View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by D-Scholarship@Pitt

RATIONALE FOR CHOOSING AN EXPLICIT CORRELATION STRUCTURE IN A MULTILEVEL ANALYSIS WITH BIVARIATE OUTCOME

by

Folefac Desire' Atem

B.Sc Physics, University of Buea, Cameroon, 2002

M.S Statistics, Wright State University, 2006

Submitted to the Graduate Faculty of

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2010

UNIVERSITY OF PITTSBURGH
Graduate School of Public Health

This dissertation was presented

by

Folefac Desire' Atem

It was defended on
July 28, 2010
and approved by

Dissertation Advisor:
Stewart Anderson, PhD
Professor
Department of Biostatistics
Graduate School of Public Health
University of Pittsburgh

Howard E. Rockette Jr, PhD
Professor
Department of Biostatistics
Graduate School of Public Health
University of Pittsburgh

Ravi Sharma, PhD
Professor
Department of Behavioral and Community Health Sciences
Graduate School of Public Health
University of Pittsburgh

Abdus Wahed,PhD
Associate Professor
Department of Biostatistics
Graduate School of Public Health
University of Pittsburgh

Copyright © by Folefac Desire' Atem

2010

Stewart Anderson, PhD

RATIONALE FOR CHOOSING AN EXPLICIT CORRELATION STRUCTURE IN A MULTILEVEL ANALYSIS WITH BIVARIATE OUTCOME

Folefac Desire' Atem, Ph.D

University of Pittsburgh, 2010

The analysis of multileveled data with bivariate outcomes is very common in the fields of education, health economics and health service research. Modeling bivariate outcomes is very useful in HIV research where the joint evolution of HIV RNA and CD4+t lymphocytes in a cohort of HIV-1 infected patient treated with active antiretroviral treatment. The use of the MIXED model method and the Generalized Estimating Equations (GEE) are the most influential recent developments in statistical practice analysis techniques used in analyzing such data. The linear mixed model takes into account all available information and accounts for both serial and cross correlation. The efficiency of the model depends on the correlation structure. Our simulations studies reveal that for smaller clusters the independent and the unstructured are highly favored while for larger clusters the independent models yields estimates with the least standard errors. Additionally, we looked at cases where the data is clustered but not longitudinal. In these cases, the compound symmetry model performed best. Furthermore, our results show that in some cases, the unstructured correlation model tend to have the smallest AICC and BIC but its estimates do not always produce estimates with the smallest standard errors. In this dissertation we formulated a rationale in choosing an explicit working correlation structures for modeling multilevel data with bivariate outcomes. We also simulated different types of data with bivariate outcomes with missingness. To guide our strategy the model selection strategies were based on optimizing AIC, CAIC, AICC BIC and standard error of estimates.

Our model has particular public health importance in clinical trials where the clinician may be interested in the joint evolution HIV RNA and CD4+t lymphocytes in a cohort of HIV-1 infected patients treated with active antiretroviral drugs.

TABLES OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

Many thanks my advisor, Dr. Anderson, as well as my committee members, Drs. Rockette, Sharma, and Wahed. I greatly appreciate all of your help, encouragement, and guidance throughout this program. I also wish to give a special thanks to my family especially to my late mother whose constant support and appreciation has made us who we are. Particular thanks are due to Emmanuel , Robert, Serge, Caroline,Christopher and Ameche who provided me with encouragement and spirit during my difficult days. I would also like to acknowledge the constant help and support of Francis Pike, The Minh Luong, Jesse YenChih Hsu, Abidemi K. Adeniji and Emmanuel Sampene.

1.0 CHAPTER ONE

1.1 Introduction

A common assumption often made in classical statistic procedures is that observations are independent and identically distributed (i.i.d). In linear models, we don't always assume that data come from identical distributions, because observations may, for example, differ in expected values. However often investigators continue to insist the error disturbance terms are independent and identically distributed. Multilevel regression models, referred to in the research literature under different names, such as 'random coefficient models' (Rubin,Klein,Wald and Theil late 1940s, Rao 1965, de Leeuw & Kreft, 1986; Longford, 1993), 'variance component models' (Longford, 1987), 'hierarchical linear models' (Raudenbush & Bryke, 1986, 1987, 1992) are constructed with the notion that we cannot assume i.i.d for the entire population despite we may for cluster within population. It is this marriage of the contextual analysis and the traditional statistical mixed model theory that gave rise to multilevel analysis, which is known as hierarchical linear model. This class of models has many applications in survey data; repeated measures; twin studies; meta analysis and multivariate data analysis. However, not much has been done in cases where the outcome variable is bivariate. For example, in most HIV studies, several markers are available to measure the quantity of virus (CD4+T lymphocytes which are a specific target of virus, CD8 + T lymphocytes) or the inflammation process (β_2 microglobuline). Another example is when pharmaceutical company introduces a new medication for high blood pressure and hence may be interested in looking at the evolution of both the systolic and diastolic readings over time (bivariate outcomes).

Several authors such as Shay et al, used the EM algorithm to fit bivariate random effects model, Sy et al used the Fisher scoring method to fit bivariate linear random effects model including an integrated Orstein – Uhlenbeck process(IOU). However, in most cases, they fail to give a coherent rationale of the choices of their working correlation structure. In this dissertation, we will use a model based on themixed model approach, then develop an algorithm in order to come up with the best correlation structure for different types of multilevel data with bivariate outcomes. It is attractive to choose the working correlation that best fit the data since the use of the optimal working correlation increases the efficiency of our estimators. We understand, however, that the selection of the correlation structure is challenging in GEE, since there is no likelihood function and consequently likelihood ratio tests cannot be applied. However that is not the case with mixed model. We hypothesize that the working correlation structure of our JMRE model will be independent of the correlation structure of our independent random effect model; independent of our sample size but will be highly dependent on cluster size (Bell et el 2008) and on the cross correlation (Fort et el 2003).

2.0 CHAPTER TWO

2.1 Statistical models used in multilevel analysis

In this section, we shall discuss some statistical models used for multilevel data. Our primary focus will be on multilevel analysis but we shall discuss other methods like MANOVA and repeated measures ANOVA and models involving generalized estimation Equations (GEE). In this chapter, we also introduce the multilevel model for the linear case, with two levels. In subsequent chapters, we shall discuss the bivariate cases. We may give a brief discussion of the non linear case.

2.2 Models

A statistical model may be defined as a functional relationship between random variables, where the observed data are said to be a realization of these random variables. The underlying distribution of these random variables is often partly specified because we merely affirm that the distribution belongs to some parametric family. This assertion is stronger if our assumed distribution is part of the exponential family. In both cases, the model is also only partly specified and one of the standard statistical tasks is to estimate the values of the unknown parameters.

2.2.1 MANOVA and Repeated-Measure ANOVA

The MANOVA approach is similar to ANOVA , where (M for Multivariate) can be estimated with corrected matrix of sums of squares and cross products. This model can be used to simultaneously test hypotheses about the mean effects of multiple outcomes by using one of the following multivariate tests: Roy's largest root (Morrison, 1976), Wilks's Lambda, Pillai's trace or Hotelling-Lawley trace. The MANOVA approach treats the vector of measurements on each subject as a multivariate observation and uses multivariate tests to assess the effects of covariates.

The MANOVA model can be represented as a linear function of the covariates X, see equation 2.7 below.

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{2.1}$$

where

Y is an $n \times p$ matrix representing the p measurements of n subjects in the study;

X is an $n \times k$ design matrix representing the values of k predictor variables for the n subjects;

$\boldsymbol{\beta}$ is a $k \times p$ matrix of coefficients ; and

$\boldsymbol{\varepsilon}$ is an $n \times p$ matrix of random errors, with $\boldsymbol{\varepsilon}_i \sim \mathbf{N}(\mathbf{0}, \boldsymbol{\Sigma})$.

The $p \times p$ matrix, , $\boldsymbol{\Sigma}$, is of the form:

$$\begin{pmatrix} \sigma_{11}^2 & \sigma_{21} \cdots & \sigma_{p1} \\ \sigma_{21} & \sigma_{22}^2 \cdots & \sigma_{p2} \\ \vdots & \vdots & \vdots \end{pmatrix},$$

The variance covariance structure above is known as the unstructured covariance structure as no structure is imposed on its elements. However, the MANOVA model assumes the variance –covariance matrix Σ is the same for all n subjects.

In equation (2.1), the ith-rows of **Y** represent the vector and the jth-column for all the n measurements. The columns of the design matrix, **X,** represent the values of the independent variables for the n subjects. They might be indicators of baseline predictors or group membership. For instance, if we are studying the effect of a drug, age and gender on a dependent variable, the **X** matrix would have three columns, the first indicating drug (medication or placebo), the second indicating ages and the third one indicating the gender of the n subjects. In MANOVA models, one major limitation is that the values of the predictors are not allowed to vary over time.

The repeated measure ANOVA is one of the oldest and the simplest method for analyzing repeated measures data because it expresses the effect of treatment in terms of the familiar ANOVA model and uses standard F test to assess their significance. The analysis method is by decomposing the total variance into the following components: a) between-subject factors, b) the within subject factor c) their interaction and d) random error. The model with a single classification factor may be represented as shown below;

$$Y_{ijk} = \mu + \tau_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk}, \tag{2.2}$$

where

Y_{ijk} is the response of subject i at time j with level of predictor k;

μ is grand mean for all observations;

τ_i is the effect of level j of the within-subjects factor (time);

β_j is the effect of the between-subjects covariate;

$\gamma_{ij} \sim N(0, \sum_{\gamma})$ is the random interaction term between the repeated factor and the covariates;

ε_{ijk} is the random error associated with subject i, who is at factor level $k(i)$, at time j.

$\varepsilon_{ijk} \sim N(0, \Sigma)$, with

$$\Sigma = \begin{pmatrix} \sigma^2 & 0\ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma^2 \end{pmatrix}$$

The variance-covariance matrix, Σ, represents the covariance between pairs of repeated measurements, where the structure satisfies the sphericity condition under the null hypothesis. But more obviously, sphericity implies $\rho_{ij} = \rho$ (constant) where ρ denotes correlation for all $i \neq j$. The sphericity condition asserts that the variance of all contrast is zero. Under the null hypothesis, for the F-statistics to have an F-distributions, sphericity is a necessary and sufficient condition.

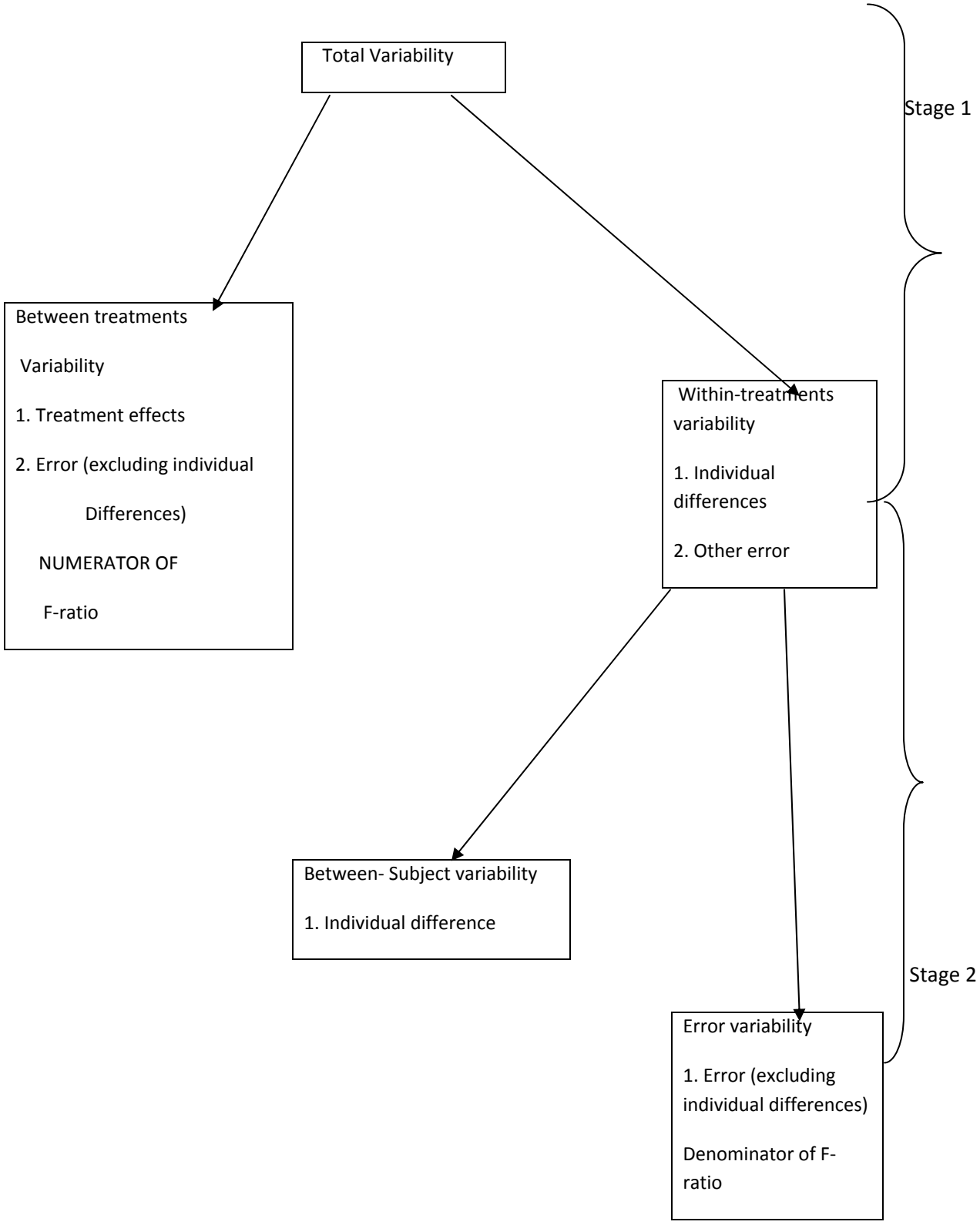

Figure 2.1.Picture Depicting Repeated MeasureAanalysis using ANOVA.

However, a sufficient but not necessary condition for sphericity is compound symmetry. One can assess sphericity using the Mauchly's test (Anderson, 1958). When the sphericity assumption is violated, the type one error rate of the ANOVA F test is generally inflated, that is, the probability of falsely rejecting the null hypothesis is higher than the selected significance level. Greenhouse and Geisser (1959) and Huynh and Fedt (1976) proposed correction factors that can be applied to the numerator and denominator degrees of freedom for the F-distribution used to make inferences.

2.2.2 Generalized Estimating Equations (GEE)

The GEE is an estimating equating procedure, which is used to estimate parameter of a the marginal model specifies only the conditional mean model is as:

$\boldsymbol{\mu}_i = \mathbf{E}(\mathbf{Y}_i \mid \mathbf{X}_i)$ but treats the parameters in Vi as nuisance parameters. The marginal expectation $\boldsymbol{\mu}_i$ depends on the covariates, $\mathbf{X}_i$, through a known link function $\mathbf{g}(\boldsymbol{\mu}_i) = \boldsymbol{\eta}_i = \mathbf{X}_i'\boldsymbol{\beta}$. A vital aspect of this model is that the mean response and the within-subject association are modeled separately. Hence, the model parameters beta have a population average interpretation. The mean and variance usually follows a distributional form a distribution within the exponential family. If the mean is correctly specified, then the method of GEE yields consistent estimator .

Hence, the model parameters beta have a population average interpretation. The mean and variance usually follows a distributional form a distribution within the exponential family. If the mean is correctly specified, then the method of GEE yields consistent estimator $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ by solving the equation below;

$$\sum_{i=1}^{n} \mathbf{D}_i' \mathbf{V}_i^{-1} (\mathbf{Y}_i - \boldsymbol{\mu}_i) = \mathbf{0}; \qquad (2.3)$$

With $\mathbf{D_i[n,p]}$ matrix of derivatives of μ_i with respect to β. For large samples, $Var(\boldsymbol{\beta})$ are robust to misspecifications of $\mathbf{V_i}$ and the underlying distributions of $\mathbf{(Y_i, X_i)}$.

2.2.3 Mixed Models

De Leeuw and Kreft (1986) introduce the terminology 'Mixed model'. The mixed effect models are statistical model in the regression analysis where it is assumed that some of the coefficients are fixed and other random. The mixed linear model or MLM is written as

$$\mathbf{Y = X\beta + Z\delta + \varepsilon} \tag{2.4}$$

With **X[n,r]**, **Z[n,p]** and $\begin{pmatrix} \boldsymbol{\varepsilon} \\ \boldsymbol{\delta} \end{pmatrix} \sim \left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega} \end{pmatrix} \right)$, we assumed that **X** and **Z** are full column rank.

However, in some cases it might be useful and intuitive writing equation 2.4 in scalar notation. This might be useful for those who have not yet have a course in matrix algebra. Therefore, equation 2.4 becomes;

$$y_i = \sum_{q=1}^{r} x_{iq}\beta_q + \sum_{s=1}^{p} z_{is}\delta_s + \varepsilon_i \tag{2.4a}$$

Equation 2.4 is known as a mixed model because the regression equation has both the fixed regression coefficient and a random regression component. Hence;

$$\mathbf{Y \sim N(XB, V)},$$

where $\mathbf{V = Z\,\Omega\,Z' + \Sigma}$

(2.5)

Equation can easily be derived by assuming that $\mathbf{var(X\beta) = 0}$, since $\boldsymbol{\beta}$ described the fixed effect parameter, also, Z is a matrix of constant. Therefore

$$\mathbf{V = Zvar(\delta)Z' + var(e)}.$$

We let $\mathbf{\Omega}$ denote $\mathbf{var(\delta)}$, also, since the random effect are assumed to follow normal distributions, we may write $\mathbf{\delta \sim N(0, \Omega)}$. Furthermore we may write $\mathbf{var(e) = \Sigma}$, the residual covariance matrix and $\mathbf{e} \sim N(\mathbf{0, \Omega})$. Therefore

$\mathbf{V = Z\Omega Z' + \Sigma}$ as required.

We may see from the above equation that, the effects of the predictors in Z are shifted from the expected values to the dispersions of the normal distribution.

The form of the dispersion matrix for the residual in (2.4) is suggestive of the common factor analysis model and this similarity would be used in extending the bivariate multilevel models to covariance structure and latent variable models.

It is suitable to parameterize both dispersion matrices $\mathbf{\Sigma}$ and $\mathbf{\Omega}$ using vectors of parameters σ and ω. We assume $\mathbf{\Sigma = \sigma^2 I}$, that is we assume the noises ε are homoskedastics (i.e. the condition that the error variance is constant over all cases in contrast to the condition of unequal error variance called heteroscedaciticity). This guarantees that if there are no random effects, then we may easily recover our classical linear model. We may also parameterize $\mathbf{\Omega}$ as linear combination of known matrices $\mathbf{C_l}$. Thus

$$\mathbf{\Omega = \omega_1 C_1 + ... + \omega_l C_l = \sum_{l=1}^{L} \omega_l C_l} \quad ,$$

(2.6)

Therefore, V also has a linear structure

$$\mathbf{V} = \boldsymbol{\omega}_1 \mathbf{Z}\mathbf{C}_1\mathbf{Z}' + \ldots + \boldsymbol{\omega}_L \mathbf{Z}\mathbf{C}_L\mathbf{Z}' + \boldsymbol{\sigma}^2\mathbf{I} = \sum_{l=1}^{L} \boldsymbol{\omega}_l \mathbf{C}_l + \boldsymbol{\sigma}^2\mathbf{I}$$

Figure 1, below shows the two levels MLM which explicitly takes into consideration the group structure into account. The two levels MLM explicit representation is shown below;

$$\mathbf{Y}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\boldsymbol{\delta}_j + \boldsymbol{\varepsilon}_j, \tag{2.6a}$$

Where $j = 1, \ldots, m$, and

$$\begin{pmatrix} \boldsymbol{\varepsilon}_j \\ \boldsymbol{\delta}_j \end{pmatrix} \sim \mathbf{N}\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_j & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}_j \end{pmatrix} \right). \tag{2.6b}$$

Using $\perp$ for independence,

$$(\boldsymbol{\varepsilon}_j, \boldsymbol{\delta}_j) \perp (\boldsymbol{\varepsilon}_l, \boldsymbol{\delta}_l) \tag{2.6c}$$

$\forall j \neq l.$

Similarly, we may assume that $\boldsymbol{\Sigma}_j = \boldsymbol{\sigma}_j^2\mathbf{I}$ and also $\boldsymbol{\Omega}_j = \boldsymbol{\Omega}.$

Therefore,

$$\mathbf{Y}_j \sim \mathbf{N}(\mathbf{X}_j\boldsymbol{\beta}, \mathbf{V}_j),$$

where

$\mathbf{V}_j = \mathbf{Z}_j\boldsymbol{\Omega}\mathbf{Z}'_j + \boldsymbol{\sigma}_j{}^2\mathbf{I},$ the $\mathbf{Y}_j$ are independent for different j, and $\mathbf{X}_j$ and $\mathbf{Z}_j$ are full column ranks. However this variance structure might be quite restrictive in the case where we have many small groups. However, in many multilevel analyses, it is assumed that all $\sigma^2{}_j = \sigma^2 \forall j$. This

assumption is often not true, so we will usually assume different variances. Furthermore, we will view $\sigma_j^2 = \sigma^2$ as no between-groups variation and all σ_j^2 treated as separate parameters in fixed effects specification.

2.2.4 Random Coefficient Models

The random coefficient model was introduced by Rao (1965). de Leeuw and Kreft(1986) introduced the random coefficient model into the multilevel literature. Longford (1993), did some work to improve on this model to accommodate cases with unbalanced data. The random coefficient model is very useful when: a) the groups are regarded as unique quantities and the researcher wishes to draw conclusions pertaining to each of these N specific groups; b) when the groups are regarded as samples from a real (hypothetical) populations and the researcher wishes to draw conclusions pertaining to these populations; and c) when the researcher wishes to test the effects of group level variable. For more on the use of random coefficient models, see Seale et al (1992). The random coefficient model (RCM) may be written as follows:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$
$$\boldsymbol{\beta} = \boldsymbol{\beta} + \boldsymbol{\delta},$$

where $$\begin{pmatrix} \boldsymbol{\varepsilon} \\ \boldsymbol{\delta} \end{pmatrix} \sim \left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega} \end{pmatrix} \right).$$

Hence, in RCM we have

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{X}\boldsymbol{\delta} + \boldsymbol{\varepsilon}.$$

Therefore the RCM is an MLM in which **Z=X**. However, the RCM in this form is not very useful, because without additional assumptions, it is not known. Nonetheless, we gave it this form to introduce the notion of random coefficients model (Rubin et el (1940s), Rao (1965), Leewu and Kraft (1986) and Longford (1993)). Furthermore, the two level RCM is given by;

$$\mathbf{Y_j = X_j\beta_j + \varepsilon_j,} \tag{2.7a}$$

$$\boldsymbol{\beta_j = \beta + \delta_j} \tag{2.7b}$$

The distribution assumptions are similar to those for the two-level MLM.

In scalar notation, our two level RCM may be easily understood and clearer, it is written as;

$$y_{ij} = x_{ij1}\beta_{j1} + ... + x_{ijp}\beta_{jp} + \varepsilon_{ij},$$
$$\beta_{js} = \beta_{js} + \delta_{js}.$$

As a consequent of the RCM, we have the random intercept model (RIM). This is the same as the RCM but for the fact that we assume that all regression coefficients that are not intercepts have no random component. Therefore, all slopes are fixed; hence the two level random intercept models;

$$\mathbf{y_j = \lambda_j 1_{nj} + X_j\beta + \varepsilon}_j,$$
$$\boldsymbol{\lambda_j = \lambda + \delta}_j.$$

2.2.5 Slopes as Outcomes Models

The slopes as outcome model (SOM) is the backbone of multilevel methodology. But in recent statistical literature , multileveled models are referred to more generally as Hierarchical Linear models. The SOM can be written as:

$$\mathbf{Y_j} = \mathbf{X_j}\boldsymbol{\beta} + \boldsymbol{\varepsilon_j},$$
$$\boldsymbol{\beta}_j = \mathbf{Z_j}\boldsymbol{\lambda} + \boldsymbol{\delta_j}, \qquad (2.8)$$

with **X**[n,p], **Z**[p,r] being fixed design matrices, and

$$\begin{pmatrix} \boldsymbol{\varepsilon_j} \\ \boldsymbol{\delta_j} \end{pmatrix} \sim \left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma_j} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega_j} \end{pmatrix} \right).$$

The uniqueness of this model as compared to the previous models is that the random coefficient $\boldsymbol{\beta}$ are themselves dependent variable in the second regression equation. The general form of the SOM is shown below:

$$\mathbf{Y} = \mathbf{WZ}\boldsymbol{\lambda} + \mathbf{X}\boldsymbol{\delta} + \boldsymbol{\varepsilon}. \qquad (2.9)$$

This shows that the SOM is an MLM in which the fixed regressors are **X=WZ** and random regressors are X.

The two level SOM is;

$$\mathbf{Y_j} = \mathbf{X_j}\boldsymbol{\beta} + \boldsymbol{\varepsilon_j},$$
$$\boldsymbol{\beta}_j = \mathbf{Z_j}\boldsymbol{\lambda} + \boldsymbol{\delta_j},$$

With **X**[n,p], **Z**[p,r] being fixed design matrices, and

$$\begin{pmatrix} \boldsymbol{\varepsilon_j} \\ \boldsymbol{\delta_j} \end{pmatrix} \sim \left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega} \end{pmatrix} \right)$$

There are also other methods of handling data with more than one level.

2.2.6 Missing Data

One of the fundamental differences between the methods described above is how they handle missing data.(Let M be a missing data indicator that take values j if $\mathbf{Y_1, ..., Y_{j-1}}$ are observed and $\mathbf{Y_j, ..., Y_k}$ are missing). The missing data mechanism is the missing completely at random (MCAR) if

$$\Pr(\mathbf{M_i} = j \mid y_{i1,...,} y_{ik}; \phi) = \phi \ \forall y_{i1}, ..., y_{ik} \tag{2.10}$$

where ϕ denotes unknown parameters.

The missing completely at random assumption is the strongest assumption. Equation 2.10 simply states that, missingness, does not depend to either observed or unobserved components of $\mathbf{Y_i}$ (Little and Rubin(1987)). GEE methods typically assume MCAR mechanism for dropouts. The dropout mechanism is missing at random (MAR) if the dropout depends on values recorded prior to dropout but not values after dropout,that is:

$$\Pr(\mathbf{M}_i = j \mid y_{ij}, ..., y_{ik}; \phi) = \Pr(\mathbf{M_i} = j \mid y_{i1}, ..., y_{i,j-1}, \phi), \forall y_{i1}, ..., y_{ik.} \tag{2.11}$$

All the different forms of the MLM discussed above assumed MAR in case of missing data. Another type of dropout mechanism is the not missing at random (NMAR). Here the dropout mechanism depends on both observed and unobserved obeservations (Little and Rubin(1987)). Both the MANOVA and the repeated- measures ANOVA are typically based on completely observed data. Observations with missing data are often dropped out in these analyses.

2.2.7 Non Linear Models.

As with the linear models discussed above which are extensions of generalized linear models, we may extend the linear mixed model to accommodate non-linear data. Non linear models are models that are fundamentally non linear in regression parameters (see eqn 2.13 below).

$$E(Y) = \beta_1 + e^{\beta_2 X} \qquad (2.13)$$

We simply condition on the random effects and assume a generalized linear model for the conditional distribution of the outcomes. In order to obtain the full model, we multiply the conditional density by the marginal density of the random effects and integrate over the random effects (similar to Bayesian methods). This integration is difficult since it is multidimensional and usually cannot be evaluated in closed form.

3.0 CHAPTER THREE

In the past decade, a large literature has arisen for analyzing univariate multilevel data (De Leeuw (1986), De Leeuw and. Kreft(1995) , De Leeuw(2005), Bryk,., Raudenbush, and Congdon, (1996)). However, there is much biomedical research in which we are interested in two outcomes. In this chapter,we introduce two ways of dealing with bivariate multilevel data. First, we introduce the Generalized Estimating Equation (GEE) approach to relate to each set of multilevel data to vital predictor variables. Then we introduce the mixed linear model approach. Furthermore, we will describe estimation and hypothesis testing issues. Finally, we will talk about missingness with two outcomes.

3.3.1 The GEE model for Bivariate outcomes

We may have a situation where our observation are clustered (not collected over time) and repeated measure design. Both use very similar approaches; hence we will only discuss the latter. In the repeated measures design, the observations might not be equally space interval time points for all individual enrolled for the study.

Hence, the bivariate repeated measures design is represented as follows:

$y_i^{(1)} = [y_{i1}^{(1)} \ldots y_{iT}^{(1)}]'$, $i = 1, \ldots, N$, denote the set of first observation for the ith individual, and

$y_i^{(2)} = [y_{i1}^{(2)} \ldots y_{iT}^{(2)}]'$, $i = 1, \ldots, N$, denote the set of second observation for the ith individual.

Following Liang and Zeger (1986) notation let

$\boldsymbol{\mu}_{\mathbf{i}}^{(.)} = E(y_i^{(.)})$,

$$\mathbf{h}_1(\boldsymbol{\mu}_{\mathbf{i}}^{(1)}) = \mathbf{X}_{\mathbf{i}}^{(1)}\boldsymbol{\beta}^{1} \text{ and } \mathbf{h}_2(\boldsymbol{\mu}_{\mathbf{i}}^{(2)}) = \mathbf{X}_{\mathbf{i}}^{(2)}\boldsymbol{\beta}^{2} \quad (3.1)$$

where $\mathbf{X}_{\mathbf{i}}^{(1)}$ is $T \times r$ matrix of non-stochastic predictor variables for the ith individual, $\boldsymbol{\beta}$ is corresponding vector of linear model parameters and h(.) is a known link function, most often a member

of exponential family of distribution. We assumed that there is no perfect correlation within either design variables (i.e, no singularities). Furthermore, **X** may consist of fixed and/or time-dependent covariates, with any appropriate degree of overlap between the two sets of independent variables.

For the GEE model, $\mathbf{v_{it}^{(1)} = var(y_{it}^{(1)}) = \varphi_1 . g_1(\mu_{it}^{(1)}),}$ where g(.) is a known variance function and φ is the dispersion parameter.

Letting $\mathbf{A_i^{(1)} = diag[g_1(\mu_{i1}^{(1)})...g_1(\mu_{iT}^{(1)})],}$ hence covariance matrix is given by

$$\mathbf{V_i^{(1)} = \varphi_1 (A_i^{(1)})^{1/2} R_{11} (A_i^{(1)})^{1/2},}$$

where $\mathbf{R_{11}}$ is the working correlation matrix among the repeated measures on the first outcome.

We may define a similar expression for $\mathbf{V_i^{(2)}}$.

$\mathbf{y}_i = \mathbf{[y_i^{(1)'} y_i^{(2)'}]'}$, the model may be written as

$$\mathbf{h(\mu_i) = X_i \beta},$$

where $\boldsymbol{\mu}_i = \begin{pmatrix} \boldsymbol{\mu}_i^{(1)} \\ \boldsymbol{\mu}_i^{(2)} \end{pmatrix}$, $\mathbf{X}_i = \begin{pmatrix} \mathbf{X}_i^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_i^{(2)} \end{pmatrix}$, $\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}^{(1)} \\ \boldsymbol{\beta}^{(2)} \end{pmatrix}$,

and h(.) is a compound function comprising of h_1 and h_2.

Therefore, the joint covariance matrix among the sets of repeated measures may written in the form

$$\mathbf{V_i} = \begin{pmatrix} \varphi_1 \mathbf{I_T} & \mathbf{0} \\ \mathbf{0} & \varphi_2 \mathbf{I_T} \end{pmatrix}^{1/2} \begin{pmatrix} \mathbf{A_i^{(1)}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_i^{(2)}} \end{pmatrix}^{1/2} \mathbf{R} \begin{pmatrix} \mathbf{A_i^{(1)}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_i^{(2)}} \end{pmatrix}^{1/2} \begin{pmatrix} \varphi_1 \mathbf{I_T} & \mathbf{0} \\ \mathbf{0} & \varphi_2 \mathbf{I_T} \end{pmatrix}^{1/2}$$

$$= \boldsymbol{\varphi}^{1/2} \mathbf{A_i^{1/2}} \mathbf{R} \mathbf{A_i^{1/2}} \boldsymbol{\varphi}^{1/2},.$$

where R the mechanism which takes into consideration the repeated measures between the two sets of observation is define as;

$\mathbf{R} = \mathbf{R}(\boldsymbol{\alpha}) = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{pmatrix}$ is the working correlation among the pair of repeated measures, each of whose elements is a function of the $(K \times 1)$ parameter, $\boldsymbol{\alpha}$.

3.3.2 Statistical Inferences

3.3.2.1 GEE Estimation

In GEE, the generalized least square estimate $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is found as a solution to the generalized estimating equation

$$\sum_{i=1}^{N} \mathbf{D}_i' \mathbf{V}_i^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_i) = \mathbf{0} \tag{3.2}$$

$$and\ \mathbf{D}_i = (\partial \boldsymbol{\mu}_i^{(1)} / \partial \boldsymbol{\beta}') = \begin{pmatrix} \partial \boldsymbol{\mu}_i^{(1)} / \partial \boldsymbol{\beta}^{(1)'} & 0 \\ 0 & \partial \boldsymbol{\mu}_i^{(2)} / \partial \boldsymbol{\beta}^{(2)'} \end{pmatrix}.$$

There is no closed-form solution to equation 3.2. However Liang and Zeger (1986) described scoring algorithm for iterating to solution for $\hat{\boldsymbol{\beta}}$. They also showed that, if $\mathbf{D}_i, \mathbf{V}_i$ and $\mathbf{y}_i - \boldsymbol{\mu}_i$ are evaluated at consistent estimators of $\boldsymbol{\alpha}$ and $\boldsymbol{\varphi}$, then $\hat{\boldsymbol{\beta}}$ is asymptotically normal and unbiased with the asymptotic covariance matrix given by

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}} = [\Sigma \mathbf{D}_i' \mathbf{V}_i^{-1} \mathbf{D}_i]^{-1} [\Sigma \mathbf{D}_i' \mathbf{V}_i^{-1} \mathbf{cov}(\mathbf{y}_i) \mathbf{V}_i^{-1} \mathbf{D}_i] [\Sigma \mathbf{D}_i' \mathbf{V}_i^{-1} \mathbf{D}_i]^{-1}.$$

$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}}$ can be estimated be evaluating these matrices at their GEE estimate and replacing $\mathrm{cov}(y_i)$ by $(\mathbf{y}_i - \boldsymbol{\mu}_i)(\mathbf{y}_i - \boldsymbol{\mu}_i)'$. Furthermore, $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}}$ is consistent even if $\mathbf{V}_i$ is specified incorrectly, that is, GEE estimator are designed to be robust to departures from the true correlation pattern. But, greater

efficiency will be realized by those correlation models closer to true correlation structure (Rochon 1996). However, in the next section, we would discuss the bivariate linear mixed model, which is not robust to departure from true correlation structure.

3.3.3 Mixed Linear Model for Bivariate Outcomes

3.3.3.1 Modeling Bivariate outcome

We generally define a bivariate mixed linear model including a random component, correlation process and independent error as follows:

Let $\mathbf{Y_i} = \begin{pmatrix} \mathbf{Y_i^{(1)}} \\ \mathbf{Y_i^{(2)}} \end{pmatrix}$, $i = 1, ..., N$, represent the set of first observation for the ith individual, with $\mathbf{Y_i^K}$ the n_i^k vector of measurements of the maker k(k=1.2).

Since, we assume the bivariate responses (example two biomarkers) are independent, similar to section 2.2.1, we may use the two following models;

$$\begin{aligned} \mathbf{Y_i^{(1)}} &= \mathbf{X_i^1\beta^1 + Z_i^1\delta_i^1 + W_i^1 + \varepsilon_i^1} \\ \mathbf{Y_i^{(2)}} &= \mathbf{X_i^2\beta^2 + Z_i^2\delta_i^2 + W_i^2 + \varepsilon_i^2} \end{aligned} \quad \text{with} \quad \begin{aligned} &\boldsymbol{\varepsilon}_\mathbf{i}^\mathbf{1} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\sigma}_{\mathbf{e}^1}^\mathbf{2}\mathbf{I}_{\mathbf{n_i}}) \quad && \boldsymbol{\varepsilon}_\mathbf{i}^\mathbf{2} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\sigma}_{\boldsymbol{\varepsilon}^2}^\mathbf{2}\mathbf{I}_{\mathbf{n_i}}) \\ &\mathbf{Y_i^{(1)}} \sim \mathrm{N}(\mathbf{0}, \mathbf{G^1}) \text{ and } && \mathbf{Y_i^{(2)}} \sim \mathrm{N}(\mathbf{0}, \mathbf{G^2}) \\ &\mathbf{W_i^1} \sim \mathrm{N}(\mathbf{0}, \mathbf{R_i^1}) && \mathbf{W_i^2} \sim \mathrm{N}(\mathbf{0}, \mathbf{R_i^2}) \end{aligned} \tag{3.3}$$

where

$\mathbf{X_i^k[n_i, p^k]}$ is the design matrix;

$\boldsymbol{\beta}^\mathbf{k}$ is a $\mathbf{p^k}$ vector of fixed effect;

$\mathbf{Z_i^k[n_i, q^k]}$ is a design matrix which is usually subset of $\mathbf{X_i^k}$;

$\boldsymbol{\delta}_\mathbf{i}^\mathbf{k}$ is a q^k vector of random effects with dimension $\mathrm{q^k} \leq \mathrm{p^k}$;

$\mathbf{W}_i^k$ is a vector of realization process which might take the form of known correlation structure.

Thus in order to model the above structure while taking into consideration the serial correlation between bivariate outcomes, we may use the following model;

$$\mathbf{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\boldsymbol{\delta}_i + \mathbf{W}_i + \boldsymbol{\varepsilon}_i \quad where \quad \begin{matrix} \boldsymbol{\varepsilon}_i \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_i) \\ \mathbf{W}_i \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_i) \\ \mathbf{Y}_i \sim \mathrm{N}(\mathbf{0}, \mathbf{G}) \end{matrix} \tag{3.4}$$

where

$$\mathbf{X}_i = \begin{pmatrix} \mathbf{X}_i^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_i^2 \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{pmatrix}, \mathbf{Z}_i = \begin{pmatrix} \mathbf{Z}_i^1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_i^2 \end{pmatrix}, \boldsymbol{\delta}_i = \begin{pmatrix} \boldsymbol{\delta}_i^1 \\ \boldsymbol{\delta}_i^2 \end{pmatrix}, \mathbf{W}_i = \begin{pmatrix} \mathbf{W}_i^1 \\ \mathbf{W}_i^2 \end{pmatrix} \text{ and } \boldsymbol{\varepsilon}_i = \begin{pmatrix} \boldsymbol{\varepsilon}_i^1 \\ \boldsymbol{\varepsilon}_i^2 \end{pmatrix}.$$

Equation 3.4 differs from 2.4 in that we have an additional term W_i is a realization of bivariate regressive process.

With the assumption that $\boldsymbol{\delta}_i$, $\mathbf{W}_i$ and $\boldsymbol{\varepsilon}_i$ are mutually independent, it follows that

$$\mathbf{var}(\mathbf{Y}_i) = \mathbf{V}_i = \mathbf{Z}_i\mathbf{G}_i\mathbf{Z}_i^{'} + \mathbf{R}_i + \boldsymbol{\Sigma}_i \tag{3.5}$$

3.3.4 Estimation of effects when V is known

If **G**, $\mathbf{R}_i$ and $\boldsymbol{\Sigma}_i$ are assumed to be known, the minimum square error linear unbiased estimator (BLUE) of β is found using generalized least square estimator and is given by;

$$\hat{\boldsymbol{\beta}} = [\mathbf{X}'(\mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R}_i + \boldsymbol{\Sigma}_i]^{-1}\mathbf{X}'(\mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R}_i + \boldsymbol{\Sigma}_i)^{-1}\mathbf{Y} =$$

$$\left[\sum_{i=1}^{n}\mathbf{X}_i^{'}(\mathbf{Z}_i\mathbf{G}_1\mathbf{Z}_i^{'} + \mathbf{R}_i + \boldsymbol{\Sigma}_i)\mathbf{X}_i\right]^{-1}\left[\sum_{i=1}^{n}\mathbf{X}_i^{'}(\mathbf{Z}_i\mathbf{G}_1\mathbf{Z}_i^{'} + \mathbf{R}_i + \boldsymbol{\Sigma}_i)^{-1}\mathbf{Y}_i\right] \tag{3.6}$$

Therefore the variance covariance of $\hat{\boldsymbol{\beta}}$ is given by;

$$\sigma^2[X'(ZGZ'+R+\Sigma)^{-1}X]^{-1} = \sigma^2\left[\sum_{i=1}^{n} X_i'(Z_iG_1Z_i^1+R_i+\Sigma_i)^{-1}X\right]^{-1}.$$

Furthermore, an unbiased estimator of σ^2 is obtained as;

$$\hat{\sigma}^2 = \frac{1}{\upsilon}\varepsilon' V^{-1}\varepsilon,$$

Where $\varepsilon = Y - X(X'V^{-1}X)^{-1}X'V^{-1}Y$ and $\upsilon = \sum_{i=1}^{n} p_i - Rank(X)$ equals the degrees of freedom for the error term.

However, if $(X'V^{-1}X)$ is not invertible, the generalized inverse would replace the inverse in equation 3.6 , provided that the functions under consideration are estimable.

In our case, assuming multivariate normality for υ_i and ε_i, $i = 1,...,n,$ then,

$$Y \sim N_{\Sigma p_i}(X\beta, \sigma^2[ZGZ' + R + \Sigma]).$$

Here, $\hat{\beta}$ and $\hat{\upsilon}$ are the restricted maximum likelihood (REML) estimators or maximum likelihood (ML) estimators.

Let us consider the situation of testing a linear hypothesis of the form $H_0 : L\beta = 0,$ where **L** is a full rank matrix.

Hence, the usual test statistic for the hypothesis above (H_0) is;

$$F_{\upsilon_1,\upsilon_2} = \frac{\hat{\beta}'L'(L(X'V^{-1}X)^{-1}L\hat{\beta}}{\hat{\sigma}^2\upsilon_1},$$

where υ_1 =Rank (L), υ_2 is the error degrees of freedom and $\mathbf{V} = \mathbf{(ZGZ' + R + \Sigma)}$.

Estimation of σ^2 and V

When the matrices **G** and **V** are unknown, their estimate might be obtained using REML or ML under the assumption of joint multivariate normality of $\boldsymbol{\upsilon}$ and $\boldsymbol{\varepsilon}$.

Most often a certain structure of **V** is assumed such that only few parameters of **V** must be estimated, let say $\boldsymbol{\theta_1, ..., \theta_s}$. Here fixed values of **V** are obtained by an iterative process and the estimate of $\boldsymbol{\beta}$ is obtained using the form of the BLUE. Then the likelihood function of V is then maximized with respect to $\boldsymbol{\theta_1, ..., \theta_s}$. in order to get the estimate of V. These two steps are iterated until a certain convergence criterion is met.

The ML estimators of $\boldsymbol{\theta_1, ..., \theta_s}$, **V** and $\boldsymbol{\sigma^2}$ are obtained by maximizing the logarithm of the normal likelihood function below;

$$l(\boldsymbol{\theta}) = -\frac{1}{2}\ln|\boldsymbol{\sigma^2 V}| - \frac{1}{2\sigma^2}\hat{\boldsymbol{\varepsilon}}'\mathbf{V}^{-1}\hat{\boldsymbol{\varepsilon}} - \frac{n}{2}\ln(2\pi) \qquad (3.7)$$

Hence the ML estimator of $\boldsymbol{\sigma^2}$ as a function of V is; $\hat{\sigma}_n^2 = \boldsymbol{\varepsilon}'\mathbf{V}^{-1}\boldsymbol{\varepsilon}/\mathbf{n}$. The ML estimates of $\boldsymbol{\theta_1, ..., \theta_s}$ is obtained by iterative schemes.

Furthermore, another set of estimator of $\boldsymbol{\theta_1, ..., \theta_s}$., **V** and σ^2 may be obtained us the REML by maximizing the functions (after profiling $\vec{\sigma}^2$)

$$-\frac{1}{2}\ln|\mathbf{V}| - \frac{1}{2}\ln|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}| - \frac{n-k}{2}\ln\hat{\boldsymbol{\varepsilon}}'\mathbf{V}^{-1}\hat{\boldsymbol{\varepsilon}} - \frac{n-k}{2}\left[1 + \ln\left(\frac{2\pi}{n-k}\right)\right],$$

Where K=rank(X). These two estimators are known to be asymptotically equivalent. Hence, since these estimators are asymptotically equivalent, one may alternate between these two estimators.

3.4.4.1 Hypothesis test for $H_0 : h(\boldsymbol{\theta}) = 0$.

If $\boldsymbol{\theta} = (\theta_1, ..., \theta_s)'$, then $\mathbf{h(\theta)}$ can be certain vector valued function of $\boldsymbol{\theta}$. The hypothesis test for $H_0 : h(\boldsymbol{\theta}) = 0$ against an alternative might be done using the following test statistics;

Likelihood ratio Test (LRT) Statistic: $\mathbf{T_L = 2[l(\hat{\theta}) - l(\hat{\theta})]}$

Wald's Statistic: $\mathbf{T_W} = nh(\hat{\boldsymbol{\theta}})'[\frac{\partial h(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^{l} I(\hat{\boldsymbol{\theta}})^{-1} \frac{\partial h(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}]^{-1} h(\hat{\boldsymbol{\theta}})$

Rao's Statistic: $\mathbf{T_R} = \frac{1}{n}\frac{\partial l}{\partial \boldsymbol{\theta}}^{l} I(\hat{\boldsymbol{\theta}}_0)^{-1} U(\hat{\boldsymbol{\theta}}_0)$,

Where $\boldsymbol{\theta}_0$ in the maximum likelihood estimate of $\boldsymbol{\theta}$ under null hypothesis and $\mathbf{I(\theta)}$ is the Fisher information matrix.

The three tests above have asymptotically χ_r distributions under the null hypothesis, with $r = rank(H(\boldsymbol{\theta}))$.

3.3.5 Estimating the Effects when V is estimated

We may come up with the BLUE of $\boldsymbol{\beta}$ and $\boldsymbol{\upsilon}$ by estimating **G** , **B** (where **B**=**R**+ $\boldsymbol{\Sigma}$)using either REML or ML. The respective estimate of $\boldsymbol{\beta}$ and $\boldsymbol{\upsilon}$ are got by solving the equations below;

$$\begin{pmatrix} \mathbf{X}'\hat{\mathbf{B}}^{-1}\mathbf{X} & \mathbf{X}'\hat{\mathbf{B}}^{-1}\mathbf{Z} \\ \mathbf{Z}'\hat{\mathbf{B}}^{-1}\mathbf{X} & \mathbf{Z}'\hat{\mathbf{B}}^{-1}\mathbf{Z} + \hat{\mathbf{G}}^{-1} \end{pmatrix} \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \vec{\boldsymbol{\upsilon}} \end{pmatrix} = \begin{pmatrix} \mathbf{X}'\mathbf{B}^{-1}\mathbf{Y} \\ \mathbf{Z}'\hat{\mathbf{B}}^{-1}\mathbf{Y} \end{pmatrix}$$

3.3.6 Missing Data for Bivariate pattern

Missing data for bivariate outcome has four possibilities: complete cases, cases with only $\mathbf{Y_1}$ observed, cases with only $\mathbf{Y_2}$ observed and cases with both variables missing.

Consider equation below;

$$f(\mathbf{Y}, M \mid \theta, \phi) = f(\mathbf{Y} \mid \theta) f(M \mid \mathbf{Y}, \phi) = \prod_{i=1}^{n} f(y_i \mid \theta) \prod_{i=1}^{n} f(M_i \mid y_i, \phi) \quad (3.8)$$

where M, ϕ and θ are defined as in equation (2.11).

$\Pr(M_{i1} = r, M_{i2} = s \mid y_{i1}, y_{i2}; \phi) = g_{rs}(y_{i1}, y_{i2}; \phi),\ r, s \in \{0,1\}$,

Where $g_{00}(y_{i1}, y_{i2}; \phi) + g_{10}(y_{i1}, y_{i2}; \phi) + g_{01}(y_{i1}, y_{i2}; \phi) + g_{11}(y_{i1}, y_{i2}; \phi) = 1$, that is g is a bivariate missing value indicator.

The dropout mechanism is NMAR if missingness of Y_j depends on unobserved values of Y_j

$$g_{11}(y_{i1}, y_{i2}; \phi) = g_{1+}(y_{i1}; \phi) g_{+1}(y_{i2}; \phi)$$
$$g_{10}(y_{i1}, y_{i2}; \phi) = g_{1+}(y_{i1}; \phi)(1 - g_{+1}(y_{i2}; \phi))$$
$$g_{01}(y_{i1}, y_{i2}; \phi) = (1 - g_{1+}(y_{i1}; \phi)) g_{+1}(y_{i2}; \phi)$$
$$g_{00}(y_{i1}, y_{i2}; \phi) = (1 - g_{1+}(y_{i1}; \phi))(1 - g_{+1}(y_{i2}; \phi))$$

The dropout mechanism is MAR if missingness of Y_1 depends on observed values of Y_1 or Y_2

and vice versa. Hence, $g_{10}(y_{i1}, y_{i2}; \phi) = g_{10}(y_{i2}; \phi)$ since y_{i1} is missing and y_{i2} is observed.

Therefore, the four missing data patterns are described below;

$$g_{11}(y_{i1}, y_{i2}; \phi) = g_{11}(\phi)$$
$$g_{10}(y_{i1}, y_{i2}; \phi) = g_{10}(y_{i2}; \phi)$$
$$g_{01}(y_{i1}, y_{i2}; \phi) = g_{01}(y_{i1}; \phi)$$
$$g_{00}(y_{i1}, y_{i2}; \phi) = 1 - g_{10}(y_{i2}; \phi) - g_{01}(y_{i1}; \phi) - g_{11}(\phi)$$

The MAR assumption above is a weaker assumption as compared to the MCAR assumption, which assumed that missingness is unrelated to outcomes.

3.3.7 Modeling of Bivariate Outcomes

There are numerous advantages in modeling bivariate outcomes. First, this model allows us to explore variation at different levels of the hierarchy (random effect) and to model the correlation structure, both the serial correlation and the cross correlation. Second, as shown by Pantazis and Touloumi(2007), the bivariatejoint multivariate random effect(JMRE) model achieves greater bias reduction in all model parameters compared to the two independent JMRE models. Third, there is a greater flexibility in dealing with independent variables. Fourth, we can easily test hypotheses on either end point individually or on both ends simultaneously. Fifth, since our bivariate multilevel markers' model is a made up of two mixed model with correlated random effect it possesses some the quality of mixed model such that one can always include cases of missing data since the models is fit by maximum likelihood (Dempster, Selwyn, Patel, and Roth, 1984; Laird and Ware, 1982) as compared to the traditional ANOVA and MAVOVA were missing data are simple deleted since it is fitted using moment-matching method. Sixth, according to McCulloch (2008), we can avoid multiple testing by forming joint models without having to resort to ad hoc methods such as Bonferroni adjustment. Furthermore, this model is parsimonious, that is, estimating only the essential number of parameters as compared to MANOVA which is based on the unstructured variance-covariance structure thus estimating a single parameter for each variance for whatever the true structure might be. Finally, since this model is a time

variant or invariant.regression model, so it can incorporate both categorical and continuous predictors that might be either.

3.3.8 Example illustrating modeling bivariate outcomes

3.3.8.1 Background

Using data from the National Health interview Survey from 1997 to 2006, we present a multilevel analysis of changes in Body Mass Index (BMI) and Number of Cigarette Smoked per day in the United States. Smoking and obesity are the leading causes of preventable mortality and morbidity in the USA and most parts of the developed world. A two stage bivariate model of changes in obesity and number of cigarette smoked per day is proposed. At the within subject stage, an individual's BMI status and the number of cigarette smoked per day are jointly modeled as functions of individual growth trajectories plus a random errors. At the between-subject stage, the parameters of the individual growth trajectories are allowed to vary as a function of differences between subjects with respect to demographic and behavioral characteristics and with respect to the four regions of the United States (Northeast, West, South and North central). Our two stage modeling techniques are more informative than standard regression because they characterize both group-level (nomothetic) and individual-level (idiographic) effects, yielding a more complete understanding of the phenomena under study.

3.3.8.2 Study Aims

Our goal for this study is to see how BMI and smoking jointly change over time while adjusting for other potential predictors (e.g., age, gender, socio-economic status, education). Furthermore, we want to know which of these predictors are jointly related to both BMI and smoking. Finally concluding remarks and reference to some application of the Joint Multivariate Random Effect (JMRE) models are given in the final section.

The bivaraite outcome BMI and smoking were transformed using the transformation log(y+1) where y is the joint distribution of BMI and smoking. Table 3.1 summarizes the variables used in this

analysis. Based on the AIC measure,the bivariate JMRE model was significantly better than two separate univariate JMRE models (-260157 vs. -322167, likelihood ratio 62010 with 8 degree of freedom, $p<0.0001$). There was 14% cross correlation between obesity and smoking. A prominent mitigating factor is the fact that most previous research showed that increase in the number of cigarettes smoked reduces obesity (Chen, Yen and Eastwood (2005)). Grossman and Saffer (2004) also looked at the association between smoking and obesity, concluding that non-smoker and former smokers are more likely to be obese than smokers. Kuchler and Variyam (2003) looked at the association between smoking and body weight and noted that after simultaneously controlling for a number of variables (age, sex, marital status, etc.), cigarette consumption does not have a direct-long term effect on body mass index. However, there have been several arguments for and against this assertion. Many researchers will argue that smoking might increase energy consumption while suppressing appetite, (Wilson (2002)). However, as described by Perkins (1987), there is a more complex relationship between obesity and smoking. Part of the effect of smoking on body weight may be due to its influence on energy expenditure; both smoking and nicotine intake have been shown to increase metabolic rate. On the other hand, some researchers stipulate that smokers tend to accumulate other risk behaviors potentially favoring weight gain, for example, low physical activity, high alcohol consumption, poor diet (Klesges, ScienceDaily, 1998). However, here we are not concerned with how cigarette smoking can be used to predict obesity or vice versa. We are interested in the joint evolution of smoking and obesity over time and on variables that are jointly associated with both smoking and obesity.

3.3.8.3 Results

Several of the variables were significantly associated with both smoking and BMI. The joint evolution of smoking and obesity was significant . Men were more likely to smoke and be obese as compared to females. Obesity and smoking were also associated with age; the highest obesity and smoking was found in the age group of 35-50 years, while obesity and smoking in those > 50 years old

was slightly less. Marital status was also a significant factor. Those who never married or were divorced were less likely to smoke and be obese compared to married people. Mandal and Chern, (2007) had a similar result for marital status using marital status to predict BMI. The socio-economic variable, poverty level indicated that those falling within higher socio-economic groups were less likely to smoke and be overweight. Higher education was also associated with lower obesity and smoking. Place of birth was also a significant variable with those born in the USA more likely to smoke and be obese. Race was associated with obesity; African Americans had higher smoking and obesity rates than those from all other races, Caucasians were also likely to smoke and be obese compared to those of Asian and Middle Eastern descent.

There were significant differences in obesity and smoking amongst the four regions in the United States. Individuals from the West and North central regions were more likely to be obese and smoke than those from the North East and South regions. However, people from the west tend to have a higher smoking and obesity rate.

Table3. 1.Modeling BMI and Smoking: Bivariate Multilevel Markers' Model

Effect	Parameter estimate	F Value	Pr > F
(Smoking+Bmi)	12.568	39460.4	<.0001
(smoking+Bmi)*time	-	23.74	<.0001
Gender			
male vs female	1.7498	505.87	<.0001
Age			
25-35 vs <25	1.5559	348.16	<.0001
35-50 vs< 25	3.1926	628.32	<.0001
>50 vs < 25	3.4098	288.56	<.0001
Marital status			
Married vs never married	0.1261	36.82	<.0001
Divorced vs never married	0.4182	0.78	0.3768
Poverty line			
1-3.99 vs < 1	-0.4182	4.73	0.0297
>4.00 vs < 1	-0.6868	30.36	<.0001
Highest education level attend			
High school (HS) vs pre-HS	0.9005	0.34	0.5378
College vs pre-HS	-0.2528	3.70	0.0544
Postgraduate vs pre-HS	-1.5707	30.74	<.0001
Birth place			
USA vs others	2.7384	158.71	<.0001
Race			
White vs others	1.0696	1.33	0.2482
Black vs others	0.6930	28.61	0.0001
Region			
Northeast vs South	-0.1873	0.62	0.4306
Northcentral vs South	0.1572	6.71	0.0096
West vs South	-1.4291	19.59	<.0001

4.0 CHAPTER FOUR

4.4.1 Model selection process

The model selection process is difficult for multilevel data since it depends on not only the question we wish to answer but the best means structure and also the optimal covariance structure. Most statisticians approach identifying the mean structure by using backward, forward and stepwise procedures (e.g. multiple testing, arbitrary significance levels used in dropping or acquiring variables). However these selection procedures are not typically used when selecting an appropriate correlation structure.

In contrast, the following six model selection criteria have good statistical properties in selecting the best correlation structure:

- Akaike Information Criterion (AIC)
- Bozdogan Corrected Akaike Information Criterion (CAIC)
- Hurvich and Tsai The Corrected Akaike Information Criterion (AICC)
- Schwatz Bayesian Information Criterion (BIC)
- Likelihood ratio test for clustered
- Hannan and Quin Information Criterion
- Standard error of estimates

The first three are the most popular criteria used in the last two decades. Ferron et al (2002) studied performance of AIC and BIC as well as the likelihood ratio test (LRT) in guiding the choice of the covariance structure for one outcome data. AIC, BIC and LRT were used to choose between the true covariance structure and one alternative structure, namely $\sigma^2\mathbf{I}$. AIC chose the right covariance structure 79% of the time versus 66% for BIC and 71% for LRT. Furthermore, he showed that AIC, BIC and LRT performed better when the sample size, the length of the repeated measures and the level of

autocorrelation were higher with the length of repeated measures being the most influential characteristics.

Al-Marshadi et al (2007) showed that new a approach, AICC, and the older BIC performed best in identifying the right correction by simulating data with the following correlation structures: Compound Symmetry (CS), First-Order Autoregressive (AR(1)), Heterogeneous First-Order Autoregressive (ARH(1)), Compound Symmetry (CSH), Independent Errors (VC), Main Diagonal(UN(1)), Toeplitz (TOEP) and Unstructured (UN) covariance structure.

4.4.2 The Akaike Information Criterion (AIC)

In order to understand the concept behind AIC, one need to understand the definition of the Kullback-Leibler information which is a measure of the distance between two density functions. The Kullback-Leibler information may be use to represent the distance between the true model and the selected model. The best model is often the model with the smallest distance between the true and selected model.

Let the true model density by $f(.)$ and the joint density for the selected model be $g(.,\vec{\theta})$, where $\vec{\theta}$ is the estimated vector of d parameters by maximum likelihood method.

The Kullback-Leibler distance is;

$$D(f,g) = E_f \log\frac{f(X)}{g(X,\vec{\theta})} = \int \log\frac{f(x)}{g(x,\vec{\theta})} f(x)dx \qquad (4.1)$$

where x is the observed sample data of n independent observations. Here the expectation provides the basis of the model selection. The asymptotic approximation for the estimation of the expectation is

$$\hat{E}(D(f,g)) = \int f(x)\log(f(x))dx + (-\log(g(x,\hat{\theta}))) + trace(\hat{\Sigma}\hat{\Omega}) \qquad (4.2)$$

where,

$\log g(x,\hat{\theta})$ is the maximized log likelihood function based on the observed data and $\hat{\theta}$ is the maximum likelihood estimate of θ,

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{d\log g(x_i,\hat{\theta})}{d\theta_u}\right]\left[\frac{d\log g(x_i,\hat{\theta})}{d\theta_v}\right], u,v = 1,2,...,d,$$

and

$$\hat{\Omega} = \frac{-1}{n}\sum_{i=1}^{n}\left[\frac{d^2\log g(x_i,\hat{\theta})}{d\theta_v d\theta_u}\right], u,v = 1,2,...,d$$

From equation (4.2) above, we noticed that, the term $\int f(x)\log(f(x))dx$ is highly dependent on the true model which is unknown. Nonetheless, this term is fixed when comparing between models. The term, $-\log g(x,\hat{\theta}) + trace(\hat{\Sigma}\hat{\Omega})$, is needed for the comparison process, which is computable for parametric model. If we assumed that the true model is contained within the family of models from which the fitted model is obtained then $-\log g(x,\hat{\theta}) + trace(\hat{\Sigma}\hat{\Omega})$ becomes $-\log g(x,\hat{\theta}) + d$. Hence, AIC is twice $-\log g(x,\hat{\theta}) + d$, that is,

AIC=2($-\log g(x,\hat{\theta}) + d$)=-2loglikelihood+2d (4.3)

From equation (4.3) above, we see that AIC imposes a penalty of two units per parameter in the model. One may look at the first term AIC as a measure of lack of fit while the second term is a penalty for estimating d parameters which enforces parsimony. But, AIC theory is more compelling than this heuristic interpretation. For more elaborate detail on the theory of AIC, see Akaike (1973, 1985), Sakamota et al. (1986) and Bozdogan (1987).

Bozdogan (1987) reviewed and adjusted the AIC in order to achieve dimension consistency, that is, consistent estimation of the order (d) of model with a finite true model order that does not increases as sample size increases. The model was presented as

$$CAIC = -2\log g(x,\hat{\theta}) + d(\log(n)+1) \tag{4.4}$$

Hurvich and Tsai (1989) show that the AIC has a small sample bias adjustment. They corrected it by proposing the Corrected Akaike Information Criterion (AICC),

$$AICC = AIC + \frac{2(d+1)(d+2)}{n-d-2} \tag{4.5}$$

Both AIC and AICC are formulated on the basis that the true model requires a very large number of parameters for its representation but not the CAIC which has a larger penalty term and results in selected model with fewer parameters than AIC and AICC.

4.4.3 Schwarz Bayesian Information Criterion (BIC)

Shabita, 1976 and Schwarz 1978 introduced a technique similar to CAIC for consistency and termed the criterion BIC for Bayesian Information Criterion. Schwarz's Bayesian Criterion (BIC) is defined as

$$BIC = -2\log g(x,\hat{\theta}) + d(\log(n^*)) \tag{4.6}$$

where $n^* = n$ for ML and $(n-p)$ for REML. Similar to the other criterion, a model with a smaller value of BIC is preferred.

AIC, AICC and CAIC are objective criteria; they do not required candidate models be nested. For nested model, we used the likelihood ratio test discussed below.

4.4.4 Likelihood Ratio Tests (LRT)

Likelihood ratio test are used for nested models because they allow sequential decisions to be made about parameters affected by treatment. Thus, a model can be selected for final inference in a traditional hypothesis-testing manner.

Let $Y_{p\times 1}$ be a p dimensional normally distributed random vector with mean μ and variance covariance parameter B. Let assume we have our data as a random sample $y_1,...,y_n$ from this population. The likelihood ratio test for the null hypothesis,

$$H_0 : B \text{ has a given covariance structure}$$

Is given by

$$L = \frac{\max_{H_0} g(B \mid data)}{\max_{unrestricted} g(B \mid data)} \tag{4.7}$$

where g(B|data) is same as the $f(y_1,...,y_n)$, the joint density function of $y_1,...,y_n$ and for maximization purposes it is treated as a function of B for given data.

5.0 CHAPTER FIVE

5.5.1 Proposal

There have been numerous publications, presentations and controversies in addressing how to choose the right covariance structure for longitudinal, univariate data (Wolfinger, 1993, 1996, Rosner and Glynn (1997), Kincaid, 2005; Ferron et al .,2002; littell et el 2000, littell et el 2006, Ferron et al., 2002; Grady and Halms 1995, James Hardin and Hilbe 2006). Some of these approaches has been base on graphical tools while others have look at the trend, sample sizes, cluster size etc. Despite all these publications, presentation and controversies in coming up with a rationale for choosing the right correlation structure, no one to the best of my knowledge has looked at any of the cases with a bivariate outcome. The problem with bivariate outcome is very challenging because in addition to the within and between correlation we usually encountered in multilevel data, there is a serial correlation that exist between the two outcome. There are many correlation structures which can be used to model longitudinal data. However, there is a limited number which we often used in our daily analysis. Below are some examples from SAS online manuel(SAS Institute, Inc.(1992)) of correlation structures together with their parameter number which we would be using for this study.

Table 5.1. Correlation Structures

Description	Structure	Example	parameters #
Variances Components	VC (default)	$\begin{bmatrix} \sigma_B^2 & 0 & 0 & 0 \\ 0 & \sigma_B^2 & 0 & 0 \\ 0 & 0 & \sigma_{AB}^2 & 0 \\ 0 & 0 & 0 & \sigma_{AB}^2 \end{bmatrix}$	1
Compound Symmetry	CS	$\begin{bmatrix} \sigma^2+\sigma_1 & \sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma^2+\sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma^2+\sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma^2+\sigma_1 \end{bmatrix}$	2
Unstructured	UN	$\begin{bmatrix} \sigma_1^2 & \sigma_{21} & \sigma_{31} & \sigma_{41} \\ \sigma_{21} & \sigma_2^2 & \sigma_{32} & \sigma_{42} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 & \sigma_{43} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_4^2 \end{bmatrix}$	$t(t+1)/2$
Banded Main diagonal	UN(1)	$\begin{pmatrix} \Sigma_1 & & 0 \\ & \ddots & \\ 0 & & \Sigma_4 \end{pmatrix}$	t
First-Order Autoregressive	AR(1)	$\sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$	2
Toeplitz	TOEP	$\begin{bmatrix} \sigma^2 & \sigma_1 & \sigma_2 & \sigma_3 \\ \sigma_1 & \sigma^2 & \sigma_1 & \sigma_2 \\ \sigma_2 & \sigma_1 & \sigma^2 & \sigma_1 \\ \sigma_3 & \sigma_2 & \sigma_1 & \sigma^2 \end{bmatrix}$	2t-1
Heterogeneous CS	CSH	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho & \sigma_1\sigma_4\rho \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho \\ \sigma_3\sigma_1\rho & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$	t+1
Huynh-Feldt	HF	$\begin{bmatrix} \sigma_1^2 & \frac{\sigma_1^2+\sigma_2^2}{2}-\lambda & \frac{\sigma_1^2+\sigma_3^2}{2}-\lambda \\ \frac{\sigma_2^2+\sigma_1^2}{2}-\lambda & \sigma_2^2 & \frac{\sigma_2^2+\sigma_3^2}{2}-\lambda \\ \frac{\sigma_3^2+\sigma_1^2}{2}-\lambda & \frac{\sigma_3^2+\sigma_2^2}{2}-\lambda & \sigma_3^2 \end{bmatrix}$	t+1

Table 5.1. Continued

Heterogeneous AR(1)	ARH(1)	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho^2 & \sigma_1\sigma_4\rho^3 \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho^2 \\ \sigma_3\sigma_1\rho^2 & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho^3 & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$	t+1

5.5.2 Replication

Example 1

We generate a continuous response to approximately replicate the AP-HP studies of patients infected by HIV at "Claude Bernard, Universite' Denis Diderot-Paris VII, 46 rue Henri-Huchard 75018 Paris. FR". The clinicians there were interested in the joint evolution of HIV CD4 and CD8 lymphocytes level for patients on HIV-AIDS patients on antiretroviral drug.

$$y_{ij} = \theta(x_{ij} + ERR) + \varepsilon_{ij} \quad (i = 1,...50;\ j = 1,...,5),$$

in order to study the joint evolution of CD4 + T lymphocytes which are a specific target of the virus, CD8 + T lymphocytes or the inflammation process (β_2 microglobuline). These markers are associated as the infection measured by HIV RNA induced inflammation and destruction of immune cells. In our study, we controlled for gender and age. As shown below the significance of our outcome is strongly dependent on our choice of correlation structure;

Table 5.2. Example of Replicates Results

Correlation structure	**F-test p-value CD4 + T lymphocytes**	**F-test p-value for (CD8+T lymphocytes)*Time**	F-test P-value Age	F-test P-value Gender
Variance component	<0.0001*	0.0687	0.3884	0.2152
Compound symmetry	<0.0001*	0.0570	0.3983	0.2394
Unstructured	<0.0001*	0.1274*	0.3413	0.2052
Banded Main diagonal	<0.0001*	0.1483	0.3568	0.2043
First-Order Autoregressive	<0.0001*	0.0680	0.3883	0.2394
Toeplitz	<0.0001*	0.0583	0.3764	0.2227
Spatial Power	<0.0001*	0.0570	0.3983	0.2394
Heterogeneous CS	<0.0001*	0.0564	0.3627	0.2251
Huynh-Feldt	<0.0001*	0.0555	0.3496	0.2114
Heterogeneous AR(1)	<0.0001*	0.0564	0.3627	0.2251

Despite the above results is just for a single replicate and one can not draw any strong conclusion but for antedoctal purposes we observe that, the Compound symmetry, First order Autoregressive and independent yield similar results. Furthermore, banded main diagonal and Toeplitz have similar results, while the others are not that similar. However our judgment of the best correlation would not be based on either identical results or similar results but rather, based on the selection processes such AIC, AICC and BIC as outline in Chapter 4 and also on the standard error of our estimates.

5.5.3 Simulation

5.3.1 Nested data (not longitudinal)

We generated bivariate normal (Z_{ix}) observation $\{(y_{i1}, y_{i2}), i = 1,...,50\}$ with 500 replicates on (Y_1, Y_2) using the equations below: $\begin{matrix} y_{i1} = \theta_1 + Z_{i1} \\ y_{i2} = \theta_2 + \theta_3 * Z_{i1} + Z_{i2} \end{matrix}$, $\theta_i \in$. By varying n and the $\theta_i \in [-1,1]$ we can obtain a correlation as small 50% between (Y_1, Y_2) and as high as 99%.

Examples of values of θ_i , $i = 1,2,3$ are 1/6,-1/5 and 2/3 respectively.

Example for two clusters, Average value for 500 replicates:

Table 5.3. Results for Nested Cases

Correlation structure	2llog(deviance)	Joint Standard error of estimate	Standard error of estimate
Independent	294	0.5499	0.0152
Compound Symmetry	294	0.5482	0.0151

Average value for 6 clusters

Correlation Structure	-2llog	Joint Standard error of estimate	Standard error of estimate
Independent	945.8	0.4030	0.0077
Unstructured	941.4	0.4398	0.0077
Toeplitz, AR(1) & Compound symmetry	945.5	0.3660	0.0077

From the above, we come up with the following conclusion:

For observations are clustered (not collected over time), There was little difference between the Independent, Compound Symmetry and First Order Autoregressive. Similar to Bell et el (2008) our cluster size were between 1 to 15 and our sample sizes were varying between 50 to 100 per cluster. As our cluster size and sample size increases the Independent tends to perform best.

5.5.4 Simulating Longitudinal data.

In order to come up with a rationale for choosing the correlation structure, we first generate a Gaussian bivariate data with sample size 50 and 100 replicate, using the equation times point of x variable with correlated random effect(error). We generated two level bivaraiate models in which the observations are nested within groups. For instance a continuous outcome was generated as a linear function of k predictors. Furthermore, we assumed one and/or a combination of the five correlation structure (Independent, Compound Symmetry, First Order Autoregressive, Toeplitz and Unstructured) . For each time point we simulated data with same bivariate correlation and different combination. We started with a 2×2 correlation matrix where there are just two different correlation structure, i.e. Independent and Compound Symmetric. For this case we simulated data with both having the same cross correlation, that is , Independent versus Independent, Compound Symmetry versus Compound Symmetry and a combination of them, i.e. Independent versus Compound Symmetry. Then we move up to the $3 \times 3,\ 4 \times 4$, 5×5 etc . For our analysis, all covaraiance term for the first level was model as $\Sigma = \sigma^2 I$, the serial correlation was model using different structures in order to come up with a rationale for choosing the best correlation with least standard error.

$$y_{ij}^{(1)} = 1 + 0.1000(x_i + ERR) + e_{ij}$$
$$y_{ij}^{(2)} = 1 + 0.2098(x_i + ERR) + e_{ij} \quad \text{for } i = 1,...,n;\ j = 1,...,t \qquad (5.3.2.1)$$

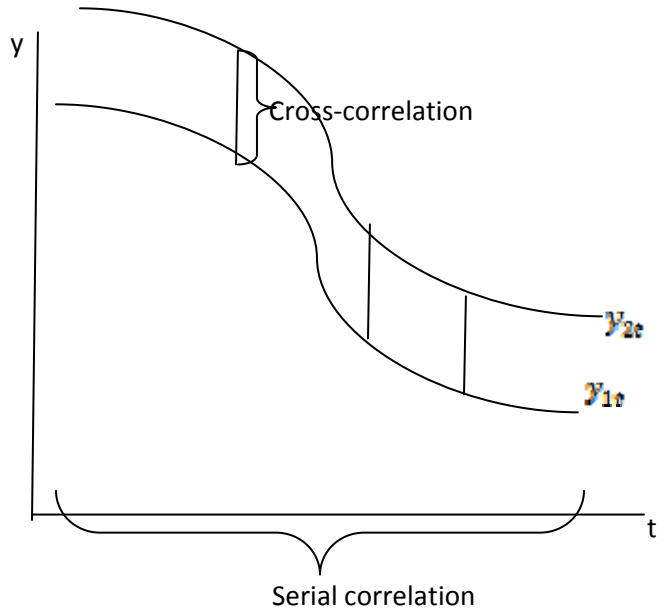

Figure 5.1. Example of Bivariate Plot

For a 2×2 with t=1, 2, we generated 2 bivariate normal distributions of sample size 50. This was then replicated 100 times. These distributions were created with a combination of correlation structures 5.1 below. We then generated a "random error "i.e.(.12,0.03) distribution that followed a normal distribution with mean 0.12 and variance 0 .03. The outcomes of the first two time point normal distribution, are labeled as e11 and e21 and the outcomes of the second two time points normal distribution are labeled as e12 and e22 respectively.

Two normal distributions were generated as follows,Y11=1+(.1000*(xi+error))+e11 and Y21=1+(0.2898*(xi+error))+e21 .

The errors (e11,e21) are the actual outcomes of the first two time points normal distribution as stated earlier. We then followed a similar scheme for the second set of two time points' normal distributions. Here we have Y11=1+(.1000*(xi+error))+e11 and Y21=1+(0.2898*(xi+error))+e21. The same procedure was followed to t=3, 4, 5, etc.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} and \begin{pmatrix} 1 & 0.4 \\ 0.4 & 1 \end{pmatrix}. \qquad (5.1)$$

We first established that the choice of the serial correlation in our JMRE is independent of the simulated structure of our two random effect model with correlated random effect.

Table 5.4. Average Values for our 500 Replicates for t=2

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	831.5125747	854.0272098	854.0272098	854.0272098	792.4723240
AIC	839.5125747	862.0272098	862.0272098	862.0272098	816.4723240
AICC	839.6505057	862.1651408	862.1651408	862.1651408	817.5787070
BIC	847.1606667	869.6753018	869.6753018	869.6753018	839.4166001
alpha	-9.1604014	-9.1664364	-9.1664364	-9.1664364	-9.1604066
beta1	31.0216966	31.0216966	31.0216966	31.0216966	31.0216966
beta2	-0.0014112	-0.0014112	-0.0014112	-0.0014112	-0.0014112
beta3	0.0101815	0.0101815	0.0101815	0.0101815	0.0101815
beta5	0.0033003	0.0034461	0.0034461	0.0034461	0.0032792
e_alpha	1.4354172	0.9985101	0.9985101	0.9985101	0.9824989
e_beta1	0.2631727	0.2344216	0.2344216	0.2344216	0.3264953
e_beta2	0.0695045	0.0950186	0.0950186	0.0950186	0.0766067
e_beta3	0.0823257	0.0973239	0.0973239	0.0973239	0.0992148
e_beta5	0.0356111	0.0246496	0.0246496	0.0246496	0.0240607
p_alpha	5.5519739E-7	6.049219E-11	6.049219E-11	6.049219E-11	6.05561E-11
p_beta1	3.32233E-102	1.44031E-106	1.44031E-106	1.44031E-106	4.838645E-92
p_beta2	0.4314124	0.5299733	0.5299733	0.5299733	0.4605343
p_beta3	0.4775434	0.5279397	0.5279397	0.5279397	0.5329074
p_beta5	0.5757831	0.4568330	0.4568330	0.4568330	0.4353011

The Independent performed best follow by the Compound Symmetry. The Independent correlation structure tends to have the least standard error however this does not always correspond to the smallest AICC and BIC. From the above it is easy to conclude that the Unstructured performed best

since it AICC and BIC are the smallest, furthermore it tends to have the least standard. However, this is not clear cut, since the unstructured does not always converge, it happens to fail to converge in about 40% of it cases. .One possible explanation is that likelihoods results so that the minimization procedure is not able to produce convergence. This problem seemed to be most apparent the unstructured convaruance pattern.

For the $t=3$ we generated data with the following correlation structure and their combinations:

$$\begin{pmatrix} 1 & 0.4 & 0.7 \\ 0.4 & 1 & 0.39 \\ 0.7 & 0.39 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0.4 & 0.4 \\ 0.4 & 1 & 0.4 \\ 0.4 & 0.4 & 1 \end{pmatrix} \text{and} \begin{pmatrix} 1 & 0.4 & 0.16 \\ 0.4 & 1 & 0.4 \\ 0.16 & 0.4 & 1 \end{pmatrix}.$$

Table 5.5. Average Values for our 500 Replicates for t=3

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	1148.22	1180.92	1180.92	1180.92	1102.99
AIC	1156.22	1188.92	1188.92	1188.92	1126.99
AICC	1156.33	1189.02	1189.02	1189.02	1127.80
BIC	1163.87	1196.57	1196.57	1196.57	1149.93
alpha	-8.8792422	-8.8666235	-8.8666235	-8.8666235	-8.7224126
beta1	31.0098313	31.0098313	31.0098313	31.0098313	31.0098313
beta2	0.0051315	0.0051315	0.0051315	0.0051315	0.0051315
beta3	-0.0057786	-0.0057786	-0.0057786	-0.0057786	-0.0057786
beta5	-0.0026483	-0.0029695	-0.0029695	-0.0029695	-0.0065822
e_alpha	1.3801776	0.9007078	0.9007078	0.9007078	0.8381520
e_beta1	0.2548272	0.2165301	0.2165301	0.2165301	0.3091154
e_beta2	0.0510253	0.0709417	0.0709417	0.0709417	0.0545545
e_beta3	0.0617839	0.0735216	0.0735216	0.0735216	0.0736783
e_beta5	0.0341131	0.0221392	0.0221392	0.0221392	0.0203750
p_alpha	1.1292241E-6	7.606553E-10	7.606553E-10	7.606553E-10	3.708049E-11
p_beta1	3.81656E-105	4.42998E-112	4.42998E-112	4.42998E-112	2.224803E-93
p_beta2	0.4530199	0.5614956	0.5614956	0.5614956	0.4744171
p_beta3	0.4322918	0.4839922	0.4839922	0.4839922	0.4830315
p_beta5	0.5469386	0.4141046	0.4141046	0.4141046	0.4596071

For the case of a $t=3$ matrix, that is t=1,2,3. We quickly notice that the choice of the correlation structure does not depend on our simulation structure. However, of the $4+C_2^4=10$ combinations, our simulation shows that, the Unstructured followed by the independent structure performed best.

For the t=4matrix, there are sevaral different correlation models, we simulated data with the following correlation models; independent, compound symmetric, First order autoregressive, Toeplitz and Unstructure. For the model with bivariate outcomes using the 4*4 correlation models, there were a total of fifteen different combinations of the two independent corralation models. However, we first established that the choice of our correlation does not depends on the independent simulated structure.

Table 5.6. Average Values for our 500 Replicates for t=4

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	1125.18	1176.52	1176.52	1176.52	1104.49
AIC	1132.75	1184.52	1184.52	1184.52	1127.23
AICC	1132.84	1184.62	1184.62	1184.62	1127.97
BIC	1139.98	1192.17	1192.17	1192.17	1148.98
alpha	-9.1443393	-9.1454427	-9.1454427	-9.1454427	-9.1313151
beta1	31.0630501	31.0630501	31.0630501	31.0630501	31.0630501
beta2	-0.0010717	-0.0010717	-0.0010717	-0.0010717	-0.0010717
beta3	0.0035442	0.0035442	0.0035442	0.0035442	0.0035442
beta5	0.0028763	0.0029224	0.0029224	0.0029224	0.0025314
e_alpha	1.3558993	0.9125308	0.9125308	0.9125308	0.9878383
e_beta1	0.2633106	0.2161295	0.2161295	0.2161295	0.9878383
e_beta2	0.0508053	0.0729195	0.0729195	0.0729195	0.0508571
e_beta3	0.0479144	0.0674071	0.0674071	0.0674071	0.0512086
e_beta5	0.0336094	0.0225514	0.0225514	0.0225514	0.0243153
p_alpha	5.0359452E-7	3.751868E-11	3.751868E-11	3.751868E-11	8.270418E-11
p_beta1	2.06605E-103	1.59496E-111	1.59496E-111	1.59496E-111	4.329218E-94
p_beta2	0.4747985	0.5855728	0.5855728	0.5855728	0.4742428
p_beta3	0.4787319	0.5821697	0.5821697	0.5821697	0.4966984
p_beta5	0.5922629	0.4591166	0.4591166	0.4591166	0.5126173

Table 5.7. Average Values for our 500 Replicates for t=5

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	1385.02	1479.65	1479.65	1479.65	1362.14
AIC	1391.02	1487.65	1487.65	1487.65	1382.14
AICC	1391.07	1487.73	1487.73	1487.73	1382.59
BIC	1396.75	1495.29	1495.29	1495.29	1401.26
alpha	-8.8552459	-8.8691460	-8.8691460	-8.8691460	-8.8551239
beta1	30.9957876	30.9957876	30.9957876	30.9957876	30.9957876
beta2	0.000506778	0.000506778	0.000506778	0.000506778	0.000506778
beta3	0.0021149	0.0021149	0.0021149	0.0021149	0.0021149
beta5	-0.0040646	-0.0037161	-0.0037161	-0.0037161	-0.0040486
e_alpha	1.3193034	0.8590256	0.8590256	0.8590256	0.9380357
e_beta1	0.2505538	0.1928062	0.1928062	0.1928062	0.2924777
e_beta2	0.0344394	0.0542960	0.0301847	0.0378633	0.0344352
e_beta3	0.0343780	0.0518760	0.0518760	0.0518760	0.0346648
e_beta5	0.0326854	0.0212237	0.0212237	0.0212237	0.0230563
p_alpha	2.582159E-7	4.047317E-11	4.047317E-11	4.047317E-11	7.114741E-11
p_beta1	5.47716E-104	2.63594E-117	2.63594E-117	2.63594E-117	4.061695E-92
p_beta2	0.5833253	0.7057174	0.7057174	0.7057174	0.5832711
p_beta3	0.6132161	0.7262323	0.7262323	0.7262323	0.6158577
p_beta5	0.5897531	0.4318724	0.4318724	0.4318724	0.5133816

For, t=4 and t=5, the unstructured still performed best , closedly follow by the independent structure.

Table 5.8. Average Values for our 500 Replicates for t=6

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	1716.46	1805.04	1805.04	1805.04	1672.67
AIC	1724.35	1813.04	1813.04	1813.04	1695.74
AICC	1724.42	1813.10	1813.10	1813.10	1696.24
BIC	1731.89	1820.68	1820.68	1820.68	1717.80
alpha	-8.9780465	-9.0361787	-9.0361787	-9.0361787	-8.8966496
beta1	30.9937281	30.9937281	30.9937281	30.9937281	30.9937281
beta2	-0.0052023	-0.0052023	-0.0052023	-0.0052023	-0.0052023
beta3	-0.0110074	-0.0110074	-0.0110074	-0.0110074	-0.0110074
beta5	-0.000056033	0.0014136	0.0014136	0.0014136	-0.0021075
e_alpha	1.3349430	0.8078129	0.8078129	0.8078129	0.8178826
e_beta1	0.2392514	0.1791568	0.1791568	0.1791568	0.2944956
e_beta2	0.0282578	0.0444458	0.0444458	0.0444458	0.0271108
e_beta3	0.0318448	0.0441987	0.0441987	0.0441987	0.0401114
e_beta5	0.0330932	0.0199528	0.0199528	0.0199528	0.0199645
p_alpha	7.9031033E-7	2.717182E-12	2.717182E-12	2.717182E-12	3.417216E-12
p_beta1	3.07229E-104	4.70866E-120	2.717182E-120	2.717182E-120	1.052374E-95
p_beta2	0.4936211	0.6295475	0.6295475	0.6295475	0.4796045
p_beta3	0.3843371	0.4989231	0.4989231	0.4989231	0.4630468
p_beta5	0.5391019	0.3801430	0.3801430	0.3801430	0.4156699

Table 5.9. Average Values for our 500 Replicates for t=7

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	2029.14	2109.71	2109.71	2109.71	2005.53
AIC	2036.23	2117.71	2117.71	2117.71	2027.64
AICC	2036.28	2117.77	2117.77	2117.77	2028.03
BIC	2043.01	2125.36	2125.36	2125.36	2048.77
alpha	-8.9449276	-8.9460238	-8.9460238	-8.9460238	-8.9777822
beta1	31.0252292	31.0252292	31.0252292	31.0252292	31.0252292
beta2	0.0020883	0.0020883	0.0020883	0.0020883	0.0020883
beta3	-0.0048663	0.0332403	0.0332403	-0.0048663	-0.0048663
beta5	-0.0013963	0.0175089	0.0175089	-0.0013866	-0.000573112
e_alpha	1.2086144	0.7076256	0.7076256	0.7076256	0.8183471
e_beta1	0.2271918	0.1694522	0.1694522	0.1694522	0.2654506
e_beta2	0.0272245	0.0381057	0.0381057	0.0381057	0.0276664
e_beta3	0.0211743	0.0332403	0.0332403	0.0332403	0.0215225
e_beta5	0.0299999	0.0175089	0.0175089	0.0175089	0.0201420
p_alpha	5.1966357E-9	1.07978E-17	1.07978E-17	1.07978E-17	2.744026E-13
p_beta1	1.04156E-109	1.15306E-123	1.15306E-123	1.15306E-123	7.011467E-99
p_beta2	0.4958068	0.6049481	0.6049481	0.6049481	0.5009605
p_beta3	0.4623217	0.6084078	0.6084078	0.6084078	0.4687942
p_beta5	0.5984166	0.3851127	0.3851127	0.3851127	0.4504902

Table 5.10. Average Values for our 500 Replicates for t=8

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	2244.61	2378.91	2378.91	2378.91	2197.86
AIC	2252.51	2386.91	2386.91	2386.91	2221.11
AICC	2252.56	2386.97	2386.97	2386.97	2221.49
BIC	2260.05	2394.56	2394.56	2394.56	2243.34
alpha	-8.7793607	-8.7827632	-8.7827632	-8.7827632	-8.9946186
beta1	30.9460679	30.9460679	30.9460679	30.9460679	30.9460679
beta2	0.0054272	0.0054272	0.0054272	0.0054272	0.0054272
beta3	0.0022572	0.0022572	0.0022572	0.0022572	0.0022572
beta5	-0.0056540	-0.0055499	-0.0055499	-0.0055499	-0.000260569
e_alpha	1.2877977	0.7251197	0.7251197	0.7251197	0.7546744
e_beta1	0.2223351	0.1502541	0.1502541	0.1502541	0.2727362
e_beta2	0.0189020	0.0313436	0.0313436	0.0313436	0.0176663
e_beta3	0.0214642	0.0315017	0.0315017	0.0315017	0.0268602
e_beta5	0.0319210	0.0179212	0.0179212	0.0179212	0.0184412
p_alpha	1.8198774E-6	5.647444E-14	5.647444E-14	5.647444E-14	3.810323E-14
p_beta1	1.76074E-109	2.94648E-128	2.94648E-128	2.94648E-128	2.478376E-99
p_beta2	0.4862315	0.6471529	0.6471529	0.6471529	0.4644752
p_beta3	0.4484235	0.5678381	0.5678381	0.5678381	0.5167973
p_beta5	0.5625845	0.3625634	0.3625634	0.3625634	0.4949608

Table 5.11. Average Values for our 500 Replicates for t=10

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	3041.08	3098.64	3098.64	3098.64	2975.79
AIC	3048.13	3106.64	3106.64	3106.64	2997.23
AICC	3048.17	3106.68	3106.68	3106.68	2997.49
BIC	3054.88	3114.29	3114.29	3114.29	3017.73
alpha	-9.0514522	-9.0551879	-9.0551879	-9.0551879	-8.9916734
beta1	31.0369501	31.0369501	31.0369501	31.0369501	31.0369501
beta2	-0.000581632	-0.000581632	-0.000581632	-0.000581632	-0.000581632
beta3	0.0023988	0.0023988	0.0023988	0.0023988	0.0023988
beta5	0.000657439	0.000751159	0.000751159	0.000751159	-0.000834688
e_alpha	1.0585338	0.5310884	0.5310884	0.5310884	0.4474552
e_beta1	0.1995601	0.1469233	0.1469233	0.1469233	0.2367521
e_beta2	0.0159472	0.0227940	0.0227940	0.0227940	0.0160064
e_beta3	0.0158007	0.0216051	0.0216051	0.0216051	0.0160532
e_beta5	0.0262224	0.0130400	0.0130400	0.0130400	0.0107902
p_alpha	1.721019E-10	2.654029E-23	2.654029E-23	2.654029E-23	5.441584E-26
p_beta1	4.1839E-113	1.87931E-130	1.87931E-130	4.42163E-130	3.43178E-106
p_beta2	0.5254997	0.6344728	0.6344728	0.2446118	0.5260909
p_beta3	0.5310678	0.6234939	0.6234939	0.2570241	0.5362563
p_beta5	0.6421729	0.4259176	0.4259176	0.4259176	0.5178218

Table 5.12. Average Values for our 500 Replicates for t=12

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	3378.94	3553.84	3553.84	3553.84	3353.22
AIC	3385.83	3561.84	3561.84	3561.84	3375.12
AICC	3385.86	3561.88	3561.88	3561.88	3375.34
BIC	3392.43	3569.49	3569.49	3569.49	3396.05
alpha	-9.1080746	-9.0462745	-9.0462745	-9.0462745	-9.0669617
beta1	31.0112884	31.0112884	31.0112884	31.0112884	31.0112884
beta2	0.000512589	0.000512589	0.000512589	0.000512589	0.000512589
beta3	0.000128370	0.000128370	0.000128370	0.000128370	0.000128370
beta5	0.0012954	0.0011430	0.0011430	0.0011430	0.0016636
e_alpha	1.1815334	0.5919199	0.5919199	0.5919199	0.7719584
e_beta1	0.2035089	0.1232949	0.1232949	0.1232949	0.2421513
e_beta2	0.0120470	0.0196425	0.0196425	0.0196425	0.0122240
e_beta3	0.0094608	0.0178579	0.0178579	0.0178579	0.0095677
e_beta5	0.0292739	0.0146427	0.0146427	0.0146427	0.0189825
p_alpha	3.8648438E-9	1.570601E-19	1.570601E-19	1.570601E-19	4.685902E-13
p_beta1	7.46477E-112	7.43714E-137	7.43714E-137	7.43714E-137	2.20515E-102
p_beta2	0.5033257	0.6548355	0.6548355	0.6548355	0.5086131
p_beta3	0.4796641	0.6652823	0.6652823	0.6652823	0.4830252
p_beta5	0.6534399	0.4175166	0.4175166	0.4175166	0.5627389

Table 5.13. Average Values for our 500 Replicates for t=14

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	3939.06	4171.39	4171.39	4171.39	3869.82
AIC	3946.26	4179.39	4179.39	4179.39	3891.46
AICC	3946.29	4179.42	4179.42	4179.42	3891.64
BIC	3953.15	4187.04	4187.04	4187.04	3912.14
alpha	-8.8891048	-8.8960687	-8.8960687	-8.8960687	-8.9427418
beta1	31.0124512	31.0124512	31.0124512	31.0124512	31.0124512
beta2	0.0014259	0.0014259	0.0014259	0.0014259	0.0014259
beta3	-0.0011435	0.0158957	-0.0011435	-0.0011435	0.0011435
beta5	-0.0029495	-0.0027677	-0.0027677	-0.0027677	-0.0016563
e_alpha	1.1565394	0.5665937	0.5665937	0.5665937	0.4782024
e_beta1	0.1994223	0.1132665	0.1132665	0.1132665	0.2337331
e_beta2	0.0076405	0.0155132	0.0155132	0.0155132	0.0076368
e_beta3	0.0095442	0.000307293	0.0158957	0.0158957	0.0097882
e_beta5	0.0286710	0.0140043	0.0140043	0.0140043	0.0116641
p_alpha	1.670014E-7	1.943024E-17	1.943024E-17	1.943024E-17	3.560225E-22
p_beta1	2.5415E-110	3.2475E-140	3.2475E-140	1.76818E-139	1.12583E-99
p_beta2	0.4788661	0.6873799	0.6873799	0.6873799	0.4783636
p_beta3	0.4940675	0.6576269	0.6576269	0.6576269	0.5016979
p_beta5	0.5927980	0.2950559	0.2950559	0.2950559	0.4507584

As t gets bigger, the parameter estimate of the unstructure tend to be slightly different from those of other correlation structure. In order to investigate which parameter estimates is the best. We decided to vary our number of replicates and see which of the corretion structure produces varying parameters estimates. On doing this, we discovered that the unstructured covariance produces parameter estimates which are slightly different. However, when we increase the sample size to about 1000, the parameter estimates of the unstructured are similar to those of other structures. We conclude that, the unstructured might be overparameterized or yield a flat likelikelihood, so that in each case, parameter estimates with some differences are produced.

Another reason for this might be due to the fact the our JMRE model uses the Newton Raphson algorithm so it might easily converge to a local maximum. Also, despite the fact that the unstructured covariance had the smallest AICC and BIC, the estimates of it standard errors were not always the least as shown in table 5.5 above. Nonetheless, as t gets big the independent structure is more favored to the first order autoregressive.

Furthermore, the unstructured tend to perform worst, in some cases were there was paucity of information that won't support estimation of the order of 1/2t(t+1) different variance and covariances. However, in cases were it converges, it produces least AICC and BIC but their standard errors are not always the smallest as shown above .

5.5.5 Simulating missing data.

Similar to the complete case, missing data were generated using equation (5.3.2.1) above. The cutoff values were set at $y_{ij}^{(1)} > 20$ and $y_{ij}^{(2)} < -9.9$ for some time point for the light missing, that is, missing values ranging from 5% to 10%. For heavy missing, that is missingness in the range 10%-25%, the cutoff values were set at $y_{ij}^{(1)} > 20.8$ and $y_{ij}^{(2)} < -8.8$. This cutoffs were based on previous observation thus leading to RAM missingness. For example for t= 6, this cutoff were set for the third observation leading to missing values for the fourth, fifth and sixth observations.
As for missing data case, we do not just concentrate on their AIC, AICC, BIC and standard error, for choosing the best correlation structure. We also look at the estimates and compare them for consistency relative to the complete cases.

For the cases with 5% to 10% missingness, there was little or no difference as compared to thecomplete. Table below shows average of 500 replicates for t=2 and t=5 respectively are are very similar to the complete case. Similar results were obtained for other values of t.

Table 5.14. Average Values for our 500 Replicates for Light Missing Data

Variables	VC	CS	AR(1)	Toeplitz	UN
LL	753.0942015	775.5174722	775.5174722	775.5174722	719.2288173
AIC	761.0942015	783.5174722	783.5174722	783.5174722	743.2288173
AICC	761.2406130	783.6638837	783.6638837	783.6638837	744.4053029
BIC	768.7422935	791.1655642	791.1655642	791.1655642	766.1730934
alpha	-9.0519590	-9.0341916	-9.0341916	-9.0341916	-9.1179562
beta1	31.2386436	31.2630820	31.2630820	31.2630820	31.2217012
beta2	-0.0124304	-0.0225840	-0.0225840	-0.0225840	-0.0038970
beta3	-0.0123865	-0.0123045	-0.0123045	-0.0123045	-0.0133057
beta5	0.000258310	0.000167173	0.000167173	-0.000167173	0.0020650
e_alpha	1.3361633	0.9182532	0.9182532	0.9182532	0.9333733
e_beta1	0.2542471	0.2260443	0.2260443	0.2260443	0.3127217
e_beta2	0.0672357	0.0928593	0.0928593	0.0928593	0.0736013
e_beta3	0.0799449	0.0963796	0.0963796	0.0963796	0.0949337
e_beta5	0.0329735	0.0225277	0.0225277	0.0225277	0.0226980
p_alpha	4.624668E-7	4.002804E-11	4.002804E-11	4.002804E-11	3.463902E-12
p_beta1	6.43985E-101	1.35405E-105	1.35405E-105	1.35405E-105	1.089667E-91
p_beta2	0.3560214	0.4872401	0.4872401	0.4872401	0.4006887
p_beta3	0.5213506	0.5754326	0.5754326	0.5754326	0.5648049
p_beta5	0.5379178	0.4444352	0.4444352	0.4444352	0.4318931

LL	1344.74	1434.20	1434.20	1434.20	1323.13
AIC	1350.74	1442.20	1442.20	1442.20	1343.13
AICC	1350.79	1442.28	1442.28	1442.28	1343.60
BIC	1356.48	1449.85	1449.85	1449.85	1362.25
alpha	-9.1954411	-9.0411733	-9.0411733	-9.0411733	-9.3220818
beta1	30.9420667	30.9661801	30.9661801	30.9661801	30.9505209
beta2	0.0099266	0.0029635	0.0029635	0.0029635	0.0089504
beta3	0.0106274	0.0167055	0.0167055	0.0167055	0.0146307
beta5	0.0052340	0.0011142	0.0011142	0.0011142	0.0081998
e_alpha	1.2755769	0.8363002	0.8363002	0.8363002	0.9033590
e_beta1	0.2496026	0.1945668	0.1945668	0.1945668	0.2834154
e_beta2	0.0362037	0.0568606	0.0568606	0.0568606	0.0361110
e_beta3	0.0343713	0.0531855	0.0531855	0.0531855	0.0344725
e_beta5	0.0315158	0.0206046	0.0206046	0.0206046	0.0221530
p_alpha	1.2145919E-7	1.774E-15	1.774E-15	1.774E-15	3.675551E-14
p_beta1	3.55951E-107	2.35872E-117	2.35872E-117	2.35872E-117	1.775712E-99
p_beta2	0.5243166	0.7073603	0.7073603	0.7073603	0.5770099
p_beta3	0.5485995	0.6724820	0.6724820	0.6724820	0.4965575
p_beta5	0.4735809	0.3422082	0.3422082	0.3422082	0.2976364

For the cases with heavy missing values, that is , for missing values greater than 10%, as shown below for t=5 and t=7 respectively, the independent yield the most consistent estimate with least AIC, AIC, BIC and standard error. Despite the Unstructured model seems to have the smallest AIC, AICC and BIC, it estimates are less consistence and slightly differs from those of the complete cases.

Table 5.15. Average Values for our 500 Replicates for Heavy Missing Data

Variable	VC	CS	AR(1)	Toeplitz	UN
LL	1240.52	1327.16	1327.16	1327.16	1216.48
AIC	1246.52	1335.16	1335.16	1335.16	1236.68
AICC	1246.58	1335.25	1335.25	1335.25	1237.20
BIC	1252.26	1342.80	1342.80	1342.80	1255.99
alpha	-8.8615627	-8.7986733	-8.7986733	-8.7986733	-9.0000195
beta1	31.0468938	31.0887730	31.0887730	31.0887730	31.0713937
beta2	0.0238820	0.0261552	0.0261552	0.0261552	0.0270299
beta3	0.0069954	0.0247372	0.0247372	0.0247372	0.0305866
beta5	-0.0055759	-0.0078299	-0.0078299	-0.0078299	-0.0044476
e_alpha	1.3385822	0.8664918	0.8664918	0.8664918	0.9270806
e_beta1	0.2542116	0.2025049	0.2025049	0.2025049	0.2840744
e_beta2	0.0363943	0.0610743	0.0610743	0.0610743	0.0360436
e_beta3	0.0373062	0.0578629	0.0578629	0.0578629	0.0370110
e_beta5	0.0332355	0.0214439	0.0214439	0.0214439	0.0228947
p_alpha	3.5115402E-7	5.920223E-11	5.920223E-11	5.920223E-11	2.339829E-10
p_beta1	2.02313E-105	4.59791E-115	4.59791E-115	4.59791E-115	4.932763E-95
p_beta2	0.4459724	0.5958762	0.5958762	0.5958762	0.4495144
p_beta3	0.3952714	0.5461329	0.5461329	0.5461329	0.3851594
p_beta5	0.6224742	0.4708600	0.4708600	0.4708600	0.4198751
LL	1777.90	1850.54	1850.54	1850.54	1756.11
AIC	1784.70	1858.54	1858.54	1858.54	1777.91
AICC	1784.75	1858.60	1858.60	1858.60	1778.34
BIC	1791.21	1866.19	1866.19	1866.19	1798.76
alpha	-9.2027133	-9.1923218	-9.1923218	-9.1923218	-9.2325722
beta1	31.2271603	31.2851555	31.2851555	31.2851555	31.2131445
beta2	-0.0053934	-0.0125191	-0.0125191	-0.0125191	-0.0034191
beta3	0.0059580	0.0074366	0.0074366	0.0074366	0.0068393
beta5	0.0074930	0.0046553	0.0046553	0.0046553	0.0057744
e_alpha	1.0854645	0.6484990	0.6484990	0.6484990	0.7566547
e_beta1	0.2174790	0.1675197	0.1675197	0.1675197	0.2585426
e_beta2	0.0263282	0.0368793	0.0368793	0.0368793	0.0271991
e_beta3	0.0214734	0.0324625	0.0324625	0.0324625	0.0220867
e_beta5	0.0267427	0.0159146	0.0159146	0.0159146	0.0184809
p_alpha	1.206781E-9	2.736209E-19	2.736209E-19	2.736209E-19	2.337936E-16
p_beta1	8.7240E-114	3.46597E-125	3.46597E-125	3.46597E-125	2.61091E-103
p_beta2	0.5491173	0.6503149	0.6503149	0.6503149	0.5849490
p_beta3	0.5391965	0.6709841	0.6709841	0.6709841	0.5550990
p_beta5	0.6404481	0.4867923	0.4867923	0.4867923	0.5077491

a)For data with bivariate outcome, the normality assumptions are very important and should always be verified.

b) The choice of our correlation structure is independent of the correlation structures of our two independent models. (Guinand et el(1999),Ford et el(2003)).

c) The correlation structure depends on the cluster size (Bell et el (2008)).

5.5.6 Rationale for choosing an explicit correlation structure

Following are our recommendations based on the simulation studies.

a)If the observation are collected over time and t=1,2,that is, for data with just the baseline and final readings used the independent correlation structure.

b) For 2<t<6 use the unstructured correlation structure.

c) For smaller sample (about 100 and less), as t increases beyound six i.e t>=6 the independent correlation structure is favored .

d) If your sample size is large (about 500) and 2<t<10 used unstructured correlation structure.

e) If the observations are clustered (not collected over time) then use the compound symmetry if the number of clusters is less than or equals to 9 but if the number of clusters is greater than 9, then use the independent correlation structure.

f) If the data is clustered and also collected over time, run a model with the independent and the compound syymetry, then use the AICC and BIC measure to discern the best correlation structure.

g) For data with missing data, if there less than 10% missingness, analyszed the data as if it were complete. But is missingness is more than 10% use the Independent structure.

However, one great limitation in our simulation strategy is that we have used just five correlation structures, that is, Independent, Compound Symmetry, First Order Autoregressive, Toeplitz and the Unstructured. As we know there are finitely many different serial patterns. This problem seems

to be widely studied in engineering, where two or more signals may overlap or be related and they proposed simulating data using cross correlation term(Guinand et el(1999)).

5.5.7 CONCLUSIONS

Numerous methods have been introduced and discussed on handling correlated data but little work has been done in cases with a bivariate outcomes. However there are many advantages in using bivariate models including: (1)This model allows us to explore variation at different levels of the hierarchy and to model the correlation structure; (2). a shown by Pantazis and Touloumi(2007), the bivariate model achieves the greater bias reduction in all model parameters compared to the two independent models; and (3)there is a greater flexibility in dealing with independent variables. However, the extent to which dependent structure would be considered will depend on the question at hand, i.e. ,the objectives of our analysis. If the interest lies primarily in the population response means and the impact of covariates on these means then a detailed consideration of the correlation structure might not be of significant importance. However, there could be loss of efficiency if the assumed working correlation is far from the true correlation (Gardiner 2009). On the other hand, if we are interested in both marginal and subject-specific inferences, for example, in estimating the growth trajectories of individuals (e.g, Potthoff and Roy (1964)), a careful evaluation regarding the appropriate covariance structure is needed. The linear mixed model can be used for both marginal and subject specific inference, for example on the subject-specific inference mean $E(Y_i \mid X_i, Z_i) = X_i\beta + Z_i\delta$ and the population mean $E(Y_i \mid X_i) = X_i\beta.$ The significance test for this approach depends very highly on our chosen covariance structure . So that we must be careful when choosing it. It is vital to note that the fixed effects estimates with different covariance structures may yield the same values, even though the standard errors of these estimates can widely vary.

A major aim of data analysis using linear mixed model is to define an adequate error covariance structure in order to obtain efficient estimates of regression parameters. However, to properly estimate the covariance structure, the normality assumption of the random effect must be met. Once both conditions are met, various authors (Bock, 1989; Bryk & Raudenbush, 1996; Goldstein, 2003: Hoeksma & Knol, 2001; Plewis, 2001: Raudenbush, 1989; Snijders, 1996) have argued that multilevel models are most suitable for analysis of longitudinal and data with hierarchical structure.

APPENDIX:SAS Results Codes and Simulation Codes

Code for section 3.3.8

```
data data.test1;

set rks1946_pitt_edu_003;  run;

data data.temp;
      set data.test1;
    keep region northeast northcentral west bmi sex poverty poverty1 poverty2
     educ earnings earnings_cat2  earnings_cat1 cigsday male  age
     youngadult   middleage   oldadult     year   marstat   alcamt   alcamt_cat1
      alcamt_cat2 alcamt_cat3 marstat_cat1 marstat_cat2 msasize msasize_cat1
msasize_cat2   alcdaysyr highschool college postgrad racea usborn usbornyes
hispeth nothispanic  white black  time;
if  year< =1996 AND year~=. then delete;
if year=1997 then time=1;
else if year=1998 then time=2;
else if year=1999 then time=3;
else if year=2000 then time=4;
else if year=2001 then time=5;
else if year=2002 then time=6;
else if year=2003 then time=7;
else if year=2004 then time=8;
else if year=2005 then time=9;
else if year=2006 then time=10;
 if poverty in (1,3.99) then poverty1=1; else poverty1=0;
 else if poverty >4.00 then poverty2=1; else poverty2=0;
if region = 1 then northeast=1; else northeast=0;
else if region=2 then northcentral=1; else northcentral=0;
else if region=4 then west=1; else west=0;
if alcamt=0 then delete;
else if alcamt>95 then delete;
else if 7<alcamt<=14 then alcamt_cat1=1; else atcamt_cat1=0;
else if 14<alcamt<=28 then alcamt_cat2=1; else atcamt_cat2=0;
else if alcamt>28 then alcamt_cat3=1; else atcamt_cat1=0;
if age>=99 then delete;
if 25<age<=35 then youngadult=1; else youngadult=0;
else if 35<age<=50 then middleage=1; else middleage=0;
else if age>50 then oldadult=1; else oldadult=0;
if usborn=20 then usbornyes=1; else usbornyes=0;
if earnings >=97 then delete;
if educ >=97 then delete;
if hispeth=10 then nothispanic=1; else nothispanic=0;
if racea=100 then white=1; else white=0;
else if racea=200 then black=1; else black=0;
if educ in (09,10,11,12,13,14,15) then highschool=1; else highschool=0;
else if educ in (16,17,18,19) then college=1; else college=0;
else if educ >=20 then postgrad=1; else postgrad=0;
if earnings in(04,05,06,07,08,09) then earnings_cat1=1; else earnings_cat1=0;
else if earnings in(10,11) then earnings_cat2=1; else earnings_cat2=0;
if  educ>=96  then delete;
if cigsday=0 and cigsday=0 ~=. then delete;
if cigsday=0  then delete;
```

```
if msasize <=10 and msasize~=. then msasize_cat1=1;else msasize_cat1=0;
else if 10<msasize <=11  then msasize_cat2=1;else msasize_cat2=0;
else if 0=<marstat<=4 then marstat_cat1=1;else marstat_cat1=0;
else if 5=<marstat<=7 then marstat_cat2=1;else marstat_cat2=0;
if 0=<sex=1 then male=1;else male=0;
run;

proc sort data =data.temp out=temp1;
     by year;
run;

symbol1 v=dot c=salmon;
 title ‘BMI  1997- 2006 '
proc boxplot data =temp1;
     plot BMI*year;
 run;

symbol1 v=dot c=salmon;
title 'Cigarrete smoked per day 1997- 2006 ';
proc boxplot data =temp1;
    plot cigsday*year;
run;

proc means data=data.temp  mean std alpha=0.05 lclm uclm ;
      class Year;
        var cigsday;
title 'Mean Cigerrate and Confidence';
 title2 '1997-2006';
 run;
 proc means data=data.temp mean std alpha=0.05 lclm uclm ;
   class Year;
     var BMI;
title 'Mean BMI and Confidence';
 title2 '1997-2006';
  run;

 proc univariate data=data.temp;
      var bmi;
          histogram;
           qqplot / normal(mu=est sigma=est);
run;
proc univariate data=data.temp;
  var cigsday;
     histogram;
       qqplot / normal(mu=est sigma=est);
run;

/* arranging data for bivariate analysis*/
data sim_1;
set data.temp;
drop bmi;
var=0;
rename cigsday=y;
run;

data sim_2;
set data.temp;
```

```
drop cigsday;
var=1;
rename bmi=y;
run;

/* transforming data since it look skewed*/
data sim_final;
set sim_1 sim_2;
     logy = log(y+1);
run;

/*Bivariate model*/
proc mixed data=sim_final ;
  class  year;
    model logy = var var*year male youngadult middleage oldadult marstat_cat1
marstat_cat2  poverty1  poverty2              highschool     college  postgrad
usbornyes white black northeast northcentral west/s;
      random  var var*year / type=VC   sub=year G GCORR;
run;

/* Univariate models */
proc mixed data=sim_1;
 class  year;
  model  y  =   male  youngadult  middleage  oldadult  marstat_cat1  marstat_cat2
poverty1 poverty2
    highschool college postgrad usbornyes white black northeast northcentral
west/s;
      random  year / type=VC   sub=year G GCORR;
run;
proc mixed data=sim_2;
 class  year;
   model  y  =  male  youngadult  middleage  oldadult  marstat_cat1  marstat_cat2
poverty1 poverty2
    highschool college postgrad usbornyes white black northeast northcentral
west/s;
     random  year / type=CS   sub=year G GCORR;
run;
```

Simulation codes

T=2

```
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
```

```
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {
1 .4,
.4 1
};
mu = { 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" }; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1 0 ,
0 1
};
mu = { 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/
```

```
data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;


Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;

format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 3;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            output;
      end;
      drop Y11 Y21 Y31 Y12 Y22 Y32;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;
```

```
data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
```

```
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
```

```
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

T=3

```
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {
1 .4 .4,
.4 1 .4,
.4 .4 1
};
mu = {0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31"}; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1 0 0,
0 1 0 ,
0 0 1
```

```
};
mu = {0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
Y32=1-(.1000*(xi+error)) + e32;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 3;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
```

```
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            output;
      end;
      drop Y11 Y21 Y31 Y12 Y22 Y32;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
```

```
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
```

```
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
```

```
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
```

```
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

T=4

```
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {
1 .4 .4 .4,
.4 1 .4 .4,
.4 .4 1 .4,
.4 .4 .4 1
```

```
};
mu = {0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41"}; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1.00 0.40 0.70 0.15,
0.40 1.00 0.39 0.69,
0.70 0.39 1.00 0.40,
0.15 0.69 0.40 1.00
};
mu = {0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32" "e42"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
```

```
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;
Y41=1+(.2098*(xi+error)) + e41;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
Y32=1-(.1000*(xi+error)) + e32;
Y42=1-(.1000*(xi+error)) + e42;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 4;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y12 Y22 Y32 Y42 ;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
```

```
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
```

```
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
```

```
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

T=5

```
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
```

```
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {
1.00 0.40 0.27 0.60  0.7,
0.40 1.00 0.75 0.01  0.4,
0.27 0.75 1.00 0.30  0.2,
0.60 0.01 0.30 1.00  0.4,
0.70 0.40 0.20 0.40  1.0
};
mu = {0 0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51"}; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1    0    0    0    0,
0    1    0    0    0,
0    0    1    0    0,
0    0    0    1    0,
0    0    0    0    1,
};
mu = {0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;
```

```
col={ "e12" "e22" "e32" "e42" "e52"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;
Y41=1+(.2098*(xi+error)) + e41;
Y51=1+(.2098*(xi+error)) + e51;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
Y32=1-(.1000*(xi+error)) + e32;
Y42=1-(.1000*(xi+error)) + e42;
Y52=1-(.1000*(xi+error)) + e52;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 5;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            output;
      end;
```

```
      drop Y11 Y21 Y31 Y41 Y51 Y12 Y22 Y32 Y42 Y52 ;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;
```

```
ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;
```

```
ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
```

```
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;
```

T=6

```
%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {
1.0  0.4  0.4  0.4  0.4  0.4,
0.4  1.0  0.4  0.4  0.4  0.4,
0.4  0.4  1.0  0.4  0.4  0.4,
0.4  0.4  0.4  1.0  0.4  0.4,
0.4  0.4  0.4  0.4  1.0  0.4,
0.4  0.4  0.4  0.4  0.4  1.0
};
```

```
mu = {0 0 0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51" "e61"}; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1.000000 0.40000 0.1600 0.0256 0.01024 0.004096,
0.400000 1.00000 0.4000 0.1600 0.02560 0.010240,
0.160000 0.40000 1.0000 0.4000 0.16000 0.025600,
0.025600 0.16000 0.4000 1.0000 0.40000 0.160000,
0.010240 0.02560 0.1600 0.4000 1.00000 0.400000,
0.004096 0.01024 0.0256 0.1600 0.40000 1.000000
};
mu = {0 0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32" "e42" "e52" "e62"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
```

```
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;
Y41=1+(.2098*(xi+error)) + e41;
Y51=1+(.2098*(xi+error)) + e51;
Y61=1+(.2098*(xi+error)) + e61;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
Y32=1-(.1000*(xi+error)) + e32;
Y42=1-(.1000*(xi+error)) + e42;
Y52=1-(.1000*(xi+error)) + e52;
Y62=1-(.1000*(xi+error)) + e62;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 6;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            else if time = 6 then do; Y1 = Y61; Y2 = Y62; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y51 Y61 Y12 Y22 Y32 Y42 Y52 Y62;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
```

```
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
```

```
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

T=7

```
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
```

```
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {
1    0    0    0    0    0    0,
0    1    0    0    0    0    0,
0    0    1    0    0    0    0,
0    0    0    1    0    0    0,
 0    0    0    0    1    0    0,
 0    0    0    0    0    1    0,
 0    0    0    0    0    0    1
};
mu = {0 0 0 0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51" "e61" "e71"}; /*renames the column
names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1.0  0.4  0.4  0.4  0.4  0.4  0.4,
0.4  1.0  0.4  0.4  0.4  0.4  0.4,
```

```
0.4  0.4  1.0  0.4  0.4  0.4  0.4,
0.4  0.4  0.4  1.0  0.4  0.4  0.4,
0.4  0.4  0.4  0.4  1.0  0.4  0.4,
0.4  0.4  0.4  0.4  0.4  1.0  0.4,
0.4  0.4  0.4  0.4  0.4  0.4  1.0
};
mu = {0 0 0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32" "e42" "e52" "e62" "e72"}; /*renames the column
names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;
Y41=1+(.2098*(xi+error)) + e41;
Y51=1+(.2098*(xi+error)) + e51;
Y61=1+(.2098*(xi+error)) + e61;
Y71=1+(.2098*(xi+error)) + e71;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
Y32=1-(.1000*(xi+error)) + e32;
Y42=1-(.1000*(xi+error)) + e42;
Y52=1-(.1000*(xi+error)) + e52;
```

```
Y62=1-(.1000*(xi+error)) + e62;
Y72=1-(.1000*(xi+error)) + e72;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 7;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            else if time = 6 then do; Y1 = Y61; Y2 = Y62; end;
            else if time = 7 then do; Y1 = Y71; Y2 = Y72; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y51 Y61 Y71 Y12 Y22 Y32 Y42 Y52 Y62 Y72;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
```

```
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;
```

```
%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

T=8

```
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
```

```
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {
1.0  0.4  0.4  0.4  0.4  0.4  0.4  0.4,
0.4  1.0  0.4  0.4  0.4  0.4  0.4  0.4,
0.4  0.4  1.0  0.4  0.4  0.4  0.4  0.4,
0.4  0.4  0.4  1.0  0.4  0.4  0.4  0.4,
0.4  0.4  0.4  0.4  1.0  0.4  0.4  0.4,
0.4  0.4  0.4  0.4  0.4  1.0  0.4  0.4,
0.4  0.4  0.4  0.4  0.4  0.4  1.0  0.4,
0.4  0.4  0.4  0.4  0.4  0.4  0.4  1.0
};
mu = {0 0 0 0 0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51" "e61" "e71" "e81"}; /*renames the column
names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1.0000000 0.4000000 0.160000 0.02560 0.01024 0.004096 0.0016384 0.0006536,
0.4000000 1.0000000 0.400000 0.16000 0.02560 0.010240 0.0040960 0.0016384,
0.1600000 0.4000000 1.000000 0.40000 0.16000 0.025600 0.0102400 0.0040960,
0.0256000 0.1600000 0.400000 1.00000 0.40000 0.160000 0.0256000 0.0102400,
0.0102400 0.0256000 0.160000 0.40000 1.00000 0.400000 0.1600000 0.0256000,
0.0040960 0.0102400 0.025600 0.16000 0.40000 1.000000 0.4000000 0.1600000,
0.0016384 0.0040960 0.010240 0.02560 0.16000 0.400000 1.0000000 0.4000000,
0.0006536 0.0016384 0.004096 0.01024 0.02560 0.160000 0.4000000 1.0000000
};
mu = {0 0 0 0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;
```

```
col={ "e12" "e22" "e32" "e42" "e52" "e62" "e72" "e82"}; /*renames the column
names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;
Y41=1+(.2098*(xi+error)) + e41;
Y51=1+(.2098*(xi+error)) + e51;
Y61=1+(.2098*(xi+error)) + e61;
Y71=1+(.2098*(xi+error)) + e71;
Y81=1+(.2098*(xi+error)) + e81;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
Y32=1-(.1000*(xi+error)) + e32;
Y42=1-(.1000*(xi+error)) + e42;
Y52=1-(.1000*(xi+error)) + e52;
Y62=1-(.1000*(xi+error)) + e62;
Y72=1-(.1000*(xi+error)) + e72;
Y82=1-(.1000*(xi+error)) + e82;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 8;
```

```
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            else if time = 6 then do; Y1 = Y61; Y2 = Y62; end;
            else if time = 7 then do; Y1 = Y71; Y2 = Y72; end;
            else if time = 8 then do; Y1 = Y81; Y2 = Y82; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y51 Y61 Y71 Y81 Y12 Y22 Y32 Y42 Y52 Y62 Y72 Y82;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
```

```
data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
```

```
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
```

```
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

```
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

T=10

```
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;
```

```
sigma = {
1    0    0    0    0    0    0    0    0     0,
0    1    0    0    0    0    0    0    0     0,
0    0    1    0    0    0    0    0    0     0,
0    0    0    1    0    0    0    0    0     0,
0    0    0    0    1    0    0    0    0     0,
0    0    0    0    0    1    0    0    0     0,
0    0    0    0    0    0    1    0    0     0,
0    0    0    0    0    0    0    1    0     0,
0    0    0    0    0    0    0    0    1     0,
0    0    0    0    0    0    0    0    0     1

};
mu = {0 0 0 0 0 0 0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51" "e61" "e71" "e81" "e91" "e101"};
/*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1    0.4    0.4    0.4    0.4    0.4    0.4    0.4    0.4     0.4,
0.4    1    0.4    0.4    0.4    0.4    0.4    0.4    0.4     0.4,
0.4    0.4    1    0.4    0.4    0.4    0.4    0.4    0.4     0.4,
 0.4    0.4   0.4    1    0.4    0.4    0.4    0.4    0.4     0.4,
 0.4    0.4   0.4   0.4    1     0.4    0.4    0.4    0.4     0.4,
 0.4    0.4   0.4   0.4    0.4    1     0.4    0.4    0.4     0.4,
 0.4    0.4   0.4   0.4    0.4   0.4    1      0.4    0.4     0.4,
 0.4    0.4   0.4   0.4    0.4   0.4    0.4    1      0.4     0.4,
 0.4    0.4   0.4   0.4    0.4   0.4    0.4    0.4    1       0.4,
 0.4    0.4   0.4   0.4    0.4    0.4    0.4    0.4    0.4     1

};
mu = {0 0 0 0 0 0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32" "e42" "e52" "e62" "e72" "e82" "e92" "e102"};
/*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
```

```
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;
Y41=1+(.2098*(xi+error)) + e41;
Y51=1+(.2098*(xi+error)) + e51;
Y61=1+(.2098*(xi+error)) + e61;
Y71=1+(.2098*(xi+error)) + e71;
Y81=1+(.2098*(xi+error)) + e81;
Y91=1+(.2098*(xi+error)) + e91;
Y101=1+(.2098*(xi+error)) + e101;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
Y32=1-(.1000*(xi+error)) + e32;
Y42=1-(.1000*(xi+error)) + e42;
Y52=1-(.1000*(xi+error)) + e52;
Y62=1-(.1000*(xi+error)) + e62;
Y72=1-(.1000*(xi+error)) + e72;
Y82=1-(.1000*(xi+error)) + e82;
Y92=1-(.1000*(xi+error)) + e92;
Y102=1-(.1000*(xi+error)) + e102;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 10;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
```

```
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            else if time = 6 then do; Y1 = Y61; Y2 = Y62; end;
            else if time = 7 then do; Y1 = Y71; Y2 = Y72; end;
            else if time = 8 then do; Y1 = Y81; Y2 = Y82; end;
            else if time = 9 then do; Y1 = Y91; Y2 = Y92; end;
            else if time = 10 then do; Y1 = Y101; Y2 = Y102; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y51 Y61 Y71 Y81 Y91 Y101 Y12 Y22 Y32 Y42 Y52 Y62
Y72 Y82 Y92 Y102;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
```

```
data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
```

```
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
```

```
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

```
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

T=12

```
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;
```

```
sigma = {
     1    0    0    0    0    0    0    0    0     0     0     0,
     0    1    0    0    0    0    0    0    0     0     0     0,
     0    0    1    0    0    0    0    0    0     0     0     0,
     0    0    0    1    0    0    0    0    0     0     0     0,
     0    0    0    0    1    0    0    0    0     0     0     0,
     0    0    0    0    0    1    0    0    0     0     0     0,
     0    0    0    0    0    0    1    0    0     0     0     0,
     0    0    0    0    0    0    0    1    0     0     0     0,
     0    0    0    0    0    0    0    0    1     0     0     0,
     0    0    0    0    0    0    0    0    0     1     0     0,
     0    0    0    0    0    0    0    0    0     0     1     0,
     0    0    0    0    0    0    0    0    0     0     0     1
};
mu = {0 0 0 0 0 0 0 0 0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51" "e61" "e71" "e81" "e91" "e101" "e111"
"e121"}; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1  0.4  0.4  0.4   0.4  0.4   0.4   0.4    0.4     0.4   0.4  0.4,
0.4  1  0.4  0.4   0.4  0.4   0.4   0.4    0.4     0.4   0.4  0.4,
0.4  0.4  1  0.4   0.4  0.4   0.4   0.4    0.4     0.4   0.4  0.4,
0.4  0.4 0.4   1   0.4  0.4   0.4   0.4    0.4     0.4   0.4  0.4,
0.4  0.4 0.4 0.4   1    0.4   0.4   0.4    0.4     0.4   0.4  0.4,
0.4  0.4 0.4 0.4  0.4   1     0.4   0.4    0.4     0.4   0.4  0.4,
0.4  0.4   0.4  0.4  0.4  0.4    1   0.4    0.4     0.4   0.4   0.4,
0.4  0.4   0.4  0.4  0.4  0.4  0.4   1     0.4     0.4   0.4   0.4,
0.4  0.4   0.4  0.4  0.4  0.4  0.4   0.4   1       0.4   0.4   0.4,
0.4  0.4   0.4  0.4  0.4  0.4  0.4   0.4   0.4     1     0.4   0.4,
0.4  0.4   0.4  0.4  0.4  0.4  0.4   0.4   0.4     0.4   1     0.4,
0.4  0.4   0.4  0.4  0.4  0.4  0.4   0.4   0.4     0.4   0.4   1

};
mu = {0 0 0 0 0 0 0 0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32" "e42" "e52" "e62" "e72" "e82" "e92" "e102" "e112"
"e122"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
```

```
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;
Y41=1+(.2098*(xi+error)) + e41;
Y51=1+(.2098*(xi+error)) + e51;
Y61=1+(.2098*(xi+error)) + e61;
Y71=1+(.2098*(xi+error)) + e71;
Y81=1+(.2098*(xi+error)) + e81;
Y91=1+(.2098*(xi+error)) + e91;
Y101=1+(.2098*(xi+error)) + e101;
Y111=1+(.2098*(xi+error)) + e111;
Y121=1+(.2098*(xi+error)) + e121;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
Y32=1-(.1000*(xi+error)) + e32;
Y42=1-(.1000*(xi+error)) + e42;
Y52=1-(.1000*(xi+error)) + e52;
Y62=1-(.1000*(xi+error)) + e62;
Y72=1-(.1000*(xi+error)) + e72;
Y82=1-(.1000*(xi+error)) + e82;
Y92=1-(.1000*(xi+error)) + e92;
Y102=1-(.1000*(xi+error)) + e102;
Y112=1-(.1000*(xi+error)) + e112;
Y122=1-(.1000*(xi+error)) + e122;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;
```

```
data atem&j;
      set test4&j;
      do time = 1 to 12;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            else if time = 6 then do; Y1 = Y61; Y2 = Y62; end;
            else if time = 7 then do; Y1 = Y71; Y2 = Y72; end;
            else if time = 8 then do; Y1 = Y81; Y2 = Y82; end;
            else if time = 9 then do; Y1 = Y91; Y2 = Y92; end;
            else if time = 10 then do; Y1 = Y101; Y2 = Y102; end;
            else if time = 11 then do; Y1 = Y111; Y2 = Y112; end;
            else if time = 12 then do; Y1 = Y121; Y2 = Y122; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y51 Y61 Y71 Y81 Y91 Y101 Y111 Y121 Y12 Y22 Y32 Y42
Y52 Y62 Y72 Y82 Y92 Y102 Y112 Y122;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
```

```
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;
```

```
%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

T=14

```
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
```

```
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma ={
1  0.4  0.4   0.4  0.4   0.4  0.4  0.4   0.4    0.4   0.4   0.4   0.4    0.4,
0.4   1 0.4   0.4  0.4   0.4  0.4  0.4   0.4    0.4   0.4   0.4   0.4    0.4,
0.4  0.4  1   0.4  0.4   0.4  0.4  0.4   0.4    0.4   0.4   0.4   0.4    0.4,
0.4  0.4  0.4   1  0.4   0.4  0.4  0.4   0.4    0.4   0.4   0.4   0.4    0.4,
0.4  0.4  0.4  0.4   1   0.4  0.4  0.4   0.4    0.4   0.4   0.4   0.4    0.4,
0.4  0.4  0.4  0.4 0.4    1   0.4  0.4   0.4    0.4   0.4   0.4   0.4    0.4,
0.4  0.4  0.4  0.4 0.4   0.4    1  0.4   0.4    0.4   0.4   0.4   0.4    0.4,
0.4  0.4  0.4  0.4 0.4   0.4  0.4    1   0.4    0.4   0.4   0.4   0.4    0.4,
0.4  0.4  0.4  0.4 0.4   0.4  0.4  0.4    1     0.4   0.4   0.4   0.4    0.4,
0.4  0.4  0.4  0.4 0.4   0.4  0.4  0.4   0.4    1.0   0.4   0.4   0.4    0.4,
0.4  0.4  0.4  0.4 0.4   0.4  0.4  0.4   0.4    0.4     1   0.4   0.4    0.4,
0.4  0.4  0.4  0.4 0.4   0.4  0.4  0.4   0.4    0.4   0.4     1   0.4    0.4,
0.4  0.4  0.4  0.4 0.4   0.4  0.4  0.4   0.4    0.4   0.4   0.4     1    0.4,
0.4  0.4  0.4  0.4 0.4   0.4  0.4  0.4   0.4    0.4   0.4   0.4   0.4     1

};
mu = {0 0 0 0 0 0 0 0 0 0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51" "e61" "e71" "e81" "e91" "e101" "e111"
"e121" "e131" "e141"}; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
 1    0    0    0    0    0    0    0    0     0     0     0     0,
 0    1    0    0    0    0    0    0    0     0     0     0     0,
 0    0    1    0    0    0    0    0    0     0     0     0     0,
 0    0    0    1    0    0    0    0    0     0     0     0     0,
 0    0    0    0    1    0    0    0    0     0     0     0     0,
 0    0    0    0    0    1    0    0    0     0     0     0     0,
 0    0    0    0    0    0    1    0    0     0     0     0     0,
 0    0    0    0    0    0    0    1    0     0     0     0     0,
 0    0    0    0    0    0    0    0    1     0     0     0     0,
 0    0    0    0    0    0    0    0    0     1     0     0     0,
 0    0    0    0    0    0    0    0    0     0     1     0     0,
 0    0    0    0    0    0    0    0    0     0     0     1     0,
```

```
 0    0    0    0    0    0    0    0    0    0    0    0    0    1
};
mu = {0 0 0 0 0 0 0 0 0 0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32" "e42" "e52" "e62" "e72" "e82" "e92" "e102" "e112"
"e122" "e132" "e142"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;
Y41=1+(.2098*(xi+error)) + e41;
Y51=1+(.2098*(xi+error)) + e51;
Y61=1+(.2098*(xi+error)) + e61;
Y71=1+(.2098*(xi+error)) + e71;
Y81=1+(.2098*(xi+error)) + e81;
Y91=1+(.2098*(xi+error)) + e91;
Y101=1+(.2098*(xi+error)) + e101;
Y111=1+(.2098*(xi+error)) + e111;
Y121=1+(.2098*(xi+error)) + e121;
Y131=1+(.2098*(xi+error)) + e131;
Y141=1+(.2098*(xi+error)) + e141;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
```

```
Y32=1-(.1000*(xi+error)) + e32;
Y42=1-(.1000*(xi+error)) + e42;
Y52=1-(.1000*(xi+error)) + e52;
Y62=1-(.1000*(xi+error)) + e62;
Y72=1-(.1000*(xi+error)) + e72;
Y82=1-(.1000*(xi+error)) + e82;
Y92=1-(.1000*(xi+error)) + e92;
Y102=1-(.1000*(xi+error)) + e102;
Y112=1-(.1000*(xi+error)) + e112;
Y122=1-(.1000*(xi+error)) + e122;
Y132=1-(.1000*(xi+error)) + e132;
Y142=1-(.1000*(xi+error)) + e142;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 14;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            else if time = 6 then do; Y1 = Y61; Y2 = Y62; end;
            else if time = 7 then do; Y1 = Y71; Y2 = Y72; end;
            else if time = 8 then do; Y1 = Y81; Y2 = Y82; end;
            else if time = 9 then do; Y1 = Y91; Y2 = Y92; end;
            else if time = 10 then do; Y1 = Y101; Y2 = Y102; end;
            else if time = 11 then do; Y1 = Y111; Y2 = Y112; end;
            else if time = 12 then do; Y1 = Y121; Y2 = Y122; end;
            else if time = 13 then do; Y1 = Y131; Y2 = Y132; end;
            else if time = 14 then do; Y1 = Y141; Y2 = Y142; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y51 Y61 Y71 Y81 Y91 Y101 Y111 Y121 Y131 Y141 Y12
Y22 Y32
Y42 Y52 Y62 Y72 Y82 Y92 Y102 Y112 Y122 Y132 Y142;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
```

```
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
```

```
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
```

```
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=n,iterations=N); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

/********Code that follows an algorithm which simulates data from
Nested Multivariate normal distributions**********/
proc format;
```

```
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {
1 .0,
.0 1
};
mu = { 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" }; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {
1 0 ,
0 1
};
mu = { 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
```

```
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/


X1=1/5+sqrt(1)*rannor(0);
X2=1/6+1/3*sqrt(1)*rannor(0)+sqrt(1)*rannor(0);
age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=X1 + e11;
Y22=X2 + e21;



format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;


data sim_1;
      set test4&j;
      drop Y22;
      var=0;
      rename Y11=y;
run;
data sim_2;
      set test4&j;
      drop Y11;
      var=1;
```

```
      rename Y22=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var  age/s ;
      random var  / type=vc sub=id;

run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 ) drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 ) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 ) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var  age/s ;
      random var  / type=cs sub=id;
      run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 ) drop=_name_);
      var estimate;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 ) drop=_name_
_label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 ) drop=_name_
_label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var  age/s ;
      random var  / type=AR(1) sub=id;
      run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 ) drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 ) drop=_name_
_label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 ) drop=_name_
_label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
```

```
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var age/s ;
      random var  / type=Toep sub=id;

run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 ) drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 ) drop=_name_
_label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 ) drop=_name_
_label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var  age/s ;
      random var  / type=UN sub=id;

run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 ) drop=_name_
_label_);
      var StdErr;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 ) drop=_name_
_label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=50,iterations=500); /*to call macro, sample of n
repeated N times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2  e_alpha e_beta1 e_beta2
p_alpha p_beta1 p_beta2 ;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2  e_alpha e_beta1 e_beta2
p_alpha p_beta1 p_beta2 ;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2  e_alpha e_beta1 e_beta2
p_alpha p_beta1 p_beta2 ;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2  e_alpha e_beta1 e_beta2
p_alpha p_beta1 p_beta2 ;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2  e_alpha e_beta1 e_beta2
p_alpha p_beta1 p_beta2 ;
run;



/*Generating missing data*/

/*Light missing*/
/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;
```

```
%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {1.00 0.40 0.27 0.60  0.7,
 0.40 1.00 0.75 0.01  0.4,
 0.27 0.75 1.00 0.30  0.2,
 0.60 0.01 0.30 1.00  0.4,
0.70 0.40 0.20 0.40  1.0};
mu = {0 0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51"}; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {1    0    0    0    0,
0    1    0    0    0,
0    0    1    0    0,
0    0    0    1    0,
0    0    0    0    1,
};
mu = {0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
```

```
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32" "e42" "e52"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;
Y31=1+(.2098*(xi+error)) + e31;
if (1+(.2098*(xi+error)) + e11)>=20 then do;
      Y41=1+(.2098*(xi+error)) + e41;
end;

if (1+(.2098*(xi+error)) + e11)>=20 then do;
      Y51=1+(.2098*(xi+error)) + e51;
end;


if Y41~=. or Y51~=. then do;
      Y42=1-(.1000*(xi+error)) + e42;
      Y52=1-(.1000*(xi+error)) + e52;
end;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;
Y32=1-(.1000*(xi+error)) + e32;


if Y51=. and Y42<-9.9 then Y42=.;
if Y51=. and Y52<-9.9 then Y52=.;
format sex lsex.;
run;
```

```
proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 5;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y51 Y12 Y22 Y32 Y42 Y52 ;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
```

```
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
```

```
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=50,iterations=500); /*to call macro, sample of 1000
repeated n times*/
```

```
proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

```
/*Heavy missing*/

/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
```

```
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {1.00 0.40 0.27 0.60  0.7,
 0.40 1.00 0.75 0.01  0.4,
 0.27 0.75 1.00 0.30  0.2,
 0.60 0.01 0.30 1.00  0.4,
0.70 0.40 0.20 0.40  1.0};
mu = {0 0 0 0 0};
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51"}; /*renames the column names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {1    0    0    0    0,
0    1    0    0    0,
0    0    1    0    0,
0    0    0    1    0,
0    0    0    0    1,
};
mu = {0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32" "e42" "e52"}; /*renames the column names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
```

```
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
Y21=1+(.2098*(xi+error)) + e21;

if (1+(.2098*(xi+error)) + e31)>=20.8 then do;
      Y31=1+(.2098*(xi+error)) + e41;
end;
if (1+(.2098*(xi+error)) + e41)>=20.8 then do;
      Y41=1+(.2098*(xi+error)) + e41;
end;

if (1+(.2098*(xi+error)) + e51)>=20.6 then do;
      Y51=1+(.2098*(xi+error)) + e51;
end;


if Y41~=. or Y51~=. or Y31~=. then do;
    Y32=1-(.1000*(xi+error)) + e32;
      Y42=1-(.1000*(xi+error)) + e42;
      Y52=1-(.1000*(xi+error)) + e52;
end;

Y12=1-(.1000*(xi+error)) + e12;
Y22=1-(.1000*(xi+error)) + e22;


if Y31=. and Y32<-8.8 then Y32=.;
if Y41=. and Y42<-8.8 then Y42=.;
if Y51=. and Y52<-8.8 then Y52=.;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;

data atem&j;
      set test4&j;
      do time = 1 to 5;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
```

```
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y51 Y12 Y22 Y32 Y42 Y52 ;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
```

```
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
```

```
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
```

```
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=50,iterations=500); /*to call macro, sample of 1000
repeated n times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
```

```
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

/********Code that follows an algorithm which simulates data from
2 Multivariate normal distributions**********/
proc format;
value lsex 0='male'
1='female';

run;


%macro trivariate_normal(sample=, iterations=);
data result;
      set _null_;
run;
data csresult;
      set _null_;
run;
data arresult;
      set _null_;
run;
data tpresult;
      set _null_;
run;
data unresult;
      set _null_;
run;

%do j=1 %to &iterations; /*to iterate*/


PROC IML;

sigma = {1    0    0    0    0    0    0,
   0    1    0    0    0    0    0,
   0    0    1    0    0    0    0,
   0    0    0    1    0    0    0,
   0    0    0    0    1    0    0,
   0    0    0    0    0    1    0,
    0    0    0    0    0    0    1
};
mu = {0 0 0 0 0 0 0};
```

```
n = &sample;
seed = -1;
q=NROW(sigma); /* calculate number of variables */
MUMAT=REPEAT(mu,n,1);/*matrix mumat is created by repeating the row vector is
mu times to produce a nx3 matrix*/
SROOT=ROOT(sigma); /* calculate cholesky root of cov matrix */
Z=NORMAL(REPEAT(seed,n,q)); /* generate normals */
x=Z*SROOT+MUMAT; /*finally the multivariate random variables are created*/ /*
premultiply by cholesky root */

col={ "e11" "e21" "e31" "e41" "e51" "e61" "e71"}; /*renames the column
names*/
CREATE test1 FROM x [colname=col];
APPEND FROM x ;

sigma = {1.0  0.4  0.4  0.4  0.4  0.4  0.4,
 0.4  1.0  0.4  0.4  0.4  0.4  0.4,
 0.4  0.4  1.0  0.4  0.4  0.4  0.4,
 0.4  0.4  0.4  1.0  0.4  0.4  0.4,
 0.4  0.4  0.4  0.4  1.0  0.4  0.4,
 0.4  0.4  0.4  0.4  0.4  1.0  0.4,
 0.4  0.4  0.4  0.4  0.4  0.4  1.0
};
mu = {0 0 0 0 0 0 0};
n = &sample;
seed = -2;
q=NROW(sigma);
MUMAT=REPEAT(mu,n,1);
SROOT=ROOT(sigma);
Z=NORMAL(REPEAT(seed,n,q));
x=Z*SROOT+MUMAT;

col={ "e12" "e22" "e32" "e42" "e52" "e62" "e72"}; /*renames the column
names*/
CREATE test2 FROM x [colname=col];
APPEND FROM x ;

QUIT;
/*PROC UNIVARIATE DATA=test1 NOPRINT;
QQPLOT;
RUN;*/ /*test is good*/

data test3;

set test1;
set test2;
id=_n_;
run;


data test4&j.; /*TO GET EACH DATA SET THAT IS CREATED!!!!!*/
set test3;

/*A normal variate X with mean MU and variance S2 can be generated with this
code:*/

age=40+sqrt(10)*rannor(0); /*dist='N(40,10)'; */
```

```
error= 0.12+sqrt(0.03)*rannor(0); /*dist='N(.12,.03)'; */
sex=ranbin(0,1,.5); /*ranbin(Seed_3,n,p);*/
xi=100+sqrt(20)*rannor(0); /*dist='N(100,20)'; */

Y11=1+(.2098*(xi+error)) + e11;
if (1+(.2098*(xi+error)) + e21)>=20.8 then do;
      Y21=1+(.2098*(xi+error)) + e21;
end;
if (1+(.2098*(xi+error)) + e31)>=20.8 then do;
      Y31=1+(.2098*(xi+error)) + e41;
end;
if (1+(.2098*(xi+error)) + e41)>=20.8 then do;
      Y41=1+(.2098*(xi+error)) + e41;
end;

if (1+(.2098*(xi+error)) + e51)>=20.6 then do;
      Y51=1+(.2098*(xi+error)) + e51;
end;
if (1+(.2098*(xi+error)) + e61)>=20.6 then do;
      Y61=1+(.2098*(xi+error)) + e61;
end;
if (1+(.2098*(xi+error)) + e61)>=20.6 then do;
      Y71=1+(.2098*(xi+error)) + e71;
end;


if Y41~=. or Y51~=. or Y31~=. or Y41~=. or Y51~=. or Y61~=. or Y71~=. then
do;
    Y22=1-(.1000*(xi+error)) + e22;
    Y32=1-(.1000*(xi+error)) + e32;
      Y42=1-(.1000*(xi+error)) + e42;
      Y52=1-(.1000*(xi+error)) + e52;
    Y62=1-(.1000*(xi+error)) + e62;
      Y62=1-(.1000*(xi+error)) + e62;
    Y72=1-(.1000*(xi+error)) + e72;
end;



if Y31=. and Y32<-8.8 then Y32=.;
if Y41=. and Y42<-8.8 then Y42=.;
if Y51=. and Y52<-8.8 then Y52=.;


Y12=1-(.1000*(xi+error)) + e12;

if Y51=. and Y42<-8.8 then Y42=.;
if Y51=. and Y52<-8.8 then Y52=.;
if Y61=. and Y62<-8.8 then Y62=.;
if Y71=. and Y72<-8.8 then Y72=.;
format sex lsex.;
run;

proc freq data=test4&j.; /*to see the distribution of gender*/
table sex;
run;
```

```
data atem&j;
      set test4&j;
      do time = 1 to 7;
            if time = 1 then do; Y1 = Y11; Y2 = Y12; end;
            else if time = 2 then do; Y1 = Y21; Y2 = Y22; end;
            else if time = 3 then do; Y1 = Y31; Y2 = Y32; end;
            else if time = 4 then do; Y1 = Y41; Y2 = Y42; end;
            else if time = 5 then do; Y1 = Y51; Y2 = Y52; end;
            else if time = 6 then do; Y1 = Y61; Y2 = Y62; end;
            else if time = 7 then do; Y1 = Y71; Y2 = Y72; end;
            output;
      end;
      drop Y11 Y21 Y31 Y41 Y51 Y61 Y71 Y12 Y22 Y32 Y42 Y52 Y62 Y72;
run;

data sim_1;
      set atem&j;
      drop Y2;
      var=0;
      rename Y1=y;
run;
data sim_2;
      set atem&j;
      drop Y1;
      var=1;
      rename Y2=y;
run;

data sim_final&j;
      set sim_1 sim_2;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=vc sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
para(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
error(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
pvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
```

```
run;


data estimate;
      set stat;
      set para;
      set error;
      set pvalue;
run;
data result;
      set result estimate;
run;


ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=cs sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cserror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
cspvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;

data estimate;
      set stat;
      set cspara;
      set cserror;
      set cspvalue;
run;
data csresult;
      set csresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=AR(1) sub=id;
      repeated / type=vc grp=var sub=id;
```

```
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
arpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set arpara;
      set arerror
      set arpvalue;
run;
data arresult;
      set arresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j ;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=Toep sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tperror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
tppvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
```

```
run;
data estimate;
      set stat;
      set tppara;
      set tperror
      set tppvalue;
run;
data tpresult;
      set tpresult estimate;
run;

ods output FitStatistics=fit SolutionF=beta;
proc mixed data=sim_final&j;
      class var sex id;
      model y = var var*time age/s ;
      random var var*time / type=UN sub=id;
      repeated / type=vc grp=var sub=id;
run;
proc transpose data = fit out = stat(rename=(col1=LL col2=AIC col3=AICC
col4=BIC) drop=_name_);
      var value;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpara(rename=(col1=alpha col2=beta1 col3=beta2 col4=beta3 col5=beta5)
drop=_name_);
      var estimate;
run;

proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unerror(rename=(col1=e_alpha col2=e_beta1 col3=e_beta2 col4=e_beta3
col5=e_beta5) drop=_name_ _label_);
      var StdErr;
run;
proc transpose data = beta(where=(not (effect='var' & var=1))) out =
unpvalue(rename=(col1=p_alpha col2=p_beta1 col3=p_beta2 col4=p_beta3
col5=p_beta5) drop=_name_ _label_);
      var Probt;
run;
data estimate;
      set stat;
      set unpara;
      set unerror;
      set unpvalue;
run;
data unresult;
      set unresult estimate;
run;

%end;
%mend trivariate_normal; /*to close macro*/
%trivariate_normal(sample=50,iterations=500); /*to call macro, sample of 1000
repeated n times*/

proc means data =result ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
```

```
run;
proc means data =csresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =arresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =tpresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;

proc means data =unresult ;
var LL AIC AICC BIC alpha beta1 beta2 beta3 beta5 e_alpha e_beta1 e_beta2
e_beta3 e_beta5
p_alpha p_beta1 p_beta2 p_beta3 p_beta5;
run;
```

BIBLOGRAPHY

Akaike, H. (1973). Information Theory and an Extension of the Maximum Likelihood Principle, In *2nd International Symposium on Information Theory,* Ed. B.N. Petrov and F. Csaki, pp. 267-281, Budapest: Akademia Kiado.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control,* 19(6): 716–723.

Akaike, H. (1985). Prediction and entropy, In *A Celebration of Statistics,* Ed. A. C. Atkinson and S. E. Fienberg, pp. 1-24, New York: Springer.

Ali-Hussein AL-Marshadi (2007). A New Approach Guide the Selection of the Covariance Structure in Mixed Model. Research Journal of Medicine and Medical Sciences, 2(2):88-97.

Annie-Qu and Lee, J.L. (2008). Model Diagnostic tests for Selecting Informative Correlation Structure in Correlated Data. *Biometrika*, 95, 4, pp. 891-905.

Bagiella, E., Sloan, R.P. and Heitjan, D.F. (2000). Mixed-effects models in psychophysiology. 37, 13-20. Cambridge University Press.

Bell, B.A., Ferron, J.M., Kromrey, J.D. (2008). Cluster Size in Multilevel Models: The Impact of Sparse Data Structures on Point and Interval Estimates in Two-level Models. Section on Survey Methods-JSM.

Bock, R.D. (1989). *Multilevel analysis of educational data*. San Diego, CA: Academic Press.

Bozdogan, H. (1987). Model selection and Akaike's Information Criterion (AIC): the general theory and its analytical extensions. *Psychometrika* 52, 345-370.

Brono, R., Arnau, J. and Balluerka, N. (2007). Using Linear mixed models in Longitudinal Studies: Application of SAS PROC MIXED. Revista Electro'nica de Metrodological Aplicada. Vol. 12, No. 2, pp. 15-31.

Brown, H. and Prescott, R. (2006). Applied Mixed Models in Medicine, 2nd Edition. Statistics In Practice. Wiley and Sons, Ltd. Copyright.

Bryk, A.S. and Raudenbush, S.W. (1992). Hierarchical Linear Models for Social and Behavioural Research: Applications and data analysis methods. Newbury Park, CA: Sage Publications.

Bryk, A.S., Raudenbush, S.W. and Congdon, R.T. (1996). HLM. Hierarchical linear and Nonlinear Modelling with the HLM/2L and HLM/3L programs. Chicago, IL: Scientific Software International.

Burnham, K.P., White, G.C. and Anderson, D.R. (1995). Model Selection Strategy in Analysis of capture Recapture Data. *Biometrics,* 52: 888-898.

Cook, A. and Daponte, B.A. (2008). Demographic Analysis of the Rise in the Prevalence of the US Population Overweight and/or Obese. *Popul Res Policy Rev,* 27:403-426.

De Leeuw, J. and Kreft, I.G.G. (1995). Random coefficient models for multilevel analysis. *Journal of Educational Statistics*, 20:171-190.

De Leeuw, J. and Kreft, I.G.G. (1995). Questioning Multilevel Models. American Educational Research Association and American Statistical Association. Vol. 20, No. 2, Summer, 1995, pp. 171-189.

De Leeuw, J. (2005). Centering in Multilevel analysis. In B.S. Everitt and D.C. Howell, editors, Encyclopedia in Statistics Behavioral Science, pages 247-249. Wiley, New York.

Dempster, A.P., Selwyn, M.S., Patel, C.P. and Roth, A.J. (1984). Statistical and computational aspects of mixed model analysis. *Applied Statistics, 33:203-214.*

Fort, A. et al. (2003). A performance and Complexity Comparison of Auto-correlation and Cross-correlation for ofdm burst synchronization. 0-7803-7663. IEEE.

Gardiner, J.C., Luo, Z. and Roman, L.A. (2009). Fixed effects, random effects and GEE : What are the differences? Statistics in Medicine. *Statist. Med*, 28: 221-239.

Gelman, A. Multilevel (hierarchical) modeling (2006). What it can and cannot do. *Technometrics*, 48: 432-435.

Goldstein, H. (2003). *Multilevel statistical models* (3rd ed.). London: Oxford University Press.

Gomez, B.E., Schaalje, G.B. and Fellingham, G.W. (2005). Performance of Kenward-Roger Method when the Covariance Structure is Selected Using AIC and BIC. Communication in Statistics Simulation and Computation, 34: 377-392.

Guinand, P.S., Kerr, R.W. and Moher, M. (1999). Serial Interference Cancellation for Highly Correlated Users. 0-7803-5582-2/99/IEEE.

Hoeksma, J.B. and Knol, D.L. (2001). Testing Predictive Developmental Hypothesis. Multivariate Behavioral Research, 36: 227-248.

Jindal, A. and Psounis, K. (2005). Modeling Spatially-correlated Data of Sensor Networks with Irregular Topologies. IEEE Xplore.

Khattree, R. and Naik, D.N. (2005). Applied Multivariate Statistics with SAS software, second edition.

King, T., Kavanagh, A.M., Jolley, D., Turrell, G. and Crawford, D. *(2006):* Weight and place: a multilevel cross-sectional survey of area-level social disadvantage and overweight/obesity in Australia. *International Journal of Obesity,* 30: 281-287.

Kristjansson, S.D., Kircher, J.C. and Webb, A.K. (2007). Multilevel models for repeated measures research in psychophysiology. *An introduction to growth curve modeling. Psychophysiology,* 44: 728-736.

Laird, N.M. and Ware, J.H. (1982). Random-effects models for longitudinal data. *Biometrics*, 38: 963-974.

Lin, T.H. and Dayton, C.M. (1997). Model Selection Information Criteria for Nested latent Class models. Journal of Education and Behavioral Statistics., Vol. 22, No. 3, pp. 249-264.

Little, R.J. and Rubin, D.B. (2002). Statistical Analysis with missing data, second edition. Wiley series in probability and statistics.

Little, R.J.A. (1995). Modeling the Drop-Out Mechanism in Repeated-Measure Studies Journal ofAmerican Statistical Association, Vol. 90, No. 431, pp. 1112-1121.

Littell, R.C., Milliken, G.A., Stroup, W.W. and Wolfinger, R.D. (1996). SAS system for mixed models. Cary, NC: SAS Institute.

Longford, N.T. (1987). A fast scoring algorithm for maximum likelihood estimation in unbalance mixed models with nested random effects. *Biometrika,* 74: 817-827.

Longford, N.T. (1995). Random coefficient models, in G. Arminger, C.C Clogg, and M.E Sobel, eds. Handbook of statistical modeling for Social and Behavioral Science. Plenum Press, London, 519-577.

Longford, N.T. (2001). Random Coefficient Models. Oxford University Press, Oxford, UK, 1993. Plewis, I. Explanatory models for relating growth processes. Multivariate Behavioral Research, 36: 207-225.

Mandal, B. and Chern, W.S. (2006). A Multilevel Approach to Model Obesity and Overweight in the United State. *Department of Agriculture, Environmental & Development Economics, The Ohio State University, 2120 Fyffe Road, Columbus, OH 43210, USA.*

McCulloch, C. (2008). Joint modeling of mixed outcome types using latent variables. *Statistics Methods in Medical Research* ; 17:53-73.

Muenz, L.R. and Rubinstein, L.V. (1985). Markov Models for Covariance Dependence of Binary Sequences. *Biometric*, Vol. 41, Issue 191-101.

Ogden, C.L., Carroll, M.D., McDowell, M.A. and Flegal, K.M. (2007). Obesity Among Adults in the United States – no change since 2003—2004. *NCHS data brief no 1. Hyattsville, MD. National Center for Health Statistics.*

Ogden, C.L, Carroll, M.D. and Flegal, K.M. (2008). High Body Mass Index for Age Among US Children and Adolescents, 2003—2006. *JAMA; 299, (20): 2401-2405.*

Pantazis, N. and Touloumi, G. (2007). Fitting bivariate models for longitudinal data with informative drop-outs using MLwiN. *Multilevel Modeling Newsletter Vol. 18, 10-18.*

Pawitan, Y. (2001). In all likelihood: Statistical Modelling and Inference Using likelihood. Oxford Science publications.

Perkins et al. (1987). Weight Gain Associated with Decreases in Smoking Rate and Nicotine Intake. Substance Use & Misued, Vol. 22, No. 6, pp. 575-581.

Prentice, R.L. (1988). Correlated Binary Regression with Covariate Specific to Each Binary Observation. *Biometric,* 44: 1033-1048.

Raman, R. and Hedeker, D. (2005). A mixed-effects regression model for three-level ordinal response data. Statistics in Medicine. *Statist.*, 24: 3331-3345.

Rao, C.R. (1965). The theory of least squares when parameters are stochastic and its application to the analysis of growth curves. *Biometrika 52, 447-458.*

Rao, C.R. (1973). *Linear statistical inference and it application (2nd ed.).* New York:Wiley.

Raudenbush, S.W. (1989). The analysis of longitudinal multilevel data. *International Journal Of Educational Research*, 13: 721-740.

Raudenbush, S.W., Bryk, A.S., Cheong, Y.F. and Congdon, R.T. (2000). HLM5: Hierarchical Linear and Nonlinear Modelling. Lincolnwood, IL: Scientific Software International.

Reinsel, G. (1985). Mean square Error Properties of Emperical Bayes Estimator in Multivariate Random Effects General Linear Model. Journal of the American Statistical Association, Vol. 80, No. 391, Theory and Methods.

Rice, N. and Jones, A. (1997). Multilevel Models and Health Economics. Health Economics, Vol. 6: 561-575.

Rochon, J. (1996). Analyzing Bivariate Repeated Measures for Discrete and Continuous Outcome Variables. *Biometrics,* 52: 740-750.

Rock et al. (2010). U.S. Department of Health and Human Services. *Healthy People 2010)*. 2nd ed. With Understanding and Improving Health and Objectives for Improving Health. 2 vols. Washington, DC: U.S. Government Printing Office.

Sakamoto, Y., Ishiguro, M. and Kitagawa, G. (1986). Akaike information criterion statistics. KTK Scientific Publisher, Tokyo, Japan.

SAS Institute, Inc. (1992). SAS/STAT User's Guide: Version 6(Vol. 2, 4th ed.) Cary, NC:SAS Institute, Inc.

Schwarz, G.E. (1978). Estimating the dimension of a model. *Annals of Statistics* 6(2): 461–464.

Sclove, S.L. (1987). Application of Model selection Criteria to some problems in Multivariate Analysis. *Psychometrika*, Vol. 52, No. 3, pp. 333-343.

Snijders, T. (1996). Analysis of longitudinal data using the hierarchical linear model. Quality and Quantity, 30, 405-426.

Thie'baut,R. et al. (2002). Bivariate Linear Mixed model using SAS proc MIXED. Computing Methods Programs Biomed. 2002 November; 69(3): 249-256.

Verbeke, G. and Molenberghs, G. (Eds.). (1997). Linear mixed models in practice: A SAS oriented approach. New York: Springer-Verlag.

Wilson, M.M. (2002). Cigarette Smoking and Weight Loss in Nursing Home Residents. Online Medicine.

Wolfinger, R.D. (1993). Covariance structure selection in general mixed models. Communications in Statistics, Simulation and Computation, 22, 1079-1106.

Wu, M.C. and Bailey, K.R. (1989). Estimation and Comparison of Changes in the presence of Informative RightCensoring: Conditional Linear Model. *Biometrics*, Vol. 45, No. 3, (Sep. 1989), pp. 939-955.