11-28-2007

Selecting the Working Correlation Structure by a New Generalized AIC Index for Longitudinal Data

Wei-Lun Lin

SELECTING THE WORKING CORRELATION STRUCTURE BY A NEW GENERALIZED AIC INDEX FOR LONGITUDINAL DATA

by

Wei-Lun Lin

Under the Direction of Jiawei Liu

ABSTRACT

The analysis of longitudinal data has been a popular subject for the recent years. The growth of the Generalized Estimating Equation (GEE) Liang & Zeger, 1986) is one of the most influential recent developments in statistical practice for this practice. GEE methods are attractive both from a theoretical and a practical standpoint. In this paper, we are interested in the influence of different "working" correlation structures for modeling the longitudinal data. Furthermore, we propose a new AIC-like method for the model assessment which generalized AIC from the point of view of the data generating. By comparing the difference of the log-likelihood functions between different correlation models, we define the exact $\tilde{n}$ value to create an interval for our model selection. In this thesis, we combine the GEE method and a new generalized AIC Index for the longitudinal data with different correlation structures.

INDEX WORDS: Longitudinal data, Generalized Estimating Equation, Working Correlation, Generalized AIC Index

SELECTING THE WORKING CORRELATION STRUCTURE BY A NEW GENERALIZED AIC INDEX FOR LONGITUDINAL DATA

by

Wei-Lun Lin

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in the College of Arts and Sciences

Georgia State University

2007

SELECTING THE WORKING CORRELATION STRUCTURE BY A NEW GENERALIZED AIC INDEX FOR LONGITUDINAL DATA

by

Wei-Lun Lin

Major Professor:	Jiawei Liu
Committee:	Yu-sheng Hsu
	Yuanhui Xiao

Electronic Version Approved:

Office of Graduate Studies
College of Art and Sciences
Georgia State University
December 2007

ACKNOWLEDGEMENTS

First of all, I would like to acknowledge my thesis advisor Dr. JiaWei Liu for spending her precious time to direct me every week. She always gave me a clear direction to follow and also taught me with her generous patience. Under her guidance, I may feel a lot easier to finish this thesis. In addition, I sincerely appreciate the help of Dr. Yu-sheng Hsu. He helps me to clarify most of the definitions in this thesis.

Furthermore, I want to thank to my classmates who help me on gathering relative information and resources for my thesis. Finally, appreciates for all the people who had ever given me a hand. Thank you all.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter One: Introduction

There was a huge growing interest in the collection of longitudinal data for the last three decades of the 20th century and the statistical analysis of longitudinal data has been the topic of numerous statistical papers in recent years. Several books on the topic have also been published, for example Diggle et al. (1994) and Jones (1993). Such data naturally occur when repeated observations are taken on individuals, or the data is taken on clusters or groups of subjects sharing similar characteristics. One of a great method dealing with longitudinal data is the generalized estimating equations (GEE) method. The GEE method introduced by Liang and Zeger (1986) have been widely used over the past decade to analyze longitudinal data. The method uses a generalized quasi-score function estimate for the regression coefficients, and moment estimated for the correlation parameters. About the model selection, there are a lot of methods we may use. In this paper, we will apply the generalized AIC Index for the best model selection. In other words, we combine the GEE method and the generalized AIC Index for our longitudinal data study.

I organized this thesis in following order. Chapter one is the Introduction, which lists the motivation and the main ideas. Chapter two is the definition of the longitudinal data and the GEE method. The background information is listed on the Chapter three which discuss the Model selection and generalized AIC statistic method. After all the definitions and methods are explained, the simulation study is arranged in Chapter four. Chapter five we use the same method for the real data and test the effect of our methodology. The last chapter is the conclusion which discusses the result of our study and also mention about some relative future researches

Chapter Two: Longitudinal data

A longitudinal study is a correlational research study that involves repeated observations of the same items over long periods of time, often many decades. Longitudinal studies are often used in psychology and biology to study developmental trends across the life span. It may be difficult to develop formal models to summarize trends and covariance, yet there may be rich information in the data.

2.1 Longitudinal Data Analysis

In longitudinal data individuals are repeatedly measured through time which enables the direct study of change (Diggle, Heagerty, Liang & Zeger 2002). Each individual will have certain special characteristics, and measurements on several topics or variables may be taken each time an individual is measured. The reporting times can be different from individual to individual in number, dates and time between reporting. This deviation from equal-spaced, equal quantity time points, producing a ragged time indexing of the data, is common in longitudinal studies and it causes grief for many data analysts. Researchers have done much works on this kind of study. For example, Rao (1965), Grizzle and Allen (1969), and Hui (1984) have discussed methods based on fitting growth curves to the repeated observation for each subject, Fearn (1975) discussed a Bayesian approach to growth curve modeling, Harville (1977) and Laird and Ware (1982) developed random-effects models in which repeated observations for a subject are assumed to share a common random component, Azzalini (1984) discussed models in which autoregressive error structure was assumed where the auto correlation decreases as a geometric function of the time between two observations. Ware (1985) has also presented an overview of linear models for Gaussian longitudinal data. One possible objective of statistical analysis is to describe the marginal expectation of the outcome variable as a function of the covariates while accounting for the correlation among the repeated observations for a given subject. With the outcome variable being approximately Gaussian, a large class of linear models is available for analysis.

2.2 Quasi-Likelihood

Before introducing the Generalized estimating equation, the basic idea of this methodology is Quasi-likelihood approach. Quasi-likelihood was first proposed by Wedderburn (1974) and later examined extensively by McCullagh (1983). It is a methodology for regression that requires few assumptions about the distribution of the dependent variable and hence can be used with a variety of outcomes. In quasi-likelihood, we distinguish only the relationship between the outcome mean and covariates and between the mean and variance. Consider the observations y_{ij} for time t_{ij}, $j = 1,...,n_i$ and subjects $i = 1,...,K$. Here y_{ij} is the outcome variable and x_{ij} is a $p \times 1$ vector of covariates. Let y_i be the $n_i \times 1$ vector $(y_{i1},...,y_{in_i})'$ and x_i be the $n_i \times p$ matrix $(x_{i1},...,x_{in_i})'$ for the *i*th subject. Define u_i to be the expectation of y_i and suppose that

$$u_i = h(x_i\beta)$$

where β is a $p \times 1$ vector of parameters. The inverse of h is referred to as the "link" function (McCullagh and Nelder, 1983). In quasi-likelihood, the variance, v_i, of y_i is expressed as a known function, *g,* of the expectation, u_i, i.e.,

$$v_i = g(u_i)/\phi$$

where ϕ is a scale parameter. Since we only focus on β, ϕ is treated as a nuisance parameter.

The quasi-likelihood estimator is the solution of the score-like equation system

$$S_k(\beta) = \sum_{i=1}^{K} \frac{\partial u_i}{\partial \beta_k} v_i^{-1}(y_i - u_i) = 0, \quad k = 1,..., p \tag{2.1}$$

The solution can be obtained by an iteratively reweighted least squares. The resulting estimator is asymptotically Gaussian under mild regularity conditions (McCullagh, 1983)

2.3 Generalized Estimate Equation Method

The GEE method of Liang and Zeger (1986) is a conceptually and notationally straightforward generalization of quasi-likelihood regression to longitudinal responses. To apply the quasi-likelihood approach to the analysis of longitudinal data, we must consider the mean and covariance of the vector of responses, y_i, for the *i*th subject. Let $R_i(\alpha)$ be the $n_i \times n_i$ working correlation matrix for each y_i, where α is an unknown parameter. Of course the observation times and correlation matrix may differ from subject to subject. The working covariance matrix for y_i is given by

$$Vi = A_i^{1/2} R_i(\alpha) A_i^{1/2} \tag{2.2}$$

where A_i is an $n_i \times n_i$ diagonal matrix with $g(u_{ij})$ as the *j*th diagonal element. We would like estimators that are consistent and have consistent variance estimates even when $R_i(\alpha)$ is incorrect. Our extension of equations (2.1) to the longitudinal data case is given by

$$\sum_{i=1}^{K} D_i' V_i^{-1} S_i = 0 \qquad (2.3)$$

Here $Si = y_i - u_i$ with $u_i = (u_{i1},...,u_{in})'$ and $Di = \frac{\partial u_i}{\partial \beta}$. $Di'Vi^{-1}$ does not depend on the y's generally, so that equations (2.3) converge to 0 and hence have consistent roots as long as $ESi = 0$. For Gaussian outcomes equations (2.3) are the score equations for β. While the estimating equations depends on α as well as β, they can be expressed as a function of β along by first replacing α in equations (2.2) and (2.3) by a $K^{1/2}$-consistent estimator, $\hat{\alpha}(Y,\beta,\phi)$, and then replacing ϕ in $\hat{\alpha}$ by a $K^{1/2}$-consistent estimator, $\hat{\phi}(Y,\beta)$. For any given $R_i(\alpha)$, the estimate, $\hat{\beta}_R$, of β is defined as the solution of

$$\sum_{i=1}^{K} U_i \left\{\beta, \hat{\alpha}\left[\beta, \hat{\phi}(\beta)\right]\right\} = 0 \qquad (2.4)$$

Under mild regularity conditions, Liang and Zeger (1986) show that as $K \to \infty$, $\hat{\beta}_R$ is a consistent estimator of β and that $K^{1/2}(\hat{\beta}_R - \beta)$ is asymptotically multivariate Gaussian with covariance matrix V_R given by

$$\begin{aligned} V_R &= \lim_{K\to\infty} K(\sum_{i=1}^{K} D_i'V_i^{-1}D_i)^{-1}[\sum_{i=1}^{K} D_i'V_i^{-1}\operatorname{cov}(y_i)V_i^{-1}D_i](\sum_{i=1}^{K} D_i'V_i^{-1}D_i)^{-1} \\ &= \lim_{K\to\infty} K(V_1^{-1}V_0V_1^{-1}) \end{aligned} \qquad (2.5)$$

where the covariance of y_i is the actual rather than the assumed covariance. To solve the GEE for $\hat{\beta}_R$, we iteratively solve for the regression coefficients and the correlation and scale parameters, α and ϕ. Given an estimate of $R_i(\alpha)$ and of ϕ, we can calculate an updated estimate of β by iteratively reweighted least squares as described by McCullagh and Nelder (1983). Given an estimate of β, we can calculate standardized residuals, $r_{ij} = (y_{ij} - \hat{u}_{ij}) / \sqrt{[\hat{V}_1^{-1}]_{jj}}$, which are used to consistently estimate α and ϕ. These two steps are iterated until convergence.

In this paper, we specify that a known function of the marginal expectation of the dependent variate is a linear function of the covariates, and assume that the variance is a known function of the mean. In addition, we specify a "working" correlation matrix for the observations for each subject. This set-up leads to generalized estimating equations (GEEs) which give consistent estimators of the regression coefficients and of their variances under weak assumptions about the actual correlation among a subject's observations.

2.4 Gaussian assumption

We apply a simply linear regression $y_{ij_{ij}} = \beta_i x_{ij} + e_{ij}$ to our simulation step in this paper, where *i* is the subject number and *j* is for time point. Our strategy for parameter estimation in the general linear model is to consider simultaneous estimation of the parameter of interest, β, and of the covariance V_0 .parameters, σ^2 and V_0, using the likelihood function, where *V* is a block-diagonal matrix with common non-zero blocks. The general linear model for longitudinal data treat y as a realization of a multivariate Gaussian random vector, Y, with

$$Y \sim MVN(X\beta, \sigma^2 V)$$

In our simulation study progress, we suppose the data is Gaussian distribution, that is, a normal distribution. Since our data may not only have one variable, we use multiple-normal distribution to generate our data.

2.5 Covariance (Correlation) Selection

Since the GEE method is much related to the "working" correlation matrix. We may use different correlation matrices to test the results. There are tons of correlation structures in our Mathematics and Statistics field. In this thesis, we use four common correlation structures such as Independent, Compound symmetric, Toeplitz and Unstructured. For our simulation study, we consider both a 2×2 and a 4×4 correlation models. In a simple 2×2 simulation, we only have two different correlation structures which are independent and Compound symmetric correlation; however, we may have four different correlation structures as in a 4x4 model. All the correlation matrices and parameter number are listed below:

For 2×2 models

Independent

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Parameter number: 0

Compound symmetric

$$\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

Parameter number:1

For 4×4 models:

Independent

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Parameter number: 0

Compound symmetric

$$\begin{bmatrix} 1 & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho \\ \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & 1 \end{bmatrix}$$

Parameter number: 1

Toeplitz

$$\begin{bmatrix} 1 & \rho_1 & \rho_2 & \rho_3 \\ \rho_1 & 1 & \rho_1 & \rho_2 \\ \rho_2 & \rho_1 & 1 & \rho_1 \\ \rho_3 & \rho_2 & \rho_1 & 1 \end{bmatrix}$$

Parameter number: 3

Unstructured

$$\begin{bmatrix} 1 & \rho_1 & \rho_2 & \rho_3 \\ \rho_1 & 1 & \rho_4 & \rho_5 \\ \rho_2 & \rho_4 & 1 & \rho_6 \\ \rho_3 & \rho_5 & \rho_6 & 1 \end{bmatrix}$$

Parameter number: 6

We use these different correlation matrices to be our working correlation matrices in our simulation study.

Chapter Three: Background Information

Model selection is the task of selecting a statistical model from a set of potential models, given data. In its most basic forms, this is one of the fundamental tasks of scientific inquiry. Determining the principle behind a series of observations is often linked directly to a mathematical model predicting those observations.

Goodness-of-fit is generally determined using chi-square statistics. The complexity is generally measured by counting the number of free parameters in the model.

Model selection techniques can be considered as estimators of some physical quantity, such as the probability of the model producing the given data. The bias and variance are both important measures of the quality of this estimator. Asymptotic efficiency is also often considered. A standard example of model selection is that of curve fitting, where, given a set of points and other background knowledge (e.g. points are a result of i.i.d. samples), we must select a function that describes the best curve.

3.1 Model Selection

Lindsay and Liu (2005) emphasize the point of view that the models under consideration are almost always false, if viewed realistically, and so we should analyze model adequacy from that point of view. They investigate this issue in large samples by looking at the Generalized AIC indices, which are designed to serve as one-number summary measures of model adequacy. They also define these index to be the maximum sample size at which samples from the model and those from the true data generating mechanism are nearly indistinguishable. Those definitions lead us to some new ways of viewing models as flawed but useful.

3.2 Generalized AIC Index

Our next context is to find the $\widetilde{n}$. Before testing to generate a Generalized AIC index, we assess the quality of a particular model element m_θ , the one that best approximates the true sampling distribution τ . As an alternative to this approximation question, one could ask how well $m_{\hat{\theta}}$ approximate τ , where $\hat{\theta}$ is an estimator of θ. Because of the randomness of $\hat{\theta}$, the accuracy of this approximation is a random quantity. The testing indices there estimated the AIC of the best independence model where "best" meant using the best parameter values, which are unknown. We may ask: How well will the model, using estimated parameters, fit future samples from τ . With this perspective, Akaike (1974) proposed the AIC index which adds the dimension of the model as a penalty to the negative of the maximized loglikelihood.

$$AIC(M_\delta) = -2\hat{\ell}(m_\delta) + 2k_\delta \tag{3.1}$$

In this section, we notated that $\{\mathrm{M}_\delta : \delta \in \Delta\}$ for a class of models indexed by δ , where each model depends on some finite number $k = k(\delta)$ of real parameters. And k_δ is the number of parameters of the model (dimension), and $\hat{\ell}(m_\delta)$ is the loglikelihood for model m_δ, evaluated at the maximum likelihood estimators. The selection of a model based on this criterion is conventionally done by selecting the model with the smallest value of AIC.

We define the relative risk in using model *M*, together with parameter estimators $\hat{\theta}$, at sample size *n*, to be

$$R(M,n) = -\int \log m_{\hat{\theta}}(x)\tau(x)dx - \int \log \frac{m_{\hat{\theta}}}{m_{\theta}}\tau(x)dx \qquad (3.2)$$

$$= \qquad A \qquad + \qquad B$$

Hence the theoretically best AIC model in $R(M,n)$, could be a false model (if the first term is positive), and could very well depend on n. The theoretically best AIC model M_{δ} in terms of $R(M_{\delta},n)$, minimizes the risk over δ.

Equation A is asymptotically equivalent to $-\frac{\hat{\ell}}{n}+\frac{k}{2n}$ and equation B is asymptotically equivalent to $\frac{k}{2n}$. Consider we simulate samples from model or from true distribution at new sample size $\tilde{n}$, the parameter then estimate from data of size $\tilde{n}$. The second term in (3.2) hence is $\frac{k}{2\tilde{n}}$.

The AIC would depend on sample size *n*. Our generalization of AIC would simply estimate the relative risk at all sample size $\tilde{n}$ via the formula:

$$GAIC = AIC(m_{\delta},\tilde{n}) = [-\frac{\hat{\ell}}{n}+\frac{k}{2n}+\frac{k}{2\tilde{n}}]*2n \qquad (3.3)$$

The conventional AIC estimates the risk at $\tilde{n} = n$. Under this setting we would expect the best model to depend on the choice of $\tilde{n}$. For example, the best model at target sample size $\tilde{n}$ =500 is then a property of the class of models considered and the true data mechanism, but not the de facto sample size *n*.

We now turn the generalized AIC criterion into a sample size index. To simplify matters, we first reduce our attention to the best model of each fixed size. We define the standardized maximum loglikelihood $\hat{\ell}_k$ to be

$\hat{\ell}_k$ = max loglikelihood of all k-dimensional candidate models

And we let $\hat{M}(k)$ represent the best model of size k. In order to find the best AIC model, we minimize over k.

If follows that saying $\hat{M}(k)$ is better than $\hat{M}(k-1)$ is equivalent to

$$\frac{-\hat{\ell}_{k-1}}{n} + \frac{(k-1)}{2n} + \frac{(k-1)}{2\tilde{n}} \geq \frac{-\hat{\ell}_k}{n} + \frac{k}{2n} + \frac{k}{2\tilde{n}}$$

That is,

$$\frac{1}{2\tilde{n}} \leq \frac{\hat{\ell}_k}{n} - \frac{\hat{\ell}_{k-1}}{n} - \frac{1}{2n} \tag{3.4}$$

If the right-handed side in (3.4) is negative, then $\hat{M}(k)$ is worse than $\hat{M}(k-1)$ for all value of $\tilde{n}$. On the other hand, if the right-handed side in (3.4) is positive, we obtain a range of $\tilde{n}$ values for which $\hat{M}(k)$ is better than $\hat{M}(k-1)$. After a little transformation we may find

$$\tilde{n} = n(k-(k-1))/(2(\hat{\ell}_k - \hat{\ell}_{k-1}) + ((k-1)-k)) \tag{3.5}$$

Since the value of the GAIC index $\tilde{n}$ could depend strongly on the test statistic that is being used. If we wish $\tilde{n}$ reflect usefulness of the model, then the test statistic must be sensitive to those model failures which we consider most important. It is obvious that $0 \le \tilde{n} \le n$. If the $\tilde{n}$ computed from (3.5) is out of the range $[0, n]$, it means $\hat{M}(k)$ is not better than $\hat{M}(k-1)$.

We are interested about the result by the combination of GEE and the GAIC. For the next section, we will use this model selection technique combine the GEE method to do the simulation.

Chapter Four: Simulation

In order to test our method, we start with a simple case which is a 2×2 correlation matrix. In other words, there are only two different correlation structures such as independent and Compound symmetric. After the simple simulation, we apply our method to a more complicated case which is a 4×4 correlation matrix. We may have four different correlation structures of this case such as independent, Compound symmetric, Toeplitz and unstructured. The goal for us is to find the best model by comparing the $\tilde{n}$ table and lack of fit curve in each sample size.

4.1 Simulation for 2x2 models

Our starting point is the data in Table 4.1 of Rencher (1995). The data consists of blood glucose measurement (y) at two time points and the glucose measurement one hour after sugar intake (x). We fit a simple linear regression model between y and x for the data using the GEE method with identity link function $(g(u) = u)$, and totally unspecified working correlation structure. We suppose x is a random Gaussian distribution with sample size equal 2, mean equal 100 and 20 is the standard deviation. Then we generate y_{ij} by the equation

$$y_{ij} = 1 + 0.1098 x_i + e_{ij} \qquad (4.1)$$

where $e_i = (e_{i1}, e_{i2})$ is a multivariate random normal distribution with covariance *w*. Furthermore, before testing the method, we need to choose two different correlations to simulate our data, which are close to independent and Compound symmetric correlation.

The first correlation matrix $w_1 = \begin{bmatrix} 1 & 0.01 \\ 0.01 & 1 \end{bmatrix}$, that is a very close to independent correlation. We simulate subjects which have two correlated measurements each at size $n.$=200, 500, 800, 1000. By using the GEE method, we keep updating the "β" in an error range of 10^{-6}. The program will only stop when the difference within new β and old β is less than our error range. Finally, we may construct our GAIC $\tilde{n}$ by computing the Loglikelihood for both models. Follow the equations (3.5), our new $\tilde{n}$ equation is

$$\tilde{n} = n(k1 - k)/(2(ELL - LL) + (k - k1)) \qquad (4.2)$$

where n is the sample size, k and LL represent the parameter number and loglikelihood for Independent case; k1 and ELL represent the same thing but for Compound symmetric case.

We use S-plus software to do the simulation for us; the code is listed in Appendix A. The result is listed as table 4.2.

Table 4.1 $\tilde{n}$ values for correlation structure w_1 $(\rho = 0.01)$

Independent Simulation Model	
Sample size	Independent vs. Compound symmetric
200	-0.001889143
500	-0.00517425
800	-0.009782553
1000	-0.01578773

If the $\tilde{n}$ value is smaller than 0, we may use the correlation matrix with lower parameter numbers as well. Otherwise, we may use the correlation matrix with higher parameter numbers. Follow the table 4.2 we may see that all the $\tilde{n}$ values are negative numbers. This tells us that we may use Independent correlation to be our best model.

After finishing the Independent simulation model, we are interested about Compound symmetric simulation model. We do the same steps with Compound symmetric correlation model for which w=$\begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$. We generate n=200, 500, 800 and 1000 data to do our test. The result is listed as table 4.3

Table 4.2 $\tilde{n}$ values for correlation structure w_2 $(\rho = 0.03)$

Compound symmetric Simulation Model	
Sample size	Independent vs. Compound symmetric
200	-0.01124281
500	-0.02334381
800	-0.02650104
1000	-0.02232003

From the output table, we can see that all the $\tilde{n}$ values are negative numbers. We may make a conclusion, that is, no matter what our sample size is, we still may use the independent correlation models.

4.2 Simulation for 4x4 Models

For a 4×4 correlation matrix, there are several different correlation models, such as independent, Compound symmetric, Toeplitz and unstructured. We first generate a Gaussian distribution of 8000 numbers to be our four times of x variables. The regression equation we used is still $y_{ij} = 1 + 0.1098x_i + e_{ij}$ for $i = 1,...,n$; $j = 1,...,4$. In order to compare the difference between sample sizes, we do the sampling from these 8000 observations. We choose 200, 500, 800 and 1000 to be our four different sample size. Each subject in the sample has four related measurements. Since we have already simulated independent and Compound symmetric correlation structures in 4.1, we just choose another two complex correlation structure models.

4.21 Toeplitz Simulation Models

The starting correlation matrices $w_3 = \begin{bmatrix} 1 & 0.4 & 0.7 & 0.15 \\ 0.4 & 1 & 0.39 & 0.69 \\ 0.7 & 0.39 & 1 & 0.4 \\ 0.15 & 0.69 & 0.4 & 1 \end{bmatrix}$. Then we use GEE method to estimate the equation and find the loglikelihood in each kind of correlation models. In order to apply this correlation matrix to GEE method, we need to make sure our matrix is positive definite. Since $\det(w3) = 0.1723003$, we may continue our simulation steps. The equation (4.1) only compares the independent to the Compound symmetric correlations models. Since we have two more correlation structures right now, we also need two more equation to find $\tilde{n}$ by comparing the Compound symmetric to Toeplitz and Toeplitz to unstructured correlations. The equation (4.3) is used for compute the $\tilde{n}$ value between Compound symmetric to Toeplitz and equation (4.4) is used to compute the $\tilde{n}$ value between Toeplitz to unstructured.

$$\tilde{n} = n(k2 - k1)/(2(CLL - ELL) + (k1 - k2)) \tag{4.3}$$

$$\tilde{n} = n(k3 - k2)/(2(ULL - CLL) + (k2 - k3)) \tag{4.4}$$

where n is the sample size, k1 and ELL represent the parameter number and loglikelihood for Compound symmetric case; k1 and CLL represent the same thing but for Toeplitz case; k2 and ULL are for Unstructured case. The S-plus code is listed in Appendix B. The $\tilde{n}$ result is listed as Table 4.3 and Table 4.4.

Table 4.3 $\tilde{n}$ values for correlation structure w_3

correlation structure w_3			
Sample size	Independent vs. Compound symmetric	Compound symmetric vs. Toeplitz	Toeplitz vs. Untructured
200	1.702304	1.939537	-121.5784
500	1.409049	1.830033	-236.4277
800	1.484285	1.997326	-258.6532
1000	1.384215	1.846223	-136.116

Table 4.4 Best model selection for correlation structure w_3

correlation structure w_3				
Sample size	Independent	Compound symmetric	Toeplitz	Unstructured
200	(0, 1.702304)	(1.702304, 1.939537)	(1.939537, 200)	-
500	(0, 1.409049)	(1.409049, 1.830033)	(1.830033, 500)	-
800	(0, 1.484285)	(1.484285, 1.997326)	(1.997326, 800)	-
1000	(0, 1.384215)	(1.384215, 1.846223)	(1.846223, 1000)	-

Follow Table 4.4, we may divide our conclusion into four different sample sizes.

1) For n=200, one may use independent correlation model when n is less than 1.7; for n between 1.7 to 1.9, we may choose the Compound symmetric correlation model and for sample size n greater than 1.9, the Toeplitz correlation model will be our best selection. The last column of Table 4.3 are all negative numbers, we may skip this comparison.

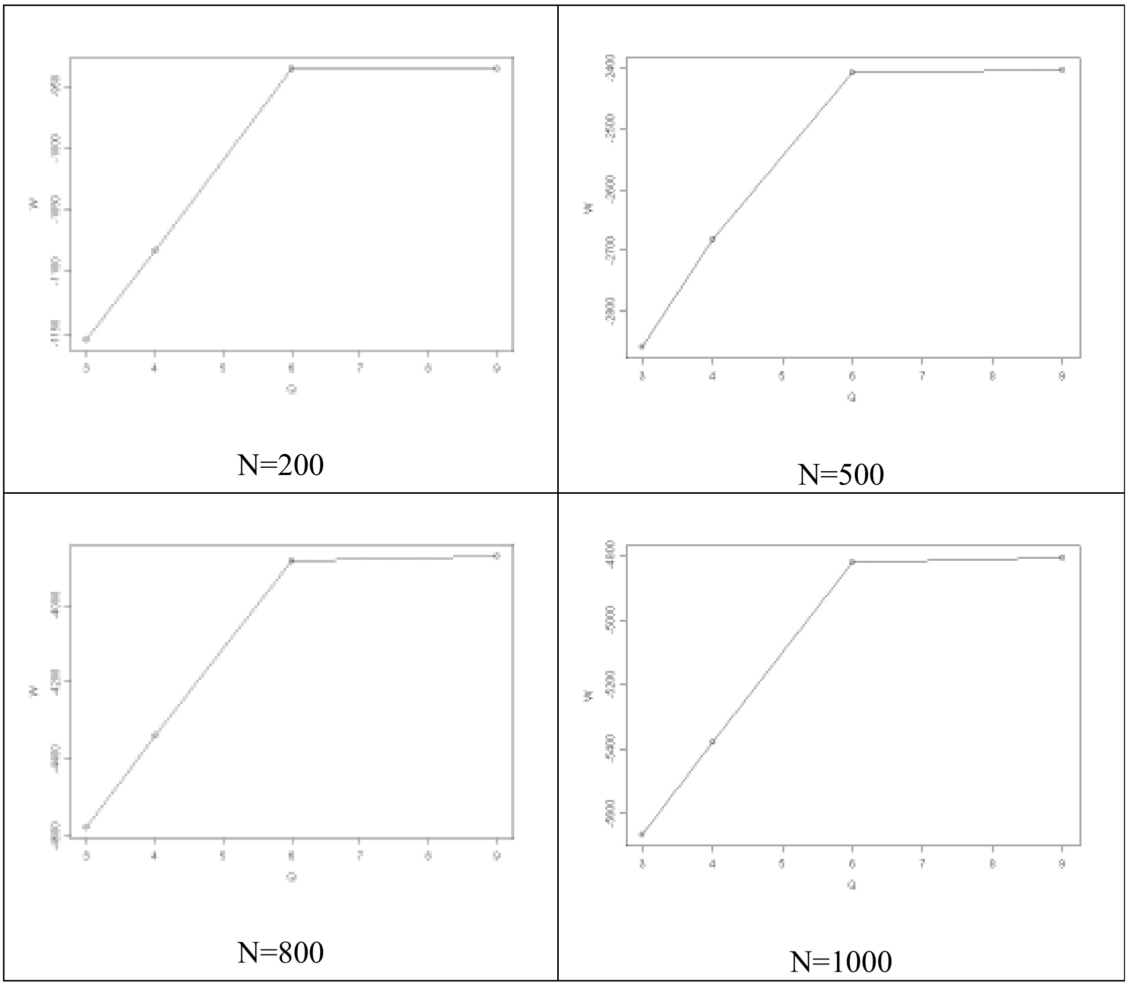
2) For n=500, we may use independent correlation model when n is less than 1.4; for n between 1.4 to 1.8, we may choose the Compound symmetric correlation model and for sample size n greater than 1.8, the Toeplitz correlation model will still be our best selection.

3) For n=800, we may choose independent correlation model when n is less than 1.5; for n between 1.5 to 2, we may use the Compound symmetric correlation model and for sample size n greater than 1.8, we can use the Toeplitz correlation model as well.

4) For n=1000, we may use independent correlation model when n is less than 1.4; for n between 1.4 to 1.8, we may choose the Compound symmetric correlation model and for sample size n greater than 1.8, the Toeplitz correlation model will still be our best selection.

For the overall of the result, we may say that for a very small sample size n less than 1.4, we can just use independent correlation as well; for size n between 1.4 to 2, we need to use the Compound symmetric correlation model; and for n greater than 2, we may use Toeplitz correlation model as well. Furthermore, we can see that this method can effectively find the corresponding correlation structured.

In order to see the change of loglikelihood value in different models, we also plot a graph as Figure 4.1 for connecting all the four likelihood value in the corresponding model. The vertical axis W is the loglikelihood values within four different models and the horizontal axis Q is the number of the parameters.



N=200

N=500

N=800

N=1000

Figure 4.1 The model lack of fit curve for correlation structure w_3

4.22 Unstructured Simulation Models

The last model is the unstructured correlation model. We just change our starting correlation $w_4 = \begin{bmatrix} 1 & 0.4 & 0.27 & 0.6 \\ 0.4 & 1 & 0.75 & 0.01 \\ 0.27 & 0.75 & 1 & 0.3 \\ 0.6 & 0.01 & 0.3 & 1 \end{bmatrix}$. We still need to check our matrix is positive definite. Since $\det(w4) = 0.1288293$, we may continue our simulation steps. Repeat all the rest steps and output our result as Table 4.50, Table 4.6 and Figure 4.2

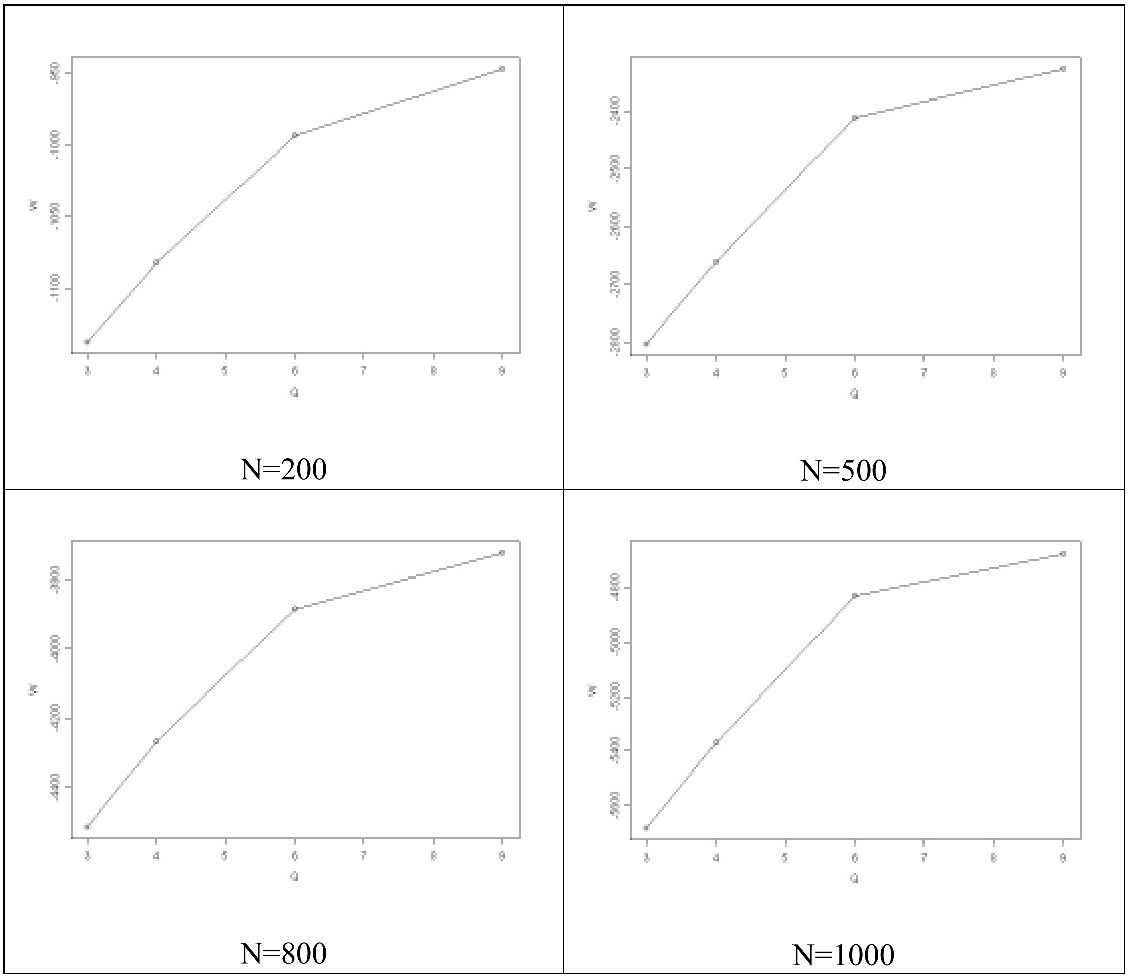
Table 4.5 $\tilde{n}$ values for correlation structure w_4

correlation structure w_4			
Sample size	Independent vs. Compound symmetric	Compound symmetric vs. Toeplitz	Toeplitz vs. Untructured
200	1.815224	2.266745	6.56367
500	1.748966	2.025032	8.973657
800	1.623625	2.09331	7.455487
1000	1.579191	1.8607	9.691426

Table 4.6 Best model selection for correlation structure w_4

correlation structure w_4				
Sample size	Independent	Compound symmetric	Toeplitz	Unstructured
200	(0, 1.815224)	(1.815224, 2.266745)	(2.266745, 6.56367)	(6.56367, 200)
500	(0, 1.748966)	(1.748966, 2.025032)	(2.025032, 8.973657)	(8.973657, 500)
800	(0, 1.623625)	(1.623625, 2.09331)	(2.09331, 7.455487)	(7.455487, 800)
1000	(0, 1.579191)	(1.579191, 1.8607)	(1.8607, 9.691426)	(9.691426, 1000)

For the overall of the result, we may find out that all the $\tilde{n}$ value for the first three correlation structures are very small. Therefore, when n is greater than 10, we may use Unstructured correlation model to be our best model.



N=200

N=500

N=800

N=1000

Figure 4.2 The model lack of fit curve for correlation structure w_4

From Figure 4.2, we can see that there still exist a little difference between the loglikelihood values in each correlation models. To sum up, the $\widetilde{n}$ value has found the corresponding correlation structure in a short time by our simulation result. Take the result table 4.3 for example, we may choose the Toeplitz correlation while n is greater than 2. We could consider this working correlation structure w_3 maybe too close to our Toeplitz correlation. Due to this result, we may test a new correlation which is between Compound symmetric and Toeplitz correlation structures. The correlation structure we test $w_5 = \begin{bmatrix} 1 & 0.1 & 0.1 & 0.05 \\ 0.1 & 1 & 0.07 & 0.08 \\ 0.1 & 0.07 & 1 & 0.1 \\ 0.05 & 0.08 & 0.1 & 1 \end{bmatrix}$. We use the same methodology to test this new correlation structure and construct the $\tilde{n}$ value as table 4.7; furthermore, we put our best model selection as table 4.8.

Table 4.7 $\tilde{n}$ values for correlation structure w_5

correlation structure w_5			
Sample size	Independent vs. Compound symmetric	Compound symmetric vs. Toeplitz	Toeplitz vs. Untructured
200	65.6931	99.30862	-
500	97.6872	143.2394	-
800	91.89772	151.0378	-
1000	90.21411	144.2176	-

Table 4.8 Best model selection for correlation structure w_5

correlation structure w_5				
Sample size	Independent	Compound symmetric	Toeplitz	Unstructured
200	(0, 65.6931)	(65.6931, 99.30862)	(99.30862, 200)	-
500	(0, 97.6872)	(97.6872, 143.2394)	(143.2394, 500)	-
800	(0, 91.89772)	(91.89772, 151.0378)	(151.0378, 800)	-
1000	(0, 90.21411)	(90.21411, 144.2176)	(144.2176, 1000)	-

Under table 4.8, we shall conclude that for a small sample less than 90, we could use the independent correlation model; for sample size between 90 and 145, we shall choose the Compound symmetric correlation model; for sample size greater than 145, we choose Toeplitz correlation as our best model. Comparing the result in table 4.5 and table 4.7, it is obviously that the power of finding the corresponding correlation structure strongly depends on the working correlation we set.

After finishing the simulation study, we may use a real data to prove our method. Chapter five we will put our method into practice, in other words, we use a real data to test our methodology.

Chapter Five: Real data

After the simulation step, we still need to put our methodology into a real data which exist in our true living. In order to test our method, we find a biological data from a hospital. This is a biological data from Weiss (2005) which discuss the relationship between several variables and the systolic blood pressure.

5.1 Data Background

The response variable of this data is the systolic blood pressure (SYS). Observations are taken repeatedly on nurses over the course of a day. This data set has data taken during the first day of participation and during their waking hours. At each blood pressure reading, the nurses also rate their mood on several dimensions and record their posture. A machine records the average number of motions per minute made by the subjects during the preceding 5 minutes, called MNACT5. Also available are phase and day, but they are not to be included in this analysis. Two of the mood variables Happy (HAP) and Stress (STR) are ratings on a 1-5 scale by the subjects of how they feel at the moment that the blood pressure measure is taken. POSTURE is coded as SIT, STAND or RECLINE. MNACT5 should be included in the analysis, but we are not interested in drawing conclusions about it. Family History, FH123, is coded NO, YES, or YESYES if 0, 1 or 2 respectively of the subject’s parents had a history of hypertension. Our study is to describe how the MANCT5, moods, POSTURE, AGE and FH123 affect the systolic blood pressure.

5.2 Result

In order to complete this study, we choose all the six variables such as MANCT5, POSTURE, HAP, STR, AGE and FH123 to be our explanatory variables. Considering the POSTURE and the FH123 variables both contain three categories, we use the dummy variable to separate them to two variables each. The response variable is still the systolic blood pressure (SYS). By the regression method, our equation will be

$$y_{ij} = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 + e_{ij}$$

where all the variables have been coded as Table 5.1

Table 5.1 Summary of variables coding

x_1	Vector with all constant equal 1
x_2	MANCT5
x_3	POSTURE-1(Dummy variable)
x_4	POSTURE-2(Dummy variable)
x_5	Family History 123-1(Dummy variable)
x_6	Family History 123-2(Dummy variable)
x_7	STR
x_8	HAP
x_9	AGE
e_{ij}	Error term

We still apply our GEE method and the Generalized AIC index to this real data, and we randomly choose six different time points to each subject. The $\tilde{n}$ result and the best model selection interval are listed as Table 5.2 and Table 5.3.

Table 5.2 $\tilde{n}$ values for Real Data

Real Data			
Sample size	Independent vs. Compound symmetric	Compound symmetric vs. Toeplitz	Toeplitz vs. Untructured
200	4.718622	17.20437	107.8815

Table 5.3 Best model selection for Real Data

Real Data				
Sample size	Independent	Compound symmetric	Toeplitz	Unstructured
200	(0, 4.718622)	(4.718622, 17.20437)	(17.2043, 107.881)	(107.8815, 200)

From our result table, we may find that for very small sample size n less than 4.7, we shall select independent correlation model; for sample size from 4.7 to 17, we could use the Compound symmetric correlation model; for sample size from 17 to 108, Toeplitz correlation model will be our choice; for the sample size greater than 108, our best model will be the unstructured model. We also plot a graph for the four loglikelihood values via the patameter numbers as Figure 5.1.

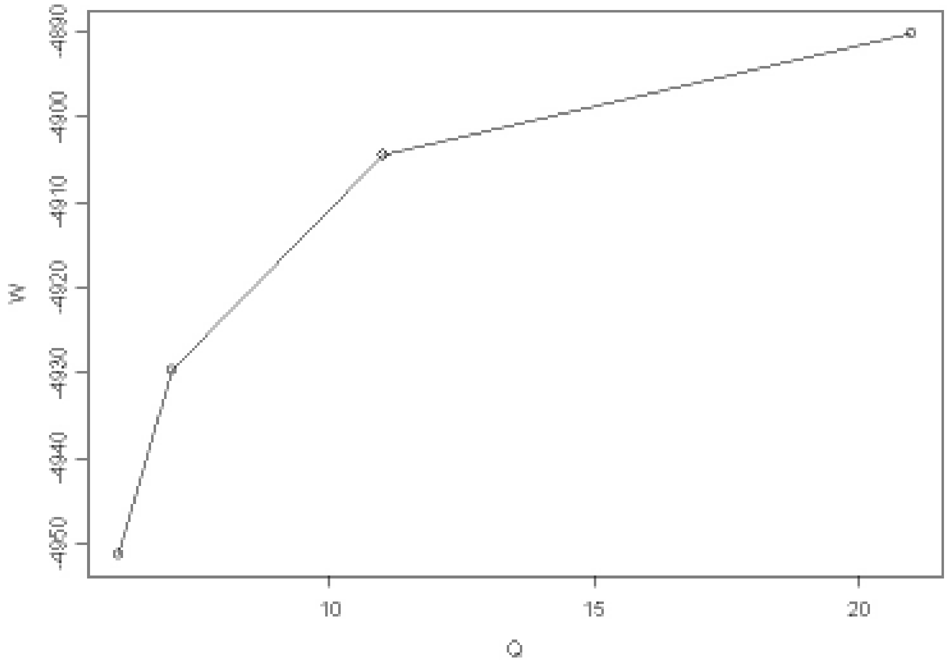


Figure 5.1 The model lack of fit curve for Real Data

From the figure, we can find that our choice will be the Toeplitz correlation model for a specific sample size. For a greater sample size, we still need to use the more complex model such as unstructured correlation model.

Chapter Six: Conclusion and Further research

In summary, our methodology is using GEE method to estimate the longitudinal data and find the exact intervals by $\tilde{n}$ value in Generalized AIC Index for our model selection. This study has found the best model selection for different correlation structures in a longitudinal data. In our simulation result, we have successfully found the specific intervals for model selection in both 2 by 2 and 4 by 4 working correlation models; furthermore, we also apply our algorithm to the real data. The result for the real data is also very consistent as our simulation study. We find the exact $\tilde{n}$ value in each correlation structure and also the best model selection interval. However, we are under the assumption of discrete repeated measurements. The working correlation structure W(t) depends on time point t that is discrete. There are still some time point of measurement t is continuous in our real life. Nonetheless, we need advanced method and techniques to solve those kinds of problems.

References

[1] AKAIKE, H. (1974). A new look at the statistical model identification. System identification and time-series analysis. IEEE *Trans. Automatic Control*, 19 716-724

[2] Azzalini, A. (1984). Estimation and hypothesis testing for collections of autoregressive time series. *Biometrika* 71, 85-90.

[3] Crouder, M. (1995). On the use of a working correlation matrix in using generalized linear models for repeated measures. *Biometrika* 82, 407-410.

[4] Diggle, P. J. (1994). Informative drop-out in longitudinal data analysis. *Applied Statistics* vol. 43, 49-93.

[5] Diggle, P. J., Heagerty, P., Liang, K.Y. and Zeger, S. L. (2002). *The Analysis of Longitudinal data* (2nd ed.). Oxford: Oxford University Press.

[6] Fearn, T. (1975). A bayesian approach to growth curves. *Biometrika* 62. 89-100

[7] Fitzmaurice, G. M. and Laird, N. M. (1993). A likelihood-based method for analyzing longitudinal binary responses. *Biometrika* 80, 141-151.

[8] Fitzmaurice, G. M., Laird, N. M. and Rotinitzky, A. G. (1993). Regression Models for discrete longitudinal responses. *Statistical Science* vol. 8, No. 3, 284-309.

[9] Grizzle, J. E. and Allen, D. M. (1969). Analysis of growth and dose response curves. *Biometrika* 25, 357-381.

[10] Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association* 72, 320-338.

[11] Hui, S. L. (1984). Curve fitting for repeated measurements made at irregular time points. *Biometrika* 40, 691-697.

[12] Jones, R. H. (1993). *Longitudinal data with serial correlation: A state-space approach.* New York: Chapman and Hall.

[13] Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrika* 38, 963-974.

[14] Liang, K. Y. and Zeger, S. L. (1986). Longitudinal data analysis for discrete and continuous outcomes. *Biometrika*, 42, 121-130.

[15] Liang, K. Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, 73, 13-22.

[16] Lindsay, B. and Liu, J (2005) Model assessment tools for a model false world. Technical report. Pennsylvania State University and Georgia State University.

[17] McCullagh, P. (1983). *Generalized Linear Models*. London: Chapman and Hall

[18] McCullagh, P. and Nelder, J. A. (1983). Quasi-likelihood functions. *Annals of Statistics* 11, 59-67.

[19] Rao, C. R (1965). The theory of least squares when the parameters are stochastic and its application to the analysis of growth curves. *Biometrika* 52, 447-458.

[20] Rencher, A. (1995). *Methods of multivariate analysis,* NY

[21] Weiss, R (2005). *Modeling longitudinal data*, Spring-verlag, NY.

[22] Taris, T. W. (2000). *A primer in Longitudinal Data Analysis*. Longon, Thousand Oaks and New Delhi: Sage.

[23] Walls, T. A. and Joseph, L. S. (2006). *Models for intensive longitudinal data.* Oxford: Oxford University Press.

[24] Wedderburn, R. W. M. (1974). Quasi-likelihood functions, generalized linear models, and the Gauss Newton method. *Biometrika* 61, 439-447.

Appendix A: Splus Code(Compound symmetric simulation model with sample size=1000)

```
m=matrix(c(1,0.01,0.01,1),nrow=2,ncol=2)
x=y=e=numeric()

for (i in 1:2000)
{
xi=rnorm(2,100,20)
a=rnorm(2)
ei=a%*%m
yi=0.1098*xi+ei
x=c(x,xi)
y=c(y,yi)
e=c(e,ei)
}

oid=1:2000
ns=1000
id=sample(oid,ns)

Xm=matrix(c(x[2*id-1], x[2*id]),ns,2)
Ym=matrix(c(y[2*id-1],y[2*id]),ns,2)
m1=m2=matrix(c(1,0,0,1), nrow=2,ncol=2)

########################################
# Compound symmetric

B=0
old.B=1
var=1
rho=0
maxit=0

while(abs(old.B-B)>1E-6 & maxit <100)
{
maxit=maxit+1
old.B=B

V=solve(m1)

sum1=0
sum2=0
for (i in 1:ns)
{
sum1=sum1+Xm[i,]%*%V%*%Ym[i,]
sum2=sum2+Xm[i,]%*%V%*%Xm[i,]
}
sum3=solve(sum2)
Betahat=sum1%*%sum3
B=Betahat[1,1]

Rm=(Ym-(Xm*B))
var = sum(Rm^2)/(2*ns-1)

Rm=Rm/sqrt(var)
rho=0
for(i in 1:ns)
{
   rho = rho+Rm[i,1]*Rm[i,2]/(ns-1)
}


m1=matrix(var*c(1,rho,rho,1), nrow=2, ncol=2)

print(c(B,maxit,var, rho))

}
V1=solve(m1)

ALL=0
g=2
for (i in 1:ns)
{
ELL=ELL+sum(-(2*pi)^(g/2)-det(m)^(1/2)-t(Rm[i,])%*%V1%*%Rm[i,])

}

########################################
# Independence

B=0
old.B=1
var=1
rho=0
maxit=0

while(abs(old.B-B)>1E-6 & maxit <100)
{
maxit=maxit+1
old.B=B

V=solve(m2)

sum1=0
sum2=0
for (i in 1:ns)
{
sum1=sum1+Xm[i,]%*%V%*%Ym[i,]
sum2=sum2+Xm[i,]%*%V%*%Xm[i,]
}
sum3=solve(sum2)
Betahat=sum1%*%sum3
B=Betahat[1,1]

Rm=(Ym-(Xm*B))
var = sum(Rm^2)/(2*ns-1)
m2=matrix(var*c(1,0,0,1), nrow=2, ncol=2)
print(c(B,maxit,var, rho))

}

V2=solve(m2)

LL=0
g=2
for (i in 1:ns)
{
LL=LL+sum(-(2*pi)^(g/2)-det(m2)^(1/2)-t(Rm[i,])%*%V2%*%Rm[i,])
}
```

```
LL


k=2
k1=3

nt=ns*(k-k1)/(2*(LL-ELL)+(k1-k))
nt
```

Appendix B: Splus Code(Unstructured simulation model with sample size=1000)

```
m=matrix(c(1,0.4,0.27,0.6,0.4,1,0.75,0.01,0.27,0.75,1,0.3,0.6,0.01,0.3,1),4,4)


x=matrix(0,4*2000,2)
y=rep(0,4*2000)
e=rep(0,4*2000)



for (i in 1:2000)
{
xi=rnorm(4,100,20)
ei=rmvnorm(1, mean=rep(0,4), cov=m, d=4)
yi=1+0.1098*xi+ei
x[(4*i-3):(4*i),1:2]=c(1,1,1,1,xi)
y[(4*i-3):(4*i)]=yi
e[(4*i-3):(4*i)]=ei
}


oid=1:2000
ns=500
id=sample(oid,ns,replace=F)
Xm=matrix(0,4*ns,2)
Ym=rep(0,4*ns)
for(k in 1:ns)
{
   Xm[(4*k-3):(4*k),1:2]=c(x[(4*id[k]-3):(4*id[k]),1:2])
   Ym[(4*k-3):(4*k)]=y[(4*id[k]-3):(4*id[k])]
}



#########################################
# Independence

m1=matrix(c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1), 4,4)
B=c(0,0)
old.B=c(1,1)
var=1
rho=0
maxit=0

while(max(abs(old.B-B))>1E-6 & maxit <100)
{
maxit=maxit+1
old.B=B

V=solve(m1)


sum1=rep(0,2)
sum2=matrix(0,2,2)
for (i in 1:ns)
{
   xi=Xm[(4*(i-1)+1):(4*i),1:2]
   yi=Ym[(4*i-3):(4*i)]
   sum1=sum1+t(xi)%*%V%*%yi
   sum2=sum2+t(xi)%*%V%*%xi
}
B=solve(sum2)%*%sum1


Rm=(Ym-(Xm%*%B))
var = sum(Rm^2)/(4*ns-2)

m1=matrix(var*c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1), 4,4)

print(c(maxit,B,var))

}
R=matrix(Rm,nrow=ns,ncol=4,byrow=T)
V1=solve(m1)
LL=0
g=4
for (i in 1:ns)
{
   LL=LL+sum(-log((2*pi))*g/2-log(det(m1))/2-t(R[i,])%*%V1%*%R[i,]/2)
   }
#########################################################

# Exchangable

m2=matrix(c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1),4,4)
B=c(0,0)
old.B=c(1,1)
var=1
rho=0
maxit=0

while(max(abs(old.B-B))>1E-6 & maxit <100)
{
maxit=maxit+1
old.B=B

V=solve(m2)


sum1=rep(0,2)
sum2=matrix(0,2,2)
for (i in 1:ns)
{
   xi=Xm[(4*(i-1)+1):(4*i),1:2]
   yi=Ym[(4*(i-1)+1):(4*i)]
   sum1=sum1+t(xi)%*%V%*%yi
   sum2=sum2+t(xi)%*%V%*%xi
}
B=solve(sum2)%*%sum1

Rm=(Ym-(Xm%*%B))
var = sum(Rm^2)/(4*ns-2)


rho=0
for(i in 1:ns)
{
   rho = rho+Rm[4*i-3]*Rm[4*i-2]+Rm[4*i-3]*Rm[4*i-1]+Rm[4*i-3]*Rm[4*i]
         +Rm[4*i-2]*Rm[4*i-1]+Rm[4*i-2]*Rm[4*i]+Rm[4*i-1]*Rm[4*i]
}
```

```
rho=rho/var/(6*ns-2)


m2=matrix(var*c(1,rho,rho,rho,rho,1,rho,rho,rho,rho,1,rho,rho,rho,rho,1), 4,4)

print(c(maxit,B,var, rho))

}

R=matrix(Rm,nrow=ns,ncol=4,byrow=T)

V2=solve(m2)
ELL=0
g=4
for (i in 1:ns)
{
   ELL=ELL+sum(-log((2*pi))*g/2-log(det(m2))/2-t(R[i,])%*%V2%*%R[i,]/2)
}

#########################################
# CS

m3=matrix(c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1),4,4)
B=c(0,0)
old.B=c(1,1)
var=1
rho1=0
rho2=0
rho3=0
maxit=0

while(max(abs(old.B-B))>1E-6 & maxit <100)
{
maxit=maxit+1
old.B=B

V=solve(m3)


sum1=rep(0,2)
sum2=matrix(0,2,2)
for (i in 1:ns)
{
   xi=Xm[(4*i-3):(4*i),1:2]
   yi=Ym[(4*i-3):(4*i)]
   sum1=sum1+t(xi)%*%V%*%yi
   sum2=sum2+t(xi)%*%V%*%xi
}
B=solve(sum2)%*%sum1

Rm=(Ym-(Xm%*%B))
var = sum(Rm^2)/(4*ns-2)


rho1=0
rho2=0
rho3=0
for(i in 1:ns)
{
   rho1 = rho1+Rm[4*i-3]*Rm[4*i-2]+Rm[4*i-2]*Rm[4*i-1]+Rm[4*i-1]*Rm[4*i]
   rho2 = rho2+Rm[4*i-3]*Rm[4*i-1]+Rm[4*i-2]*Rm[4*i]
   rho3 = rho3+Rm[4*i-3]*Rm[4*i]
}
rho1=rho1/var/(3*ns-2)
rho2=rho2/var/(2*ns-2)
rho3=rho3/var/(ns-2)

m3=matrix(var*c(1,rho1,rho2,rho3,rho1,1,rho1,rho2,rho2,rho1,1,rho1,rho3,rho2,r
   ho1,1), 4,4)

print(c(maxit,B,var, rho1,rho2,rho3))

}

R=matrix(Rm,nrow=ns,ncol=4,byrow=T)

V3=solve(m3)
CLL=0
g=4
for (i in 1:ns)
{
   CLL=CLL+sum(-log((2*pi))*g/2-log(det(m3))/2-t(R[i,])%*%V3%*%R[i,]/2)
   }

########################################
# Unstructed


m4=matrix(c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1),4,4)
B=c(0,0)
old.B=c(1,1)
var=1
rhovec=rep(0,6)
maxit=0

while(max(abs(old.B-B))>1E-6 & maxit <100)
{
maxit=maxit+1
old.B=B

V4=solve(m4)


sum1=rep(0,2)
sum2=matrix(0,2,2)
for (i in 1:ns)
{
   xi=Xm[(4*i-3):(4*i),1:2]
   yi=Ym[(4*i-3):(4*i)]
   sum1=sum1+t(xi)%*%V%*%yi
   sum2=sum2+t(xi)%*%V%*%xi
}
B=solve(sum2)%*%sum1

Rm=(Ym-(Xm%*%B))
var = sum(Rm^2)/(4*ns-2)

rhovec=rep(0,6)
for(i in 1:ns)
{
   rhovec[1] = rhovec[1]+Rm[4*i-3]*Rm[4*i-2]
   rhovec[2] = rhovec[2]+Rm[4*i-3]*Rm[4*i-1]
```

```
    rhovec[3] = rhovec[3]+Rm[4*i-3]*Rm[4*i]
    rhovec[4] = rhovec[4]+Rm[4*i-2]*Rm[4*i-1]
    rhovec[5] = rhovec[5]+Rm[4*i-2]*Rm[4*i]
    rhovec[6] = rhovec[6]+Rm[4*i-1]*Rm[4*i]
}
rhovec=rhovec/var/(ns-2)

m4=matrix(var*c(1,rhovec[1],rhovec[2],rhovec[3],rhovec[1],1,rhovec[4],rhovec[5
    ],
           rhovec[2],rhovec[4],1,rhovec[6],rhovec[3],rhovec[5],rhovec[6],1),
    4,4)

print(c(maxit,B,var,rhovec))

}

R=matrix(Rm,nrow=ns,ncol=4,byrow=T)

V4=solve(m4)
ULL=0
g=4
for (i in 1:ns)
{
    ULL=ULL+sum(-log((2*pi))*g/2-log(det(m4))/2-t(R[i,])%*%V4%*%R[i,]/2)
    }


##############################################################################
    ###############
#N*

k=3
k1=4
k2=6
k3=9

nt=(k1-k)*ns/(2*(ELL-LL)+(k-k1))
nt1=(k2-k1)*ns/(2*(CLL-ELL)+(k1-k2))
nt2=(k3-k2)*ns/(2*(ULL-CLL)+(k2-k3))

LL
ELL
CLL
ULL
nt
nt1
nt2

Q=c(k,k1,k2,k3)
W=c(LL,ELL,CLL,ULL)

plot(Q,W)
lines(Q,W)
```

Appendix C: Splus Code (Real data)

```
# Real Data

sample(1:20,6)
k=1+8
repn=6
ns=200
indm=matrix(c(1,    0,    0,    0,    0,    0,
              0,    1,    0,    0,    0,    0,
              0,    0,    1,    0,    0,    0,
              0,    0,    0,    1,    0,    0,
              0,    0,    0,    0,    1,    0,
              0,    0,    0,    0,    0,    1), 6, 6)

rhovec=rep(0,15)
corrm=matrix(c(1,      rhovec[1], rhovec[2], rhovec[3], rhovec[4], rhovec[5],
            rhovec[1],        1, rhovec[6], rhovec[7], rhovec[8], rhovec[9],
            rhovec[2],rhovec[6],         1,rhovec[10],rhovec[11],rhovec[12],
            rhovec[3],rhovec[7],rhovec[10],         1,rhovec[13],rhovec[14],
            rhovec[4],rhovec[8],rhovec[11],rhovec[13],         1,rhovec[15],
            rhovec[5],rhovec[9],rhovec[12],rhovec[14],rhovec[15],         1),
   6,6)

print(Real.data2)
Xm=matrix(0,repn*ns,k)
Ym=rep(0,repn*ns)

x1=rep(1,repn*ns)
x2=rep(0,repn*ns)
x3=rep(0,repn*ns)
x4=rep(0,repn*ns)
x5=rep(0,repn*ns)
x6=rep(0,repn*ns)
x7=rep(0,repn*ns)
x8=rep(0,repn*ns)
x9=rep(0,repn*ns)
y=rep(0,repn*ns)


for (i in 1:1200)
{
x2[i]=c(Real.data2[i,3])
x3[i]=c(Real.data2[i,4])
x4[i]=c(Real.data2[i,5])
x5[i]=c(Real.data2[i,6])
x6[i]=c(Real.data2[i,7])
x7[i]=c(Real.data2[i,8])
x8[i]=c(Real.data2[i,9])
x9[i]=c(Real.data2[i,10])
y[i]=c(Real.data2[i,2])
}
X=c(x1,x2,x3,x4,x5,x6,x7,x8,x9)
Xm=matrix(c(X),repn*ns,k)
Ym=matrix(c(y),repn*ns,1)



#########################################
# Independence
```

```
m1=indm

B=rep(0,k)
old.B=rep(1,k)
var=1
rho=0
maxit=0

while(max(abs(old.B-B))>1E-6 & maxit <100)
{

maxit=maxit+1
old.B=B

V=solve(m1)


sum1=rep(0,k)
sum2=matrix(0,k,k)

for (i in 1:ns)
{
    xi=Xm[(repn*(i-1)+1):(repn*i),1:k]
    yi=Ym[(repn*(i-1)+1):(repn*i)]
#   sum1=as.numeric(sum1)
    sum1=sum1+t(xi)%*%V%*%yi
    sum2=sum2+t(xi)%*%V%*%xi
    print(c(i,sum1))
}

B=solve(sum2)%*%sum1


Rm=(Ym-(Xm%*%B))

var = sum(Rm^2)/(repn*ns-k)


m1=matrix(var*indm, repn,repn)

print(c(maxit,B,var))

}

R=matrix(Rm,nrow=ns,ncol=repn,byrow=T)
V1=solve(m1)
LL=0
g=repn
for (i in 1:ns)
{
    LL=LL+sum(-log((2*pi))*g/2-log(det(m1))/2-t(R[i,])%*%V1%*%R[i,]/2)
}
##########################################################

# Exchangable

m2=indm
B=rep(0,k)
old.B=rep(1,k)
var=1
rho=0
```

```
maxit=0

while(max(abs(old.B-B))>1E-6 & maxit <100)
{
maxit=maxit+1
old.B=B

V=solve(m2)



sum1=rep(0,k)
sum2=matrix(0,k,k)
for (i in 1:ns)
{
   xi=Xm[(repn*(i-1)+1):(repn*i),1:k]
   yi=Ym[(repn*(i-1)+1):(repn*i)]
   sum1=sum1+t(xi)%*%V%*%yi
   sum2=sum2+t(xi)%*%V%*%xi
}

B=solve(sum2)%*%sum1

Rm=(Ym-(Xm%*%B))
var = sum(Rm^2)/(repn*ns-k)

rho=0
for(i in 1:ns)
{
   rho = rho+Rm[6*i-5]*Rm[6*i-4]+Rm[6*i-5]*Rm[6*i-3]+Rm[6*i-5]*Rm[6*i-2]
            +Rm[6*i-5]*Rm[6*i-1]+Rm[6*i-5]*Rm[6*i]+Rm[6*i-4]*Rm[6*i-3]
            +Rm[6*i-4]*Rm[6*i-2]+Rm[6*i-4]*Rm[6*i-1]+Rm[6*i-4]*Rm[6*i]
            +Rm[6*i-3]*Rm[6*i-2]+Rm[6*i-3]*Rm[6*i-1]+Rm[6*i-3]*Rm[6*i]
            +Rm[6*i-2]*Rm[6*i-1]+Rm[6*i-2]*Rm[6*i]+Rm[6*i-1]*Rm[6*i]
}
rho=rho/var/(15*ns-k)


corrm=matrix(c(1, rho, rho, rho, rho, rho,
             rho,   1, rho, rho, rho, rho,
             rho, rho,   1, rho, rho, rho,
             rho, rho, rho,   1, rho, rho,
             rho, rho, rho, rho,   1, rho,
             rho, rho, rho, rho, rho,   1), 6,6)


m2=matrix(var*corrm, repn,repn)

print(c(maxit,B,var,rho))

}

R=matrix(Rm,nrow=ns,ncol=repn,byrow=T)

V2=solve(m2)
ELL=0
g=repn
for (i in 1:ns)
{
   ELL=ELL+sum(-log(2*pi)*g/2-log(det(m2))/2-t(R[i,])%*%V2%*%R[i,]/2)}
```

```
#########################################
# CS

m3=indm

B=rep(0,k)
old.B=rep(1,k)
var=1
rho1=0
rho2=0
rho3=0
rho4=0
rho5=0
maxit=0

while(max(abs(old.B-B))>1E-6 & maxit <100)
{
maxit=maxit+1
old.B=B

V=solve(m3)


sum1=rep(0,k)
sum2=matrix(0,k,k)
for (i in 1:ns)
{
   xi=Xm[(repn*(i-1)+1):(repn*i),1:k]
   yi=Ym[(repn*(i-1)+1):(repn*i)]
   sum1=sum1+t(xi)%*%V%*%yi
   sum2=sum2+t(xi)%*%V%*%xi
}

B=solve(sum2)%*%sum1

Rm=(Ym-(Xm%*%B))
var = sum(Rm^2)/(repn*ns-k)


rho1=0
rho2=0
rho3=0
rho4=0
rho5=0

for(i in 1:ns)
{
   rho1 = rho1+Rm[6*i-5]*Rm[6*i-4]+Rm[6*i-4]*Rm[6*i-3]+Rm[6*i-3]*Rm[6*i-2]
            +Rm[6*i-2]*Rm[6*i-1]+Rm[6*i-1]*Rm[6*i]
   rho2 = rho2+Rm[6*i-5]*Rm[6*i-3]+Rm[6*i-4]*Rm[6*i-2]+Rm[6*i-3]*Rm[6*i-
   1]+Rm[6*i-2]*Rm[6*i]
   rho3 = rho3+Rm[6*i-5]*Rm[6*i-2]+Rm[6*i-4]*Rm[6*i-1]+Rm[6*i-3]*Rm[6*i]
   rho4 = rho4++Rm[6*i-5]*Rm[6*i-1]+Rm[6*i-4]*Rm[6*i]
   rho5 = rho5++Rm[6*i-5]*Rm[6*i]
}
rho1=rho1/var/(5*ns-k)
rho2=rho2/var/(4*ns-k)
rho3=rho3/var/(3*ns-k)
rho4=rho4/var/(2*ns-k)
rho5=rho5/var/(ns-k)
```

```
corrm=matrix(c(1, rho1, rho2, rho3, rho4, rho5,
             rho1,   1, rho1, rho2, rho3, rho4,
             rho2, rho1,   1, rho1, rho2, rho3,
             rho3, rho2, rho1,   1, rho1, rho2,
             rho4, rho3, rho2, rho1,   1, rho1,
             rho5, rho4, rho3, rho2, rho1,   1), 6,6)


m3=matrix(var*corrm, repn,repn)

print(c(maxit,B,var,rho1,rho2,rho3,rho4,rho5))

}

R=matrix(Rm,nrow=ns,ncol=repn,byrow=T)

V3=solve(m3)
CLL=0
g=repn
for (i in 1:ns)
{
   CLL=CLL+sum(-log(2*pi)*g/2-log(det(m3))/2-t(R[i,])%*%V3%*%R[i,]/2)
}

#########################################
# Unstructed


m4=indm

B=rep(0,k)
old.B=rep(1,k)
var=1
rhovec=rep(0,15)

maxit=0

while(max(abs(old.B-B))>1E-6 & maxit <100)
{
maxit=maxit+1
old.B=B

V4=solve(m4)

sum1=rep(0,k)
sum2=matrix(0,k,k)
for (i in 1:ns)
{
   xi=Xm[(repn*(i-1)+1):(repn*i),1:k]
   yi=Ym[(repn*(i-1)+1):(repn*i)]
   sum1=sum1+t(xi)%*%V%*%yi
   sum2=sum2+t(xi)%*%V%*%xi
}

B=solve(sum2)%*%sum1

Rm=(Ym-(Xm%*%B))
var = sum(Rm^2)/(repn*ns-k)

rhovec=rep(0,15)
for(i in 1:ns)
```

```
{
   rhovec[1] = rhovec[1]+Rm[6*i-5]*Rm[6*i-4]
   rhovec[2] = rhovec[2]+Rm[6*i-5]*Rm[6*i-3]
   rhovec[3] = rhovec[3]+Rm[6*i-5]*Rm[6*i-2]
   rhovec[4] = rhovec[4]+Rm[6*i-5]*Rm[6*i-1]
   rhovec[5] = rhovec[5]+Rm[6*i-5]*Rm[6*i]
   rhovec[6] = rhovec[6]+Rm[6*i-4]*Rm[6*i-3]
   rhovec[7] = rhovec[7]+Rm[6*i-4]*Rm[6*i-2]
   rhovec[8] = rhovec[8]+Rm[6*i-4]*Rm[6*i-1]
   rhovec[9] = rhovec[9]+Rm[6*i-4]*Rm[6*i]
   rhovec[10] = rhovec[10]+Rm[6*i-3]*Rm[6*i-2]
   rhovec[11] = rhovec[11]+Rm[6*i-3]*Rm[6*i-1]
   rhovec[12] = rhovec[12]+Rm[6*i-3]*Rm[6*i]
   rhovec[13] = rhovec[13]+Rm[6*i-2]*Rm[6*i-1]
   rhovec[14] = rhovec[14]+Rm[6*i-2]*Rm[6*i]
   rhovec[15] = rhovec[15]+Rm[6*i-1]*Rm[6*i]
}

rhovec=rhovec/var/(ns-k)

corrm=matrix(c(      1,rhovec[1], rhovec[2], rhovec[3], rhovec[4], rhovec[5],
             rhovec[1],        1, rhovec[6], rhovec[7], rhovec[8], rhovec[9],
             rhovec[2],rhovec[6],         1,rhovec[10],rhovec[11],rhovec[12],
             rhovec[3],rhovec[7],rhovec[10],         1,rhovec[13],rhovec[14],
             rhovec[4],rhovec[8],rhovec[11],rhovec[13],         1,rhovec[15],
             rhovec[5],rhovec[9],rhovec[12],rhovec[14],rhovec[15],         1),
   6,6)


m4=matrix(var*corrm, repn,repn)

print(c(maxit,B,var,rhovec))

}

R=matrix(Rm,nrow=ns,ncol=repn,byrow=T)

V4=solve(m4)
ULL=0
g=repn
for (i in 1:ns)
{
   ULL=ULL+sum(-log(2*pi)*g/2-log(det(m4))/2-t(R[i,])%*%V4%*%R[i,]/2)
}

print(c(LL,ELL,CLL,ULL))

################################################################################
   ###############
#N*

k=6
k1=7
k2=11
k3=21

nt=ns*(k-k1)/(2*(LL-ELL)+(k1-k))
nt1=ns*(k1-k2)/(2*(ELL-CLL)+(k2-k1))
nt2=ns*(k2-k3)/(2*(CLL-ULL)+(k3-k2))

nt
```

```
nt1
nt2

Q=c(k,k1,k2,k3)
W=c(LL,ELL,CLL,ULL)

plot(Q,W)
lines(Q,W)
```