CONDITIONAL MAXIMUM LIKELIHOOD ESTIMATION FOR LOGISTIC PANEL DATA MODELS WITH NON-RESPONSES

OPEYO PETER OTIENO

MASTER OF SCIENCE

(Mathematics - Statistics Option)

PAN AFRICAN UNIVERSITY

INSTITUTE FOR BASIC SCIENCES TECHNOLOGY AND INNOVATION

2014

CONDITIONAL MAXIMUM LIKELIHOOD ESTIMATION FOR LOGISTIC PANEL DATA MODELS WITH NON-RESPONSES

OPEYO PETER OTIENO

MS300-0005/12

A Thesis submitted to Pan African University, Institute for Basic Sciences Technology and Innovation in partial fulfillment of the requirements for the degree of Master of Science in Mathematics (Statistics Option)

2014

Declaration

This thesis is my original work and has not been submitted to any other university.

Signature:..... Date: .....

Opeyo Peter Otieno

This thesis has been submitted for examination with our approval as University supervisors.

1. Signature:  Date: 06/11/2014

   Dr. Olusanya Elisa Olubusoye

   Department of Statistics, University of Ibadan, Ibadan, Oyo-State, Nigeria

2. Signature: ..... Date:.....

   Prof. Leo O. Odongo

   Department of Statistics and Actuarial Science, Kenyatta University, Nairobi, Kenya

Dedication

I dedicate this work to my Mum Carren Anyango, my Uncle Meshack Oduor, my loving wife Lilian Amondi and my beloved Daughters Sheryl Clare Atieno and Metrine Valary Akinyi.

Acknowledgements

I would, first and foremost, like to thank the Almighty God, the provider, for giving me the gift of life, knowledge and wisdom to successfully pursue this study.

That this study is courtesy of a scholarship, my appreciation goes to the African Union Commission for granting me the opportunity to study M.Sc. Mathematics at the Pan African University Institute for Basic Sciences, Technology and Innovation (PAUISTI). Through their continuous financial support, this study has succeeded.

Special acknowledgments go to my course supervisors Dr. O. E. Olubusoye and Prof. Leo O. Odongo for their dedicated guidance and support throughout the research. They put me back on track when I was falling out more so on time management during the course.

I would like to take this opportunity to acknowledge the important role of Prof. Kinyanjui Mathew and Mr. Henry Kissinger for their guidance and motivation in the struggle of researching in this field.

I wish to thank the PAUISTI administration and staff for their committed support during the whole study period. Without forgetting my classmates in Statistics Option and all the pioneer students of PAUISTI for the determination, devotion and support towards this achievement, I sincerely say thank you to you all.

I acknowledge the continued encouragement and support from my relatives with special thanks going to my esteemed uncle Meshack Oduor for setting the grounds for my education this far. May the Almighty reward you abundantly.

Special acknowledgment also goes to my parents Carren and Charles, for their prayers, encouragement and the sacrifice they made to inculcate a strong foundation for education in my childhood.

Lastly, I would like to acknowledge my dear wife Lilian and daughter Sheryl who missed me while I was away on study leave. Your understanding and strong will gave me the zeal to sail through the academic voyage.

Table of Contents

Declaration ..... ***ii***

Dedication ..... ***iii***

Acknowledgements ..... ***iv***

List of Tables ..... ***x***

List of Figures ..... ***xi***

List of Symbols, Abbreviations and Acronyms ..... ***xii***

Abstract ..... ***xiii***

CHAPTER 1 ..... ***1***

INTRODUCTION 1

1.1 Background Introduction 1

1.1.1 Panel Data 1

1.1.2 Panel Data Model 2

1.2 Statement of the Problem 4

1.3 Research Questions 5

1.4 Objectives 5

1.4.1 General Objective 5

1.4.2 Specific Objectives 6

1.5 Justification for the Study 6

1.6 Organization of the Thesis 7

CHAPTER 2 ..... ***8***

LITERATURE REVIEW 8

2.1 Introduction 8

2.2 The Review of Literature 8

2.3 Research Gap 11

2.4 Conclusion 11

CHAPTER 3 12

BINARY CHOICE PANEL DATA MODELS 12

3.1 Introduction 12

3.1.1 Binary Choice Variable 12

3.2 The Linear Probability Model 12

3.2.1 The Model 12

3.2.2 Weaknesses of the Linear Probability Model 13

3.3 Logistic Model for Binary Response 14

3.3.1 The Model 14

3.3.2 Assumptions of the Logistic Model 15

3.4 Estimation of Logistic Model 17

3.4.1 Incidental Parameter Problem 17

3.4.2 The Unconditional Likelihood Function 18

3.4.3 Conditional Likelihood Function for Logistic Panel Data Model 18

3.5 Newton-Raphson Algorithm 21

3.6 Unbalanced Panels 21

3.6.1 Causes of Unbalancedness 21

3.6.2 Nonresponses in Panel Data 22

3.7 Techniques for Dealing with Missing Data 24

3.7.1 Deletion Procedures 24

3.7.2 Replacement Procedures 25

3.7.3 Model-based Procedures 27

CHAPTER 4 29

METHODOLOGY AND DATA ANALYSIS 29

4.1 Introduction 29

4.2 Parameter Estimation 29

4.3 Monte Carlo Simulation 32

4.4 Discussion 45

CHAPTER 5 47

SUMMARY, CONCLUSION AND RECOMMENDATION 47

5.1 Summary 47

5.2 Conclusion 48

5.3 Recommendations 48

REFERENCES 49

APPENDIX 53

A1. R CODES FOR MONTE CARLO SIMULATION 53

A1.1 n= 50 with 10% missingness 53

A1.2 n= 50 with 30% missingness.....60

A1.3 n= 100 with 10% missingness.....67

A1.4 n= 100 with 30% missingness.....74

A1.5 n= 250 with 10% missingness.....81

A1.6 n= 250 with 30% missingness.....88

List of Tables

Table 4.1: Description of variables 33

Table 4.2: Simulation values for n= 50, T=2, percentage of missingness=10%.....34

Table 4.3: Simulation values for n= 50, T=2, percentage of missingness=30%.....35

Table 4.4: Simulation values for n= 100, T=2, percentage of missingness=10%.....36

Table 4.5: Simulation values for n= 100, T=2, percentage of missingness=30%.....37

Table 4.6: Simulation values for n= 250, T=2, percentage of missingness=10%.....38

Table 4.7: Simulation values for n= 250, T=2, percentage of missingness=30%.....39

List of Figures

Figure 4.1: Median Bias for 10% missingness with varying sample sizes 40

Figure 4.2: Median Bias variation across estimators.....41

Figure 4.3: MAD variation across estimators 42

Figure 4.4: Mean Bias variation across estimators.....43

Figure 4.5: RMSE variation across estimators.....44

List of Symbols, Abbreviations and Acronyms

N	Number of observations or units in the population
T	Time period
T_i	Time period for the i-th unit
n	Number of Units in sample
y_{it}	Response variable of the i-th unit at time t
X	Explanatory (independent variable)
β	Model parameter
$\hat{\beta}$	Parameter estimator
c_i	Individual effects of the i-th unit
FE	Fixed Effect
RE	Random Effect
LPM	Linear Probability Model
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimation
OLS	Ordinary Least Squares
iid	Independent and Identically Distributed
$\bar{y}_R$	Estimate of a population mean based on using just the cases responding in the sample.
N_{NR}	Total number of non-responders in the population
$\bar{Y}_R$	Population average for responders in the population
$\bar{Y}_{NR}$	Population average for non-responders in the population
MAD	Median Absolute Deviation
RMSE	Root Mean Square Error

Abstract

In analyzing most survey data in which the dependent variable is a binary choice variable taking values 1 or 0 for success or failure respectively it is feasible to consider the conditional probabilities of the dependent variable. Under strict exogeneity, this conditional probability equals the expected value of the dependent variable. This treatment calls for a nonlinear function which will ensure that the conditional probability lies between 0 and 1 and such functions yield the probit model and the logit model. For panel data econometrics, such nonlinear panel data models require conditioning the probabilities on the minimal sufficient statistic for the fixed effects so as to curb the incidental parameter problem. Solving the joint probability distribution function by maximum likelihood method yields consistent '*conditional maximum likelihood estimate'* for the model parameters in cases when the data set is complete (or balanced) with no cases of missing observations. In cases of missing observations in the covariates, researchers employ several imputation techniques to make the data complete. Imputation, however, brings about a bias in the covariate and this bias is propagated to the parameter estimates. This study considers the susceptibility of nonlinear logistic panel data model with single fixed effects to imputation by investigating the bias arising from various imputation methods. The study developed a conditional maximum likelihood estimator for nonlinear binary choice logistic panel data model in the presence of missing observations. A Monte Carlo simulation was designed to determine the magnitude of bias arising from some common imputation techniques and recommend better techniques to be used in order to improve model performance in the presence of missing observations in econometrics panel data analysis. The simulation results show that the parameter estimates for the conditional logistic model are less biased than those from the unconditional logistic model without sacrificing on the precision. Mean imputation and median imputation preserve precision of the parameter estimates better than last value carried forward.

CHAPTER 1

INTRODUCTION

This chapter gives the background of the study by introducing the concept of panel data econometrics and mentions various approaches to panel data models' estimation where the conditional maximum likelihood estimation is mentioned as a solution to the incidental parameter problem in logistic panel data model. It also gives the problem statement, research questions, objectives and justification of the study.

1.1 Background Introduction

1.1.1 Panel Data

Panel data, also called longitudinal data or cross-sectional time series data, are data where multiple cases (individuals, firms etc) were observed over two or more time periods. In Panel data, there are two kinds of information: the cross-sectional information reflected in the differences between individuals, and the time-series or within-subject information reflected in the changes within individuals over time. Panel data accounts for such individual heterogeneity by allowing us to control for variables which we cannot observe or measure or individual specific time invariant variables that change over time but not across individuals.

When there is some unknown variable or variables that cannot be controlled for that affect the dependent variable the estimates of coefficients derived from regression may be subject to omitted variable bias. Consequently, the use of ordinary multiple regression techniques on panel

data such as OLS may not be ideal. As such, Panel data regression techniques allow us to adequately take advantage of the different types of information available in a panel dataset.

1.1.2 Panel Data Model

A general panel data model is of the form

$$y_{it} = \boldsymbol{x_{it}\beta} + c_i + \gamma_t + u_{it}, \quad i = 1,\ldots,\mathrm{N};\ \mathrm{t} = 1,\ldots,\mathrm{T} \tag{1.1}$$

where the parameters c_i and γ_t represent the individual specific and time specific effects respectively. Assuming only the individual specific effects c_i then the equation (1.1) takes the form

$$y_{it} = \boldsymbol{x_{it}\beta} + c_i + u_{it} \tag{1.2}$$

The relationship between c_i and u_{it} determines whether the relation (1.2) is treated as fixed or random effects model. This is to say that if c_i is correlated with $\boldsymbol{x_{it}}$then the model has only u_{it} as the stochastic part and c_i is treated as fixed (non-random). Consequently, we have a fixed effect panel data model. Otherwise, it is a random effect model, if c_i becomes part of the stochastic part of (2) so that

$$y_{it} = \boldsymbol{x_{it}\beta} + v_{it} \tag{1.3}$$

where $v_{it} = c_i + u_{it}$. Equation (1.3) is the random effect model.

In estimating panel data model parameters, therefore, there exist generally two categories of models, fixed effects models (FE) and random effects (RE) models. With the former, one does not estimate the effects of the variables that are individual specific and time invariant but rather controls for them or 'partials them out'. The latter (RE models) estimate the effects of these time invariant variables. These estimates may be biased since other omitted variables are uncontrolled for.

If the dependent variable y_{it} is continuous then the parameters in panel data model can be estimated. The approaches used so far in estimating panel data models with fixed effects aim at

controlling for these effects by eliminating their presence from the model and estimating the coefficients of the regressors. If on the other hand the dependent variable is categorical, then specific nonlinear functions that preserve the structure of the dependent variable are considered. Such nonlinear functions include among others, the logit, probit and Poisson models. Among the approaches explored to estimate fixed effects models include:

- Demeaning variables- where the within subject means (averages) are subtracted from each observed value of the variables. This ensures that the constant nuisance factor for each subject attains a value 0 (zero) for each case and do not therefore appear in any further analysis. This approach is known to work best for linear regression models but fails in logistic regression.
- Unconditional maximum likelihood - here dummy variables are created for each subject (except one) and included in the model i.e. N-1 dummies introduced. For large N, estimating N-1 dummy coefficients and k explanatory coefficients becomes too tedious and time consuming yet our interest may not be in the so many coefficients produced. Estimating linear regressions by unconditional maximum likelihood produces consistent estimates with the demeaning variables method but for logistic regressions, these estimates are biased.
- Conditional maximum likelihood estimation – this is the most preferred method for logistic regressions. Here, the conditional maximum likelihood 'conditions' the fixed effects out of the likelihood function (Chamberlain, 1980). This is done by conditioning the likelihood function on the total number of events observed for each subject.

The concepts of conditional maximum likelihood for nonlinear panel data models has been tackled in several studies from cases with only a single fixed effect to multiple fixed effects. For static linear models, consistent estimates for the parameters are obtained by simply differencing

out the fixed effects. For nonlinear panel data models, however, there exist the well-known incidental parameter problem realized by Neyman and Scott (1948) in which the number of fixed effects increases with increasing sample size. Incidental parameters are such parameters whose dimension increases with sample size. For example, as n approaches infinity, the number of fixed effects increases and so they are incidental parameters. Such parameters cannot be consistently estimated (Baltagi, 2001). Other attempts to solve the incidental parameter problem by Hausman, Hall and Griliches (1984) succeeded for the Poisson and negative binomial models with single fixed effects. Manski (1987) generalised the logit model and developed a conditional maximum score estimation of binary response models.

As much as parameter estimation of panel data models is possible, complications arise when the loss of efficiency is desired for the panels that are unbalanced (Matyas and Lovrics, 1991). Such unbalancedness in panel data is brought about by delayed entry, early exit or intermittent non-response from a study unit. For the former two causes of unbalancedness, each individual is observed T_i times and analysis of the panel data models is still feasible. However, in cases of intermittent non-responses a need to establish the nature and cause of the non-response suffices. Approaches suggested in literature on how to handle missing observations become valid in such cases. This study therefore examines the impact of missing data on the conditional maximum likelihood estimation procedures in nonlinear panel data models for discrete choice dependent variable. Using simulations with various types of missing data, we shall attempt to recommend the best techniques to be used in order to improve the treatment of missing data.

1.2 Statement of the Problem

In order to estimate the model parameters in a logistic likelihood function, the complete data set [**Y X**] should be available where **Y** is an NTx1 matrix of zeros and ones while **X** is an NTxk matrix of explanatory variables. With instances of missing observations due to intermittent

nonresponses in either **Y** or **X**, the attempt to complete the data vector [**Y X**] through imputation introduces a bias on the sample mean to the specific variable. As such, this bias is propagated to the estimated parameters, $\widehat{\boldsymbol{\beta}}$.

How much biased the parameter estimates $\widehat{\boldsymbol{\beta}}$ would be due to nonresponse is thus worth investigating. There is, however, limited literature on nonresponse bias especially for panel data models and specifically binary response models. As such, econometricians need to reckon with the important aspects of imputation techniques which may minimize the bias in the parameter estimates thereby increasing efficiency of the estimation procedure adopted. This study therefore delves into the problem of non-response bias induced in the conditional maximum likelihood estimation of nonlinear panel data models (specifically for the logistic panel data model).

1.3 Research Questions

Following the problem statement above, the following research questions are germane to this study:

1. How does one estimate parameters for logistic panel data models in the presence of non-responses in the explanatory variables?
2. What is the magnitude of the bias introduced when values are imputed for nonresponses in a nonlinear (logistic) panel data model?

1.4 Objectives

This section outlines the general objective and specific objectives of the study.

1.4.1 General Objective

To estimate logistic panel data models by conditional maximum likelihood estimation in the presence of imputed missing observations due to nonresponses.

1.4.2 Specific Objectives

1. To derive the conditional maximum likelihood estimators of the parameters for logistic panel data models in the presence of missing observations.
2. To evaluate the magnitude of bias arising from using common imputation techniques to replace missing observations.

1.5 Justification for the Study

Most researchers normally attempt to make inferences about a population by drawing a random sample and studying relationships among the measurements contained in the sample. Any mismatch between the average characteristics of respondents in a sample and the average characteristics of the population can lead to serious inferential problems which may misdirect policy actions. To curb this, much attention need to be given to the problem of non-response bias both at stages of data collection and data analysis. Non-responses inevitably lead to samples having unequal number of entries per unit and thereby lead to biased estimates of the characteristics of interest under study. Thus, this prompts the empirical researchers to, at all times, consider the possibility of non-responses being present.

Moreover, in collecting real life data, missing observations as a result of intermittent nonresponses are always encountered. For instance, in engineering, stock market data, economic data, experimental data etc., not all information are captured in numerical form. These missing observations in effect, makes the analysis of real life data difficult thus leading to inaccurate findings which eventually results into incorrect inferences. Due to this, there is need for more studies to be carried out on the best approaches to be applied when computing missing values specifically for panel data models. Such studies will determine reliable techniques which will estimate missing observations accurately. Furthermore, new insights will be provided regarding the most appropriate modelling techniques for different missing data patterns.

1.6 Organization of the Thesis

Following the background introduction above, the rest of the thesis is organized as follows. Chapter 2 focuses on literature review and covering developments on nonlinear binary choice panel data models. Chapter 3 reviews the procedures of estimating logistic panel data model by conditional maximum likelihood approach. In addition, the concepts of unbalancedness in panel data sets and the various techniques of dealing with missing observations that lead to the unbalancedness are also discussed in this chapter. Chapter 4 incorporates the concept of missingness to conditional maximum likelihood estimation of nonlinear binary choice logistic model. Monte Carlo simulation results are also given and discussed in this chapter to assess the impact of missingness on the bias of the parameter estimates. Chapter 5 which is the last gives the summary, conclusions from the study and consequently provides recommendations for further study, based on the key findings from this work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter focuses on reviewing the work done by previous researchers (both empirical and theoretical) that are relevant to the problem of study. The main purpose of this chapter is to offer an overall view on the approaches developed so far in the estimation of panel data models (linear and nonlinear) and a detailed presentation of basic binary response logistic regression. This enables us to gain an insight of our research while avoiding repetition of the work and mistakes already done by others.

2.2 The Review of Literature

Panel data econometrics has greatly developed since the handbook chapter by Chamberlain (1984). Panel data methods so far studied are necessary for understanding individual specific behaviors. The analysis of two way models, both fixed and random effects, has been well worked out in the linear case in studies by Baltagi (1995, 2001). Greene (2000, chapter 14) shows that individual specific dummy variable coefficients can be estimated using group specific averages of residuals. By least squares dummy variables (LSDV) approach, the slope parameters in linear models can also be estimated using simple first differences.

Although for linear cases, regression using mean deviations sweeps out the fixed effects, there are a few analogous cases of nonlinear models, where the fixed effects can be eliminated, that have been identified in literature. Among them are the binomial logit model (Greene, 2000), Poisson and negative binomial regressions (Hausman, Hall and Griliches 1984) and exponential regression model (Munkin and Trivedi, 2000 and Greene, 2001). Differently put, when studying static linear models, fixed effects do not generally cause any problem, since they can easily be

differenced out to allow consistent estimation of the relevant parameters. The incidental parameter problem identified by Neyman and Scott (1948), motivated a rich literature on the estimation of single fixed effects nonlinear panel data models. Rasch (1960, 1961) considered the first model in the literature - the logit model. Later, Manski (1987) generalized this to develop a conditional maximum score estimator for binary response models that remains consistent under weak assumptions on the distribution of the errors. On the same breath, Hausman, Hall and Griliches (1984) used the relationship between the Poisson and multinomial distribution to solve the incidental parameter problem in the Poisson regression model (and Negative Binomial) in the presence of a single fixed effect. Like in the logistic case, this results in a conditional likelihood approach that can be used to obtain consistent estimates of the parameters of interest. Charbonneau (2012) extended the works of Hausman, Hall and Griliches (1984) by considering the adaptability of nonlinear panel data models to multiple fixed effects. From Monte Carlo simulations by Charbonneau (2012), the conditional ML parameter estimates proved less biased than other estimates for the logistic model.

With a more general approach to the problem, Hahn and Newey (2004) show that when N and T grow at the same rate, the fixed effects estimator is asymptotically biased and the asymptotic confidence intervals are wrong. They suggest two bias correction methods (the panel Jackknife and the analytic bias correction).

Most of the models so far studied are however considered mainly for cases with balanced panels in which no missing data due to nonresponses exist. The problem of non-response is normally ignorable for a regression model of interest if inference can be made about the model without caring about the process that causes the missing data. Certain conditions that allow one to neglect the selection process are given by Rubin (1976) and Little and Rubin (1987) for cross sectional case. Specifically, Little and Rubin (1987) introduced the concepts of missing at

random (MAR), missing not at random (MNAR) and missing completely at random (MCAR). Tsikriktis (2005) gave detailed overviews on various techniques of dealing with missing data which he categorized into three: deletion procedures, replacement procedures and model based procedures.

Griliches and Hausman (1986) noted that a frequent drawback of using panel data is the insignificant results produced by the 'within' approach to their analysis, which are often blamed on the errors of measurement magnified by this approach. They provide a variety of errors-in-variables models for panel data, but for a continuous dependent variable. The problem changes when the dependent variable is discrete.

Stefanski and Carroll (1985) studied errors in variables in the logistic regression model and suggested a bias-adjusted estimator. Kao and Schnell (1987) extended the results of Stefanski and Carroll to panel data and showed that, with errors in variables, the conditional maximum-likelihood estimator for a binary regression panel data model is asymptotically biased. They also introduced a bias-corrected estimator, which was examined asymptotically when the measurement error is small but non-negligible.

Individuals present in the data base may not be observed during the same period (unbalanced panels) or there may be 'holes' in the observation panel leading to incomplete panels. In literature, there exist two possibilities of estimating an econometric model with these kinds of incomplete panels. We can either use appropriate (unbalanced) estimation methods (Biorn, 1981 and Baltagi, 1985), which are in general quite complex or drop from the panel those individuals for which the observations are not complete and carry out the estimation on a balanced and complete sub-panel of the original one.

Verbeek and Nijman (1990) show that if we have unbalanced or incomplete panels we can use the usual estimators of panel data models based on a balanced and complete sub-panel. These

estimators are (asymptotically) unbiased and consistent under quite general conditions for the case when the observations are missing at random (except for the OLS).

2.3 Research Gap

From the available literature, it is evident that not much study on binary choice panel data and the logistic model has explored the concepts of nonresponse bias. As much as imputation techniques exist that can make datasets complete for ease of parameter estimation, the magnitudes of the biases introduced into the parameter estimates are not substantively quantified. This means that, hitherto no concrete procedure exists to suggest an appropriate imputation technique in the estimation of panel data models. A study in this area will therefore add on to the existing theoretical knowledge.

2.4 Conclusion

The available literature iterates that conditional maximum likelihood estimates are consistent even for the logistic model although with smaller bias compared to the unconditional MLE. Imputation also biases the covariates' averages. A study that combines these two biases, due to logistic regression and imputation, is worthwhile.

CHAPTER 3

BINARY CHOICE PANEL DATA MODELS

3.1 Introduction

This chapter introduces the basic concepts of binary choice models and the developments leading to the logistic model. This is done in the context of panel data sets for which each of the n individuals is observed for T time periods. The estimation technique of conditional ML is also explored for balanced panels. In addition, this chapter also highlights some of the causes of unbalancedness in panel data sets and the various common imputation techniques often employed in dealing with nonresponse or missing situations.

3.1.1 Binary Choice Variable

In many economic studies, the dependent variable is categorical indicating a success or a failure of an event. Such dependent variable is normally represented by a binary choice variable $y_{it} = 1$ if the event happens and 0 if it does not happen for individual i at time t. In fact if p_{it} is the probability of success for individual i at time t, then $E(y_{it}) = 1 \times p_{it} + 0 \times (1 - p_{it}) = p_{it}$ and this is usually modelled as a function of some explanatory variables. That is,

$$p_{it} = \Pr(y_{it} = 1) = E(y_{it}|x_{it}, c_i) = F(\boldsymbol{x_{it}\beta} + c_i)$$

3.2 The Linear Probability Model

3.2.1 The Model

Consider the linear regression model

$$y_{it} = \beta_1 + \beta_2 x_{2it} + \cdots + \beta_K x_{Kit} + c_i + u_{it}$$

$$y_{it} = \boldsymbol{x_{it}\beta} + c_i + u_{it} \qquad (3.1)$$

where y_{it} is a binary response variable, $\boldsymbol{x_{it}}$is a 1xK vector of observed explanatory variables (including a constant), $\boldsymbol{\beta}$ is a K x1 vector of parameters, c_i is an unobserved time invariant individual effect, and u_{it} is a zero-mean residual uncorrelated with all the terms on the right-hand side. Here, we assume strict exogeneity holds i.e. the residual u_{it} is uncorrelated with all x-variables over the entire time period spanned by the panel.

Since the dependent variable is binary, it is natural to interpret the expected value of y as a probability. Indeed, under random sampling, the unconditional probability that y equals one is equal to the unconditional expected value of y, i.e. E (y) = Pr (y = 1). As such,

$$Pr(y_{it} = 1|\boldsymbol{x_{it}}, c_i) = E(y_{it} = 1|\boldsymbol{x_{it}}, c_i; \beta)$$

So if the model (3.1) above is correctly specified, we have

$$\left.\begin{aligned} Pr(y_{it} = 1|\boldsymbol{x_{it}}, c_i) &= \boldsymbol{x_{it}\beta} + c_i \\ Pr(y_{it} = 0|\boldsymbol{x_{it}}, c_i) &= 1 - (\boldsymbol{x_{it}\beta} + c_i) \end{aligned}\right\} \qquad (3.2)$$

Equation (3.2) is a binary response model. In this particular model the probability of success (i.e. y = 1) is a linear function of the explanatory variables in the vector x. Hence this is called a linear probability model (LPM) for which OLS or the within estimation techniques can be used to obtain the parameter estimates. This LPM however has limitations when used to estimate the parameters for a discrete choice variable.

3.2.2 Weaknesses of the Linear Probability Model

One undesirable property of the LPM is that we can get predicted "probabilities" either less than zero or greater than one. Of course a probability by definition falls within the [0, 1] interval, so predictions outside this range are meaningless and somewhat embarrassing.

A related problem is that, conceptually, it does not make sense to say that a probability is linearly related to a continuous independent variable for all possible values. If it were, then

continually increasing this explanatory variable would eventually drive $P(y = 1|x)$ above one or below zero.

A third problem with the LPM, is that the residual is heteroskedastic. A possible way of solving this problem is to obtain estimates of the standard errors that are robust to heteroskedasticity.

A fourth problem is that the residual is not normally distributed. This implies that inference in small samples cannot be based on the usual suite of normality-based distributions such as the t test.

3.3 Logistic Model for Binary Response

3.3.1 The Model

To address the problems of LPM, a nonlinear binary response model is used where we write our nonlinear binary response model as

$$Pr(y_{it} = 1|\boldsymbol{x_{it}}, c_i) = G(\beta_1 + \beta_2 x_{2it} + \cdots + \beta_K x_{Kit} + c_i)$$

$$Pr(y_{it} = 1|\boldsymbol{x_{it}}, c_i) = G(\boldsymbol{x_{it}\beta} + c_i) \quad (3.3)$$

where G is a function taking on values strictly between zero and one: i.e.0 < G(z) < 1, for all real numbers z. The fact that 0 <G($\boldsymbol{x_{it}\beta} + c_i$) < 1 ensures that the estimated response probabilities are strictly between zero and one, which thus addresses the main limitation of using LPM. G is a cumulative density function (cdf), monotonically increasing in the index z (i.e. $z = \boldsymbol{x_{it}\beta} + c_i$), with

$$\left.\begin{aligned} Pr(y_{it} = 1|\boldsymbol{x_{it}}, c_i) \to 1 \; as \;\; \boldsymbol{x_{it}\beta} + c_i \to \infty \\ Pr(y_{it} = 1|\boldsymbol{x_{it}}, c_i) \to 0 \; as \;\; \boldsymbol{x_{it}\beta} + c_i \to -\infty \end{aligned}\right\} \quad (3.4)$$

Thus G is a nonlinear function, and hence we cannot use a linear regression model for estimation. Various non-linear functions for G have been suggested in the literature and the most common

ones are the logistic distribution, yielding the logit model, and the standard normal distribution, yielding the probit model. In the logistic model, G takes the form,

$$G(\boldsymbol{x_{it}\beta} + c_i) = \frac{e^{\boldsymbol{x_{it}\beta}+c_i}}{1 + e^{\boldsymbol{x_{it}\beta}+c_i}} \tag{3.5}$$

which is between zero and one for all values of $\boldsymbol{x_{it}\beta}$**.** This is the cumulative distribution function (CDF) for a logistic variable.

3.3.2 Assumptions of the Logistic Model

In Logistic estimation, there does not exist many of the key assumptions of linear regression and general linear models that are based on ordinary least squares algorithms – particularly regarding linearity, normality, homoscedasticity, and measurement level. As such, logistic regression has the following unique characteristics to be mentioned:

i. it does not need a linear relationship between the dependent and independent variables. Logistic regression can handle all sorts of relationships, because it applies a non-linear log transformation to the predicted odds ratio,

ii. the independent variables do not need to be multivariate normal – although multivariate normality yields a more stable solution. Also the error terms (the residuals) do not need to be multivariate normally distributed,

iii. homoscedasticity is not needed,

iv. it can handle ordinal and nominal data as independent variables. The independent variables do not need to be metric (interval or ratio scaled).

The following assumptions, however, still apply:

A1: Binary logistic regression requires the dependent variable to be binary and ordinal logistic regression requires the dependent variable to be ordinal.

A2: For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome. Since logistic regression assumes that P(y=1) is the probability of the event occurring, it is necessary that the dependent variable is coded accordingly.

A3: The model should be fitted correctly. Neither over fitting nor under fitting should occur. That is only the meaningful variables should be included, but also all meaningful variables should be included. A good approach to ensure this is to use a stepwise method to estimate the logistic regression.

A4: The error terms need to be independent. Logistic regression requires each observation to be independent. That is that the data-points should not be from any dependent samples design, e.g., before-after measurements, or matched pairings.

A5: The model should have little or no multicollinearity. That is that the independent variables should be independent from each other. However, there is the option to include interaction effects of categorical variables in the analysis and the model. If multicollinearity is present centering the variables might resolve the issue, i.e. deducting the mean of each variable, before the logistic regression is estimated.

A6: Logistic regression assumes linearity of independent variables and log odds. Whilst it does not require the dependent and independent variables to be related linearly, it requires that the independent variables are linearly related to the log odds.

A7: Large sample sizes should be available; the sample size should be at least ten times the number of parameters to be estimated.

3.4 Estimation of Logistic Model

3.4.1 Incidental Parameter Problem

For Panel data, the presence of individual effects complicates the parameter estimation significantly.

Consider the fixed effects panel data model, $y_{it} = \boldsymbol{x_{it}\beta} + c_i + u_{it}$ with $\Pr(y_{it} = 1) = F(\boldsymbol{x_{it}\beta} + c_i)$.

In this case c_i and $\boldsymbol{\beta}$ are unknown parameters to be estimated and as $N \to \infty$ for fixed T, the number of parameters c_i increases with N . As such c_i cannot be consistently estimated for fixed T. This is known as the incidental parameter problem in statistics, first discussed by Neyman and Scott (1948) and later reviewed by Lancaster (2000).

For linear panel data regression model, when T is fixed, only β can be estimated consistently by first getting rid of c_i using the within transformation. This is possible for the linear case because the MLE of β and c_i are asymptotically independent (Hsiao 2003). For qualitative binary choice model with fixed T, this is not possible as demonstrated by Chambelain (1980).

Hsiao (2003) simply illustrates how the inconsistency of the ML estimate of $c_i$is transmitted into inconsistency for $\widehat{\boldsymbol{\beta}}_{mle}$.This is done in the context of a logistic model with one regressor x_{it} that is observed over two periods, with x_{i1} = 0 and x_{i2} = 1where as $N \to \infty$ with $T = 2$, $plim\hat{\beta}_{mle} = 2\beta$. Greene (2004a) shows that despite the large number of incidental parameters, one can still force maximum likelihood estimation for the fixed effects model by including a large number of dummy variables. Using Monte Carlo experiments, he showed that the fixed effects MLE is biased even when $T$is large. For N = 1000, T = 2 and 200 replications, this bias is 100%, confirming the results derived by Hsiao (2003). However, this bias improves as T increases. For

example, when N = 1000 and T = 10 this bias is 16% and when N = 1000 and T = 20 this bias is 6.9%.

3.4.2 The Unconditional Likelihood Function

The logistic model is estimated by means of Maximum Likelihood (ML). That is, the ML estimate of β is the particular vector $\hat{\beta}^{ML}$ that gives the greatest likelihood of observing the outcomes in the sample $\{y_1, y_2, ...\}$ conditional on the explanatory variables x.

By assumption, the probability of observing y_{it}= 1 is $G(\boldsymbol{x_{it}\beta} + c_i)$ while the probability of observing y_{it}= 0 is $1 - G(\boldsymbol{x_{it}\beta} + c_i)$. It then follows that the probability of observing the entire sample is

$$L(y|\boldsymbol{x};\beta) = \prod_{i\in l} G(\boldsymbol{x_{it}\beta} + c_i) \prod_{i\in m} [1 - G(\boldsymbol{x_{it}\beta} + c_i)] \tag{3.6}$$

where *l* refers to the observations for which y = 1 and *m* to the observations for which y = 0.

We can rewrite this as

$$L(y|\boldsymbol{x};\beta) = \prod_{i=1}^{N} (G(\boldsymbol{x_{it}\beta} + c_i))^{y_i} [1 - G(\boldsymbol{x_{it}\beta} + c_i)]^{1-y_i} \tag{3.7}$$

because when y = 1 we get $G(\boldsymbol{x_{it}\beta} + c_i)$ and when y = 0 we get $[1 - G(\boldsymbol{x_{it}\beta} + c_i)]$.

The log likelihood function for the sample is

$$lnL(y|\boldsymbol{x};\beta) = \sum_{i=1}^{N} \{y_i lnG(\boldsymbol{x_{it}\beta} + c_i) + (1 - y_i) ln[1 - G(\boldsymbol{x_{it}\beta} + c_i)]\} \tag{3.8}$$

The MLE of β maximizes this log likelihood function.

3.4.3 Conditional Likelihood Function for Logistic Panel Data Model

If G is the logistic CDF then we obtain the logistic log likelihood:

$$lnL(y|\boldsymbol{x};\beta) = \sum_{i=1}^{N}\left\{y_i ln\left(\frac{e^{\boldsymbol{x_{it}\beta}+c_i}}{1+e^{\boldsymbol{x_{it}\beta}+c_i}}\right) + (1-y_i) ln\left(\frac{1}{1+e^{\boldsymbol{x_{it}\beta}+c_i}}\right)\right\} \quad (3.9)$$

Estimating the parameters in this model is not easy as it is specified since the unobserved individual characteristics, c_i are also not known. In linear models, it is easy to eliminate c_i by means of first differencing or using within transformation. If we attempt to estimate c_i directly by adding N-1 individual dummy variables to the logistic specification, this will result in severely biased and inconsistent estimates of β unless T is large due to the incidental parameters problem.

One important advantage of the logistic model over the probit model is that it is possible to obtain a consistent estimator of β without making any assumptions about how c_i is related to $\boldsymbol{x_{it}}$ (however, strict exogeneity must hold).

This is possible, because the logistic functional form enables us to eliminate c_i from the estimating equation, once we condition on the "minimal sufficient statistic" for c_i. As such we obtain the conditional likelihood function whose parameters are estimated.

To see this, assume T = 2, and consider the following conditional probabilities:

$$Pr(y_{i1} = 0, y_{i2} = 1|x_{i1}, x_{i2}, c_i, y_{i1} + y_{i2} = 1) \quad (3.10)$$

and

$$Pr(y_{i1} = 1, y_{i2} = 0|x_{i1}, x_{i2}, c_i, y_{i1} + y_{i2} = 1) \quad (3.11)$$

It is worth noting here that conditioning is on $y_{i1} + y_{i2} = 1$, i.e. that y_{it} changes between the two time periods. For the logistic functional form, we have

$$Pr(y_{i1} + y_{i2} = 1|x_{i1}, x_{i2}, c_i,) = \frac{e^{\boldsymbol{x_{i1}\beta}+c_i}}{1+e^{\boldsymbol{x_{i1}\beta}+c_i}}\frac{1}{1+e^{\boldsymbol{x_{i2}\beta}+c_i}} + \frac{1}{1+e^{\boldsymbol{x_{i1}\beta}+c_i}}\frac{e^{\boldsymbol{x_{i2}\beta}+c_i}}{1+e^{\boldsymbol{x_{i2}\beta}+c_i}}$$

or simply

$$Pr(y_{i1} + y_{i2} = 1|x_{i1}, x_{i2}, c_i,) = \frac{e^{\boldsymbol{x_{i1}\beta}+c_i} + e^{\boldsymbol{x_{i2}\beta}+c_i}}{[1+e^{\boldsymbol{x_{i1}\beta}+c_i}][1+e^{\boldsymbol{x_{i2}\beta}+c_i}]} \quad (3.12)$$

Similarly,

$$Pr(y_{i1}=0, y_{i2}=1|x_{i1}, x_{i2}, c_i) = \frac{1}{1+e^{\boldsymbol{x_{i1}\beta}+c_i}} \frac{e^{\boldsymbol{x_{i2}\beta}+c_i}}{1+e^{\boldsymbol{x_{i2}\beta}+c_i}} \tag{3.13}$$

hence, conditional probability on $y_{i1}+y_{i2}=1$,

$$Pr(y_{i1}=0, y_{i2}=1|x_{i1}, x_{i2}, c_i, y_{i1}+y_{i2}=1) = \frac{e^{\boldsymbol{x_{i2}\beta}+c_i}}{e^{\boldsymbol{x_{i1}\beta}+c_i}+e^{\boldsymbol{x_{i2}\beta}+c_i}}$$

$$Pr(y_{i1}=0, y_{i2}=1|x_{i1}, x_{i2}, y_{i1}+y_{i2}=1) = \frac{e^{(\boldsymbol{x_{i2}-x_{i1}})\boldsymbol{\beta}}}{1+e^{(\boldsymbol{x_{i2}-x_{i1}})\boldsymbol{\beta}}} \tag{3.14}$$

The key result here is that the $c_i's$ are eliminated. It also follows that

$$Pr(y_{i1}=1, y_{i2}=0|x_{i1}, x_{i2}, y_{i1}+y_{i2}=1) = \frac{1}{1+e^{(\boldsymbol{x_{i2}-x_{i1}})\boldsymbol{\beta}}} \tag{3.15}$$

Probabilities (3.14) and (3.15) are conditional on $y_{i1}+y_{i2}=1$ and are independent of c_i.

The probability distribution function is thus given as

$$Pr(y_{i1}, y_{i2}|x_{i1}, x_{i2}, y_{i1}+y_{i2}=1) = \begin{cases} 1 & if\ (y_{i1}, y_{i2}) = (0,0) or (1,1) \\ \frac{1}{1+e^{(\boldsymbol{x_{i2}-x_{i1}})\boldsymbol{\beta}}} & if\ (y_{i1}, y_{i2}) = (1,0) \\ \frac{e^{(\boldsymbol{x_{i2}-x_{i1}})\boldsymbol{\beta}}}{1+e^{(\boldsymbol{x_{i2}-x_{i1}})\boldsymbol{\beta}}} & if\ (y_{i1}, y_{i2}) = (0,1) \end{cases} \tag{3.16}$$

The conditional log likelihood function is therefore given as

$$lnL = \sum_{i=1}^{N} \left\{ d_{01i} ln\left(\frac{e^{(\boldsymbol{x_{i2}-x_{i1}})\boldsymbol{\beta}}}{1+e^{(\boldsymbol{x_{i2}-x_{i1}})\boldsymbol{\beta}}}\right) + d_{10i} ln\left(\frac{1}{1+e^{(\boldsymbol{x_{i2}-x_{i1}})\boldsymbol{\beta}}}\right) \right\} \tag{3.17}$$

Where d_{01i} selects the individuals for which the dependent variable changed from 0 to 1 while d_{10i} selects the cases for which the dependent variable changed from 1 to 0.

Hence, by maximizing the conditional log likelihood function (3.17) we obtain consistent estimates of β, regardless of whether c_i and x_{it} are correlated.

The trick is thus to condition the likelihood on the outcome series (y_{i1}, y_{i2}), and in the more general case. For example, if T = 3, we can condition on $\sum_t y_{it} = 1$, with possible sequences

(1,0,0) , (0,1,0) , (0,0,1) , or on $\sum_t y_{it} = 2$ with possible sequences (1,1,0) , (0,1,1) , (1,0,1). The general conditional probability of the response variable $(y_{i1}, y_{i2}, \ldots, y_{iT})$ given $\sum_t y_{it}$ is

$$Pr\left(y_{i1}, y_{i2}, \ldots, y_{iT} \middle| \boldsymbol{X_i}, \sum_t y_{it}\right) = \frac{e^{(\sum_t y_{it} \boldsymbol{x_{it}} \boldsymbol{\beta})}}{\sum_{d \in B_i} e^{(\sum_t d_{it} \boldsymbol{x_{it}} \boldsymbol{\beta})}} \quad (3.18)$$

where $B_i = \{(d_{i1}, d_{i2}, \ldots, d_{iT}) | d_{it} = 0,1 \; and \sum_t d_{it} = \sum_t y_{it}\}$

3.5 Newton-Raphson Algorithm

To obtain the values of β that maximize 3.17, we use the Newton-Raphson algorithm. Starting from an initial estimate, $\boldsymbol{\beta}^{(0)}$, the algorithm consists of iterating the estimate at step h as

$$\boldsymbol{\beta}^{(h)} = \boldsymbol{\beta}^{(h-1)} + \boldsymbol{J}\left(\boldsymbol{\beta}^{(h-1)}\right)^{-1} \boldsymbol{s}\left(\boldsymbol{\beta}^{(h-1)}\right) \quad (3.19)$$

Where, $\boldsymbol{s}(\boldsymbol{\beta}) = \frac{\partial \mathbf{lnL}}{\partial \boldsymbol{\beta}}$ is the score vector and $\boldsymbol{J}(\boldsymbol{\beta}) = -\frac{\partial^2 \mathbf{lnL}}{\partial \boldsymbol{\beta} \, \partial \boldsymbol{\beta}'}$ is the observed information matrix

3.6 Unbalanced Panels

In a panel data set, there are observations across cross-sectional units (e.g. individuals or firms), and across time periods. Often such a data-set can be represented by a completely filled in matrix of N units and T periods. In the unbalanced data case, however, the number of observations per time period varies. Equivalently we might say that the number of observations per unit is not always the same. We can handle this by letting T be the total number of time periods and N_t be the number of observations in each period for t=1, 2,…,T.

3.6.1 Causes of Unbalancedness

Unbalancedness in panel data is as a result of three major factors, namely;

a. Delayed Entry

Here, an individual in panel study joins into the panel after several time periods have passed during which other individuals were being studied.

b. Early Exit

Some individuals may opt out of a study before the intended time for the study elapses. This may be considered as a case of attrition.

c. Intermittent Non-response from a Study Unit

Despite certain surveys having equal number of units and time periods throughout the study, some study units may not provide all the information required. This is commonly referred to as item nonresponse and as such the entire data set is rendered unbalanced.

For the former two causes of unbalancedness, each individual is observed T_i times (i=1, 2,…,n) and analysis of the panel data models is still feasible. However, in cases of intermittent non-responses a need to establish the nature and cause of the non-response suffices and the approaches suggested in literature on how to handle missing observations become valid in such cases.

3.6.2 Nonresponses in Panel Data

Intermittent non-responses lead to missing observations within the data set and ,therefore, to decide how to handle missing data, one must establish why they are missing. Four missing data mechanisms have been discussed in literature. Considering the four general mechanisms moving from the simplest to the most general, we have;

a. Missingness Completely at Random

A variable is missing completely at random if the probability of missingness is the same for all units, for example, if each survey respondent decides whether to answer a question by rolling a die and refusing to answer if a particular face shows up. If data are missing completely at random, then throwing out cases with missing data does not bias inferences.

b. Missingness at Random

Most missingness is not completely at random, as can be seen from the data themselves. A more general assumption, missing at random, is that the probability that a variable is missing depends only on available information. When an outcome variable is missing at random, it is acceptable to exclude the missing cases, as long as the regression controls for all the variables that affect the probability of missingness to avoid nonresponse bias.

This missing-at-random assumption (a more formal version of which is sometimes called the ignorability assumption) in the missing-data framework is basically same sort of assumption as ignorability in the causal framework. Both require that sufficient information has been collected that we can "ignore" the assignment mechanism (assignment to treatment, assignment to nonresponse).

c. Missingness that Depends on Unobserved Predictors

Missingness is no longer "at random" if it depends on information that has not been recorded and this information also predicts the missing values. A familiar example from medical studies is that if a particular treatment causes discomfort, a patient is more likely to drop out of the study. This missingness is not at random (unless "discomfort" is measured and observed for all patients). If missingness is not at random, it must be explicitly modelled, or else biased inferences will be inevitable.

d. Missingness that Depends on the Missing Value Itself

Finally, a particularly difficult situation arises when the probability of missingness depends on the (potentially missing) variable itself. For example, suppose that people with higher earnings are less likely to reveal them. In the extreme case (for example, all persons earning more than ksh.100,000 refuse to respond), this is called censoring, but even the probabilistic case causes difficulty.

3.7 Techniques for Dealing with Missing Data

There are three ways to treat missing data according to Kline (1998): (a) to delete them, (b) to replace (impute) the missing data with estimated scores and (c) to model the distribution of missing data and estimate them based on certain parameters. Each one of these families of techniques is discussed below.

3.7.1 Deletion Procedures

We have two deletion procedures;

a. Listwise Deletion

This method eliminates from further analysis all cases with any missing data. As a result, it sacrifices a large amount of data (Malhotra, 1987). In the regression context, this usually means complete-case analysis: excluding all units for which the outcome or any of the inputs are missing.

Two problems arise with complete-case analysis:

i. If the units with missing values differ systematically from the completely observed cases, this could bias the estimates obtained from complete-case analysis.

ii. If many variables are included in a model, there may be very few complete cases, so that most of the data would be discarded for the sake of a simple analysis.

Despite the fact that the large loss of data reduces statistical power and accuracy (Little and Rubin, 1987), listwise deletion is the default option for analysis in most statistical software packages. Listwise deletion results in conservative results, since by reducing the sample size, it also results in a decrease in statistical power. Hence, it tends to make fewer variables statistically significant.

b. Pairwise Deletion

Pairwise deletion deletes cases only from those statistical analyses that require the information. For example, if a respondent is missing information on variable A, the respondent's data could still be used to calculate other correlations, such as the one between variables B and C.

Compared to listwise deletion, pairwise deletion preserves much more information that would have been lost if the researcher was using listwise deletion (Roth, 1994).

3.7.2 Replacement Procedures

Before discussing replacement procedures in depth, it is important to note that empirical researchers should be careful before they start replacing data. Data replacement does not compensate for a badly designed instrument or for poor data collection. Overall, replacement procedures can be used in certain cases, as long as the researcher has a good reason for replacing. In general, the replacement procedures are easy to perform, and some are included as options in statistical packages. The most important advantages of these procedures are the retention of the sample size and, consequently, of statistical power in subsequent analyses. To a greater or lesser extent, all replacement procedures are biased if there is a non-random distribution of missing values.

Many different missing data replacement procedures have been developed over the years. In general, it has been found that the differences between the various methods decrease with: (a) larger sample size, (b) a smaller percentage of missing values, (c) fewer missing variables and (d) a decrease in the level of the correlations between the variables (Raymond, 1986).

However, Kromrey and Heines (1994) reported that this is not the case if the effects of the treatments on the analytical statistics are taken into account. With larger sample sizes, in fact, the differences between the various replacement procedures are found to increase; this provides further evidence that in assessing the effectiveness of missing data treatments, both the accuracy

of estimating the value of missing data and the accuracy of estimating the statistical effects have to be considered.

Several types of replacement procedures can be distinguished: mean-based, regression-based and hot-deck imputation, last value carried forward among others.

a. Mean Substitution

There are three variants of mean substitution: total mean substitution, subgroup mean substitution and case mean substitution.

Under total mean substitution, the missing value of a variable is replaced by the mean on the item for all respondents answering the question. According to the subgroup mean substitution, the missing value is replaced by the mean of the subgroup of which the respondent is a member. The third variant of mean substitution is the case mean substitution, which replaces missing values with the intra-individual mean of the respondent for all non-missing items. Mean imputation, unfortunately, can severely distort the distribution for this variable, leading to complications with summary measures including, notably, underestimates of the standard deviation. Moreover, mean imputation distorts relationships between variables by "pulling" estimates of the correlation toward zero. The median of the available values can also be used to substitute for the missing values just in a similar way to mean substitution.

b. Regression Imputation

This is a two-step approach: first, the researcher estimates the relationships among variables, and then uses the regression coefficients to estimate the missing value (Frane, 1976). The underlying assumption of regression imputation is the existence of a linear relationship between

the predictors and the missing variable. The technique also assumes that values are missing at random (i.e. a missing value is not related to the value of the predictors).

c. Hot-deck Imputation

According to this technique, the researcher should replace a missing value with the actual score from a similar case in the dataset.

d. Last Value Carried Forward

In evaluations of interventions where pre-treatment measures of the outcome variable are also recorded, a strategy that is sometimes used is to replace missing outcome values with the pre-treatment measure. This is often thought to be a conservative approach (that is, one that would lead to underestimates of the true treatment effect).

e. Using Information from Related Observations

Suppose we are missing data regarding the income of fathers of children in a dataset. Why not fill these values in with mother's report of the values? This is a plausible strategy, although these imputations may propagate measurement error. Also we must consider whether there is any incentive for the reporting person to misrepresent the measurement for the person about whom he or she is providing information.

f. Indicator Variables for Missingness of Categorical Predictors

For unordered categorical predictors, a simple and often useful approach to imputation is to add an extra category for the variable indicating missingness.

3.7.3 Model-based Procedures

Finally, we now explain three main approaches to model based procedures.

a. Nonresponse Weighting

Suppose, for instance, that only one variable has missing data. We could build a model to predict the nonresponse in that variable using all the other variables. The inverse of predicted

probabilities of response from this model could then be used as survey weights to make the complete-case sample representative of the full sample. This method becomes more complicated when there is more than one variable with missing data. Moreover, as with any weighting scheme, there is the potential that standard errors will become erratic if predicted probabilities are close to 0 or 1.

b. Maximum Likelihood Estimation with Missing Data

The maximum likelihood approach to analyzing missing data has many different forms. In its simplest form, it assumes that the observed data are a sample drawn from a multivariate normal distribution (DeSarbo et al., 1986). The parameters are estimated by available data, and then missing scores are estimated based on the parameters just estimated. Contrary to the techniques discussed above, maximum likelihood procedures allow explicit modelling of missing data that is open to scientific analysis and critique.

c. Expectation Maximization

The expectation maximization algorithm is an iterative process (Laird, 1988; Ruud, 1991). The first iteration estimates missing data and then parameters using maximum likelihood. The second iteration re-estimates the missing data based on the new parameter estimates and then recalculates the new parameters estimates based on actual and re-estimated missing data (Little and Rubin, 1987). The approach continues until there is convergence in the parameter estimates.

CHAPTER 4

METHODOLOGY AND DATA ANALYSIS

4.1 Introduction

Having explored the general theory of the above binary choice model and the basics on missing observations leading to unbalanced panels under study in the preceding chapter, this chapter is dedicated to estimating a logistic model by conditional maximum likelihood approach in the presence of missing observations within the covariates. Using Monte Carlo simulations, the biases in the parameter estimates shall be compared and consequently imputation procedures with reduced biases are suggested.

A description of the simulated data is given in section 4.3 where the general statistical features of panel data are investigated.

4.2 Parameter Estimation

Consider the logistic panel data model given by

$$P(\boldsymbol{y_{it}} = \mathbf{1}|\boldsymbol{x_{it}}, \boldsymbol{\beta}, c_i) = \frac{e^{\boldsymbol{x_{it}\beta}+c_i}}{1 + e^{\boldsymbol{x_{it}\beta}+c_i}} \tag{4.1}$$

where $\boldsymbol{x_{it}}$ is the vector of covariates. In the presence of missing observations in the vector $\boldsymbol{x_{it}}$, we express it as a sum of two vectors $\boldsymbol{x_{it_s}}$ and $\boldsymbol{x_{it_I}}$ for the sample-present covariate values and the missing covariate values respectively. Therefore, the model (4.1) above is written as

$$P(\boldsymbol{y_{it}} = \mathbf{1}|\boldsymbol{x_{it}}, \boldsymbol{\beta}, c_i) = \frac{e^{[\boldsymbol{x_{it_s}}+\boldsymbol{x_{it_I}}]\boldsymbol{\beta}+c_i}}{1 + e^{[\boldsymbol{x_{it_s}}+\boldsymbol{x_{it_I}}]\boldsymbol{\beta}+c_i}} \tag{4.2 a}$$

and

$$P(\boldsymbol{y_{it}} = \mathbf{0}|\boldsymbol{x_{it}}, \boldsymbol{\beta}, c_i) = \frac{1}{1 + e^{[\boldsymbol{x_{it_s}}+\boldsymbol{x_{it_I}}]\boldsymbol{\beta}+c_i}} \tag{4.2 b}$$

For a panel data set with only two time periods we have $t = 1, 2$ and we consider the conditional probabilities:

$$Pr(y_{i1} = 0, y_{i2} = 1|\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, c_i, y_{i1} + y_{i2} = 1) \qquad (4.3\ a)$$

and

$$Pr(y_{i1} = 1, y_{i2} = 0|\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, c_i, y_{i1} + y_{i2} = 1) \qquad (4.3\ b)$$

The conditioning is done on $y_{i1} + y_{i2} = 1$ as this sum is a sufficient statistic for the individual effects, c_i . For the logistic functional form, we have $Pr(y_{i1} + y_{i2} = 1|\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, c_i,) = \frac{e^{[x_{i1_s}+x_{i1_I}]\boldsymbol{\beta}+c_i}}{1+e^{[x_{i1_s}+x_{i1_I}]\boldsymbol{\beta}+c_i}} \frac{1}{1+e^{[x_{i2_s}+x_{i2_I}]\boldsymbol{\beta}+c_i}} + \frac{1}{1+e^{[x_{i1_s}+x_{i1_I}]\boldsymbol{\beta}+c_i}} \frac{e^{[x_{i2_s}+x_{i2_I}]\boldsymbol{\beta}+c_i}}{1+e^{[x_{i2_s}+x_{i2_I}]\boldsymbol{\beta}+c_i}}$

which yield, upon simplification,

$$Pr(y_{i1} + y_{i2} = 1|\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, c_i,) = \frac{e^{[x_{i1_s}+x_{i1_I}]\boldsymbol{\beta}+c_i} + e^{[x_{i2_s}+x_{i2_I}]\boldsymbol{\beta}+c_i}}{\left[1 + e^{[x_{i1_s}+x_{i1_I}]\boldsymbol{\beta}+c_i}\right]\left[1 + e^{[x_{i2_s}+x_{i2_I}]\boldsymbol{\beta}+c_i}\right]} \qquad (4.4)$$

and

$$Pr(y_{i1} = 0, y_{i2} = 1|x_{i1}, x_{i2}, c_i) = \frac{1}{1 + e^{[x_{i1_s}+x_{i1_I}]\boldsymbol{\beta}+c_i}} \frac{e^{[x_{i2_s}+x_{i2_I}]\boldsymbol{\beta}+c_i}}{1 + e^{[x_{i2_s}+x_{i2_I}]\boldsymbol{\beta}+c_i}} \qquad (4.5)$$

The conditional probabilities on $y_{i1} + y_{i2} = 1$ are thus expressed as,

$$Pr(y_{i1} = 0, y_{i2} = 1|\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, c_i, y_{i1} + y_{i2} = 1) = \frac{e^{[x_{i2_s}+x_{i2_I}]\boldsymbol{\beta}+c_i}}{e^{[x_{i1_s}+x_{i1_I}]\boldsymbol{\beta}+c_i} + e^{[x_{i2_s}+x_{i2_I}]\boldsymbol{\beta}+c_i}}$$

$$Pr(y_{i1} = 0, y_{i2} = 1|\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, y_{i1} + y_{i2} = 1) = \frac{e^{\{(x_{i2_s}+x_{i2_I})-(x_{i1_s}+x_{i1_I})\}\boldsymbol{\beta}}}{1 + e^{\{(x_{i2_s}+x_{i2_I})-(x_{i1_s}+x_{i1_I})\}\boldsymbol{\beta}}} \qquad (4.6)$$

Notice that the individual fixed effects,c_i's, are eliminated through the conditioning on $y_{i1} + y_{i2} = 1$. It also follows that

$$Pr(y_{i1} = 1, y_{i2} = 0|\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, y_{i1} + y_{i2} = 1) = \frac{1}{1 + e^{\{(x_{i2_s}+x_{i2_I})-(x_{i1_s}+x_{i1_I})\}\boldsymbol{\beta}}} \qquad (4.7)$$

Equations (4.6) and (4.7) can be expressed as

$$Pr(y_{i1}=0, y_{i2}=1|\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, y_{i1}+y_{i2}=1) = \frac{e^{\Delta x_{i_I}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}} \tag{4.8}$$

and

$$Pr(y_{i1}=1, y_{i2}=0|\boldsymbol{x_{i1}}, \boldsymbol{x_{i2}}, y_{i1}+y_{i2}=1) = \frac{e^{-\Delta x_{i_s}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}} \tag{4.9}$$

respectively, where $\Delta \boldsymbol{x_{i_I}} = \left(\boldsymbol{x_{i2_I}} - \boldsymbol{x_{i1_I}}\right)$ and $\Delta \boldsymbol{x_{i_s}} = \left(\boldsymbol{x_{i2_s}} - \boldsymbol{x_{i1_s}}\right)$.

The conditional log likelihood function can thus be obtained using equations (4.8) and (4.9) as

$$lnL = \sum_{i=1}^{N}\left\{d_{01i} ln\left(\frac{e^{\Delta x_{i_I}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}}\right) + d_{10i} ln\left(\frac{e^{-\Delta x_{i_s}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}}\right)\right\} \tag{4.10}$$

where d_{01i}selects the individuals for which the dependent variable changed from 0 to 1 while d_{10i}selects the cases for which the dependent variable changed from 1 to 0.

The score vector and observed information matrix are respectively,

$$\boldsymbol{s(\beta)} = \frac{\partial \mathbf{lnL}}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{N}\left\{d_{01i}\left[\Delta \boldsymbol{x'}_{i_I} - \left(\frac{-\Delta \boldsymbol{x'}_{i_s} e^{-\Delta x_{i_s}\boldsymbol{\beta}} + \Delta \boldsymbol{x'}_{i_I} e^{\Delta x_{i_I}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}}\right)\right]\right.$$

$$\left. - d_{10i}\left[\Delta \boldsymbol{x'}_{i_s} + \left(\frac{-\Delta \boldsymbol{x'}_{i_s} e^{-\Delta x_{i_s}\boldsymbol{\beta}} + \Delta \boldsymbol{x'}_{i_I} e^{\Delta x_{i_I}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}}\right)\right]\right\} \tag{4.11}$$

$$\boldsymbol{J(\beta)} = -\frac{\partial^2 \mathbf{lnL}}{\partial \boldsymbol{\beta}\, \partial \boldsymbol{\beta'}} = \sum_{i=1}^{N}\left\{d_{01i}\left[\left(\frac{\Delta \boldsymbol{x'}_{i_s}{}^2 e^{-\Delta x_{i_s}\boldsymbol{\beta}} + \Delta \boldsymbol{x'}_{i_I}{}^2 e^{\Delta x_{i_I}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}}\right) - \left(\frac{-\Delta \boldsymbol{x'}_{i_s} e^{-\Delta x_{i_s}\boldsymbol{\beta}} + \Delta \boldsymbol{x'}_{i_I} e^{\Delta x_{i_I}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}}\right)^2\right]\right.$$

$$\left. + d_{10i}\left[\left(\frac{\Delta \boldsymbol{x_{i_s}}{}^2 e^{-\Delta x_{i_s}\boldsymbol{\beta}} + \Delta \boldsymbol{x'}_{i_I}{}^2 e^{\Delta x_{i_I}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}}\right) - \left(\frac{-\Delta \boldsymbol{x'}_{i_s} e^{-\Delta x_{i_s}\boldsymbol{\beta}} + \Delta \boldsymbol{x'}_{i_I} e^{\Delta x_{i_I}\boldsymbol{\beta}}}{e^{-\Delta x_{i_s}\boldsymbol{\beta}} + e^{\Delta x_{i_I}\boldsymbol{\beta}}}\right)^2\right]\right\}$$

By using the Newton-Raphson algorithm equation (3.19) consistent estimates of the parameter can be obtained.

4.3 Monte Carlo Simulation

In this section, we present the results of Monte Carlo simulation to investigate the properties of conditional ML estimator developed in section 4.2 above. For this, we focus on the conditional ML estimator for the logistic model given by the maximization of (4.10). The simulation results will compare the conditional ML estimator to the unconditional ML estimator which estimates all the fixed effects by putting in dummies. The unconditional ML estimator for the logistic model, however, is subject to the incidental parameter problem.

To account for different possible features of the data, this comparison will be made for two sets of data, one complete (balanced) and the other incomplete (unbalanced) due to intermittent nonresponses. The latter data set is balanced by imputing the missing observations and substituting the imputed vector $\boldsymbol{x_{it_I}}$into the conditional log likelihood function (4.10) where the imputation methods described in section 3.7 are employed. Both panel sets are applied to the estimation of the following model:

$y_{it} = 1(\boldsymbol{x_{it}\beta} + c_i + v_{it} \geq 0) \quad i = 1,2,\dots,n \quad t = 1,2,\dots,T$ where $\boldsymbol{x_{it}}$ is a vector of five explanatory variables drawn from uniform, binomial and normal distributions (see Table 4.1) and the error term v_{it} has a logistic distribution. All other parameters, beta1 to beta5, of the model necessary to calculate the dependent variable y were fixed as β_1=1, β_2=-1, β_3=1, β_4=1 and β_5=1. Having determined all the independent variables, the dependent variable, y, was calculated from the relation $y_{it} = 1\big(c_i + \beta_1 x^1{}_{it} + \beta_2 x^2{}_{it} + \beta_3 x^3{}_{it} + \beta_4 x^4{}_{it} + \beta_5 x^5{}_{it} x_5 + v_{it} \geq 0\big) \quad i = 1,2,\dots,n \quad t = 1,2,\dots,T$ where v_{it} follows a logistic distribution given by $v_{it} = ln\left|\frac{u_{it}}{1+u_{it}}\right|$ with u_{it} being a standard normal random variable. The fixed effects c_i are obtained as functions of x_1 and t by the relation $c_i = \frac{\sqrt{t}\sum x_1}{n} + a_i$ with a_i being a standard normal random variable as well.

Table 4.1: Description of variables

Variable	Type	
x_1	continuous	N~(0, 1)
x_2	continuous	U~ (0, 1)
x_3	continuous	N~ (0.5, 0.5)
x_4	discrete	B~(nT, 2, 0.65)
x_5	discrete	Binary

Three different sample sizes were used for both the balanced and unbalanced sets of data fitted to the model i.e. n = 50, 100 and 250. In addition, for each sample size; we vary the proportion of missingness from 10% to 30% by randomly deleting the desired proportion of observations from the data set and imputing them back through mean imputation, last value carried forward imputation and median imputation. Whenever fixed effects are estimated, the coefficients are truncated in order to ensure convergence. The summarized results for 1000 replications are given in Tables 4.2 to 4.7. For both estimators (unconditional MLE and conditional MLE) considered, we report the median bias, the median absolute deviation (MAD), the mean bias, and the root mean squared error (RMSE) for all the five coefficient estimates.

Table 4.2: Simulation values for n= 50, T=2, percentage of missingness=10%

			Sample size n= 50, T=2, percentage of missingness=10%				
Model	Balanced/Unbalanced		Parameter	Median Bias	MAD	Mean Bias	RMSE
Unconditional Logistic (With FE)	Balanced		β_1	-0.0910	0.2961	-0.1289	0.4744
			β_2	0.1718	0.7306	0.1752	1.1299
			β_3	-0.0603	0.4218	-0.1165	0.6786
			β_4	-0.0758	0.3291	-0.1193	0.5285
			β_5	-0.1364	0.5041	-0.2033	1.1341
	Unbalanced (But imputed)	Mean Imputation	β_1	-0.0575	0.2935	-0.1027	0.4629
			β_2	0.1148	0.7680	0.1289	1.1361
			β_3	-0.0392	0.4128	-0.0750	0.6734
			β_4	-0.0425	0.3367	-0.0865	0.5289
			β_5	-0.2349	0.4946	-0.3011	1.3780
		Last Value carried forward	β_1	0.0571	0.2435	0.0398	0.4067
			β_2	-0.0075	0.7142	-0.0195	1.0669
			β_3	0.1066	0.4020	0.0725	0.6392
			β_4	0.0912	0.3080	0.0638	0.4811
			β_5	-0.1576	0.4620	-0.2084	1.0964
		Median Imputation	β_1	-0.0935	0.2957	-0.1340	0.4718
			β_2	0.0951	0.7587	0.1151	1.1285
			β_3	-0.0293	0.4165	-0.0686	0.6739
			β_4	-0.0201	0.3356	-0.0620	0.5193
			β_5	-0.1140	0.4626	-0.1524	1.0553
Conditional Logistic	Balanced		β_1	-0.0762	0.2936	-0.1127	0.4625
			β_2	0.1507	0.7219	0.1585	1.1108
			β_3	-0.0451	0.4168	-0.1006	0.6655
			β_4	-0.0599	0.3233	-0.1030	0.5163
			β_5	-0.1223	0.4948	-0.1907	1.1798
	Unbalanced (But imputed)	Mean Imputation	β_1	-0.0435	0.2888	-0.0873	0.4522
			β_2	0.0997	0.7566	0.1134	1.1182
			β_3	-0.0259	0.4071	-0.0601	0.6617
			β_4	-0.0289	0.3290	-0.0712	0.5181
			β_5	-0.2202	0.4864	-0.2863	1.4248
		Last Value carried forward	β_1	0.0702	0.2386	0.0528	0.4015
			β_2	-0.0206	0.7056	-0.0326	1.0522
			β_3	0.1162	0.3950	0.0852	0.6312
			β_4	0.1042	0.3039	0.0767	0.4754
			β_5	-0.1437	0.4553	-0.1965	1.1465
		Median Imputation	β_1	-0.0782	0.2903	-0.1183	0.4602
			β_2	0.0804	0.7480	0.0999	1.1111
			β_3	-0.0162	0.4123	-0.0538	0.6625
			β_4	-0.0077	0.3316	-0.0472	0.5095
			β_5	-0.0999	0.4573	-0.1419	1.1211

Table 4.3: Simulation values for n= 50, T=2, percentage of missingness=30%

			Sample size n= 50, T=2, percentage of missingness=30%				
Model	Balanced/Unbalanced		Parameter	Median Bias	MAD	Mean Bias	RMSE
Unconditional Logistic (With FE)	Balanced		β_1	-0.0977	0.2815	-0.1324	0.4513
			β_2	0.0889	0.6959	0.1507	1.1536
			β_3	-0.1312	0.4111	-0.1439	0.6980
			β_4	-0.0839	0.3190	-0.1294	0.5218
			β_5	-0.1143	0.4932	-0.1849	1.2575
	Unbalanced (But imputed)	Mean Imputation	β_1	-0.0462	0.2745	-0.0682	0.4572
			β_2	0.0621	0.7774	0.0610	1.2152
			β_3	-0.0178	0.4811	-0.0385	0.7640
			β_4	0.0234	0.3522	-0.0149	0.5470
			β_5	-0.3333	0.5109	-0.3637	0.8748
		Last Value carried forward	β_1	0.3029	0.2138	0.2801	0.4434
			β_2	-0.3884	0.6439	-0.3295	1.0663
			β_3	0.3486	0.3847	0.3206	0.7053
			β_4	0.3583	0.2718	0.3335	0.5501
			β_5	0.0169	0.4217	-0.0210	0.9962
		Median Imputation	β_1	-0.0819	0.2603	-0.1229	0.4631
			β_2	0.0099	0.7636	0.0447	1.2013
			β_3	-0.0132	0.4716	-0.0206	0.7568
			β_4	0.0784	0.3488	0.0352	0.5589
			β_5	-0.0097	0.4590	-0.1847	1.6086
Conditional Logistic	Balanced		β_1	-0.0815	0.2781	-0.1161	0.4396
			β_2	0.0768	0.6865	0.1340	1.1341
			β_3	-0.1168	0.4052	-0.1275	0.6841
			β_4	-0.0699	0.3150	-0.1130	0.5094
			β_5	-0.0991	0.4864	-0.1757	1.3442
	Unbalanced (But imputed)	Mean Imputation	β_1	-0.0328	0.2711	-0.0540	0.4482
			β_2	0.0474	0.7656	0.0472	1.1979
			β_3	-0.0061	0.4762	-0.0248	0.7526
			β_4	0.0354	0.3470	-0.0015	0.5387
			β_5	-0.3172	0.5035	-0.3462	0.8575
		Last Value carried forward	β_1	0.3115	0.2107	0.2891	0.4454
			β_2	-0.3947	0.6366	-0.3379	1.0564
			β_3	0.3560	0.3806	0.3291	0.7019
			β_4	0.3662	0.2686	0.3419	0.5505
			β_5	0.0282	0.4166	-0.0131	1.0638
		Median Imputation	β_1	-0.0695	0.2535	-0.1082	0.4526
			β_2	-0.0026	0.7516	0.0312	1.1844
			β_3	-0.0007	0.4637	-0.0072	0.7457
			β_4	0.0900	0.3449	0.0478	0.5516
			β_5	0.0056	0.4508	-0.1873	1.7749

Table 4.4: Simulation values for n= 100, T=2, percentage of missingness=10%

			Sample size n= 100, T=2 , percentage of missingness=10%				
Model	Balanced/Unbalanced		Parameter	Median Bias	MAD	Mean Bias	RMSE
Unconditional Logistic (With FE)	Balanced		β_1	-0.0723	0.1842	-0.0760	0.2921
			β_2	0.0882	0.4448	0.0886	0.7197
			β_3	-0.0575	0.2796	-0.0645	0.4245
			β_4	-0.0509	0.2169	-0.0667	0.3295
			β_5	-0.0542	0.3214	-0.0644	0.5043
	Unbalanced (But imputed)	Mean Imputation	β_1	-0.0375	0.1887	-0.0477	0.2823
			β_2	0.0433	0.4901	0.0471	0.7456
			β_3	-0.0184	0.2966	-0.0290	0.4355
			β_4	-0.0142	0.2157	-0.0285	0.3321
			β_5	-0.1332	0.3145	-0.1511	0.5304
		Last Value carried forward	β_1	0.1018	0.1695	0.0906	0.2785
			β_2	-0.0969	0.4431	-0.0993	0.6851
			β_3	0.1110	0.2712	0.1037	0.4234
			β_4	0.1258	0.2084	0.1044	0.3226
			β_5	-0.0507	0.3050	-0.0744	0.4820
		Median Imputation	β_1	-0.0705	0.1868	-0.0771	0.2870
			β_2	0.0342	0.4789	0.0403	0.7397
			β_3	-0.0063	0.2955	-0.0221	0.4318
			β_4	0.0075	0.2227	-0.0129	0.3253
			β_5	-0.0072	0.3099	-0.0229	0.4811
Conditional Logistic	Balanced		β_1	-0.0649	0.1834	-0.0687	0.2881
			β_2	0.0812	0.4420	0.0812	0.7139
			β_3	-0.0510	0.2787	-0.0572	0.4204
			β_4	-0.0433	0.2148	-0.0595	0.3256
			β_5	-0.0470	0.3196	-0.0575	0.5000
	Unbalanced (But imputed)	Mean Imputation	β_1	-0.0312	0.1871	-0.0408	0.2791
			β_2	0.0361	0.4863	0.0402	0.7402
			β_3	-0.0119	0.2940	-0.0222	0.4321
			β_4	-0.0069	0.2138	-0.0217	0.3292
			β_5	-0.1263	0.3123	-0.1438	0.5250
		Last Value carried forward	β_1	0.1074	0.1684	0.0964	0.2786
			β_2	-0.1028	0.4406	-0.1050	0.6815
			β_3	0.1167	0.2696	0.1095	0.4222
			β_4	0.1312	0.2069	0.1102	0.3225
			β_5	-0.0448	0.3034	-0.0678	0.4780
		Median Imputation	β_1	-0.0637	0.1856	-0.0701	0.2831
			β_2	0.0272	0.4760	0.0335	0.7344
			β_3	0.0002	0.2925	-0.0155	0.4286
			β_4	0.0138	0.2214	-0.0062	0.3228
			β_5	-0.0016	0.3072	-0.0165	0.4777

Table 4.5: Simulation values for n= 100, T=2, percentage of missingness=30%

			Sample size n= 100, T=2, percentage of missingness=30%				
Model	Balanced/Unbalanced		Parameter	Median Bias	MAD	Mean Bias	RMSE
Unconditional Logistic (With FE)	Balanced		β_1	-0.0350	0.2030	-0.0542	0.2939
			β_2	0.0571	0.4212	0.0649	0.6997
			β_3	-0.0396	0.2874	-0.0622	0.4292
			β_4	-0.0393	0.2089	-0.0516	0.3186
			β_5	-0.0896	0.3201	-0.1008	0.5002
	Unbalanced (But imputed)	Mean Imputation	β_1	0.0132	0.1841	-0.0030	0.2798
			β_2	-0.0593	0.5078	-0.0210	0.7839
			β_3	0.0818	0.3000	0.0467	0.4860
			β_4	0.0667	0.2306	0.0587	0.3542
			β_5	-0.2915	0.3283	-0.3170	0.5968
		Last Value carried forward	β_1	0.3590	0.1419	0.3477	0.4117
			β_2	-0.3936	0.4105	-0.3979	0.7441
			β_3	0.3942	0.2360	0.3832	0.5412
			β_4	0.3950	0.1892	0.3859	0.4777
			β_5	0.0403	0.2560	0.0345	0.3992
		Median Imputation	β_1	-0.0475	0.1829	-0.0630	0.2874
			β_2	-0.0797	0.4962	-0.0392	0.7810
			β_3	0.0935	0.3069	0.0620	0.4865
			β_4	0.1041	0.2393	0.0914	0.3613
			β_5	0.0027	0.2973	-0.0108	0.4640
Conditional Logistic	Balanced		β_1	-0.0282	0.2020	-0.0471	0.2904
			β_2	0.0506	0.4187	0.0577	0.6942
			β_3	-0.0329	0.2853	-0.0550	0.4252
			β_4	-0.0323	0.2067	-0.0445	0.3152
			β_5	-0.0832	0.3175	-0.0936	0.4955
	Unbalanced (But imputed)	Mean Imputation	β_1	0.0195	0.1830	0.0033	0.2778
			β_2	-0.0649	0.5047	-0.0270	0.7792
			β_3	0.0875	0.2981	0.0526	0.4834
			β_4	0.0725	0.2289	0.0645	0.3529
			β_5	-0.2832	0.3260	-0.3090	0.5897
		Last Value carried forward	β_1	0.3626	0.1409	0.3515	0.4142
			β_2	0.3970	0.4081	-0.4014	0.7429
			β_3	-0.3976	0.2343	0.3869	0.5422
			β_4	0.3986	0.1879	0.3896	0.4796
			β_5	0.0460	0.2542	0.0400	0.3974
		Median Imputation	β_1	-0.0409	0.1814	-0.0565	0.2841
			β_2	-0.0853	0.4931	-0.0451	0.7764
			β_3	0.0985	0.3048	0.0678	0.4842
			β_4	0.1094	0.2377	0.0970	0.3605
			β_5	0.0091	0.2960	-0.0048	0.4612

Table 4.6: Simulation values for n= 250, T=2, percentage of missingness=10%

			Sample size n= 250, T=2, , percentage of missingness=10%				
Model	Balanced/Unbalanced		Parameter	Median Bias	MAD	Mean Bias	RMSE
Unconditional Logistic (With FE)	Balanced		β_1	-0.0222	0.1213	-0.0310	0.1787
			β_2	0.0205	0.2935	0.0153	0.4325
			β_3	-0.0359	0.1718	-0.0382	0.2609
			β_4	-0.0276	0.1295	-0.0330	0.1921
			β_5	-0.0500	0.1957	-0.0579	0.2884
	Unbalanced (But imputed)	Mean Imputation	β_1	-0.0001	0.1189	-0.0061	0.1757
			β_2	-0.0249	0.3078	-0.0234	0.4479
			β_3	-0.0011	0.1766	0.00003	0.2647
			β_4	0.0067	0.1370	0.0034	0.1969
			β_5	-0.1288	0.1938	-0.1310	0.3121
		Last Value carried forward	β_1	0.1403	0.1068	0.1270	0.2071
			β_2	-0.1694	0.2837	-0.1557	0.4536
			β_3	0.1387	0.1639	0.1301	0.2780
			β_4	0.1373	0.1273	0.1286	0.2254
			β_5	-0.0629	0.1817	-0.0610	0.2709
		Median Imputation	β_1	-0.0303	0.1167	-0.0369	0.1779
			β_2	-0.0227	0.3088	-0.0296	0.4445
			β_3	0.0058	0.1759	0.0050	0.2635
			β_4	0.0201	0.1338	0.0185	0.1963
			β_5	0.0049	0.1809	0.0021	0.2756
Conditional Logistic	Balanced		β_1	-0.0194	0.1208	-0.0283	0.1777
			β_2	0.0176	0.2928	0.0126	0.4312
			β_3	-0.0333	0.1714	-0.0355	0.2597
			β_4	-0.0250	0.1291	-0.0303	0.1911
			β_5	-0.0472	0.1950	-0.0552	0.2872
	Unbalanced (But imputed)	Mean Imputation	β_1	0.0025	0.1186	-0.0035	0.1751
			β_2	-0.0273	0.3070	-0.0259	0.4469
			β_3	0.0013	0.1762	0.0026	0.2640
			β_4	0.0092	0.1363	0.0060	0.1964
			β_5	-0.1262	0.1932	-0.1282	0.3102
		Last Value carried forward	β_1	0.1426	0.1064	0.1292	0.2080
			β_2	-0.1713	0.2829	-0.1578	0.4533
			β_3	0.1407	0.1633	0.1323	0.2784
			β_4	0.1394	0.1269	0.1308	0.2263
			β_5	-0.0602	0.1813	-0.0584	0.2696
		Median Imputation	β_1	-0.0278	0.1163	-0.0343	0.1769
			β_2	-0.0251	0.3080	-0.0321	0.4435
			β_3	0.0084	0.1754	0.0075	0.2628
			β_4	0.0227	0.1333	0.0211	0.1960
			β_5	0.0075	0.1806	0.0046	0.2749

Table 4.7: Simulation values for n= 250, T=2, percentage of missingness=30%

			Sample size n= 250, T=2, percentage of missingness=30%				
Model	Balanced/Unbalanced		Parameter	Median Bias	MAD	Mean Bias	RMSE
Unconditional Logistic (With FE)	Balanced		β_1	-0.0236	0.1198	-0.0309	0.1789
			β_2	0.0208	0.2892	0.0254	0.4361
			β_3	-0.0154	0.1612	-0.0258	0.2571
			β_4	-0.0325	0.1283	-0.0319	0.1983
			β_5	-0.0475	0.2137	-0.0605	0.3154
	Unbalanced (But imputed)	Mean Imputation	β_1	0.0266	0.1203	0.0136	0.1780
			β_2	-0.0823	0.3292	-0.0884	0.4975
			β_3	0.0798	0.1743	0.0760	0.2936
			β_4	0.0646	0.1440	0.0701	0.2223
			β_5	-0.2303	0.2009	-0.2509	0.3944
		Last Value carried forward	β_1	0.3672	0.0942	0.3594	0.3851
			β_2	-0.4144	0.2678	-0.4091	0.5744
			β_3	0.4144	0.1485	0.4065	0.4685
			β_4	0.3984	0.1165	0.3999	0.4332
			β_5	0.0941	0.1585	0.0901	0.2547
		Median Imputation	β_1	-0.0327	0.1155	-0.0436	0.1815
			β_2	-0.0895	0.3209	-0.1048	0.4965
			β_3	0.0992	0.1780	0.0898	0.2958
			β_4	0.1013	0.1397	0.1048	0.2348
			β_5	0.0658	0.1935	0.0570	0.3018
Conditional Logistic	Balanced		β_1	-0.0211	0.1195	-0.0282	0.1780
			β_2	0.0181	0.2887	0.0227	0.4347
			β_3	-0.0127	0.1607	-0.0231	0.2561
			β_4	-0.0298	0.1281	-0.0292	0.1973
			β_5	-0.0449	0.2132	-0.0578	0.3140
	Unbalanced (But imputed)	Mean Imputation	β_1	0.0293	0.1199	0.0160	0.1777
			β_2	-0.0844	0.3283	-0.0906	0.4967
			β_3	0.08210	0.1739	0.0782	0.2935
			β_4	0.0668	0.1436	0.0724	0.2225
			β_5	-0.2275	0.2005	-0.2480	0.3920
		Last Value carried forward	β_1	0.3686	0.0940	0.3609	0.3864
			β_2	-0.4156	0.2671	-0.4104	0.5747
			β_3	0.4158	0.1480	0.4078	0.4694
			β_4	0.3998	0.1162	0.4013	0.4343
			β_5	0.0960	0.1581	0.0921	0.2549
		Median Imputation	β_1	-0.0303	0.1152	-0.0411	0.1804
			β_2	-0.0918	0.3202	-0.1069	0.4958
			β_3	0.1015	0.1776	0.0919	0.2958
			β_4	0.1033	0.1393	0.1069	0.2353
			β_5	0.0678	0.1932	0.0592	0.3016

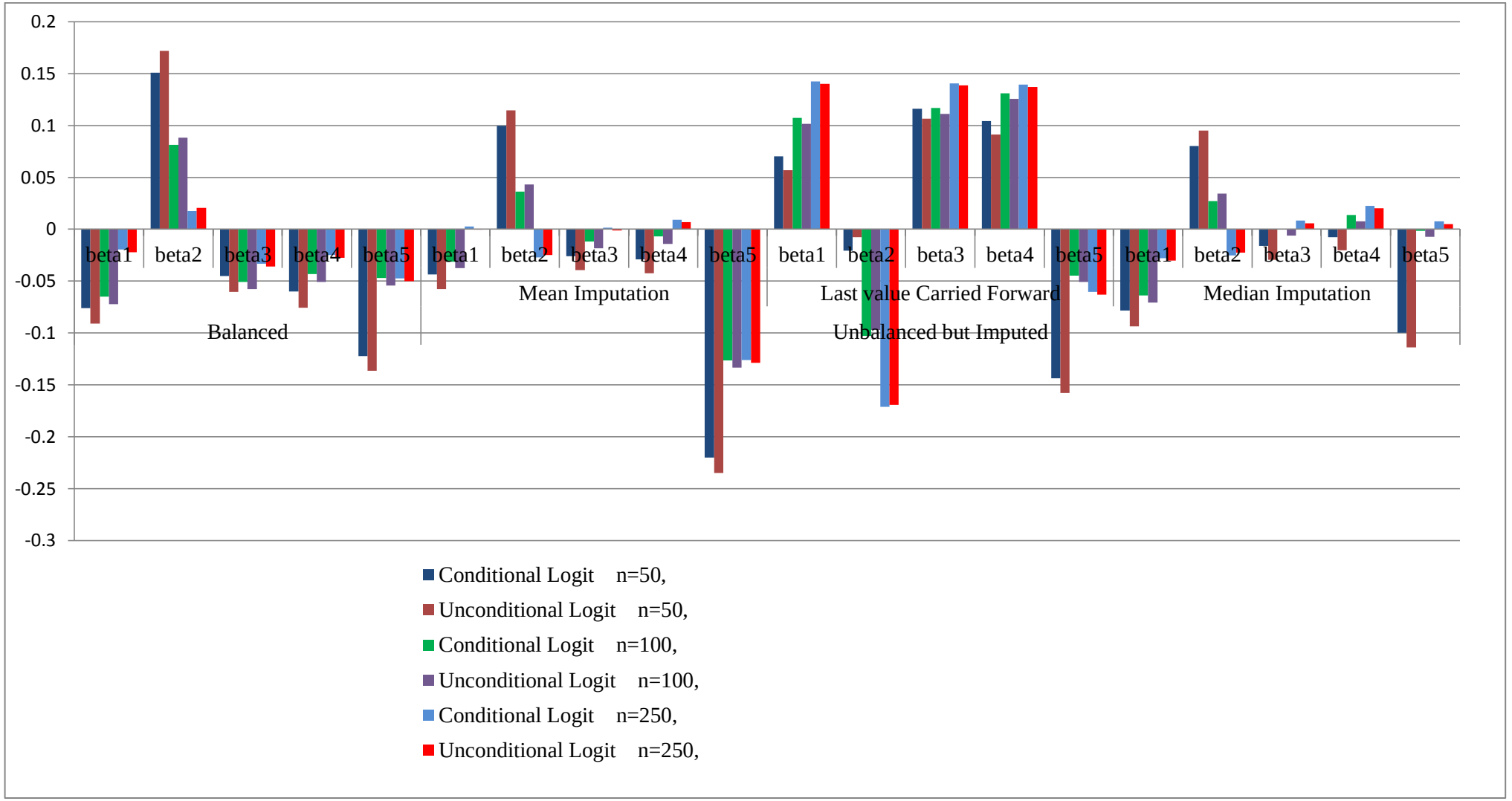

Figure 4.1: Median Bias for 10% missingness with varying sample sizes

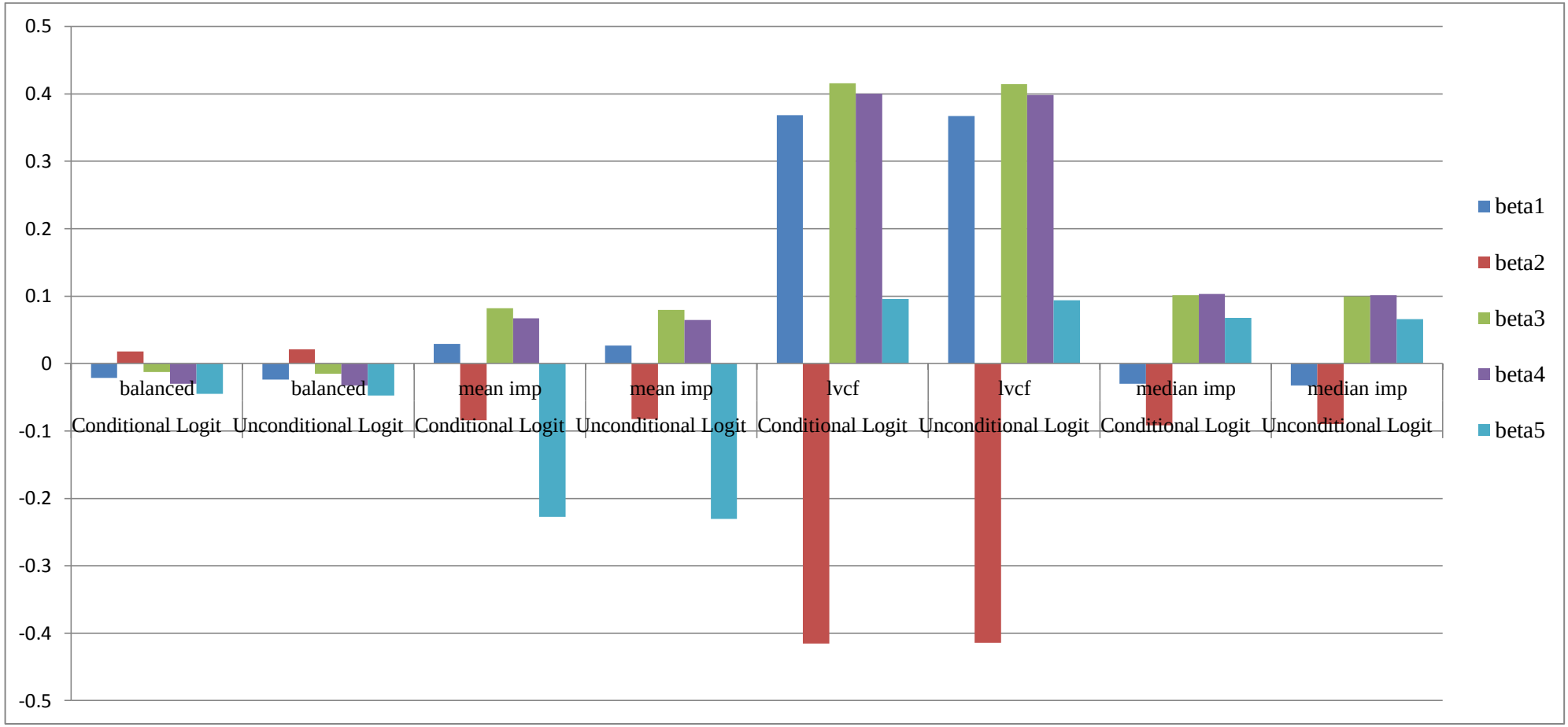

Figure 4.2: Median Bias variation across estimators

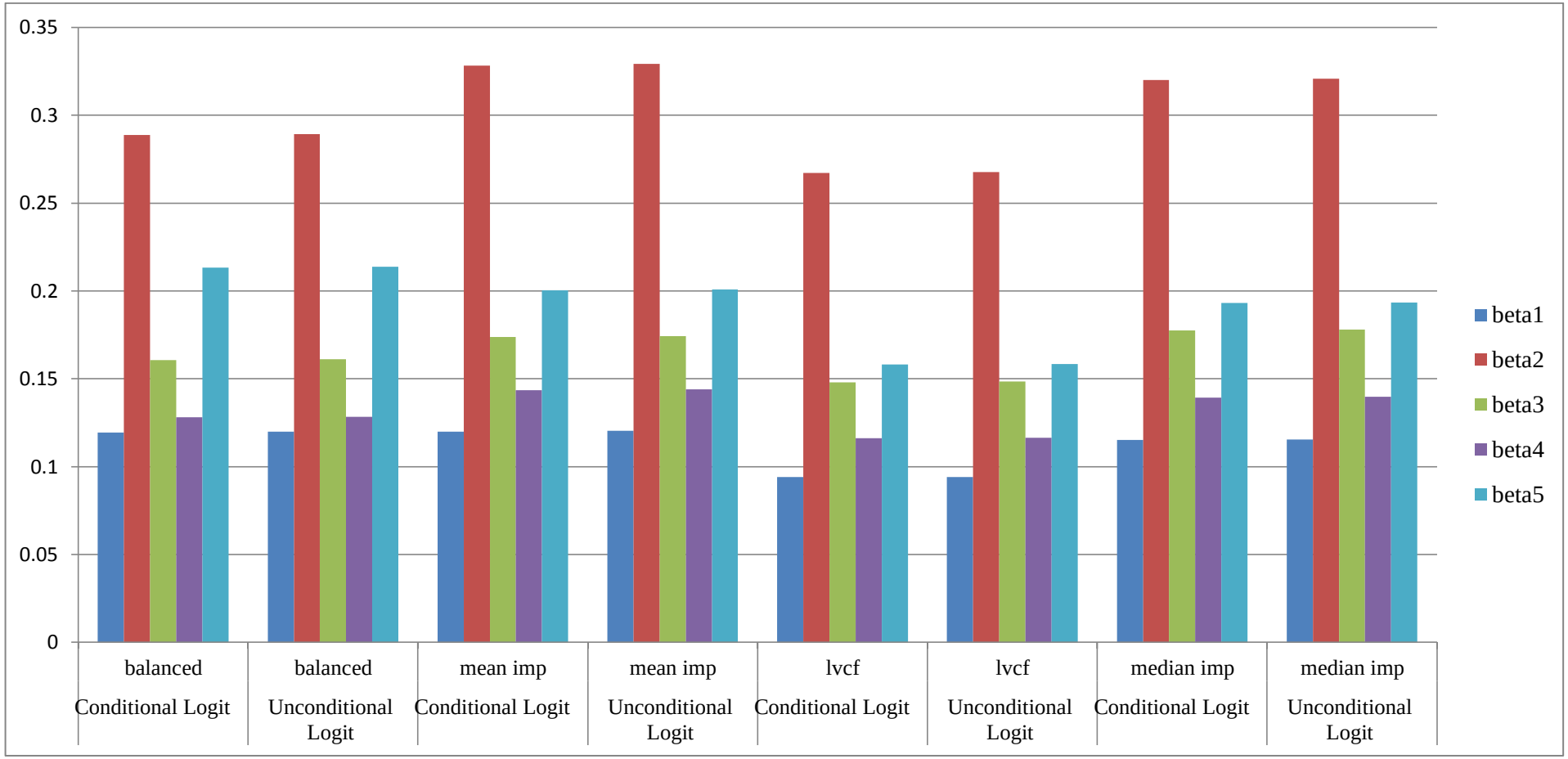

Figure 4.3: MAD variation across estimators

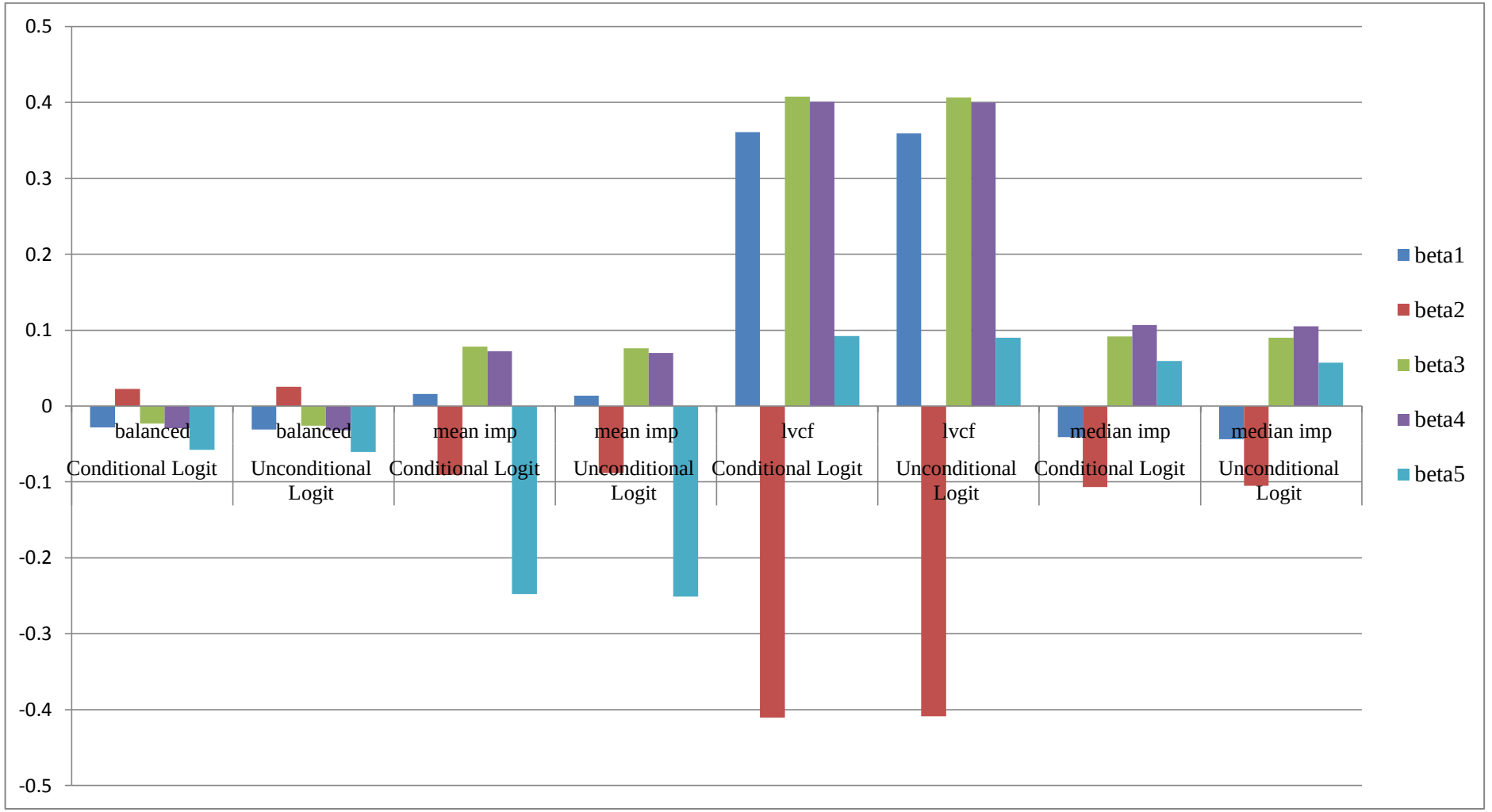

Figure 4.4: Mean Bias variation across estimators

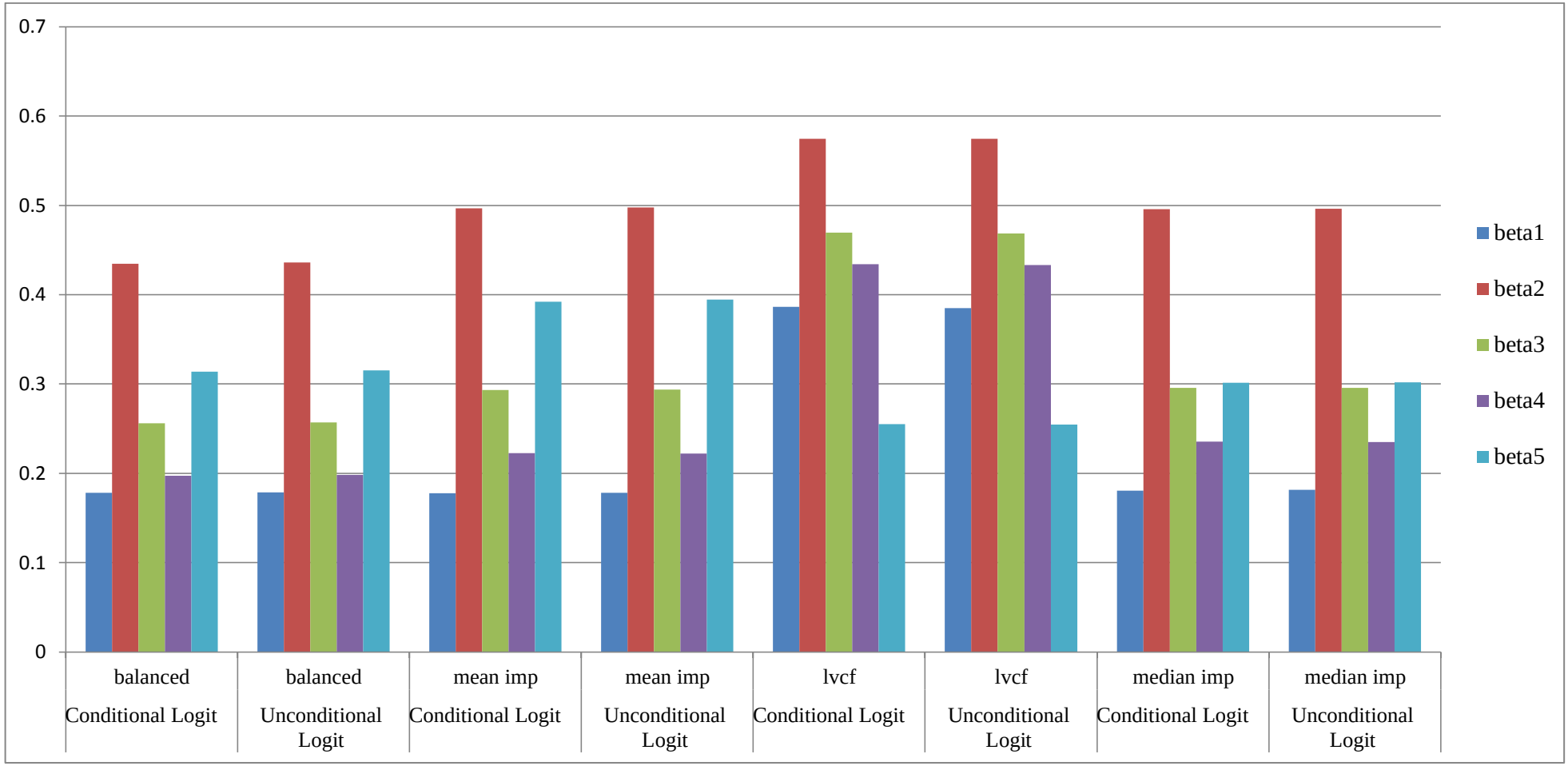

Figure 4.5: RMSE variation across estimators

4.4 Discussion

Similar to most empirical studies, sample size matters, both for the bias and the precision of the parameter estimates. Indeed, all the reported measures (median bias, median absolute deviation, mean bias and the root mean square errors) are observed to reduce significantly as the sample size increases for both the unconditional and conditional logistic models. Figures 2 and 4 give the bias variations across estimators. The curves follow the same trend for the positive parameters; beta1, beta3, beta4 and beta5. For beta2, whose value was set as -1, the trend is opposite to that of the positive parameters and almost symmetric about the horizontal axis.

The conditional maximum likelihood estimators yield lower median biases than the unconditional maximum likelihood estimators irrespective of sample size and the imputation technique used (Figure 4.1). This attribute supports the superiority of conditional ML estimation over the unconditional ML estimation for the logistic panel data model.

Comparatively, for n=250, imputation by last value carried forward (LVCF) increases both the mean and median biases with respect to the balanced panel set. The bias estimates obtained when mean and median imputation techniques were performed are however inconsistent although most of them also indicate larger magnitudes than those from balanced data set.

From Figure 4.3 we observe that LVCF provides the smallest MAD for all the five parameters irrespective of the percentage missingness and sample size. Mean and median imputation however increase the MAD for all the parameters.

The median biases are generally lower for the conditional ML estimator than for the unconditional estimator. Median bias is observed to reduce in absolute value as sample size increases an indication that for very large samples the estimators become unbiased, that is, the conditional estimator is asymptotically unbiased. The data set obtained by median imputation gives lower median biases for all the parameter estimates beta1 to beta5. Figures 4.2 to 4.5

however show very significant changes when we impute by carrying forward the last or previous value. As a matter of fact this is expected given that the last value being carried forward may have been an individual specific value which could not satisfactorily replace the missing value for the succeeding individual.

Imputation by last value carried forward yields the highest RMSE values for most of the parameter estimates. Figure 4.5 shows that the RMSE values are not deviated in value as much from the complete or balanced case when mean and median imputations are performed. The same trend is observed from Figure 4.3 for the MAD values. Beta2 however yields the highest RMSE and MAD values across all sets of data. This can be attributed to the fact that the set value for beta2 was less than zero. As such, mean imputation and median imputations tend to preserve precision of the estimates better than last value carried forward.

CHAPTER 5

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

In this study, we have discussed the estimation method and procedures for estimating nonlinear (binary choice logistic) panel data regression models.

The major concern being the effect of non-responses (missingness) in the parameter estimates, we have developed an analogous estimation process for the logistic panel data model in the presence of imputed values to replace the missing observations. Detailed derivations of the conditional maximum likelihood estimator for the logistic panel data model are discussed. In particular, we eliminate the incidental parameters from the logistic model thereby curbing the incidental parameter problem which would otherwise have made parameter estimation complicated. The likelihood functions were derived from the logistic probability distribution of the dependent variable and solved using the Newton-Raphson algorithm due to the nonlinearity of the likelihood functions. The maximum likelihood estimates for the parameters are obtainable easily if the data set is balanced. In the cases of unbalancedness we employed three simple imputation techniques (mean imputation, last value carried forward and median imputation) to make the data balanced. Through Monte Carlo simulations, comparisons are made for the imputation techniques so as to assess the magnitude of bias and efficiency of each imputation technique on the parameter estimates. For both estimators (unconditional MLE and conditional MLE), we reported the median bias, the median absolute deviation (MAD), the mean bias, and the root mean squared error (RMSE) for all the five coefficient estimates.

5.2 Conclusion

A key importance of deriving estimators is to increase the theoretical understanding of the estimation process and also reduce the computational complexity while estimating logistic panel data models. For completeness we examined the properties of the estimators in this study through Monte Carlo simulations. As observed from the Monte Carlo results, unbalancedness in a data set biases the parameter estimates and the different imputation techniques employed in this study respond differently to the bias and efficiency of the estimates. The simulation results also show that the parameter estimates for the conditional logistic model are less biased than those from the unconditional logistic model without sacrificing on the precision for balanced panels. Two imputation techniques, mean imputation and median imputation, prove to be superior over the LVCF technique whenever a panel data set is unbalanced. In fact, mean imputation and median imputations tend to preserve precision of the estimates better than last value carried forward.

5.3 Recommendations

This study gives an insight into the impact of missingness due to non-responses in binary choice panel data. Section 3.7 outlined various imputation techniques only three of which have been considered in the Monte Carlo simulations. The analysis given herein does not, therefore, provide a once and for all clear-cut answer to the question of how nonresponse in panel data should be handled. As a recommendation, further developments can be done on this study by considering other imputation techniques and also using different time periods greater than T=2.

REFERENCES

1. Baltagi, B.H., (1985). Pooling cross-sections with unequal time-series lengths, *Economic Letters*, **18**, 133-136.
2. Baltagi, B.H. (1995). *Econometric Analysis of Panel Data*, New York, John Wiley.
3. Baltagi, B.H. (2001). *Econometric Analysis of Panel Data, 2nd edition*, New York, John Wiley.
4. Biorn, E., (1981). Estimating economic relations from incomplete cross section and time series data, *Journal of Econometrics* **16**, 221-236
5. Chamberlain, G., (1980). Analysis of Covariance with Qualitative Data, *Review of Economic Studies* **47**, 225-238
6. Chamberlain, G., (1984). Panel Data: *Handbook of Econometrics, vol. 2*, (Griliches Z. and Intriligator M.D. eds.), 1247-1318, Elsevier Science, Amsterdam.
7. Charbonneau K. B. (2012). *Multiple Fixed Effects in Nonlinear Panel Data Models-Theory and Evidence*, Princeton University
8. DeSarbo, W.S., Green, P.E. and Carroll, J.D., (1986). Missing data in product-concept testing. *Decision Sciences* **17**, 163–185.
9. Frane, J.W., (1976). Some simple procedures for handling missing data in multivariate analysis. *Psychometrika* **41**, 409–415.
10. Greene, W. (2000). *Econometric Analysis, 4th ed.*, Prentice Hall, Englewood Cliffs.
11. Greene, W. (2001). *Estimating Sample Selection Models with Panel Data*, Manuscript, Department of Economics, Stern School of Business, NYU.
12. Greene, W., (2004a). The behaviour of the maximum likelihood estimator of limited dependent variable models in the presence of fixed effects. *Econometrics Journal* **7**(1): 98.

13. Griliches, Z. and Hausman, J., **(**1986**).** Errors in variables in panel data, *Journal of Econometrics* **31**(1), 93-118.

14. Hahn, J., and Newey, W. (2004). Jackknife and Analytical Bias Reduction for Nonlinear Panel data models, *Econometrica*, **72**(4), 1295-1319.

15. Hausman, J., Hall, B.H. and Griliches, Z., (1984). Econometric models for count data with an application to the patents-R&D Relationship. *Econometrica* **52** (4), 909-938.

16. Hsiao, C., (2003). *Analysis of Panel Data*, 2nd edition, New York, Cambridge University Press.

17. Kao, C. and Schnell, J. F., (1987). Errors in variables in panel data with a binary dependent variable, *Economics Letters* **24**(4), 339-342

18. Kline, R.B., (1998). *Principles and Practice of Structural Equation Modelling*. Guilford Press, New York.

19. Kromrey, J.D. and Heines, C.V. (1994). Non-randomly missing data in multiple regression: an empirical comparison of common missing-data. *Educational and Psychological Measurement* **54**(3), 573–593.

20. Laird, N.M., (1988). Missing data in longitudinal studies. *Statistics in Medicine* **7**, 305–315.

21. Lancaster, T., (2000). The incidental parameter problem since 1948, *Journal of Econometrics*, **95**, 391–413.

22. Little, R. J. A. and Rubin, D. B. (1987). *Statistical analysis with missing data*. New York: John Wiley & Sons.

23. Malhotra, N.K., (1987). Analysing marketing research data with incomplete information on the dependent variable. *Journal of Marketing Research* **24**, 74–84.

24. Manski, C. F., (1987). Semi-parametric Analysis of Random Effects Linear Models from Binary Panel Data, *Econometrica*, **55**(2), 357-362.

25. Matyas, L. and Lovrics, L. (1991). Missing observations and panel data- A Monte-Carlo analysis, *Economics Letters* **37**(1), 39-44.

26. Munkin, M. and Trivedi, P., (2000). *Econometric Analysis of a Self-Selection Model with Multiple Outcomes Using Simulation-Based Estimation: An Application to the Demand for Healthcare*, Manuscript, Department of Economics, Indiana University.

27. Neyman, J. and Scott, E.L., (1948). Consistent estimation from partially consistent observations. *Econometrica* **16**, 1-32.

28. Rasch, G. (1960). *Probabilistic Models for Some Intelligence and Attainment Tests*, Denmarks Paedagogiske Institute, Copenhagen.

29. Rasch, G. (1961). *On the general laws and the meaning of measurement in psychology*, Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, 4.

30. Raymond, M.R., (1986). Missing data in evaluation research. *Evaluation and the Health Profession* **9**, 395–420.

31. Roth, P.L., (1994). Missing data: a conceptual review for applied psychologists. *Personnel Psychology* **47** (3), 537–560.

32. Rubin, D.B. (1976). Inference and missing data. *Biometrika* **63**, 581-592

33. Ruud, P.A., (1991). Extensions of estimation methods using the EM algorithm. *Journal of Econometrics* **49**, 305–341.

34. Stefanski, L.A. and Carroll R.J. (1985), Covariate measurement error in logistic regression, *The Annals of Statistics*, 1335-1351.

35. Tsikriktsis, N., (2005). A review of techniques for treating missing data in OM survey research. *Journal of Operations Management* **24**(1), 53-62.

36. Verbeek, M.J.C.M. and Nijman, T.E., (1990). *Testing for selectivity bias in panel data models*, Tilburg University, Centre for Economic Research.

APPENDIX

A1. R CODES FOR MONTE CARLO SIMULATION

A1.1 n= 50 with 10% missingness

```
set.seed(123467)        # Use this to make the randomly generated data the same each time you run the simulation#

j= 1:5
A <- array(0, dim=c(N.iter,5))
B <- array(0, dim=c(N.iter,5))
C <- array(0, dim=c(N.iter,5))
D <- array(0, dim=c(N.iter,5))
E <- array(0, dim=c(N.iter,5))
F <- array(0, dim=c(N.iter,5))
G <- array(0, dim=c(N.iter,5))
H <- array(0, dim=c(N.iter,5))

for(i in 1:N.iter){

N.iter=1000
n=50                    # vary n
t=2          # vary t and change time in pData
nt=n*t
b1=1
b2=-1
b3=1
b4=1
b5=1
ai=rnorm(n,0,1)
```

```
x1=rnorm(nt,0,1)
x2 <- runif(nt,0,1)
x3 <- rnorm(nt,0.5,0.5)
x4 <- rbinom(nt,2,0.65)              # nt different such numbers
hit=rnorm(nt,0,1)
uit=rnorm(nt,0,1)
vit=log(abs(uit/(1+uit)))
x5 <- ifelse(x1+hit>0, 1, 0)
alphai <- sqrt(t)*sum(x1)/n+ai #alphai simulation
ci <- kronecker(alphai ,matrix(1,t,1))#kronecker of alphai
wit=ci+b1*x1+b2*x2+b3*x3+b4*x4+b5*x5
zit= wit+vit
y <- ifelse(zit>0, 1, 0)
pData <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n),
y,zit,ci,x1,x2,x3,x4,x5)
```

 # Randomly Insert A Certain Proportion Of NAs Into A Dataframe #

```
pData2<- cbind(x1,x2,x3,x4,x5)
###################################
NAins <-  NAinsert <- function(df, prop = .1){
 n <- nrow(df)
 m <- ncol(df)
 num.to.na <- ceiling(prop*n*m)
 id <- sample(0:(m*n-1), num.to.na, replace = FALSE)
 rows <- id %/% m + 1
 cols <- id %% m + 1
 sapply(seq(num.to.na), function(x){
 df[rows[x], cols[x]] <<- NA
 }
 )
 return(df)
```

```
}
##############
# TRY IT OUT #
##############
XXX<-NAins(pData2, .1)
XXX
```

 # Mean Imputation #

```
dat<-sapply(seq_len(ncol(XXX)),function(i) {XXX[,i][is.na(XXX[,i])]<-
mean(XXX[,i],na.rm=TRUE);XXX[,i]})
dat2 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat)
dat2
```

 # Previous value carried forward Imputation #

```
fillNAByPreviousData <- function(column) {
   # At first we find out which columns contain NAs
   navals <- which(is.na(column))
   # and which columns are filled with data.
   filledvals <- which(! is.na(column))
   # If there would be no NAs following each other, navals-1 would give the
   # entries we need. In our case, however, we have to find the last column filled for
   # each value of NA. We may do this using the following sapply trick:
   fillup <- sapply(navals, function(x) max(filledvals[filledvals < x]))
   # And finally replace the NAs with our data.
   column[navals] <- column[fillup]
   column
}
 dat1=fillNAByPreviousData(XXX)
dat3 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat1)
dat3
```

 # Median Imputation #

```
f=function(x){
  x<-as.numeric(as.character(x)) #first convert each column into numeric if it is from factor
  x[is.na(x)] =median(x, na.rm=TRUE) #convert the item with NA to median value from the
column
  x #display the column
}
dat.1=data.frame(apply(XXX,2,f))
dat4<- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat.1)
dat4

glm.out1 = glm(y ~x1+x2+x3+x4+x5,family=binomial(logit), data=pData)
clogit1=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = pData)
glm.out2 = glm(y ~X1+X2+X3+X4+X5,family=binomial(logit), data=dat2)
clogit2=clogit((y ~ (X1 +X2+X3+X4+X5)),  strata (id), data = dat2)
glm.out3 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat3)
clogit3=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat3)
glm.out4 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat4)
clogit4=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat4)

A[i,] <- glm.out1$coef[j+1]
B[i,] <- clogit1$coef[j]
C[i,] <- glm.out2$coef[j+1]
D[i,] <- clogit2$coef[j]
E[i,] <- glm.out3$coef[j+1]
F[i,] <- clogit3$coef[j]
G[i,] <- glm.out4$coef[j+1]
H[i,] <- clogit4$coef[j]
}
```

#median absolute deviation#

Median Absolute Deviation (MAD) or Absolute Deviation Around the Median as stated in the title, is a robust measure of central tendency. Robust statistics are statistics with good

performance for data drawn from a wide range of non-normally distributed probability distributions. Unlike the standard mean/standard deviation combo, MAD is not sensitive to the presence of outliers.#

```
madA<- c(mad(A[,1],constant=1), mad(A[,2], constant=1), mad(A[,3], constant=1), mad(A[,4], constant=1), mad(A[,5], constant=1))
madB<- c(mad(B[,1],constant=1), mad(B[,2], constant=1), mad(B[,3], constant=1), mad(B[,4], constant=1), mad(B[,5], constant=1))
madC<- c(mad(C[,1],constant=1), mad(C[,2], constant=1), mad(C[,3], constant=1), mad(C[,4], constant=1), mad(C[,5], constant=1))
madD<- c(mad(D[,1],constant=1), mad(D[,2], constant=1), mad(D[,3], constant=1), mad(D[,4], constant=1), mad(D[,5], constant=1))
madE<- c(mad(E[,1],constant=1), mad(E[,2], constant=1), mad(E[,3], constant=1), mad(E[,4], constant=1), mad(E[,5], constant=1))
madF<- c(mad(F[,1],constant=1), mad(F[,2], constant=1), mad(F[,3], constant=1), mad(F[,4], constant=1), mad(F[,5], constant=1))
madG<- c(mad(G[,1],constant=1), mad(G[,2], constant=1), mad(G[,3], constant=1), mad(G[,4], constant=1), mad(G[,5], constant=1))
madH<- c(mad(H[,1],constant=1), mad(H[,2], constant=1), mad(H[,3], constant=1), mad(H[,4], constant=1), mad(H[,5], constant=1))
```

#median bias#

```
median.biasA<- c(b1-median(A[,1]), b2-median(A[,2]),b3-median(A[,3]),b4-median(A[,4]),b5-median(A[,5]))
median.biasB<- c(b1-median(B[,1]), b2-median(B[,2]),b3-median(B[,3]),b4-median(B[,4]),b5-median(B[,5]))
median.biasC<- c(b1-median(C[,1]), b2-median(C[,2]),b3-median(C[,3]),b4-median(C[,4]),b5-median(C[,5]))
median.biasD<- c(b1-median(D[,1]), b2-median(D[,2]),b3-median(D[,3]),b4-median(D[,4]),b5-median(D[,5]))
median.biasE<- c(b1-median(E[,1]), b2-median(E[,2]),b3-median(E[,3]),b4-median(E[,4]),b5-median(E[,5]))
```

```
median.biasF<- c(b1-median(F[,1]), b2-median(F[,2]),b3-median(F[,3]),b4-median(F[,4]),b5-median(F[,5]))
median.biasG<- c(b1-median(G[,1]), b2-median(G[,2]),b3-median(G[,3]),b4-median(G[,4]),b5-median(G[,5]))
median.biasH<- c(b1-median(H[,1]), b2-median(H[,2]),b3-median(H[,3]),b4-median(H[,4]),b5-median(H[,5]))
```

#mean bias#

```
mean.biasA<- c(b1-mean(A[,1]), b2-mean(A[,2]),b3-mean(A[,3]),b4-mean(A[,4]),b5-mean(A[,5]))
mean.biasB<- c(b1-mean(B[,1]), b2-mean(B[,2]),b3-mean(B[,3]),b4-mean(B[,4]),b5-mean(B[,5]))
mean.biasC<- c(b1-mean(C[,1]), b2-mean(C[,2]),b3-mean(C[,3]),b4-mean(C[,4]),b5-mean(C[,5]))
mean.biasD<- c(b1-mean(D[,1]), b2-mean(D[,2]),b3-mean(D[,3]),b4-mean(D[,4]),b5-mean(D[,5]))
mean.biasE<- c(b1-mean(E[,1]), b2-mean(E[,2]),b3-mean(E[,3]),b4-mean(E[,4]),b5-mean(E[,5]))
mean.biasF<- c(b1-mean(F[,1]), b2-mean(F[,2]),b3-mean(F[,3]),b4-mean(F[,4]),b5-mean(F[,5]))
mean.biasG<- c(b1-mean(G[,1]), b2-mean(G[,2]),b3-mean(G[,3]),b4-mean(G[,4]),b5-mean(G[,5]))
mean.biasH<- c(b1-mean(H[,1]), b2-mean(H[,2]),b3-mean(H[,3]),b4-mean(H[,4]),b5-mean(H[,5]))
```

#residual errors#

```
rmseA<-c(sqrt(mean((b1-A[,1])^2)), sqrt(mean((b2-A[,2])^2)), sqrt(mean((b3-A[,3])^2)), sqrt(mean((b4-A[,4])^2)), sqrt(mean((b5-A[,5])^2)))
rmseB<-c(sqrt(mean((b1-B[,1])^2)), sqrt(mean((b2-B[,2])^2)), sqrt(mean((b3-B[,3])^2)), sqrt(mean((b4-B[,4])^2)), sqrt(mean((b5-B[,5])^2)))
rmseC<-c(sqrt(mean((b1-C[,1])^2)), sqrt(mean((b2-C[,2])^2)), sqrt(mean((b3-C[,3])^2)), sqrt(mean((b4-C[,4])^2)), sqrt(mean((b5-C[,5])^2)))
```

```
rmseD<-c(sqrt(mean((b1-D[,1])^2)), sqrt(mean((b2-D[,2])^2)), sqrt(mean((b3-D[,3])^2)),
sqrt(mean((b4-D[,4])^2)), sqrt(mean((b5-D[,5])^2)))
rmseE<-c(sqrt(mean((b1-E[,1])^2)), sqrt(mean((b2-E[,2])^2)), sqrt(mean((b3-E[,3])^2)),
sqrt(mean((b4-E[,4])^2)), sqrt(mean((b5-E[,5])^2)))
rmseF<-c(sqrt(mean((b1-F[,1])^2)), sqrt(mean((b2-F[,2])^2)), sqrt(mean((b3-F[,3])^2)),
sqrt(mean((b4-F[,4])^2)), sqrt(mean((b5-F[,5])^2)))
rmseG<-c(sqrt(mean((b1-G[,1])^2)), sqrt(mean((b2-G[,2])^2)), sqrt(mean((b3-G[,3])^2)),
sqrt(mean((b4-G[,4])^2)), sqrt(mean((b5-G[,5])^2)))
rmseH<-c(sqrt(mean((b1-H[,1])^2)), sqrt(mean((b2-H[,2])^2)), sqrt(mean((b3-H[,3])^2)),
sqrt(mean((b4-H[,4])^2)), sqrt(mean((b5-H[,5])^2)))

median.biasA
madA
mean.biasA
rmseA

median.biasB
madB
mean.biasB
rmseB

median.biasC
madC
mean.biasC
rmseC

median.biasD
madD
mean.biasD
rmseD

median.biasE
```

```
madE
mean.biasE
rmseE

median.biasF
madF
mean.biasF
rmseF

median.biasG
madG
mean.biasG
rmseG

median.biasH
madH
mean.biasH
rmseH
```

A1.2 n= 50 with 30% missingness

```
set.seed(1234687)

j= 1:5
A <- array(0, dim=c(N.iter,5))
B <- array(0, dim=c(N.iter,5))
C <- array(0, dim=c(N.iter,5))
D <- array(0, dim=c(N.iter,5))
E <- array(0, dim=c(N.iter,5))
F <- array(0, dim=c(N.iter,5))
G <- array(0, dim=c(N.iter,5))
```

```
H <- array(0, dim=c(N.iter,5))

for(i in 1:N.iter){

N.iter=1000
n=50                        # vary n
t=2          # vary t and change time in pData
nt=n*t
b1=1
b2=-1
b3=1
b4=1
b5=1
ai=rnorm(n,0,1)
x1=rnorm(nt,0,1)
x2 <- runif(nt,0,1)
x3 <- rnorm(nt,0.5,0.5)
x4 <- rbinom(nt,2,0.65)
hit=rnorm(nt,0,1)
uit=rnorm(nt,0,1)
vit=log(abs(uit/(1+uit)))
x5 <- ifelse(x1+hit>0, 1, 0)
alphai <- sqrt(t)*sum(x1)/n+ai #alphai simulation
ci <- kronecker(alphai ,matrix(1,t,1))#kronecker of alphai
wit=ci+b1*x1+b2*x2+b3*x3+b4*x4+b5*x5
zit= wit+vit
y <- ifelse(zit>0, 1, 0)
pData <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n),
y,zit,ci,x1,x2,x3,x4,x5)
```

 # Randomly Insert A Certain Proportion Of NAs Into A Dataframe #

```
pData2<- cbind(x1,x2,x3,x4,x5)

NAins <-  NAinsert <- function(df, prop = .3){
 n <- nrow(df)
 m <- ncol(df)
 num.to.na <- ceiling(prop*n*m)
 id <- sample(0:(m*n-1), num.to.na, replace = FALSE)
 rows <- id %/% m + 1
 cols <- id %% m + 1
 sapply(seq(num.to.na), function(x){
 df[rows[x], cols[x]] <<- NA
 }
 )
 return(df)
}
##############
# TRY IT OUT #
##############
XXX<-NAins(pData2, .3)
XXX
```

 # Mean Imputation #

```
dat<-sapply(seq_len(ncol(XXX)),function(i) {XXX[,i][is.na(XXX[,i])]<-
mean(XXX[,i],na.rm=TRUE);XXX[,i]})
dat2 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat)
dat2
```

 # Previous value carried forward Imputation #

```
fillNAByPreviousData <- function(column) {
  navals <- which(is.na(column))
    filledvals <- which(! is.na(column))
```

```
    fillup <- sapply(navals, function(x) max(filledvals[filledvals < x]))
    column[navals] <- column[fillup]
    column
}
 dat1=fillNAByPreviousData(XXX)
dat3 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat1)
dat3
```

 # Median Imputation #

```
f=function(x){
   x<-as.numeric(as.character(x))
   x[is.na(x)] =median(x, na.rm=TRUE)
   x
}
dat.1=data.frame(apply(XXX,2,f))
dat4<- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat.1)
dat4

glm.out1 = glm(y ~x1+x2+x3+x4+x5,family=binomial(logit), data=pData)
clogit1=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = pData)
glm.out2 = glm(y ~X1+X2+X3+X4+X5,family=binomial(logit), data=dat2)
clogit2=clogit((y ~ (X1 +X2+X3+X4+X5)),  strata (id), data = dat2)
glm.out3 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat3)
clogit3=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat3)
glm.out4 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat4)
clogit4=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat4)

A[i,] <- glm.out1$coef[j+1]
B[i,] <- clogit1$coef[j]
C[i,] <- glm.out2$coef[j+1]
D[i,] <- clogit2$coef[j]
E[i,] <- glm.out3$coef[j+1]
```

```
F[i,] <- clogit3$coef[j]
G[i,] <- glm.out4$coef[j+1]
H[i,] <- clogit4$coef[j]

}
```

#median absolute deviation#

```
madA<- c(mad(A[,1],constant=1), mad(A[,2], constant=1), mad(A[,3], constant=1), mad(A[,4], constant=1), mad(A[,5], constant=1))
madB<- c(mad(B[,1],constant=1), mad(B[,2], constant=1), mad(B[,3], constant=1), mad(B[,4], constant=1), mad(B[,5], constant=1))
madC<- c(mad(C[,1],constant=1), mad(C[,2], constant=1), mad(C[,3], constant=1), mad(C[,4], constant=1), mad(C[,5], constant=1))
madD<- c(mad(D[,1],constant=1), mad(D[,2], constant=1), mad(D[,3], constant=1), mad(D[,4], constant=1), mad(D[,5], constant=1))
madE<- c(mad(E[,1],constant=1), mad(E[,2], constant=1), mad(E[,3], constant=1), mad(E[,4], constant=1), mad(E[,5], constant=1))
madF<- c(mad(F[,1],constant=1), mad(F[,2], constant=1), mad(F[,3], constant=1), mad(F[,4], constant=1), mad(F[,5], constant=1))
madG<- c(mad(G[,1],constant=1), mad(G[,2], constant=1), mad(G[,3], constant=1), mad(G[,4], constant=1), mad(G[,5], constant=1))
madH<- c(mad(H[,1],constant=1), mad(H[,2], constant=1), mad(H[,3], constant=1), mad(H[,4], constant=1), mad(H[,5], constant=1))
```

#median bias#

```
median.biasA<- c(b1-median(A[,1]), b2-median(A[,2]),b3-median(A[,3]),b4-median(A[,4]),b5-median(A[,5]))
median.biasB<- c(b1-median(B[,1]), b2-median(B[,2]),b3-median(B[,3]),b4-median(B[,4]),b5-median(B[,5]))
median.biasC<- c(b1-median(C[,1]), b2-median(C[,2]),b3-median(C[,3]),b4-median(C[,4]),b5-median(C[,5]))
```

```
median.biasD<- c(b1-median(D[,1]), b2-median(D[,2]),b3-median(D[,3]),b4-median(D[,4]),b5-median(D[,5]))
median.biasE<- c(b1-median(E[,1]), b2-median(E[,2]),b3-median(E[,3]),b4-median(E[,4]),b5-median(E[,5]))
median.biasF<- c(b1-median(F[,1]), b2-median(F[,2]),b3-median(F[,3]),b4-median(F[,4]),b5-median(F[,5]))
median.biasG<- c(b1-median(G[,1]), b2-median(G[,2]),b3-median(G[,3]),b4-median(G[,4]),b5-median(G[,5]))
median.biasH<- c(b1-median(H[,1]), b2-median(H[,2]),b3-median(H[,3]),b4-median(H[,4]),b5-median(H[,5]))
```

#mean bias#

```
mean.biasA<- c(b1-mean(A[,1]), b2-mean(A[,2]),b3-mean(A[,3]),b4-mean(A[,4]),b5-mean(A[,5]))
mean.biasB<- c(b1-mean(B[,1]), b2-mean(B[,2]),b3-mean(B[,3]),b4-mean(B[,4]),b5-mean(B[,5]))
mean.biasC<- c(b1-mean(C[,1]), b2-mean(C[,2]),b3-mean(C[,3]),b4-mean(C[,4]),b5-mean(C[,5]))
mean.biasD<- c(b1-mean(D[,1]), b2-mean(D[,2]),b3-mean(D[,3]),b4-mean(D[,4]),b5-mean(D[,5]))
mean.biasE<- c(b1-mean(E[,1]), b2-mean(E[,2]),b3-mean(E[,3]),b4-mean(E[,4]),b5-mean(E[,5]))
mean.biasF<- c(b1-mean(F[,1]), b2-mean(F[,2]),b3-mean(F[,3]),b4-mean(F[,4]),b5-mean(F[,5]))
mean.biasG<- c(b1-mean(G[,1]), b2-mean(G[,2]),b3-mean(G[,3]),b4-mean(G[,4]),b5-mean(G[,5]))
mean.biasH<- c(b1-mean(H[,1]), b2-mean(H[,2]),b3-mean(H[,3]),b4-mean(H[,4]),b5-mean(H[,5]))
```

#residual errors#

```
rmseA<-c(sqrt(mean((b1-A[,1])^2)), sqrt(mean((b2-A[,2])^2)), sqrt(mean((b3-A[,3])^2)), sqrt(mean((b4-A[,4])^2)), sqrt(mean((b5-A[,5])^2)))
```

```
rmseB<-c(sqrt(mean((b1-B[,1])^2)), sqrt(mean((b2-B[,2])^2)), sqrt(mean((b3-B[,3])^2)),
sqrt(mean((b4-B[,4])^2)), sqrt(mean((b5-B[,5])^2)))
rmseC<-c(sqrt(mean((b1-C[,1])^2)), sqrt(mean((b2-C[,2])^2)), sqrt(mean((b3-C[,3])^2)),
sqrt(mean((b4-C[,4])^2)), sqrt(mean((b5-C[,5])^2)))
rmseD<-c(sqrt(mean((b1-D[,1])^2)), sqrt(mean((b2-D[,2])^2)), sqrt(mean((b3-D[,3])^2)),
sqrt(mean((b4-D[,4])^2)), sqrt(mean((b5-D[,5])^2)))
rmseE<-c(sqrt(mean((b1-E[,1])^2)), sqrt(mean((b2-E[,2])^2)), sqrt(mean((b3-E[,3])^2)),
sqrt(mean((b4-E[,4])^2)), sqrt(mean((b5-E[,5])^2)))
rmseF<-c(sqrt(mean((b1-F[,1])^2)), sqrt(mean((b2-F[,2])^2)), sqrt(mean((b3-F[,3])^2)),
sqrt(mean((b4-F[,4])^2)), sqrt(mean((b5-F[,5])^2)))
rmseG<-c(sqrt(mean((b1-G[,1])^2)), sqrt(mean((b2-G[,2])^2)), sqrt(mean((b3-G[,3])^2)),
sqrt(mean((b4-G[,4])^2)), sqrt(mean((b5-G[,5])^2)))
rmseH<-c(sqrt(mean((b1-H[,1])^2)), sqrt(mean((b2-H[,2])^2)), sqrt(mean((b3-H[,3])^2)),
sqrt(mean((b4-H[,4])^2)), sqrt(mean((b5-H[,5])^2)))

median.biasA
madA
mean.biasA
rmseA

median.biasB
madB
mean.biasB
rmseB

median.biasC
madC
mean.biasC
rmseC

median.biasD
madD
```

```
mean.biasD
rmseD

median.biasE
madE
mean.biasE
rmseE

median.biasF
madF
mean.biasF
rmseF

median.biasG
madG
mean.biasG
rmseG

median.biasH
madH
mean.biasH
rmseH
```

A1.3 n= 100 with 10% missingness

```
set.seed(12346789)

j= 1:5
A <- array(0, dim=c(N.iter,5))
B <- array(0, dim=c(N.iter,5))
C <- array(0, dim=c(N.iter,5))
D <- array(0, dim=c(N.iter,5))
```

```
E <- array(0, dim=c(N.iter,5))
F <- array(0, dim=c(N.iter,5))
G <- array(0, dim=c(N.iter,5))
H <- array(0, dim=c(N.iter,5))

for(i in 1:N.iter){

N.iter=1000
n=100                    # vary n
t=2                      # vary t and change time in pData
nt=n*t
b1=1
b2=-1
b3=1
b4=1
b5=1
ai=rnorm(n,0,1)
x1=rnorm(nt,0,1)
x2 <- runif(nt,0,1)
x3 <- rnorm(nt,0.5,0.5)
x4 <- rbinom(nt,2,0.65)          # nt different such numbers
hit=rnorm(nt,0,1)
uit=rnorm(nt,0,1)
vit=log(abs(uit/(1+uit)))
x5 <- ifelse(x1+hit>0, 1, 0)
alphai <- sqrt(t)*sum(x1)/n+ai #alphai simulation
ci <- kronecker(alphai ,matrix(1,t,1))#kronecker of alphai
wit=ci+b1*x1+b2*x2+b3*x3+b4*x4+b5*x5
zit= wit+vit
y <- ifelse(zit>0, 1, 0)
pData <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n),
y,zit,ci,x1,x2,x3,x4,x5)
```

 # Randomly Insert A Certain Proportion Of NAs Into A Dataframe #

```
pData2<- cbind(x1,x2,x3,x4,x5)

NAins <-  NAinsert <- function(df, prop = .1){
 n <- nrow(df)
 m <- ncol(df)
 num.to.na <- ceiling(prop*n*m)
 id <- sample(0:(m*n-1), num.to.na, replace = FALSE)
 rows <- id %/% m + 1
 cols <- id %% m + 1
 sapply(seq(num.to.na), function(x){
 df[rows[x], cols[x]] <<- NA
 }
 )
 return(df)
}
###############
# TRY IT OUT #
###############
XXX<-NAins(pData2, .1)                                    ##### Inserts NA even in id and
time!!!! #
XXX
```

 # Mean Imputation #

```
dat<-sapply(seq_len(ncol(XXX)),function(i) {XXX[,i][is.na(XXX[,i])]<-
mean(XXX[,i],na.rm=TRUE);XXX[,i]})
dat2 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat)
dat2
```

 # Previous value carried forward Imputation #

```
fillNAByPreviousData <- function(column) {
   navals <- which(is.na(column))
   filledvals <- which(! is.na(column))
    fillup <- sapply(navals, function(x) max(filledvals[filledvals < x]))
   column[navals] <- column[fillup]
   column
}
 dat1=fillNAByPreviousData(XXX)
dat3 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat1)
dat3
```

 # Median Imputation #

```
f=function(x){
  x<-as.numeric(as.character(x))
  x[is.na(x)] =median(x, na.rm=TRUE)
  x
}
dat.1=data.frame(apply(XXX,2,f))
dat4<- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat.1)
dat4

glm.out1 = glm(y ~x1+x2+x3+x4+x5,family=binomial(logit), data=pData)
clogit1=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = pData)
glm.out2 = glm(y ~X1+X2+X3+X4+X5,family=binomial(logit), data=dat2)
clogit2=clogit((y ~ (X1 +X2+X3+X4+X5)),  strata (id), data = dat2)
glm.out3 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat3)
clogit3=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat3)
glm.out4 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat4)
clogit4=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat4)

A[i,] <- glm.out1$coef[j+1]
```

```
B[i,] <- clogit1$coef[j]
C[i,] <- glm.out2$coef[j+1]
D[i,] <- clogit2$coef[j]
E[i,] <- glm.out3$coef[j+1]
F[i,] <- clogit3$coef[j]
G[i,] <- glm.out4$coef[j+1]
H[i,] <- clogit4$coef[j]

}
```

#median absolute deviation#

```
madA<- c(mad(A[,1],constant=1), mad(A[,2], constant=1), mad(A[,3], constant=1), mad(A[,4], constant=1), mad(A[,5], constant=1))
madB<- c(mad(B[,1],constant=1), mad(B[,2], constant=1), mad(B[,3], constant=1), mad(B[,4], constant=1), mad(B[,5], constant=1))
madC<- c(mad(C[,1],constant=1), mad(C[,2], constant=1), mad(C[,3], constant=1), mad(C[,4], constant=1), mad(C[,5], constant=1))
madD<- c(mad(D[,1],constant=1), mad(D[,2], constant=1), mad(D[,3], constant=1), mad(D[,4], constant=1), mad(D[,5], constant=1))
madE<- c(mad(E[,1],constant=1), mad(E[,2], constant=1), mad(E[,3], constant=1), mad(E[,4], constant=1), mad(E[,5], constant=1))
madF<- c(mad(F[,1],constant=1), mad(F[,2], constant=1), mad(F[,3], constant=1), mad(F[,4], constant=1), mad(F[,5], constant=1))
madG<- c(mad(G[,1],constant=1), mad(G[,2], constant=1), mad(G[,3], constant=1), mad(G[,4], constant=1), mad(G[,5], constant=1))
madH<- c(mad(H[,1],constant=1), mad(H[,2], constant=1), mad(H[,3], constant=1), mad(H[,4], constant=1), mad(H[,5], constant=1))
```

#median bias#

```
median.biasA<- c(b1-median(A[,1]), b2-median(A[,2]),b3-median(A[,3]),b4-median(A[,4]),b5-median(A[,5]))
```

```
median.biasB<- c(b1-median(B[,1]), b2-median(B[,2]),b3-median(B[,3]),b4-median(B[,4]),b5-median(B[,5]))
median.biasC<- c(b1-median(C[,1]), b2-median(C[,2]),b3-median(C[,3]),b4-median(C[,4]),b5-median(C[,5]))
median.biasD<- c(b1-median(D[,1]), b2-median(D[,2]),b3-median(D[,3]),b4-median(D[,4]),b5-median(D[,5]))
median.biasE<- c(b1-median(E[,1]), b2-median(E[,2]),b3-median(E[,3]),b4-median(E[,4]),b5-median(E[,5]))
median.biasF<- c(b1-median(F[,1]), b2-median(F[,2]),b3-median(F[,3]),b4-median(F[,4]),b5-median(F[,5]))
median.biasG<- c(b1-median(G[,1]), b2-median(G[,2]),b3-median(G[,3]),b4-median(G[,4]),b5-median(G[,5]))
median.biasH<- c(b1-median(H[,1]), b2-median(H[,2]),b3-median(H[,3]),b4-median(H[,4]),b5-median(H[,5]))
```

#mean bias#

```
mean.biasA<- c(b1-mean(A[,1]), b2-mean(A[,2]),b3-mean(A[,3]),b4-mean(A[,4]),b5-mean(A[,5]))
mean.biasB<- c(b1-mean(B[,1]), b2-mean(B[,2]),b3-mean(B[,3]),b4-mean(B[,4]),b5-mean(B[,5]))
mean.biasC<- c(b1-mean(C[,1]), b2-mean(C[,2]),b3-mean(C[,3]),b4-mean(C[,4]),b5-mean(C[,5]))
mean.biasD<- c(b1-mean(D[,1]), b2-mean(D[,2]),b3-mean(D[,3]),b4-mean(D[,4]),b5-mean(D[,5]))
mean.biasE<- c(b1-mean(E[,1]), b2-mean(E[,2]),b3-mean(E[,3]),b4-mean(E[,4]),b5-mean(E[,5]))
mean.biasF<- c(b1-mean(F[,1]), b2-mean(F[,2]),b3-mean(F[,3]),b4-mean(F[,4]),b5-mean(F[,5]))
mean.biasG<- c(b1-mean(G[,1]), b2-mean(G[,2]),b3-mean(G[,3]),b4-mean(G[,4]),b5-mean(G[,5]))
mean.biasH<- c(b1-mean(H[,1]), b2-mean(H[,2]),b3-mean(H[,3]),b4-mean(H[,4]),b5-mean(H[,5]))
```

#residual errors#

```
rmseA<-c(sqrt(mean((b1-A[,1])^2)), sqrt(mean((b2-A[,2])^2)), sqrt(mean((b3-A[,3])^2)),
sqrt(mean((b4-A[,4])^2)), sqrt(mean((b5-A[,5])^2)))
rmseB<-c(sqrt(mean((b1-B[,1])^2)), sqrt(mean((b2-B[,2])^2)), sqrt(mean((b3-B[,3])^2)),
sqrt(mean((b4-B[,4])^2)), sqrt(mean((b5-B[,5])^2)))
rmseC<-c(sqrt(mean((b1-C[,1])^2)), sqrt(mean((b2-C[,2])^2)), sqrt(mean((b3-C[,3])^2)),
sqrt(mean((b4-C[,4])^2)), sqrt(mean((b5-C[,5])^2)))
rmseD<-c(sqrt(mean((b1-D[,1])^2)), sqrt(mean((b2-D[,2])^2)), sqrt(mean((b3-D[,3])^2)),
sqrt(mean((b4-D[,4])^2)), sqrt(mean((b5-D[,5])^2)))
rmseE<-c(sqrt(mean((b1-E[,1])^2)), sqrt(mean((b2-E[,2])^2)), sqrt(mean((b3-E[,3])^2)),
sqrt(mean((b4-E[,4])^2)), sqrt(mean((b5-E[,5])^2)))
rmseF<-c(sqrt(mean((b1-F[,1])^2)), sqrt(mean((b2-F[,2])^2)), sqrt(mean((b3-F[,3])^2)),
sqrt(mean((b4-F[,4])^2)), sqrt(mean((b5-F[,5])^2)))
rmseG<-c(sqrt(mean((b1-G[,1])^2)), sqrt(mean((b2-G[,2])^2)), sqrt(mean((b3-G[,3])^2)),
sqrt(mean((b4-G[,4])^2)), sqrt(mean((b5-G[,5])^2)))
rmseH<-c(sqrt(mean((b1-H[,1])^2)), sqrt(mean((b2-H[,2])^2)), sqrt(mean((b3-H[,3])^2)),
sqrt(mean((b4-H[,4])^2)), sqrt(mean((b5-H[,5])^2)))

median.biasA
madA
mean.biasA
rmseA

median.biasB
madB
mean.biasB
rmseB

median.biasC
madC
mean.biasC
rmseC
```

```
median.biasD
madD
mean.biasD
rmseD

median.biasE
madE
mean.biasE
rmseE

median.biasF
madF
mean.biasF
rmseF

median.biasG
madG
mean.biasG
rmseG

median.biasH
madH
mean.biasH
rmseH
```

A1.4 n= 100 with 30% missingness

```
set.seed(12346879)

j= 1:5
A <- array(0, dim=c(N.iter,5))
```

```
B <- array(0, dim=c(N.iter,5))
C <- array(0, dim=c(N.iter,5))
D <- array(0, dim=c(N.iter,5))
E <- array(0, dim=c(N.iter,5))
F <- array(0, dim=c(N.iter,5))
G <- array(0, dim=c(N.iter,5))
H <- array(0, dim=c(N.iter,5))

for(i in 1:N.iter){

N.iter=1000
n=100                    # vary n
t=2          # vary t and change time in pData
nt=n*t
b1=1
b2=-1
b3=1
b4=1
b5=1
ai=rnorm(n,0,1)
x1=rnorm(nt,0,1)
x2 <- runif(nt,0,1)
x3 <- rnorm(nt,0.5,0.5)
x4 <- rbinom(nt,2,0.65)
hit=rnorm(nt,0,1)
uit=rnorm(nt,0,1)
vit=log(abs(uit/(1+uit)))
x5 <- ifelse(x1+hit>0, 1, 0)
alphai <- sqrt(t)*sum(x1)/n+ai #alphai simulation
ci <- kronecker(alphai ,matrix(1,t,1))#kronecker of alphai
wit=ci+b1*x1+b2*x2+b3*x3+b4*x4+b5*x5
zit= wit+vit
```

```
y <- ifelse(zit>0, 1, 0)
pData <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n),
y,zit,ci,x1,x2,x3,x4,x5)
```

 # Randomly Insert A Certain Proportion Of NAs Into A Dataframe #

```
pData2<- cbind(x1,x2,x3,x4,x5)

NAins <-  NAinsert <- function(df, prop = .3){
 n <- nrow(df)
 m <- ncol(df)
 num.to.na <- ceiling(prop*n*m)
 id <- sample(0:(m*n-1), num.to.na, replace = FALSE)
 rows <- id %/% m + 1
 cols <- id %% m + 1
 sapply(seq(num.to.na), function(x){
 df[rows[x], cols[x]] <<- NA
 }
 )
 return(df)
}

XXX<-NAins(pData2, .3)                                    ##### Inserts NA even in id and
time!!!! #
XXX
```

 # Mean Imputation #

```
dat<-sapply(seq_len(ncol(XXX)),function(i) {XXX[,i][is.na(XXX[,i])]<-
mean(XXX[,i],na.rm=TRUE);XXX[,i]})
dat2 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat)
dat2
```

 # Previous value carried forward Imputation #

```
fillNAByPreviousData <- function(column) {
   navals <- which(is.na(column))
   filledvals <- which(! is.na(column))
   fillup <- sapply(navals, function(x) max(filledvals[filledvals < x]))
   column[navals] <- column[fillup]
   column
}
 dat1=fillNAByPreviousData(XXX)
dat3 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat1)
dat3
```

 # Median Imputation #

```
f=function(x){
  x<-as.numeric(as.character(x))
  x[is.na(x)] =median(x, na.rm=TRUE)
  x
}
dat.1=data.frame(apply(XXX,2,f))
dat4<- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat.1)
dat4

glm.out1 = glm(y ~x1+x2+x3+x4+x5,family=binomial(logit), data=pData)
clogit1=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = pData)
glm.out2 = glm(y ~X1+X2+X3+X4+X5,family=binomial(logit), data=dat2)
clogit2=clogit((y ~ (X1 +X2+X3+X4+X5)),  strata (id), data = dat2)
glm.out3 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat3)
clogit3=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat3)
glm.out4 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat4)
clogit4=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat4)
```

```
A[i,] <- glm.out1$coef[j+1]
B[i,] <- clogit1$coef[j]
C[i,] <- glm.out2$coef[j+1]
D[i,] <- clogit2$coef[j]
E[i,] <- glm.out3$coef[j+1]
F[i,] <- clogit3$coef[j]
G[i,] <- glm.out4$coef[j+1]
H[i,] <- clogit4$coef[j]

}
```

#median absolute deviation#

```
madA<- c(mad(A[,1],constant=1), mad(A[,2], constant=1), mad(A[,3], constant=1), mad(A[,4], constant=1), mad(A[,5], constant=1))
madB<- c(mad(B[,1],constant=1), mad(B[,2], constant=1), mad(B[,3], constant=1), mad(B[,4], constant=1), mad(B[,5], constant=1))
madC<- c(mad(C[,1],constant=1), mad(C[,2], constant=1), mad(C[,3], constant=1), mad(C[,4], constant=1), mad(C[,5], constant=1))
madD<- c(mad(D[,1],constant=1), mad(D[,2], constant=1), mad(D[,3], constant=1), mad(D[,4], constant=1), mad(D[,5], constant=1))
madE<- c(mad(E[,1],constant=1), mad(E[,2], constant=1), mad(E[,3], constant=1), mad(E[,4], constant=1), mad(E[,5], constant=1))
madF<- c(mad(F[,1],constant=1), mad(F[,2], constant=1), mad(F[,3], constant=1), mad(F[,4], constant=1), mad(F[,5], constant=1))
madG<- c(mad(G[,1],constant=1), mad(G[,2], constant=1), mad(G[,3], constant=1), mad(G[,4], constant=1), mad(G[,5], constant=1))
madH<- c(mad(H[,1],constant=1), mad(H[,2], constant=1), mad(H[,3], constant=1), mad(H[,4], constant=1), mad(H[,5], constant=1))
```

#median bias#

```
median.biasA<- c(b1-median(A[,1]), b2-median(A[,2]),b3-median(A[,3]),b4-median(A[,4]),b5-median(A[,5]))
```

```
median.biasB<- c(b1-median(B[,1]), b2-median(B[,2]),b3-median(B[,3]),b4-median(B[,4]),b5-median(B[,5]))
median.biasC<- c(b1-median(C[,1]), b2-median(C[,2]),b3-median(C[,3]),b4-median(C[,4]),b5-median(C[,5]))
median.biasD<- c(b1-median(D[,1]), b2-median(D[,2]),b3-median(D[,3]),b4-median(D[,4]),b5-median(D[,5]))
median.biasE<- c(b1-median(E[,1]), b2-median(E[,2]),b3-median(E[,3]),b4-median(E[,4]),b5-median(E[,5]))
median.biasF<- c(b1-median(F[,1]), b2-median(F[,2]),b3-median(F[,3]),b4-median(F[,4]),b5-median(F[,5]))
median.biasG<- c(b1-median(G[,1]), b2-median(G[,2]),b3-median(G[,3]),b4-median(G[,4]),b5-median(G[,5]))
median.biasH<- c(b1-median(H[,1]), b2-median(H[,2]),b3-median(H[,3]),b4-median(H[,4]),b5-median(H[,5]))
```

#mean bias#

```
mean.biasA<- c(b1-mean(A[,1]), b2-mean(A[,2]),b3-mean(A[,3]),b4-mean(A[,4]),b5-mean(A[,5]))
mean.biasB<- c(b1-mean(B[,1]), b2-mean(B[,2]),b3-mean(B[,3]),b4-mean(B[,4]),b5-mean(B[,5]))
mean.biasC<- c(b1-mean(C[,1]), b2-mean(C[,2]),b3-mean(C[,3]),b4-mean(C[,4]),b5-mean(C[,5]))
mean.biasD<- c(b1-mean(D[,1]), b2-mean(D[,2]),b3-mean(D[,3]),b4-mean(D[,4]),b5-mean(D[,5]))
mean.biasE<- c(b1-mean(E[,1]), b2-mean(E[,2]),b3-mean(E[,3]),b4-mean(E[,4]),b5-mean(E[,5]))
mean.biasF<- c(b1-mean(F[,1]), b2-mean(F[,2]),b3-mean(F[,3]),b4-mean(F[,4]),b5-mean(F[,5]))
mean.biasG<- c(b1-mean(G[,1]), b2-mean(G[,2]),b3-mean(G[,3]),b4-mean(G[,4]),b5-mean(G[,5]))
mean.biasH<- c(b1-mean(H[,1]), b2-mean(H[,2]),b3-mean(H[,3]),b4-mean(H[,4]),b5-mean(H[,5]))
```

#residual errors#

```
rmseA<-c(sqrt(mean((b1-A[,1])^2)), sqrt(mean((b2-A[,2])^2)), sqrt(mean((b3-A[,3])^2)),
sqrt(mean((b4-A[,4])^2)), sqrt(mean((b5-A[,5])^2)))
rmseB<-c(sqrt(mean((b1-B[,1])^2)), sqrt(mean((b2-B[,2])^2)), sqrt(mean((b3-B[,3])^2)),
sqrt(mean((b4-B[,4])^2)), sqrt(mean((b5-B[,5])^2)))
rmseC<-c(sqrt(mean((b1-C[,1])^2)), sqrt(mean((b2-C[,2])^2)), sqrt(mean((b3-C[,3])^2)),
sqrt(mean((b4-C[,4])^2)), sqrt(mean((b5-C[,5])^2)))
rmseD<-c(sqrt(mean((b1-D[,1])^2)), sqrt(mean((b2-D[,2])^2)), sqrt(mean((b3-D[,3])^2)),
sqrt(mean((b4-D[,4])^2)), sqrt(mean((b5-D[,5])^2)))
rmseE<-c(sqrt(mean((b1-E[,1])^2)), sqrt(mean((b2-E[,2])^2)), sqrt(mean((b3-E[,3])^2)),
sqrt(mean((b4-E[,4])^2)), sqrt(mean((b5-E[,5])^2)))
rmseF<-c(sqrt(mean((b1-F[,1])^2)), sqrt(mean((b2-F[,2])^2)), sqrt(mean((b3-F[,3])^2)),
sqrt(mean((b4-F[,4])^2)), sqrt(mean((b5-F[,5])^2)))
rmseG<-c(sqrt(mean((b1-G[,1])^2)), sqrt(mean((b2-G[,2])^2)), sqrt(mean((b3-G[,3])^2)),
sqrt(mean((b4-G[,4])^2)), sqrt(mean((b5-G[,5])^2)))
rmseH<-c(sqrt(mean((b1-H[,1])^2)), sqrt(mean((b2-H[,2])^2)), sqrt(mean((b3-H[,3])^2)),
sqrt(mean((b4-H[,4])^2)), sqrt(mean((b5-H[,5])^2)))

median.biasA
madA
mean.biasA
rmseA

median.biasB
madB
mean.biasB
rmseB

median.biasC
madC
mean.biasC
rmseC
```

median.biasD
madD
mean.biasD
rmseD

median.biasE
madE
mean.biasE
rmseE

median.biasF
madF
mean.biasF
rmseF

median.biasG
madG
mean.biasG
rmseG

median.biasH
madH
mean.biasH
rmseH

A1.5 n= 250 with 10% missingness

set.seed(1234678)

```
j= 1:5
A <- array(0, dim=c(N.iter,5))
B <- array(0, dim=c(N.iter,5))
C <- array(0, dim=c(N.iter,5))
D <- array(0, dim=c(N.iter,5))
E <- array(0, dim=c(N.iter,5))
F <- array(0, dim=c(N.iter,5))
G <- array(0, dim=c(N.iter,5))
H <- array(0, dim=c(N.iter,5))

for(i in 1:N.iter){

N.iter=1000
n=250                    # vary n
t=2                    # vary t and change time in pData
nt=n*t
b1=1
b2=-1
b3=1
b4=1
b5=1
ai=rnorm(n,0,1)
x1=rnorm(nt,0,1)
x2 <- runif(nt,0,1)
x3 <- rnorm(nt,0.5,0.5)
x4 <- rbinom(nt,2,0.65)          # nt different such numbers
hit=rnorm(nt,0,1)
uit=rnorm(nt,0,1)
vit=log(abs(uit/(1+uit)))
x5 <- ifelse(x1+hit>0, 1, 0)
```

```
alphai <- sqrt(t)*sum(x1)/n+ai #alphai simulation
ci <- kronecker(alphai ,matrix(1,t,1))#kronecker of alphai
wit=ci+b1*x1+b2*x2+b3*x3+b4*x4+b5*x5
zit= wit+vit
y <- ifelse(zit>0, 1, 0)
pData <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n),
y,zit,ci,x1,x2,x3,x4,x5)
```

 # Randomly Insert A Certain Proportion Of NAs Into A Dataframe #

```
pData2<- cbind(x1,x2,x3,x4,x5)

NAins <-  NAinsert <- function(df, prop = .1){
 n <- nrow(df)
 m <- ncol(df)
 num.to.na <- ceiling(prop*n*m)
 id <- sample(0:(m*n-1), num.to.na, replace = FALSE)
 rows <- id %/% m + 1
 cols <- id %% m + 1
 sapply(seq(num.to.na), function(x){
 df[rows[x], cols[x]] <<- NA
 }
 )
 return(df)
}

XXX<-NAins(pData2, .1)
XXX
```

 # Mean Imputation #

```
dat<-sapply(seq_len(ncol(XXX)),function(i) {XXX[,i][is.na(XXX[,i])]<-
mean(XXX[,i],na.rm=TRUE);XXX[,i]})
```

```
dat2 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat)
dat2
```

 # Previous value carried forward Imputation #

```
fillNAByPreviousData <- function(column) {
   navals <- which(is.na(column))
    filledvals <- which(! is.na(column))
    fillup <- sapply(navals, function(x) max(filledvals[filledvals < x]))
      column[navals] <- column[fillup]
   column
}
 dat1=fillNAByPreviousData(XXX)
dat3 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat1)
dat3
```

 # Median Imputation #

```
f=function(x){
  x<-as.numeric(as.character(x))
  x[is.na(x)] =median(x, na.rm=TRUE)
  x }
dat.1=data.frame(apply(XXX,2,f))
dat4<- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat.1)
dat4

glm.out1 = glm(y ~x1+x2+x3+x4+x5,family=binomial(logit), data=pData)
clogit1=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = pData)
glm.out2 = glm(y ~X1+X2+X3+X4+X5,family=binomial(logit), data=dat2)
clogit2=clogit((y ~ (X1 +X2+X3+X4+X5)),  strata (id), data = dat2)
glm.out3 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat3)
clogit3=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat3)
glm.out4 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat4)
clogit4=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat4)
```

```
A[i,] <- glm.out1$coef[j+1]
B[i,] <- clogit1$coef[j]
C[i,] <- glm.out2$coef[j+1]
D[i,] <- clogit2$coef[j]
E[i,] <- glm.out3$coef[j+1]
F[i,] <- clogit3$coef[j]
G[i,] <- glm.out4$coef[j+1]
H[i,] <- clogit4$coef[j]

}
```

#median absolute deviation#

```
madA<- c(mad(A[,1],constant=1), mad(A[,2], constant=1), mad(A[,3], constant=1), mad(A[,4], constant=1), mad(A[,5], constant=1))
madB<- c(mad(B[,1],constant=1), mad(B[,2], constant=1), mad(B[,3], constant=1), mad(B[,4], constant=1), mad(B[,5], constant=1))
madC<- c(mad(C[,1],constant=1), mad(C[,2], constant=1), mad(C[,3], constant=1), mad(C[,4], constant=1), mad(C[,5], constant=1))
madD<- c(mad(D[,1],constant=1), mad(D[,2], constant=1), mad(D[,3], constant=1), mad(D[,4], constant=1), mad(D[,5], constant=1))
madE<- c(mad(E[,1],constant=1), mad(E[,2], constant=1), mad(E[,3], constant=1), mad(E[,4], constant=1), mad(E[,5], constant=1))
madF<- c(mad(F[,1],constant=1), mad(F[,2], constant=1), mad(F[,3], constant=1), mad(F[,4], constant=1), mad(F[,5], constant=1))
madG<- c(mad(G[,1],constant=1), mad(G[,2], constant=1), mad(G[,3], constant=1), mad(G[,4], constant=1), mad(G[,5], constant=1))
madH<- c(mad(H[,1],constant=1), mad(H[,2], constant=1), mad(H[,3], constant=1), mad(H[,4], constant=1), mad(H[,5], constant=1))
```

#median bias#

```
median.biasA<- c(b1-median(A[,1]), b2-median(A[,2]),b3-median(A[,3]),b4-median(A[,4]),b5-median(A[,5]))
median.biasB<- c(b1-median(B[,1]), b2-median(B[,2]),b3-median(B[,3]),b4-median(B[,4]),b5-median(B[,5]))
median.biasC<- c(b1-median(C[,1]), b2-median(C[,2]),b3-median(C[,3]),b4-median(C[,4]),b5-median(C[,5]))
median.biasD<- c(b1-median(D[,1]), b2-median(D[,2]),b3-median(D[,3]),b4-median(D[,4]),b5-median(D[,5]))
median.biasE<- c(b1-median(E[,1]), b2-median(E[,2]),b3-median(E[,3]),b4-median(E[,4]),b5-median(E[,5]))
median.biasF<- c(b1-median(F[,1]), b2-median(F[,2]),b3-median(F[,3]),b4-median(F[,4]),b5-median(F[,5]))
median.biasG<- c(b1-median(G[,1]), b2-median(G[,2]),b3-median(G[,3]),b4-median(G[,4]),b5-median(G[,5]))
median.biasH<- c(b1-median(H[,1]), b2-median(H[,2]),b3-median(H[,3]),b4-median(H[,4]),b5-median(H[,5]))
```

#mean bias#

```
mean.biasA<- c(b1-mean(A[,1]), b2-mean(A[,2]),b3-mean(A[,3]),b4-mean(A[,4]),b5-mean(A[,5]))
mean.biasB<- c(b1-mean(B[,1]), b2-mean(B[,2]),b3-mean(B[,3]),b4-mean(B[,4]),b5-mean(B[,5]))
mean.biasC<- c(b1-mean(C[,1]), b2-mean(C[,2]),b3-mean(C[,3]),b4-mean(C[,4]),b5-mean(C[,5]))
mean.biasD<- c(b1-mean(D[,1]), b2-mean(D[,2]),b3-mean(D[,3]),b4-mean(D[,4]),b5-mean(D[,5]))
mean.biasE<- c(b1-mean(E[,1]), b2-mean(E[,2]),b3-mean(E[,3]),b4-mean(E[,4]),b5-mean(E[,5]))
mean.biasF<- c(b1-mean(F[,1]), b2-mean(F[,2]),b3-mean(F[,3]),b4-mean(F[,4]),b5-mean(F[,5]))
mean.biasG<- c(b1-mean(G[,1]), b2-mean(G[,2]),b3-mean(G[,3]),b4-mean(G[,4]),b5-mean(G[,5]))
```

```
mean.biasH<- c(b1-mean(H[,1]), b2-mean(H[,2]),b3-mean(H[,3]),b4-mean(H[,4]),b5-
mean(H[,5]))
```

#residual errors#

```
rmseA<-c(sqrt(mean((b1-A[,1])^2)), sqrt(mean((b2-A[,2])^2)), sqrt(mean((b3-A[,3])^2)),
sqrt(mean((b4-A[,4])^2)), sqrt(mean((b5-A[,5])^2)))
rmseB<-c(sqrt(mean((b1-B[,1])^2)), sqrt(mean((b2-B[,2])^2)), sqrt(mean((b3-B[,3])^2)),
sqrt(mean((b4-B[,4])^2)), sqrt(mean((b5-B[,5])^2)))
rmseC<-c(sqrt(mean((b1-C[,1])^2)), sqrt(mean((b2-C[,2])^2)), sqrt(mean((b3-C[,3])^2)),
sqrt(mean((b4-C[,4])^2)), sqrt(mean((b5-C[,5])^2)))
rmseD<-c(sqrt(mean((b1-D[,1])^2)), sqrt(mean((b2-D[,2])^2)), sqrt(mean((b3-D[,3])^2)),
sqrt(mean((b4-D[,4])^2)), sqrt(mean((b5-D[,5])^2)))
rmseE<-c(sqrt(mean((b1-E[,1])^2)), sqrt(mean((b2-E[,2])^2)), sqrt(mean((b3-E[,3])^2)),
sqrt(mean((b4-E[,4])^2)), sqrt(mean((b5-E[,5])^2)))
rmseF<-c(sqrt(mean((b1-F[,1])^2)), sqrt(mean((b2-F[,2])^2)), sqrt(mean((b3-F[,3])^2)),
sqrt(mean((b4-F[,4])^2)), sqrt(mean((b5-F[,5])^2)))
rmseG<-c(sqrt(mean((b1-G[,1])^2)), sqrt(mean((b2-G[,2])^2)), sqrt(mean((b3-G[,3])^2)),
sqrt(mean((b4-G[,4])^2)), sqrt(mean((b5-G[,5])^2)))
rmseH<-c(sqrt(mean((b1-H[,1])^2)), sqrt(mean((b2-H[,2])^2)), sqrt(mean((b3-H[,3])^2)),
sqrt(mean((b4-H[,4])^2)), sqrt(mean((b5-H[,5])^2)))
```

```
median.biasA
madA
mean.biasA
rmseA

median.biasB
madB
mean.biasB
rmseB

median.biasC
```

madC
mean.biasC
rmseC

median.biasD
madD
mean.biasD
rmseD

median.biasE
madE
mean.biasE
rmseE

median.biasF
madF
mean.biasF
rmseF

median.biasG
madG
mean.biasG
rmseG

median.biasH
madH
mean.biasH
rmseH

A1.6 n= 250 with 30% missingness

set.seed(123456879)

```
j= 1:5
A <- array(0, dim=c(N.iter,5))
B <- array(0, dim=c(N.iter,5))
C <- array(0, dim=c(N.iter,5))
D <- array(0, dim=c(N.iter,5))
E <- array(0, dim=c(N.iter,5))
F <- array(0, dim=c(N.iter,5))
G <- array(0, dim=c(N.iter,5))
H <- array(0, dim=c(N.iter,5))

for(i in 1:N.iter){

N.iter=1000
n=250                    # vary n
t=2                    # vary t and change time in pData
nt=n*t
b1=1
b2=-1
b3=1
b4=1
b5=1
ai=rnorm(n,0,1)
x1=rnorm(nt,0,1)
x2 <- runif(nt,0,1)
x3 <- rnorm(nt,0.5,0.5)
x4 <- rbinom(nt,2,0.65)          # nt different such numbers
hit=rnorm(nt,0,1)
uit=rnorm(nt,0,1)
vit=log(abs(uit/(1+uit)))
x5 <- ifelse(x1+hit>0, 1, 0)
```

```
alphai <- sqrt(t)*sum(x1)/n+ai #alphai simulation
ci <- kronecker(alphai ,matrix(1,t,1))#kronecker of alphai
wit=ci+b1*x1+b2*x2+b3*x3+b4*x4+b5*x5
zit= wit+vit
y <- ifelse(zit>0, 1, 0)
pData <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n),
y,zit,ci,x1,x2,x3,x4,x5)
```

 # Randomly Insert A Certain Proportion Of NAs Into A Dataframe #

```
pData2<- cbind(x1,x2,x3,x4,x5)

NAins <-  NAinsert <- function(df, prop = .3){
 n <- nrow(df)
 m <- ncol(df)
 num.to.na <- ceiling(prop*n*m)
 id <- sample(0:(m*n-1), num.to.na, replace = FALSE)
 rows <- id %/% m + 1
 cols <- id %% m + 1
 sapply(seq(num.to.na), function(x){
 df[rows[x], cols[x]] <<- NA
 }
 )
 return(df)
}

XXX<-NAins(pData2, .3)
XXX
```

 # Mean Imputation #

```
dat<-sapply(seq_len(ncol(XXX)),function(i) {XXX[,i][is.na(XXX[,i])]<-
mean(XXX[,i],na.rm=TRUE);XXX[,i]})
```

```
dat2 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat)
dat2
# Previous value carried forward Imputation #
fillNAByPreviousData <- function(column) {
   navals <- which(is.na(column))
    filledvals <- which(! is.na(column))
   fillup <- sapply(navals, function(x) max(filledvals[filledvals < x]))
   column[navals] <- column[fillup]
   column
}
 dat1=fillNAByPreviousData(XXX)
dat3 <- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat1)
dat3

# Median Imputation #
f=function(x){
  x<-as.numeric(as.character(x))
  x[is.na(x)] =median(x, na.rm=TRUE)
  x
}
dat.1=data.frame(apply(XXX,2,f))
dat4<- data.frame(id = rep(paste("stdnt", 1:n, sep = "_"), each = t),time = rep(1:2, n), y,ci,dat.1)
dat4

glm.out1 = glm(y ~x1+x2+x3+x4+x5,family=binomial(logit), data=pData)
clogit1=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = pData)
glm.out2 = glm(y ~X1+X2+X3+X4+X5,family=binomial(logit), data=dat2)
clogit2=clogit((y ~ (X1 +X2+X3+X4+X5)),  strata (id), data = dat2)
glm.out3 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat3)
clogit3=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat3)
glm.out4 = glm(y ~ x1+x2+x3+x4+x5,family=binomial(logit), data=dat4)
clogit4=clogit((y ~ (x1 +x2+x3+x4+x5)),  strata (id), data = dat4)
```

```
A[i,] <- glm.out1$coef[j+1]
B[i,] <- clogit1$coef[j]
C[i,] <- glm.out2$coef[j+1]
D[i,] <- clogit2$coef[j]
E[i,] <- glm.out3$coef[j+1]
F[i,] <- clogit3$coef[j]
G[i,] <- glm.out4$coef[j+1]
H[i,] <- clogit4$coef[j]

}
```

#median absolute deviation#

```
madA<- c(mad(A[,1],constant=1), mad(A[,2], constant=1), mad(A[,3], constant=1), mad(A[,4], constant=1), mad(A[,5], constant=1))
madB<- c(mad(B[,1],constant=1), mad(B[,2], constant=1), mad(B[,3], constant=1), mad(B[,4], constant=1), mad(B[,5], constant=1))
madC<- c(mad(C[,1],constant=1), mad(C[,2], constant=1), mad(C[,3], constant=1), mad(C[,4], constant=1), mad(C[,5], constant=1))
madD<- c(mad(D[,1],constant=1), mad(D[,2], constant=1), mad(D[,3], constant=1), mad(D[,4], constant=1), mad(D[,5], constant=1))
madE<- c(mad(E[,1],constant=1), mad(E[,2], constant=1), mad(E[,3], constant=1), mad(E[,4], constant=1), mad(E[,5], constant=1))
madF<- c(mad(F[,1],constant=1), mad(F[,2], constant=1), mad(F[,3], constant=1), mad(F[,4], constant=1), mad(F[,5], constant=1))
madG<- c(mad(G[,1],constant=1), mad(G[,2], constant=1), mad(G[,3], constant=1), mad(G[,4], constant=1), mad(G[,5], constant=1))
madH<- c(mad(H[,1],constant=1), mad(H[,2], constant=1), mad(H[,3], constant=1), mad(H[,4], constant=1), mad(H[,5], constant=1))
```

#median bias#

```
median.biasA<- c(b1-median(A[,1]), b2-median(A[,2]),b3-median(A[,3]),b4-median(A[,4]),b5-median(A[,5]))
median.biasB<- c(b1-median(B[,1]), b2-median(B[,2]),b3-median(B[,3]),b4-median(B[,4]),b5-median(B[,5]))
median.biasC<- c(b1-median(C[,1]), b2-median(C[,2]),b3-median(C[,3]),b4-median(C[,4]),b5-median(C[,5]))
median.biasD<- c(b1-median(D[,1]), b2-median(D[,2]),b3-median(D[,3]),b4-median(D[,4]),b5-median(D[,5]))
median.biasE<- c(b1-median(E[,1]), b2-median(E[,2]),b3-median(E[,3]),b4-median(E[,4]),b5-median(E[,5]))
median.biasF<- c(b1-median(F[,1]), b2-median(F[,2]),b3-median(F[,3]),b4-median(F[,4]),b5-median(F[,5]))
median.biasG<- c(b1-median(G[,1]), b2-median(G[,2]),b3-median(G[,3]),b4-median(G[,4]),b5-median(G[,5]))
median.biasH<- c(b1-median(H[,1]), b2-median(H[,2]),b3-median(H[,3]),b4-median(H[,4]),b5-median(H[,5]))
```

#mean bias#

```
mean.biasA<- c(b1-mean(A[,1]), b2-mean(A[,2]),b3-mean(A[,3]),b4-mean(A[,4]),b5-mean(A[,5]))
mean.biasB<- c(b1-mean(B[,1]), b2-mean(B[,2]),b3-mean(B[,3]),b4-mean(B[,4]),b5-mean(B[,5]))
mean.biasC<- c(b1-mean(C[,1]), b2-mean(C[,2]),b3-mean(C[,3]),b4-mean(C[,4]),b5-mean(C[,5]))
mean.biasD<- c(b1-mean(D[,1]), b2-mean(D[,2]),b3-mean(D[,3]),b4-mean(D[,4]),b5-mean(D[,5]))
mean.biasE<- c(b1-mean(E[,1]), b2-mean(E[,2]),b3-mean(E[,3]),b4-mean(E[,4]),b5-mean(E[,5]))
mean.biasF<- c(b1-mean(F[,1]), b2-mean(F[,2]),b3-mean(F[,3]),b4-mean(F[,4]),b5-mean(F[,5]))
mean.biasG<- c(b1-mean(G[,1]), b2-mean(G[,2]),b3-mean(G[,3]),b4-mean(G[,4]),b5-mean(G[,5]))
```

```
mean.biasH<- c(b1-mean(H[,1]), b2-mean(H[,2]),b3-mean(H[,3]),b4-mean(H[,4]),b5-
mean(H[,5]))

#residual errors#
rmseA<-c(sqrt(mean((b1-A[,1])^2)), sqrt(mean((b2-A[,2])^2)), sqrt(mean((b3-A[,3])^2)),
sqrt(mean((b4-A[,4])^2)), sqrt(mean((b5-A[,5])^2)))
rmseB<-c(sqrt(mean((b1-B[,1])^2)), sqrt(mean((b2-B[,2])^2)), sqrt(mean((b3-B[,3])^2)),
sqrt(mean((b4-B[,4])^2)), sqrt(mean((b5-B[,5])^2)))
rmseC<-c(sqrt(mean((b1-C[,1])^2)), sqrt(mean((b2-C[,2])^2)), sqrt(mean((b3-C[,3])^2)),
sqrt(mean((b4-C[,4])^2)), sqrt(mean((b5-C[,5])^2)))
rmseD<-c(sqrt(mean((b1-D[,1])^2)), sqrt(mean((b2-D[,2])^2)), sqrt(mean((b3-D[,3])^2)),
sqrt(mean((b4-D[,4])^2)), sqrt(mean((b5-D[,5])^2)))
rmseE<-c(sqrt(mean((b1-E[,1])^2)), sqrt(mean((b2-E[,2])^2)), sqrt(mean((b3-E[,3])^2)),
sqrt(mean((b4-E[,4])^2)), sqrt(mean((b5-E[,5])^2)))
rmseF<-c(sqrt(mean((b1-F[,1])^2)), sqrt(mean((b2-F[,2])^2)), sqrt(mean((b3-F[,3])^2)),
sqrt(mean((b4-F[,4])^2)), sqrt(mean((b5-F[,5])^2)))
rmseG<-c(sqrt(mean((b1-G[,1])^2)), sqrt(mean((b2-G[,2])^2)), sqrt(mean((b3-G[,3])^2)),
sqrt(mean((b4-G[,4])^2)), sqrt(mean((b5-G[,5])^2)))
rmseH<-c(sqrt(mean((b1-H[,1])^2)), sqrt(mean((b2-H[,2])^2)), sqrt(mean((b3-H[,3])^2)),
sqrt(mean((b4-H[,4])^2)), sqrt(mean((b5-H[,5])^2)))

median.biasA
madA
mean.biasA
rmseA

median.biasB
madB
mean.biasB
rmseB

median.biasC
```

madC
mean.biasC
rmseC

median.biasD
madD
mean.biasD
rmseD

median.biasE
madE
mean.biasE
rmseE

median.biasF
madF
mean.biasF
rmseF

median.biasG
madG
mean.biasG
rmseG

median.biasH
madH
mean.biasH
rmseH