluster:endCluster,4,2) = Xt2(4)
    ENDIF

    corrStructure(likeCluster) = corrStructureInput
    CALL getCorrType(corrStructure(likeCluster))
    CALL getCorrStructure(rhoInput, C(startCluster,:,:))
    DO j=startCluster, endCluster
      rho(j) = rhoInput
    
```

```

rhoEstimate(:, :, j) = rhoInput
C(j, :, :) = C(startCluster, :, :)
ENDDO
startCluster = endCluster + 1
ENDSUBROUTINE readLine

REAL FUNCTION calcX(X, mu)
REAL :: X, mu
calcX = log(mu/(1-mu))-X
ENDFUNCTION calcX

SUBROUTINE readParameters()
INTEGER :: startCluster
REAL :: beta(p)

beta = 1.0
startCluster = 1
likeCluster = 1
OPEN (UNIT=2, FILE="Info.dat")
READ(2, *)
READ(2, *)
READ(2, *)
IF(.NOT. stationary) THEN
READ(2, *)
ENDIF

      DO WHILE (.NOT. EOF(2))
CALL readLine(startCluster)
likeCluster = likeCluster + 1
END DO
CALL evaluateMu(beta, mu)

DO j=1,K
print *, j, mu(j, :)
ENDDO
CLOSE(2)
print *, "Cluster Boundaries", clusterBoundaries
ENDSUBROUTINE readParameters

SUBROUTINE readK()
INTEGER :: KInput

```

```

numLikeClusters=0
  DO WHILE (.NOT. EOF(2))
READ(2, *) KInput
K = K + KInput
numLikeClusters = numLikeClusters +1
END DO

CLOSE(2)
ENDSUBROUTINE readK

SUBROUTINE readMethods
OPEN (UNIT=2, FILE="Info.dat")
READ(2, *) numMethods, (usedMethods(i), i = 1, numMethods)
READ(2, *) combined
READ(2, *) stationary
IF(.NOT. stationary) THEN
READ(2, *) transType
ENDIF

ENDSUBROUTINE readMethods

SUBROUTINE readData()
CALL readMethods()
CALL readK()
CALL Allocate()
CALL readParameters()
IF(.NOT. stationary) THEN
clusterBoundaries(1,2)=clusterBoundaries(numLikeClusters,2)
ENDIF
numLikeClusters=1
ENDSUBROUTINE readData

SUBROUTINE writeY()
INTEGER :: numGroup
REAL :: muInput, rhoInput
CHARACTER (5) :: corrStructure

OPEN (UNIT=3, FILE="Binary.csv")
OPEN (UNIT=2, FILE="Info.dat")

98 FORMAT(5(A12, ","))

```



```

WRITE(3, 98) "Clusters", "Mu", "Rho", "Correlation"
99 FORMAT(I5, ",", 2(F6.3, ","), A5, ",")

READ(2, *)
READ(2, *)
READ(2, *) stationary
IF(.NOT. stationary) THEN
READ(2, *) transType
ENDIF
DO WHILE (.NOT. EOF(2))
READ(2, *) numGroup, muInput, rhoInput, corrStructure
WRITE(3, 99) numGroup, muInput, rhoInput, corrStructure
END DO

100 FORMAT(4(I5, ","))
DO method = 1, numMethods
WRITE (3,*) ""
SELECT CASE (usedMethods(method))
CASE ('B')
WRITE (3,*) "Bahadur,"
CASE ('Q')
WRITE (3,*) "Qaqish,"
CASE ('K')
WRITE (3,*) "Kanter,"
END SELECT
DO i = 1, K
WRITE (3,100) y(method, i, :)
ENDDO
ENDDO

CLOSE (3)

ENDSUBROUTINE writeY

SUBROUTINE writeSummaryStatistics()
INTEGER :: j
LOGICAL dummy(REPS)
INTEGER :: numGroup
REAL :: muInput, rhoInput, xInput(p)
CHARACTER (5) :: corrStructureInput

OPEN (UNIT=4, FILE="Summary.csv")

```

```

OPEN (UNIT=2, FILE="Info.dat")

98 FORMAT(7(A12, ","))
WRITE(4, 98) "Clusters", "Mu_1", "Rho", "X_11", "X_12", "Correlation"
99 FORMAT(I5, ",", 4(F6.3, ","), A5)

READ(2, *)
READ(2, *)
READ(2, *) stationary
IF(.NOT. stationary) THEN
READ(2, *) transType
ENDIF
likeCluster = 1
  DO WHILE (.NOT. EOF(2))
READ(2, *) numGroup, muInput, rhoInput, xInput, corrStructureInput
WRITE(4, 99) numGroup, muInput, rhoInput, x(likeCluster, 1, :),
              corrStructureInput
likeCluster = likeCluster + numGroup
END DO

DO method = 1, numMethods
WRITE (4,*) ""
SELECT CASE (usedMethods(method))
CASE ('B')
WRITE (4,*) "Bahadur,"
CASE ('Q')
WRITE (4,*) "Qaqish,"
CASE ('K')
WRITE (4,*) "Kanter,"
END SELECT

WRITE (4,100) "Variable", "Mean", "SD", "95% CI", "% RR Violations"
100 FORMAT(4(A9, ","), A20, ",")

DO j = 1, p
CALL writeSummary("Beta", j, betaEstimate(:,method,j), dummy)
ENDDO

IF(combined) THEN
DO likeCluster = 1, numLikeClusters
CALL writeSummary("Rho ", clusterBoundaries(likeCluster,1),
  rhoEstimate(:,method,clusterBoundaries(likeCluster,1)),
  rangeViolated(:,method,clusterBoundaries(likeCluster,1)))

```

```

ENDDO
ELSE
DO i = 1, K
CALL writeSummary("Rho ", i, rhoEstimate(:,method,i),
    rangeViolated(:,method,i))
ENDDO
ENDIF

ENDDO

CALL printCorr()
close(4)

ENDSUBROUTINE writeSummaryStatistics

SUBROUTINE writeSummary(parameterName, parameterNum,
    estimates, rangeViolated)
CHARACTER (4) :: parameterName
INTEGER :: parameterNum
REAL estimates(REPS), mean, sd, numRRViolations, CI(2),
    REAL estimatesSorted(REPS)
LOGICAL rangeViolated(REPS)

100 FORMAT(A5, I3, ",", 2(F12.7, ","),
    """"(", F12.7, ",", F12.7, ")""", I3, ",")

mean = sum(estimates)/REPS
sd = sqrt(sum((estimates-mean)**2)/real(REPS-1))
CALL SVRGN (REPS, estimates, estimatesSorted)
CI(1) = estimatesSorted(int(0.025*REPS))
CI(2) = estimatesSorted(int(0.975*REPS))

numRRViolations =0

IF(parameterName == "Rho ") THEN
DO rep = 1,REPS
numRRViolations = numRRViolations + rangeViolated(rep)
ENDDO
numRRViolations = 100.0*numRRViolations/real(REPS)
WRITE(4,100) parameterName, parameterNum, mean, sd, CI(1), CI(2),
    int(numRRViolations)
ELSE

```

```

WRITE(4,100) parameterName, parameterNum, mean, sd, CI(1), CI(2)
ENDIF
ENDSUBROUTINE

```

```

SUBROUTINE printCorr()
WRITE (4,*) ""
WRITE (4,*) ""

DO method = 1, numMethods
WRITE (4,*) ""
SELECT CASE (usedMethods(method))
CASE ('B')
WRITE (4,*) "Bahadur,"
CASE ('Q')
WRITE (4,*) "Qaqish,"
CASE ('K')
WRITE (4,*) "Kanter,"
END SELECT
CALL writeCorr(method)
ENDDO
END SUBROUTINE printCorr

```

```

SUBROUTINE writeCorr(method)
INTEGER :: method
REAL :: estimate(T,T), variance(T,T)

101 FORMAT(4(F5.3, ","), ",", 4(F5.2, ","))
102 FORMAT(A20, ",", A10, F5.2, ",,," , A30)

```

```

DO likeCluster = 1,numLikeClusters
WRITE(4, 102) "Correlation Estimate", "rho = ",
    rho(clusterBoundaries(likeCluster,1)),
    "Variance estimate x 10^2 ="
estimate = sum(corrEstimate(likeCluster,:,method,:,1)/REPS
variance = sum(corrEstimate(likeCluster,:,method,:,1)**2)/REPS
    - (estimate)**2
DO i = 1,T
WRITE(4, 101) estimate(i,:), 100*variance(i,:)
ENDDO
ENDDO

END SUBROUTINE writeCorr

```

```

SUBROUTINE writeData()
CALL writeSummaryStatistics()
ENDSUBROUTINE writeData

ENDMODULE IO

MODULE rhoRelated

USE commonData
USE correlationRelated
USE Bahadur
USE Qaqish
USE Kanter
IMPLICIT NONE

CONTAINS

LOGICAL FUNCTION valid(testRho)
real testRho
SELECT CASE (usedMethods(method))
CASE ('B')
valid = validBahadur(testRho)
CASE ('Q')
valid = validQaqish(testRho)
CASE ('K')
valid = validKanter(testRho)
END SELECT
ENDFUNCTION valid

! iteratively find valid rho limit with accuracy 1/EPSILON
REAL RECURSIVE FUNCTION getRhoBoundary(rhoValid, rhoNotValid)
    RESULT (rhoMidpoint)
REAL, INTENT(IN) :: rhoValid, rhoNotValid
rhoMidpoint = (rhoValid + rhoNotValid)/2.0
IF (abs(rhoValid - rhoNotValid) > EPSILON(rhoValid)) THEN
IF (valid(rhoMidpoint)) THEN
rhoMidpoint = getRhoBoundary(rhoMidpoint, rhoNotValid)
ELSE
rhoMidpoint = getRhoBoundary(rhoValid, rhoMidpoint)
ENDIF
ENDIF
ENDFUNCTION getRhoBoundary

```

```

SUBROUTINE setRestrictions_i()
IF(usedMethods(method)=='K' .AND. corrType==2) THEN
CALL rhoRestrictionMAKanter
ELSE
! if rho values in info.dat are not valid then exit
IF (.NOT.valid(rho(i))) THEN
PRINT *, "rho_i = ", rho(i), " and mu_i1 = ", mu(i,1),
      " is not valid for method ", usedMethods(method)
      CALL EXIT(-1)
ELSE
! set lower rho boundary limit
IF (valid(-1.0)) THEN
rhoRestriction(method,i,1) = -1.0
ELSE
rhoRestriction(method,i,1) = getRhoBoundary(rho(i), -1.0)
ENDIF

! set uppper rho boundary limit
IF (valid(1.0)) THEN
rhoRestriction(method,i,2) = 1.0
ELSE
rhoRestriction(method,i,2) = getRhoBoundary(rho(i), 1.0)
ENDIF
ENDIF
ENDIF
ENDSUBROUTINE setRestrictions_i

SUBROUTINE setRestrictions()
DO method = 1,numMethods
i=1
CALL setRestrictions_i()
DO i=2,K
IF (mu(i,1).ne.mu(i-1,1)) THEN
CALL setRestrictions_i()
ELSE
rhoRestriction(:,i,:) = rhoRestriction(:,i-1,:)
gama(i) = gama(i-1)
ENDIF
ENDDO
ENDDO

```

```
ENDSUBROUTINE setRestrictions
```

```
SUBROUTINE estimateRhoi(mu)
  INTEGER j
  REAL r(T), numerator, denominator, mu(T), estimatedRho

  r = 0; numerator=0; denominator=0

  r = (y(method,i,:)-mu)/sqrt(mu*(1-mu))
  DO j = 1,(T-1)
    denominator = denominator + r(j)**2
    numerator = numerator + r(j)*r(j+1)
  ENDDO
  denominator = denominator + r(T)**2
  estimatedRho = (T/(T-1))*numerator/denominator

  rangeViolated(rep,method,i) =
    (estimatedRho <= rhoRestriction(method, i, 1)) +
    (estimatedRho >= rhoRestriction(method, i, 2))
  rhoEstimate(rep,method, i) = estimatedRho
ENDSUBROUTINE estimateRhoi
```

```
SUBROUTINE estimateRho(mu)
  REAL, INTENT(IN):: mu(K,T)
  DO i=1,K
    CALL estimateRhoi(mu(i,:))
  ENDDO
ENDSUBROUTINE estimateRho
```

```
SUBROUTINE estimateRhoCluster(mu)
  REAL r(K,T), numerator, denominator, mu(K,T), estimatedRho

  r=0; numerator=0; denominator=0
  DO i = clusterBoundaries(likeCluster,1),
    clusterBoundaries(likeCluster,2)
    r(i,:) = (y(method,i,:)-mu(i,:))/sqrt(mu(i,:)*(1-mu(i,:)))
    DO j = 1, (T-1)
      numerator = numerator + r(i,j)*r(i,j+1)
      denominator = denominator + r(i,j)**2
    ENDDO
    denominator = denominator + r(i,T)**2
  ENDDO
```

```

estimatedRho = (T*numerator)/((T-1)*denominator)
rangeViolated(rep,method,
clusterBoundaries(likeCluster,1):clusterBoundaries(likeCluster,2)) =
(estimatedRho <=
    rhoRestriction(method, clusterBoundaries(likeCluster,1),1)).OR.
(estimatedRho >=
    rhoRestriction(method, clusterBoundaries(likeCluster,1), 2))
rhoEstimate(rep,method,
clusterBoundaries(likeCluster,1):clusterBoundaries(likeCluster,2)) =
    estimatedRho
ENDSUBROUTINE estimateRhoCluster

SUBROUTINE estimateRhoCombined(mu)
REAL, INTENT(IN):: mu(K,T)
DO likeCluster = 1,numLikeClusters
CALL estimateRhoCluster(mu)
ENDDO
ENDSUBROUTINE estimateRhoCombined

ENDMODULE rhoRelated

MODULE estimate

USE MSIMSL
USE commonData
USE rhoRelated
USE correlationRelated
IMPLICIT NONE

CONTAINS

SUBROUTINE GQL(beta, F, p)
INTEGER, INTENT(IN) :: p
REAL, INTENT(IN) :: beta(p)
REAL, INTENT(OUT) :: F(p)
REAL :: mu(K,T), C(T,T), INVC(T,T), temp(p)

CALL evaluateMu(beta, mu)
IF(COMBINED) THEN
CALL estimateRhoCombined(mu)
ELSE
CALL estimateRho(mu)

```



```

ENDIF

F=0
DO i=1,K
CALL getCorrStructure(rhoEstimate(rep,method,i), C)
CALL LINRG (T, C, T, INVC, T)
CALL SGEMV ('N', p, T, 1.0, MATMUL(TRANSPPOSE(X(i,:,:)),INVC), p,
          (y(method,i,:) - mu(i,:)), 1, 0.0, temp, 1)
F = F + temp
ENDDO
ENDSUBROUTINE GQL

SUBROUTINE estimationGQL()
REAL :: betaGuess(p), FNORM

DO method = 1, numMethods
betaGuess = (1.0,1.0)
CALL NEQNF(GQL, ERRREL, p, ITMAX, betaGuess,
          betaEstimate(rep,method,:), FNORM)
202 FORMAT(10(F7.3))
201 FORMAT(A7, 3(I3), 10(F8.4))
ENDDO
ENDSUBROUTINE estimationGQL

RECURSIVE SUBROUTINE Newton(beta, iter)
REAL :: beta(p), betaNew(p), residual(p), coeff(p,p)
REAL :: C(T,T), sigmaInv(T,T), mu(K,T), A(T,T), del(P,T)
INTEGER :: iter

residual = 0
coeff = 0
C = 0
iter = iter + 1

CALL evaluateMu(beta, mu)

IF(COMBINED) THEN
CALL estimateRhoCombined(mu)
ELSE
CALL estimateRho(mu)
ENDIF
DO i=1,K

```

```

CALL getCorrStructure(rhoEstimate(rep,method,i), C)
A = 0.0; do j = 1, T; A(j,j) = mu(i,j)*(1-mu(i,j)); end do
CALL LINRG (T, MATMUL(MATMUL(sqrt(A), C), sqrt(A)), T, sigmaInv, T)
del = MATMUL(TRANSPPOSE(X(i,:,:), A)
coeff = coeff + MATMUL(MATMUL(del, sigmaInv), TRANSPPOSE(del))
residual = residual +
                MATMUL(MATMUL(del, sigmaInv), (y(method,i,:) - mu(i,:)))
ENDDO

CALL LINRG (p, coeff, p, coeff, p)
betaNew = beta + MATMUL(coeff, residual)

IF (SNRM2(P, (betaNew-beta), 1)/SNRM2(P, betaNew, 1))>=ERRREL .AND.
    count<=ITMAX) THEN
CALL Newton(betaNew, iter)
ENDIF
beta = betaNew
ENDSUBROUTINE Newton

SUBROUTINE estimationNEWTON()
DO method = 1, numMethods
betaEstimate(rep,method,:) = (1.0,1.0)
CALL Newton(betaEstimate(rep,method,:), 0)
202 FORMAT(10(F7.3))
201 FORMAT(A7, 2(I3), 10(F8.4))
ENDDO
ENDSUBROUTINE estimationNEWTON

END MODULE estimate

MODULE generate

USE commonData
USE Bahadur
USE Qaqish
USE Kanter
IMPLICIT NONE

CONTAINS

SUBROUTINE generateData()
DO method = 1, numMethods
CALL RNSET(seedStart+rep)

```

```

DO i = 1, K
SELECT CASE (usedMethods(method))
CASE ('B')
CALL generateBahadur()
CASE ('Q')
CALL generateQaqish()
CASE ('K')
CALL generateKanter()
END SELECT
ENDDO
ENDDO
ENDSUBROUTINE generateData

ENDMODULE generate

MODULE corrEst

USE commonData
IMPLICIT NONE

CONTAINS

SUBROUTINE CORR_AB(a,b,r)
REAL a, b, r(K,T), numerator, denominator

DO likeCluster = 1,numLikeClusters
numerator=0; denominator=0
DO i = clusterBoundaries(likeCluster,1),
      clusterBoundaries(likeCluster,2)
numerator = numerator + r(i,a)*r(i,b)
DO j = 1,T
denominator = denominator + r(i,j)**2
ENDDO
ENDDO
corrEstimate(likeCluster,rep,method,a,b) = T*numerator/denominator
ENDDO
END SUBROUTINE CORR_AB

SUBROUTINE estimationCorr()
REAL r(K,T), a,b

DO method = 1,numMethods

```

```

      r=0
DO likeCluster = 1,numLikeClusters
DO i = clusterBoundaries(likeCluster,1),
      clusterBoundaries(likeCluster,2)
r(i,:) = (y(method,i,:)-mu(i,:))/sqrt(mu(i,:)*(1-mu(i,:)))
ENDDO
ENDDO

DO a = 1,(T-1)
DO b = (a+1),T
CALL CORR_AB(a,b,r)
corrEstimate(:,rep,method,b,a) = corrEstimate(:,rep,method,a,b)
ENDDO
ENDDO
ENDDO
ENDSUBROUTINE estimationCorr

ENDMODULE corrEst

MODULE correlationRelated

USE commonData

IMPLICIT NONE

CONTAINS

SUBROUTINE getCorrStructure(rho, C)
REAL, INTENT(IN) :: rho
REAL, INTENT(INOUT) :: C(T,T)
INTEGER :: j, k

SELECT CASE (corrType)
CASE (AR1)
C = 1
DO j = 1, T
DO k = 1, T
C(j, k) = rho**(abs(j-k))
ENDDO
ENDDO

CASE (MA)
C = 0

```

```

DO j = 1, T
DO k = j-1, j+1
IF (k>=1 .AND. k<=T) THEN
C(j, k) = rho**(abs(k-j))
ENDIF
ENDDO
ENDDO

CASE (EX)
C = rho
DO j = 1, T
C(j, j) = 1
ENDDO

CASE (IND)
C = 0
DO j = 1, T
C(j, j) = 1
ENDDO

CASE DEFAULT
print *, "Correlation Structure ", corrStructure, "not valid."
CALL EXIT(-1)

END SELECT

ENDSUBROUTINE getCorrStructure

SUBROUTINE getCorrType(corrStructure)
CHARACTER (5) :: corrStructure

IF (corrStructure == "AR1") THEN
corrType = AR1
ELSEIF (corrStructure == "MA") THEN
corrType = MA
ELSEIF (corrStructure == "EX") THEN
corrType = EX
ELSEIF (corrStructure == "Ind") THEN
corrType = IND
ELSE
print *, "Correlation Structure ", corrStructure, "not valid."
CALL EXIT()

```

```

ENDIF
ENDSUBROUTINE getCorrType

END MODULE correlationRelated

MODULE Bahadur

USE commonData
USE correlationRelated

IMPLICIT NONE

CONTAINS

! joint prob of y1,...,ys
REAL FUNCTION BahadurJoint(s, rho)
INTEGER, INTENT (IN) :: s
INTEGER :: j, m
REAL :: sum, product, rho, C(T,T)

CALL getCorrStructure(rho, C)
sum = 0
product = 1
DO j = 1, s
DO m = (j+1), s
sum = sum + C(j,m) * (y(method,i,j)-mu(i,j))*(y(method,i,m)-mu(i,m)) /
(sqrt(mu(i,j)*(1-mu(i,j)))*sqrt(mu(i,m)*(1-mu(i,m))))
ENDDO
product = product * mu(i,j)**y(method,i,j) *
(1-mu(i,j))**(1-y(method,i,j))
ENDDO
BahadurJoint = product*(1 + sum)
ENDFUNCTION BahadurJoint

! conditional prob of y(s+1) | y1,...,ys
REAL FUNCTION BahadurCondProbGiven_s(s, rho)
INTEGER, INTENT (IN) :: s
INTEGER :: temp
REAL jointT, rho

temp = y(method,i, s+1)
y(method,i, s+1) = 1
jointT = BahadurJoint(s, rho)

```

```

IF(jointT==0) THEN
BahadurCondProbGiven_s = 2.0
ELSE
BahadurCondProbGiven_s = BahadurJoint(s+1, rho)/jointT
ENDIF
y(method,i, s+1) = temp
ENDFUNCTION BahadurCondProbGiven_s

SUBROUTINE generateBahadur()
INTEGER :: j

DO j=1,T
CALL RNBIN (1, 1, BahadurCondProbGiven_s(j-1, rho(i)), y(method,i,j))
ENDDO
ENDSUBROUTINE generateBahadur

LOGICAL FUNCTION validBahadurProb_s(s, rho)
INTEGER, INTENT(IN) :: s
REAL :: probability, rho
INTEGER r

validBahadurProb_s = .TRUE.
DO r = 1, 2**s
DO j = 1, s
y(method,i,j) = MOD(r/2**(j-1), 2)
ENDDO
probability = BahadurJoint(s, rho)
validBahadurProb_s = validBahadurProb_s*validProbability(probability)
IF (s.NE.1) THEN
probability = BahadurCondProbGiven_s(s-1, rho)
validBahadurProb_s = validBahadurProb_s*validProbability(probability)
ENDIF
ENDDO
ENDFUNCTION validBahadurProb_s

LOGICAL FUNCTION validBahadur(rho)
REAL :: rho
INTEGER s

validBahadur = .TRUE.
DO s = 1, T
validBahadur = validBahadur*validBahadurProb_s(s, rho)
ENDDO

```

```

ENDFUNCTION validBahadur

END MODULE Bahadur

MODULE Qaqish

USE commonData

IMPLICIT NONE

CONTAINS

SUBROUTINE setb(b, rho, successful)
  REAL :: b(T,T), a, rho
  LOGICAL :: successful
  INTEGER :: j, m

  SELECT CASE (corrType)
    CASE (AR1)
      b = 0
      DO m = 2, T
        b(m,m-1) = rho
      ENDDO

    CASE (MA)
      if(abs(rho)<0.5) THEN

        IF(rho==0.0) THEN
          b = 0
        ELSE

          a = (sqrt(1-4*rho**2)-1)/(2*rho)

          DO m = 1, T
            DO j = 1, (m-1)
              b(m,j) = (a**j-a**(-j))/(a**(-m)-a**m)
            ENDDO
          ENDDO
        ENDIF
      ELSE
        successful = .FALSE.
      RETURN
    
```



```

ENDIF

CASE (EX)

  if(rho>(-1.0/(T-1.0)) .AND. rho<1) THEN
    DO m =1,T
      b(m,:) = rho/(1+(m-2)*rho)
    ENDDO
  ELSE
    successful = .FALSE.
    RETURN
  ENDIF

CASE DEFAULT
  print *, "DEFAULTED"

END SELECT

successful = .TRUE.

ENDSUBROUTINE setb

REAL FUNCTION getLambda(b, m)
  INTEGER :: j, m
  REAL :: b(T,T)

  getLambda = mu(i,m)
  DO j =1,(m-1)
    getLambda = getLambda + b(m,j)*(y(method,i,j)-mu(i,j))
  ENDDO
ENDFUNCTION getLambda

LOGICAL FUNCTION validQaqishProb_s(s, b)
  INTEGER, INTENT(IN) :: s
  REAL, INTENT(IN) :: b(T,T)
  REAL :: lambda
  INTEGER r

  validQaqishProb_s = .TRUE.
  DO r = 1, 2**s
    DO j = 1, s
      y(method,i,j) = MOD(r/2**(j-1), 2)
    
```

```

ENDDO
IF (s.NE.1) THEN
  lambda = getLambda(b, s)
  validQaqishProb_s = validQaqishProb_s*validProbability(lambda)
ENDIF
ENDDO
ENDFUNCTION validQaqishProb_s

LOGICAL FUNCTION validQaqish(rho)
REAL :: b(T,T), rho
INTEGER s

  validQaqish = .TRUE.
  CALL setb(b, rho, validQaqish)
  DO s = 1, T
    validQaqish = validQaqish*validQaqishProb_s(s, b)
  ENDDO

ENDFUNCTION validQaqish

SUBROUTINE generateQaqish()
INTEGER :: j
REAL :: b(T,T)
LOGICAL :: successful

  CALL setb(b, rho(i), successful)

  CALL RNBIN (1, 1, mu(i,1), y(method,i,1))
  DO j=2,T
    CALL RNBIN (1, 1, getLambda(b, j), y(method,i,j))
  ENDDO
ENDSUBROUTINE generateQaqish

END MODULE Qaqish

MODULE Kanter

USE commonData

IMPLICIT NONE

```

CONTAINS

```

SUBROUTINE generateARKanter()
  INTEGER :: s(T-1), d(T-1), j

  CALL RNBIN (1, 1, mu(i,1), y(method,i,1) )

  IF (1-2*mu(i,1)*(1-rho(i)) .ne. 0) THEN
    gama(i) = rho(i)/(1-2*mu(i,1)*(1-rho(i)))
    CALL RNBIN ((T-1), 1, gama(i), s)
  ELSE
    s = 0
  ENDIF

  CALL RNBIN ((T-1), 1, mu(i,1)*(1-gama(i))/(1-2*gama(i)*mu(i,1)), d)

  DO j = 1, (T-1)
    y(method,i,j+1) = s(j)*MOD(y(method,i,j)+d(j),2) + (1-s(j))*d(j)
  ENDDO
ENDSUBROUTINE generateARKanter

SUBROUTINE generateMAKanter()
  INTEGER :: s(T+1), d(T+1), j

  IF ( gama(i) .ne. 0 ) THEN
    CALL RNBIN (T+1, 1, gama(i), s)
    CALL RNBIN (T+1, 1,
      (gama(i)+1-sqrt((gama(i)+1)**2-8*gama(i)*mu(i,1)))/(4*gama(i)),d)
  ELSE
    s = 0
  CALL RNBIN (T+1, 1, mu(i,1), d)
  ENDIF

  DO j = 1, T
    y(method,i,j) = s(j+1)*MOD(d(j)+d(j+1),2) + (1-s(j+1))*d(j+1)
  ENDDO
ENDSUBROUTINE generateMAKanter

SUBROUTINE generateEXKanter()
  INTEGER :: s(T), d(T), j
  INTEGER :: yi0(1)
  REAL :: mu_star, p0, p(T), q(T)

```

```

mu_star = 0.5
DO j = 1,T
mu_star = min(mu_star, 0.5 - abs(0.5-mu(i,j)))
ENDDO
p0 = mu_star/(rho(i)*(1-mu_star)+mu_star) - 0.000001

DO j = 1,T
IF (mu(i,1)<0.5) THEN
p(j) = -sqrt(rho(i)*p0*mu(i,j)*(1-mu(i,j))/(1-p0)) + mu(i,j)
ELSE
p(j) = sqrt(rho(i)*p0*mu(i,j)*(1-mu(i,j))/(1-p0)) + mu(i,j)
ENDIF
q(j) = (mu(i,j)-p(j))/(p0*(1-2*p(j)))
CALL RNBIN (1, 1, p(j), d(j))
CALL RNBIN (1, 1, q(j), s(j))
ENDDO

CALL RNBIN (1, 1, p0, yi0)

DO j = 1, T
  y(method,i,j) = s(j)*MOD(yi0(1)+d(j),2) + (1-s(j))*d(j)
ENDDO
ENDSUBROUTINE generateEXKanter

SUBROUTINE generateKanter()
SELECT CASE (corrType)
CASE (AR1)
CALL generateARKanter()
CASE (MA)
CALL generateMAKanter()
CASE (EX)
CALL generateEXKanter()
END SELECT
ENDSUBROUTINE generateKanter

SUBROUTINE rhoRestrictionMAKanter()
REAL g
  INTEGER NDEG
PARAMETER (NDEG=2)
REAL COEFF(NDEG+1)
COMPLEX ZERO(NDEG)
EXTERNAL WRCRN, ZPLRC

```

```

COEFF(1) = mu(i,1)*rho(i)*(1-mu(i,1))-mu(i,1)*(1-2*mu(i,1))
COEFF(2) = (1+mu(i,1))*(1-2*mu(i,1)) + 2*rho(i)*mu(i,1)*(mu(i,1)-1)
COEFF(3) = 2*mu(i,1)-1

IF (COEFF(1)==0 .AND. COEFF(2)==0 .AND. COEFF(3)==0) THEN
  gama(i) = 0
ELSE
  CALL ZPLRC (NDEG, COEFF, ZERO)
  gama(i) = (REAL(ZERO(1))-mu(i,1))/(REAL(ZERO(1))*(2*REAL(ZERO(1))-1))
ENDIF

IF (mu(i,1)<=0.5) THEN
  rhoRestriction(method,i,1) = 0
  rhoRestriction(method,i,2) = (1-2*mu(i,1))/(2*(1-mu(i,1)))
  IF (mu(i,1)==0.5) THEN
    rhoRestriction(method,i,2) = 0
  ENDIF
ELSE
  g=(-2*mu(i,1)+2*sqrt(2*mu(i,1)-1))/(2*(mu(i,1)-1))
  rhoRestriction(method,i,1) = ((2*mu(i,1)-1)*(1-g)*
    (1+g-sqrt((g+1)**2-8*mu(i,1)*g))+4*mu(i,1)*g-8*mu(i,1)**2*g) /
    (8*mu(i,1)*g*(1-mu(i,1)))
  rhoRestriction(method,i,2) = 0
ENDIF
ENDSUBROUTINE rhoRestrictionMAKanter

LOGICAL FUNCTION validKanter(rho)
REAL :: rho
SELECT CASE (corrType)
CASE (AR1)
  validKanter = validARKanter(rho)
CASE (MA)
  validKanter = validMAKanter(rho)
CASE (EX)
  validKanter = validEXKanter(rho)
END SELECT
ENDFUNCTION validKanter

LOGICAL FUNCTION validARKanter(rho)
REAL :: s(T), d(T)
REAL :: rho
validARKanter = .TRUE.

```

```

IF (1-2*mu(i,1)*(1-rho) .ne. 0) THEN
s(1) = rho/(1-2*mu(i,1)*(1-rho))
d(1) = mu(i,1)*(1-s(1))/(1-2*s(1)*mu(i,1))
ENDIF

IF (s(1)<0 .OR. s(1)>1) THEN
validARKanter = .FALSE.
ENDIF
IF (d(1)<0 .OR. d(1)>1) THEN
validARKanter = .FALSE.
ENDIF
ENDFUNCTION validARKanter

LOGICAL FUNCTION validMAKanter(rho)
REAL :: s(T), d(T)
REAL :: rho
validMAKanter = .TRUE.
ENDFUNCTION validMAKanter

LOGICAL FUNCTION validEXKanter(rho)
REAL :: s(T), d(T), p, mu_star
REAL :: rho
validEXKanter = .TRUE.

IF (0 > rho) THEN
validEXKanter = .FALSE.
ELSE
mu_star = 0.5
DO j = 1,T
mu_star = min(mu_star, 0.5 - abs(0.5-mu(i,j)))
ENDDO
p = mu_star/(rho*(1-mu_star)+mu_star) - 0.0000001

DO j = 1,T
IF (mu(i,1)<0.5) THEN
d(j) = -sqrt(rho*p*mu(i,j)*(1-mu(i,j))/(1-p)) + mu(i,j)
ELSE
d(j) = sqrt(rho*p*mu(i,j)*(1-mu(i,j))/(1-p)) + mu(i,j)
ENDIF
s(j) = (mu(i,j)-d(j))/(p*(1-2*d(j)))
IF (s(j)<0.0 .OR. s(j) >1.0) THEN
validEXKanter = .FALSE.

```

```
ENDIF
IF (d(j)<0 .OR. d(j) >1) THEN
validEXKanter = .FALSE.
ENDIF
ENDDO
ENDIF
ENDFUNCTION validEXKanter

END MODULE Kanter
```