Estimation of Finite Population Total in the Face of Missing Values Using Model Calibration and Model Assistance on Semiparametric and Nonparametric Models

Pius Nderitu Kihara

A thesis submitted in fulfilment for the degree of Doctor of Philosophy in Statistics in the Jomo Kenyatta University of Agriculture and Technology

2012

DECLARATION

This thesis is my original work and has not been presented for a degree in any other university

Signature.. Date.........................

Pius Nderitu Kihara

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

Signature.. Date.........................

Prof. Romanus Odhiambo

JKUAT, Kenya

Signature.. Date.........................

Dr. John Kihoro

JKUAT, Kenya

DEDICATION

To my wife Mary Wangari and daughter Pauline.

ACKNOWLEDGEMENTS

I thank the everlasting God for granting me enough grace to carry out my studies without whose will I could not have achieved anything. I highly appreciate the company of various people who made this less stressful, interesting and successful. I thank my two supervisors , Prof. R.O. Odhiambo and Dr. J.M. Kihoro and a friend Dr. G. Orwa for their presence and inspiration during the study. I also thank them for their moral and academic support.

Special thanks to my family, to whom I have dedicated this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

SYMBOLS

Φ_s..........chi square distance measure

π_i...........inclusion probability

d_i...........inverse of inclusion probability

ω_i...........design weights

I_i............indicator variable

ξ.............regression model

E_ξ...........model expectation

E_p...........design expectation

y..............survey variable

x..............indepedent variable

Z.............indepedent categorical vector

$\mathbf{X}$.............model matrix

t_i..............ith cluster total

X_t............population total of the indepedent variable x

Y_t.............population total of the depedent variable y

$\widehat{X}$.............horvitz thonpson estimator of X_t

$\widehat{Y}$..............horvitz thonpson estimator of Y_t

N.............population size

n.............sample size

S..............sample

T_n............sequence of estimators

R.............number of samples

ρ..............number of populations

M............size of the population of clusters

m.............number of clusters included in the sample

ϵ_i............ error term assumed i.i.d

ABBREVIATIONS

R_B......... Relative Bias

R_E......... Relative Efficiency

MSE..... Mean Squared Error

i.i.d........ identically and indepedently distributed

HT........ Horvitz Thompson

ABSTRACT

Estimation of finite population total using model calibration and model assistance on semiparametric and nonparametric models and in the presence of auxilliary information is considered. In particular, a class of estimators based on penalized splines are proposed for one stage and two stage sampling. Firstly, estimation of finite population total using internal calibration, model calibration and model assistance on nonparametric models based on kernel methods have been considered by several authors. We have considered such model calibration and model assistance estimation based on penalized splines and extended the estimation to two stage sampling. Secondly, estimation of finite population total using internal calibration and model assistance on semiparametric models based on kernel methods have also been considered by several authors. In this thesis, we have extended this to consinder model calibration, based the estimation on penalized splines and extended the estimation to two stage sampling consindering two scenarios. In the first scenario, the auxilliary information is only available at the cluster level and in the second scenario, the auxilliary information is available both at the element level and at the cluster level. We have shown that the proposed estimators are robust in the face of misspecified models, are asymptotic design unbiased, have reduced model bias, are consistent and asymptotic normal. We have shown that estimators based on penalized splines perform better than corresponding kernel based estimators while model calibrated estimators perform better than internally calibrated estimators. We also recommend some areas for further research.

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background Information

Use of auxiliary information in estimation of missing values and descriptive parameters of a survey variable in a finite population has become fairly common. Census data, administrative registers and previous surveys provide a wide and growing range of variables eligible to be employed to increase the precision of estimation procedures. A simple way to incorporate known population totals of auxiliary variables is through ratio and regression estimation. More general situations are handled by means of generalized regression estimation (Sarndal,1980) and calibration estimation (Deville and Sarndal, 1992). These methods have been proposed within a model assisted approach to inference for the finite population. The processes of estimation of population total and mean starts first with the point estimation of the missing values based on auxiliary variable. Then tecniques like calibration and model assistance are employed on the values to estimate population parameters and or any other required analysis of the data are carried out. There are various methods of handling missing data. Roughly, they can be classified into four groups; complete case analysis, imputation based methods, weighting methods and fully model based procedures.

In complete case analysis, if some variables are not observed for some of the units, these units are omitted from the analysis. The complete cases are then analyzed as they are. This method can lead to serious biases and inefficiency. (Little and Rubin, 1987).

The concept of multiple imputations refers to replacing each missing value with more than one imputed value. The goal is to combine the simplicity of imputation strategies with unbiasedness.(Rubin and Schenker,1986, Little and Rubin,1987). A problem with simple imputation procedure is that this may yield inconsistent point estimates if the data is not missing at random, that is, if the missingness depend on either the unobserved or observed value. Another problem is that the variability of the estimators is underestimated since imputed values are treated as observed values, (Tanner and Wong, 1987).

The fully model based procedures rely on modeling the missing data using estimation methods such as maximum likelihood. They are based on model assumptions which are in most cases untestable, hence sensitivity analysis should be part of the analysis,(Scharfstein et al,2003). Use of semiparametric and nonparametric techniques help relax upon assumptions made.

The weighting methods approach is based on the complete cases where they are weighted with the inverse of the inclusion probability, (Flanders and Greenland, 1991, Zhao and Lipsitz, 1992). Cases with low inclusion probability gain more influence in the analysis thus representing the probable missing values in the neighborhood. In most cases, the inclusion probability is unknown. It can be estimated using nonparametric techniques like kernel based density estimation, (Carpenter and Kenward, 2005).

The reasoning towards use of nonparametric and semiparametric modeling techniques for the missing values include the following. First, an initial nonparametric estimate may well suggest a suitable parametric model such as linear regression. That is, it may give the data more of a chance to speak for themselves in choosing the model to be fitted (Silverman, 1985). Secondly, known facts suggest a tentative model which in turn suggest a particular examination and analysis of data or the need to acquire further data or suggest a modified model resulting in an iterative procedure (Box, 1980, Hastie and Tibshirani, 1987, Simonoff, 1996). It is very important to note that parametric models would be very efficient if the model is correcly specified. However, if the assumed model is mis-specified, inferences can lead to misleading interpretations of data.

Considered is a super population regression model which is denoted by ξ given as

$$y_i = \mu(x_i) + \epsilon \tag{1.1}$$

where $\mu(x_i)$ is a smooth function. Given n pair of observations $(x_i, y_i), \ldots (x_n, y_n)$ from a population of size N, of interest is the estimator $\hat{\mu}(x)$ of $\mu(x) = E_\xi(y/x)$. A nonparametric method like local polynomial or splines could be used for this estimation.

In some circumstances, the auxilliiary information is such that it contains a component whose parametric structure is known and a component that need to enter the estimation nonparametrically. Consider a case where auxiliary information consists of a single univariate term x that is to enter estimation nonparamtrically and a vector Z composed of an arbitrary number of linear terms . Consider super population regression model, ξ given by

$$\begin{aligned} E_\xi(y_i) &= g(x_i, Z_i) \\ &= \mu(x_i) + Z_i\beta \end{aligned} \tag{1.2}$$

where Z_i is a vector of the categorical or continuous auxiliary variables and $\mu(x_i)$ is a nonparametric component. The interest is to find an estimator $\hat{g}(x_i, Z_i)$ of $g(x_i, Z_i)$. This is semiparametric estimation. Breidt et al (2007) uses a sample estimate of the form

$$\hat{g}_i = \hat{\mu}(x_i) + Z_i\hat{\beta} \tag{1.3}$$

Once missing data has been modelled, it is now possible to use it to estimate population total using several techniques. One such technique is calibration. Suppose $U = \{1, 2, \ldots, N\}$is the set of labels for the finite population. Let (y_i, x_i)be the respective values of the study variable y and the vector of auxiliary variables x attached to i^{th} unit. The question is how to estimate population total $Y_t = \sum_{i=1}^{N} y_i$ effectively using the known population totals $X_t = \sum_{i=1}^{N} x_i$ at the estimation stage. If we let $s = \{1, 2, \ldots, n\}$ be the set of sampled units under a general sampling design p, and let $\pi_i = p(i \in s)$ be the first order inclusion probabilities, then the conventional calibration estimator for total Y_t is defined by $\hat{Y} = \sum_{i \in s} w_i y_i$ where $w_i' s$ are design weights such that for a given metric, are as close as possible in an average sense to the $d_i = \frac{1}{\pi_i}$ and are obtained by minimizing a given distance measure between the $w_i' s$ and $d_i' s$ subject to constraints

$$\begin{aligned} \sum_{i \in s} w_i x_i &= \sum_{i=1}^{N} x_i \\ &= X_t \end{aligned} \tag{1.4}$$

This type of calibration is called internal calibration. Consider models for the super population ξ, such that $E_\xi(y_i) = \mu(x_i)$, where $\mu(x_i)$ is a known function of x_i. The model calibration estimator for population total Y_t is $\widetilde{Y} = \sum_{i \in s} w_i y_i$ with weights sought to minimize a given distance measure subject to new constraints

$$\begin{aligned} \sum_{i \in s} w_i &= N, \\ \sum_{i \in s} w_i \hat{\mu}_i &= \sum_{i=1}^{N} \hat{\mu}_i \end{aligned} \tag{1.5}$$

where $\hat{\mu}_i = \hat{\mu}(x_i)$. In this context, calibration is performed with respect to the population mean of the fitted values $\hat{\mu}_i$ (model calibration).

Another technique is model assistance. It is important to improve the precision of estimators while still relying on the sampling design as the primary probability generating mechanism. Model assistance is intended to provide good efficiency if the model is correctly specified, but maintain desirable properties like design consistency if the model is mis-specified.

1.2 Statement of the problem

Of interest is the estimation of population total in the face of some unobserved values but in the presence of auxilliary information. A choice is to be made on the approach of modelling these missing values. As noted earlier, Parametric models have been used and though very efficient when the model is correctly specified, but fail terribly when the model is mis-specified. But in most survey problems, the parametric structure of the population is unknown hence the need to find a more robust approach of modelling the missing values. Nonparametric methods may be used but again a choice among the various nonparametric methods such as local polynomial and spline methods has to be made. The auxilliary information may however contain some part which needs to enter the model parametrically, for example some categorical data, and another part to enter the model nonparametrically in which case semiparametric modelling need to be used.

Internal calibration, model calibration and model assistance have been applied to kernel based nonparametric methods in one stage and two stage sampling. For two stage sampling, approach has been to use these techniques at the estimation of population total while using design estimation for cluster totals ignoring presence of auxilliary information at element level. It would be important to employ model calibration and model assistance to penalzed splines, extend to two stage sampling and study the performace as compared to when kernel methods are used. For two stage sampling, application of the above techniques even at the estimation of cluster totals to take advantage of auxilliary information within clusters should be considered.

Again,internal calibration and model assistance have been used on semiparametric models based on kernel methods in one stage sampling. Model calibration has not been employed on semiparametric models. It would be neccesary to apply model calibration and model assistance on semiparametric models based on penalized splines, extend to two stage sampling and again study the performance compared to when kernel methods are used. Consideration of the auxilliary information available at element level is also important.

A comparison between model calibrated and internally calibrated estimators is necessary to find out which one uses the auxilliary information more efficiently and in which circumstances. This is important in order to protect against blind calibration.

1.3 Objectives of the study

1. Derive an estimator of finite population total using model calibration and assistance on nonparametric models and using penalized splines
2. Derive an estimator of finite population total using model calibration and assistance on semiparametric models and using penalized splines
3. Compare the performance of kernel methods and penalized splines when employed to model calibration on semiparametric and nonparametric models
4. Derive a model calibrated and assisted estimator for two stage sampling
5. Carry out sensitivity analysis on the semiparametric estimation

1.4 Significance of the study

Most oftnely, the population structure of the variable of interest is not known. By using nonparametric methods, we ensure that we have a reliable estimator even if we do not know the population structure of the variables. We derive an estimator that incoporates model calibration so that once weights have been built for a variable depedent on a given set of auxiliary variables, the weights can be used for any other variable that is depedent on the same set of auxiliary variables. The control totals for the auxiliary information are normaly available. We derive an estimator such that if the weights derived are applied to a sample of the auxiliary information, we reproduce the control totals. This is reassuring to the user.By using a design procedure as the primary sample generating mechanism, we ensure that the estimator is robust and does not fail.

In some cases, the population structure for some of the auxiliary variables is known. For example, some may be categorical variables which would imply they are better used in a parametric model. We derive a semiparametric estimator that incoporates nonparametric part for the variables whose structure is unkown and a parametric component.

The derived estimators are general and any nonparametric method may be used. The study includes a comparison of the commonly used kernel methods and penalized splines. This is significant to the user to enable him choose the best. Penalized splines make it easy to incorporate multiple covariates as well as combination of categorical variables. Also makes computation of estimators for data sets with regions of sparse data easier.

We now describe how the thesis is organized.

1.5 Outline of The Thesis

The rest of the thesis is organized as follows. In section(2.1) we review calibration and model assistance techniques in one stage sampling in relation to our first objective. In section(2.2) we review the nonparametric techniques that we have considered in this study and whose performance we compare in line with objective two. Section (2.3) is a review of two stage sampling in view of objective three. In section(3.1) we derive estimators for one stage sampling and whose asymptotic properties we derive in section (3.2) while in section (3.3), we derive estimators under two stage sampling and their asymptotic properties in section (3.4). Section (4.1) is a study of the empirical properties under one stage sampling and in section (4.2) a study of the empirical properties under two stage sampling is carried out.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Calibration and Model Assistance in Estimation of Population Total

We now describe some techniques used to estimate population total where some values of interest have not been observed but imputed. These techniques include internal calibration, model calibration and model assistance.

2.1.1 Internal Calibration

The notion of calibration was introduced by Deville and Sarndal (1992) in the context of using auxiliary information from survey data. Suppose $U = \{1, 2, \ldots, N\}$is the set of labels for the finite population. Let (y_i, x_i)be the respective values of the study variable y and the auxiliary variable x attached to i^{th} unit. Of interest is the estimation of population total $Y_t = \sum_{i=1}^{N} y_i$ effectively using the known population totals $X_t = \sum_{i=1}^{N} x_i$ at the estimation stage. If we let $s = \{1, 2, \ldots, n\}$ be the set of sampled units under a general sampling design p, and let $\pi_i = p(i \in s)$ be the first order inclusion probabilities, then the conventional calibration estimator for total Y_t is defined by $\hat{Y} = \sum_{i \in s} w_i y_i$ where $w_i' s$ are design weights such that for a given metric, are as close as possible in an average sense to the $d_i = \frac{1}{\pi_i}$ and are obtained by minimizing a given distance measure between the $w_i' s$ and $d_i' s$ subject to constraint(1.4)

The most commonly used distance measure is the chi-squared distance

$$\Phi_s = \sum_{i \in s} \frac{(w_i - d_i)^2}{q_i d_i} \tag{2.1}$$

where $q_i' s$ are known positive constants uncorrelated with the $d_i' s$. (Deville and Sarndal,1992). Minimizing this chi-squared distance subject to equation (1.4), they obtained the Langrange equation

$$L = \sum_{i \in s} \frac{(w_i - d_i)^2}{q_i d_i} - 2\lambda \left(\sum_{i=1}^{N} w_i x_i - \sum_{i=1}^{N} x_i) \right) \tag{2.2}$$

where λ is the langrange constant to be determined. Differentiating L with respect to w_i ,equating to zero and solving for w_i then

$$w_i = d_i + \left\{ \frac{d_i q_i x_i}{\sum_{i=1}^{n} d_i q_i x_i^2} \right\} \left\{ \sum_{i=1}^{N} x_i - \sum_{i=1}^{n} d_i x_i \right\} \tag{2.3}$$

Particular choices for q_i yield different forms of the estimator in $\sum_{i \in s} w_i y_i$. Substituting this weight w_i in $\sum_{i \in s} w_i y_i$, they derived the generalized regression estimator of the population total given by

$$YTC = \sum_{i=1}^{n} d_i y_i + \left\{ \frac{\sum_{i=1}^{n} d_i q_i x_i y_i}{\sum_{i=1}^{n} d_i q_i x_i^2} \right\} \left\{ \sum_{i=1}^{N} x_i - \sum_{i=1}^{n} d_i x_i \right\} \tag{2.4}$$

see Deville and Sarndal (1992).

This can be written as

$$YTC = \widehat{Y} + (X_t - \widehat{X})' \widehat{B} \tag{2.5}$$

where the regressions coefficient $\widehat{B} = \left\{ \frac{\sum_{i=1}^{n} d_i q_i x_i y_i}{\sum_{i=1}^{n} d_i q_i x_i^2} \right\}$ while $\widehat{X}$ and $\widehat{Y}$ are Horvitz-Thompson estimators of X_t and Y_t respectively. The approximate variance derived by Deville and Sarndal (1992) is

$$v(YTC) = \frac{1}{2} \sum_{i \neq j} \sum_{\in U} (\pi_i \pi_j - \pi_{ij}) (d_i \epsilon_i - d_j \epsilon_j)^2 \tag{2.6}$$

where $\epsilon_i = y_i - \widehat{B} x_i$. They derived an estimator for the variance as

$$\hat{v}(YTC) = \frac{1}{2} \sum_{i \neq j} \sum_{\in s} \frac{\pi_i \pi_j - \pi_{ij}}{\pi_{ij} (w_i \epsilon_i - w_j \epsilon_j)^2} \tag{2.7}$$

The population estimator (2.5) is quite general and includes some well known estimators as particular cases. If $q_i = \frac{1}{x_i}$ then the estimator (2.5) reduces to the ratio estimator studied by Cochran (1997)

$$YTC = \widehat{Y}\left(\frac{X_t}{\widehat{X}}\right) \tag{2.8}$$

If $q_i = 1$, then estimator (2.5) reduces to general regression estimator

$$YTC = \widehat{Y} + \left(X_t - \widehat{X}\right) \tag{2.9}$$

The definition of YTC is equivalent to a generalized regression estimator, which is derived as a model assisted estimator assuming a linear regression model with variance structure provided by the diagonal matrix with elements $\frac{1}{q_i}$ (Deville and sarndal, 1992, section 1). Hence YTC implicitly relies on a linear relationship between the auxiliary variables and the survey variable.

Although alternative distance measures have also been considered, (Deville and Sarndal ,1992), all resulting estimators are asymptotically equivalent to the one obtained from minimizing the chi-squared distance measure(2.1).

Wu and Sitter (2001) added the constraint $\sum_{i\in s} w_i = N$ on the weights and developed a new estimator of the population total by minimizing (2.1) subject to the constraints $\sum_{i\in s} w_i = N$,$\sum_{i\in s} w_i x_i = X_t$. They introduced the Lagrange equation

$$L = \sum_{i\in s} \frac{(w_i - d_i)^2}{q_i d_i} - 2\lambda\left(\sum_{i=1}^{N} w_i x_i - \sum_{i=1}^{N} x_i\right) - 2V\left(\sum_{i\in s} w_i - N\right) \tag{2.10}$$

whereλ and V are the langrange constants to be determined. Differentiating (2.10) with respect to w_i, equating the derivative to zero and solving for w_i they obtained

$$w_i = (\lambda x_i + V)q_i d_i + d_i \tag{2.11}$$

and when substituted in $\sum_{i\in s} w_i y_i$ yields the estimator given below.

$$\begin{aligned} YTC &= \sum_{i=1}^{n} d_i y_i \\ &+ \left(N - \sum_{i=1}^{n} d_i\right) \left\{ \frac{\sum_{i=1}^{n} d_i q_i y_i}{\sum_{i=1}^{n} d_i q_i} - \frac{\sum_{i=1}^{n} d_i q_i \left[x_i - \frac{\sum_{i=1}^{n} d_i q_i x_i}{\sum_{i=1}^{n} d_i q_i}\right] \left[y_i \frac{\sum_{i=1}^{n} d_i q_i y_i}{\sum_{i=1}^{n} d_i q_i}\right]}{\sum_{i=1}^{n} d_i q_i \left[x_i - \frac{\sum_{i=1}^{n} d_i q_i x_i}{\sum_{i=1}^{n} d_i q_i}\right]^2} \right\} \\ &+ \left(X - \sum_{i=1}^{n} d_i X_i\right) \frac{\sum_{i=1}^{n} d_i q_i \left[x_i - \frac{\sum_{i=1}^{n} d_i q_i x_i}{\sum_{i=1}^{n} d_i q_i}\right] \left[y_i - \frac{\sum_{i=1}^{n} d_i q_i y_i}{\sum_{i=1}^{n} d_i q_i}\right]}{\sum_{i=1}^{n} d_i q_i \left[x_i - \frac{\sum_{i=1}^{n} d_i q_i x_i}{\sum_{i=1}^{n} d_i q_i}\right]^2} \end{aligned}$$

This can be written as

$$\widehat{YTC} = \widehat{Y} + \left(N - \sum_{i \in s} d_i\right) \widehat{A} + \left(X - \widehat{X}\right) \widehat{BTC} \qquad (2.12)$$

where $\widehat{BTC} = \frac{\sum_{i \in s} d_i q_i (x_i - \breve{x})(y_i - \breve{y})}{\sum_{i \in s} d_i q_i (x_i - \breve{x})^2}$ and $\widehat{A} = \breve{y} - \widehat{BTC}\breve{x}$ with $\breve{x} = \frac{\sum_{i \in s} d_i q_i x_i}{\sum_{i \in s} d_i q_i}$ and $\breve{y} = \frac{\sum_{i \in s} d_i q_i y_i}{\sum_{i \in s} d_i q_i}$

The term $(N - \sum_{i \in s} d_i)\,\widehat{A}$ was found to be negligible due to the costraint $\sum_{i \in s} w_i = N$ and the fact that d_i is very close to w_i which in turn means $\sum_{i \in s} d_i \rightarrow \sum_{i \in s} w_i = N$. The term was therefore dropped to obtain

$$\widehat{YTC} = \widehat{Y} + \left(X - \widehat{X}\right) \widehat{BTC} \qquad (2.13)$$

It is noted that there are two basic components in the construction of calibration estimators namely; a distance measure and a set of calibration equations. The choice of a distance measure is less critical in terms of efficiency since the resulting estimators are all asymptotically equivalent to the one obtained by using a chi-squared distance with a certain choice of $q_i's$.

Calibration equation (1.4) is routinely used by many survey organizations and is referred to as benchmark constraint. Benchmark constraints are often imposed in practice for two reasons; the surveyor may believe that the weights which give perfect estimates for the auxiliary information should give a good estimate for the study variable and the auxiliary information may only be available at the aggregate level i.e. only the auxiliary total X_t is known.

2.1.2 Model Calibration

Statisticians in fields such as demography sometimes insist on benchmarking over lots of variables to match the known totals from a census at the risk of worsening the efficiency of the estimators. On the other hand, if complete auxiliary information $x_1, \ldots, x_N$ is known which is usually the case in most survey problems, a very compelling question to ask would be; What is the best calibration equation to be used in the construction of the calibration estimator?

By noting that it is the relationship between y and x hopefully captured by the working model that determines how well the auxiliary information should be used, Wu and sitter (2001) proposed more complex models and generalized the calibration procedure by means of model calibration. In particular, they consider generalized linear models and nonlinear parametric regression models for the super population model ξ, such that $E_{\xi}(y_i) = \mu(x_i)$, where $\mu(x_i)$ is a known function of x_i. They proposed model calibration estimator for population total Y_t to be $\widetilde{Y} = \sum_{i \in s} w_i y_i$ with weights sought to minimize the distance measure (2.1) subject to the then new constraints(1.5) where $\hat{\mu}_i = \hat{\mu}(x_i)$ was parametrically obtained.

In this context, calibration is performed with respect to the population mean of the fitted values $\hat{\mu}_i$ (model calibration). They obtained the estimator

$$Y_{ws} = \widehat{Y} + \left\{ \sum_{i=1}^{N} \hat{\mu}_i - \sum_{i=1}^{N} d_i \hat{\mu}_i \right\} \hat{B}_{WS} \tag{2.14}$$

Where $\hat{\mu}_i = \hat{\mu}(x_i$, $\hat{B}_{WS} = \frac{\sum_{i \in s} d_i q_i (\hat{\mu}_i - \breve{\mu})(y_i - \breve{y})}{\sum_{i \in s} d_i q_i (\hat{\mu}_i - \breve{\mu})^2}$, $\breve{y} = \frac{\sum_{i \in s} d_i q_i y_i}{\sum_{i \in s} d_i q_i}$ and
$\breve{\mu} = \frac{\sum_{i \in s} d_i q_i \hat{\mu}_i}{\sum_{i \in s} d_i q_i}$.

The variance was derived as

$$var(Y_{ws}) = \sum_{i=1}^{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{\mu}_i \hat{B}_{WS}}{\pi_i} \right) \left(\frac{y_j - \hat{\mu}_j \hat{B}_{WS}}{\pi_j} \right) \left(\frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}} \right) \tag{2.15}$$

If X was a random vector with k components, the benchmark constraint (1.4) would consist of k equations, while constraint (1.5) would have only one equation involving the single data reduction variable $\mu(X)$. The single calibration equation (1.5) is indeed more general than the constraints resulting from (1.4) because of the unspecified function $\mu(.)$. Wu and Sitter(2001) show that for any k dimensional vector $X = (x_1, \ldots, x_k)$, if we use $\mu(X) = \theta_0 + \theta_1 x_1 + \ldots + \theta_k x_k$, where $\theta = (\theta_0, \ldots, \theta_k)$ are estimated by ordinary least squares, then the calibration estimator of the population total obtained by using the single constant (1.5) is identical to the one using equation (1.4). The Conventional calibration based on (1.4) is therefore just a special member of the many in the class of calibration estimators.

Model calibration makes this estimator retain efficiency even when the fitted values of y are biased. However, use of parametric model for ξ would require a priori knowledge of specific parametric structure of the population. Without this knowledge, we may end up with a mis-specified model. If there are many variables of interest each of which has x as a covariate, then we would have to establish a parametric relation between x and each of the variables.

Montanari and Ranalli (2003) proposed to use nonparametric method to obtain $\mu(.)$. In particular, they use neural networks and local polynomial. Otieno, Mwita and Kihara (2007) extended this to two stage sampling using kernel functions to fit the mean functions. We note that any nonparametric method such as kernel methods can be used to recover the fitted values for the non sampled units. Such estimators are however challenging to employ in cases of multiple covariates and when data is sparse. Another challenge is how to incorporate categorical covariates. It is therefore necessary to consider other methods to recover the fitted values such as splines.

2.1.3 Estimation by Model Assistance on Nonparametric and Semiparametric Models

The concept of nonparametric models within a model assisted framework was first introduced by Briedt and Opsomer in 2000 in estimating population parameters like population total and mean. The estimator was based on local polynomial smoothing. For a population of size N and where an auxiliary variable x is fully observed, given a sample s of size n for which values for y are fully observed, they proposed the following estimator for population total of the variable y.

$$\hat{Y}_{gen} = \sum_{i \in s} \left(\frac{y_i - \hat{\mu}(x_i)}{\pi_i} \right) + \sum_{j=1}^{N} \hat{\mu}(x_j) \tag{2.16}$$

Where $j = 1, 2, \ldots, N$ and $i = 1, 2, \ldots, n$. $\hat{\mu}(x_i)$ were obtained using local polynomial, a kernel nonparametric method. π_i is the inclusion probability into the sample. $\hat{\mu}(x_i)$ is a smooth function of a single variable x. The first term in (2.16) is an adjustment for bias while the second is an estimator of population total.

The estimator could also be wrtten as

$$\hat{Y}_{gen} = \sum_{i \in s} \frac{y_i}{\pi_i} + \left(\sum_{j=1}^{N} \hat{\mu}(x_j) - \sum_{i \in s} \frac{\hat{\mu}(x_i)}{\pi_i} \right) \tag{2.17}$$

The first term in (2.17) is a design estimator while the second is model component. Therefore, when the sample comprises of the whole population, the model component reduces to zero since $\pi_i = 1$ and $s = N$. We therefore have the actual population total. Among other desirable properties, this estimator has been found to be calibrated with respect to auxiliary variables (internal calibration), though not calibrated with respect to the fitted values $\hat{\mu}(x_i)$. However, this estimator experiences a twin problem of how to determine the optimal bandwidth h and how to determine the optimal degrees (q) of the local polynomial. A higher degree polynomial yields a smoother $\hat{\mu}(.)$ but worsens the boundary variance. These challenges are fairly discussed in Simonoff(1996) and we avoid repeating the discussion here. The ad hock rule is to choose a bandwidth equal to a quater of the data range. Otieno,Mwita and Kihara(2007) showed the rule to be quite reliable. We use this rule in this study. Another challenge with the estimator is that if the fitted values of y are biased, then this estimator loses efficiency. Moreover, accounting for more than one auxiliary variable could be a problem in practice.

When some variable is to enter the estimator of a missing value parametrically, as noted earlier, then the estimator (1.3) has been used. Recall (1.2) has a sample based estimator (1.3).

Let $s \subset U$ be the sample of size n drawn from a population U according to sampling deign $p(s)$ with one way and two way inclusion probabilities $\pi_i = \sum_{i \in s} p(s)$ and $\pi_{ij} = \sum_{i,j \in s} p(s)$ respectively. If the $g_i = g(x_i, z_i)$ were available, Sarndal et al (1992) notes that it would be possible to construct a difference estimator for the population total as

$$y_{dif} = \sum_U g_i + \sum_s \frac{y_i - g_i}{\pi_i} \tag{2.18}$$

Which is design unbiased and has design variance

$$var_p(y_{dif}) = \sum \sum_U (\pi_{ij} - \pi_i \pi_j) \frac{y_i - g_i}{\pi_i} \frac{y_j - g_j}{\pi_j} \tag{2.19}$$

This design variance is small if the deviation between y_i and g_i are small. This estimator is not feasible, since it requires knowledge of all the x_i , z_i and y_i for the population to calculate. Instead, Breidt et al(2007) constructs the following feasible estimator by replacing the g_i with the sample based estimators (1.3)

$$y_{reg} = \sum_U \hat{g}_i + \sum_s \frac{y_i - \hat{g}_i}{\pi_i} \tag{2.20}$$

Defining $\widehat{Y} = \sum_s \frac{y_i}{\pi_i}$ and similarly for $\widehat{Z}$, an equivalent expression for y_{reg} is given by

$$y_{reg} = \widehat{Y} + (\sum_U z_i - \hat{z})\hat{\beta} + \sum_U \hat{\mu}(x)_i - \sum_s \frac{\hat{\mu}(x_i)}{\pi_i} \tag{2.21}$$

This shows that the semi parametric estimator can be interpreted as a traditional linear regression survey estimator using the parametric model component

$z\beta$, with an additional correction term for the nonparametric component of the model. This estimator shares some desirable properties with the fully parametric regression estimators. It is found to be location and scale invariant, and it is internally calibrated for both the parametric and the nonparametric components, in the sense that $\hat{x}_{reg} = \sum_U x_i$ and $\hat{z}_{reg} = \sum_U z_i$. Breidt et al,(2007), showed the estimator (2.20) is design consistent with the rate $\sqrt{n}$, in the sense that $y_{reg} = \sum_U y_i + O_p(\frac{1}{\sqrt{n}})$

The central limit theorem for y_{reg} exists whenever it exists for the expansion estimator $\widehat{Y}$

If $\frac{\frac{\widehat{Y}}{N} - \frac{\sum_U y_i}{N}}{\sqrt{\hat{V}\frac{\widehat{Y}}{N}}} \xrightarrow{d} N(0,1)$ with $\hat{V}(\widehat{y}) = \frac{1}{N^2}\sum\sum_s \frac{\pi_{ij}-\pi_i\pi_j}{\pi_{ij}}\frac{y_i}{\pi_i}\frac{y_j}{\pi_j}$ for a given sampling design, then, Breidt et al (2007) shows that we also have

$$\frac{y_{reg} - \sum_U y_i}{\sqrt{\hat{V}(y_{reg})}} \xrightarrow{d} N(0,1)$$

with

$$\hat{V}(y_{reg}) = \sum\sum_s \frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}} \frac{y_i - \hat{g}_i}{\pi_i} \frac{y_j - \hat{g}_i}{\pi_j} \tag{2.22}$$

As noted earlier, this estimator y_{reg} is internally calibrated based on the benchmark constraint (1.4). It would be interesting therefore to also include model calibration for such an estimator. That is, to base our calibration on the fitted values g_i. Model calibration will make this estimator retain efficiency even when the fitted values of y are biased. We propose to introduce this in the next chapter of this study.

In the next section, we describe the nonparametric techniques that we consider in this thesis in fulfilment of objective three.

2.2 Nonparametric Modeling Techniques

2.2.1 Nadaraya Watson Kernel smoothing

Consider a super population regression model ξ given as

$$y_i = \mu(x_i) + \epsilon \tag{2.23}$$

Where $\mu(x_i)$ is a smooth function.

Consider n pair of observations$(x_i, y_i), \ldots (x_n, y_n)$ from a population of size N. We are interested in $\mu(x) = E_\xi(y/x)$ which is considered smooth. Consider estimates of the form $\hat{\mu}(x) = \sum_{i=1}^n \omega(x, x_i) y_i$ where $\omega(x, x_i)$ is a collection of weights. Consider the weights

$$\omega(x, x_i) = \frac{K(\frac{x_i - x}{h})}{\sum_{i=1}^n K(\frac{x_i - x}{h})}$$

where K is a kernel function and h the bandwidth, (Simonoff, 1996). This results in the following estimator for a target x_j in the population.

$$\hat{\mu}(x_j) = \sum_{i=1}^n \omega(x_j, x_i) y_i \tag{2.24}$$

where $j = 1, 2 \ldots N$. and $i = 1, 2 \ldots .n$. This form was proposed by Nadaraya (1964) and Watson (1964).

2.2.2 Local polynomial

A second approach is local polynomial regression. The objective is to minimize

$$\sum_{j\ =1}^n \{y_j - \beta_0 - \beta_1(x_j - x_i) ... \beta_q(x_j - x_i)^q\}\ ^2 K(x_j - x_i) \tag{2.25}$$

with respect to $\beta = (\beta_0 \beta_1 \ldots \beta_p)$. β_0 estimates $\mu(x_i)$ while$\beta_1 \ldots \beta_p$ estimates higher order derivatives of $\mu(x_i)$ while q is the degree of the polynomial, (Simonoff, 1996). The corresponding estimator can be obtain from a local polynomial smoother

$$S_{lpsi}^T = \epsilon_1^T (X_{si}^T W_{si} X_{si})^{-1} X_{si}^T W_{si} \tag{2.26}$$

as

$$\hat{\mu}(x_i) = S_{lpsi}^T Y_s \tag{2.27}$$

Where

$$\epsilon_1 = (1, 0, \ldots, 0)^T, Y_s = (y_1, y_2, \ldots, y_n)^T, W_{si} = K((x_1 - x_i)/h), \ldots, K((x_n - x_i)/h)$$

and

$$X_{si} = \begin{bmatrix} 1 & (x_1 - x_i) & \ldots & (x_1 - x_i)^q \\ . & & & \\ . & & & \\ . & & & \\ 1 & (x_n - x_i) & \ldots & (x_n - x_i)^q \end{bmatrix}$$

When $q = 0$, it can be shown that we have the Nadaraya Watson kernel smoother, (Breidt and Opsomer, 2000).

2.2.3 Splines

We now describe splines in more detail since it is our area of interest.

The term spline originally referred to a tool used by draftsmen to draw curves. According to Luke Keele,(2008), splines are piecewise regression functions we constrain to join at points called knots. In their simplest form, splines are regression models with a set of dummy variables on the right hand side of the model that are used to force the regression line to change direction at some point along the range of auxilliary variable x. For some simplest regression splines, the piecewise functions are linear; a constraint that is later relaxed. In essence, separate regression lines are fitted within the regions between the knots, and the knots tie together the piecewise regression fits. Again, splines are a local model with local fits between the knots that allow us to estimate the functional form from the data, (Luke Keele, 2008)

Like local polynomial regression, the analyst must make several modeling decisions with splines. With splines, one must choose the degree of polynomial for the piecewise regression functions, the number of knots and the location of knots, (Breidt et al, 2005). It has been found that while the fit is invariant to some of the modeling choices, the analyst must focus on how smooth the fit should be. For some types of splines, the number of knots will control the amount of smoothing, while for other types of splines, a smoothing parameter controls the smoothing.(Breidt et al, 2005).

There are several different types of splines. For example, there are regression splines, cubic splines, B-splines, penalized-splines, natural splines, thin-plate splines, and smoothing splines to name but a few, (De Boor,2001). Moreover, there are often combinations such as natural cubic B-splines. The wide variety of splines partially stem from the progress in research on splines. Often a new type of spline either supplants an older type of spline or adds a refinement to existing methods. (Luke Keele, 2008, Rupert et al 2003). Penalized splines are more complex than regression splines, but they work on the same principle.

The logic behind a regression spline for example is to estimate two separate regression lines that will be joined at the kink in the data. The first regression line will approximate the negative dependency between two variables and the second regression line will approximate the upturn in the functional form. To estimate the spline model, we need to specify the point where the two lines will be joined. Additional piecewise fits would require additional knots. For a single knot k_1 we can write the following regression spline model.

$$y = \beta_0 + \beta_1 x + \beta_2 (x - k_1)_+ + \epsilon \tag{2.28}$$

Where $(x - k_1)_+ = x - k_1$ if $x > k_1$ and 0 if $x \leq k_1$. Below is the model matrix, **X**, constructed after applying a two basis functions to the x variable.

$$X = \begin{bmatrix} 1 & (k_1 - x_1) & 0 \\ . & & \\ . & & \\ 1 & (k_1 - x_{k_1-1}) & 0 \\ 1 & 0 & 0 \\ 1 & 0 & (x_{k_1+1} - k_1) \\ . & & \\ . & & \\ 1 & 0 & (x_n - k_1) \end{bmatrix}$$

Once the model matrix has been formed, estimating a spline fit between x and y is simple. We use the new model matrix to construct the hat matrix $H = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. Application of the hat matrix to the data vector for the outcome produces a set of predictions that form the nonparametric spline estimate of the relationship between x and y. Therefore the spline estimate is $\hat{\beta} = Hy$,

Luke Keele,(2008) finds that the simple regression splines described above to estimate nonlinear dependence between x and y are not suitable for most applied smoothing problems. It's overly restrictive to only estimate piecewise functions that are linear between the knots. To estimate more curvilinear functional forms, the solution is to combine piecewise regression functions with polynomial regression by representing each piecewise regression function as a piecewise polynomial regression function. Piecewise polynomials offer two advantages; First, piecewise polynomials allow for nonlinearity between the knots. Second, piecewise polynomial regression functions ensure that the first derivatives are defined at knots which guarantees that the spline estimate will not have sharp corners.

For the spline model in the last section, we could estimate piecewise polynomial fits by adding x^2 to the basis and squaring the results from the basis functions. These alterations form a quadratic spline basis with single knot at k_1.See Luke Keele,(2008).

Typically, cubic spline bases are used instead of quadratic bases to allow for more flexibility in fitting peaks and valleys in the data. (Breidt et al 2005). A spline model with a cubic basis and two knots k_1 and k_2 is formed from the following linear regression model.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x - k_1)^3_+ + \beta_5 (x - k_2)^3_+ + \epsilon$$

The spline estimate is again the predictions from the hat matrix applied to the outcome variable. To form the hat matrix, we must first construct a model matrix that contains the correct bases. For this example, the model will contain the following data vectors

$$x_1 = x,\ x_2 = x^2,\ x_3 = x^3,\ x_4 = (x - k_1)^3_+, x_5 = (x - k_2)^3_+$$

where x represents the original predictor variable.

The model matrix will consist of a 1 and the above five variables. We use the model matrix to form a hat matrix that is applied to the outcome variable, and the predictions from this model serve as the spline estimate of the possibly nonlinear relationship between x and y. The number of parameters used to construct the spline estimate is controlled by the number of knots. If there are k knots, with a cubic basis, the function will require $k + 4$ regression coefficients (including the intercept). The cubic basis allows for flexible fits to nonlinearity between the knots and eliminates any sharp corners in the resulting estimate. The later is true since the first derivative exists for $(x - k_1)^3_+$ and it follows that the first derivative will also exist for any linear combination of terms of the model data vectors, (Rupert et al, 2003). In any spline model the analyst must select the number of knots and decide where they should be placed along the range of x. Stone (1986) found that where the knots are placed matters less than how many knots are used. Standard practice is to place knots at evenly placed intervals in the data to ensure that there is enough data within each region of x to get a smooth fit. Knots are mostly placed at either quartiles or quintiles in the data. If the data has an obvious feature, it may be useful to place the knots in a less automatic fashion(Luke Keele,(2008). In our study, we have placed the knots at quintiles rather than at quartiles so that we have a higher number of knots which is better because number of knots chosen affects the amount of smoothing applied to the data by controlling the number of piecewise fits. A spline with two knots will be linear and globally smooth since there is only one piecewise function. Increasing the number of knots increases the number of piecewise functions fit to the data allowing for greater flexibility. If one selects a large enough number of knots, the spline model will interpolate between the data points, since more knots shrink the amount of data used for each piecewise function. The number of knots effectively acts as a span parameter for splines. If one uses a small number of knots, the spline estimate will be overly smooth with little variability but may be biased. Using a high number of knots implies little bias but increases variability in the fit and may result in over fitting, (Rupert et al, 2003, Breidt et al, 2005) but this can be solved by penalizing the splines.

If we want a flexible estimate of the statistical relationship between two variables, both splines and local polynomial regression can provide such an estimate with few assumptions about the functional form. It is easy to have a surfeit of local parameters which produces overly nonlinear nonparametric estimates that overly fit data. Penalized splines are a nonparametric regression technique that relies on principles of statistical theory to minimize the possibility of over fitting.

2.2.3.1 Penalized splines

It is possible to over fit both parametric and nonparametric regression models. Over fit statistical models have too many parameters relative to the amount of data and cause random variation in the data to appear as a systematic effects. A solution of over fitting is penalized estimation, (Eilers and Marx, 1996). Here, for each parameter used in the model, a penalty is added to the model. For the P-spline estimator for the function $\mu(x)$, from the regression estimate

$$\hat{\mu}(x, \beta) = \beta_0 + \beta_1 x + \ldots + \beta_q x^q + \sum_{\kappa=1}^{k} \beta_{q+\kappa}(x - k_\kappa)_+^q \tag{2.29}$$

We bind $\sum_{\kappa=1}^{k} \beta_{q+\kappa}^2$ by some constant, while leaving the polynomial coefficients $\beta_0, \ldots, \beta_q$ unconstrained. Breidt et al(2005) obtained an estimate for β by minimizing

$$\sum_{i\, \in U} (y_i - \mu(x_i, \beta))^2 + \alpha \sum_{\kappa\, =1}^{k} \beta_{\kappa+q}^2 \tag{2.30}$$

for some fixed constant $\alpha \geq 0$ that determines the smoothness of the obtained fit.

They obtained a sample design consistent estimator for β as

$$\hat{\beta} = (X_s^T W_s X_s + A\alpha)^{-1} X_s^T W_s Y_s$$

where X_s is a sub matrix of a matrix X which inturn consist of rows $X_i^T = \{1, x_i, \ldots, x_i^q, (x_i - k_1)_+^q, \ldots, (x_i - k_\kappa)_+^q\}$for $i \in U$, $A_\alpha = diag\{0, \ldots, 0, \alpha, \ldots, \alpha\}$ with $q + 1$ zeros on the diagonal followed by k penalty constants α.

The corresponding nonparametric sample fit is

$$\mu(x_i; \hat{\beta}) = X_i^T \hat{\beta} \tag{2.31}$$

Consider matrices F and R with rows $(1, x_i, \ldots, x_i^q)$ and $((x_i - k_1)_+^q, \ldots, (x_i - k_\kappa)_+^q)$ respectively. Jiang(1996), constructed a design based $\sqrt{n}$-consistent estimator for α as

$$\hat{\alpha}_s = \frac{tr((A^T V A)^{-1} A^T R R^T A)}{tr((A^T V A)^{-1} A^T A)} \tag{2.32}$$

where $V = var(Y)$ and A is a matrix such that $A^T F = 0$

When some of the auxilliary information contain a parametric component like categorical data, nonparametric modelling may not be sufficient. In our study, we introduce semiparametric modelling technique which suits such a scenario.

2.3 Estimation of Population Total Under Two Stage Sampling

The application of internal calibration and model assistance on nonparametric models in the estimatetion of population total under two stage sampling was introduced by Breidt et al in 2005. They Consider a population U partitioned into M clusters each of size N_i so that the population of clusters is $C = 1, \dots, i, \dots, M$. For all clusters $i \in s$,an auxiliary vector x_i is available considered to be a scalar. At stage one, a probability sample s of clusters is drawn from C according to a fixed design $p_1(.)$, where $p_1(s)$ is the probability of drawing the sample s from C. They let m be the size of s. The cluster inclusion probabilities are $\pi_i = p(i \in s)$ and $\pi_{ij} = p(i, j \in s)$. p_1 refers to first stage design. From every sampled cluster $i \in s$, a probability sample s_i of elements is drawn according to a fixed size design $p_i(.)$ with inclusion probabilities $\pi_{k/i} = p(k \in s_i/i \in s)$ and $\pi_{kl/i} = p(k, l \in s_i/i \in s)$. They let n_i be the size of s_i and assumed invariance and independence of the second stage design and let $t_i, i = 1, 2, \dots, M$ be the ith cluser total.

They Let $\hat{t}_s = \left[\hat{t}_i\right]_{i \in s}$ be the m vector of $\hat{t}_i' s$ obtained in the sample of clusters, where $\hat{t}_i = \sum_{k \in s_i} \frac{y_{ik}}{\pi_k}$ is the Horvitz-Thompson design estimate of ith cluster total. In their approach to using the auxilliry information and using local polynomial, Breidt et al (2005) assumed as a working model that the finite population scatter $(x_i, t_i)_{i \in C}$ is a realization from a superpopulation model ξ in which

$$t_i = \mu(x_i) + \epsilon_i \tag{2.33}$$

They obtained the nonparametric local polynomial estimator of $\mu(x_i)$ as

$$\hat{\mu(x_i)} = S_{lpsi}^T \hat{t_s} \tag{2.34}$$

and a local polynomial regression estimator for population total as

$$\hat{Y_{gen2}} = \sum_{i \in s} \frac{\hat{t_i}}{\pi_i} + \left(\sum_{i \in C} \hat{\mu}(x_i) - \sum_{i \in s} \frac{\hat{\mu}(x_i)}{\pi_i} \right) \tag{2.35}$$

We consider modelling $\hat{\mu(x_i)}$ by way of penalized splines and perform model calibration on $\hat{\mu(x_i)}$. We consider the case when auxilliary information is available at cluter level alone like did Breidt et al in 2005 and the case when auxilliary information is available at both cluster and element level.

2.4 Sensitivity Analysis

Sensitivity analysis have been used when fitting missing values using fully model based procedures that rely on estimation methods such as maximum likelihood. This is because they are based on model assumptions which are in most cases untestable,(Schafstein et al ,2003,). We carry out a Sentivity analysis to test the robustness of model calibrated estimators as compared to internally calibrated ones based on semiparametric modelling. This is a new area we have ventured into.

CHAPTER THREE

3.0 PROPOSED MODEL CALIBRATED AND ASSISTED ESTIMATOR

3.1 Estimators for One Stage Sampling Based on Penalized Splines

Consider a population U of size N for which several values for a random variable y are missing. Suppose a multidimensional covariate consisting of a single variable x that is to enter the estimation nonparametrically and a categorical vector Z are fully observed. Let s be a sample of size n from the population and for which all the variables are fully observed. Consider the matrix X_c with rows

$$X_{ci}^T = \{1, x_i, \ldots, x_i^q, (x_i - k_1)_+^q, \ldots, (x_i - k_k)_+^q\} \tag{3.1}$$

for $i \in U$, and let X_{cs} be the sub matrix of X_c consisting of those rows for which $j \in s$. let Y be the vector of response values y_i for $i \in U$ and $A_\alpha = (0, \ldots, 0, \alpha, \ldots, \alpha)$, with $q+1$ zeros on the diagonal followed by the penalty constant α repeated k times . Consider also the diagonal matrix of inverse inclusion probabilities $W = j \in U(\frac{1}{\pi_j})$ and its sample sub matrix $W_s = j \in s(\frac{1}{\pi_j})$.
Define a superpopulation model

$$Y_i = \mu_i + Z_i\beta \tag{3.2}$$

and let the semiparametric estimator for $E_\xi(y_i)$ be

$$\hat{g}_i = \hat{\mu}_i + Z_i\hat{\beta} \tag{3.3}$$

The design weighted penalized spline smoother vector at x_i similar to the local polynomial smoother (2.26) due to Breidt and Opsomer,(2000) is

$$S_{spsi}^T = X_{ci}\left(X_{cs}^T W_s X_{cs} + A_\alpha\right)^{-1} X_{cs}^T W_s \tag{3.4}$$

This is such that when applied to the sample Y_s it yields the nonparametric fit at x_i. That is $\hat{\mu}_i = S^T_{spsi} Y_s$.

The sample smoother matrix is a matrix with rows S^T_{spsi}.

$$S_{sps} = \left[S^T_{spsi}, i \in s\right] \tag{3.5}$$

Accordingly, we therefore have the following estimators resulting from equations (3.3), (3.4) and (3.5) for fixed α and under regularity conditions in section(3.2.1)

$$\hat{\beta} = \left(Z^T_s S_{sps} Z_s\right)^{-1} Z^T_s S_{sps} Y_s \tag{3.6}$$

so that an estimator for $Z_i\beta$ becomes $Z_i\hat{\beta}$. Now, from (3.2), an estimator $\hat{\mu}_i$ for the smooth function μ_i is obtained by smoothing the residue $\left(Y_s - Z^T_s\hat{\beta}\right)$ so that we get

$$\hat{\mu}_i = S^T_{spsi}\left(Y_s - Z^T_s\hat{\beta}\right) \tag{3.7}$$

where $\hat{\mu}_i$ and x_i are defined for every $i \in U$. We can now rewrite the resulting semiparametric fit $\hat{g}_i$ as

$$\hat{g}_i = S^T_{spsi}\left(Y_s - Z^T_s\hat{\beta}\right) + Z_i\left(Z^T_s S_{sps} Z_s\right)^{-1} Z^T_s S_{sps} Y_s \tag{3.8}$$

We now propose a semiparametric model assisted model calibrated estimator of population total to be

$$\hat{y}_{sm} = \sum_{i \in s} w_i y_i \tag{3.9}$$

with w_i obtained by minimizing the chi square distance measure (2.1) subject to the constraints $\sum_{i\in s} w_i = N$ and $\sum_{i\in s} w_i\hat{g}_i = \sum_{i\in U}\hat{g}_i$. We introduce the lagrange procedure in the minimization of equation(2.1) to obtain an equation of the form

$$l = \sum_{i\in s}\frac{(w_i - d_i)^2}{q_i d_i} - 2\lambda(\sum_{i\in s} w_i\hat{g}_i - \sum_{i\in U}\hat{g}_i) - 2v(\sum_{i\in s} w_i - N) \tag{3.10}$$

Where λ is the lagrange's multiplier and v is the penalty constant. Differentiating (3.10) with respect to w_i and equating to zero we get

$$\begin{aligned}\frac{\partial l}{\partial w_i} &= \frac{2(w_i - d_i)}{q_i d_i} - 2\lambda\hat{g}_i - 2v \\ &= 0\end{aligned} \tag{3.11}$$

which gives

$$w_i = (\lambda \hat{g}_i + v) q_i d_i + d_i \tag{3.12}$$

solving for λ and v we have

$$\begin{aligned} w_i = d_i + (N - \sum_{i \in s} d_i) & \left\{ \frac{d_i q_i}{\sum_{i \in s} d_i q_i} - \left\{ \frac{q_i d_i \left(\hat{g}_i - \frac{\sum_{i \in s} d_i q_i \hat{g}_i}{\sum_{i \in s} d_i q_i} \right) \left(1 - \frac{\sum_{i \in s} d_i q_i}{\sum_{i \in s} d_i q_i} \right)}{\sum_{i \in s} q_i d_i \left(\hat{g}_i - \frac{\sum_{i \in s} d_i q_i \hat{g}_i}{\sum_{i \in s} d_i q_i} \right)^2} \right\} \right\} \\ & + \left\{ \sum_{i \in U} \hat{g}_i - \sum_{i \in s} d_i \hat{g}_i \right\} \left\{ \frac{q_i d_i \left(\hat{g}_i - \frac{\sum_{i \in s} d_i q_i \hat{g}_i}{\sum_{i \in s} d_i q_i} \right) \left(1 - \frac{\sum_{i \in s} d_i q_i}{\sum_{i \in s} d_i q_i} \right)}{\sum_{i \in s} q_i d_i \left(\hat{g}_i - \frac{\sum_{i \in s} d_i q_i \hat{g}_i}{\sum_{i \in s} d_i q_i} \right)^2} \right\} \end{aligned} \tag{3.13}$$

Substitutig w_i in (3.9) we have

$$\hat{y}_{sm} = \sum_{i \in s} d_i y_i + (N - \sum_{i \in s} d_i) \left\{ \frac{\sum_{i \in s} d_i q_i y_i}{\sum_{i \in s} d_i q_i} - \hat{\beta}_m \right\} + \left\{ \sum_{i \in U} \hat{g}_i - \sum_{i \in s} d_i \hat{g}_i \right\} \hat{\beta}_m \tag{3.14}$$

where $\hat{\beta}_m = \left\{ \frac{\sum_{i \in s} q_i d_i \left(\hat{g}_i - \frac{\sum_{i \in s} d_i q_i \hat{g}_i}{\sum_{i \in s} d_i q_i} \right) \left(y_i - \frac{\sum_{i \in s} d_i q_i y_i}{\sum_{i \in s} d_i q_i} \right)}{\sum_{i \in s} q_i d_i \left(\hat{g}_i - \frac{\sum_{i \in s} d_i q_i \hat{g}_i}{\sum_{i \in s} d_i q_i} \right)^2} \right\}$

<u>Theorem 1</u>: The term $(N - \sum_{i \in s} d_i) \left\{ \frac{\sum_{i \in s} d_i q_i y_i}{\sum_{i \in s} d_i q_i} - \hat{\beta}_m \right\}$ is neglible. The proof is provided in appendix 1.

It therefore suffices to write our estimator as

$$\hat{y}_{sm} = \sum_{i \in s} \frac{y_i}{\pi_i} + \left\{ \sum_{i \in U} \hat{g}_i - \sum_{i \in s} \frac{\hat{g}_i}{\pi_i} \right\} \hat{\beta}_m \tag{3.15}$$

where $\hat{g}_i = Z_i \hat{\beta} + \hat{\mu}(x_i)$ and $d_i = (\pi_i)^{-1}$ with $\hat{\beta}$ and $\hat{\mu}$ computed as defined in (3.6) and (3.7) respectively.

We propose an estimator of finite population mean as

$$\hat{\bar{y}}_{sm} = \frac{1}{N} \sum_{i \in s} \frac{y_i}{\pi_i} + \frac{1}{N} \left\{ \sum_{i \in U} \hat{g}_i - \sum_{i \in s} \frac{\hat{g}_i}{\pi_i} \right\} \hat{\beta}_m \tag{3.16}$$

If local polynomial is used to fit the missing values, then the semiparametric estimator for $E_{\xi}(y_i)$ is given as

$$\hat{g}_i = S_{lpsi}^T \left(Y_s - Z_s^T \hat{\beta} \right) + Z_i \left(Z_s^T S_{lps} Z_s \right)^{-1} Z_s^T S_{lps} Y_s \tag{3.17}$$

where

$$S_{lps} \quad = \quad \left[S_{lpsi}^T, i \in s\right] \tag{3.18}$$

A Nadaraya Watson fit is obtained as in (3.17) but with the degree q in S_{lpsi}^T being zero.

When no part of the auxilliary information require to enter the estimation parametrically, then nonparametric estimation would be sufficient. We define the nonparametric sample fit for$E_\xi(y_i)$ based on penalized splines as

$$\hat{\mu}_i = S_{spsi}^T Y_s \tag{3.19}$$

while if based on local polynomial we have

$$\hat{\mu}_i = S_{lpsi}^T Y_s \tag{3.20}$$

We minimize the chi square distance measure (2.1) subject to the constraints $\sum_{i\in s} w_i = N$ and $\sum_{i\in s} w_i \hat{\mu}_i = \sum_{i\in U} \hat{\mu}_i$ and solve for w_i to obtain

$$w_i = (\lambda \hat{\mu}_i + v) q_i d_i + d_i \tag{3.21}$$

Solving for v and λ then substituting w_i into the noparametric equivalent of (3.9) we obtain

$$\hat{y}_{np} = \sum_{i\in s} \frac{y_i}{\pi_i} + \left\{ \sum_{i\in U} \hat{\mu}_i - \sum_{i\in s} \frac{\hat{\mu}_i}{\pi_i} \right\} \hat{\beta}_m \tag{3.22}$$

in which case $\hat{\beta}_m = \left\{ \frac{\sum_{i\in s} q_i d_i \left(\hat{\mu}_i - \frac{\sum_{i\in s} d_i q_i \hat{\mu}_i}{\sum_{i\in s} d_i q_i} \right) \left(y_i - \frac{\sum_{i\in s} d_i q_i y_i}{\sum_{i\in s} d_i q_i} \right)}{\sum_{i\in s} q_i d_i \left(\hat{\mu}_i - \frac{\sum_{i\in s} d_i q_i \hat{\mu}_i}{\sum_{i\in s} d_i q_i} \right)^2} \right\}$

A corresponding estimator for population mean is therefore

$$\hat{\bar{y}}_{np} = \frac{1}{N} \sum_{i\in s} \frac{y_i}{\pi_i} + \frac{1}{N} \left\{ \sum_{i\in U} \hat{\mu}_i - \sum_{i\in s} \frac{\hat{\mu}_i}{\pi_i} \right\} \hat{\beta}_m \tag{3.23}$$

3.2 Theoretical Properties Under One Stage Sampling

In this section, we show that

$\hat{y}_{sm} = \sum_{i=1}^n \frac{y_i}{\pi_i} + \left\{ \sum_{i=1}^N \hat{g}_i - \sum_{i=1}^n \frac{\hat{g}_i}{\pi_i} \right\} \hat{\beta}_m$

where $\hat{g}_i = \hat{g}(x_i, Z_i) + \epsilon_i$,is design unbiased, consistent and asymptotic normally distributed under the assumptions listed below. These properties are confirmed in the empirical analysis and hold even when the model is mis-spcefied. Assumptions 2-5 are similar to those of Breidt et al(2005) while assumption 6 is from Wu and Sitter(2001).

3.2.1 Assumptions

1. We assume that there is a sequence of finite populations indexed by ρ each of size N_ρ but which for ease of representation we write N and each estimated by $\hat{y}_{sm\rho}$ but which we again simply write as $\hat{y}_{sm}$.

2. As $\rho \to \infty, n \to \infty, N \to \infty$, the number of knots $k \to \infty$ while bandwidth $h \to 0$.

3. For each ρ, the x_i, for $i = 1, 2, ..., N$ are indepedent and identically distributed $F(x) = \int_{-\infty}^{x} f(t)dt$ where $f(.)$ is a density with compact support $[a_x, b_x]$ and $f(x) > 0$ for all $x \in [a_x, b_x]$. The Z_i have bounded suport.

4. For each ρ the x_i and Z_i are considered fixed with respect to the model ξ while the errors ϵ_i are indepedent and have mean zero, variance $var(x_i, Z_i)$ and compact support, uniformly for each ρ.

5. Every element in a population has an inclusion probability $\pi_i > 0$ and any two distinct elements have a joint inclusion probability $\pi_{ij} > 0$

6. The sampling design is regular so that the inclusion probabilities are independent of response measurements and satisfies the conditions $\max_{i \in s} \frac{n}{N\pi_i} = 0(1)$, where π_i is the inclusion probability, and $\sum_{i \in s} \frac{g_i}{\pi_j} - \sum_{i=1}^{N} g_i = 0_p(Nn^{-\frac{1}{2}})$.

Assumption 1 ensures there is a sequence of estimates which are necessary in establishing consistency. Assumption 2 ensures that as the population size grows, the amount of data within a neighbourihood defined by a knot in case of splines or bandwidth in case of kernel smoothing is reasonable and does not become exeptionaly large. Assumption 3 ensures that the $\{x_i\}$ are a random sample from a continuous distribution. Assumption 4 is necessary to make the results in later sections to look like traditional(non-asymptotic) finite population results while assumption 5 ensures every element has a probability, not zero, of being included in the sample. First condition in assumption 6 says that no basic design weight is dispropotionaly large while the second condition is equivalent to assuming that Horvitz thompson estimator for $\sum_{i=1}^{N} g_i$ is asymptotically normally distributed.

3.2.2 Asymptotic Design Unbiasedness

Let E_p be design expectation and E_ξ model based expectation. We need to show that $\{E_p(\hat{y}_{sm})\} = Y_t$. Under assumptios 1-6, and the fact that with respect to design expectation, $\hat{g}_i$ and $\hat{\beta}_m$ are treated as a constants and the fact that $E_P(I_i) = \pi_i$.

Now,

$$\begin{aligned}
\{E_p(\hat{y}_{sm})\} &= E_p\left\{\sum_{i\in s}\frac{y_i}{\pi_i} + \left\{\sum_{i\in U}\hat{g}_i - \sum_{i\in s}\frac{\hat{g}_i}{\pi_i}\right\}\hat{\beta}_m\right\} \\
&= E_p\left\{\sum_{i\in U}\frac{y_i I_i}{\pi_i} + \left\{\sum_{i\in U}\hat{g}_i - \sum_{i\in U}\frac{\hat{g}_i I_i}{\pi_i}\right\}\hat{\beta}_m\right\} \\
&= \sum_{i\in U}\frac{E_p y_i I_i}{\pi_i} + \sum_{i\in U} E_p\hat{g}_i\hat{\beta}_m - \sum_{i\in U}\frac{E_p(\hat{g}_i\hat{\beta}_m I_i)}{\pi_i} \\
&= \sum_{i\in U}\frac{E_p y_i I_i}{\pi_i} + \sum_{i\in U} E_p\hat{g}_i\hat{\beta}_m - \sum_{i\in U}\frac{(\hat{g}_i\hat{\beta}_m E_p I_i)}{\pi_i} \\
&= \sum_{i\in U}\frac{\bar{y}}{1} + \sum_{i\in U}\hat{g}_i\hat{\beta}_m - \sum_{i\in U}\frac{\hat{g}_i\hat{\beta}_m}{1} \qquad (3.24)
\end{aligned}$$

Thus we have, $\sum_{i\in U}\bar{y} = N\bar{y} = Y_t$

3.2.3 Model Bias Reduction

We show that $\hat{y}_{sm} = \sum_{i=1}^{n}\frac{y_i}{\pi_i} + \left\{\sum_{i=1}^{N}\hat{g}_i - \sum_{i=1}^{n}\frac{\hat{g}_i}{\pi_i}\right\}\hat{\beta}_m$

has reduced model bias. Now, $\hat{\beta}_m$ is an estimate of the change in Y_t when g_i is increased by a unit. The rationale of this estimator is that if $\sum_{i\in s}\frac{\hat{g}_i}{\pi_i}$ is below average, we should expect the estimate of the population total Y_t to be below average by an amount $\left\{\sum_{i\in U}\hat{g}_i - \sum_{i\in s}\frac{\hat{g}_i}{\pi_i}\right\}\hat{\beta}_m$ due to regression of y_i on $\hat{g}_i$. Cochran(1997). The estimate $\hat{g}_i$ need not be free from bias. If $\hat{g}_i - y_i = D$, so that the estimate is perfect except for a constant bias D, then with $\hat{\beta}_m = 1$ the regression estimate becomes

$$\sum_{i\,\in s}\frac{y_i}{\pi_i} + \left\{\sum_{i\in U}\hat{g}_i - \sum_{i\in s}\frac{\hat{g}_i}{\pi_i}\right\} = \sum_{i\in U}\hat{g}_i + \left\{\sum_{i\in s}\frac{y_i}{\pi_i} - \sum_{i\in s}\frac{\hat{g}_i}{\pi_i}\right\} \tag{3.25}$$

The first term to the right is population total estimate and the second term to the right is an adjustment for bias. This regression estimate is consistent in the sense that when the sample comprises the whole population, then$\sum_{i\in U}\hat{g}_i = \sum_{i\in s}\frac{\hat{g}_j}{\pi_j}$ and the regression estimate reduces to $\sum_{i\in s}\frac{y_i}{\pi_i}$. See Firth and Bennett (2006).Thus, establishing a CLT for generalized difference estimator is essentially the same as establishing a CLT for Horvitz-Thompson estimator.

3.2.4 Design Consistency

Under assumptions 1-6, the chebycheve's inequality, $pr[|x_n - \theta| > \epsilon] \leq E_P\frac{|x_n-\theta|^2}{\epsilon^2}$ and a sequence of the estimates $\hat{y}_{sm\rho}$ but which we simply write as $\hat{y}_{sm}$ for ease of representation. We have that $pr[|\hat{y}_{sm} - Y_t| > \epsilon] \leq E_P\frac{|\hat{y}_{sm}-Y_t|^2}{\epsilon^2}$

but since $\hat{y}_{sm}$ is unbiased for Y_t, then the mean squared error can consistently be estimated by $var(\hat{y}_{sm})$, so that $pr[|\hat{y}_{sm} - Y_t| > \epsilon] \leq \frac{var\{\hat{y}_{sm}\}}{\epsilon^2}$

and $\lim_{\rho\to\infty} pr[|E_p\hat{y}_{sm} - Y_t| > \epsilon] \leq \lim_{\rho\to\infty} \frac{var\{\hat{y}_{sm}\}}{\epsilon^2}$. We see that,

$$\begin{aligned}\lim_{\rho\to\infty}\frac{var\{\hat{y}_{sm}\}}{\epsilon^2} &= \lim_{\rho\to\infty} E_p\sum_{i=1}^{n}\sum_{j=1}^{n}(\frac{y_i - g_i\hat{\beta}_m}{\pi_i})(\frac{y_j - g_j\hat{\beta}_m}{\pi_j})(\frac{\pi_{ij}-\pi_i\pi_j}{\pi_{ij}})\frac{1}{\epsilon^2} \\ &= \lim_{\rho\to\infty} E_p\sum_{i=1}^{N}\sum_{j=1}^{N}(\frac{y_i - g_i\hat{\beta}_m}{\pi_i})(\frac{y_j - g_j\hat{\beta}_m}{\pi_j})(\frac{\pi_{ij}-\pi_i\pi_j}{\pi_{ij}})\frac{I_iI_j}{\epsilon^2} = 0\end{aligned} \tag{3.26}$$

Since $E_p(\pi_{ij}) = \pi_i\pi_j, E_p(\pi_i\pi_j) = \pi_i\pi_j, E_p(\pi_i) = \pi_i$ and $E_p(I_iI_j) = \pi_{ij} \leq \pi_i\pi_j$ See Kott P.S.(2003). We must therefore also have that,

$\lim_{\rho\to\infty} pr[|E_p\hat{y}_{sm} - Y_t| > \epsilon] \to 0$, that is , $\hat{y}_{sm} \xrightarrow{p} Y_t$

3.2.5 Asymptotic Normality

Theorem 2: Let g_i be the population fit assumed known for every population element, then

$$y^*_{sm} = \sum_{i=1}^{n}\frac{y_i}{\pi_i} + \left\{\sum_{i=1}^{N} g_i - \sum_{i=1}^{n}\frac{g_i}{\pi_i}\right\}\beta^*_m \tag{3.27}$$

where

$\beta^*_m = \frac{\sum_{j=1}^{N}\frac{1}{\pi_i}q_i(g_i-\bar{g})(y_i-\bar{Y})}{\sum_{i=1}^{N}\frac{1}{\pi_i}q_i(g_i-\bar{g})^2}$, $\bar{g} = N^{-1}\sum_{i=1}^{N} g_i$ and $\bar{y} = N^{-1}\sum_{i=1}^{N} y_i$

is an asymptotic normal estimator for population total in the sense that $\frac{(y^*_{sm}-Y_t)}{var^{1/2}(y^*_{sm})} \xrightarrow{d} N(0,1)$ as $\rho \to \infty$

proof: Consider the Horvitz - Thompson estimator

$$y_{ht} = \sum_{i=1}^{n}\frac{y_i}{\pi_i} \tag{3.28}$$

with variance

$$V(y_{ht}) = \sum_{i,j\in U}(\pi_{ij} - \pi_i\pi_j)\frac{y_i}{\pi_i}\frac{y_j}{\pi_j} \tag{3.29}$$

This estimator is known to be normaly distributed. If population fits $\mu_i = e_i^T\left(X^T_{Ui}W_{Ui}X_{Ui}\right)^{-1}X^T_{Ui}W_{Ui}\hat{Y}_s$ are known, under assumptions 1-6, and by theorem 3 of Breidt and Opsomer(2000) and theorem 3.2 of Thompson(1997), the resulting difference estimator

$$\hat{Y}^{*}_{diff} = \sum_{i\in s} \frac{y_i}{\pi_i} + \left(\sum_{i\in U} \mu_i - \sum_{i\in s} \frac{\mu_i}{\pi_i} \right) \tag{3.30}$$

is found to be asymptotic normal in the sense that $\frac{\hat{Y}^{*}_{diff} - Y_t}{var^{1/2}(\hat{Y}^{*}_{diff})} \xrightarrow{d} N(0,1)$ as $\rho \to \infty$. The estiamtor's variance

$$V(\hat{Y}^{*}_{diff}) = \sum_{i,j\in U} (\pi_{ij} - \pi_i\pi_j) \frac{y_i - \mu_i}{\pi_i} \frac{y_j - \mu_j}{\pi_j} \tag{3.31}$$

is less than that of Horvitz-Thomson estimator due to $y_i - \mu_i$ and $y_j - \mu_j$. They find that establishing a CLT for $\hat{Y}^{*}_{diff}$ is the same as establishing a CLT for the Horvitz-Thompson estimator. The estimator y^{*}_{sm} is similar to $\hat{Y}^{*}_{diff}$ with the difference being the regression coefficient and the fact that penalized spline is used for the fits. Its variance,

$$V(\hat{Y}^{*}_{sm}) = \sum_{i,j\in U} (\pi_{ij} - \pi_i\pi_j) \frac{y_i - g_i\beta^{*}_{m}}{\pi_i} \frac{y_j - g_j\beta^{*}_{m}}{\pi_j} \tag{3.32}$$

is still less than that of Horvitz-Thomson estimator. By theorem 3 of Breidt and Opsomer(2000) and theorem 3.2 of Thompson(1997), y^{*}_{sm} is aymptotic normal. Establishing a CLT for y^{*}_{sm} is the same as establishing a CLT for $\hat{Y}^{*}_{diff}$ which is in turn the same as establishing a CLT for the Horvitz-Thompson estimator.

Theorem 3: Let y^{*}_{sm} be as defined in theorem 2. Then,
$\frac{(y^{*}_{sm} - Y_t)}{var^{1/2}(y^{*}_{sm})} \xrightarrow{d} N(0,1)$ as $\rho \to \infty$ implies that $\frac{(\hat{y}_{sm} - Y_t)}{var^{1/2}(\hat{y}_{sm})} \xrightarrow{d} N(0,1)$
where

$$var(\hat{y}_{sm}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \left(\frac{y_i - \hat{g}_i\hat{\beta}_m}{\pi_i} \right) \left(\frac{y_j - \hat{g}_j\hat{\beta}_m}{\pi_j} \right) \left(\frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}} \right) \tag{3.33}$$

proof: We need to show that $(\hat{y}_{sm} - Y_t)$ converges to $(y^{*}_{sm} - Y_t)$ in distribution. This would imply that $\hat{y}_{sm}$ inherits limiting distributional properties of $\hat{\hat{y}}_{sm}$. This, coupled with theorem 2 would proof the above.

Now,

$(\hat{y}_{sm} - Y_t) = \sum_{i=1}^{N} \frac{y_i I_i}{\pi_i} + \sum_{i=1}^{N} \hat{g}_i\hat{\beta}_m - \sum_{i=1}^{N} \frac{\hat{g}_i I_i \hat{\beta}_m}{\pi_i} - \sum_{i=1}^{N} y_i$ and $(y^{*}_{sm} - Y_t) =$

$\sum_{i=1}^{N} \frac{y_i I_i}{\pi_i} + \sum_{i=1}^{N} g_i \beta_m^* - \sum_{i=1}^{N} \frac{g_i I_i \beta_m^*}{\pi_i} - \sum_{i=1}^{N} y_i$

Clearly,

$$\hat{y}_{sm} - y_{sm}^* = \sum_{i=1}^{N} \left(\hat{g}_i \hat{\beta}_m - g_i \beta_m^*\right) \left(1 - \frac{I_i}{\pi_i}\right) \tag{3.34}$$

Taking limits of the expectation,we have

$$\lim_{\rho \to \infty} E_P \left\{\hat{y}_{sm} - y_{sm}^*\right\} = \lim_{\rho \to \infty} E_P \left\{\sum_{i=1}^{N} \left(\hat{g}_i \hat{\beta}_m - g_i \beta_m^*\right) \left(1 - \frac{I_i}{\pi_i}\right)\right\} \tag{3.35}$$

It can be seen that the design expectation of $\hat{y}_{sm} - y_{sm}^*$ approaches zero since design expectation of I_i is π_i. Thus the limit is zero. This is convergence in mean which implies convergence in probability and convergence in distribution.

The right hand term in (3.34) can be written as

$$\left(\hat{\beta}_m - \beta_m^*\right) \sum_{i\ =1}^{N} \hat{g}_i \left(1 - \frac{I_i}{\pi_i}\right) + \beta_m^* \sum_{i\ =1}^{N} \left(\hat{g}_i - g_i\right) \left(1 - \frac{I_i}{\pi_i}\right) \tag{3.36}$$

Now, $\hat{g}_i - g_i = O_p(1)$ from lemma 4 in Breidt and Opsomer(2000), $\hat{\beta}_m - \beta_m^* = O_p(1)$ from an argument similar to that of lemma 4 in Montanari and Ranalli(2003). $\sum_{i=1}^{N} \hat{g}_i \left(1 - \frac{I_i}{\pi_i}\right) = O_p(Nn^{-1/2})$ and $\sum_{i=1}^{N} \left(\hat{g}_i - g_i\right) \left(1 - \frac{I_i}{\pi_i}\right) = O_p(Nn^{-1/2})$ from the proof of theorem 2 in Breidt and Opsomer(2000). Therefore the term in equation(3.36) is of order $O_p(Nn^{-1/2})$.

The estimators $\hat{y}_{np}$ and $\hat{y}_{sm}$ only differ in the way the missing values were obtained. In $\hat{y}_{np}$, a nonparametric method is used while for $\hat{y}_{sm}$ a semiparametric method is used. Generally, their structure is similar. The theoretical properties for $\hat{y}_{np}$ are obtained in quite a similar manner as for $\hat{y}_{sm}$

3.3 Extensions To Two Stage Sampling

We consider a case where auxilliary information is available only at the cluster level and when it is availbale at both element and cluster levels.

3.3.1 Auxilliary Information at Cluster Level Only

Consider a population U partitioned into M clusters each of size N_i so that the population of clusters is $C = \{1, \ldots, i, \ldots, M\}$. For all clusters $i \in C$, an auxiliary variable x is observed and a categorical vector Z is also available. At stage one, a probability sample s of clusters is drawn from C according to a fixed design $p_1(.)$, where $p_1(s)$ is the probability of drawing the sample s from C. let m be the size of s. The cluster inclusion probabilities $\pi_i = p(i \in s)$ and $\pi_{ij} = p(i, j \in s)$ are assumed to be strictly positive. p_1 refers to first stage design. From every sampled cluster $i \in s$, a probability sample s_i of elements is drawn according to a fixed size design $p_2(.)$ with inclusion probabilities $\pi_{k/i} = p(k \in s_i/i \in s)$ and $\pi_{kl/i} = p(k, l \in s_i/i \in s)$ which are strictly positive. We let n_i be the size of s_i and assume invariance and independence of the second stage design. Let $t_i = g(x_i, Z_i) + \epsilon_i, i = 1, 2, \ldots, M$ be the i^{th} cluser total, where $g(x_i, Z_i)$ is a smooth function of x and Z.

Let $\hat{t}_s = \left[\hat{t}_i\right]_{i \in s}$ be the m dimension vector of $\hat{t}_i's$ obtained in the sample of clusters, where $\hat{t}_i = \sum_{k \in s_i} \frac{y_{ik}}{\pi_k}$ is the Horvitz-Thompson design estimate of ith cluster total. Define the spline model matrix X_c to contain bases that are functions of $\hat{t}_i$ and define the sub matrix $W_s = j \in s \left\{\frac{1}{\pi_j}\right\}$

Let ξ_1 denote a super population of clusters model and ξ_{11} denote a super population of cluster elements model. Define the semiparametric population estimator for $E_{\xi_1}(t_i)$ as

$$\begin{aligned} \hat{g}_i &= \hat{g}(x_i, Z_i) \\ &= \hat{\mu}_i + Z_i\hat{\beta} \end{aligned} \tag{3.37}$$

If the fits are based on penalized splines, we then have the estimators

$$\hat{\beta} = \left(Z_s^T S_{sps} Z_s\right)^{-1} Z_s^T S_{sps} \hat{t}_s \tag{3.38}$$

and

$$\hat{\mu}_i = S_{spsi}^T \left(\hat{t}_s - Z_s^T \hat{\beta}\right) \tag{3.39}$$

so that

$$\hat{g}_i = S_{spsi}^T\left(\hat{t}_s - Z_s^T\hat{\beta}\right) + Z_i\left(Z_s^T S_{sps} Z_s\right)^{-1} Z_s^T S_{sps}\hat{t}_s \tag{3.40}$$

If the fits are based on local polynomial, we then have the estimators

$$\hat{\beta} = \left(Z_s^T S_{lps} Z_s\right)^{-1} Z_s^T S_{lps}\hat{t}_s \tag{3.41}$$

and

$$\hat{\mu}_i = S_{lpsi}^T\left(\hat{t}_s - Z_s^T\hat{\beta}\right) \tag{3.42}$$

so that

$$\hat{g}_i = S_{lpsi}^T\left(\hat{t}_s - Z_s^T\hat{\beta}\right) + Z_i\left(Z_s^T S_{lps} Z_s\right)^{-1} Z_s^T S_{lps}\hat{t}_s \tag{3.43}$$

A Nadaraya Watson fit is obtained as in (3.43) but with the polynomial degree in S_{lpsi}^T being zero.

We propose a semiparametric model assisted model calibrated estimator of population total to be

$$\hat{y}_{sm2} = \sum_{i\in s} w_i\hat{t}_i \tag{3.44}$$

with w_i obtained by minimizing the chi square distance measure (2.1) subject to the constraints $\sum_{i\in s} w_i = M$ and $\sum_{i\in s} w_i\hat{g}_i = \sum_{i\in C}\hat{g}_i$. $d_i = \pi_i^{-1}$. $q_i's$ are known positive constants uncorrelated with the $d_i's$. We introduce the lagrange procedure in the minimization of equation (2.1) to obtain an equation of the form

$$l = \sum_{i\in s}\frac{(w_i - d_i)^2}{q_i d_i} - 2\lambda(\sum_{i\in s} w_i\hat{g}_i - \sum_{i\in C}\hat{g}_i) - 2v(\sum_{i\in s} w_i - M) \tag{3.45}$$

Where λ is the lagrange's multiplier and v is the penalty constant.

Differentiating equation (3.45) with respect to w_i and equating to zero we get

$$\begin{aligned}\frac{\partial l}{\partial w_i} &= \frac{2(w_i - d_i)}{q_i d_i} - 2\lambda\hat{g}_i - 2v \\ &= 0\end{aligned} \tag{3.46}$$

which gives

$$w_i = (\lambda\hat{g}_i + v)q_i d_i + d_i \tag{3.47}$$

Solving for λ and v we have

$$w_i = d_i + (M - \sum_{i\in s} d_i)\left\{\frac{d_iq_i}{\sum_{i\in s} d_iq_i} - \left\{\frac{q_id_i\left(\hat{g}_i - \frac{\sum_{i\in s} d_iq_i\hat{g}_i}{\sum_{i\in s} d_iq_i}\right)\left(1 - \frac{\sum_{i\in s} d_iq_i}{\sum_{i\in s} d_iq_i}\right)}{\sum_{i\in s} q_id_i\left(\hat{g}_i - \frac{\sum_{i\in s} d_iq_i\hat{g}_i}{\sum_{i\in s} d_iq_i}\right)^2}\right\}\right\}$$
$$+ \left\{\sum_{i\in C} \hat{g}_i - \sum_{i\in s} d_i\hat{g}_i\right\}\left\{\frac{q_id_i\left(\hat{g}_i - \frac{\sum_{i\in s} d_iq_i\hat{g}_i}{\sum_{i\in s} d_iq_i}\right)\left(1 - \frac{\sum_{i\in s} d_iq_i}{\sum_{i\in s} d_iq_i}\right)}{\sum_{i\in s} q_id_i\left(\hat{g}_i - \frac{\sum_{i\in s} d_iq_i\hat{g}_i}{\sum_{i\in s} d_iq_i}\right)^2}\right\} \quad (3.48)$$

Substituting w_i in (3.44) we have that

$$\hat{y}_{sm2} = \sum_{i\in s} d_i\hat{t}_i + (M - \sum_{i\in s} d_i)\left\{\frac{\sum_{i\in s} d_iq_i\hat{t}_i}{\sum_{i\in s} d_iq_i} - \hat{\beta}_{m2}\right\} + \left\{\sum_{i\in C} \hat{g}_i - \sum_{i\in s} d_i\hat{g}_i\right\}\hat{\beta}_{m2} \quad (3.49)$$

where $\hat{\beta}_{m2} = \left\{\frac{\sum_{i\in s} q_id_i\left(\hat{g}_i - \frac{\sum_{i\in s} d_iq_i\hat{g}_i}{\sum_{i\in s} d_iq_i}\right)\left(\hat{t}_i - \frac{\sum_{i\in s} d_iq_i\hat{t}_i}{\sum_{i\in s} d_iq_i}\right)}{\sum_{i\in s} q_id_i\left(\hat{g}_i - \frac{\sum_{i\in s} d_iq_i\hat{g}_i}{\sum_{i\in s} d_iq_i}\right)^2}\right\}$

Like in one stage sampling, the term $(M - \sum_{i\in s} d_i)\left\{\frac{\sum_{i\in s} d_iq_i\hat{t}_i}{\sum_{i\in s} d_iq_i} - \hat{\beta}_m\right\}$ is neglible. The proof cleary follow from the proof of theorem 1. We therefore rewrite the estimator as

$$\hat{y}_{sm2} = \sum_{i\in s} \frac{\hat{t}_i}{\pi_i} + \left\{\sum_{i\in C} \hat{g}_i - \sum_{i\in s} \frac{\hat{g}_i}{\pi_i}\right\}\hat{\beta}_{m2} \quad (3.50)$$

where $\hat{g}_i = Z_i\hat{\beta} + \hat{\mu}(x_i)$ and $d_i = (\pi_i)^{-1}$. with $\hat{\beta}$ and $\hat{\mu}$ computed as defined in equations (3.38) and (3.39) respectively.

We now derive the variance of this estimator. Suppose the sample comprises the whole population of clusters, then $\hat{y}_{sm2} = \sum_{i=1}^{m} \frac{\hat{t}_i}{\pi_i}$ which is the Horvitz-Thompson (HT) design based estimator. The variance of HT estimator under two stage sampling design can be written as the sum of two components

$$\begin{aligned} var_p(\hat{y}_{sm2}) &= V_1\left(E_{11}\left[\hat{y}_{sm2}\right]\right) + E_1\left(v_{11}\left[\hat{y}_{sm2}\right]\right) \\ &= \sum_{i\in C}\sum_{j\in C}(\pi_{ij} - \pi_i\pi_j)\frac{t_i}{\pi_i}\frac{t_j}{\pi_j} + \sum_{i\in C}\frac{V_i}{\pi_i} \end{aligned} \quad (3.51)$$

where $V_i = V_{11}(\hat{t}_i) = \sum_k^m \sum_l^m \left(\pi_{kl/i} - \pi_{k/i}\pi_{l/i}\right)\frac{y_{ik}}{\pi_{k/i}}\frac{y_{il}}{\pi_{l/i}}$, is the variance component at element level. (See Breidt and Opsomer, (2000) and Otieno et al(2007))

When $\hat{y}_{sm2}$ has the model component $\left\{\sum_{i=1}^{M} \hat{g}_i - \sum_{i=1}^{m} \frac{\hat{g}_i}{\pi_i}\right\} \hat{\beta}_{m2}$, its design variance becomes

$$\sum_{i\ \in C} \sum_{j\ \in C} (\pi_{ij} - \pi_i \pi_j) \frac{t_i - \hat{g}_i \hat{\beta}_{m2}}{\pi_i} \frac{t_j - \hat{g}_j \hat{\beta}_{m2}}{\pi_j} + \sum_{i\ \in C} \frac{V_i}{\pi_i} \tag{3.52}$$

Only the variance at the cluster level is affected by the model. V_i is non random due to invariance.

For two stage sampling, a corresponding estimator of finite population mean is therefore derived from the estimator of the total to give

$$\hat{\bar{y}}_{sm2} = \frac{1}{N} \sum_{i \in s} \frac{\hat{t}_i}{\pi_i} + \frac{1}{N} \left\{ \sum_{i \in C} \hat{g}_i - \sum_{i \in s} \frac{\hat{g}_i}{\pi_i} \right\} \hat{\beta}_{m2} \tag{3.53}$$

Dropping the regression coefficient(which resulted from model calibration) $\hat{\beta}_{m2}$ from $\hat{y}_{sm2}$ we have the corresponding internally calibrated estimator

$$\hat{y}_{reg2} = \sum_{i \in s} \frac{\hat{t}_i}{\pi_i} + \left\{ \sum_{i \in C} \hat{g}_i - \sum_{i \in s} \frac{\hat{g}_i}{\pi_i} \right\} \tag{3.54}$$

Now, when no part of the auxiliary information is to enter the estimation process parametrically, a nonparametric estimator would again be sufficient. Define the penalized splines nonparametric sample fit for $E_{\xi_1}(t_i)$ as

$$\hat{\mu}_{ti} = S_{spsi}^{T} \hat{t}_s \tag{3.55}$$

and a local polynomial nonparametric sample fit as

$$\hat{\mu}_{ti} = S_{lpsi}^{T} \hat{t}_s \tag{3.56}$$

minimizing the chi square distance measure(2.1) subject to the constraints $\sum_{i \in s} w_i = M$ and $\sum_{i \in s} w_i \hat{\mu}_{ti} = \sum_{i \in C} \hat{\mu}_{ti}$ and solving for w_i, v and λ then subsistuting to the nonparametric equivalent of (3.44)we then have that

$$\hat{y}_{np2} = \sum_{i \in s} \frac{\hat{t}_i}{\pi_i} + \left\{ \sum_{i \in C} \hat{\mu}_{ti} - \sum_{i \in s} \frac{\hat{\mu}_{ti}}{\pi_i} \right\} \hat{\beta}_{m2} \tag{3.57}$$

in which case

$$\hat{\beta}_{m2} = \left\{ \frac{\sum_{i \in s} q_i d_i \left(\hat{\mu}_{ti} - \frac{\sum_{i \in s} d_i q_i \hat{\mu}_{ti}}{\sum_{i \in s} d_i q_i} \right) \left(\hat{t}_i - \frac{\sum_{i \in s} d_i q_i \hat{t}_i}{\sum_{i \in s} d_i q_i} \right)}{\sum_{i \in s} q_i d_i \left(\hat{\mu}_{ti} - \frac{\sum_{i \in s} d_i q_i \hat{\mu}_{ti}}{\sum_{i \in s} d_i q_i} \right)^2} \right\} \tag{3.58}$$

Following the same approach in deriving variance as in the case of $\hat{y}_{sm2}$, we have variance of $\hat{y}_{np2}$ as

$$\sum_{i\,\in C}\sum_{j\,\in C}(\pi_{ij}-\pi_i\pi_j)\frac{t_i-\hat{\mu}_{ti}\hat{\beta}_{m2}}{\pi_i}\frac{t_j-\hat{\mu}_{tj}\hat{\beta}_{m2}}{\pi_j}+\sum_{i\,\in C}\frac{V_i}{\pi_i} \tag{3.59}$$

where $V_i = V_{11}(\hat{t}_i) = \sum_k^m \sum_l^m \left(\pi_{kl/i} - \pi_{k/i}\pi_{l/i}\right) \frac{y_k}{\pi_{k/i}} \frac{y_l}{\pi_{l/i}}$

The cosrresponding nonparametric estimator of the population mean is therefore

$$\hat{\bar{y}}_{np2} = \frac{1}{N}\sum_{i\in s}\frac{\hat{t}_i}{\pi_i} + \frac{1}{N}\left\{\sum_{i\in C}\hat{\mu}_{ti} - \sum_{i\in s}\frac{\hat{\mu}_{ti}}{\pi_i}\right\}\hat{\beta}_{m2} \tag{3.60}$$

while a corresponding internally calibrated esimator for population total is

$$\hat{y}_{gen2} = \sum_{i\in s}\frac{\hat{t}_i}{\pi_i} + \left\{\sum_{i\in C}\hat{\mu}_{ti} - \sum_{i\in s}\frac{\hat{\mu}_{ti}}{\pi_i}\right\} \tag{3.61}$$

3.3.2 Auxilliary Information at Both Element and Cluster Level

Now, consider the case where there is also auxilliary information known at element level such that for each element in the ith cluster, a variable x_i that is to be used in noparametric estimation and a categorical vector Z_i are available. Suppose not all elements in a given cluster are available and have to be imputed. We use model calibration in the estimation of the cluster total to obtain.

$$\hat{t}_{ism} = \sum_{k\in s_i} d_{k/i}y_{ik} + \left\{\sum_{k=1}^{N_i}\hat{g}_{ik} - \sum_{k\in s_i} d_{k/i}\hat{g}_{ik}\right\}\hat{\beta}_{mc} \tag{3.62}$$

where $\hat{\beta}_{mc} = \left\{ \frac{\sum_{k\in s_i} q_{ik}d_{k/i}\left(\hat{g}_{ik} - \frac{\sum_{k\in s_i} d_{k/i}q_{ik}\hat{g}_{ik}}{\sum_{k\in s_i} d_{k/i}q_{ik}}\right)\left(y_{ik} - \frac{\sum_{k\in s_i} d_{k/i}q_{ik}y_{ik}}{\sum_{k\in s_i} d_{k/i}q_{ik}}\right)}{\sum_{k\in s_i} q_{ik}d_{k/i}\left(\hat{g}_{ik} - \frac{\sum_{k\in s_i} d_{k/i}q_{ik}\hat{g}_{ik}}{\sum_{k\in s_i} d_{k/i}q_{ik}}\right)^2} \right\}$

and $\hat{g}_{ik} = Z_{ik}\hat{\beta} + \hat{\mu}(x_{ik}) = E_{\xi_{11}}(y_{ik})$ with $\hat{\beta}$ derived as before but using values from the cluster.

Accordingly, we have the estimator for population total as

$$\hat{y}_{ssm2} = \sum_{i\in s}\frac{\hat{t}_{ism}}{\pi_i} + \left\{\sum_{i\in C}\hat{g}_i - \sum_{i\in s}\frac{\hat{g}_i}{\pi_i}\right\}\hat{\beta}_{m2} \tag{3.63}$$

where in $\hat{\beta}_{m2} = \left\{ \frac{\sum_{i\in s} q_i d_i \left(\hat{g}_i - \frac{\sum_{i\in s} d_i q_i \hat{g}_i}{\sum_{i\in s} d_i q_i} \right) \left(\hat{t}_{ism} - \frac{\sum_{i\in s} d_i q_i \hat{t}_{ism}}{\sum_{i\in s} d_i q_i} \right)}{\sum_{i\in s} q_i d_i \left(\hat{g}_i - \frac{\sum_{i\in s} d_i q_i \hat{g}_i}{\sum_{i\in s} d_i q_i} \right)^2} \right\}$
A corresponging internally calibrated estimator will therefore be

$$\hat{y}_{regreg2} = \sum_{i\in s} \frac{\hat{t}_{reg}}{\pi_i} + \left\{ \sum_{i\in C} \hat{g}_i - \sum_{i\in s} \frac{\hat{g}_i}{\pi_i} \right\} \tag{3.64}$$

where now

$$\hat{t}_{reg} = \sum_{k\in s_i} d_{k/i} y_{ik} + \left\{ \sum_{k=1}^{N_i} \hat{g}_{ik} - \sum_{k\in s_i} d_{k/i} \hat{g}_{ik} \right\} \tag{3.65}$$

Again, if the sample comprises the whole population of clusters, then $\hat{y}_{ssm2} = \sum_{i=1}^{m} \frac{\hat{t}_{ism}}{\pi_i}$ which is the Horvitz-Thompson (HT) design based estimator. We then have that

$$\begin{aligned} var_p(\hat{y}_{ssm2}) &= V_1 \left(E_{11} \left[\hat{y}_{ssm2} \right] \right) + E_1 \left(v_{11} \left[\hat{y}_{ssm2} \right] \right) && (3.66) \\ &= \sum_{i\in C} \sum_{j\in C} (\pi_{ij} - \pi_i \pi_j) \frac{t_{ism}}{\pi_i} \frac{t_{jsm}}{\pi_j} + \sum_{i\in C} \frac{V_i}{\pi_i} && (3.67) \end{aligned}$$

where the variance component at element level
$V_i = V_{11}(\hat{t}_{ism}) = \sum_k^m \sum_l^m \left(\pi_{kl/i} - \pi_{k/i} \pi_{l/i} \right) \frac{y_{ik} - \hat{g}_{ik} \hat{\beta}_{mc}}{\pi_{k/i}} \frac{y_{il} - \hat{g}_{il} \hat{\beta}_{mc}}{\pi_{l/i}}$ due to the model component $\left\{ \sum_{k=1}^{N_i} \hat{g}_{ik} - \sum_{k\in s_i} d_{k/i} \hat{g}_{ik} \right\} \hat{\beta}_{mc}$. When $\hat{y}_{ssm2}$ has the model component $\left\{ \sum_{i=1}^{M} \hat{g}_i - \sum_{i=1}^{m} \frac{\hat{g}_i}{\pi_i} \right\} \hat{\beta}_{m2}$, its design variance becomes

$$\sum_{i\,\in C} \sum_{j\,\in C} \left(\pi_{ij} - \pi_i \pi_j \right) \frac{t_{ism} - \hat{g}_i \hat{\beta}_{m2}}{\pi_i} \frac{t_{jsm} - \hat{g}_j \hat{\beta}_{m2}}{\pi_j} + \sum_{i\,\in C} \frac{V_i}{\pi_i} \tag{3.68}$$

When the sample of clusters comprise the whole poipulation of clusters and the element samples comprise all the elements in a cluster then $\hat{y}_{sm2} = \hat{y}_{ssm2}$. It is therefore enough to find the asymptotic properties of $\hat{y}_{sm2}$ which would clearly apply to $\hat{y}_{ssm2}$.

We obtain a nonparametric estimator by letting $\hat{\mu}_{tik} = E_{\xi_{11}}(y_{ik})$ so that

$$\hat{t}_{inp} = \sum_{k\in s_i} d_{k/i} y_{ik} + \left\{ \sum_{k=1}^{N_i} \hat{\mu}_{tik} - \sum_{k\in s_i} d_{k/i} \hat{\mu}_{tik} \right\} \hat{\beta}_{mc} \tag{3.69}$$

where $\hat{\beta}_{mc} = \left\{ \frac{\sum_{k\in s_i} q_{ik} d_{k/i}\left(\hat{\mu}_{tik} - \frac{\sum_{k\in s_i} d_{k/i} q_{ik}\hat{\mu}_{tik}}{\sum_{k\in s_i} d_{k/i} q_{ik}}\right)\left(y_{ik} - \frac{\sum_{k\in s_i} d_{k/i} q_{ik} y_{ik}}{\sum_{k\in s_i} d_{k/i} q_{ik}}\right)}{\sum_{k\in s_i} q_{ik} d_{k/i}\left(\hat{\mu}_{tik} - \frac{\sum_{k\in s_i} d_{k/i} q_{ik}\hat{\mu}_{tik}}{\sum_{k\in s_i} d_{k/i} q_{ik}}\right)^2} \right\}$
So that we have the estimator for population total as

$$\hat{y}_{nnp2} = \sum_{i\in s}\frac{\hat{t}_{inp}}{\pi_i} + \left\{\sum_{i\in C}\hat{\mu}_{ti} - \sum_{i\in s}\frac{\hat{\mu}_{ti}}{\pi_i}\right\}\hat{\beta}_{m2} \tag{3.70}$$

Where now $\hat{\beta}_{m2} = \left\{ \frac{\sum_{i\in s} q_i d_i\left(\hat{\mu}_{ti} - \frac{\sum_{i\in s} d_i q_i \hat{\mu}_{ti}}{\sum_{i\in s} d_i q_i}\right)\left(\hat{t}_{inp} - \frac{\sum_{i\in s} d_i q_i \hat{t}_{inp}}{\sum_{i\in s} d_i q_i}\right)}{\sum_{i\in s} q_i d_i\left(\hat{\mu}_{ti} - \frac{\sum_{i\in s} d_i q_i \hat{\mu}_{ti}}{\sum_{i\in s} d_i q_i}\right)^2} \right\}$
A corresponging internally calibrated estimator is therefore

$$\hat{y}_{gengen2} = \sum_{i\in s}\frac{\hat{t}_{gen}}{\pi_i} + \left\{\sum_{i\in C}\hat{\mu}_{ti} - \sum_{i\in s}\frac{\hat{\mu}_{ti}}{\pi_i}\right\} \tag{3.71}$$

where now

$$\hat{t}_{gen} = \sum_{k\in s_i} d_{k/i} y_{ik} + \left\{\sum_{k=1}^{N_i}\hat{\mu}_{tik} - \sum_{k\in s_i} d_{k/i}\hat{\mu}_{tik}\right\} \tag{3.72}$$

3.4 Theoretical Properties Under Two Stage Sampling

In this section, we show that $\hat{y}_{sm2} = \sum_{i=1}^{m}\frac{\hat{t}_i}{\pi_i} + \left\{\sum_{i=1}^{M}\hat{g}_i - \sum_{i=1}^{m}\frac{\hat{g}_i}{\pi_i}\right\}\hat{\beta}_{m2}$ where $\hat{g}_i = Z_i\hat{\beta} + \hat{\mu}(x_i) + \epsilon_i$ is design unbiased, consistent and asymptotic normaly distributed. These properties hold under the following mild assumptions. These assumptions are similar to those in (3.2.1) with little modification to fit two stage sampling. They therefore have similar relevance to the proofs.

3.4.1 Assumptions

1. We assume that there is a sequence of finite populations indexed by ρ each of size N_ρ but which we simply write N for ease of representation.

2. As $\rho \to \infty, N, n, M, m, N_i, n_i \to \infty$, the number of knots $k \to \infty$ while bandwidth $h \to 0$.

3. For each ρ, the x_i, for $i = 1, 2, ..., M$ are indepedent and identically distributed $F(x) = \int_{-\infty}^{x} f(t)dt$ where $f(.)$ is a density with compact support $[a_x, b_x]$ and $f(x) > 0$ for all $x \in [a_x, b_x]$. The Z_i have bounded suport.

4. For each ρ the x_i and Z_i are considered fixed with respect to the model ξ_1 while the errors ϵ_{i1} are indepedent and have mean zero, variance $var(x_i, Z_i)$ and compact support, uniformly for each ρ.

5. For each ρ the x_{ik} and Z_{ik} are considered fixed with respect to the model ξ_{11} while the errors ϵ_{i11} are indepedent and have mean zero, variance $var(x_{ik}, Z_{ik})$ and compact support, uniformly for each ρ.

6. The sampling design is regular so that the inclusion probabilities are independent of response measurements and satisfies the following conditions
a) $\max_{i \in s} \frac{m}{M\pi_i} = 0(1)$, and $\max_{k \in s_i} \frac{n_i}{N_i \pi_{k/i}} = 0(1)$
b) $\sum_{i \in s} \frac{g_i}{\pi_j} - \sum_{i=1}^{M} g_i = 0_p(Mm^{-\frac{1}{2}})$ and $\sum_{k \in s_i} \frac{g_{ik}}{\pi_{k/i}} - \sum_{k=1}^{N_i} g_{ik} = 0_p(N_i n_i^{-\frac{1}{2}})$.

3.4.2 Asymptotic Design Unbiasedness

Let E_{p_1} be design expectation and E_{ξ_1} model based expectation. We need to show that $\{E_{p_1}(\hat{y}_{sm2})\} = Y_t$. We note that $\hat{t}_i$ is a Horvitz Thompson design estimator which is unbiased fot t_i . The proof follow from assumptions 1-6, the fact that $E_{P_1}(I_i) = \pi_i$ and that with respect to design expectation $\hat{g}_i$ and $\hat{\beta}_{m2}$ are treated as constants.

Now,

$$\{E_{p_1}(\hat{y}_{sm2})\} = E_{p_1}\left\{\sum_{i \in s} \frac{\hat{t}_i}{\pi_i} + \left\{\sum_{i \in C} \hat{g}_i - \sum_{i \in s} \frac{\hat{g}_i}{\pi_i}\right\} \hat{\beta}_{m2}\right\}$$

$$
\begin{aligned}
&= E_{p_1}\left\{\sum_{i\in C}\frac{\hat{t}_i I_i}{\pi_i} + \left\{\sum_{i\in C}\hat{g}_i - \sum_{i\in C}\frac{\hat{g}_i I_i}{\pi_i}\right\}\hat{\beta}_{m2}\right\} \\
&= \left\{\sum_{i\in C}\frac{E_{p_1}\hat{t}_i I_i}{\pi_i} + \left\{\sum_{i\in C}E_{p_1}\hat{g}_i\hat{\beta}_{m2} - \sum_{i\in C}\frac{E_{p_1}(\hat{g}_i I_i\hat{\beta}_{m2})}{\pi_i}\right\}\right\} \\
&= \left\{\sum_{i\in C}\frac{t_i}{1} + \left\{\sum_{i\in C}\hat{g}_i\hat{\beta}_{m2} - \sum_{i\in C}\frac{\hat{g}_i\hat{\beta}_{m2}}{1}\right\}\right\} \qquad (3.73)
\end{aligned}
$$

Thus we have, $\sum_{i\in C} t_i = Y_t$.

3.4.3 Model Bias Reduction

We show that $\hat{y}_{sm2} = \sum_{i=1}^{m}\frac{\hat{t}_i}{\pi_i} + \left\{\sum_{i=1}^{M}\hat{g}_i - \sum_{i=1}^{m}\frac{\hat{g}_i}{\pi_i}\right\}\hat{\beta}_{m2}$ has reduced model bias. $\hat{\beta}_{m2}$ is an estimate of the change in Y_t when g_i is increased by a unit. If $\sum_{i\in s}\frac{\hat{g}_i}{\pi_i}$ is below average, we should expect the population total Y_t to be below average by an amount $\left\{\sum_{i\in C}\hat{g}_i - \sum_{i\in s}\frac{\hat{g}_i}{\pi_i}\right\}\hat{\beta}_{m2}$ due to regression of $\hat{t}_i$ on $\hat{g}_i$. Cochran(1997). Again, the estimate $\hat{g}_i$ need not be free from bias. If $\hat{g}_i - \hat{t}_i = D$, so that the estimate is perfect except for a constant bias D, then with $\hat{\beta}_m = 1$ the regression estimate becomes

$$\sum_{i\ \in s}\frac{\hat{t}_i}{\pi_i} + \left\{\sum_{i\in C}\hat{g}_i - \sum_{i\in s}\frac{\hat{g}_i}{\pi_i}\right\} = \sum_{i\in C}\hat{g}_i + \left\{\sum_{i\in s}\frac{\hat{t}_i}{\pi_i} - \sum_{i\in s}\frac{\hat{g}_i}{\pi_i}\right\} \qquad (3.74)$$

This regression estimate is consistent in the sense that when the sample comprises the whole population, then $\sum_{i\in C}\hat{g}_i = \sum_{i\in s}\frac{\hat{g}_i}{\pi_i}$ and the regression estimate reduces to $\sum_{i\in s}\frac{\hat{t}_i}{\pi_i}$. See Firth and Bennett (2006). Again, establishing a CLT for $\hat{y}_{sm2}$, which is a generalized diference estimator is essentially the same as establishing a CLT for Horvitz-Thompson estimator.

3.4.4 Design Consistency

Under asumptions 1-6, the chebycheve's inequality and a sequence of the estimates $\hat{y}_{sm2\rho}$ but which we simply write $\hat{y}_{sm2}$, We have that $pr[|\hat{y}_{sm2} - Y_t| > \epsilon] \leq$

$E_{P_1} \frac{|\hat{y}_{sm2} - Y_t|^2}{\epsilon^2}$

but since $\hat{y}_{sm2}$ is unbiased for Y_t, then the mean squared error is consistently estimated by $var(\hat{y_{sm2}})$, so that $pr[|\hat{y}_{sm2} - Y_t| > \epsilon] \leq \frac{var\{\hat{y}_{sm2}\}}{\epsilon^2}$ and

$$\lim_{\rho \to \infty} pr[|E_{p_1}\hat{y}_{sm2} - Y_t| > \epsilon] \leq \lim_{\rho \to \infty} \frac{var\,\{\hat{y}_{sm2}\}}{\epsilon^2}$$

We see that,

$$\begin{aligned}
\lim_{\rho \to \infty} \frac{var\,\{\hat{y}_{sm2}\}}{\epsilon^2} &= \lim_{\rho \to \infty} E_{p_1} \sum_{i=1}^{m} \sum_{j=1}^{m} (\frac{t_i - \hat{g}_i \hat{\beta}_{m2}}{\pi_i})(\frac{t_j - \hat{g}_j \hat{\beta}_{m2}}{\pi_j})(\frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}}) \frac{1}{\epsilon^2} \\
&+ \lim_{\rho \to \infty} E_{p_1} \sum_{i=1}^{M} \left\{ \sum_{k}^{m} \sum_{l}^{m} \left(\pi_{kl/i} - \pi_{k/i} \pi_{l/i} \right) \frac{y_k}{\pi_{k/i}} \frac{y_l}{\pi_{l/i}} \right\} \frac{1}{\epsilon^2 \pi_i} \\
&= \lim_{\rho \to \infty} E_{p_1} \sum_{i=1}^{M} \sum_{j=1}^{M} (\frac{t_i - \hat{g}_i \hat{\beta}_{m2}}{\pi_i})(\frac{t_j - \hat{g}_j \hat{\beta}_{m2}}{\pi_j})(\frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}}) \frac{I_i I_j}{\epsilon^2} \\
&+ \lim_{\rho \to \infty} E_{p_1} \sum_{k}^{M} \sum_{l}^{M} \left\{ \left(\pi_{kl/i} - \pi_{k/i} \pi_{l/i} \right) \frac{y_k}{\pi_{k/i}} \frac{y_l}{\pi_{l/i}} \right\} \frac{I_i I_j}{\epsilon^2 \pi_i} = 0 \qquad (3.75)
\end{aligned}$$

Since $E_{p_1}(\pi_{ij}) = \pi_i \pi_j, E_{p_1}(\pi_i \pi_j) = \pi_i \pi_j, E_{p_1}(\pi_i) = \pi_i$ and $E_{p_1}(I_i I_j) = \pi_{ij} \leq \pi_i \pi_j$ Which reduces the brackets of probabilities to zero. We must therefore also have that

$$\lim_{\rho \to \infty} pr[|E_{p_1}\hat{y}_{sm2} - Y_t| > \epsilon] \to 0$$

That is ,

$$\hat{y}_{sm2} \xrightarrow{p} Y_t$$

3.4.5 Asymptotic Normality

Theorem 4: Let g_i be the population fit assumed known for every population element and

$$y^*_{sm2} = \sum_{i=1}^{m} \frac{\hat{t}_i}{\pi_i} + \left\{ \sum_{i=1}^{M} g_i - \sum_{i=1}^{m} \frac{g_i}{\pi_i} \right\} \beta^*_{m2} \tag{3.76}$$

where

$\beta^*_{m2} = \frac{\sum_{j=1}^{M} \frac{1}{\pi_i} q_i (g_i - \bar{g})(\hat{t}_i - \bar{t})}{\sum_{i=1}^{M} \frac{1}{\pi_i} q_i (g_i - \bar{g})^2}$ and $\bar{g} = \sum_{i=1}^{M} g_i$

Then, y^*_{sm2} is asymptotic normal in the sense that $\frac{(y^*_{sm2} - Y_t)}{var^{1/2}(y^*_{sm2})} \xrightarrow{d} N(0,1)$ as $\rho \to \infty$

Proof: The proof follow from the proof of theorem 2, but in this case the sampling units are the clusters.

Theorem 5: Let y^*_{sm2} be as defined in theorem 4.

Then, $\frac{(y^*_{sm2} - Y_t)}{var^{1/2}(y^*_{sm2})} \xrightarrow{d} N(0,1)$ as $\rho \to \infty$ implies that $\frac{(\hat{y}_{sm2} - Y_t)}{var^{1/2}(\hat{y}_{sm2})} \xrightarrow{d} N(0,1)$

Where,

$$var(\hat{y}_{sm2}) = \sum_{i=1}^{m} \sum_{j=1}^{m} \left(\frac{\hat{t}_i - \hat{g}_i \hat{\beta}_{m2}}{\pi_i} \right) \left(\frac{\hat{t}_j - \hat{g}_j \hat{\beta}_{m2}}{\pi_j} \right) \left(\frac{\pi_{ij} - \pi_i \pi_j}{\pi_{ij}} \right) \tag{3.77}$$

proof: We need to show that $(\hat{y}_{sm2} - Y_t)$ converges to $(y^*_{sm2} - Y_t)$ in distribution. This would imply that $\hat{y}_{sm2}$ inherits limiting distributional properties of y^*_{sm2}. This, coupled by theorem 4 would proof the above.

Now,

$(\hat{y}_{sm2} - Y_t) = \sum_{i=1}^{M} \frac{\hat{t}_i I_i}{\pi_i} + \sum_{i=1}^{M} \hat{g}_i \hat{\beta}_{m2} - \sum_{i=1}^{M} \frac{\hat{g}_i I_i \hat{\beta}_{m2}}{\pi_i} - \sum_{i=1}^{M} \hat{t}_i$

And $(y^*_{sm2} - Y_t) = \sum_{i=1}^{M} \frac{\hat{t}_i I_i}{\pi_i} + \sum_{i=1}^{M} g_i \beta^*_{m2} - \sum_{i=1}^{M} \frac{g_i I_i \beta^*_{m2}}{\pi_i} - \sum_{i=1}^{M} \hat{t}_i$.

Clearly,

$$\hat{y}_{sm2} - y^*_{sm2} = \sum_{i=1}^{M} \left(\hat{g}_i \hat{\beta}_{m2} - g_i \beta^*_{m2} \right) \left(1 - \frac{I_i}{\pi_i} \right) \tag{3.78}$$

Taking limits of the expectation,we have

$$\lim_{\rho \to \infty} E_{P_1} \left\{ \hat{y}_{sm2} - y^*_{sm2} \right\} = \lim_{\rho \to \infty} E_{P_1} \left\{ \sum_{i=1}^{M} \left(\hat{g}_i \hat{\beta}_{m2} - g_i \beta^*_{m2} \right) \left(1 - \frac{I_i}{\pi_i} \right) \right\} \tag{3.79}$$

It can be seen that the design expectation of $\hat{y}_{sm2} - y^*_{sm2}$ approaches zero since design expectation of I_i is π_i. This is convergence in mean which implies convergence in probability and convergence in distribution.

The right hand term in (3.78) can be written as

$$\left(\hat{\beta}_{m2}-\beta^*_{m2}\right)\sum_{i\,=1}^{M}\hat{g}_i\left(1-\frac{I_i}{\pi_i}\right)+\beta^*_{m2}\sum_{i\,=1}^{M}\left(\hat{g}_i-g_i\right)\left(1-\frac{I_i}{\pi_i}\right) \tag{3.80}$$

Now, $\hat{g}_i - g_i = O_p(1)$ from lemma 4 in Breidt and Opsomer(2000), $\hat{\beta}_{m2} - \beta^*_{m2} = O_p(1)$ from an argument similar to that of lemma 4 in Montanari and Ranalli(2003). $\sum_{i=1}^{M}\hat{g}_i\left(1-\frac{I_i}{\pi_i}\right) = O_p(Mm^{-1/2})$ and $\sum_{i=1}^{M}\left(\hat{g}_i-g_i\right)\left(1-\frac{I_i}{\pi_i}\right) = O_p(Mm^{-1/2})$ from the proof of theorem 2 in Breidt and Opsomer(2000). Therefore the term in equation(3.80) is of order $O_p(Mm^{-1/2})$.

The estimators $\hat{y}_{ssm2}$, $\hat{y}_{np2}$ and $\hat{y}_{nnp2}$ have the same structure as $\hat{y}_{sm2}$, with the only difference being how the fitted values are obtained. Its therefore clear that they posses the theoretical properties; design unbiasedness, consistency and asymptotic normality with the proofs obtained in quite a similar manner as for $\hat{y}_{sm2}$.

CHAPTER FOUR

4.0 EMPIRICAL ANALYSIS

4.1 Empirical Analysis Under One Stage Sampling

For the analysis, using R program we simulated a population of independent and identically distributed variable x using uniform (0.1). For nonparametric estimation, the dependent population values y were generated from the following mean functions which are similar to but not exactly the same as those used by Breidt and Opsomer(2000). This is so that we could compare the performance of the model calibrated estimator that we have proposed and the internaly calibrated estimator they proposed. This comparison is done by generating our own data and employing the estimators to see which yields better results. We note that it would be erroneous to simply pick their results and compare with our results even when the mean functions below are the same.This is because the generated data depends on the random numbers generated at each of the replication. Still, simulation software actually generate pseudo random numbers and not random numbers. These pseudo random numbers are generated using a congruential generator coded in a software and which differs from sofware to software. Simply picking results from two authors and directly comparing them assumes the same software(and hence the same random number generator) and same set of pseudo random numbers at every simulation replication are used by the authors. This is not feasible.In fact it is possible with many softwares for a data analyst to create his or her own generator to suit his or her analysis. It is for these reasons that we could not compare our results directly with theirs but instead use their estimators with our own data.

1. linear $2 + 5x$
2. quadratic $(2 + 5x)^2$
3. bump $(2 + 5x) + exp(-200(2 + 5x)^2)$
4. exponential $exp(-8x)$
5. cycle1 $2 + sin(2\pi x)$
6. cycle2 $2 + sin(8\pi x)$

For semiparametric estimation, the dependent population values y were generated from the following mean functions which are similar to those used by Breidt and Opsomer(2000) but we have added a categorical matrix Z.

1. linear $Z\beta' + 2 + 5x$
2. quadratic $Z\beta' + (2 + 5x)^2$
3. bump $Z\beta' + (2 + 5x) + exp(-200(2 + 5x)^2)$
4. exponential $Z\beta' + exp(-8x)$
5. cycle1 $Z\beta' + 2 + sin(2\pi x)$
6. cycle2 $Z\beta' + 2 + sin(8\pi x)$

Z is the matrix (Z_1, Z_2, Z_3),where Z_1 is a matrix of 2s with dimension N, the population size. Z_2 is a matrix of alternating 3s,4s and 5s with dimension N, while Z_3 is a matrix of alternating 6s,7s and 8s with dimension N. $\beta = (1, 2, 3)$ is the vector of coefficients.

We consider the following estimators in the analysis.

1. Horvitz Thompson, $\hat{y}_{ht}$ with inclusion probability $\pi_i = \frac{n}{N}$
2. Model Calibrated Model Assisted Estimator $\hat{y}_{sm}$(3.15) proposed, for which we considered three cases; $\hat{y}_{sm}$ based on penalized splines which from now on will be denoted by $\hat{y}_{smsp}$ and in which $\hat{g}_i$ is as defined in (3.8), $\hat{y}_{sm}$ based on local polynomial which we denote by $\hat{y}_{smlp}$ in which $\hat{g}_i$ is as defined in (3.17) and $\hat{y}_{sm}$ based on Nadaraya Watson kernel smoothing which we denote by $\hat{y}_{smnw}$ in which $\hat{g}_i$ is as defined in (3.17) but with polynomial degree zero.
3. Internally Calibrated Model Assisted Estimator $\hat{y}_{reg}$,(2.20), for which we consider the three cases; $\hat{y}_{reg}$ based on penalized splines which we denote by $\hat{y}_{regsp}$ in which $\hat{g}_i$ is as defined in (3.8), $\hat{y}_{reg}$ based on local polynomial which we denote by $\hat{y}_{reglp}$ in which $\hat{g}_i$ is as defined in (3.17)and $\hat{y}_{reg}$ based on Nadaraya Watson kernel smoothing which we denote by $\hat{y}_{regnw}$ in which $\hat{g}_i$ is as defined in (3.17) but with polynomial degree zero.

The first is a design based estimator while the others are semiparametric estimators which are model assisted. For the Nadaraya Watson kernel smoothing we consider equal probability sampling. For model calibrated estimators, the weight q_i is set to 1 for ease of computations. We used the standard epernecknikov kernel $K(u) = 3/4(1 - u^2), u \leq 1$ for the kernel based estimators. A bandwidth of 0.25 was used for the kernel based estimators. This was based on the ad hoc rule of $1/4^{th}$ of the data rage. For the penalised spline estimators, the knots were placed at equidistance. That is, $0.2, 0.4, 0.6, 0.8,$ thus dividing the data rage into five equal segments.

The populations were of size N=300. Samples of size n=30 were generated by simple random sampling. The sample size is ten percent of the population. For each population of x and mean function, 100 replicate samples were generated and the estimates calculated. The population was kept fixed during these 100 replicates in order to evaluate the design averanged performance of the estimators. This enabled estimation of design bias, design variance and design mean squared error. Trials involving 500 and 1000 replicate samples yielded same results as for 100 replicates.

The performance of any estimator say y_{est} in $\hat{y}_{ht}$, $\hat{y}_{smsp}$, $\hat{y}_{smlp}$, $\hat{y}_{smnw}$, $\hat{y}_{regsp}$, $\hat{y}_{reglp}$, $\hat{y}_{regnw}$ was evaluated using its relative bias R_B and relative efficiency R_E defined by

$$R_B = \frac{\sum_{r=1}^{R} (y_{est} - Y_t)}{R * Y_t} \tag{4.1}$$

where R is the replicate number of samples and relative efficiency

$$R_E = \frac{MSE(y_{est})}{MSE(y_{ht})} \tag{4.2}$$

where y_{est} was calculated from the R^{th} simulated sample.

The $\hat{y}_{ht}$ estimator was used as the baseline comparison. Large values of relative efficiencies,($R_E > 1$ or Inverse $R_E < 1$) represent higher efficiency for the design estimator $\hat{y}_{ht}$ over the estimator y_{est} that its being compared with.

We also carried out a Sensitivity Analysis by looking at the effects that ignoring a variable in the categorical matrix would have on the estimators.

For nonparametric estimation, we compared the performance of the Horvitz-Thompson estimator with the model calibrated model assisted estimator $\hat{y}_{np}$,(3.22), for which we consider the three cases; $\hat{y}_{np}$ based on penalized splines which we denote as $\hat{y}_{npsp}$ in which $\hat{\mu}_i$ is as defined in (3.19), $\hat{y}_{np}$ based on local polynomial which we denote by $\hat{y}_{nplp}$ in which $\hat{\mu}_i$ is as defined in (3.20) and $\hat{y}_{np}$ based on Nadaraya Watson kernel which we denote by $\hat{y}_{npnw}$ in which $\hat{\mu}_i$ is as defined in (3.20) but with polynomial degree zero. We also compared the performance of the Horvitz-Thompson estimator with the internally calibrated model assisted estimator $\hat{y}_{gen}$,(2.16), for which we consider the three cases; $\hat{y}_{gen}$ based on penalized splines which we denote as $\hat{y}_{gensp}$ in which $\hat{\mu}_i$ is as defined in (3.19), $\hat{y}_{gen}$ based on local polynomial which we denote by $\hat{y}_{genlp}$ in which $\hat{\mu}_i$ is as defined in (3.20)and $\hat{y}_{gen}$ based on Nadaraya Watson kernel which we denote by $\hat{y}_{gennw}$ in which $\hat{\mu}_i$ is as defined in (3.20) but with degree zero.

4.1.1 Normality Test

Before we could compare the performance of our proposed model calibrated estimators with the design and internally calibrated estimtors, we carried out a Shapiro-Wilk test for normality and obtained the p-values in table (4.1). A p-value greater than the set α significance level means normality is established. At $\alpha = 0.05$, we can see that the proposed estimators are normal. A sample of normal graphs are provided in appendix 2.

Table 4.1: Shapiro-Wilk p-values(one stage)

	$\hat{y}_{smsp}$	$\hat{y}_{smlp}$	$\hat{y}_{smnw}$	$\hat{y}_{npsp}$	$\hat{y}_{nplp}$	$\hat{y}_{npnw}$
Linear	0.763	0.799	0.731	0.439	0.499	0.428
Quadratic	0.872	0.826	0.694	0.158	0.165	0.157
Bump	0.781	0.679	0.505	0.601	0.631	0.589
Exponential	0.523	0.511	0.511	0.692	0.692	0.692
Cycle 1	0.471	0.456	0.461	0.863	0.863	0.863
Cycle 2	0.637	0.637	0.643	0.613	0.613	0.613

4.1.2 Consistency Test

We carried out a consistency test by varying the the sample size n_ρ,population size N_ρ for each population $\rho = 1, 2, 3$ and obtaining the diference between the estimate and actual total Y_t. The estimated total is taken to be the average from the R replicates. The results in the table (4.2) show the estmators are consistent in the sense that the diferences decrease consistently with an increase in population and sample sizes.

Table 4.2: Absolute Diferences for One Stage Sampling

	$n_1 = 30, N_1 = 300$	$n_2 = 450, N_2 = 900$	$n_3 = 1350, N_3 = 1800$
Mean Function	$\hat{y}_{smsp} - Y_t$	$\hat{y}_{smsp} - Y_t$	$\hat{y}_{smsp} - Y_t$
Linear	5.75	5.74	5.27
Quadratic	18.94	18.66	18.67
Bump	12.24	5.33	3.57
Exponential	10.57	9.15	5.98
Cycle 1	4.58	5.06	4.11
Cycle 2	7.24	5.63	5.41
Mean Function	$\hat{y}_{smlp} - Y_t$	$\hat{y}_{smlp} - Y_t$	$\hat{y}_{smlp} - Y_t$
Linear	5.76	5.54	5.31
Quadratic	33.13	29.82	26.84
Bump	7.61	7.67	5.85
Exponential	10.52	10.01	8.51
Cycle 1	14.87	13.46	10.41
Cycle 2	25.64	16.87	16.43
Mean Function	$\hat{y}_{smnw} - Y_t$	$\hat{y}_{smnw} - Y_t$	$\hat{y}_{smnw} - Y_t$
Linear	13.63	12.99	12.17
Quadratic	40.26	41.72	34.04
Bump	17.81	15.55	13.505
Exponential	30.23	27.51	23.95
Cycle 1	30.67	30.56	26.39
Cycle 2	36.73	35.37	25.66
Mean Function	$\hat{y}_{npsp} - Y_t$	$\hat{y}_{npsp} - Y_t$	$\hat{y}_{npsp} - Y_t$
Linear	4.89	4.71	3.62
Quadratic	13.62	13.63	10.62
Bump	11.24	5.60	2.94
Exponential	8.97	8.45	4.18
Cycle 1	4.16	4.33	3.52
Cycle 2	6.53	5.11	4.21
Mean Function	$\hat{y}_{nplp} - Y_t$	$\hat{y}_{nplp} - Y_t$	$\hat{y}_{nplp} - Y_t$
Linear	5.16	4.92	4.33
Quadratic	29.29	29.31	23.39
Bump	7.22	6.34	5.12
Exponential	9.78	8.71	8.81
Cycle 1	12.94	12.41	8.29
Cycle 2	19.77	16.02	14.53
Mean Function	$\hat{y}_{npnw} - Y_t$	$\hat{y}_{npnw} - Y_t$	$\hat{y}_{npnw} - Y_t$
Linear	12.39	11.82	10.07
Quadratic	37.19	38.12	33.74
Bump	15.75	13.90	10.60
Exponential	27.71	23.83	19.49
Cycle 1	29.55	29.07	23.12
Cycle 2	33.25	31.30	22.62

4.1.3 Bias

From the comparative analysis, we obtained the following results. They are organized as follows. The first three are model calibrated estimators based on penalized splines, local polynomial and nadaraya watson kernel respectively. The fourth is a design estimator. The fifth, sixth and seventh are internally calibrated estimators based on penalized splines, local polynomial and Nadaraya Watson kernel respectively.

Table 4.3: Relative Biases (semiparametric)

	$\hat{y}_{smsp}$	$\hat{y}_{smlp}$	$\hat{y}_{smnw}$	$\hat{y}_{ht}$	$\hat{y}_{regsp}$	$\hat{y}_{reglp}$	$\hat{y}_{regnw}$
Linear	0.008	0.008	0.018	0.008	0.019	0.038	0.301
Quadratic	0.024	0.023	0.024	0.023	0.465	1.587	2.016
Bump	0.014	0.016	0.028	0.016	0.029	0.167	0.259
Exponential	0.001	0.004	0.009	0.010	0.002	0.017	0.103
Cycle 1	0.005	0.008	0.014	0.010	0.007	0.022	0.073
Cycle 2	0.006	0.005	0.008	0.015	0.007	0.005	0.019

From table (4.3) we observe that the biases are very small given that the population totals were in hundreds, pointing to unbiasedness. Comparing the model calibrated estimators with their corresponding internal calibrated estimators, that is $\hat{y}_{smsp}$ with $\hat{y}_{regsp}$,$\hat{y}_{smlp}$ with $\hat{y}_{reglp}$ and $\hat{y}_{smnw}$ with $\hat{y}_{regnw}$, we see that model calibration results in reduced bias than internal calibration.

Table 4.4: Relative Biases (nonparametric)

	$\hat{y}_{npsp}$	$\hat{y}_{nplp}$	$\hat{y}_{npnw}$	$\hat{y}_{ht}$	$\hat{y}_{gensp}$	$\hat{y}_{genlp}$	$\hat{y}_{gennw}$
Linear	0.010	0.009	0.020	0.008	0.019	0.038	0.301
Quadratic	0.020	0.021	0.022	0.019	0.443	2.187	2.536
Bump	0.024	0.027	0.029	0.027	0.039	0.216	0.349
Exponential	0.003	0.004	0.012	0.002	0.0016	0.019	0.143
Cycle 1	0.007	0.010	0.017	0.013	0.009	0.025	0.077
Cycle 2	0.006	0.006	0.008	0.005	0.010	0.011	0.018

From table (4.4) we again observe that the biases are very small. Again Comparing the model calibrated estimators with their corresponding internal calibrated estimators, that is $\hat{y}_{gensp}$ with $\hat{y}_{gensp}$,$\hat{y}_{nplp}$ with $\hat{y}_{genlp}$ and $\hat{y}_{npnw}$ with $\hat{y}_{gennw}$, we see that model calibration results in reduced bias than internal calibration just like in semiparametric estimation.

4.1.4 Relative Efficiency

Table 4.5: Relative Efficiency (semiparametric)

	$\hat{y}_{smsp}$	$\hat{y}_{smlp}$	$\hat{y}_{smnw}$	$\hat{y}_{ht}$	$\hat{y}_{regsp}$	$\hat{y}_{reglp}$	$\hat{y}_{regnw}$
Linear	1.180	1.301	1.752	1	3.639	34.583	1641.763
Quadratic	1.229	1.627	1.930	1	3.153	11308.16	15369.7
Bump	1.277	1.411	1.832	1	2.561	545.785	431.521
Exponential	1.417	1.827	1.873	1	3.081	784.672	479.265
Cycle 1	1.361	1.901	2.136	1	2.909	7.962	71.887
Cycle 2	1.218	1.267	1.970	1	3.361	10.251	12.003

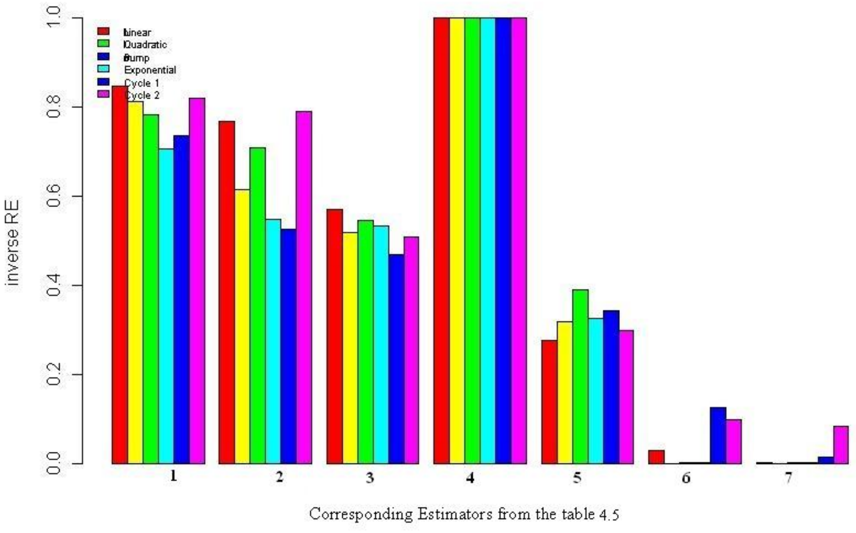

Figure 4.1: Inverse RE (Semiparametric onestage)

The estimators 1 to 7 in figure (4.1) represent the estimators $\hat{y}_{smsp}$ to $\hat{y}_{regnw}$ respectively in the table (4.5). We used the design estimator as the bases for comparison since all the model based estimators are new in the context of semiparametric estimation and so that we could carry out a sensitivity analysis when some categorical variables are not included in a semiparametric estimator. From table (4.5) and figure (4.1), all the model calibrated estimators $\hat{y}_{smsp}$,$\hat{y}_{smlp}$ and $\hat{y}_{smnw}$ have performance very close to that of design estimator even though none of them performs better than the design estimator. The model calibrated estimator based on penalied splines $\hat{y}_{smsp}$ performs better than the other model calibrated estimators based on kernel methods. For the internally calibrated estimators, only the penalized spline one has a performace close to that of the design estimator while the kernel based estmators are found to perform poorly, again illustrating the power of penalized spline estimators. The internally calibrated penalized spline estimator $\hat{y}_{regsp}$ does not however fare better than the kernel based model calibrated estimators $\hat{y}_{smlp}$ and $\hat{y}_{smnw}$. It would appear that whether to model-calibrate or not is more significant than the choice of the nonparametric function to use to fit the missing values.

Table 4.6: Relative Efficiency (nonparametric)

	$\hat{y}_{npsp}$	$\hat{y}_{nplp}$	$\hat{y}_{npnw}$	$\hat{y}_{ht}$	$\hat{y}_{gensp}$	$\hat{y}_{genlp}$	$\hat{y}_{gennw}$
Linear	1.390	1.501	1.965	1	3.340	30.573	1591.803
Quadratic	1.329	1.337	2.230	1	3.633	11190.10	14329.17
Bump	1.579	1.721	2.134	1	2.963	540.725	433.542
Exponential	1.527	1.922	1.979	1	4.108	780.290	496.361
Cycle 1	1.601	1.936	2.112	1	3.600	6.783	66.856
Cycle 2	1.468	1.562	1.988	1	2.991	10.001	13.123

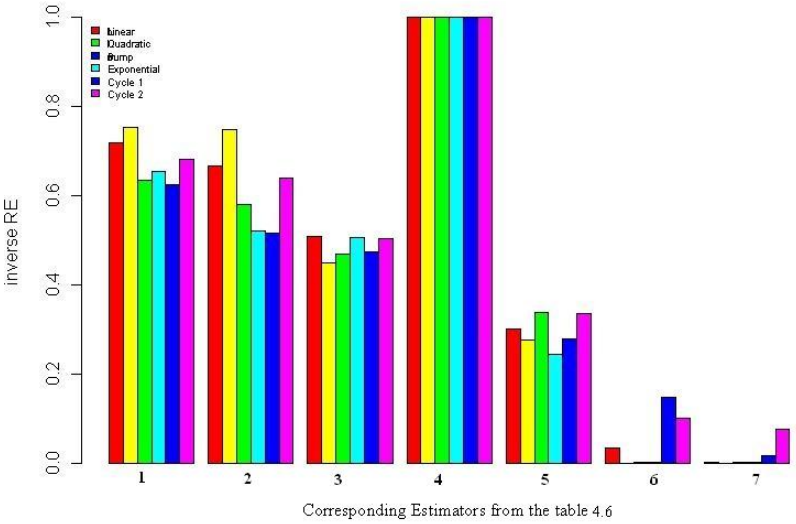

Figure 4.2: Inverse RE (Nonparametric Onestage)

The estimators 1 to 7 in figure (4.2) represent the estimators $\hat{y}_{npsp}$ to $\hat{y}_{gennw}$ respectively in the table (4.6). From table (4.6) and figure (4.2), all the model calibrated estimators $\hat{y}_{npsp}$,$\hat{y}_{nplp}$ and $\hat{y}_{npnw}$ hace performance close to the design estimator just like in semiparametric estimation, but none performs better than the design estimator. The model calibrated estimator based on penalized splines $\hat{y}_{npsp}$ performs better than the other model calibrated estimators. For the internally calibrated estimators, only the one based on penalized splines has a performance close to the performance of the design estimator. The internally calibrated penalized spline does not fare better than the kernel based model calibrated estimators $\hat{y}_{nplp}$ and $\hat{y}_{npnw}$ just like in the case of semiparametric model.

4.1.5 Bias on Sensitivity Analysis

Table 4.7: Bias on Removing Z_3(semiparametric)

	$\hat{y}_{smsp}$	$\hat{y}_{smlp}$	$\hat{y}_{smnw}$	$\hat{y}_{ht}$	$\hat{y}_{regsp}$	$\hat{y}_{reglp}$	$\hat{y}_{regnw}$
Linear	0.012	0.029	0.029	0.012	0.015	0.276	0.103
Quadratic	0.041	0.068	0.045	0.039	0.045	1.201	0.233
Bump	0.014	0.030	0.018	0.016	0.017	0.373	0.113
Exponential	0.145	0.131	0.151	0.146	0.143	0.591	0.176
Cycle 1	0.012	0.011	0.017	0.012	0.012	0.225	0.049
Cycle 2	0.010	0.011	0.023	0.010	0.014	0.113	0.105

Looking at table (4.7) we observe that the biases still remain very small even after the vector Z_3 is dropped. Comparing the model calibrated estimators with their corresponding internally calibrated estimators, that is $\hat{y}_{smsp}$ with $\hat{y}_{regsp}$, $\hat{y}_{smlp}$ with $\hat{y}_{reglp}$ and $\hat{y}_{smnw}$ with $\hat{y}_{regnw}$, we observe that the model calibrated estimators remain less biased than their corresponding internally calibrated estimators. Same observations were made even when Z_1 and Z_2 were the omitted vectors.

4.1.6 Relative Efficiency on Sensitivity

Table 4.8: Relative Efficiency on Removing Z_3(semiparametric)

	$\hat{y}_{smsp}$	$\hat{y}_{smlp}$	$\hat{y}_{smnw}$	$\hat{y}_{ht}$	$\hat{y}_{regsp}$	$\hat{y}_{reglp}$	$\hat{y}_{regnw}$
Linear	1.144	1.344	2.016	1	4.192	580.989	91.503
Quadratic	1.297	4.031	4.528	1	4.353	1953.703	58.408
Bump	1.308	2.020	2.365	1	3.117	851.133	77.568
Exponential	1.6910	2.230	2.636	1	3.697	1903.492	33.693
Cycle 1	1.456	2.150	2.8882	1	3.179	701.722	19.286
Cycle 2	1.352	2.311	3.205	1	3.511	64.356	115.493

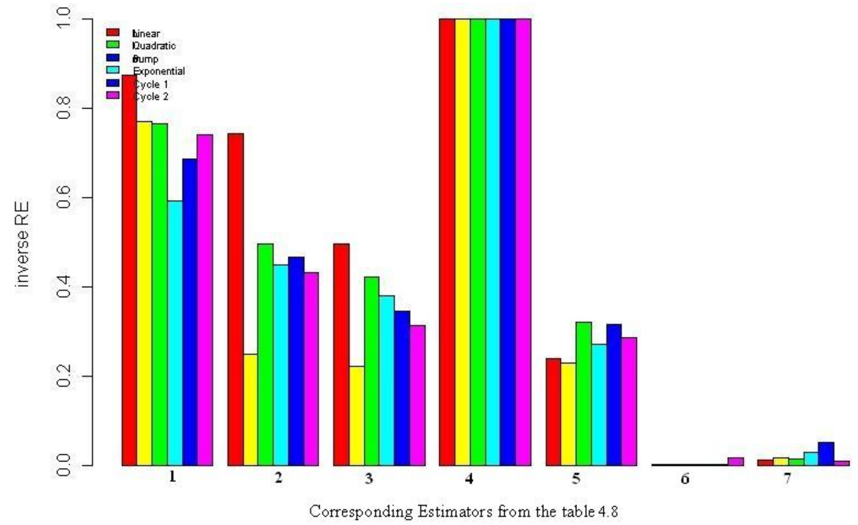

Figure 4.3: Inverse RE on removing Z_3

The estimators 1 to 7 in figure (4.3) represent the estimators $\hat{y}_{smsp}$ to $\hat{y}_{regnw}$ respectively in the table (4.8). We observe, from table (4.8) and figure (4.3) that all the model calibrated estimators have performance close to the performance of the design estimator with the one based on penalized splines being the best. Of the internally calibrated estimators, only the one based on penalized penalized splines has a performance close to the performance of the design estimator. The kernel based ones, $\hat{y}_{reglp}$ and $\hat{y}_{regnw}$ perform poorly compared to the design estimator. It is certain therefore that model calibrated estimators are more robust than the corresponding internally calibrated estimators. Same observations were made even when Z_1 and Z_2 were dropped.

4.2 Empirical Analysis Under Two Stage Sampling

4.2.1 Auxilliary Information at Cluster Level Only

We simulated a population of independent and identically distributed variable x using uniform(0.1) and a categorical matrix Z. For each generated x_i and vector Z_i and for each mean function , $N_i = 100$ element values were generated as follows for semiparametric modelling.

$$y_{ik} = \frac{g(x_i, Z_i)}{N_i} + \frac{\epsilon_{ik}}{\sqrt{N_i}}, \{\epsilon_{ik}\}\, iidN(0, 0.1) \tag{4.3}$$

where y_{ik} is the kth element in the ith cluster and $g(x_i, Z_i)$ which we simply write g_i is the mean function for t_i, the cluster total, obtained semiparametrically. For nonparametric estimation element values were generated as

$$y_{ik} = \frac{\mu(x_i)}{N_i} + \frac{\epsilon_{ik}}{\sqrt{N_i}}, \{\epsilon_{ik}\}\, iidN(0, 0.1) \tag{4.4}$$

$\mu(\hat{x_i})$ is the mean function for t_i.

At stage one, a sample of clusters was generated by simple random sampling with sample size m=50. At stage two, within each of the selected clusters, sub samples of elements of size n_i were generated by simple random sampling. We considered the case where $n_i = 50$ for all clusters and the case $n_i = N_i$ which is just but one stage sampling. Since no auxilliary information within the clusters is considered, cluster totals were estimated using the Horvitz Thompson design estimator. For each pair (x_i, Z_i) in the case of semiparametric modelling or for each x_i in case of nonparametric modelling, and for each generating function described earlier , R=100 replicate samples of clusters were generated and the estimates calculated.Trials involving 500 and 1000 replicate samples yielded same results as for 100 replicates.

We consider the following estimators in the analysis.

1. Horvitz Thompson, $\hat{y}_{ht2}$ with inclusion probability $\pi_i = \frac{m}{M}$

2. Model Calibrated Model Assisted Semiparametric Estimator $\hat{y}_{sm2}$, (3.50), proposed, for which we considered three cases;$\hat{y}_{sm2}$ based on penalized splines which we denote by $\hat{y}_{smsp2}$ in which $\hat{g}_i$ is as defined in (3.40), $\hat{y}_{sm2}$ based on local polynomial which is denoted by $\hat{y}_{smlp2}$ in which $\hat{g}_i$ is as defined in (3.43) and $\hat{y}_{sm2}$ based on Nadaraya Watson kernel smoothing which we denote by $\hat{y}_{smnw2}$ in which $\hat{g}_i$ is as defined in (3.43) but with polynomial degree zero.

3. Internally Calibrated Model Assisted Semiparameric Estimator $\hat{y}_{reg2}$, (3.54), for which we consider the three cases; $\hat{y}_{reg2}$ based on penalized splines which we denote by $\hat{y}_{regsp2}$ in which $\hat{g}_i$ is as defined in (3.40), $\hat{y}_{reg2}$ based on local polynomial which we denote by $\hat{y}_{reglp2}$ in which $\hat{g}_i$ is as defined in (3.43)and $\hat{y}_{reg2}$ based on Nadaraya Watson kernel smoothing which we denote by $\hat{y}_{regnw2}$ in which $\hat{g}_i$ is as defined in (3.43) but with polynomial degree zero.

The performance of any estimator say y_{est} in $\hat{y}_{ht2}$, $\hat{y}_{smsp2}$, $\hat{y}_{smlp2}$, $\hat{y}_{smnw2}$, $\hat{y}_{regsp2}$, $\hat{y}_{reglp2}$, $\hat{y}_{regnw2}$ was evaluated using its relative bias R_B and relative efficiency R_E defined ealier

Like in one stage sampling, we also carried out a Sensitivity Analysis by looking at the effects that ignoring a variable in the categorical matrix would have on the estimators.

For nonparameric estimation, we compare the performance of the three set of estimators. First if the design estimator $\hat{y}_{ht2}$. Second is the model calibrated estimators $\hat{y}_{np2}$, (3.57), for which we consider three cases; $\hat{y}_{np2}$ based on penalized splines denoted by $\hat{y}_{npsp2}$ in which $\hat{\mu}_i$ is as defined in (3.55), $\hat{y}_{np2}$ based on local polynomial denoted by $\hat{y}_{nplp2}$ in which $\hat{\mu}_i$ is as defined in (3.56)and $\hat{y}_{np2}$ based on Nadaraya Watson kernel which we denote as $\hat{y}_{npnw2}$ in which $\hat{\mu}_i$ is as defined in (3.56) but with polynomial degree zero.Third is the internally calibrated estimators $\hat{y}_{gen2}$, (3.61), for which we consider three cases; $\hat{y}_{gen2}$ based on penalized splines denoted by $\hat{y}_{gensp2}$ in which $\hat{\mu}_i$ is as defined in (3.55), $\hat{y}_{gen2}$ based on local polynomial denoted by $\hat{y}_{genlp2}$ in which $\hat{\mu}_i$ is as defined in (3.56)and $\hat{y}_{gen2}$ based on Nadaraya Watson kernel which we denote as $\hat{y}_{gennw2}$ in which $\hat{\mu}_i$ is as defined in (3.56) but with polynomial degree

4.2.1.1 Normality Test

Before the comparative analysis, we carried out a shapiro-wilk test for normality for the proposed model calibrated estimators and obtained the following results. At $\alpha = 0.05$, normality is proven. A sample of normal graphs are provided in appendix 2.

Table 4.9: Shapiro-Wilk p-values(two stage)

	$\hat{y}_{smsp2}$	$\hat{y}_{smlp2}$	$\hat{y}_{smnw2}$	$\hat{y}_{npsp2}$	$\hat{y}_{nplp2}$	$\hat{y}_{npnw2}$
Linear	0.497	0.550	0.502	0.439	0.599	0.428
Quadratic	0.672	0.507	0.473	0.555	0.555	0.555
Bump	0.761	0.748	0.744	0.829	0.828	0.828
Exponential	0.343	0.337	0.311	0.231	0.166	0.166
Cycle 1	0.497	0.497	0.497	0.567	0.578	0.578
Cycle 2	0.700	0.699	0.699	0.713	0.748	0.744

4.2.1.2 Consistency Test

We carried out a consistency test by varying the the sample size m_ρ,population size M_ρ for each population $\rho = 1, 2, 3$ and obtaining the diference between the estimate and actual total Y_t. The estimated total is taken to be the average from the R replicates. The results in the table (4.10) show the estmators are consistent.

Table 4.10: Absolute Diferences for Two Stage Sampling

	$m_1 = 30, M_1 = 300$	$m_2 = 450, M_2 = 900$	$m_3 = 1350, M_3 = 1800$
Mean Function	$\hat{y}_{smsp2} - Y_t$	$\hat{y}_{smsp2} - Y_t$	$\hat{y}_{smsp2} - Y_t$
Linear	8.45	8.44	7.10
Quadratic	34.53	29.42	29.11
Bump	25.12	20.32	16.17
Exponential	14.19	13.22	9.58
Cycle 1	8.82	8.45	8.26
Cycle 2	10.24	8.92	8.81
Mean Function	$\hat{y}_{smlp2} - Y_t$	$\hat{y}_{smlp2} - Y_t$	$\hat{y}_{smlp2} - Y_t$
Linear	8.90	8.56	7.07
Quadratic	37.22	33.64	30.39
Bump	10.66	10.64	8.49
Exponential	14.65	14.22	12.55
Cycle 1	18.89	17.44	14.50
Cycle 2	28.77	19.17	19.15
Mean Function	$\hat{y}_{smnw2} - Y_t$	$\hat{y}_{smnw2} - Y_t$	$\hat{y}_{smnw2} - Y_t$
Linear	16.66	15.34	15.13
Quadratic	44.29	44.45	35.034
Bump	20.17	18.76	18.21
Exponential	32.73	29.73	25.65
Cycle 1	31.72	31.15	28.41
Cycle 2	37.21	37.33	27.09
Mean Function	$\hat{y}_{npsp2} - Y_t$	$\hat{y}_{npsp2} - Y_t$	$\hat{y}_{npsp2} - Y_t$
Linear	6.78	6.67	6.35
Quadratic	32.16	27.77	27.00
Bump	23.32	18.66	15.84
Exponential	12.54	11.67	8.89
Cycle 1	6.99	6.71	6.02
Cycle 2	8.20	7.88	6.95
Mean Function	$\hat{y}_{nplp2} - Y_t$	$\hat{y}_{nplp2} - Y_t$	$\hat{y}_{nplp2} - Y_t$
Linear	6.99	6.28	5.37
Quadratic	34.36	31.10	27.36
Bump	8.76	8.09	6.55
Exponential	12.85	12.11	10.78
Cycle 1	16.98	14.64	12.53
Cycle 2	24.00	15.13	14.75
Mean Function	$\hat{y}_{npnw2} - Y_t$	$\hat{y}_{npnw2} - Y_t$	$\hat{y}_{npnw2} - Y_t$
Linear	14.85	13.48	13.10
Quadratic	40.44	40.43	31.64
Bump	17.53	16.67	15.52
Exponential	30.77	27.56	24.44
Cycle 1	29.27	28.45	27.46
Cycle 2	35.49	34.03	26.28

4.2.1.3 Bias

Below are the results obtained. As was the case in previous tables, the first three are model calibrated estimators, fourth is a design estimator and the last three are the internally calibrated estimators.

Table 4.11: Relative Biases (semiparametric-one level covariate)

	$\hat{y}_{smsp2}$	$\hat{y}_{smlp2}$	$\hat{y}_{smnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{regsp2}$	$\hat{y}_{reglp2}$	$\hat{y}_{regnw2}$
Linear	0.009	0.009	0.018	0.010	0.021	0.038	0.308
Quadratic	0.034	0.033	0.034	0.033	0.445	1.595	2.216
Bump	0.024	0.026	0.038	0.026	0.039	0.177	0.279
Exponential	0.003	0.006	0.011	0.012	0.004	0.019	0.105
Cycle 1	0.006	0.010	0.017	0.013	0.010	0.026	0.077
Cycle 2	0.006	0.005	0.009	0.017	0.010	0.007	0.022

From table (4.11), we observe that the biases are very small again pointing to unbiasedness. Comparing each model calibrated estimator with its corresponding internally calibrated estimator, $\hat{y}_{smsp2}$ with $\hat{y}_{regsp2}$, $\hat{y}_{smlp2}$ with $\hat{y}_{reglp2}$ and $\hat{y}_{smnw2}$ with $\hat{y}_{regnw2}$, we see that model calibration results in reduced bias than internal calibration.

Table 4.12: Relative Biases (nonparametric-one level covariate)

	$\hat{y}_{npsp2}$	$\hat{y}_{nplp2}$	$\hat{y}_{npnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{gensp2}$	$\hat{y}_{genlp2}$	$\hat{y}_{gennw2}$
Linear	0.008	0.008	0.018	0.011	0.024	0.035	0.401
Quadratic	0.044	0.043	0.044	0.043	0.443	1.515	2.198
Bump	0.034	0.036	0.045	0.019	0.046	0.197	0.309
Exponential	0.003	0.005	0.011	0.011	0.008	0.020	0.112
Cycle 1	0.007	0.012	0.018	0.012	0.012	0.024	0.082
Cycle 2	0.006	0.006	0.008	0.019	0.012	0.009	0.021

From table (4.12), Comparing $\hat{y}_{npsp2}$ with $\hat{y}_{gensp2}$, $\hat{y}_{nplp2}$ with $\hat{y}_{genlp2}$ and $\hat{y}_{npnw2}$ with $\hat{y}_{gennw2}$, we still see that model calibration resulting in reduced bias than internal calibration just like was the case in semiparametric estimation above.

4.2.1.4 Relative Efficiency

Table 4.13: Relative Efficiency (Semiparametric one level covariate)

	$\hat{y}_{smsp2}$	$\hat{y}_{smlp2}$	$\hat{y}_{smnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{regsp2}$	$\hat{y}_{reglp2}$	$\hat{y}_{regnw2}$
Linear	1.286	1.541	1.976	1	4.039	36.583	1538.704
Quadratic	1.829	2.227	2.530	1	3.753	10309.77	14332.5
Bump	1.977	2.116	2.542	1	3.244	546.585	432.271
Exponential	2.010	2.425	2.493	1	3.679	787.646	469.355
Cycle 1	1.861	2.451	2.639	1	3.529	8.922	74.437
Cycle 2	1.928	1.967	2.678	1	4.061	11.001	13.113

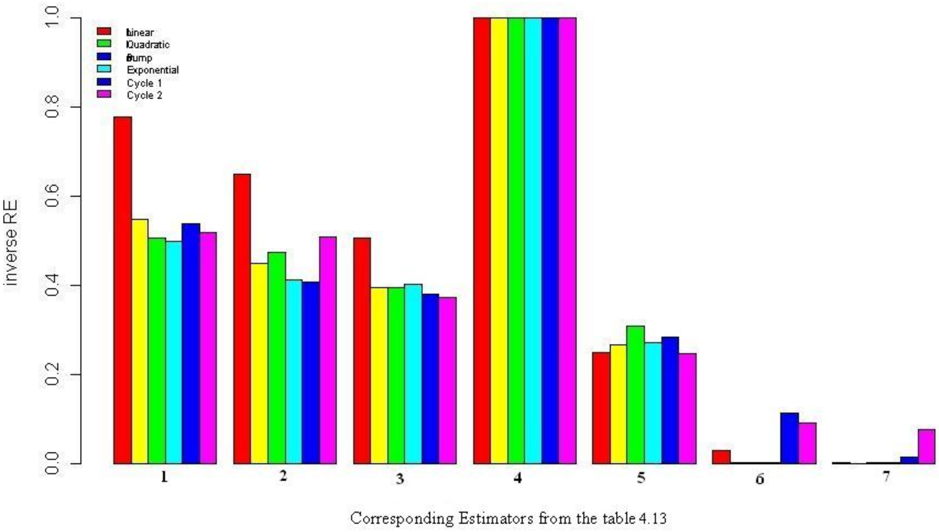

Figure 4.4: Inverse RE (Semiparametric one level covariate)

The estimators 1 to 7 in figure (4.4) represent the estimators $\hat{y}_{smsp2}$ to $\hat{y}_{regnw2}$ respectively in the table (4.13). As in one stage sampling, we used the design estimator as the bases for comparison since all the model based estimators are new in the context of semiparametric estimation and so that we could carry out a sensitivity analysis when some categorical variables are not included in a semiparametric estimator. From table (4.13) and figure (4.4), all the model calibrated estimators $\hat{y}_{smsp2}$, $\hat{y}_{smlp2}$ and $\hat{y}_{smnw2}$ have performances close to the performance of the design estimator but none performs better than the design estimator. The model calibrated estimator based on penalized splines, $\hat{y}_{smsp2}$ performs better than the other model calibrated estimators based on kernel methods. The internally calibrated estimator $\hat{y}_{regsp2}$ based on penalized splines has a performance close to the design estimator while kernel based perform poorly compared to the design estimator, again illustrating the power of penalized spline estimators. $\hat{y}_{regsp2}$ does not however fare better than the kernel based model calibrated estimators $\hat{y}_{smlp2}$ and $\hat{y}_{smnw2}$. This confirms that the choice of whether to model calibrate or not is more significant than the choice of the nonparametric procedure to use to fit the missing values.

Table 4.14: Relative Efficiency (nonparametric one level covariate)

	$\hat{y}_{npsp2}$	$\hat{y}_{nplp2}$	$\hat{y}_{npnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{gensp2}$	$\hat{y}_{genlp2}$	$\hat{y}_{gennw2}$
Linear	1.586	1.839	1.998	1	4.339	34.681	1538.704
Quadratic	1.730	2.422	2.639	1	4.213	10117.12	12372.6
Bump	1.872	2.006	2.361	1	3.004	526.345	437.431
Exponential	1.980	2.143	2.198	1	3.638	790.001	471.349
Cycle 1	1.881	2.215	2.479	1	4.126	9.721	72.467
Cycle 2	1.956	2.065	2.529	1	4.361	12.013	15.102

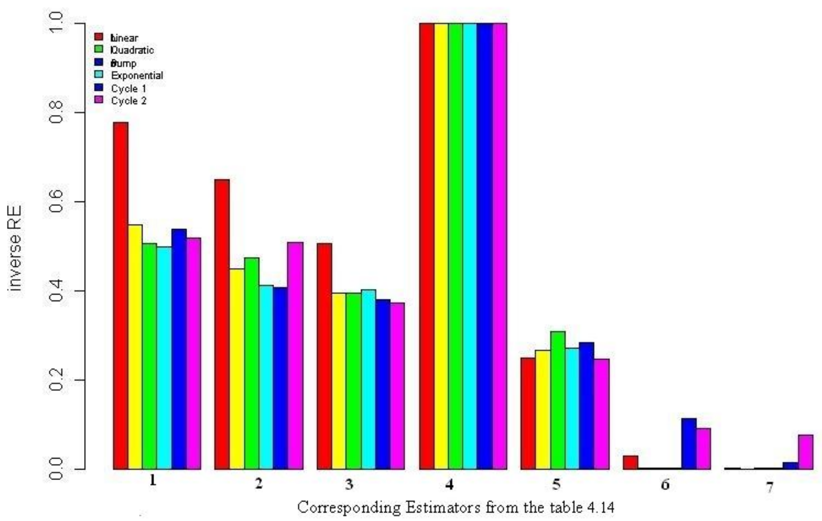

Figure 4.5: Inverse RE (Nonparametric one level covariate)

The estimators 1 to 7 in figure (4.5) represent the estimators $\hat{y}_{npsp2}$ to $\hat{y}_{gennw2}$ respectively in the table (4.14). From table (4.14) and figure (4.5), all the model calibrated estimators $\hat{y}_{npsp2}$, $\hat{y}_{nplp2}$ and $\hat{y}_{npnw2}$ have performances close to the performance of the design estimator but again none performs better than it. $\hat{y}_{npsp2}$ performs better than the other model calibrated estimators. For the internally calibrated estimators, only the penalized spline one has a performance close to that of the design estimator while the kernel based ones perform poorly compared to the design estimator, again illustrating the power of penalized splines estimators.

4.2.1.5 Bias on Sensitivity Analysis

Table 4.15: Bias on Removing Z_3(semiparametric-one level covariate)

	$\hat{y}_{smsp2}$	$\hat{y}_{smlp2}$	$\hat{y}_{smnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{regsp2}$	$\hat{y}_{reglp2}$	$\hat{y}_{regnw2}$
Linear	0.022	0.039	0.039	0.023	0.026	0.292	0.113
Quadratic	0.062	0.089	0.066	0.061	0.065	1.231	0.263
Bump	0.025	0.052	0.038	0.035	0.038	0.400	0.133
Exponential	0.244	0.232	0.249	0.245	0.243	0.690	0.277
Cycle 1	0.023	0.022	0.028	0.022	0.024	0.235	0.059
Cycle 2	0.019	0.020	0.029	0.021	0.024	0.123	0.117

Looking at table (4.15), we observe that the biases still remain very small even after the variable Z_3 is dropped. Comparing $\hat{y}_{smsp2}$ with $\hat{y}_{regsp2}$, $\hat{y}_{smlp2}$ with $\hat{y}_{reglp2}$ and $\hat{y}_{smnw2}$ with $\hat{y}_{regnw2}$, we observe that the model calibrated estimators remain less biased than their corresponding internally calibrated estimators. Same observations were made even when Z_1 and Z_2 were the omitted variables.

4.2.1.6 Relative Efficiency on Sensitivity

Table 4.16: Relative Efficiency on Removing Z_3(semiparametric one level covariate)

	$\hat{y}_{smsp2}$	$\hat{y}_{smlp2}$	$\hat{y}_{smnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{regsp2}$	$\hat{y}_{reglp2}$	$\hat{y}_{regnw2}$
Linear	1.944	2.144	2.816	1	4.992	579.968	92.573
Quadratic	2.007	4.834	5.326	1	5.152	1950.456	60.206
Bump	2.008	2.822	3.169	1	3.921	856.100	77.568
Exponential	1.992	2.533	2.942	1	3.986	1900.425	36.234
Cycle 1	1.952	2.669	3.389	1	3.689	704.142	19.906
Cycle 2	1.973	2.944	3.901	1	4.131	66.337	120.063

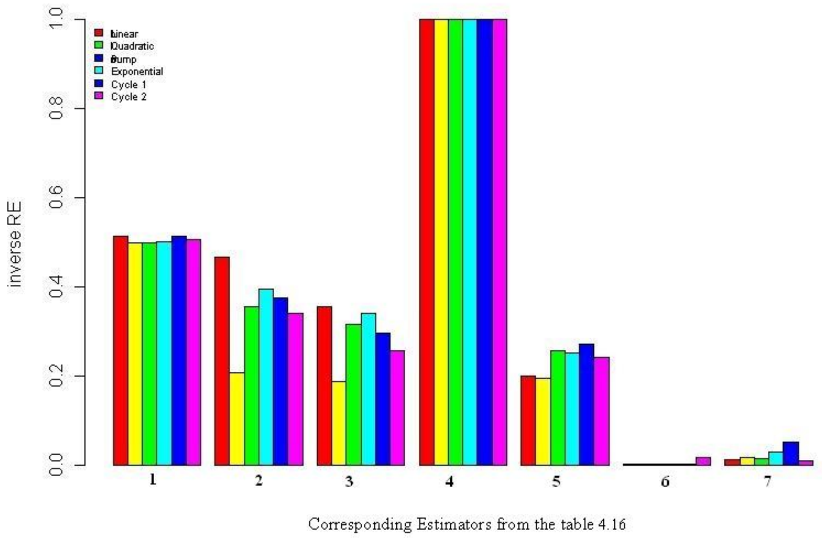

Figure 4.6: Inverse RE on removing Z_3 (Semiparametric one level covariate)

The estimators 1 to 7 in figure (4.6) represent the estimators $\hat{y}_{smsp2}$ to $\hat{y}_{regnw2}$ respectively in the table (4.16). We observe, from table (4.16) and figure(4.6), that all the model calibrated estimators have performances close to the performance of the design estimator with the penalized spline one being the best. Of the internally calibrated estimators, only the penalized spline one has a performance close to the performance of the design estimator with the kernel based ones performing poorly compared to the design estimator. The observations are similar to those obtained before dropping Z_3

4.2.2 Auxilliary Information at Both Element and Cluster Levels

We simulated a population of independent and identically distributed variable x using uniform(0.1) and a categorical matrix Z. For each generated x_i and vector Z_i and for each mean function , $N_i = 100$ element values were generated as follows for semiparametric modelling.

$$y_{ik} = \frac{g(x_i, Z_i)}{\sqrt{N_i}} + \frac{\epsilon_{ik}}{\sqrt{N_i}}, \{\epsilon_{ik}\}\, iidN(0, 0.1) \tag{4.5}$$

Where y_{ik} is the kth element in the ith cluster and $g(x_i, Z_i)$ which we simply write g_i is the mean function for t_i, the cluster total, obtained semiparametrically. For nonparametric estimation element values were generated as

$$y_{ik} = \frac{\mu(x_i)}{N_i} + \frac{\epsilon_{ik}}{\sqrt{N_i}}, \{\epsilon_{ik}\}\, iidN(0, 0.1) \tag{4.6}$$

$\mu(x_i)$ is the mean function for t_i. For simplicity, within each cluster the auxiliary information at element level x_{ik} for semiparametric modelling was generated using the linear and quadratic mean functions and working backward to obtain

$$x_{ik} = \frac{y_{ik} - 2 - Z_{ik}\beta'}{5} \tag{4.7}$$

and

$$x_{ik} = \frac{\sqrt{y_{ik} - Z_{ik}\beta'} - 2}{5} \tag{4.8}$$

Where the squareroot is assumed posive and Z_{ik} is the matrix (Z_{i1}, Z_{i2}, Z_{i3}),where Z_{i1} is a matrix of 1s, Z_{i2} is a matrix of 2s,3s and 4s, while Z_{i3} is a matrix of 5s, 6s and 7s. β is the matrix (1,2,3). For nonparametric modelling the aixilliary information was obtained as

$$x_{ik} = \frac{y_{ik} - 2}{5} \tag{4.9}$$

and

$$x_{ik} = \frac{\sqrt{y_{ik}} - 2}{5} \tag{4.10}$$

This provides the auxilliary information that we use to model missing values within clusters.

For each pair (x_i, Z_i) and mean function, $R = 100$ replicate samples of clusters were generated. At stage one, a sample of clusters was generated by simple random sampling with sample size $m = 50$. At stage two, within each of the selected clusters, sub samples of elements of size n_i were generated by simple random sampling. We considered the case where $n_i = 50$ for all clusters and the case $n_i = N_i$ which is just but one stage sampling. Cluster totals for the clusters were estimated using the estimators employed in one stage sampling. Using the estimated totals of the clusters and the estimators described next, estimates of the population total were generated.

The estimation was such that a similar estimator to the one used in the estimation of cluster totals is used at the estimation of the population total. For example, if at cluster level we use model calibrated estimator based on penalized splines to estimate the cluster totals, then we also use the model calibrated estimator based on penalized splines to estimate the population total.

We consider the following estimators in the analysis.

1. Horvitz Thompson, $\hat{y}_{ht2}$ with inclusion probability $\pi_i = \frac{m}{M}$

2. The Model Calibrated Model Assisted Semiparametric Estimator $\hat{y}_{ssm2}$, (equation 3.63) that we have proposed. We considered three cases based on the nonparametric method used to obtain the mean estimate. These are; $\hat{y}_{ssmsp2}$ whepenalized splines are used , $\hat{y}_{ssmlp2}$ when local polynomial is used and $\hat{y}_{ssmnw2}$ in case of Nadaraya Watson kernel smoothing.

3. Internally Calibrated Model Assisted Semiparameric Estimator $\hat{y}_{regreg2}$, (equation 3.64). We also consider three cases; $\hat{y}_{regregsp2}$ in case of penalized splines, $\hat{y}_{regreglp2}$ in case of local polynomial and $\hat{y}_{regregnw2}$ for Nadaraya Watson kernel smoothing.

The performance of any estimator say y_{est} in $\hat{y}_{ht2}$, $\hat{y}_{ssmsp2}$, $\hat{y}_{ssmlp2}$, $\hat{y}_{ssmnw2}$, $\hat{y}_{regregsp2}$, $\hat{y}_{regreglp2}$, $\hat{y}_{regregnw2}$ was evaluated using its relative bias R_B and relative efficiency R_E defined ealier

We also carried out a Sensitivity Analysis by looking at the effects that ignoring a variable in the categorical matrix would have on the estimators. We dropped values available at cluster level. Same effects would be expected if an auxilliary variable at element level is dropped since the processes of estimation at both stages are similar.

For nonparameric estimation, we compare the performance of the three sets of estimators.

1. Design estimator $\hat{y}_{ht2}$

2. model calibrated estimator $\hat{y}_{nnp2}$, (equation 3.70), for which we consider three cases. $\hat{y}_{nnpsp2}$ denote $\hat{y}_{nnp2}$ when its based on penalized splines, $\hat{y}_{nnplp2}$ denote $\hat{y}_{nnp2}$ when its based on local polynomial and $\hat{y}_{nnpnw2}$ represent $\hat{y}_{nnp2}$ based on Nadaraya Watson kernel smoothing.

3. Internally calibrated estimators $\hat{y}_{gengen2}$, (equation 3.71), where we also look at three cases;$\hat{y}_{gengensp2}$ to denote $\hat{y}_{gengen2}$ based on penalized splines , $\hat{y}_{gengenlp2}$ to denote $\hat{y}_{gengen2}$ based on local polynomial and $\hat{y}_{gengennw2}$ to represent $\hat{y}_{gengen2}$ based on Nadaraya Watson kernel smoothing.

We report on the observations for the case where the auxilliary information at element level was obtained from the linear function. Similar observations were obtained when the auxilliary information at the element level was obtained from the quadratic function. Clearly, the results would similary not be different if any of the six generating functions is considered.

From the analysis, we obtained the following results.

4.2.2.1 Bias

Table 4.17: Relative Biases (semiparametric-two level covariates)

	$\hat{y}_{ssmsp2}$	$\hat{y}_{ssmlp2}$	$\hat{y}_{ssmnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{regregsp2}$	$\hat{y}_{regreglp2}$	$\hat{y}_{regregnw2}$
Linear	0.015	0.015	0.025	0.017	0.028	0.048	0.328
Quadratic	0.041	0.039	0.041	0.039	0.516	1.645	2.906
Bump	0.031	0.036	0.040	0.036	0.048	0.247	0.339
Exponential	0.013	0.016	0.021	0.023	0.014	0.030	0.125
Cycle 1	0.012	0.015	0.023	0.019	0.018	0.034	0.086
Cycle 2	0.012	0.010	0.015	0.022	0.017	0.013	0.028

From table (4.17), we observe that the biases are very small again pointing to unbiasedness. Comparing $\hat{y}_{ssmsp2}$ with $\hat{y}_{regregsp2}$, $\hat{y}_{ssmlp2}$ with $\hat{y}_{regreglp2}$ and $\hat{y}_{ssmnw2}$ with $\hat{y}_{regregnw2}$, we see that model calibration results in reduced bias than internal calibration. Comparing table (4.17) with table (4.11) and comparing the observations for corresponding estimators, for example $\hat{y}_{ssmsp2}$ with $\hat{y}_{smsp2}$, $\hat{y}_{ssmlp2}$ with $\hat{y}_{smlp2}$ and so on, we see that biases are higher when we apply modelling even within the clusters(table (4.17)) than when we used design estimation within clusters(table (4.11)).This is expected because a design method is always more efficient.

Table 4.18: Relative Biases (nonparametric-two level covariates)

	$\hat{y}_{nnpsp2}$	$\hat{y}_{nnplp2}$	$\hat{y}_{nnpnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{gengensp2}$	$\hat{y}_{gengenlp2}$	$\hat{y}_{gengennw2}$
Linear	0.012	0.012	0.023	0.016	0.029	0.038	0.407
Quadratic	0.053	0.054	0.052	0.053	0.453	1.565	2.258
Bump	0.045	0.046	0.046	0.030	0.057	0.307	0.320
Exponential	0.010	0.012	0.019	0.018	0.016	0.031	0.122
Cycle 1	0.010	0.015	0.022	0.015	0.015	0.028	0.086
Cycle 2	0.009	0.009	0.010	0.022	0.015	0.012	0.024

From table (4.18), Comparing $\hat{y}_{nnpsp2}$ with $\hat{y}_{gengensp2}$, $\hat{y}_{nnplp2}$ with $\hat{y}_{gengenlp2}$ and $\hat{y}_{nnpnw2}$ with $\hat{y}_{gengennw2}$, we see that model calibration resulted in reduced bias than internal calibration just like was the case in one stage sampling.

Comparing table (4.18) with table (4.12) and comparing the observations for corresponding estimators, for example $\hat{y}_{nnpsp2}$ with $\hat{y}_{npsp2}$, $\hat{y}_{nnplp2}$ with $\hat{y}_{nplp2}$ and so on, we see that biases are higher when we apply modelling even within the clusters(table (4.18)) than when we used design estimation within clusters(table (4.12)). This, as stated earlier is expected since a design method is more efficient.

4.2.2.2 Relative Efficiency

Table 4.19: Relative Efficiency (semiparametric-two level covariates)

	$\hat{y}_{ssmsp2}$	$\hat{y}_{ssmlp2}$	$\hat{y}_{ssmnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{regregsp2}$	$\hat{y}_{regreglp2}$	$\hat{y}_{regregnw2}$
Linear	1.497	1.242	2.175	1	4.229	38.573	1550.004
Quadratic	2.027	2.431	2.730	1	3.933	10330.00	14370.6
Bump	2.168	2.320	2.743	1	3.454	560.659	446.332
Exponential	2.213	2.630	2.691	1	1.890	791.657	480.553
Cycle 1	2.059	2.641	2.841	1	3.731	10.945	77.077
Cycle 2	2.131	2.172	2.879	1	4.259	13.456	14.321

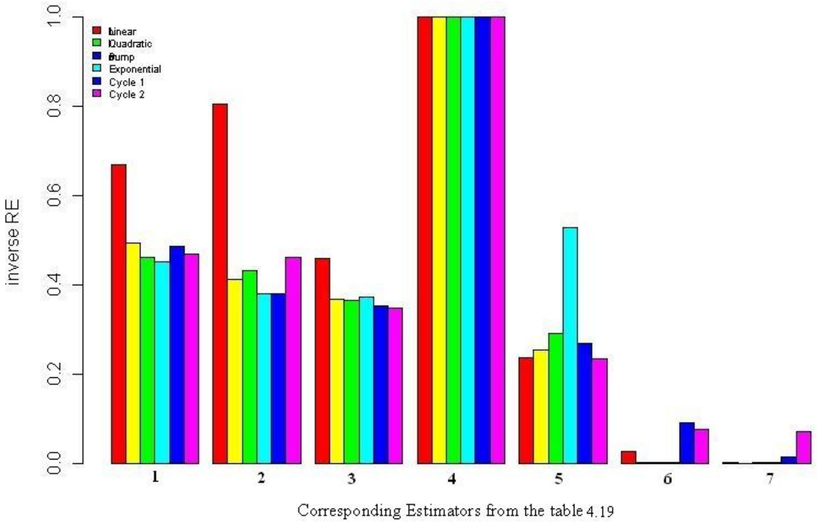

Figure 4.7: Inverse RE (semiparametric two level covariate)

The estimators 1 to 7 in figure (4.7) represent the estimators $\hat{y}_{ssmsp2}$ to $\hat{y}_{regregnw2}$ respectively in the table (4.19). As in previous cases, we used the design estimator as the bases for comparison. From table (4.19) and figure (4.7), all the model calibrated estimators $\hat{y}_{ssmsp2}$, $\hat{y}_{ssmlp2}$ and $\hat{y}_{ssmnw2}$ have performances close to the design estimator but as has been the case so far, none performs better than it. $\hat{y}_{ssmsp2}$ performs better than the other model calibrated estimators. For the internally calibrated estimators, only the penalized spline one has a performance close to that of the design estimator while the kernel based ones perform poorly to the design estimator, again illustrating the power of penalized spline estimators. Comparing table (4.19) with table (4.13) and comparing the observations for corresponding estimators, for example $\hat{y}_{ssmsp2}$ with $\hat{y}_{smsp2}$, $\hat{y}_{ssmlp2}$ with $\hat{y}_{smlp2}$ and so on, we see that applying modelling even within the clusters(table (4.19)) does not result in better than when we used design estimation within clusters(table (4.13)). This is expected because as mentioned severaly, a design method is more efficient. However, the estimators in using modelling within clusters are still performing closely to the design estimator and hence still very reliable.

Table 4.20: Relative Efficiency (nonparametric-two level covariates)

	$\hat{y}_{nnpsp2}$	$\hat{y}_{nnplp2}$	$\hat{y}_{nnpnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{gengensp2}$	$\hat{y}_{gengenlp2}$	$\hat{y}_{gengennw2}$
Linear	1.783	2.136	2.208	1	4.542	36.683	1541.342
Quadratic	1.931	2.623	2.842	1	4.418	10134.21	12389.6
Bump	2.072	2.205	2.571	1	3.214	531.350	447.443
Exponential	2.180	2.342	2.396	1	4.842	801.211	491.356
Cycle 1	2.079	2.417	2.682	1	4.331	9.923	84.234
Cycle 2	2.159	2.270	2.727	1	4.565	13.020	17.162

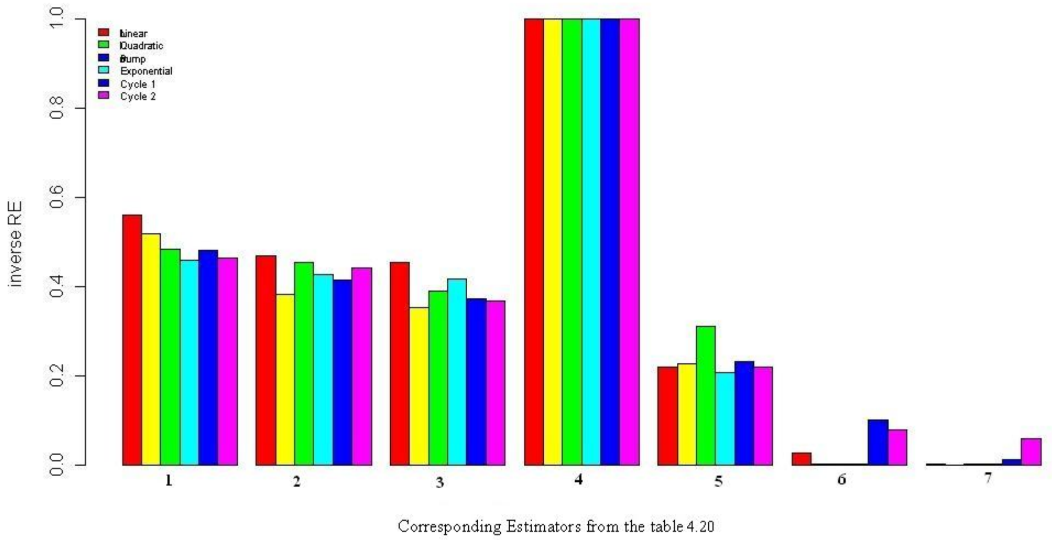

Figure 4.8: Inverse RE (nonparametric two level covariate)

The estimators 1 to 7 in figure (4.8) represent the estimators $\hat{y}_{nnpsp2}$ to $\hat{y}_{gengennw2}$ respectively in the table (4.20). From table (4.20) and figure (4.8) , all the model calibrated estimators $\hat{y}_{nnpsp2}$, $\hat{y}_{nnplp2}$ and $\hat{y}_{nnpnw2}$ have performances close to that of the design estimator but none of them performs better than it. $\hat{y}_{nnpsp2}$ performs better than the other model calibrated estimators. For the internally calibrated estimators, only the penalized spline one has a performance close to performance of the design estimator while the kernel based ones perform poorly compared to the design estimator, confirming once more the power of penalized splines estimators over kernel besed ones.

A Comparison of table (4.20) with table (4.14) reveals similar relatioships as obeserved in comparison of table (4.19) with table (4.13) in the above section.

4.2.2.3 Bias on Sensitivity Analysis

Table 4.21: Bias on Removing Z_3(semiparametric-two level covariates)

	$\hat{y}_{ssmsp2}$	$\hat{y}_{ssmlp2}$	$\hat{y}_{ssmnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{regregsp2}$	$\hat{y}_{regreglp2}$	$\hat{y}_{regregnw2}$
Linear	0.024	0.040	0.040	0.024	0.029	0.302	0.173
Quadratic	0.063	0.092	0.066	0.063	0.067	1.250	0.274
Bump	0.026	0.054	0.041	0.035	0.040	0.431	0.161
Exponential	0.252	0.252	0.253	0.246	0.261	0.710	0.302
Cycle 1	0.024	0.024	0.029	0.022	0.026	0.242	0.063
Cycle 2	0.021	0.022	0.031	0.022	0.028	0.155	0.152

Looking at table (4.21), we observe that the biases still remain very small even after the variable Z_3 is dropped meaning the estimators still perform well.

Comparing table (4.21) with table (4.15) and comparing the observations for corresponding estimators, for example $\hat{y}_{ssmsp2}$ with $\hat{y}_{smsp2}$, $\hat{y}_{ssmlp2}$ with $\hat{y}_{smlp2}$ and so on, reveals that biases are higher when we apply modelling within the clusters(table (4.21) than when we used design estimation within clusters(table (4.15).

4.2.2.4 Relative Efficiency on Sensitivity

Table 4.22: Relative Efficiency on Removing Z_3(semiparametric-two level covariates)

	$\hat{y}_{ssmsp2}$	$\hat{y}_{ssmlp2}$	$\hat{y}_{ssmnw2}$	$\hat{y}_{ht2}$	$\hat{y}_{regregsp2}$	$\hat{y}_{regreglp2}$	$\hat{y}_{regregnw2}$
Linear	1.952	2.214	2.897	1	5.112	582.348	93.783
Quadratic	2.017	4.911	5.525	1	5.892	1955.006	63.786
Bump	2.022	2.889	3.312	1	4.021	860.134	81.532
Exponential	2.112	2.634	2.992	1	4.289	1931.129	42.245
Cycle 1	1.992	2.745	3.429	1	3.987	715.164	22.923
Cycle 2	2.194	3.004	4.101	1	4.934	72.356	126.912

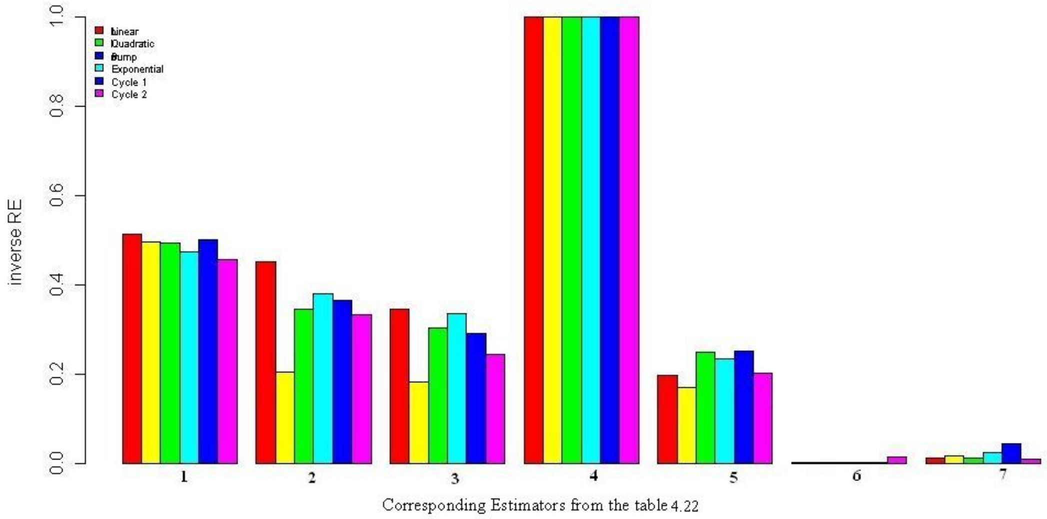

Figure 4.9: Inverse RE on Removing Z_3(Semiparametric two level covariates

The estimators 1 to 7 in figure (4.9) represent the estimators $\hat{y}_{ssmsp2}$ to $\hat{y}_{regregnw2}$ respectively in the table (4.22). We observe, from table (4.22) and figure(4.9) that all the model calibrated estimators have performances close to the design estimator illustrating the robustness of the model calibrated estimators. For the internally calibrated estimators, only the one based on penalized splines has a performance close to that of the design estimator while the kernel based ones perform poorly.

CHAPTER FIVE

5.0 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary and Conclusions

We have derived a general model calibrated model assited nonparametric estimator for population total and mean as envisaged in the first objective. We have shown that the estimator is design unbiased, consistent and asymptotic normal. The estimator has been shown to perform better than the nonparametric internally calibrated estimator. Any nonparametric smoothing method may be used to fit the missing values. We have considered three specific methods;penalized splines, local polynomial and Nadaraya Watson kernel smoothing.

We have also derived a general model calibrated model assited semiparametric estimator for population total and mean as envisaged in the second objective and derived an estimate for the variance. We have shown that the estimator is design unbiased, consistent and asymptotic normal. The estimator has been shown to perform better than the semiparametric internally calibrated estimator. We have again considered penalized splines, local polynomial and Nadaraya Watson kernel smoothing methods to fit the missing values.

When penalized splines are used to fit the missing values, both model calibrated and internally calibrated estimators performs better than when local polynomial or nadaraya watston smoothing are used. In fact,only when penalized splines were used did we have the internally calibrated estimators having a performance close to that of the design estimator. It is observed that even though using penalized splines results in a more efficient model calibrated or internally calibrated estimator than when kernel based methods are used, an internally calibrated estimator that uses penalized splines is less efficient than a model calibrated estimator that uses kernel based method to fit missing values. Thus, to model calibrate or not is more significant than the choice of the nonparametric method to use to fit the missing values.

By letting our auxilliary information be available at the cluster level and letting the cluster be the sampling units, we have extended our estimation to two stage sampling. The resulting model calibrated estimators(both nonparametric and semiparametric) have been shown to be design unbiased, consistent and asymptotic normal. We have also considered a case where auxilliary information is available at both element and cluster level in which case we have used the various models for the estimation even within the clusters. Even though this has not yielded better results than when design estimation is used for the estimations within clusters, the difference is very small, thus the results are still reliable. Thus, we have shown that in cases where some elements within clusters are unreachable but auxilliary information is available at element level, we can take advantage of this auxilliary information to obtain cluster totals which are then used in the estimation of population total. If there is a possibility that some clusters may be unreachable, it means there is a posibility too that some cluster elements may be unreachable.

When some of the categorical variables are not considered in estimation,the model calibrated semiparametric estimators(for both one stage and two stage sampling) remain robust still having performances close to the performance of the design estimator even though none performed better than the design estimator. The internally calibrated estimators were shown to perform poorly when some of the categorical variables are dropped. In a real world problem where we may not have or may not be sure that we have all the relevant auxilliary information about a variable, model calibrated estimators would therefore be the estimators of choice. The Horvitz Thompson design estimator performs better than all the other estimators considered.

5.2 Recommendations

We would recommend further research on using semiparametric and nonparametric models in the presence of auxilliary information to estimate population parameters like total and mean at any given time for a population that grows with time. That is, population size is time dependent. This is a broad area if we were to think of application of internal calibration,model calibration and model assistance techniques.

Consider two random variables say Y_1 and Y_2 generated by two different systems and suppose they are both depedent on the same set of auxiliary variables(both continuous and categorical), it would be interesting to study the use of nonparametric and semiparametric techniques to model the relationship between the populations of the two random variables, more so if the generating systems are non-terminating.

REFERENCES

Box, G.E.P. (1980). Sampling and Baye's Inference in Scientific Modeling and Robustness. *Journal of the Royal Statistical Society, series A, 143,383-40.*

Breidt, F. J. and Opsomer, J.D. (2000). Local Polynomial Regression Estimation in Survey Sampling. *Annals of Statistics, 28, 1026-1053.*

Breidt, F.J. Claeskens, G. and Opsomer, J.D. (2005). Model Assisted Estimation for Complex Surveys Using Penalized Splines. *Biometrika, 92,831-846.*

Breidt, F. J. Kim, J.Y. and Opsomer, J.D. (2005). Nonparametric Regression Estimation of Finite Population Totals Under Two Stage Sampling. *Annals of Statistics, 25, 1026-1053.*

Breidt, F.J. Opsomer, J.D. Alicia, A.J. and Ranalli, G. (2007). Semiparametric Model Assisted Estimation for Natural Resource Surveys. *Statistics Canada, Catalogue No. 12-001.*

Carpenter, J. and Kenward, M. (2005). Contribution To Discussion of Greenland's 'Multiple-bias Modelling For The Analysis of Observational Data'. *Journal of the royal statistical society, series A,.*

Chen, J. and Sitter, R.R. (1999).A Pseudo Empirical Likelihood Approach to the Effective Use of Auxilliary Information in Complex Surveys. *Statistica Sinica 9,385-406*

Cochran W.G. (1997). Sampling Techniques (3rd ed.). *New York: John Wiley and sons.*

De Boor, C. (2001). A Practical Guide To Splines (Revised Edition). *Springer, New York.*

Deville, J.C. and Sarndal C.E. (1992). Calibration Estimators in Survey Sampling. *Journal of the American Statistical Association 87,376-82.*

Eilers, P.H.C. and Marx B.D. (1996). Flexible Smoothing with B-Splines and Penalties (with discussion). *Statistical Science 11, 89-121.*

Firth, D. And Bennet, K.E. (2006). Robust Models in Probability Sampling. *Journal of Royal Statistical Society. B, 17, 267-278.*

Flanders, W.D. and Greenland, S. (1991). Logistic Analysis of Studies With Two Stage Sampling: A Comparison of Four Approaches. *stat med :Jan 15-Feb 15,16(1-3), 117-32.*

Hastie, T. and Tibshirani, R. (1987). Generalized Addidative Models: Some Applications. *Journal of the American Statistical Association, 82,371-386.*

Jiang, J. (1996). REML estimation: Asymptotic Behavior and Related Topics. *Ann. Statist. 24(1), 255-286.*

Kott, P.S. (2003). On Calibration Weighting. *Journal of Official Statistics. 16,379-399*

Little, R. and Rubin, D. (1987). Statistical Analysis with Missing Data. *New York: John Wiley.*

Luke Keele,(2008).Semiparametric Regression for the Social Sciences. *New York: John Wiley and Sons.*

Montanari, G.E. and Ranalli, M.G. (2003). Nonparametric Model Calibration Estimation in Survey Sampling. *Journal of American Statistical Association. 100, 1429-1442.*

Nadaraya, E.A. (1964). On Estimation Regression. *Theory of Probability and its Applications, 9,142-142.*

Otieno, R.O, Mwita, P. and Kihara, P.N. (2007). Nonparametric Model Assisted Model Calibrated Estimation in Two Stage Survey Sampling. *The East African Journal of Statistics, Vol 1, No.3, 261-281.*

Rubin,D.B. Schenker, N.(1986). Multiple Imputation for Nonresponse in Surveys.*New York: John Wiley and Sons.*

Rupert,D. Wand, M.P. and Carroll, R.J.(2003). Semiparametric Regression.*New York: John Wiley and Sons.*

Sarndal, C.E. (1980). On -Inverse weighting versus best Linear Unbiased Weighting in Probability Sampling. *Biometrika, 67,639-650.*

Scharfstein, D.O. Daniels, M.J. and Robins, J.M. (2003). Incorporatimg Prior Beliefs About Selection Bias Into The Analysis of Randomized Trials With Missing Outcomes. *Biostatistics,4,4, 495-512.*

Silverman, B.W. (1985). Some Aspects of the Spline Smoothing Approach to Nonparametric Regression Curve Fitting. *Journal of the royal statistical society, series B, 47, 1-21.*

Simonoff, J. (1996). Smoothing Methods in Statistics. *New York: Springer.*

Stone, C.J.(1986). The Dimensionality Reduction Principle for Generalized Additive Models. *Annals of Statistics, 14,590-606.*

Tanner, M.A. and Wong,W.H.(1987).The Calculation of Posterior Distributions by Data Augmentation. *Journal of American Statistical Association, 82, 528-540.*

Thompson, M.E. (1997). Theory of Sample Surveys. *Chapman Hall, London*

Watson, G. (1964). Smooth Regression Analysis. *Sankya A, 26,359-72.*

Wu, C, and Sitter, R.R. (2001). A Model Calibration Approach to Using Complete Auxiliary Information from Survey Data. *Journal of American Statistical Association, 96, 185-93.*

Zhao, L.P. and Lipsitz, S. (1992). Design and Analysis of Two Stage Studies. *Statistics in modeling, 11,769-782.*

Appendix 1: Proofs

Proof of Theorem 1

Firstly, due to the constraint $\sum_{i\in s} w_i = N$ and the fact that w_i is chosen to be as close as possible to d_i, then $\sum_{i\in s} d_i$ is also very close to $\sum_{i\in s} w_i$ when N is large. In fact sometimes $\sum_{i\in s} d_i$ is used to estimate the population size N in the constraints (1.5). See Wu and sitter,(2001). The term $N - \sum_{i\in s} d_i$ is therefore the error term in the estimation of N. Under SRSWOR, $\sum_{i\in s} d_i = N$. In our analysis where we have assumed equal probability sampling, the term reduces to zero.

Secondly, by theorem 1 of Chen and Sitter(1999), if L_c is a set of all constant sequences $C = \{c_1, c_2, ...\}$ such $N^{-1}\sum_{i=1}^{N}(c_i - \overline{C}_N)^2 \to c \neq 0$ as $N \to \infty$ where $\overline{C}_N = N^{-1}\sum_{i=1}^{N} c_i$ then we have the following mean estimator for a population of unknown size obtained by minimizing a chi-square distance measure.

$$\hat{\bar{y}}_{EC} = \frac{\sum_{i\in s} d_i y_i}{\sum_{i\in s} d_i} + \left\{ \frac{\sum_{i\in U} c_i}{N} - \frac{\sum_{i\in s} d_i c_i}{\sum_{i\in s} d_i} \right\} \hat{\beta}_{mc} + O_p(\frac{1}{\sqrt{n}}) \tag{5.1}$$

where $\hat{\beta}_{mc}$ is defined as $\hat{\beta}_m$ but with $q_i = 1$. But since $\sum_{i\in s} d_i$ is an estimator for N, by modifying the theorem and letting $g_i \in L_c$, one has that

$$\hat{y}_{sm} = \sum_{i\in s} \frac{y_i}{\pi_i} + \left\{ \sum_{i\in U} \hat{g}_i - \sum_{i\in s} \frac{\hat{g}_i}{\pi_i} \right\} \hat{\beta}_m + O_p(\frac{N}{\sqrt{n}}) \tag{5.2}$$

Thirdly, under an equal probability sampling scheme in which $\sum_{i\in s} d_i$ is unbiased for N, we have from condition (a) of assumption 6 in section (3.2.1) that $\sum_{i\in s} d_i = O(N)$.

Suppose we let $q_i = 1$, and $\hat{g}_i$ be unbiased for y_i. By arguments similar to lemma 4 of Breidt et al (2000), lemma 4 of Montanari and Ranalli (2003) and condition (b) of assumption 6 in section (3.2.1), $\frac{\sum_{i\in s} d_i q_i y_i}{\sum_{i\in s} d_i q_i} - \hat{\beta}_m = O_p(\frac{1}{\sqrt{n}})$ and hence $(N - \sum_{i\in s} d_i)(\frac{\sum_{i\in s} d_i q_i y_i}{\sum_{i\in s} d_i q_i} - \hat{\beta}_m) = O_p(\frac{N}{\sqrt{n}})$

Appendix 2: Sample Normal Graphs

Graphs for Spline Model Calibrated Estimator

Nonparametric one stage Estimation (cycle 1 function)

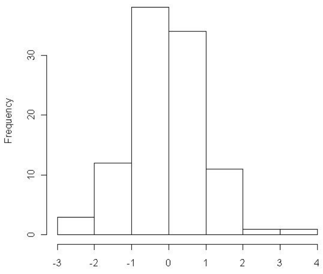

Figure 5.1: Spline Graph for Population ρ_1(nonparametric one stage)

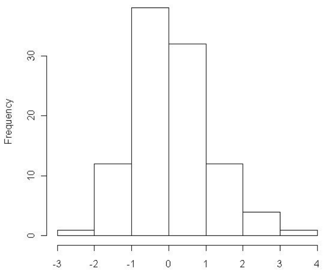

Figure 5.2: Spline Graph for Population ρ_2(nonparametric one stage)

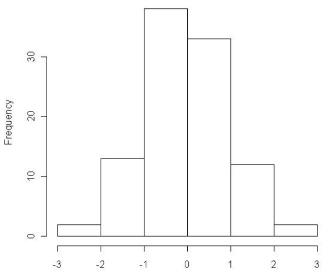

Figure 5.3: Spline Graph for Population ρ_3(nonparametric one stage)

Semiparametric one stage Estimation (cycle 1 function)

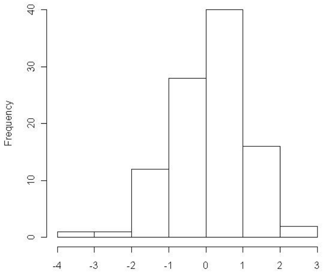

Figure 5.4: Spline Graph for Population ρ_1(semiparametric one stage)

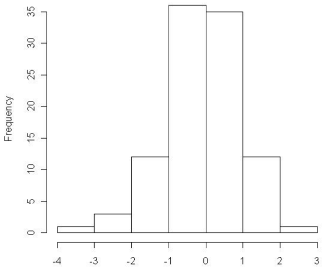

Figure 5.5: Spline Graph for Population ρ_2(semiparametric one stage)

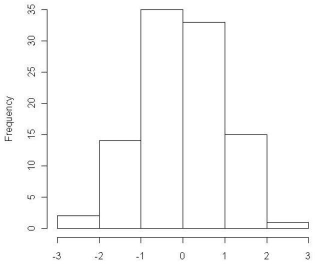

Figure 5.6: Spline Graph for Population ρ_3(semiparametric one stage)

Two stage Estimation (cycle 1 function)

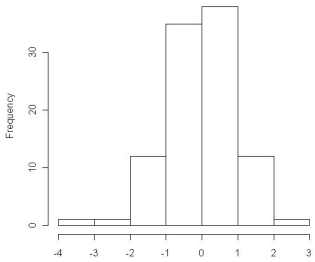

Figure 5.7: Spline Graph for Population ρ_1(semiparametric two stage)

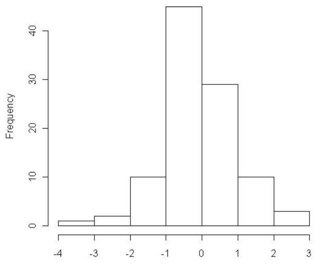

Figure 5.8: Spline Graph for Population ρ_2(semiparametric two stage)

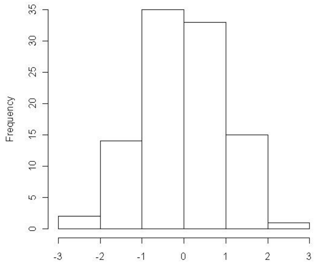

Figure 5.9: Spline Graph for Population ρ_3(semiparametric two stage)

Graphs for Local Polynomial Model Calibrated Estimator

Nonparametric one stage Estimation (cycle 1 function)

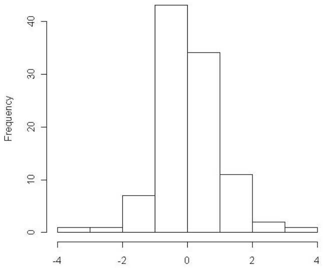

Figure 5.10: Local Polynomial Graph for Population ρ_1(nonparametric one stage)

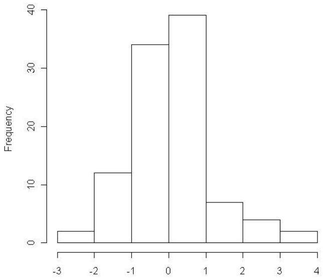

Figure 5.11: Local Polynomial Graph for Population ρ_2(nonparametric one stage)

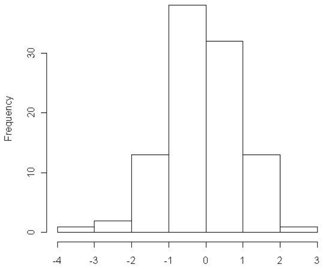

Figure 5.12: Local Polynomial Graph for Population ρ_3(nonparametric one stage)

Semiparametric one stage Estimation (cycle 1 function)

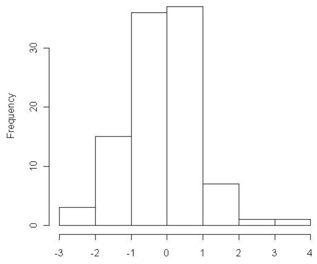

Figure 5.13: Local Polynomial Graph for Population ρ_1(semiparametric one stage)

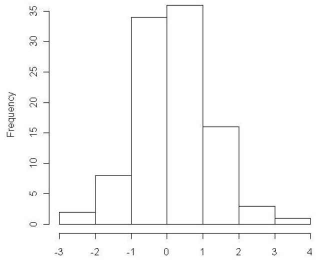

Figure 5.14: Local Polynomial Graph for Population ρ_2(semiparametric one stage)

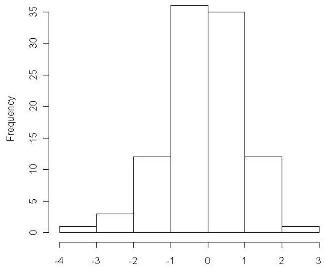

Figure 5.15: Local Polynomial Graph for Population ρ_3(semiparametric one stage)

Two stage Estimation (cycle 1 function)

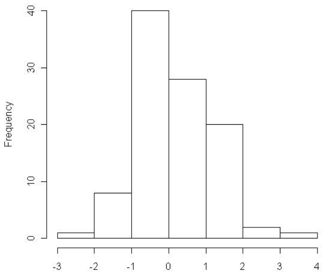

Figure 5.16: Local Polynomial Graph for Populationρ_1(semiparametric two stage)

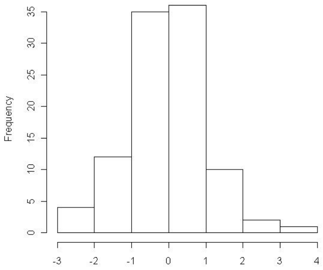

Figure 5.17: Local Polynomial Graph for Population ρ_2(semiparametric two stage)

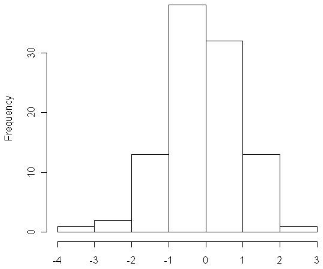

Figure 5.18: Local Polynomial Graph for Population ρ_3(semiparametric two stage)

Appendix 3: Code for Nonparametric Estimation

```
unipop<-data.frame(runif(300)
names(unipop)<-"a"
#unipop
xpop<-c((unipop$a))
ypop<-2+sin(4*360*xpop)
#ypop
popsize<-300
samplesize<-0.1*popsize
samplenumber<-10
knotsnumber<-4
knotsmatrix<-c(0.2,0.4,0.6,0.8)
#polynomialdegree
q<-3
h<-0.5
inclusionprobability<-samplesize/popsize
di<-1/inclusionprobability
qi<-1
types_of_estimators<-7
estimatorsmatrix<-matrix(0,types_of_estimators,samplenumber)
relative_biase_matrix<-matrix(0,1,types_of_estimators)
relative_mse_matrix<-matrix(0,1,types_of_estimators)
absolute_biase_matrix<-matrix(0,1,types_of_estimators)
bias_numerator_matrix<-matrix(0,types_of_estimators,samplenumber)
mean_squared_error_matrix<-matrix(0,types_of_estimators,1)
```

```
localpolynomial_x_matrix<-matrix(1,samplesize,1)
locallinear_x_matrix<-matrix(1,samplesize,2)
e<-c(1,0)
polynomial_diagonal_matrix<-matrix(0,samplesize,samplesize)
for(r in 1:samplenumber)
{ similar_de<-0
mu_spline_nu<-0
y_nu<-0
mu_local_nu<-0
mu_linear_nu<-0
q<-3
xsample<-sample(unipop$a,samplesize)
ysample<-2+sin(4*360*xsample)
splines_sample_mean_estimates<-matrix(0,1,samplesize)
splines_pop_mean_estimates<-matrix(0,1,popsize)
penaltymatrix<-matrix(0,1+q+knotsnumber,1+q+knotsnumber)
samplemodelmatrix<-matrix(0,samplesize,1+q+knotsnumber)
popmodelmatrix<-matrix(0,popsize,1+q+knotsnumber)
i<-0
#elements model matrix
for(x in xsample)
{
i<-i+1
ith_x_samplemodelmatrix<-c(1,x,x^2,x^3,(x-0.2)^q,(x-0.4)
^q,(x-0.6)^q,(x-0.8)^q)
samplemodelmatrix[i,]<-ith_x_samplemodelmatrix
}
#penalty matrix and setting -ve values in model matrix to zero
```

```
for(t in (2+q):(1+q+knotsnumber))
{
penaltymatrix[t,t]<-1600*(samplesize/4)^4
for(z in 1:samplesize)
{
if (samplemodelmatrix[z,t]<=0) samplemodelmatrix[z,t]<-0
}
}
#print(samplemodelmatrix)
bu<-solve(t(samplemodelmatrix)%*%samplemodelmatrix
+penaltymatrix)%*%t(samplemodelmatrix)%*%ysample
#spline sample means estimates
p<-0
for(y in ysample)
        {
p<-p+1
splines_sample_mean_estimates[1,p]<-
t(samplemodelmatrix[p,])%*%bu
#some regression step values
y_nu<-(y_nu)+di*qi*y
similar_de<-(similar_de)+di*qi
mu_spline_nu<-(mu_spline_nu)+
di*qi*splines_sample_mean_estimates[1,p]
#print(similar_de)
}
#print(similar_de)
y_avg<-(y_nu)/(similar_de)
spline_mu_avg<-(mu_spline_nu)/(similar_de)
```

```
#the regression step bmc
spline_bmc_nu<-0
spline_bmc_de<-0
for(v in 1:samplesize)
{
spline_bmc_nu<-(spline_bmc_nu)+di*qi*
(splines_sample_mean_estimates[1,v]-spline_mu_avg)*
(ysample[v]-(y_avg))
spline_bmc_de<-(spline_bmc_de)+di*qi*
(splines_sample_mean_estimates[1,v]-spline_mu_avg)^2
}
spline_bmc<-(spline_bmc_nu)/(spline_bmc_de)
#print(spline_bmc)
#element model matrix for entire population
j<-0
for(x in xpop)
{j<-1+j
ith_x_popmodelmatrix<-c(1,x,x^2,x^3,(x-0.2)^q,
(x-0.4)^q,(x-0.6)^q,(x-0.8)^q)
popmodelmatrix[j,]<-ith_x_popmodelmatrix
}
#setting -ve values in model matrix to zero
for(t in (2+q):(1+q+knotsnumber))
{
for(z in 1:popsize)
{
if (popmodelmatrix[z,t]<=0) popmodelmatrix[z,t]<-0
}
```

```
}
#population mean estimates
for(y in 1:popsize)
        {
splines_pop_mean_estimates[1,y]<-t(popmodelmatrix[y,])%*%bu

}
#polynomial estimations
# kernel, x matrix ,and mean estmates for the sample
polynomial_sample_mean_estimates<-matrix(0,2,samplesize)
for(xi in 1:samplesize)
{
for(xj in 1:samplesize)
{
polynomial_diagonal_matrix[xj,xj]<-(3/4)*(1-((xsample[xj]-
xsample[xi])/h)^2)locallinear_x_matrix[xj,2]<-xsample[xj]-
xsample[xi]
}
polynomial_sample_mean_estimates[1,xi]<-
solve(t(localpolynomial_x_matrix)%*%(polynomial_diagonal_
matrix)%*%(localpolynomial_x_matrix))%*
%t(localpolynomial_x_matrix)%*
%(polynomial_diagonal_matrix)%*%ysample
polynomial_sample_mean_estimates[2,xi]<-t(e)%*
%solve(t(locallinear_x_matrix)%*%(polynomial_
diagonal_matrix)%*%(locallinear_x_matrix))%*
%t(locallinear_x_matrix)%*%
(polynomial_diagonal_matrix)%*%ysample
```

```
#print("y")
#print(ysample[xi])
#print("est")
#print(polynomial_sample_mean_estimates[1,xi])
#print(polynomial_sample_mean_estimates[2,xi])
}
mu_local_nu<-sum(polynomial_sample_mean_estimates[1,]
*di*qi)
mu_linear_nu<-sum(polynomial_sample_mean_estimates[2,]
*di*qi)
local_mu_avg<-(mu_local_nu)/similar_de
linear_mu_avg<-(mu_linear_nu)/similar_de
#print(mu_local_nu)
#print(mu_linear_nu)
#print(local_mu_avg)
#print(linear_mu_avg)
#print("anpter")
local_bmc_nu<-0
local_bmc_de<-0
linear_bmc_nu<-0
linear_bmc_de<-0
for(w in 1:samplesize)
{
local_bmc_nu<-local_bmc_nu+di*qi*(polynomial_sample_
mean_estimates[1,w]-local_mu_avg)*(ysample[w]-y_avg)
local_bmc_de<-local_bmc_de+di*qi*(polynomial_sample_mean_
estimates[1,w]-local_mu_avg)^2
linear_bmc_nu<-linear_bmc_nu+di*qi*(polynomial_sample_
mean_estimates[2,w]-linear_mu_avg)*(ysample[w]-y_avg)
linear_bmc_de<-linear_bmc_de+di*qi*(polynomial_sample_
mean_estimates[2,w]-linear_mu_avg)^2
}
local_bmc<-local_bmc_nu/local_bmc_de
linear_bmc<-linear_bmc_nu/linear_bmc_de
#print(spline_bmc)
#print(local_bmc)
#print(linear_bmc)
# kernel, x matrix ,and mean estmates for the entire population
polynomial_pop_mean_estimates<-matrix(0,2,popsize)
for(xi in 1:popsize)
{
for(xj in 1:samplesize)
{
polynomial_diagonal_matrix[xj,xj]<-(3/4)*(1-((xsample[xj]-
xpop[xi])/h)^2)locallinear_x_matrix[xj,2]<-xsample[xj]-xpop[xi]
}
polynomial_pop_mean_estimates[1,xi]<-
solve(t(localpolynomial_x_matrix)%*%(polynomial_
diagonal_matrix)%*%(localpolynomial_x_matrix))%*
%t(localpolynomial_x_matrix)%*
%(polynomial_diagonal_matrix)%*%ysample
polynomial_pop_mean_estimates[2,xi]<-
t(e)%*%solve(t(locallinear_x_matrix)%*%
(polynomial_diagonal_matrix)%*
%(locallinear_x_matrix))%*%t(locallinear_x_matrix)%*
%(polynomial_diagonal_matrix)%*%ysample
```

```
}
#population total estimation
ht_total<-sum(ysample*di)
actual_total<-sum(ypop)
estimatorsmatrix[1,r]<-ht_total+(sum(splines_pop_mean_
estimates)-sum(splines_sample_mean_estimates*di))
*spline_bmc
estimatorsmatrix[2,r]<-ht_total+(sum(splines_pop_mean_
estimates)-sum(splines_sample_mean_estimates*di))
estimatorsmatrix[3,r]<-ht_total+(sum(polynomial_pop_mean_
estimates[1,])-sum(polynomial_sample_mean_estimates[1,]
*di))*local_bmc
estimatorsmatrix[4,r]<-ht_total+(sum(polynomial_pop_mean_
estimates[1,])-sum(polynomial_sample_mean_estimates[1,]*di))
estimatorsmatrix[5,r]<-ht_total+(sum(polynomial_pop_mean_
estimates[2,])-sum(polynomial_sample_mean_estimates[2,]*di))
*linear_bmc
estimatorsmatrix[6,r]<-ht_total+(sum(polynomial_pop_mean_
estimates[2,])-sum(polynomial_sample_mean_estimates[2,]*di))
estimatorsmatrix[7,r]<-ht_total
#bias_numerator_matrix[1,r]<-spline_total-actual_total
#bias_numerator_matrix[2,r]<-ht_total-actual_total
}
print(actual_total)
print(estimatorsmatrix)
#average_absolute biases rb
absolute_biase_matrix[1,1]<-sum(abs(estimatorsmatrix[1,]
-actual_total))/(samplenumber*actual_total)
absolute_biase_matrix[1,2]<-sum(abs(estimatorsmatrix[2,]
-actual_total))/(samplenumber*actual_total)
absolute_biase_matrix[1,3]<-sum(abs(estimatorsmatrix[3,]
-actual_total))/(samplenumber*actual_total)
absolute_biase_matrix[1,4]<-sum(abs(estimatorsmatrix[4,]
-actual_total))/(samplenumber*actual_total)
absolute_biase_matrix[1,5]<-sum(abs(estimatorsmatrix[5,]
-actual_total))/(samplenumber*actual_total)
absolute_biase_matrix[1,6]<-sum(abs(estimatorsmatrix[6,
-actual_total))/(samplenumber*actual_total)
absolute_biase_matrix[1,7]<-sum(abs(estimatorsmatrix[7,]
-actual_total))/(samplenumber*actual_total)
#relative biases
relative_biase_matrix[1,1]<-sum(estimatorsmatrix[1,]
-actual_total)/(samplenumber*actual_total)
relative_biase_matrix[1,2]<-sum(estimatorsmatrix[2,]
-actual_total)/(samplenumber*actual_total)
relative_biase_matrix[1,3]<-sum(estimatorsmatrix[3,]
-actual_total)/(samplenumber*actual_total)
relative_biase_matrix[1,4]<-sum(estimatorsmatrix[4,]
-actual_total)/(samplenumber*actual_total)
relative_biase_matrix[1,5]<-sum(estimatorsmatrix[5,]
-actual_total)/(samplenumber*actual_total)
relative_biase_matrix[1,6]<-sum(estimatorsmatrix[6,]
-actual_total)/(samplenumber*actual_total)
relative_biase_matrix[1,7]<-sum(estimatorsmatrix[7,]
-actual_total)/(samplenumber*actual_total)
```

```
#relative mse’s
relative_mse_matrix[1,1]<-(var(estimatorsmatrix[1,])+
(relative_biase_matrix[1,1])^2)
/(var(estimatorsmatrix[1,])+(relative_biase_matrix[1,1])^2)
relative_mse_matrix[1,2]<-(var(estimatorsmatrix[2,])+
(relative_biase_matrix[1,2])^2)
/(var(estimatorsmatrix[1,])+(relative_biase_matrix[1,1])^2)
relative_mse_matrix[1,3]<-(var(estimatorsmatrix[3,])+
(relative_biase_matrix[1,3])^2)/
(var(estimatorsmatrix[1,])+(relative_biase_matrix[1,1])^2)
relative_mse_matrix[1,4]<-(var(estimatorsmatrix[4,])+
(relative_biase_matrix[1,4])^2)/
(var(estimatorsmatrix[1,])+(relative_biase_matrix[1,1])^2)
relative_mse_matrix[1,5]<-(var(estimatorsmatrix[5,])+
(relative_biase_matrix[1,5])^2)/
(var(estimatorsmatrix[1,])+(relative_biase_matrix[1,1])^2)
relative_mse_matrix[1,6]<-(var(estimatorsmatrix[6,])+
(relative_biase_matrix[1,6])^2)/
(var(estimatorsmatrix[1,])+(relative_biase_matrix[1,1])^2)
relative_mse_matrix[1,7]<-(var(estimatorsmatrix[7,])+
(relative_biase_matrix[1,7])^2)/
(var(estimatorsmatrix[1,])+(relative_biase_matrix[1,1])^2)
#output
print(relative_biase_matrix)
print(absolute_biase_matrix)
print(relative_mse_matrix)
shapiro.test(estimatorsmatrix[1,])
shapiro.test(estimatorsmatrix[2,])
shapiro.test(estimatorsmatrix[3,])
shapiro.test(estimatorsmatrix[4,])
shapiro.test(estimatorsmatrix[5,])
shapiro.test(estimatorsmatrix[6,])
shapiro.test(estimatorsmatrix[7,])
```

Appendix 4: Code for Semiparametric Estimation

```
popsize<-300
unipop<-data.frame(runif(popsize))
names(unipop)<-"a"
z1<-rep(c(2),popsize)
z2<-rep(c(3,4,5),popsize/3)
z3<-rep(c(6,7,8),c(popsize/3,popsize/3,popsize/3))
linearz<-matrix(0,popsize,3)
linearmatrix<-matrix(0,popsize,1)
indepedentvariables<-matrix(0,popsize,4)
linearz[,1]<-z1
linearz[,2]<-z2
linearz[,3]<-z3
linregressor<-c(1,2,3)
linearmatrix<-linearz%*%(linregressor)
xpop<-c((unipop$a))
indepedentvariables[,1]<-z1
indepedentvariables[,2]<-z2
indepedentvariables[,3]<-z3
```

```
indepedentvariables[,4]<-xpop

ypop<-linearz%*%(linregressor)+(2+5*xpop)^2


samplesize<-0.1*popsize

samplenumber<-10

knotsnumber<-4

knotsmatrix<-c(0.2,0.4,0.6,0.8)

#polynomialdegree

q<-3

h<-0.5

inclusionprobability<-samplesize/popsize

di<-1/inclusionprobability

qi<-1

types_of_estimators<-4

estimatorsmatrix<-matrix(0,types_of_estimators,samplenumber)

relative_biase_matrix<-matrix(0,1,types_of_estimators)

relative_mse_matrix<-matrix(0,1,types_of_estimators)

absolute_biase_matrix<-matrix(0,1,types_of_estimators)

bias_numerator_matrix<-matrix(0,types_of_estimators,samplenumber)

mean_squared_error_matrix<-matrix(0,types_of_estimators,1)

localpolynomial_x_matrix<-matrix(1,samplesize,1)

locallinear_x_matrix<-matrix(1,samplesize,2)

linear_smoother_matrix<-matrix(0,samplesize,samplesize)

local_smoother_matrix<-matrix(0,samplesize,samplesize)

linear_pop_smoother_matrix<-matrix(0,popsize,samplesize)

local_pop_smoother_matrix<-matrix(0,popsize,samplesize)

e<-c(1,0)

polynomial_diagonal_matrix<-matrix(0,samplesize,samplesize)
```

```
for(r in 1:samplenumber)
{       similar_de<-0
                mu_spline_nu<-0
                y_nu<-0
                mu_local_nu<-0
                mu_linear_nu<-0


        q<-3
        indsample<-indepedentvariables[sample(popsize,
        samplesize,replace=FALSE),]
        #print(indsample)
        ysample<-indsample[,1:3]%*%(linregressor)+
        (2+5*indsample[,4])^2
        splines_sample_mean_estimates<-matrix(0,1,samplesize)
        splines_pop_mean_estimates<-matrix(0,1,popsize)
        penaltymatrix<-matrix(0,1+q+knotsnumber,1+q+knotsnumber)
        samplemodelmatrix<-matrix(0,samplesize,1+q+knotsnumber)
        popmodelmatrix<-matrix(0,popsize,1+q+knotsnumber)

        #print(ysample)


        # kernel, x matrix ,and mean estmates for the sample
        polynomial_sample_mean_estimates<-matrix(0,2,samplesize)
        for(xi in 1:samplesize)
        {
        for(xj in 1:samplesize)
                {
        polynomial_diagonal_matrix[xj,xj]<-(3/4)*(1-
        ((indsample[,4][xj]-indsample[,4][xi])/h)^2)
        locallinear_x_matrix[xj,2]<-indsample[,4][xj]
        -indsample[,4][xi]
                }
        loc<-solve(t(localpolynomial_x_matrix)%*%(polynomial_
        diagonal_matrix)%* %(localpolynomial_x_matrix))%*
        %t(localpolynomial_x_matrix)%*%(polynomial_diagonal_
        matrix)
        lin<-t(e)%*%solve(t(locallinear_x_matrix)
        %*%(polynomial_diagonal_matrix)%*%(locallinear_x_matrix))
        %*%t(locallinear_x_matrix)%*%(polynomial_diagonal_matrix)
        local_smoother_matrix[xi,]<-loc
        linear_smoother_matrix[xi,]<-lin
                }
        linsmooth<-1-(di/popsize)*(linear_smoother_matrix)
        locsmooth<-1-(di/popsize)*(local_smoother_matrix)
        linbhat<-solve(t(indsample[,1:3])%*%(1-linsmooth)%*
        %(indsample[,1:3]))%*%t(indsample[,1:3])%*%(1-linsmooth)%
        *%ysample
        locbhat<-solve(t(indsample[,1:3])%*%(1-locsmooth)%*
        %(indsample[,1:3]))%*%t(indsample[,1:3])%*%(1-locsmooth)%*
        %ysample
        #sample mks
        linmks<-linear_smoother_matrix%*%(ysample-(indsample[,1:3])
        %*%linbhat)
        locmks<-local_smoother_matrix%*%(ysample-(indsample[,1:3])
        %*%locbhat)
        #sample means
```

```
        lingk<-linmks+(indsample[,1:3])%*%linbhat
        locgk<-locmks+(indsample[,1:3])%*%locbhat
        #print(lingk)
        #print(locgk)
        for(xi in 1:popsize)
        {
                for(xj in 1:samplesize)
                {
                polynomial_diagonal_matrix[xj,xj]<-(3/4)*(1-
                ((xsample[xj]-xpop[xi])/h)^2)
                locallinear_x_matrix[xj,2]<-xsample[xj]-
                indepedentvariables[,4][xi]
                }
poploc<-solve(t(localpolynomial_x_matrix)%*
%(polynomial_diagonal_matrix)%*%(localpolynomial_x_matrix))%*
%t(localpolynomial_x_matrix)%*%(polynomial_diagonal_matrix)
poplin<-t(e)%*%solve(t(locallinear_x_matrix)%*
%(polynomial_diagonal_matrix)%*%(locallinear_x_matrix))%*
%t(locallinear_x_matrix)%*%(polynomial_diagonal_matrix)
                local_pop_smoother_matrix[xi,]<-poploc
                linear_pop_smoother_matrix[xi,]<-poplin
        }
        #population mks
        poplinmks<-linear_pop_smoother_matrix%*%(ysample-
        (indsample[,1:3])%*%linbhat)
        poplocmks<-local_pop_smoother_matrix%*%(ysample-
        (indsample[,1:3])%*%locbhat)
        #population means
```

```
poplingk<-poplinmks+(indepedentvariables[,1:3])%*%linbhat
poplocgk<-poplocmks+(indepedentvariables[,1:3])%*%locbhat
 #estimating bmc
        p<-0
        for(y in ysample)
{
        p<-p+1
        y_nu<-(y_nu)+di*qi*y
        similar_de<-(similar_de)+di*qi
        }
y_avg<-(y_nu)/(similar_de)
mu_local_nu<-sum(locgk*di*qi)
mu_linear_nu<-sum(lingk*di*qi)
local_mu_avg<-(mu_local_nu)/similar_de
linear_mu_avg<-(mu_linear_nu)/similar_de
local_bmc_nu<-0
local_bmc_de<-0
linear_bmc_nu<-0
linear_bmc_de<-0
local_bmc_nu<-di*qi*t(locgk-local_mu_avg)%*%(ysample-y_avg)
        local_bmc_de<-di*qi*(locgk-local_mu_avg)^2
        linear_bmc_nu<-di*qi*t(lingk-linear_mu_avg)%*
        %(ysample-y_avg)
        linear_bmc_de<-di*qi*(lingk-linear_mu_avg)^2
        local_bmc<-sum(local_bmc_nu)/sum(local_bmc_de)
        linear_bmc<-sum(linear_bmc_nu)/sum(linear_bmc_de)
        #print(local_bmc)
        #print(linear_bmc)
        i<-0
        #elements model matrix
        for(x in indsample[,4])
        {
        i<-i+1
       ith_x_samplemodelmatrix<-c(1,x,x^2,x^3,(x-0.2)^q,
       (x-0.4)^q,(x-0.6)^q,(x-0.8)^q)
        samplemodelmatrix[i,]<-ith_x_samplemodelmatrix
        }
        #penalty matrix and
        #setting -ve values in model matrix to zero
        for(t in (2+q):(1+q+knotsnumber))
        {
        penaltymatrix[t,t]<-1600*(samplesize/4)^4
        for(z in 1:samplesize)
                {
                 if (samplemodelmatrix[z,t]<=0)
                 samplemodelmatrix[z,t]<-0
                }

        }
samplebsk<-solve(t(samplemodelmatrix)%*
%(samplemodelmatrix)
+penaltymatrix)%*%t(samplemodelmatrix)
spline_smoother_matrix<-samplemodelmatrix%*
%samplebsk
splinesmooth<-1-(di/popsize)*(spline_smoother_matrix)
```

```
        splinebhat<-solve(t(indsample[,1:3])%*%(1-splinesmooth)%*
        %(indsample[,1:3]))%*%t(indsample[,1:3])%*
        %(1-splinesmooth)%*%ysample
        #print(splinebhat)
        splinemks<-spline_smoother_matrix%*%(ysample-
        (indsample[,1:3])%*%splinebhat)
        splinegk<-splinemks+(indsample[,1:3])%*%splinebhat



                #some regression step values
                mu_spline_nu<-sum(di*qi*splinegk)
                y_avg<-(y_nu)/(similar_de)
        spline_mu_avg<-(mu_spline_nu)/(similar_de)
        #the regression step bmc
        spline_bmc_nu<-0
        spline_bmc_de<-0
        spline_bmc_nu<-di*qi*t(splinegk-spline_mu_avg)%*
        %(ysample-y_avg)
        spline_bmc_de<-di*qi*(splinegk-spline_mu_avg)^2
        spline_bmc<-sum(spline_bmc_nu)/sum(spline_bmc_de)
        #print(spline_bmc)
#element model matrix for entire population
        j<-0
        for(x in indepedentvariables[,4])
                {j<-1+j
                ith_x_popmodelmatrix<-c(1,x,x^2,x^3,(x-0.2)^q,
                (x-0.4)^q,(x-0.6)^q,(x-0.8)^q)
                popmodelmatrix[j,]<-ith_x_popmodelmatrix
                }
                #setting -ve values in model matrix to zero
                for(t in (2+q):(1+q+knotsnumber))
                {
                for(z in 1:popsize)
                        {
                         if (popmodelmatrix[z,t]<=0)
                         popmodelmatrix[z,t]<-0
                        }


                }
        popbsk<-solve(t(popmodelmatrix)%*%(popmodelmatrix)+
        penaltymatrix)%*%t(popmodelmatrix)
        spline_pop_smoother_matrix<-popmodelmatrix%*%samplebsk
        #print(spline_pop_smoother_matrix)
        popsplinemks<-spline_pop_smoother_matrix%*%(ysample-
        (indsample[,1:3])%*%splinebhat)
        popsplinegk<-popsplinemks+(indepedentvariables[,1:3])
        %*%splinebhat
        #poplinmks<-linear_pop_smoother_matrix%*%(ysample-
        (indsample[,1:3])%*%linbhat)
        ht<-sum(ysample*di)
        actual_total<-sum(ypop)
        estimatorsmatrix[1,r]<- ht+(sum(popsplinegk)-
        sum(splinegk*di))*spline_bmc
        estimatorsmatrix[2,r]<-ht+(sum(poplingk)-
        sum(lingk*di))*linear_bmc
        estimatorsmatrix[3,r]<-ht+(sum(poplocgk)-
        sum(locgk*di))*local_bmc
        estimatorsmatrix[4,r]<-ht
}
print(actual_total)
print(estimatorsmatrix)
#average absolute biases
absolute_biase_matrix[1,1]<-sum(abs(estimatorsmatrix[1,]
-actual_total))/(samplenumber*actual_total)
absolute_biase_matrix[1,2]<-sum(abs(estimatorsmatrix[2,]
-actual_total))/(samplenumber*actual_total)
absolute_biase_matrix[1,3]<-sum(abs(estimatorsmatrix[3,]
-actual_total))/(samplenumber*actual_total)
absolute_biase_matrix[1,4]<-sum(abs(estimatorsmatrix[4,]
-actual_total))/(samplenumber*actual_total)
#relative biases
relative_biase_matrix[1,1]<-sum(estimatorsmatrix[1,]
-actual_total)/(samplenumber*actual_total)
relative_biase_matrix[1,2]<-sum(estimatorsmatrix[2,]
-actual_total)/(samplenumber*actual_total)
relative_biase_matrix[1,3]<-sum(estimatorsmatrix[3,]
-actual_total)/(samplenumber*actual_total)
relative_biase_matrix[1,4]<-sum(estimatorsmatrix[4,]
-actual_total)/(samplenumber*actual_total)
#relative mse’s
relative_mse_matrix[1,1]<-(var(estimatorsmatrix[1,])+
(relative_biase_matrix[1,1])^2)/(var(estimatorsmatrix
[4,])+(relative_biase_matrix[1,4])^2)
relative_mse_matrix[1,2]<-(var(estimatorsmatrix[2,])+
(relative_biase_matrix[1,2])^2)/(var(estimatorsmatrix
[4,])+(relative_biase_matrix[1,4])^2)
relative_mse_matrix[1,3]<-(var(estimatorsmatrix[3,])+
(relative_biase_matrix[1,3])^2)/(var(estimatorsmatrix
[4,])+(relative_biase_matrix[1,4])^2)
relative_mse_matrix[1,4]<-(var(estimatorsmatrix[4,])+
(relative_biase_matrix[1,4])^2)/(var(estimatorsmatrix
[4,])+(relative_biase_matrix[1,4])^2)
#output
print(relative_biase_matrix)
print(absolute_biase_matrix)
print(var(estimatorsmatrix[1,]))
print(var(estimatorsmatrix[2,]))
print(var(estimatorsmatrix[3,]))
print(var(estimatorsmatrix[4,]))
print(relative_mse_matrix)
shapiro.test(estimatorsmatrix[1,])
shapiro.test(estimatorsmatrix[2,])
shapiro.test(estimatorsmatrix[3,])
shapiro.test(estimatorsmatrix[4,])
```

Appendix 5: Code for Two Stage Estimation

```
cnum=200
cpop<-data.frame(c(runif(cnum)))
names(cpop)<-"a"
```

```
csize<-200

sampledclusters<-50

withinsize<-50

variable<-matrix(0,csize,cnum)

variable[1, ]<-c(cpop$a)

sampledvariable<-matrix(0,csize,sampledclusters)

popvariable<-matrix(0,csize,cnum)

httotal<-matrix(0,1,cnum)

poptotals<-matrix(0,2,cnum)

varpopmatrix<-matrix(0,csize,cnum)


poptotals[1, ]<-c(cpop$a)

poptotals[2, ]<-c(httotal)

clusterframe<-data.frame(t(poptotals))

sumdifht<-0

sumlocpol<-0

sumlocpolmc<-0

sumloclin<-0

sumloclinmc<-0

mselocpol<-0

mselocpolmc<-0

mseloclin<-0

mseloclinmc<-0

mseht<-0

varht<-0

varlocpol<-0

varlocpolmc<-0

varloclin<-0
```

```
varloclinmc<-0
pop<-cnum
s<-sampledclusters

xs<-c(cpop$a)
for (j in 1:csize)
{
popvariable[j, ]<-c(((1+2*xs)/csize))


}
poperrvec<-c(rnorm(cnum*csize,0,0.1))
poperrors<-matrix(poperrvec,csize,cnum)
htpoptotal<-matrix(0,1,cnum)

popmatrix<-matrix(0,csize,cnum)
popmatrix<-popvariable+(poperrors/csize^0.25)
for (t in 1:cnum)
{htsum<-0
for(j in 1:csize)
{
htsum<-htsum+popmatrix[j,t]
}
htpoptotal[1,t]<-htsum


}

y<-c(htpoptotal)
pop<-cnum
```

```
s<-sampledclusters

bandwith<-c(0.1,0.25,1,2)

for (b in bandwith)

{

samplesno=5

wubmse<-matrix(0,7,5)

est<-matrix(0,5,samplesno)

for (r in 1:samplesno)

{


samx<-sample(c(cpop$a),s)


for (j in 1:csize)

{

sampledvariable[j, ]<-c(((1+2*samx)/csize))


}

errvec<-c(rnorm(sampledclusters*csize,0,0.1))

errors<-matrix(errvec,csize,sampledclusters)

htsampletotal<-matrix(0,1,sampledclusters)


samplematrix<-matrix(0,csize,sampledclusters)

samplematrix<-sampledvariable+(errors/csize^0.25)

for (t in 1:sampledclusters)

{htsum<-0

for(j in 1:withinsize)

{

htsum<-htsum+samplematrix[j,t]*(csize/withinsize)
}
htsampletotal[1,t]<-htsum


}
samy<-c(htsampletotal)
wui<-matrix(0,s,s)
e<-c(1,0)
xui<-matrix(1,s,2)
xu<-matrix(1,s,1)
totallocalin<-0
totallocpol<-0
totallocalinht<-0
totallocpolht<-0
h<-b
totalp<-0
ht<-0
qi<-1
di<-(pop/s)
dimi<-0
sumdiqi<-0
totaldiqi<-0
diyi<-0
sumnudif<-0
sumdedif<-0
for (i in 1:cnum)
{
tolk<-0
   tolp<-0
```

```
ii<-1
 for (j in 1:s)
   {

      k<-(3/(4*5^0.5))*(1-((((samx[j]-xs[i])/h)^2)/5))

          p=k*samy[j]
      tolk<-tolk+k
        tolp<-tolp+p
        dif<- samx[j]-xs[i]
       wui[ii,ii]<-(k)/h
   xui[ii,2]<-dif
        ii<-ii+1

}
mi<-(tolp/tolk)
milocpol<-solve(t(xu)%*%wui%*%xu)%*%t(xu)%*%wui%*%samy
miloclin<-t(e)%*%solve(t(xui)%*%wui%*%xui)%*%t(xui)%*
%wui%*%samy
totalp<-totalp+mi
totallocalin<-totallocalin+miloclin
totallocpol<-totallocpol+milocpol
}
totalsht<-0

for(ji in 1:s)
{
ii<-1
```

```
     talsamplek<-0
         talsamplep<-0
    for (j in 1:s)
     {
     k<-(3/(4*5^0.5))*(1-((((samx[j]-samx[ji])/h)^2)/5))
      p=k*samy[j]
        talsamplek<-talsamplek+k
         talsamplep<-talsamplep+p
      dif<- samx[j]-samx[ji]
       wui[ii,ii]<-(k)/h
   xui[ii,2]<-dif
        ii<-ii+1
      }
mj<-(talsamplep/talsamplek)
totalsht<-totalsht+di*mj
dimi<-dimi+(di*qi*mj)
    sumdiqi<-sumdiqi+(di*qi)
mjloclin<-t(e)%*%solve(t(xui)%*%wui%*%xui)%*
%t(xui)%*%wui%*%samy
  mjlocpol<-solve(t(xu)%*%wui%*%xu)%*%t(xu)%*
  %wui%*%samy
totallocalinht<-totallocalinht+mjloclin*di
totallocpolht<-totallocpolht+mjlocpol*di
 }
avgm<-(dimi/sumdiqi)
for(l in samy)
{
nu<-(di*qi*l)
```

```
de<-(di*qi)

diyi<-diyi+nu

totaldiqi<-totaldiqi+de

}

avgy<-(diyi/totaldiqi)


for(ji in 1:s )

{

     talsamplenu<-0

         talsamplede<-0

    for (j in 1:s)

     {

     k<-(3/(4*5^0.5))*(1-((((samx[j]-samx[ji])/h)^2)/5))


        p=k*samy[j]

        talsamplenu<-talsamplenu+k

         talsamplede<-talsamplede+p


      }


mj<-(talsamplenu/talsamplede)

nudif<-di*qi*(mj-avgm)*(samy[ji]-avgy)

dedif<-(di*qi*(mj-avgm)*(mj-avgm))

sumdedif<-sumdedif+dedif

sumnudif<-sumnudif+nudif


}
```

```
bmc<-sumnudif/sumdedif
for (l in samy)
{
dy<-(pop/s)*l
ht<-ht+dy
}
ynw<-(ht+(totalp-totalsht))
ylocpol<-(ht+(totallocpol-totallocpolht))
yloclin<-(ht+(totallocalin-totallocalinht))
ylocpolmc<-(ht+(totallocpol-totallocpolht)*bmc)
yloclinmc<-(ht+(totallocalin-totallocalinht)*bmc)

ynwmc<-(ht+(totalp-totalsht)*bmc)
actualp<-sum(y)
est[1,r]<-ht
est[2,r]<-ylocpol
est[3,r]<-ylocpolmc
est[4,r]<-yloclin
est[5,r]<-yloclinmc

sumdifht<-sumdifht+(ht-actualp)/(samplesno*actualp)
sumlocpol<-sumlocpol+(ylocpol-actualp)/
(samplesno*actualp)
sumlocpolmc<-sumlocpolmc+(ylocpolmc-actualp)/
(samplesno*actualp)
sumloclin<-sumloclin+(yloclin-actualp)/
(samplesno*actualp)
sumloclinmc<-sumloclinmc+(yloclinmc-actualp)/
```

```
(samplesno*actualp)
mseht<-mseht+((ht-actualp)^2)/samplesno
mselocpol<-mselocpol+((ylocpol-actualp)^2)/
samplesno
mselocpolmc<-mselocpolmc+((ylocpolmc-actualp)^2)/
samplesno
mseloclin<-mseloclin+((yloclin-actualp)^2)/samplesno
mseloclinmc<-mseloclinmc+((yloclinmc-actualp)^2)/
samplesno
}
for(r in samplesno)
{
varht<-varht+((est[1,r]-mean(est[1, ]))^2)/
(samplesno-1)
varlocpol<-varlocpol+((est[2,r]-mean(est[2, ]))^2)/
(samplesno-1)
varlocpolmc<-varlocpolmc+((est[3,r]-mean(est[3, ]))^2)/
(samplesno-1)
varloclin<-varloclin+((est[4,r]-mean(est[4, ]))^2)/
(samplesno-1)
varloclinmc<-varloclinmc+((est[5,r]-mean(est[5, ]))^2)/
(samplesno-1)


}
varht
msht<-varht+((sumdifht)^2)


varlocpol
```

```
mslocpol<-varlocpol+((sumlocpol)^2)


varlocpolmc

mslocpolmc<-varlocpolmc+((sumlocpolmc)^2)


varloclin

msloclin<-varloclin+((sumloclin)^2)


varloclinmc

msloclinmc<-varloclinmc+((sumloclinmc)^2)


mht<-var(est[1,])+((sumdifht)^2)

var(est[1,])


mlpol<-var(est[2,])+((sumlocpol)^2)

var(est[2,])


mlpolmc<-var(est[3,])+((sumlocpolmc)^2)

var(est[3,])


mllin<-var(est[4,])+((sumloclin)^2)

var(est[4,])


mllinmc<-var(est[5,])+((sumloclinmc)^2)

var(est[5,])


wubmse[ ,1]<-c(sumdifht,mseht,(mseht/mselocpolmc),

msht,(msht/mslocpolmc),mht,(mht/mlpolmc))
```

```
wubmse[ ,2]<-c(sumlocpol,mselocpol,mselocpol/
mselocpolmc,
mslocpol,mslocpol/mslocpolmc,mlpol,mlpol/mlpolmc)
wubmse[ ,3]<-c(sumlocpolmc,mselocpolmc,mselocpolmc/
mselocpolmc,mslocpolmc,mslocpolmc/mslocpolmc,mlpolmc,
mlpolmc/mlpolmc)
wubmse[ ,4]<-c(sumloclin,mseloclin,mseloclin/
mselocpolmc,msloclin,msloclin/mslocpolmc,mllin,
mllin/mlpolmc)
wubmse[ ,5]<-c(sumloclinmc,mseloclinmc,mseloclinmc/
mselocpolmc,msloclinmc,msloclinmc/mslocpolmc,
mllinmc,mllinmc/mlpolmc)
print("bandwith")
print(h)
print(" yht2 ynw2   ymc2   yln2   ylnmc2")
print(wubmse)
}
```