SOFTWARE TOOLS FOR OPTIMAL TWO-STAGE SAMPLING

A USER GUIDE

Marie Reilly PhD[1]
Agus Salim[2] **BSc**
Salaheddin Mahmud[3] **MD, MSc**

[1]Department of Epidemiology and Public Health and
[2]Department of Statistics, University College Cork
and
[3]National Cancer Registry, Ireland

Funded by: The Health Research Board, Ireland

TABLE OF CONTENTS

List of Tables

1 Missing covariate data

1.1 Introduction

Missing data is one of the most common problems in data analysis. Perhaps the most common approach when confronted with missing data is excluding the incomplete cases from analysis and proceeding to analyse the complete cases using standard methods. While valid under certain assumptions regarding the missingness mechanism, this approach results in a loss of precision due to the ignored observations. In this report, we are interested in the problem of missing covariates in regression models. In the last two decades, some analysis methods that accommodate all available cases have been developed. Those methods include *meanscore* (Reilly and Pepe, 1995), *pseudo likelihood* (Breslow and Cain, 1988), *weighted likelihood* (Flanders and Greenland, 1991) and *nonparametric maximum likelihood* (Breslow and Holubkhov, 1998). The *meanscore* method is the subject of this report.

The *meanscore* method that incorporates information from all available cases into the regression model is a likelihood-based method. For completely random missingness, this results in an improvement in efficiency over the analysis of complete cases only. More importantly, the method is applicable to a wide range of patterns of missingness known as MAR (Missing at Random), where missingness may depend on the completely observed variables but not on the unobserved value of the incompletely observed variable(s).

1.2 Meanscore

The *meanscore* method is motivated by the EM algorithm (Dempster, *et.al.*, 1977). For simplicity of notation, let Y denote the response variable, Z the complete covariates (which must be categorical) and X the covariates of interest in the regression model, where some components of X are missing. The complete covariates Z may contain some auxiliary or surrogate variables that are informative about the missing components of X. Interest is focused on estimating the parameters in the regression model f_β (Y| X).

If the relationship between Z and X was fully known, we could obtain the Maximum Likelihood Estimator (MLE) for the parameters of the regression model by using the EM algorithm, which is equivalent to solving the score equation:

$$\sum_{i\in V} S_\beta(Y_i \mid X_i) + \sum_{j\in \overline{V}} E[S_\beta(Y_j \mid X, Z_j) \mid (Y_j, Z_j)] = 0$$

where S_β (Y_i| X_i) = ∂f_β (Y| X)/ $\partial\beta$, the usual score statistic (Reilly & Pepe, 1995), V denotes the set of complete (validation) cases and $\overline{V}$ denotes the set of incomplete (non-validation) cases. Throughout this chapter we will use the same notation.

Because the exact relationship between X and Z is unknown, *Mean Score* uses a non-parametric estimate for the conditional expectation above. Each incomplete case is assigned the average score of complete cases with matching Y and Z. A little algebra shows that the *Mean Score* estimator is thus the solution to the score equation

$$\sum_{i \in V} \left(\frac{n^{Z_i, Y_i}}{n^{V(Z_i, Y_i)}} \right) S_\beta(Y_i \mid X_i) = 0$$

where n^{Z_i, Y_i} denotes the total number of cases with $\mathbf{Z}=\mathbf{Z_i}$ and $\mathbf{Y}=\mathbf{Y_i}$. And $n^{V(Z_i, Y_i)}$ denotes the number of complete (validation) cases with $\mathbf{Z}=\mathbf{Z_i}$ **and** $\mathbf{Y}=\mathbf{Y_i}$.

The *meanscore* estimator is unbiased and has asymptotic variance given by (Reilly & Pepe, 1995):

$$\frac{1}{n}(I^{-1} + I^{-1} \Omega I^{-1})$$

where:

I = the observed Fisher information matrix

$$\Omega = \sum_{(Z,Y)} \frac{n^{Z,Y}}{n} \frac{n^{\overline{V}(Z,Y)}}{n^{V(Z,Y)}} \operatorname{var}[S_\beta(Y \mid X, Z) \mid Y, Z] .$$

n = total number of observations (study size)

The term $\frac{n^{V(Z,Y)}}{n^{Z,Y}}$ is referred as the "validation sampling fraction" or "second stage sampling fraction" for the (Z,Y) stratum. It can be seen from the form of the variance formula that the variance of the estimates is a function of the number of observations and the validation sampling fraction in each (Z,Y) stratum. Thus it is possible to minimise the variance using an appropriate study design. We will develop this idea further in Chapter 2.

2 Optimal Sampling Design for two-stage studies

We define a two-stage study as a study where a response variable and some predictor variables are measured at the "first stage" for all study subjects and one or more predictor variables are collected only for a subset of the study subjects at the "second stage". The second stage subjects are selected using stratified random sampling within the strata defined by the different levels of response and first stage variables.

This type of study is popular in epidemiology where researchers usually collect information on some `cheap' variables from all study subjects while expensive variables (such as laboratory tests and radiological imaging) might only be collected for some of the study subjects.

By appropriately choosing the total number of observations and the second stage sampling fractions, such design can yield more efficient and cost effective estimates than simple random sampling.

The missing covariates setting referred to in Chapter 1 can also be viewed as a two-stage design, where the response variable Y and the complete variables Z are regarded as the first stage information and the incomplete components of X are regarded as the second stage information.

2.1 Optimal Design and Meanscore

In chapter 1 we noted that the variance of *meanscore* estimates depends on the total number of observations and the validation sampling fraction in each (Z,Y) stratum. Thus it is possible to minimise the variance using an appropriate study design.

In this chapter we will outline how one can derive optimal sampling designs for two-stage studies for the following three scenarios:

1. Where we already have the first stage data and would like to sample a specified number of observations at the second stage. For example if we already have a database or registry, and we wish to gather additional information on some subjects in order to address a research question.
2. Where a fixed budget is available and we wish to design a study that will minimise the variance of an estimate subject to the budget constraint.
3. Where a coefficient of interest is to be estimated with a specified precision and we wish to design a study that will achieve this for a minimum cost.

2.2 Optimal Design Derivation

2.2.1 Fixed second stage sample size

Suppose we wish to select n_2 subjects at the second stage in such a way as to minimise the variance of the k^{th} component of the regression coefficient $\underline{\boldsymbol{\beta}}$. This is equivalent to minimising the [k,k] element in the variance-covariance matrix **V** :

$$V_{kk} = \frac{1}{n}(I^{-1}{}_{kk} + [I^{-1}\Omega I^{-1}]_{kk})$$

with the constraint that the second stage sample size n_2 is fixed.

Note that we can write the constraint as:

$$\sum_{(Z,Y)} \rho^{V(Z,Y)} \rho^{Z,Y} = \frac{n_2}{n}$$

where $\rho^{Z,Y}$ is the prevalence (probability) of the (Z,Y) stratum, $\rho^{V(Z,Y)}$ is the second stage sampling fraction for the (Z,Y) stratum, and n is the total number of observations. A Lagrange multiplier (λ) can be used to accommodate the constraint so that in this case we would minimise:

$$V_{kk} - \lambda(\sum_{(Z,Y)} \rho^{V(Z,Y)} \rho^{Z,Y} - n_2 / n)$$

Taking the first derivative of this function with respect to $\rho^{V(Z,Y)}$ and setting it to zero yields the optimal second stage sampling fractions. After some algebraic manipulation it can be shown that the optimal second stage sampling fraction in the (Z,Y) stratum $\rho^{V(Z,Y)}$ are given by:

$$\rho^{V(Z,Y)} = \frac{\frac{n_2}{n}\sqrt{[I^{-1}Var(S_\beta \mid Z,Y)I^{-1}]_{kk}}}{\sum_{(Z,Y)} \rho^{Z,Y}\sqrt{[I^{-1}Var(S_\beta \mid Z,Y)I^{-1}]_{kk}}}$$

2.2.2 Fixed budget

Assume now that we wish to minimise the variance of the k^{th} component of the regression coefficient $\underline{\boldsymbol{\beta}}$, given that we have a fixed budget **B** available and the first stage and second stage cost per observation are known to be $\mathbf{c_1}$ and $\mathbf{c_2}$ respectively.

Using the fact that $n\sum_{(Z,Y)}\rho^{V(Z,Y)}\rho^{Z,Y}=n_2$ note that we can write the constraint as: $B=n(c_1+c_2\sum_{(Z,Y)}\rho^{V(Z,Y)}\rho^{Z,Y})$, where n is the study size and $\rho^{Z,Y}$ and $\rho^{V(Z,Y)}$ are the prevalence (probability) and the second stage sampling fraction for the (Z,Y) stratum. Again using a Lagrange multiplier, our task is to minimise:

$$V_{kk}-\lambda(nc_1+nc_2\sum_{(Z,Y)}\rho^{V(Z,Y)}\rho^{Z,Y}-B)$$

where V_{kk} is the [k,k] element of the variance covariance matrix V.

The optimal study size and second stage sampling fractions can be obtained by taking the derivatives of the function above with respect to n and $\rho^{V(Z,Y)}$, setting these to zero and solving them simultaneously.

It can be shown (see Reilly and Pepe, 1995) for more details of the theoretical derivation) that the optimal study size is given by:

$$n=B\left[c_1+\frac{\sqrt{c_1c_2}\sum_{(Z,Y)}\rho^{Z,Y}\sqrt{[I^{-1}Var(S\beta\mid Z,Y)I^{-1}]_{kk}}}{\sqrt{[I^{-1}]_{kk}-\sum_{(Z,Y)}\rho^{Z,Y}[I^{-1}Var(S_\beta\mid Z,Y)I^{-1}]_{kk}}}\right]^{-1}$$

And the optimal second stage sampling fraction for (Z,Y) stratum is given by:

$$\rho^{V(Z,Y)}=\frac{B-nc_1}{nc_2}\frac{\sqrt{[I^{-1}Var(S_\beta\mid Z,Y)I^{-1}]_{kk}}}{\sum_{(Z,Y)}\rho^{Z,Y}\sqrt{[I^{-1}Var(S_\beta\mid Z,Y)I^{-1}]_{kk}}}$$

2.2.3 Fixed precision

In this case we would like to achieve a fixed variance estimate, say δ for the k^{th} component of the regression coefficient vector **β**, while minimising the study cost.

Assume again that the first stage cost is $\mathbf{c_1}$ per observation and second stage cost is $\mathbf{c_2}$ per observation. Note that as in the fixed budget case above we can write the total study cost as $n(c_1+c_2\sum_{(Z,Y)}\rho^{V(Z,Y)}\rho^{Z,Y})$. Using a Lagrange multiplier, we now wish to minimise the following function:

$$nc_1 + nc_2 \sum_{(Z,Y)} \rho^{V(Z,Y)} \rho^{Z,Y} - \lambda(V_{kk} - \delta)$$

The optimal solution can be obtained by taking the first derivative of this function with respect to n and $\rho^{V(Z,Y)}$, setting them to zero and solving the resultant equations simultaneously.

It can be shown (Reilly & Pepe, 1995) that the optimal study size is given by:

$$n = \frac{[I^{-1} - \sum_{(Z,Y)} \rho^{Z,Y} W_{ZY}]_{kk}}{\delta} + \frac{1}{\delta} \sqrt{\frac{c_2}{c_1}} \sum_{(Z,Y)} [\rho^{Z,Y} \sqrt{W_{ZY}}]_{kk} \sqrt{[I^{-1} - \sum_{(Z,Y)} \rho^{Z,Y} W_{ZY}]_{kk}}$$

and the optimal second stage sampling fraction for the (Z,Y) stratum is given by:

$$\rho^{V(Z,Y)} = \sqrt{\frac{c_1}{c_2}} \sqrt{\frac{[W_{ZY}]_{kk}}{[I^{-1} - \sum_{(Z,Y)} \rho^{Z,Y} W_{ZY}]_{kk}}}$$

Where:

$W_{ZY} = I^{-1} Var(S_\beta \mid Z,Y) I^{-1}$

2.3 Computational Issues

The derivation of the optimal designs above was carried out without constraining the second stage sampling fractions to be less than or equal to 1 ($\rho^{V(Z,Y)} \le 1$). As a result the "optimal" second stage sampling fractions computed with these formulae can be greater than 1. Reilly and Pepe (1995) proposed the following `ad-hoc' method to overcome this problem: sample 100 % from the stratum with the largest $\rho^{V(Z,Y)} > 1$, and optimally sample from the remaining strata. This step was done iteratively until all $\rho^{V(Z,Y)} \le 1$. In more recent work (Salim and Reilly, 2000) this ad-hoc method was shown to be equivalent to the *active set* method discussed by Fletcher (1987), and hence it yields the constrained optimum solution.

In the fixed budget and fixed precision scenarios the problem is more complicated since in addition to the second stage sampling fractions we have to estimate the optimal study size (total number of observations). Note that the formula for this quantity involves the square root of $[I^{-1} - \sum_{Z,Y} \rho^{Z,Y} (I^{-1} Var(S_\beta \mid Z,Y) I^{-1})]_{kk}$. In practical examples this term can be negative and thus there is no solution. If this occurs, we proposed to sample 100% from the stratum with maximum $\rho^{Z,Y} [I^{-1} Var(S_\beta \mid Z,Y) I^{-1}]_{kk}$ and optimally sample from

the remaining strata. The step was done iteratively until $[I^{-1} - \sum_{Z,Y} \rho^{Z,Y} (I^{-1} Var(S_\beta \mid Z,Y) I^{-1})]_{kk} > 0$. This ad-hoc method has also been shown to yield the constrained optimum solution (Salim and Reilly, 2000).

3 Computer Packages

We have written the *Meanscore* and optimal sampling algorithms in three programming languanges; R, S-PLUS and STATA. In this chapter we give a brief introduction to each of these packages. Chapter 4 presents detailed instructions for installing and running the *Meanscore* function in R, S-PLUS and STATA, with an introduction section explaining features that are common to all 3 systems. Chapter 5 presents a similarly structured guide to the optimal sampling software.

The R version of the *optimal* package is also available as a web-based module for users who do not have access to R, S-PLUS or STATA. The module uses the **R-web** (Banfield, 1999) environment. The last two chapters of this report, chapter 7 and chapter 8 provide, more details about this module

3.1 R language.

R is a language and environment for statistical computing and graphics. It was chosen, because it is an open, programmable, extendable package with most statistical functions already available, is freely available and has a large user base of academic statisticians committed to constant improvement and update of the package. R is similar to the S language and environment which was developed at Bell Laboratories by John Chambers and colleagues, and in fact R can be viewed as an implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form (see http://www.gnu.org). It runs on many operating systems including Windows 9x/NT, UNIX and Mac. R can be downloaded at the R project website at http://www.r-project.org or other mirror sites around the world. The latest version available at the time of writing is **R 1.2.3**. Once installed the software includes a reference manual and help files from which one can learn.

The ***r-help*** mailing list publishes announcements about the development of *R*, the availability of new code, questions and answers about problems and solutions using *R* and so on. One can subscribe to this list by sending "subscribe" in the *body* of an e-mail (not in the subject!) to r-help-request@lists.R-project.org

There are many contributed packages that can be downloaded from the R website and this kind of contribution is the strength of R because it enables the software to seamlessly include many useful statistical functions designed by a large expert user-base. Some introductory manuals for R can be found on the R website. The Venables and Ripley (1999) book on S-PLUS can also be used as an introduction to R provided it is accompanied by its `R Complements' (http://www.stats.ox.ac.uk/pub/MASS3/) which

describe how to use the book with R. The newer book by Venables and Ripley (2000) gives a deeper introduction to programming issues and also discusses some major differences between S and R.

3.2 S-PLUS

The S-PLUS software was based on the S language, originally designed by John Chambers and colleagues at Bell laboratories. S-PLUS is sold by Mathsoft,Inc. More details about S-PLUS can be found on the S-PLUS website http://www.mathsoft.com/splus.

A good introduction to the application of S-PLUS in many statistical areas can be found in Venables and Ripley (1999). There is a huge amount of user-contributed code for S, available at the http://lib.stat.cmu.edu/DOS/S/ at Carnegie Melon University.

Discussion about the main differences between S/S-PLUS and R is available at the R-FAQ section of the R website and in the recent book by Venables and Ripley (2000). The version we used to develop our software is S-PLUS version 4.0.

3.3 STATA

STATA is a statistical software package sold by STATA corporations. It has attracted a lot of interest from biostatisticians , epidemiologists and medical researchers for its ease of use and its flexibility. In a single environment the user has access to a wide range of commands from simple tables to complex models, in any sequential order. The software has a large library of contributed functions written by users. The documentation for these functions is published in the STATA Technical Bulletin (STB) and the code made available in the STATA website. For example the programs we developed have been published in STB-58, November 2000. A powerful capability in STATA is the web-compatibility; the "search" command inside STATA allows the user to quickly and easily identify contributed functions and the "net install" command allows one to directly install from the web, as part of STATA, any function they request. These functions can be installed directly from the internet if you already have STATA software installed in your computer. The books by Rabe-Hesketh and Everitt (2000) and Hamilton (1997) provide an introduction to STATA. More detail about STATA can be found at http://www.stata.com. We developed the package using STATA 6, although STATA 7 has been released since then. However all programs written in the older version of STATA can be run in the newer version by putting the information about the older version in the beginning of the program. For example our program uses the `"version 6.0"` command to inform STATA the version we used to write it.

4 Meanscore package

4.1 Package Features

There are many similarities between our packages in R, S-PLUS and STATA. In this section we highlight common features shared by the packages, and advise you to read this section regardless of which software you intend work with. Sections 4.2, 4.3 and 4.4 deal with some minor differences in the package for the different software environments.

The *Meanscore* package contains functions to implement the *Meanscore* method (Reilly and Pepe,1995) for estimating the coefficients in a logistic regression model from two-stage data. There are 3 functions in the package:

1. MEANSCORE is called with the combined first- and second-stage data (where the missing covariate values are represented by NA in R and S-PLUS and in STATA missing values are represented by a dot)
2. MS.NPREV is called with the second-stage (i.e. complete) data and the first-stage sample sizes (or prevalences). Prior to running this function, the CODING function (3) should be run to see the order in which MS.NPREV expects the first-stage sample sizes or prevalences to be provided.
3. CODING, which recodes multiple columns of first-stage covariates into a single column and displays the coding scheme.

Two illustrative data sets are also provided with the package. In the following section we give a brief description of each dataset.

simNA Simulated dataset for illustrating the meanscore function

DESCRIPTION:

A simulated data set of 1000 observations, with 500 missing values. In STATA, this dataset is called "sim_miss". There are 3 variables in the dataset. Y is the response variable. It was generated as a Bernoulli random variable with P(Y=1) = exp(x)/(1+exp(x)), where X is the true covariate, generated as a standard normal variable ~ N(0,1). Finally, Z is the surrogate for the true covariate X, and was generated using the following rule:

$$Z = \begin{cases} 0, & x \leq 0 \\ 1, & \text{otherwise} \end{cases}$$

ectopic **The ectopic pregnancy dataset**

DESCRIPTION

This dataset, which was analysed in Table 3 of Reilly and Pepe (1995) is from a case-control study of the association between ectopic pregnancy and sexually transmitted diseases (STDs). The total sample size is 979, consisting of 264 cases and 715 controls. One year after the study began, the investigators started collecting serum samples for determining chlamydia antibody status in all cases and in a 50 percent subsample of controls. As a result, only 327 out of the 979 patients have measurements for chlamydia antibody.

The dataset has 979 observations with 5 variables arranged in the following columns:

Column 1 (Pregnancy)

The ectopic pregnancy status of patients at the time of interview

(0 = No, 1 = Yes)

Column 2 (Chlamydia)

The chlamydia antibody status of patients (0 = No, 1 = Yes).

There are some observations with missing values, indicating that at the time these patients were enrolled, the investigators had not yet started to record chlamydia antibody status.

Column 3 (Gonnorhoea)

(0 = No, 1 = Yes)

Column 4 (Contracept)

The use of contraceptives

(0 = No, 1 = Yes)

Column 5 (Sexpatr)

Multiple sex partners (0 = No, 1 = Yes)

In order to run the *Meanscore* analysis the user must specify the response variable, first stage variables and the predictor variables in the model. Multiple first stage variables may be specified but they must all be categorical, although the second stage variables can be continuous. There is a facility to fit a separate coefficient for each level of a categorical predictor variable. The implementation of these features is slightly different from software to software (see sections 4.2, 4.3 and 4.4 for the specific details for each software).

4.2 Using the Meanscore package in R

4.2.1 Installation guide

You need to have R installed in your computer. Our program has been tested under R.1.2.0, so we advise you to first update your R if you still use an older version (see section 3.1 to learn more about R). You can download our program from the following sites:

- http://www.r-project.org [R website]
- http://www.ucc.ie/ucc/depts/pubh/programs/programs.html

The zip file contains README.packages file (see Appendix B) where you can find the instructions on how to install the package. Once the package has been installed you can make the *meanscore* package available by issuing the command *'library(meanscore)'* in the R session window. The command *`help(package=meanscore)'* will open a window where you can read more details about the package.

4.2.2 Syntax and features

meanscore	Mean Score Method for Missing Covariate Data in Logistic Regression Models

Usage:

```
meanscore(y=y,x=x,z=z,factor=NULL,print.all=F)
```

Arguments:

y	: response variable (binary 0-1)
x	: matrix of predictor variables, one column of which contains some missing values (NA)
z	: matrix of the surrogate or auxiliary variables which must be categorical
factor	: **optional** factor variables; if the columns of the matrix of predictor variables have names, supply these names, otherwise supply the column numbers. MEANSCORE will fit separate coefficients for each level of the factor variables.

Value:

A list called "parameters" containing the following will be returned:

est	: the vector of estimates of the regression coefficients
se	: the vector of standard errors of the estimates
z	: Wald statistic for each coefficient

pvalue : 2-sided p-value (H_0: est=0)

when print.all = T, it will also return the following lists:

Ihat : the Fisher information matrix

varsi : variance of the score for each (ylevel,zlevel) stratum

ms.nprev	Logistic regression of two-stage data using second stage sample and first stage sample sizes or proportions (prevalences) as input

Usage:

```
ms.nprev(y=y,x=x,z=z,n1="option",prev="option",factor=NULL,print.all=F)
```

Arguments:

REQUIRED ARGUMENTS

y: response variable (should be binary 0-1)

x: matrix of predictor variables for regression model

z: matrix of any surrogate or auxiliary variables,

and one of the following:

n1: vector of the first stage sample sizes for each (y,z) stratum: must be provided in the correct order (see `coding' function)

OR

prev: vector of the first-stage or population proportions (prevalences) for each (y,z) stratum: must be provided in the correct order (see `coding' function)

OPTIONAL ARGUMENTS

print.all: logical value determining all output to be printed. The default is False (F).

factor : factor variables; if the columns of the matrix of predictor variables have names, supply these names, otherwise supply the column numbers. MS.NPREV will fit separate coefficients for each level of the factor variables.

Value:

If called with `prev' will return only:

A list called "table" containing the following:

ylevel: the distinct values (or levels) of y

zlevel: the distinct values (or levels) of z

prev: the prevalences for each (y,z) stratum

n2: the sample sizes at the second stage in each stratum defined by (y,z)

and a list called "parameters" containing:

est: the Mean score estimates of the coefficients in the logistic regression model

If called with `n1' it will return:

A list called "table" containing:

ylevel: the distinct values (or levels) of y

zlevel: the distinct values (or levels) of z

n1 : the sample size at the first stage in each (y,z) stratum

n2 : the sample sizes at the second stage in each stratum defined by (y,z)

and a list called "parameters" containing:

est : the Mean score estimates of the coefficients in the logistic regression model

se : the standard errors of the Mean Score estimates

z : Wald statistic for each coefficient

pvalue: 2-sided p-value (H_0: est=0)

If print.all=T, the following lists will also be returned:

Wzy: the weight matrix used by the mean score algorithm for each (Y,Z) stratum: this will be in the same order as n1 and prev

varsi : the variance of the score in each Y,Z stratum

Ihat : the Fisher information matrix

coding	combines two or more surrogate/auxiliary variables into a vector

DESCRIPTION

recodes a matrix of categorical variables into a vector which takes a unique value for each combination

BACKGROUND

From the matrix Z of first-stage covariates, this function creates a vector which takes a unique value for each combination as follows:

z1	z2	z3	new.z
0	0	0	1
1	0	0	2
0	1	0	3

1	1	0	4
0	0	1	5
1	0	1	6
0	1	1	7
1	1	1	8

If some of the combinations do not exist, the function will adjust accordingly: for example if the combination (0,1,1) is absent above, then (1,1,1) will be coded as 7.

The values of this new.z are reported as `new.z' in the printed output (see `Value' below)

This function should be run on second stage data prior to using the ms.nprev function, as it illustrates the order in which the call to ms.nprev expects the first-stage sample sizes to be provided.

Usage:

```
coding(x=x,y=y,z=z,return=F)
```

Arguments:

REQUIRED ARGUMENTS

y: response variable (should be binary 0-1)

x: matrix of predictor variables for regression model

z: matrix of any surrogate or auxiliary variables

OPTIONAL ARGUMENTS

return: logical value; if it's TRUE(T) the original surrogate or auxiliary variables and the re-coded auxilliary variables will be returned. The default is False (F).

Value:

This function does not return any values except if `return'=T.

If used with only second stage (i.e. complete) data, it will print the following:

ylevel : the distinct values (or levels) of y

z1 ... zi : the distinct values of first stage variables z1 ... zi

new.z : recoded first stage variables. Each value represents a unique combination of first stage variable values.

n2 : second stage sample sizes in each (`ylevel',`new.z') stratum.

If used with combined first and second stage data (i.e. with NA for missing values), in addition to the above items, the function will also print the following:

n1 : first-stage sample sizes in each (`ylevel',`new.z') stratum.

4.2.3 Examples

4.2.3.1 meanscore

The simulated dataset "simNA" (see section 4.1) has 1000 observations with dichotomous response variable (Y) in the 1st column, dichotomous surrogate variable for X, called Z in the 2nd column and continuous predictor variable (X) in the 3rd column. A randomly selected 500 of the X values have been deleted (i.e. are missing). We would like to use all the data to estimate the coefficient of X in a logistic regression model:

$$\log\frac{P(Y_i=1)}{1-P(Y_i=1)}=\beta_0+\beta_1 X_i+\varepsilon$$

```
data(simNA)
meanscore(y=simNA[,1],z=simNA[,2],x=simNA[,3])
```

OUTPUT:

```
[1] "For calls to ms.nprev, input n1 or prev in the following order!!"
     ylevel z new.z  n1  n2
[1,]      0 0     0 310 150
[2,]      0 1     1 166  85
[3,]      1 0     0 177  86
[4,]      1 1     1 347 179
$parameters
                  est         se          z    pvalue
(Intercept) 0.0493998 0.07155138  0.6904103 0.4899362
x           1.0188437 0.10187094 10.0013188 0.0000000
```

We can extract the "complete cases" and do analysis based on those cases only as follows:

```
complete_simNA[!is.na(simNA[,3]),]
summary(glm(complete[,1]~complete[,3], family="binomial"))

OUTPUT:
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
     (Intercept)    0.05258    0.09879   0.532    0.595
     complete[, 3]  1.01942    0.12050   8.460   <2e-16 ***
```

Notice that the standard error produced by *Meanscore* is smaller reflecting the additional information we gained by using the available cases.

4.2.3.2 ms.nprev

The *ms.nprev* command provides a way of doing *Meanscore* analysis if we only have the complete observations but we know the first stage sample size in each stratum. The following lines will do the *Meanscore* analysis using the *ms.nprev* command for the simulated data above. Notice that we have to run the *coding* function first to see the order in which we have to enter the vector of first stage sample sizes.

```
data(simNA)
# extract the complete cases only
complete_simNA[!is.na(simNA[,3]),]

# run the coding function
coding(x=simNA[,3], y=simNA[,1], z=simNA[,2])
[1] "For calls to ms.nprev, input n1 or prev in the following order!!"
          ylevel z new.z  n1  n2
     [1,]      0 0     0 310 150
     [2,]      0 1     1 166  85
     [3,]      1 0     0 177  86
     [4,]      1 1     1 347 179
 # supply the first stage sample sizes in the correct order
 n1_c(310,166,177,347)
 ms.nprev(x=complete[,3],z=complete[,2],y=complete[,1],n1=n1)
 [1] "please run coding function to see the order in which you"
```

```
[1] "must supply the first-stage sample sizes or prevalences"
[1] " Type ?coding for details!"
[1] "For calls to ms.nprev,input n1 or prev in the following order!!"
     ylevel z new.z  n2
[1,]      0 0     0 150
[2,]      0 1     1  85
[3,]      1 0     0  86
[4,]      1 1     1 179
[1] "Check sample sizes/prevalences"
$table
     ylevel zlevel  n1  n2
[1,]      0      0 310 150
[2,]      0      1 166  85
[3,]      1      0 177  86
[4,]      1      1 347 179

$parameters
                  est         se          z    pvalue
(Intercept) 0.0493998 0.07155138  0.6904103 0.4899362
x           1.0188437 0.10187094 10.0013188 0.0000000
```

4.3 Using the Meanscore package in S-PLUS

4.3.1 Installation guide

Our program has been tested under SPLUS 4 for windows. Some modifications may be needed for other versions. If you have S-PLUS installed on your computer, you can download our program from the following sites:

- http://lib.stat.cmu.edu/DOS/S/ [STATLIB website]
- http://www.ucc.ie/ucc/depts/pubh/programs/programs.html

The zip file contains a README file (see Appendix B) where you can find the instructions on how to install the package. Once the package has been installed you can make the *meanscore* package available by issuing the command *'library(meanscore)'* in the S-PLUS session window.

4.3.2 Syntax and features

meanscore	Mean Score Method for Missing Covariate Data in Logistic Regression Models

USAGE

meanscore(x=x,y=y,z=z,fctvar=NULL,print.all=F)

REQUIRED ARGUMENTS

y: response variable (binary 0-1)

x: matrix of predictor variables, one column of which contains some missing values (NA)

z: matrix of the surrogate or auxiliary variables which must be categorical

OPTIONAL ARGUMENTS

fctvar: optional factor variables; if the columns of the matrix of predictor variables have names, supply these names, otherwise supply the column numbers. MEANSCORE will fit separate coefficients for each level of the factor variables.

SIDE EFFECTS:

A list called "parameters" containing the following will be returned:

est : the vector of estimates of the regression coefficients

se : the vector of standard errors of the estimates

z : Wald statistic for each coefficient

pvalue : 2-sided p-value (H_0: est=0)

when print.all = T, it will also return the following lists:

Ihat : the Fisher information matrix

varsi : variance of the score for each (ylevel,zlevel) stratum

ms.nprev	Logistic regression of two-stage data using second stage sample and first stage sample sizes or proportions (prevalences) as input

BACKGROUND

This algorithm will analyse the second stage data from a two-stage design, incorporating as appropriate weights the first stage sample sizes in each of the strata defined by the first-stage variables. If the first-stage

sample sizes are unknown, you can still get estimates (but not standard errors) using estimated relative frequencies (prevalences)of the strata. To ensure that the sample sizes or prevalences are provided in the correct order, it is advisable to first run the `coding' function.

USAGE

```
ms.nprev(y=y,x=x,z=z,n1="option",prev="option",fctvar=NULL,print.all=F)
```

REQUIRED ARGUMENTS

y : response variable (should be binary 0-1)

x : matrix of predictor variables for regression model

z : matrix of any surrogate or auxiliary variables,

and one of the following:

n1 : vector of the first stage sample sizes for each (y,z) stratum: must be provided in the correct order (see `coding' function)

OR

prev : vector of the first-stage or population proportions (prevalences) for each (y,z) stratum: must be provided in the correct order (see `coding' function)

OPTIONAL ARGUMENTS

print.all : logical value determining all output to be printed. The default is False (F).

fctvar : factor variables; if the columns of the matrix of predictor variables have names, supply these names, otherwise supply the column numbers. MS.NPREV will fit separate coefficients for each level of the factor variables.

SIDE EFFECTS:

If called with `prev' will return only:

A list called "table" containing the following:

ylevel : the distinct values (or levels) of y

zlevel : the distinct values (or levels) of z

prev : the prevalences for each (y,z) stratum

n2 : the sample sizes at the second stage in each stratum defined by (y,z)

and a list called "parameters" containing:

est: the Mean score estimates of the coefficients in the logistic regression model

If called with `n1' it will return:

A list called "table" containing:

ylevel : the distinct values (or levels) of y

zlevel : the distinct values (or levels) of z

n1 : the sample size at the first stage in each (y,z) stratum

n2 : the sample sizes at the second stage in each stratum defined by (y,z)

and a list called "parameters" containing:

est : the Mean score estimates of the coefficients in the logistic regression model

se : the standard errors of the Mean Score estimates

z : Wald statistic for each coefficient

pvalue : 2-sided p-value (H0: est=0)

If print.all=T, the following lists will also be returned:

wzy : the weight matrix used by the mean score algorithm, for each Y,Z stratum: this will be in the same order as n1 and prev

varsi : the variance of the score in each Y,Z stratum

Ihat : the Fisher information matrix

coding combines two or more surrogate/auxiliary variables into a vector

DESCRIPTION

recodes a matrix of categorical variables into a vector which takes a unique value for each combination

BACKGROUND

From the matrix Z of first-stage covariates, this function creates a vector which takes a unique value for each combination as follows:

z1	z2	z3	new.z
0	0	0	1
1	0	0	2
0	1	0	3
1	1	0	4
0	0	1	5
1	0	1	6
0	1	1	7

1 1 1 8

If some of the combinations do not exist, the function will adjust accordingly: for example if the combination (0,1,1) is absent above, then (1,1,1) will be coded as 7.

The values of this new.z are reported as `new.z' in the printed output (see SIDE EFFECTS below)

This function should be run on second stage data prior to using the ms.nprev function, as it illustrates the order in which the call to ms.nprev expects the first-stage sample sizes to be provided.

USAGE

coding(x=x,y=y,z=z,output=F)

REQUIRED ARGUMENTS

y: response variable (should be binary 0-1)

x: matrix of predictor variables for regression model

z: matrix of any surrogate or auxiliary variables

OPTIONAL ARGUMENTS

output: logical value; if it's TRUE(T) the original surrogate or auxiliary variables and the re-coded auxilliary variables will be returned. The default is False (F).

SIDE EFFECTS:

This function does not return any values except if output=T.

If used with only second stage (i.e. complete) data, it will print the following:

ylevel : the distinct values (or levels) of y

z1 ... zi : the distinct values of first stage variables z1 ... zi

new.z : recoded first stage variables. Each value represents a unique combination of first stage variable values.

n2 : second stage sample sizes in each (`ylevel',`new.z') stratum.

If used with combined first and second stage data (i.e. with NA for missing values), in addition to the above items, the function will also print the following:

n1: first-stage sample sizes in each (`ylevel',`new.z') stratum.

4.3.3 Examples

4.3.3.1 meanscore

Here we again demonstrate the example we illustrated in section 4.2.3. The "simNA" data set (see section 4.1) is stored in the "simNA" matrix. The matrix is automatically available when you declare *library(meanscore)*. This implementation is slightly different from R, as in R you need to load the data matrix to make it available.

```
meanscore(y=simNA[,1],z=simNA[,2],x=simNA[,3])
```

```
[1] "For calls to ms.nprev,input n1 or prev in the following
(ylevel,new.z) order!!"
  ylevel z new.z  n1  n2
1      0 0     1 310 150
2      0 1     2 166  85
3      1 0     1 177  86
4      1 1     2 347 179
$parameters:
                   est         se          z    pvalue
(Intercept) 0.04939797 0.07155154  0.6903831 0.4899533
          x 1.01885599 0.10187166 10.0013679 0.0000000
```

4.3.3.2 ms.nprev

The *ms.nprev* command provides a way of doing *Meanscore* analysis if we only have the complete observations but we know the first stage sample size in each stratum. The following lines will do the *Meanscore* analysis using the *ms.nprev* command for the simulated data above. Notice that we have to run the *coding* function first to see the order in which we have to enter the vector of first stage sample sizes.

```
# extract the complete cases only
complete_simNA[!is.na(simNA[,3]),]
```

```
# run the coding function
coding(x=simNA[,3], y=simNA[,1], z=simNA[,2])
```

```
[1] "For calls to ms.nprev,input n1 or prev in the following (ylevel,new.z)
order!!"
```

```
  ylevel z new.z  n1  n2
1      0 0     1 310 150
2      0 1     2 166  85
3      1 0     1 177  86
4      1 1     2 347 179

# supply the first stage sample sizes in the correct order
n1_c(310,166,177,347)
ms.nprev(x=complete[,3],z=complete[,2],y=complete[,1],n1=n1)

[1] "please run coding function first, it will give you idea on which order"
[1] "you have to supply the first sample sizes. Type ?coding for details!"
[1] "For calls to ms.nprev,input n1 or prev in the following (ylevel,new.z)
order!!"
  ylevel z new.z  n2
1      0 0     1 150
2      0 1     2  85
3      1 0     1  86
4      1 1     2 179
[1] "Check sample sizes/prevalences"
$table:
     ylevel zlevel  n1  n2
[1,]      0      1 310 150
[2,]      0      2 166  85
[3,]      1      1 177  86
[4,]      1      2 347 179

$parameters:
                   est         se          z    pvalue
(Intercept) 0.04939797 0.07155154  0.6903831 0.4899533
          x 1.01885599 0.10187166 10.0013679 0.0000000
```

4.4 Using the Meanscore package in STATA

4.4.1 Installation Guide

The program has been written in STATA version 6 so you need to have STATA 6 or later to be able to use the package. Because STATA Technical Bulletin require package's name to be no longer than 7 characters, the m*eanscore* package in STATA is called *meanscor.* The *meanscor* package can be installed directly from the STATA website by following these instructions:

From inside STATA, type:

```
net cd stb
net cd stb58
net describe sg156
net install sg156
net get sg156
```

After executing the last command, the *meanscor* package is installed in your computer. To test if you have installed all the components you can type `help meanscor` or `help msnprev` and try some of the examples.

IMPORTANT:

Since we submitted the program to the STB we have improved the calling syntax and the output format of the functions. The most recent version is available on our website at:
http://www.ucc.ie/ucc/depts/pubh/programs/programs.html

This program is slightly different from the one in the STB website. Therefore, the syntax and features section below are written based on the version on our website, and there may be some minor differences from the STB help files.

4.4.2 Syntax and Features

meanscor	**Meanscore method for missing covariate data in logistic regression models**

Command line:

meanscor depvar [indepvars] [if exp] [in range] [, first[varlist] second[varlist] odd(#)

Options:

first[varlist] specifies the complete covariates (i.e. measured at the first stage)

second[varlist] specifies the incomplete covariates (i.e. measured at the second stage)

odd(0) reports regression co-efficients (default=1: reports odds ratios)

msnprev	**Meanscore method for missing covariate data in logistic regression models using validation(second-stage) data and first stage sample sizes or prevalances**

Command line:

msnprev depvar [indepvars] [if exp] [in range] [, first[varlist] prev[vecname] sample[vecname] odd(0)

Options:

first[varlist] specifies the first stage covariates

odd(0) reports regression co-efficients (default: odds ratios),

and one of the following:

sample[vecname] vector of the first stage sample sizes for each stratum

OR

prev[vecname] vector of the prevalences for each stratum. If prevalences are provided, no standard errors are estimated

NOTE: you have to run the coding function (see below) prior to using this function in order to know the order in which to enter the `prev` or `sample` vector.

coding	**orders the strata formed by different levels of dependent variable and first stage covariates**

Command line:

coding depvar [first stage variables]

Description

The coding function orders the strata formed by different levels of the dependent variable and first stage covariates. This is the order in which the vector of first stage sample sizes or prevalences must be entered before calling msnprev or any of the optimal sampling functions described in section 5.4. Within the coding function a variable called grp_yz is created. It contains the distinct groups formed by different levels of the dependent variable (Y) and first stage covariates (Z). A list is printed indicating the definition of each

stratum. For calls to msnprev, optfixn, optbud and optprec, the first-stage sample sizes or prevalences must be entered following the order of grp_yz.

4.4.3 Examples

4.4.3.1 meanscor

Again, the example illustrated in section 4.2.3 is presented. The data set is called "sim_miss" (see section 4.1). The following code in STATA will give the same results as illustrated in section 4.2.3 and 4.3.3 for R and S-PLUS.

```
use sim_miss
meanscor y x,first(z) second(x) odd(0)
```

```
meanscore estimates
------------------------------------------------------------------------------
         |      Coef.   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    cons |   .0494025   .0715522      0.690   0.490      -.0908373    .1896424
       x |    1.01891   .1018749     10.002   0.000       .8192386    1.218581
------------------------------------------------------------------------------
```

4.4.3.2 msnprev

The *ms.nprev* command provides a way of doing *Meanscore* analysis if we only have the complete observations but we know the first stage sample size in each stratum. The following lines will do the *Meanscore* analysis using the *msnprev* command for the simulated data above. Notice that we have to run the *coding* function first to see the order in which we have to enter the vector of first stage sample sizes.

```
use sim_miss
coding y z
* keep the second stage sample only
keep if !missing(x)
*input the first stage sample sizes for each stratum
matrix samp=(310,166,177,347)'
msnprev y x,first(z) sample(samp) odd(0)
```

the second stage sample sizes

group(yz)	Freq.
1	150
2	85
3	86
4	179

please check the sample sizes!

grp_yz	y	z	grp_z	n1	n2
1	0	0	1	310	150
2	0	1	2	166	85
3	1	0	1	177	86
4	1	1	2	347	179

meanscore estimates

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
cons	.0494025	.0715522	0.690	0.490	-.0908373	.1896424
x	1.01891	.1018749	10.002	0.000	.8192386	1.218581

5 Optimal package

5.1 Package Features

The *optimal* package derives optimal sampling designs for 3 different scenarios as outlined in Chapter 2. To run any of the three functions you need to have a pilot sample, typically consisting of a few observations from each stratum defined by the different levels of response variable and first stage variables.

The names of the functions are slightly different for the different software, but they implement the same algorithm. We summarise the three functions below:

fixed.n	**Optimal second stage sampling fractions, subject to fixed sample sizes at the first and second stage**

Description:
This function computes the optimal second stage sampling fractions (and sample sizes) for each stratum defined by the different levels of response variable and first stage variables, using the mean score method for logistic regression models.
Users need to provide the first stage sample size for each stratum, the second stage sample size required and the name of the predictor variable to be optimised. Optimality is with respect to the standard error of a coefficient of interest.

Before running the "fixedn" function you should run the "coding" function, to see the order in which you must supply the vector of first stage sample size.

In STATA this function is called **optfixn,** while R and S-PLUS both use the name **fixed.n**

---

budget	**Optimal sampling design for 2-stage studies with fixed budget**

Description:
This function calculates the total number of study observations and the second-stage sampling fractions that will maximise precision subject to an available budget. In addition to specifying the budget the user must also supply: (i) the unit cost of observations at the first and second stage, (ii) the vector of prevalences in each of the strata defined by the different levels of response variable and first stage variables, and (iii) the name of the variable to be optimised.

Before running the "budget" function you should run the "coding" function, to see the order in which you must supply the vector of prevalences.

In STATA this function is called **optbud,** while R and S-PLUS both use the same name **budget**.

---

precision	**Optimal sampling design for 2-stage studies with fixed precision**

Description:

This function calculates the total number of study observations and the second-stage sampling fractions that will minimise the study cost subject to a fixed variance for a specified coefficient. In addition to specifying the required variance the user must also supply (i) the unit cost of observations at the first and second stage, (ii) the vector of prevalences in each of the strata defined by different levels of dependent variable and first stage covariates and (iii) the name of the variable to be optimised.

Before running this function you should run the "coding" function, to see the order in which you must supply the vector of prevalences.

In STATA this function is called **optprec,** while R and S-PLUS both use the same name **precision**.

---

coding	**combines two or more surrogate/auxiliary variables into a vector**

The **coding** function is very important because it must be run prior to running any of the optimal sampling functions in order to see the order in which you should enter the vector of prevalences or first stage sample sizes (see **syntax and features** under the different software). This function has already been described in section 4.2.2, 4.3.2 and 4.4.2.

Illustrative datasets

There are two illustrative data sets provided as examples with the package. The data sets have slightly different names from software to software (see Table 5-1).

Table 5-1. Illustrative data sets for optimal **package in R, S-PLUS and STATA**

R	S-PLUS	STATA	Features
cass1	cass1	pilotcas	Pilot observations from CASS study (Vliestra, et.al., 1980) The response variable is mortality, the first stage variable is sex and the second stage variable is age. There are 25 observations from each stratum formed by the different levels of mortality and sex.
cass2	cass2	wtpilot	Pilot observations from CASS study (Vliestra, et.al., 1980). The response variable is mortality, there are two first stage variables, **sex** and **categorical weight**. The second stage variables are weight (continuous), age, CHF, angina, LVDBP (**lve**) and urgency of surgery (**surg**). There are 10 observations from each stratum.

5.2 Using the Optimal package in R

5.2.1 Installation guide

You need to have R installed on your computer (see section 4.2.1 for details). You can download the *optimal* package from the following sites:

- http://www.r-project.org[1] [R website]
- http://www.ucc.ie/ucc/depts/pubh/programs/programs.html

The zip file contains a README.packages file (see Appendix B) where you can find the procedures on how to install the package. Once the package has been installed you can use the functions by issuing the command *'library(optimal)'* in the R session window. The command *`help(package=optimal)'* will open a window where you can read more details about the package.

5.2.2 Syntax and features

fixed.n	**Optimal second stage sampling fractions, subject to fixed sample sizes at the first and second stage**

Usage:

fixed.n (x=x,y=y,z=z,n2=n2,factor=NULL,var="var",n1="option",prev="option",frac="option")

[1] Based on Kurt Hornik's suggestion, the **optimal** package on the R website has been renamed **twostage**

`

Arguments:

REQUIRED ARGUMENTS

y : response variable (binary 0-1)

x : matrix of predictor variables

z : matrix of the surrogate or auxiliary variables (can be more than one column)

n2 : size of second stage sample

var : The name of the predictor variable whose coefficient is to be optimised. See **DETAILS** if this is a factor variable

and one of the following:

n1 : vector of the first stage sample sizes for each (y,z) stratum

OR

prev : vector of the estimated prevalences for each (y,z) stratum, AND

frac : the second stage sampling fraction i.e., the ratio of second stage sample size to first stage sample size (NOTE: if `prev' is given, `frac' will also be required)

OPTIONAL ARGUMENTS

factor: the names of any factor variables in the predictor matrix

Value:

A list called `design' consisting of the following items:

ylevel	: the different levels of response variable
zlevel	: the different levels of first stage variables z.
n1	: the first stage sample size for each (`ylevel',`zlevel') stratum
n2	: the sample size of pilot observations for each (`ylevel',`zlevel') stratum
prop	: optimal 2nd stage sampling proportion for each (`ylevel',`zlevel') stratum
samp.2nd	: optimal 2nd stage sample size for each (`ylevel',`zlevel') stratum

and a list called `se' containing:

se : the standard errors of estimates achieved by the optimal design.

budget	**Optimal sampling design for 2-stage studies with fixed budget**

Usage:

budget (x=x,y=y,z=z,prev=prev,factor=NULL,var=NULL,b=b,c1=c1,c2=c2)

Arguments:

REQUIRED ARGUMENTS

y : response variable (binary 0-1)

x : matrix of predictor variables

z : matrix of the surrogate or auxiliary variables (can be more than one column)

prev : the prevalence of each (y,z) stratum, where (y,z) is the different levels of y and z

var : The name of the predictor variable whose coefficient is to be optimised. If this is a factor variable, see **DETAILS** at the end of this section.

b : the total budget available

c1 : the cost per first stage observation

c2 : the cost per second stage observation

OPTIONAL ARGUMENTS

factor : the names of any factor variables in the predictor matrix

Value:

The following lists will be returned:

n : the optimal number of observations (first stage sample size)

design : a list consisting of the following items:

ylevel : the different levels of the response variable

zlevel : the different levels of first stage covariates z.

prev : the prevalence of each (ylevel,zlevel) stratum

n2 : the sample size of pilot observations for each (ylevel,zlevel) stratum

prop : optimal 2nd stage sampling proportion for each (ylevel,zlevel) stratum

samp.2nd : optimal 2nd stage sample size for each (ylevel,zlevel) stratum

se : the standard error of estimates achieved by the optimal design

precision	**Optimal sampling design for 2-stage studies with fixed precision**

Usage:

precision (x=x,y=y,z=z,prev=prev,factor=NULL,var=NULL,prc=prc,c1=c1,c2=c2)

Arguments:

REQUIRED ARGUMENTS

y : response variable (binary 0-1)

x : matrix of predictor variables

z : matrix of the surrogate or auxiliary variables (can be more than one column)

prev : the prevalence of each (y,z) stratum, where (y,z) is the different levels of y and z

var : The name of the predictor variable whose coefficient is to be optimised. See **DETAILS** at the end of this section if this is a factor variable

prc : the required variance of the `var' coefficient

c1 : the cost per first stage observation

c2: the cost per second stage observation

OPTIONAL ARGUMENTS

factor : the names of any factor variables in the predictor matrix

Value:

The following lists will be returned:

n: the optimal number of observations (first stage sample size)

var: the variance of estimates achieved by the optimal design

cost: the minimum study cost

and a list called `design' consisting of the following items:

ylevel : the different levels of response variable

zlevel : the different levels of first stage covariates z.

prev : the prevalence of each (y,z) stratum

n2 : the sample size of pilot observations for each (y,,z) stratum

prop : optimal 2nd stage sampling proportion for each (y,z) stratum

samp.2nd : optimal 2nd stage sample size for each (y,z) stratum

coding	**combines two or more surrogate/auxiliary variables into a vector**

Usage:

```
coding(x=x,y=y,z=z,return=F)
```

Arguments:

REQUIRED ARGUMENTS

y: response variable (should be binary 0-1)

x: matrix of predictor variables for regression model

z: matrix of any surrogate or auxiliary variables

OPTIONAL ARGUMENTS

return: logical value; if it's TRUE(T) the original surrogate or auxiliary variables and the re-coded auxilliary variables will be returned. The default is False (F).

Value:

This function does not return any values except if `return'=T.

If used with only second stage (i.e. complete) data, it will print the following:

ylevel : the distinct values (or levels) of y

z1 ... zi : the distinct values of first stage variables z1 ... zi

new.z : recoded first stage variables. Each value represents a unique combination of first stage variable values.

n2 : second stage sample sizes in each (`ylevel',`new.z') stratum.

If used with combined first and second stage data (i.e. with NA for missing values), in addition to the above items, the function will also print the following:

n1 : first-stage sample sizes in each (`ylevel',`new.z') stratum.

DETAILS:

The response, predictor and surrogate variables have to be numeric. If you have multiple columns of z, say (z1,z2,..zn), these will be recoded into a single vector "new.z". These `new.z' values are reported as `new.z' in the output (see `value'). For example:

z1	z2	z3	new.z
0	0	0	1
1	0	0	2
0	1	0	3
1	1	0	4
0	0	1	5
1	0	1	6
0	1	1	7
1	1	1	8

If some of the value combinations do not exist in your data, the function will adjust accordingly. For example if the combination (0,1,1) is absent, then (1,1,1) will be coded as 7.

If you wish to optimise the coefficient of a factor variable, you need to specify which level of the variable to optimise. For example, if "weight" is a factor variable with 3 categories 1,2 and 3 then var="weight2" will optimise the estimation of the coefficient which measures the difference between weight=2 and the baseline (weight=1). By default the baseline is always the category with the smallest value.

5.2.3 Examples

We give an example using the pilot subsample from the CASS data discussed in Reilly (1996) and described briefly in Table 5-1. The data are in the cass2 matrix, which can be loaded using data (cass2) and a description of the data set can be seen using help(cass2). Our model is:

$$\log\frac{P(Y_i=1)}{1-P(Y_i=1)} = \beta_0 + \beta_1 SEX_i + \beta_2 weight_i + \beta_3 age_i + \beta_4 CHF_i + \beta_5 ANGINA_i + \beta_6 LVE_i + \beta_7 Surgery_i$$

where the response variable is operative mortality.

In our examples below, we use sex and categorical weight as auxiliary variables. Given an available budget of £10,000, a first-stage cost of £ 1/unit and second-stage cost of £ 0.5/unit, the code below will calculate the sampling strategy that optimises the precision of the coefficient for Surgery **(surg)** : see output below.

```
data(cass2)
y_cass2[,1]            #response variable
z_cass2[,c(2,3)]       #auxiliary variable
x_cass2[,c(2,4:9)]     #predictor variables

# run CODING function to see in which order we should enter prevalences
coding(x=x,y=y,z=z)

[1] "For calls requiring n1 or prev as input, use the following order"
      ylevel sex wtcat new.z n2
 [1,]      0   0     1     1 10
 [2,]      0   1     1     2 10
 [3,]      0   0     2     3 10
 [4,]      0   1     2     4 10
 [5,]      0   0     3     5 10
 [6,]      0   1     3     6 10
 [7,]      1   0     1     1  8
 [8,]      1   1     1     2 10
 [9,]      1   0     2     3 10
[10,]      1   1     2     4 10
[11,]      1   0     3     5 10
[12,]      1   1     3     6 10

# supplying the prevalence (from Table 5, Reilly 1996)
prev_c(.0197823937,0.0544015826,0.1339020772,0.0503214639,0.6698813056,
0.0467359050,0.0009891197,0.0022255193,0.0040801187,0.0032146390,
0.0127349159,0.0017309594)

# optimise surg coefficient
budget(x=x,y=y,z=z,var="surg",prev=prev,b=10000,c1=1,c2=0.5)


[1] "please run coding function to see the order in which you"
[1] "must supply the first-stage sample sizes or prevalences"
[1] " Type ?coding for details!"
[1] "For calls requiring n1 or prev as input, use the following order"
      ylevel sex wtcat new.z n2
```

```
 [1,]       0   0     1     1 10
 [2,]       0   1     1     2 10
 [3,]       0   0     2     3 10
 [4,]       0   1     2     4 10
 [5,]       0   0     3     5 10
 [6,]       0   1     3     6 10
 [7,]       1   0     1     1  8
 [8,]       1   1     1     2 10
 [9,]       1   0     2     3 10
[10,]       1   1     2     4 10
[11,]       1   0     3     5 10
[12,]       1   1     3     6 10
[1] "Check sample sizes/prevalences"

$n
[1] 8752

$design
      ylevel zlevel         prev n2   prop samp.2nd
 [1,]      0      1 0.0197823937 10 0.6181      107
 [2,]      0      2 0.0544015826 10 1.0000      476
 [3,]      0      3 0.1339020772 10 0.5107      598
 [4,]      0      4 0.0503214639 10 0.1465       65
 [5,]      0      5 0.6698813056 10 0.1061      622
 [6,]      0      6 0.0467359050 10 1.0000      409
 [7,]      1      1 0.0009891197  8 1.0000        9
 [8,]      1      2 0.0022255193 10 1.0000       19
 [9,]      1      3 0.0040801187 10 1.0000       36
[10,]      1      4 0.0032146390 10 1.0000       28
[11,]      1      5 0.0127349159 10 1.0000      111
[12,]      1      6 0.0017309594 10 1.0000       15

$se
                   [,1]
(Intercept) 1.181028836
sex         0.219738725
weight      0.006671856
age         0.014483114
```

```
angina      0.241016357
chf         0.074636552
lve         0.009852953
surg        0.175539553
```

5.3 Using the Optimal package in S-PLUS

5.3.1 Installation guide

Our program has been tested under SPLUS 4 for windows. Some modifications may be needed for other versions. If you have S-PLUS installed on your computer, you can download our program from the following sites:

- http://lib.stat.cmu.edu/DOS/S/ [STATLIB website]
- http://www.ucc.ie/ucc/depts/pubh/programs/programs.html

The zip file contains a README file where you can find the instructions on how to install the package. Once the package has been installed you can use the functions by issuing the command *'library(optimal)'* in the S-PLUS session window.

5.3.2 Syntax and features

fixed.n	**Optimal second stage sampling fractions, subject to fixed sample sizes at the first and second stage**

Usage:

fixed.n (x=x,y=y,z=z,n2=n2,fctvar=NULL,var="var",n1="option",prev="option",frac="option")

Arguments:

REQUIRED ARGUMENTS

y : response variable (binary 0-1)

x : matrix of predictor variables

z : matrix of the surrogate or auxiliary variables (can be more than one column)

n2 : size of second stage sample

var : The name of the predictor variable whose coefficient is to be optimised. If this is a factor variable please see DETAILS at the end of this section**.**

and one of the following:

n1 : vector of the first stage sample sizes for each (y,z) stratum OR

prev : vector of the estimated prevalences for each (y,z) stratum, AND
frac : the second stage sampling fraction i.e., the ratio of second stage sample size to first stage sample size (NOTE: if `prev' is given, `frac' will also be required)

OPTIONAL ARGUMENTS
fctvar : the names of any factor variables in the predictor matrix

SIDE EFFECTS:

A list called `design' consisting of the following items:

ylevel	: the different levels of response variable
zlevel	: the different levels of first stage variables z.
n1	: the first stage sample size for each (`ylevel',`zlevel') stratum
n2	: the sample size of pilot observations for each (`ylevel',`zlevel') stratum
prop	: optimal 2nd stage sampling proportion for each (`ylevel',`zlevel') stratum
samp.2nd	: optimal 2nd stage sample size for each (`ylevel',`zlevel') stratum

and a list called `se' containing:
se : the standard errors of estimates achieved by the optimal design.

budget **Optimal sampling design for 2-stage studies with fixed budget**

Usage:

budget (x=x,y=y,z=z,prev=prev,fctvar=NULL,var=NULL,b=b,c1=c1,c2=c2)

Arguments:

REQUIRED ARGUMENTS
y : response variable (binary 0-1)
x : matrix of predictor variables
z : matrix of the surrogate or auxiliary variables (can be more than one column)
prev : the prevalence of each (y,z) stratum, where (y,z) is the different levels of y and z
var : The name of the predictor variable whose coefficient is to be optimised. If this is a factor variable please see DETAILS at the end of this section**.**
b : the total budget available

c1 : the cost per first stage observation

c2 : the cost per second stage observation

OPTIONAL ARGUMENTS

fctvar : the names of any factor variables in the predictor matrix

SIDE EFFECTS:

The following lists will be returned:

n : the optimal number of observations (first stage sample size)

design : a list consisting of the following items:

ylevel : the different levels of the response variable

zlevel : the different levels of first stage covariates z.

prev : the prevalence of each (ylevel,zlevel) stratum

n2 : the sample size of pilot observations for each (ylevel,zlevel) stratum

prop : optimal 2nd stage sampling proportion for each (ylevel,zlevel) stratum

samp.2nd: optimal 2nd stage sample size for each (ylevel,zlevel) stratum

se : the standard error of estimates achieved by the optimal design

precision	**Optimal sampling design for 2-stage studies with fixed precision**

Usage:

```
precision (x=x,y=y,z=z,prev=prev,fctvar=NULL,var=NULL,prc=prc,c1=c1,c2=c2)
```

Arguments:

REQUIRED ARGUMENTS

y : response variable (binary 0-1)

x : matrix of predictor variables

z : matrix of the surrogate or auxiliary variables (can be more than one column)

prev : the prevalence of each (y,z) stratum, where (y,z) is the different levels of y and z

var : The name of the predictor variable whose coefficient is to be optimised. If this is a factor variable please see DETAILS at the end of this section**.**

prc : the fixed variance of `var' coefficient

c1 : the cost per first stage observation

c2 : the cost per second stage observation

OPTIONAL ARGUMENTS

fctvar: the names of any factor variables in the predictor matrix

SIDE EFFECTS:

The following lists will be returned:

n : the optimal number of observations (first stage sample size)

design : a list consisting of the following items:

ylevel : the different levels of response variable

zlevel : the different levels of first stage covariates z.

prev : the prevalence of each (ylevel,zlevel) stratum

n2 : the sample size of pilot observations for each (ylevel,zlevel) stratum

prop : optimal 2nd stage sampling proportion for each (ylevel,zlevel) stratum

samp.2nd : optimal 2nd stage sample size for each (ylevel,zlevel) stratum

var : the variance of estimates achieved by the optimal design

cost : the minimum study cost

coding	**combines two or more surrogate/auxiliary variables into a vector**

Usage:

coding(x=x,y=y,z=z,output=F)

REQUIRED ARGUMENTS

y : response variable (should be binary 0-1)

x : matrix of predictor variables for regression model

z : matrix of any surrogate or auxiliary variables

OPTIONAL ARGUMENTS

output : logical value; if it's TRUE(T) the original surrogate or auxiliary variables and the re-coded auxiliary variables will be returned. The default is False (F).

SIDE EFFECTS:

This function does not return any values except if output=T.

If used with only second stage (i.e. complete) data, it will print the following:

ylevel : the distinct values (or levels) of y

z1 ... zi : the distinct values of first stage variables z1 ... zi

new.z : recoded first stage variables. Each value represents a unique combination of first stage variable values.

n2 : second stage sample sizes in each (`ylevel',`new.z') stratum.

If used with combined first and second stage data (i.e. with NA for missing values), in addition to the above items, the function will also print the following:

n1 : first-stage sample sizes in each (`ylevel',`new.z') stratum.

DETAILS:

The response, predictor and surrogate variables have to be numeric. If you have multiple columns of z, say (z1,z2,..zn), these will be recoded into a single vector "new.z". These `new.z' values are reported as `new.z' in the output (see `value').

z1	z2	z3	new.z
0	0	0	1
1	0	0	2
0	1	0	3
1	1	0	4
0	0	1	5
1	0	1	6
0	1	1	7
1	1	1	8

If some of the value combinations do not exist in your data, the function will adjust accordingly. For example if the combination (0,1,1) is absent, then (1,1,1) will be coded as 7.

If you wish to optimise the coefficient of a factor variable, you need to specify which level of the variable to optimise. For example, if "weight" is a factor variable with 3 categories 1,2 and 3 then var="weight2" will optimise the estimation of the coefficient which measures the difference between weight=2 and the baseline (weight=1). By default the baseline is always the category with the smallest value.

5.3.3 Examples

Here, we show the same example as in section 5.2.3 above. We assume, as before, that we have a £10,000 budget and the first stage and second stage cost per observations are £ 1 and £ 0.5 respectively. Suppose we would like to optimise the precision of the urgency of surgery (**surg**) coefficient.

```
y_cass2[,1]            #response variable
z_cass2[,c(2,3)]       #auxiliary variables
x_cass2[,c(2,4:9)]     #predictor variables

# run CODING function to see in which order we should enter prevalences
coding(x=x,y=y,z=z)

[1] "For calls to ms.nprev,input n1 or prev in the following
(ylevel,new.z) order!!"
   ylevel z1 z2 new.z n2
 1      0  0  1     1 10
 2      0  1  1     2 10
 3      0  0  2     3 10
 4      0  1  2     4 10
 5      0  0  3     5 10
 6      0  1  3     6 10
 7      1  0  1     1  8
 8      1  1  1     2 10
 9      1  0  2     3 10
10      1  1  2     4 10

11      1  0  3     5 10
12      1  1  3     6 10

# enter the prevalences (from Table 5, Reilly (1996))
prev_c(.0197823937,0.0544015826,0.1339020772,0.0503214639,0.6698813056,
0.0467359050,0.0009891197,0.0022255193,0.0040801187,0.0032146390,
0.0127349159,0.0017309594)

# optimise the surg coefficient
budget(x=x,y=y,z=z,var="surg",prev=prev,b=10000,c1=1,c2=0.5)
```

```
[1] "please run coding function first, it will give you idea on which order"
[1] "you have to supply the first sample sizes. Type ?coding for details!"
[1] "For calls to ms.nprev,input n1 or prev in the following (ylevel,new.z)
order!!"
   ylevel z1 z2 new.z n2
 1      0  0  1     1 10
 2      0  1  1     2 10
 3      0  0  2     3 10
 4      0  1  2     4 10
 5      0  0  3     5 10
 6      0  1  3     6 10
 7      1  0  1     1  8
 8      1  1  1     2 10
 9      1  0  2     3 10
10      1  1  2     4 10
11      1  0  3     5 10
12      1  1  3     6 10
[1] "Check sample sizes/prevalences"
$n:
[1] 8752

$design:
   ylevel zlevel         prev n2   prop samp.2nd
 1      0      1 0.0197823937 10 0.6181      107
 2      0      2 0.0544015826 10 1.0000      476
 3      0      3 0.1339020772 10 0.5107      598
 4      0      4 0.0503214639 10 0.1465       65
 5      0      5 0.6698813056 10 0.1061      622
 6      0      6 0.0467359050 10 1.0000      409
 7      1      1 0.0009891197  8 1.0000        9
 8      1      2 0.0022255193 10 1.0000       19
 9      1      3 0.0040801187 10 1.0000       36
10      1      4 0.0032146390 10 1.0000       28
11      1      5 0.0127349159 10 1.0000      111
12      1      6 0.0017309594 10 1.0000       15
```

```
$se:
                     SE
(Intercept) 1.181027372
        sex 0.219738497
         wt 0.006671847
        age 0.014483091
     angina 0.241016110
        chf 0.074636521
        lve 0.009852944
       surg 0.175539344
```

5.4 Using the Optimal package in STATA

5.4.1 Installation guide

We have written the package in STATA version 6, so you need to have STATA version 6 or later. The *optimal* package can be installed directly from the STATA website by following these instructions:
From inside STATA, type:

```
net cd stb
net cd stb58
net describe sxd2
net install sxd2
net get sxd2
```

After executing the last command, the *optimal* package is installed in your computer. To test if you have installed all the components you can type `help optfixn` or `help optbud` or `help optprec` to try some of the examples.

IMPORTANT:
Since we submitted the program to the STB we have improved the calling syntax and the output format. The most recent version is available at http://www.ucc.ie/depts/ucc/pubh/programs/programs.html

Note that this program is slightly different from the STB version. The most noticeable change is that the functions now call the name of the variable to be optimised in the *optvar* option instead of the position of the variable in the predictor matrix.

Since the syntax and features section below are written based on the version in our website, there may be some minor differences from the STB help files.

5.4.2 Syntax and features

optfixn	**optimal sampling design for 2-stage study with fixed second stage sample size**

Command line:

optfixn depvar [indepvars] [if exp] [in range] [, first[varlist] n1[vecname] n2(#) optvar(varname) coding(#)

Options

first[varlist]	specifies the first stage variables
n1[vecname]	vector of first stage sample sizes for each stratum formed by different levels of dependent variable and first stage covariates.
n2(#)	second stage sample sizes
optvar	the covariate whose variance estimate is to be minimised (i.e. optimised). If the covariate is a factor (categorical) variable, you need to specify the level whose coefficient is to be optimised (see ANALYSIS WITH CATEGORICAL VARIABLES)
coding(#)	a logical flag: default of 0 (FALSE) means that prior to calling the optfixn function you have run the "coding" function (help coding for details) to create the vector grp_yz, containing the distinct groups (strata) formed by the different levels of response (Y) and first stage covariates (Z). If you have not run "coding" and you call the "optfixn" function with coding=1, the grp_yz vector will be created within the optfixn function, but it is imperative that the vector [vecname] is provided to optfixn in the correct order! For this reason, we strongly suggest that any call to optfixn is preceded by a call to coding. For more details about the **coding** function see section **4.4 Using the Meanscore package in STATA.**

optbud	**Optimal sampling design for 2-stage studies with fixed budget**

Command line:

optbud depvar [indepvars] [if exp] [in range] [, first[varlist] prev[vecname] b(#) c1(#) c2(#) optvar(varname) coding(#)

Options:

first[varlist]	specifies the first stage variables
prev[vecname]	vector of prevalences for each stratum formed by different levels of dependent variable and first stage covariates.
b(#)	available budget
c1(#)	cost per study subject at the first stage
c2(#)	cost per study subject at the second stage
optvar(varname)	the covariate whose variance estimate is to be minimised (i.e. optimised). If the covariate is a factor variable you need to specify the level whose coefficient is to be optimised (see ANALYSIS WITH CATEGORICAL VARIABLES).
coding(#)	a logical flag: default of 0 (FALSE) means that prior to calling the optbud function you have run the "coding" function (help coding for details) to create the vector grp_yz, containing the distinct groups (strata) formed by the different levels of response (Y) and first stage covariates (Z). If you have not run "coding" and you call the "optbud" function with coding=1, the grp_yz vector will be created within the optbud function, but it is imperative that the vector [vecname] is provided to optbud in the correct order! For this reason, we strongly suggest that any call to optbud is preceded by a call to coding. For more details about the **coding** function see at section **4.4 Using the Meanscore package in STATA.**

optprec	**Optimal sampling design for 2-stage studies with fixed precision**

Command line:

optprec depvar [indepvars] [if exp] [in range] [, first[varlist] prev[vecname] prec(#) c1(#) c2(#) optvar(varname) coding(#)

Options:

first[varlist]	specifies the first stage variables
prev[vecname]	vector of prevalences for each stratum formed by different levels of dependent variable and first stage covariates

prec(#)	the variance we want to achieve, while minimising cost
c1(#)	cost per study subject at the first stage
c2(#)	cost per study subject at the second stage
optvar(varname)	the covariate for which we want to achieve a variance=prec. If the covariate is a factor variable you need to specify the level whose coefficient is to be optimised (see ANALYSIS WITH CATEGORICAL VARIABLES).
coding(#)	a logical flag: default of 0 (FALSE) means that prior to calling the optprec function you have run the "coding" function (help coding for details) to create the vector grp_yz, containing the distinct groups (strata) formed by the different levels of response (Y) and first stage covariates (Z). If you have not run "coding" and you call the "optprec" function with coding=1, the grp_yz vector will be created within the optprec function, but it is imperative that the vector [vecname] is provided to optprec in the correct order! For this reason, we strongly suggest that any call to optprec is preceded by a call to coding. For more details about the **coding** function see section **4.4 Using the Meanscore package in STATA.**

ANALYSIS WITH CATEGORICAL VARIABLES

When we have categorical predictor variables we usually fit separate coefficients for each category. The **xi** command prefix is a standard STATA command which can be used with **optfixn, optbud** and **optprec** to accommodate this need. STATA will create some new variables with names I`varname'_`level'. For example variable Isex_1 is a categorical variable for variable SEX with level = 1. If you want to optimise the variance estimates for this variable you should set optvar(Isex_1) in the command syntax.

xi: optfixn mort i.sex age,first(sex) n1(fstsamp) n2(1000) optvar(Isex_1)

5.4.3 Examples

We illustrate the same example as discussed in sections **5.2.3** and **5.3.3**. We assume, as before, that we have a £10,000 budget and the first stage and second stage cost per observation are £ 1 and £ 0.5 respectively. The first stage variables are sex and categorical weight. Suppose we would like to optimise the precision of the urgency of the surgery (**surg**) coefficient. The following STATA commands will run the analysis:

use wtpilot
coding mort sex wtcat

** enter the prevalences in the order suggested by coding function

** NOTE: the transpose operator **'** is essential

matrix prev=(0.02,.134,.670,.054,.05,.047,.001,.004,.013,.002,.003,.002)'

** optimise the **surg** coefficient

optbud mort sex-surg,first(sex wtcat) prev(prev) optvar(surg) b(10000) c1(1) c2(0.5)

OUTPUT:

```
the second stage sample sizes
----------+-----------
group(mor |
t sex     |
wtcat)    |      Freq.
----------+-----------
        1 |         10
        2 |         10
        3 |         10
        4 |         10
        5 |         10
        6 |         10
        7 |          8
        8 |         10
        9 |         10
       10 |         10
       11 |         10
       12 |         10
----------+-----------
please check the sample sizes!

grp_yz   mort    sex     wtcat  grp_z    prev  n2_pilot
     1      0      0         1      1     .02      10
     2      0      0         2      2    .134      10
     3      0      0         3      3     .67      10
     4      0      1         1      4    .054      10
     5      0      1         2      5     .05      10
     6      0      1         3      6    .047      10
     7      1      0         1      1    .001       8
     8      1      0         2      2    .004      10
     9      1      0         3      3    .013      10
```

```
    10      1      1         1      4    .002      10
    11      1      1         2      5    .003      10
    12      1      1         3      6    .002      10
the optimal sampling fraction (sample size) for grp_yz 1 = .604 (106)
the optimal sampling fraction (sample size) for grp_yz 2 = .504 (591)
the optimal sampling fraction (sample size) for grp_yz 3 = .106 (624)
the optimal sampling fraction (sample size) for grp_yz 4 = 1 (473)
the optimal sampling fraction (sample size) for grp_yz 5 = .146 (64)
the optimal sampling fraction (sample size) for grp_yz 6 = 1 (412)
the optimal sampling fraction (sample size) for grp_yz 7 = 1 (9)
the optimal sampling fraction (sample size) for grp_yz 8 = 1 (35)
the optimal sampling fraction (sample size) for grp_yz 9 = 1 (114)
the optimal sampling fraction (sample size) for grp_yz 10 = 1 (18)
the optimal sampling fraction (sample size) for grp_yz 11 = 1 (26)
the optimal sampling fraction (sample size) for grp_yz 12 = 1 (18)
the optimal number of obs = 8756
the minimum variance for surg : .0311946
total budget spent:  10001
```

Note:

The optimal study size and second stage sampling fractions are slightly different from those obtained using R and S-PLUS, but this is simply because we rounded the prevalence vector.

6 Comparative studies

6.1 Motivation

The purpose of this chapter is to compare our methods with other approaches to the analysis of incomplete data and optimal two-stage sampling.

Meanscore is only one of many approaches to the analysis of incomplete data. There are many other methods such as *multiple imputation* and likelihood-based methods such as *pseudo-likelihood* (Cain and Breslow, 1988) and *maximum likelihood* (Breslow and Holubkhov, 1997). In section 6.2, we compare *meanscore* with *hotdeck multiple imputation,* while in section 6.3, we compare *meanscore* to other likelihood-based methods.

Our optimal sampling methodology relies on the *Meanscore* method (in particular the variance formula). We looked for published work (and associated software) which adopted an alternative approach so that we could compare the performance of our method. This is described in section 6.4

6.2 Meanscore and Hotdeck multiple imputation

Hotdeck is a non-parametric version of multiple imputation. It refers to a procedure in which each incomplete case is "filled-in" (i.e. imputed) using several cases that are sampled randomly (with replacement) from the list of similar complete cases to form several imputed data sets. The hotdeck estimate is the average of the standard estimates from all imputed data sets. However the usual multiple imputation variance formula when applied to hotdeck multiple imputation underestimates the between-imputation variance. This is because simple hotdeck multiple imputation acts as if the distribution of non-missing sample values was exactly the same as the population distribution of the values (Little and Rubin, 1987). The *Approximate Bayesian bootstrap* (ABB) method can be used to increase the between-imputation variability so that *hotdeck* can be implemented using a two-step procedure (Mander and Clayton, 2000) to yield unbiased variance estimate.

To investigate the behaviour of the various estimators and more importantly their variance estimates, we generated **200** observations of a standard normal predictor variable **X**~ N(0,1). The response variable (Y) was then generated as a Bernoulli random variable with p= exp(x)/(1+exp(x)). A dichotomous surrogate variable for X, called Z, was generated as follows:

$$Z = \begin{cases} 1, & X > 0 \\ 0, & \text{otherwise} \end{cases}$$

We deleted $100(1-\rho^V)\%$ observations of the predictor variable X, where ρ^V is the validation sampling fraction, using both a random and a balanced sampling scheme. We used two different sampling fraction, ρ^V=0.25, and ρ^V=0.5. Under each setting, 3 and 10 imputations were used to compare the effect on the estimates of the increasing number of imputations. We performed 500 simulations to study the variability of the estimates. In every simulation the data are analysed using 3 different methods; **simple hotdeck, hotdeck** using the ABB method and the *meanscore* method. We developed R functions for **simple hotdeck** and **hotdeck** using the ABB method. The function for **hotdeck** using the ABB method was built using the same procedure suggested by Mander and Clayton (2000). Several statistics were calculated and compared:

Mean(β) — the average of the estimate from all 500 simulations.

Monte Carlo SE — the 'true' standard error of the estimate computed from 500 estimates as square root of $\sum_{i=1}^{500}\frac{(\beta_i-\bar{\beta}_{500})^2}{499}$, where β_i is the estimate from the i^{th} simulation and $\bar{\beta}_{500}$ is the average of the estimate as listed in Mean(β) column.

Estimated SE — represents the average over 500 simulations of the estimated standard error using the appropriate variance formula for the methods. This was computed as the square root of $\frac{1}{500}\sum_{i=1}^{500}\text{var}(\beta_i)$, where $\text{var}(\beta_i)$ is the variance estimate from the i^{th} simulation.

Bias — measures the bias of the standard error estimates of each method: it was computed as the difference between the Monte Carlo SE and estimated standard error, relative to the Monte Carlo standard error.

95% coverage — the proportion of simulations where the nominal 95 % confidence interval for β covers the true β=1.

The results are presented Table A-1.

As expected, all methods give unbiased estimates for β=1, with the *Meanscore* giving the most stable estimate (least variability). Not surprisingly, the largest departure from β=1 occurs when the validation sampling fraction is small (ρ^v=0.25) and there are small number of imputations (3 imputations).

By comparing the '*estimated SE'* and *'Monte Carlo SE*' columns under each sampling scheme we see that simple hotdeck consistently gives standard error estimates which are biased downward for all sampling fractions and number of imputations.

Examining the interval estimates, nominal 95% confidence intervals for the simple hotdeck gives a low coverage, which is not suprising in view of the bias discussed above. This effect is more obvious for smaller validation sampling fraction, where the coverage of the nominal 95% confidence interval never exceeded 87 %. The Mean Score and ABB hotdeck meanwhile give good coverage for all settings.

We conclude that the *Meanscore* method works as well as the hotdeck multiple imputation using *Approximate Bayesian Boostrap*. This conclusion is supported by Reilly and Pepe (1997) who proved that the *meanscore* estimate has the same asympotic distribution as the *hotdeck* estimate with infinite number of imputations. But *meanscore* has an advantage since it can produce the estimates in only one pass through the data whereas *hotdeck* is more demanding on computer time, especially if the data set is large.

6.3 Meanscore and other likelihood-based method

We compare *Meanscore* with three other methods of estimation for missing covariate data. Those methods are *pseudo-likelihood* (Breslow and Cain, 1988), *weighted-likelihood* (Flanders and Greenland, 1991) and *maximum likelihood* (Breslow and Holubkhov, 1997). The S-PLUS functions for implementing the last three methods were developed by Breslow and Chatterjee. The programs can be downloaded from http://www.biostat.washington.edu/~norm/software.html. Readers interested in the theoretical details of those methods should consult the original articles.

We used the following 5 data sets to compare the performance of the methods:

1. **Simulated dataset of 1000 2nd stage observations from the CASS dataset**.
   The "cass1" pilot dataset (see Table 5-1 p. 34, for more details about this dataset) was used to obtain this 2nd stage sample, optimal with respect to **age**. We used the first stage sample size from Table 3 of Reilly (1996).

2. **Simulated dataset of 1000 2nd stage observations from the CASS dataset**.
   The "cass2" pilot dataset (see Table 5-1 p. 34 for more details about this dataset) was used to obtain this 2nd stage sample, optimal with respect to **left ventricular diastolic blood pressure (LVDBP)**. We used the first stage sample size from Table 5 of Reilly (1996).

3. **Ectopic pregnancy data set (Sherman, et.al, 1990)**.

   This dataset, which was analysed in Table 3 of Reilly and Pepe (1995) is from a case-control study of the association between ectopic pregnancy and sexually transmitted diseases(STDs). The total sample size is 979 consisting of 264 cases and 715 controls. The variables collected from the beginning of the study included gonnorhoea, contraceptive use and sexual partners. One year after the study began, the investigators started collecting serum samples for determining chlamydia antibody status in all cases and in a 50 percent subsample of controls. As a result, only 327 out of the 979 patients have measurements for **chlamydia antibody**. This dataset is described briefly in **section 4.1**.

4. **National Wilms Tumor dataset with Institutional Histology as the first stage variable (Breslow and Chatterjee, 1999)**

   This dataset comes from the National Wilms Tumor Study Group (NWTSG). There are 3 variables in the dataset; the treatment outcome (relapse or not), the type of tumor ('favourable histology' (FH) or 'unfavourable histology' (UH)) measured at the NWTSG pathology centre (Central Histology) and the type of tumor predicted by pathologists at the participating institutions (Institutional Histology). There are 4088 patients in the original dataset. In this dataset the treatment outcome is the response variable, the Institutional Histology is the first stage variable and the Central Histology is the second stage variable. A second stage sample of 1142 observations were drawn using a **"balanced"** sampling scheme. The term "balanced" here is used as the investigators attempted to *sample all relapsed cases and those with UH tumors predicted by institutional pathologists*. The sampling fraction for controls with FH tumor was chosen so that the number of controls and cases in the dataset are the same.

5. **National Wilms Tumor data with Institutional Histology and Stage of tumor as the first stage variable (Breslow and Chatterjee, 1999)**

   This dataset contain the same observations as dataset 4 above. In addition to Institutional Histology, we also have Stage of Tumor as a first stage variable.

Each dataset was analysed using 4 different methods (*meanscore*, *weighted likelihood, pseudo likelihood* and *maximum likelihood*). The results are presented in Table A-2 - Table A-6.

From those tables we note the similar performance of all methods. In particular it is encouraging to see that the performance of *Meanscore* is comparable to the full maximum likelihood method even when the 2^{nd} stage sampling fraction is small. As expected, maximum likelihood is slightly more efficient, yielding smaller standard errors. However, this small gain is achieved at the expense of a more complex algorithm and lack of robustness when the model being fit is wrong (Breslow and Chatterjee, 1999).

6.4 Optimal Two-stage (Validation) Studies

The paper by Holcroft and Spiegelman (1999) derives a method for two-stage (validation) sampling for estimation of the odd-ratio (OR) in a logistic regression analysis, essentially the same problem we addressed. These authors offer a FORTRAN programme for calculating the optimal design. We obtained the program from Dr. Spiegelman (stdls@channing.harvard.edu). The program uses the FFSQP routine by Dr. Andre Tits (andre@isr.umd.edu), which is free for non-commercial purposes. After some email correspondence we managed to get a copy of the routine, by linking it with the main program, we could run the optimal design calculations.

This program only accommodates a single predictor and one surrogate variable in the analysis. In addition the user must supply the prevalence of the outcome, P(Y=1), the sensitivity and specificity of the surrogate variable, the prevalence of the surrogate variable, P(V=1), the odd-ratio estimates and the 2^{nd} stage sampling fraction. Unfortunately the program failed to run for many settings, even using the same parameters the authors considered in the paper. For one setting for which the program successfully ran, it returned the optimal sampling design for different sampling schemes: OPTIMAL, BALANCED, RANDOM, CASE-CONTROL and PROSPECTIVE. The OPTIMAL sampling scheme here is a hybrid design which is the same as what we considered in our algorithm. Hence we compared the OPTIMAL sampling scheme with our results using the same set of parameters. However, it appears that the results given by Holcroft and Spiegelman's algorithm is counter-intuitive in the sense that their OPTIMAL design does not "like" to sample from rare cells. Our algorithm, conversely, samples more from rare cells, indicating that observations in rare cells tend to be more informative. We have communicated our findings to the authors, but at the time of this writing we have not had any response. Our assessment to their program is that it is not user-friendly, not general enough (it can only take single predictor) and in its current form, we could not recommend it.

7 Rweb modules for optimal sampling: development and configuration

Rweb is a Web based interface to the R statistical package. R is a freely distributed open source system for statistical computation and graphics (Hornik, 2000). R was initially written by Ross Ihaka and Robert Gentleman and is available for download from any CRAN (Comprehensive R Archive Network) mirror site. See R home page at <http://www.r-project.org/> for more information about downloading and installing R.

Rweb was developed by Banfield (1999) to provide an easy to use interface to all of R statistical and data management functions. Rweb is available at http://www.math.montana.edu/Rweb/index.html It comes in three versions:

- The basic Rweb code window will run on most browsers but requires knowledge of R programming,
- The JavaScript version provides a more sophisticated interface but requires a JavaScript-capable browser like Netscape Navigator or Microsoft Internet Explorer,
- Rweb modules: these are point and click interfaces that allow the user to perform standard statistical analyses on built-in or user-supplied datasets.

Rweb currently has modules for summary statistics, two-way tables, ANOVA and linear regression. We developed four additional modules for calculating the optimal two-stage sampling strategies in Reilly (1996). The following section briefly describes the implementation of these modules. The rest of the document will explain how to install and configure them.

7.1 Implementation of the optimal sampling modules:

Rweb modules are implemented as dynamically created HTML forms. Behind the scenes, the scripts collect user choices and convert them to R code. Rweb then runs R, in batch mode, with the submitted code, and returns the output (printed and graphical) in standard HTML format. The optimal sampling modules were developed using this same approach. Each module is implemented in two script files. The **setup** script builds the HTML form where users select the options they want. The **run** script analyzes and validates user input and builds a text file that contains the appropriate R commands for the selected options. For our application we created a cut down version of Rweb with new improved opening and help pages.

This version excludes the code window in order to minimise the security risks associated with giving direct access to R and operating system commands.

7.2 Software needed to run Optimal Sampling Software on Rweb

1. Unix/Linux Operating System:

Rweb was originally developed on a Sun workstation. We installed and tested Rweb on an Intel Pentium machine running RedHat Linux version 6.2 without any modifications. Linux is a free open source operating system that is widely used to power Internet servers. For more information on downloading and installing Linux visit www.redhat.com.

2. Unix Web server e.g. the Apache Web Server:

We used Apache to test Rweb in our setup. Rweb does not use any specific feature of the Apache server so it should run on any Unix Web server. Apache is a free open source web server and is available for download from many web sites including www.apche.org.

3. R version 1.1.1 or greater:

R is downloadable from http://www.r-project.org.

4. Perl version 5.004 or greater:

Perl is downloadable from http://www.perl.com/pace/pub/perldocs/latest.html.

5. The following Perl modules:

- LWP Perl module for accessing URL's through Perl. The LWP module is part of libwww and is available at: http://www.perl.com/CPAN-local//CPAN.html#www.
- CGI Perl module for uploading local files and formatting some of the html output. It can be downloaded from < http://stein.cshl.org/WWW/software/CGI/>.

6. Ghostscript.

Ghostscript is available at:http://www.cs.wisc.edu/~ghost/aladdin/get510.html.

7. The NetPBM libaray (pstopnm, ppmtogif, pnmcrop, and pnmflip)

This is used by Rweb to convert R images to GIF images. It can be downloaded from ftp://wuarchive.wustl.edu/graphics/graphics/packages/NetPBM/.

8. The Rweb package, version 1.03:

Downloadable from http://www.math.montana.edu/Rweb/Resources.html. Read the included Readme file for information on how to install and configure Rweb.

9. The optimal sampling R package (Section 5.2).

This library contains the optimal two-stage sampling functions. Consult your R documentation for information on how to install R add-on packages. The package is available from <http://www.ucc.ie/depts/ucc/pubh/programs/programs.html.>.

10. The optimal sampling Rweb modules.

This module moves the CGI scripts to the **AnalysisModules** subdirectory in the Rweb CGI directory. It also moves the data files and their associated description files to the **DataSets** directory.

8 Using the Rweb optimal sampling modules

The opening HTML page introduces the three RWeb modules for calculating the optimal two-stage sampling designs as explained above. Rweb is a Web based interface to R (a statistical analysis package) that takes the user-submitted code, runs R on the code (in batch mode), and returns the output (printed and graphical).

Clicking on a link at the bottom of this opening page brings the user to the main screen. This screen allows the user to:

1. **Select a module to use**.

   There are three modules and one supporting function named "coding". The **fixed.n** function calculates the sampling fractions at the second stage (given fixed first- and second-stage sample sizes) which will minimise the variance of a specified co-efficient in the regression model. The **budget** function calculates the first-stage sample size and the second stage sampling fractions that will maximise precision of a specified co-efficient subject to a given budget. The **precision** function calculates the first-stage sample size and the second stage sampling fractions that will minimise cost subject to a given precision for a specified co-efficient. The supporting function, **coding** combines two or more first stage covariates into a vector i.e. it recodes a matrix of categorical variables into a vector that takes a unique value for each combination. Before running any of the three modules, you should run the coding function, to see in which order you must supply the vector of prevalences.

2. **Supply the dataset to be used in the analysis**.

   Small datasets can be copied and pasted into the text box near the bottom of the screen. One can also type in a URL for a Web-accessible dataset. For testing purposes, there are two built-in datasets provided (**cass1** or **cass2**). The data must be in text format where lines represent observations and columns represent variables, are separated by spaces. The first line should contain the variable names separated by spaces.

When the module and a dataset have been selected the user clicks the ‘submit’ button. The system will attempt to open the dataset and report any errors encountered in the process. If no errors were detected, you will be presented with the analysis page. The options in this page will depend on the module you selected in the previous step. You simply follow the instructions on the screen and click ‘submit’ when you have provide d all the parameters required for the analysis.

APPENDIX

Appendix A

Results of Comparative Study of Meanscore and Hotdeck

Table A-1 Simulation Studies to Compare Meanscore and Hotdeck Multiple Imputation

N	ρ^v	No. Imp	Method	Random Sampling					Balanced Samping				
				Mean (β)	Monte Carlo SE	Estimated SE	Bias (%)	95 % coverage	Mean (β)	Monte Carlo SE	Estimated SE	Bias (%)	95 % coverage
200	0.5	10	Meanscore	1.032	0.2335	0.2268	-2.87	0.936	1.046	0.2426	0.2283	-5.89	0.948
			ABB	1.044	0.2420	0.2282	-5.70	0.942	1.055	0.2468	0.2286	-7.37	0.938
			Hotdeck	1.037	0.2370	0.2101	-11.35	0.912	1.05	0.2462	0.2131	-13.44	0.918
200	0.25	10	Meanscore	1.023	0.3166	0.2896	-8.53	0.934	1.046	0.3084	0.2866	-7.07	0.948
			ABB	1.052	0.3332	0.2934	-11.94	0.924	1.072	0.3222	0.2854	-11.42	0.942
			Hotdeck	1.029	0.3195	0.2186	-31.58	0.850	1.053	0.3113	0.2218	-28.75	0.844
200	0.5	3	Meanscore	1.037	0.2388	0.2284	-4.36	0.938	1.037	0.243	0.2279	-6.21	0.944
			ABB	1.048	0.2520	0.2336	-7.30	0.936	1.045	0.2532	0.2309	-8.81	0.942
			Hotdeck	1.04	0.2444	0.2149	-12.07	0.928	1.041	0.2476	0.2134	-13.81	0.906
200	0.25	3	Meanscore	1.064	0.3097	0.2931	-5.36	0.940	1.057	0.2925	0.2884	-1.40	0.960
			ABB	1.096	0.3482	0.2924	-16.03	0.932	1.087	0.32	0.302	-5.63	0.936
			Hotdeck	1.068	0.3146	0.2250	-28.48	0.866	1.068	0.3035	0.2276	-25.01	0.862

Comparison of Meanscore, weighted likelihood (WL), pseudo likelihood (PL) and nonparametric maximum likelihood (ML)

Table A-2 Comparison of *meanscore* and other likelihood based methods using 1000 2nd stage observations from CASS data

(optimal with respect to age, see section 6.3)

Variable	Meanscore		WL		PL		ML	
	Coef	SE	Coef	SE	Coef	SE	Coef	SE
Intercept	-7.75272	***0.62692***	-7.75284	***0.67666***	-7.59179	***0.63523***	-7.59180	***0.63485***
Sex	0.59564	***0.17130***	0.59564	***0.17340***	0.61852	***0.16995***	0.61852	***0.16486***
Age	0.06801	***0.01053***	0.06801	***0.01141***	0.06515	***0.01064***	0.06515	***0.01066***

Table A-3 Comparison of *meanscore* and other likelihood based methods using 1000 2nd stage observations from CASS data

(optimal with respect to left ventricular blood pressure, see section 6.3)

Variable	Meanscore		WL		PL		ML	
	Coef	SE	Coef	SE	Coef	SE	Coef	SE
(Intercept)	-6.09724	***1.00602***	-6.09740	***1.11720***	-5.91887	***1.01303***	-5.93357	***1.00116***
Sex	0.21135	***0.21497***	0.21134	***0.22433***	0.27818	***0.20438***	0.29358	***0.19507***
Weight	-0.01584	***0.00797***	-0.01584	***0.00817***	-0.01563	***0.00772***	-0.01533	***0.00738***
Age	0.04594	***0.01194***	0.04595	***0.01371***	0.04304	***0.01248***	0.04284	***0.01247***
Angina	0.13795	***0.29977***	0.13796	***0.30433***	0.15164	***0.29085***	0.15032	***0.29080***
Chf	0.36390	***0.09456***	0.36391	***0.09486***	0.40086	***0.09380***	0.40235	***0.09355***
LVDBP	0.02521	***0.01172***	0.02521	***0.01137***	0.02112	***0.01107***	0.02125	***0.01107***
Surg	1.04214	***0.19520***	1.04215	***0.20183***	0.99580	***0.18658***	1.00156	***0.18634***

Table A-4 Comparison of *meanscore* and other likelihood based methods using the Ectopic pregnancy data (see section 6.3)

Variable	Meanscore		WL		PL		ML	
	Coef	SE	Coef	SE	Coef	SE	Coef	SE
Intercept	-0.78892	***0.21727***	-1.58107	***0.28143***	-1.55017	***0.28756***	-1.60172	***0.28130***
Chlam	0.90485	***0.31592***	0.85737	***0.31862***	0.89721	***0.30574***	0.89400	***0.30467***
Gonn	-0.05172	***0.30077***	0.06865	***0.33482***	0.03353	***0.35763***	0.05117	***0.32208***
Contracept	-2.36030	***0.18715***	-2.25818	***0.23860***	-2.25721	***0.23934***	-2.26657	***0.23400***
sexpatr	0.74156	***0.23526***	0.85534	***0.31331***	0.81079	***0.30252***	0.87355	***0.30779***

Table A-5 Comparison of *meanscore* and other likelihood based methods using the NWTSG data with Institutional Histology as the first stage variables

Variable	Meanscore		WL		PL		ML	
	Coef	SE	Coef	SE	Coef	SE	Coef	SE
Intercept	-3.2032	***0.5708***	-3.2035	***0.5648***	-3.2408	***0.5518***	-3.2590	***0.5470***
Stage I	-1.1781	***0.2138***	-1.1787	***0.2116***	-1.0150	***0.2085***	-1.0172	***0.2087***
Stage II	-0.3742	***0.2150***	-0.3742	***0.2129***	-0.2635	***0.2064***	-0.2647	***0.2065***
Stage III	-0.1461	***0.2152***	-0.1461	***0.2110***	-0.1400	***0.2067***	-0.1407	***0.2067***
Central Hist	1.8020	***0.1522***	1.8022	***0.1496***	1.7876	***0.1526***	1.8099	***0.1324***
Age	0.0109	***0.0022***	0.0109	***0.0022***	0.0117	***0.0021***	0.0117	***0.0021***
Study	-0.2088	***0.1394***	-0.2088	***0.1383***	-0.2270	***0.1332***	-0.2282	***0.1331***

Table A-6 Comparison of *meanscore* and other likelihood based methods using the NWTSG data with Institutional Histology and Stage of Tumor as the first stage variables

Variable	Meanscore		WL		PL		ML	
	Coef	SE	Coef	SE	Coef	SE	Coef	SE
Intercept	-3.2813	***0.5495***	-3.2815	***0.5398***	-3.2565	***0.5300***	-3.2283	***0.5282***
Stage I	-1.1330	***0.1718***	-1.1335	***0.1686***	-0.9525	***0.1659***	-1.0476	***0.1502***
Stage II	-0.3572	***0.1646***	-0.3572	***0.1610***	-0.2448	***0.1578***	-0.3210	***0.1408***
Stage III	-0.3088	***0.1627***	-0.3088	***0.1566***	-0.2824	***0.1575***	-0.3340	***0.1384***
Central Hist	1.8554	***0.1530***	1.8555	***0.1481***	1.7844	***0.1498***	1.8527	***0.1293***
Age	0.0113	***0.0021***	0.0114	***0.0021***	0.0119	***0.0020***	0.0115	***0.0021***
Study	-0.2053	***0.1392***	-0.2053	***0.1369***	-0.2241	***0.1316***	-0.2299	***0.1326***

Appendix B

INSTALLATION GUIDELINES

1. Meanscore package in R

The Mean Score method for missing and auxiliary covariate data is described in the paper by Reilly & Pepe in Biometrika (1995). This likelihood-based method is asymptotically equivalent to "hot-deck" multiple imputation (Reilly & Pepe, 1997). Missingness may depend on the available response and covariate values but not on the unobserved covariate values (i.e. MAR, Missing At Random) and the method is applicable to cohort or case-control designs. The subsample of subjects on whom the incomplete covariate is available is referred to as the "validation sample" or the "second-stage sample", and the remaining subjects are the "non-validation sample" or the "first-stage sample"

The code provided here implements a Mean Score analysis for a logistic regression model where the incomplete covariate(s) may be continuous, but the first stage covariates and/or auxiliary variables must be categorical.

INSTALLATION GUIDE
The simplest installation can be done by unzipping the BINARY package directly to R_HOME/library.

For users familiar with building R packages the full features of the R help systems can be made available by using the standard command:

`make BUILD=option pkg-meanscore'

executed from R_HOME/src/gnuwin32, after you have unzipped this package to R_HOME/src/library.

Alternatively you can use the UNIX-type command:
`Rcmd install meanscore'
executed from R_HOME/src/library

With the simpler installation the function will still work properly but some nice features of the help systems may be lost.

After you have installed the package, the command

```
> library(meanscore)
```

makes available the following 3 functions:

1. MEANSCORE: this function is called with the combined first- and second-stage data where the missing values in the incomplete covariate(s) are represented by NA, the usual notation in Splus/R.
2. MS.NPREV: this function is called with the second-stage (i.e. complete) data and the first-stage sample sizes (or prevalences): if only prevalences are available, then estimates are provided but no standard errors. Prior to running this function, the CODING function (3.) should be run to see the order in which MS.NPREV expects the first-stage sample sizes or prevalences to be provided.
3. CODING: this function recodes multiple columns of first-stage covariates into a single vector and displays the coding scheme

Help on these functions and on the illustrative data sets provided can be viewed using help or ? or the HTML help file system.

This code has only been tested under R 1.2.0 for Windows, and may need some modifications for use with other versions or other operating systems.

We would be happy to hear about any bugs that you find, and to receive any comments or suggestions for improvements.

Marie Reilly PhD. & Agus Salim
Dept. of Epidemiology Dept. of Statistics
University College Cork, University College Cork,
Ireland Ireland
E-mail: marie.reilly@ucc.ie agus@stat.ucc.ie

2. Meanscore package in S-PLUS

The Mean Score method for missing and auxiliary covariate data is described in the paper by Reilly & Pepe (Biometrika, 1995). This likelihood-based method is asymptotically equivalent to "hot-deck" multiple imputation (Reilly & Pepe, Stats in Medicine 1997). Missingness may depend on the available response and covariate values but not on the unobserved covariate values (i.e. MAR, Missing At Random) and the method is applicable to cohort or case-control designs. The subsample of subjects on whom the incomplete covariate is available is referred to as the "validation sample" or the "second-stage sample", and the remaining subjects are the "non-validation sample" or the "first-stage sample"

The code provided here implements a Mean Score analysis for a logistic regression model where the incomplete covariate(s) may be continuous, but the first stage covariates and/or auxiliary variables must be categorical.

After extracting the ZIP file in the "library" subfolder of your SPlus directory, the command

```
> library(meanscore)
```

makes available the following 3 functions:

1. MEANSCORE: this function is called with the combined first- and second-stage data where the missing values in the incomplete covariate(s) are represented by NA, the usual notation in Splus/R.
2. MS.NPREV: this function is called with the second-stage (i.e. complete) data and the first-stage sample sizes (or prevalences): if only prevalences are available, then estimates are provided but no standard errors. Prior to running this function, the CODING function (3.) should be run to see the order in which MS.NPREV expects the first-stage sample sizes or prevalences to be provided.
3. CODING: this function recodes multiple columns of first-stage covariates into a single vector and displays the coding scheme

```
Help on these functions and on the illustrative data sets
provided can be viewed using help or ?

This code has only been tested under S-PLUS 4.0 for Windows,
and may need some modifications for use with other
versions or other operating systems.

We would be happy to hear about any bugs that you find, and to
receive any comments or suggestions for improvements.

Marie Reilly                  &     Agus Salim
Dept. of Epidemiology               Dept. of Statistics
University College Cork,            University College Cork,
Ireland                             Ireland
E-mail: marie.reilly@ucc.ie         E-mail: agus@stat.ucc.ie
```

3. Meanscore package in STATA

```
SUBJECT: Meanscore algorithm for missing covariate data in logistic
         regression models

AUTHORS: Marie Reilly
       Dept. of Epidemiology & Public Health

       and

       Agus Salim
       Dept. of Statistics,

       University College Cork (UCC) Cork, Ireland

SUPPORT: marie.reilly@ucc.ie


INSTALLATION (Stata version 6):

Installing from internet, please check:
```

```
[U] 20.6 How do I install an addition?
[R] net

Installing from floopy disk

type 'net from a:'
     'net install meanscor'   to install program
     'net get meanscor'       to access illustrative datasets


Installing from C drive:

type 'net from c:/[dirname]'
     'net install meanscor'    to install program
     'net get meanscor'        to access illustrative datasets

where [dirname] is the name of the directory where you put the source
files.


HELP

After installation, from inside Stata online help is available
by typing

.help meanscor

.help msnprev

.help coding
```

4. Optimal package in R

This library contains functions for calculating the optimal two-stage sampling strategies in Reilly (1996). Briefly, the methods are applicable in studies where some categorical covariates (Z) and a dichotomous outcome variable (Y) are to be measured at the first stage and additional covariates (X, which may be continuous) are to be gathered on a subsample at the second stage. Logistic regression analysis of all the data will then proceed using the Mean Score method (Reilly and Pepe, 1995). In addition to the total sample size, the variance of the Mean Score estimate depends on the second-stage sampling fractions in each of the (Y,Z) strata. Hence the study size and/or the second-stage sampling fractions can be "optimised".

The three functions here provide the optimal sampling strategies under different constraints:

fixed.n	calculates the sampling fractions at the second stage (given fixed first- and second-stage sample sizes) which will minimise the variance of a specified coefficient in the regression model.
budget	calculates the first-stage sample size and the second stage sampling fractions that will maximise precision of a specified coefficient subject to a given budget.
precision	calculates the first-stage sample size and the second stage sampling fractions that will minimise cost subject to a given precision for a specified coefficient.

Each of the functions requires pilot data on (Z,Y,X) as input:
this would typically be a small number of X observations in each of the (Y,Z) strata. Knowledge is also required of the prevalences of these strata in the population, which can be provided as estimates or can be computed from the first-stage data if available.

INSTALLATION GUIDE
The simplest installation can be done by unzipping the BINARY package directly to R_HOME/library.

For users familiar with building R packages, the following command can be used to install the package:

`make BUILD=option pkg-optimal'

executed from R_HOME/src/gnuwin32, after you have unzipped this package to R_HOME/src/library.

Alternatively you can use the UNIX command:

`Rcmd install optimal'

executed from R_HOME/src/library

After the package has been installed, the command:

library(optimal)

will make the functions available. Detailed help on each function and on the illustrative data sets (cass1, cass2) can then be viewed by using help or ? or the HTML help file system.

This code has been tested under R1.2.0 for windows, some changes may be needed for other versions or operating systems.

We would be happy to hear about any bugs that you find, and to receive any comments or suggestions for improvements.

Marie Reilly & Agus Salim
Dept. of Epidemiology Dept. of Statistics
University College Cork, University College Cork,
Ireland Ireland
E-mail: marie.reilly@ucc.ie E-mail: agus@stat.ucc.ie

5. Optimal package in S-PLUS

This library contains functions for calculating the optimal two-stage sampling strategies in Reilly (1996). Briefly, the methods are applicable in studies where some categorical covariates (Z) and a dichotomous outcome variable (Y) are to be measured at the first stage and additional covariates (X, which may be continuous) are to be gathered on a subsample at the second stage. Logistic regression analysis of all the data will then proceed using the Mean Score method (Reilly and Pepe, 1995). In addition to the total sample size, the variance of the Mean Score estimate depends on the second-stage sampling fractions in each of the (Y,Z) strata. Hence the study size and/or the second-stage sampling fractions can be "optimised".

The three functions here provide the optimal sampling strategies under different constraints:

fixed.n	calculates the sampling fractions at the second stage (given fixed first- and second-stage sample sizes) which will minimise the variance of a specified coefficient in the regression model.
budget	calculates the first-stage sample size and the second stage sampling fractions that will maximise precision of a specified coefficient subject to a given budget.
precision	calculates the first-stage sample size and the second stage sampling fractions that will minimise cost subject to a given precision for a specified coefficient.

Each of the functions requires pilot data on (Z,Y,X) as input: this would typically be a small number of X observations in each of the (Y,Z) strata. Knowledge is also required of the prevalences of these strata in the population, which can be provided as estimates or can be computed from the first-stage data if available.

After extracting the ZIP file in the "library" subfolder of your SPLUS directory, the command

library(optimal)

will make the functions available. Detailed help on each function and on the illustrative data sets (cass1, cass2) can then be viewed by issuing the help or ? command.

The library has only been tested under S-PLUS 4 for windows, so some changes may be needed for other versions or operating systems.

We would be happy to hear about any bugs that you find, and to receive any comments or suggestions for improvements.

Marie Reilly & Agus Salim
Dept. of Epidemiology Dept. of Statistics
University College Cork, University College Cork,
Ireland Ireland
E-mail: marie.reilly@ucc.ie E-mail: agus@stat.ucc.ie

6. Optimal package in STATA

SUBJECT: Optimal sampling designs using the Meanscore algorithm

AUTHORS: Marie Reilly
Dept. of Epidemiology & Public Health
and
Agus Salim
Dept. of Statistics,
University College Cork (UCC) Cork, Ireland

SUPPORT: marie.reilly@ucc.ie

```
INSTALLATION (Stata version 6):

Installing from internet, please check:

[U] 20.6 How do I install an addition?
[R] net

Installing from floopy disk

type 'net from a:'
     'net install optimal'          to install program
     'net get optimal'        to access illustrative datasets


Installing from C drive:

type 'net from c:/[dirname]'
     'net install optimal'    to install program
     'net get optimal'        to access illustrative datasets

where [dirname] is the name of the directory where you put the source
files.

HELP

After installation, from inside Stata online help is available
by typing
.help coding

.help optfixn

.help optbud

.help optprec
```

Bibliography

Banfield, J. (1999). Rweb: web based statistical analysis. *Journal of Statistical Software* **4**: 1-15.

Breslow, N.E. and Cain, K.C. (1988). Logistic regression for two-stage case control data. *Biometrika* **75**: 11-20

Breslow, N.E. and Holubkhov, R. (1997). Maximum likelihood estimation of logistic regression parameters under two-phase, outcome-dependent sampling. *J.R.Statist. Soc. B.* **59**: 447-61

Breslow, N.E. and Chatterjee, N. (1999). Design and analysis of two-phase studies with binary outcome applied to Wilms tumor prognosis. *Appl. Statist.* **48**: 457-68

Dempster, A.P., Laird, N.M., Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J.R.Statist. Soc. B.* **39**: 1-38

Flanders, W.D. and Greenland, S. (1991). Analytic methods for two-stage case-control studies and other stratified designs. *Statist. Med.* **10**: 739-47.

Fletcher, R. (1987). *Practical methods of optimization*. Chichester : Wiley

Hamilton, L.C. (1997). *Statistics with STATA 5*. Brooks/Cole Pub Co.

Holcroft, C. and Spiegelman, D. (1999). Design of validation studies for estimating the odd ratio of exposure-disease relationships when exposure is misclassified. *Biometrics* **55**: 1193-1201.

Hornik, K. (2000). The R FAQ. http://www.ci.tuwien.ac.at/~hornik/R.

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

Mander, A. and Clayton, D. (1999). sg116: Hotdeck imputation. *STATA Technical Bulletin* **54**: 32-4.

Rabe-Hesketh, S. and Everitt, B. (2000). *A handbook of statistical analyses using Stata*. Second Edition. London: CRC.

Reilly, M and Pepe, MS. (1995). A mean score method for missing and auxiliary outcome covariate data in regression models. *Biometrika* **82**: 299-314

Reilly, M. (1996). Optimal sampling strategies for two-stage studies. *Amer. J. Epidemiol*. 143:92-100

Reilly M and Pepe MS (1997). The relationship between hot-deck multiple imputation and weighted likelihood. *Statist. Med.***16**:5-19.

Rweb: Statistical analysis on the web. http://www.math.montana.edu/Rweb/

Salim, A. and Reily, M. (2000). Practical problems arising in computing optimal sampling designs for two-stage studies. Presentation at the Research student conference (RSC 2000), Cardiff.

Sherman, K.J., et.al. .1990. Sexually transmitted diseases and tubal pregnancy. *Sex. Transm.Dis*. **7**: 115-21

SPLUS. http://www.mathsoft.com/splus

STATA. http://www.stata.com/

The R Project for Statistical Computing. http://www.r-project.org

Venables, W.N. and Ripley, B.D. (1999). *Modern Applied Statistics with S-PLUS*. Third Edition. Springer.

Venables, W. N. and Ripley, B.D. (2000). *S Programming*. Springer

Vliestra,R.E.,Frye R.L., Kromnal R.A.,et.al.(1980). Risk factors and angiographic coronary artery disease: a report from the Coronary Artery Surgery Study (CASS). *Circulation* **62**:254-61