Component of Statistics Canada
Catalogue no. 12-001-X Business Survey Methods Division

Article

The construction of stratified designs in R with the package *stratification*

by Sophie Baillargeon and Louis-Paul Rivest

June 2011

Statistics Canada Statistique Canada

Canada

Survey Methodology, June 2011
Vol. 37, No. 1, pp. 53-65
Statistics Canada, Catalogue No. 12-001-X

The construction of stratified designs in R with the package *stratification*

Sophie Baillargeon and Louis-Paul Rivest [1]

Abstract

This paper introduces a R-package for the stratification of a survey population using a univariate stratification variable X and for the calculation of stratum sample sizes. Non iterative methods such as the cumulative root frequency method and the geometric stratum boundaries are implemented. Optimal designs, with stratum boundaries that minimize either the CV of the simple expansion estimator for a fixed sample size n or the n value for a fixed CV can be constructed. Two iterative algorithms are available to find the optimal stratum boundaries. The design can feature a user defined certainty stratum where all the units are sampled. Take-all and take-none strata can be included in the stratified design as they might lead to smaller sample sizes. The sample size calculations are based on the anticipated moments of the survey variable Y, given the stratification variable X. The package handles conditional distributions of Y given X that are either a heteroscedastic linear model, or a log-linear model. Stratum specific non-response can be accounted for in the design construction and in the sample size calculations.

Key Words: Linear models; Log-linear models; Optimal stratification; Survey sampling; Take-all stratum; Take-none stratum.

1. Introduction

The establishment of strata and the planning of a stratified design have been important topics in survey sampling, since the pioneering contributions of Dalenius more than sixty years ago. This work is concerned with univariate stratification where the strata are constructed using a positive stratification variable X known for all the units of the population. X is assumed to be related to the survey variable Y. Stratum h contains all the units with an X-value in the interval $[b_{h-1}, b_h)$ for $h = 1, \ldots, L$ such that $b_0 = \min X$ and $b_L = \max X + 1$, where $\min X$ and $\max X$ are respectively the minimum and the maximum values of the stratification variable.

The determination of optimal stratum boundaries has a long history, see chapter 5A of Cochran (1977). The cumulative root frequency method (cum $\sqrt{f}$) of Dalenius and Hodges (1959) provides an approximate solution to this problem. Instances where X has a skewed distribution are frequent in business surveys and have been given a special emphasis. Gunning and Horgan (2004) proposed a geometric stratification method and Hidiroglou (1986) argued that the large units should be put in a take-all stratum. Rather than relying on an approximate method for constructing the strata, Lavallée and Hidiroglou (1988) suggested an iterative algorithm that gives the optimal boundaries for a particular X variable. Their algorithm sometimes fails to converge (Detlefsen and Veum 1991) and Slanta and Krenzke (1996) have shown that in some cases the optimal boundaries are not uniquely defined. Alternative methods, such as the search algorithm of Kozak (2004), have been proposed to alleviate some of these difficulties. The assumption that the survey variable Y is the same as the stratification variable X is not realistic when calculating sample sizes and several authors, including Dayal (1985) and Sigman and Monsour (1995), proposed to allocate the sample to the strata on the basis of the *anticipated* moments of Y knowing that X is in $[b_{h-1}, b_h)$. Sweet and Sigman (1995) and Rivest (1999, 2002) suggested using these anticipated moments in the stratification algorithm of Lavallée and Hidiroglou (1988). Recently, Baillargeon and Rivest (2009) showed that putting the small units in a take-none stratum, which is not sampled, might reduce the sample size needed to reach a predetermined precision level.

This article introduces the R-package *stratification* that implements most of the methods presented above. It provides a friendly computer environment to build stratified designs and to evaluate their performance on some real populations. This package is presented by revisiting examples in the stratification literature selected to illustrate its important features. The four functions of *stratification* with the prefix `strata` construct stratified sampling designs. These functions are `strata.cumrootf`, `strata.geo`, `strata.LH`, and `strata.bh`. The first two implement the simple cum $\sqrt{f}$ and geometric stratification methods. The function `strata.LH` derives optimal stratified sampling plans using iterative algorithms while the last function handles user defined stratum boundaries. These four functions construct strata, determine stratum sample sizes and calculate the precision of the simple expansion estimator $\bar{y}_s$ of $\bar{Y}$, the population mean of some survey variable Y related to the stratification variable X.

1. Sophie Baillargeon, Département de mathématiques et de statistique, 1045, avenue de la médecine, Université Laval, Québec, (Qc) Canada G1V 0A6. E-mail: sophie.baillargeon@mat.ulaval.ca; Louis-Paul Rivest, Département de mathématiques et de statistique, 1045, avenue de la médecine, Université Laval, Québec, (Qc) Canada G1V 0A6. E-mail: louis-paul.rivest@mat.ulaval.ca.

The four `strata`-functions use Hidiroglou and Srinath's (1993) rule to allocate the n units in the sample to the strata. The stratum sample sizes are proportional to $N_h^{2q_1}$ $\overline{Y}_h^{2q_2}$ $S_{yh}^{2q_3}$, where N_h is the size of stratum h, and $\overline{Y}_h$ and S_{yh}^2 are the anticipated mean and variance of Y in stratum h. In the `strata`-functions, an allocation rule is specified by the argument `alloc` that contains the exponents (q_1, q_2, q_3); Neyman's allocation corresponds to `alloc=c(1/2,0,1/2)`. A `strata`-function takes as an input the population vector of the stratification variable X, the number of strata `Ls`, and a total sample size n or a target CV for the simple expansion estimator $\overline{y}_s$. Its output is an `R`-object of class "strata" that defines a stratified design. It contains a set of strata determined by their upper boundaries $\{b_h\}$ and stratum population and sample sizes, N_h and n_h. There is a fifth function in *stratification* called `var.strata` that takes as an input an `R`-object of class strata and a population vector of a survey variable Y and returns the variance of $\overline{y}_s$ for the input variable Y and the input stratified design.

The text contains `R` instructions to be typed in an `R` command window; these lines start with >. It also presents outputs printed in an R command window. A special typeface allows an easy identification of these `R` instructions and print-outs in the text. The appendix contains a summary table that lists all the possible arguments of the five *stratification* functions. When using this package, the `R`-instruction `help(stratification)` calls a clickable help file that provides detailed information on the package and examples that can be pasted in a command window.

2. Basic stratification methods

This section discusses two elementary stratification methods, the cumulative root frequency method of Dalenius and Hodges (1959) and the geometric method of Gunning and Horgan (2004). These two methods are exact; they do not rely on an iterative algorithm. Throughout this section $Y = X$, so that the variance of $\overline{y}_s$ is evaluated using the values of the stratification variable X. Using the same variable to stratify a population and to evaluate the precision of survey estimates might underestimate their variances. The calculation of variances when $Y \neq X$ is considered in Section 4.

2.1 Cumulative root frequency method

This stratification algorithm, presented in chapter 5A of Cochran (1977), is implemented by the function `strata.cumrootf`. Its arguments are `x`, the population vector of the stratification variable, `nclass` the number of bins of equal size for the `x`-variable, a target `CV` for $\overline{y}_s$ or a predetermined sample size `n`, the number of strata `Ls`, and an allocation rule `alloc`. This algorithm pools the `nclass` bins into `Ls` strata in such a way that the sums of the square roots of the bin frequencies are approximately equal for the `Ls` strata.

As an illustration, consider the proportion of industrial loans of N = 13,435 banks used in Cochran (1961). We stratify this population and evaluate the sample size needed for $\overline{y}_s$ to have a CV of 5% when Neyman allocation is used. The following `R`-code creates the vector of the stratification variable `loans` from Table 2 of McEvoy (1956). The function `strata.cumrootf` is then applied to the loans variable. Following Table 2 of Cochran (1961), `nclass` is set to 20 so that the strata will be created using 20 bins and `Ls=3` strata will be constructed. The output is placed in `cum`, an `R`-object of class strata. Typing `cum` or `print(cum)` in the `R` command window prints details of the sampling plan. The input arguments, either the default or as specified by the user, appear first. Then stratum information is provided such as boundaries, sizes N_h and sample sizes n_h. The third part of the print-out provides information about the sampling properties of $\overline{y}_s$.

```
> values <- c(seq(0.5, 9.5, 1), seq(12.5, 97.5, 5))
> nrep <- c(1985, 261, 339, 405, 474, 478, 506, 569, 464, 499,
  2157, 1581, 1142, 746, 512, 376, 265, 207, 126, 107, 82, 50,
  39, 25, 16, 19, 2, 3)
> loans <- rep(values, nrep)
> cum <- strata.cumrootf(x = loans, nclass = 20, CV = 0.05,
  Ls = 3, alloc = c(0.5, 0, 0.5))
> cum

Given arguments:
x = loans
nclass = 20, CV = 0.05, Ls = 3
allocation : q1 = 0.5, q2 = 0, q3 = 0.5
model = none

Strata information:
           rh |   bh anticip.Mean anticip.var    Nh   nh    fh
Stratum 1   1 | 10.2         4.12       10.46  5980   14  0.00
Stratum 2   1 | 29.6        17.92       27.74  5626   20  0.00
Stratum 3   1 | 98.5        44.47      165.83  1829   16  0.01
Total                                    13435   50  0.00

Total sample size: 50
Anticipated population mean: 15.39408
Anticipated CV: 0.0494897
```

In the `Given arguments, model=none` means that the sampling properties of $\overline{y}_s$, presented at the end of the print-out, are evaluated at $Y = X$, that is for the `loans` variable. Its mean is 15.39408 and the anticipated `CV` of 0.0494897 is that of the estimator $\overline{y}_s$ of the mean of the variable `loans` obtained with this sampling design. The stratum boundaries given in this output are (10.2, 29.6, 98.5), they are equal to those appearing at the bottom of page 349 of Cochran (1961), once the rounding used for creating the vector `loans` is accounted for. In the `Strata Information`, r_h refers to the stratum response rates that are discussed in Section 5.1. The `R`-object `cum` contains several elements that are listed by the command `names(cum)`.

```
> names(cum)
 [1] "Nh"           "nh"   "n"          "nh.nonint" "certain.info"
 [6] "opti.criteria" "bh"   "meanh"      "varh"      "mean"
[11] "stderr"       "CV"   "stratumID"  "nclassh"   "takeall"
[16] "call"         "date" "args"
```

An element in the cum strata object can be printed by typing cum$ followed by the name of the object. For instance the cum$stratumID prints the stratum of each unit in the population. The variable cum$nclassh is specific to the strata.cumrootf function; it gives how the nclass=20 original bins have been pooled into three strata;

```
> cum$nclassh
[1] 2 4 14
```

Thus, in this stratification, strata 1, 2 and 3 contain respectively 2, 4 and 14 of the nclass=20 original bins.

2.2 Geometric method

The geometric stratification method has been introduced by Gunning and Horgan (2004). It sets the stratum boundaries to $b_h = \min X \times (\max X / \min X)^{h/L}$, for $h = 1, ..., L-1$. Once the boundaries b_h are determined, the stratum sample size calculations are the same as those carried out in strata.cumrootf.

As an illustration we stratify the four populations presented in Gunning et Horgan (2004), Debtors, USbanks, UScities, and UScolleges, into Ls=5 strata. The last three populations were considered in Cochran's (1961) investigations. These four populations are stored in *stratification*; the command data(Debtors) calls the first one. Rather than specifying a target CV we set the sample size to n = 100 following Gunning and Horgan (2004). The following commands create the R-object pop1 that contains the stratified design for the Debtors population.

```
> data(Debtors)
> pop1 <- strata.geo(x = Debtors, n = 100, Ls = 5,
  alloc = c(0.5, 0, 0.5))
```

Table 1 summarizes the geometric stratified designs for the four study populations. It reproduces Table 4 of Gunning and Horgan (2004) partially. There are however some minor differences caused by different rounding strategies. More details about *stratification* rounding methods are available in the help file.

Table 1
Stratified designs for four populations with n = 100

Population	CV		1	2	3	4	5
Debtors	0.0359	b_h	148.28	549.67	2,037.60	7,553.33	
		N_h	1,054	1,267	732	265	51
		n_h	3	14	27	33	23
UScities	0.0145	b_h	18.17	33.01	59.98	108.98	
		N_h	364	418	130	87	39
		n_h	18	28	17	20	17
UScolleges	0.0183	b_h	434.00	941.76	2,043.61	4,434.60	
		N_h	94	255	198	74	56
		n_h	3	15	27	20	35
USbanks	0.0107	b_h	118.59	200.92	340.39	576.68	
		N_h	114	116	64	39	24
		n_h	13	20	25	18	24

2.3 Take-all stratum

In Table 1, the fifth stratum for the USbanks population is a take-all stratum since $n_5 = N_5 = 24$. Under Neymann allocation, the fifth stratum gets a sample size n_5 larger than the stratum size N_5. Then strata.geo automatically identifies this stratum as a take-all stratum and allocates the $n - N_5$ units for the first four strata using Neyman allocation. This adjustment is important to have a sample size of $n = 100$ as specified in the strata.geo arguments.

To illustrate this point, we use the function strata.bh to make an allocation without a take-all stratum adjustment. This function allocates the sample and calculates the precision of $\bar{y}_s$ for a predetermined set of stratum boundaries. By setting takeall.adjust=FALSE, Neyman allocation is used in the five strata and since $n_5 > N_5$ one has $n_5 = N_5$. The following R-code gets the geometric stratum boundaries $\{b_h\}$ in the strata object adjust; it then uses the strata.bh function with the geometric stratum boundaries to get the sampling design without adjusting for a take-all stratum five in the noadjust strata object.

```
> data(USbanks)
> adjust <- strata.geo(x = USbanks, n = 100, Ls = 5,
  alloc = c(0.5, 0, 0.5))
> noadjust <- strata.bh(x = USbanks, bh = adjust$bh,
  n = 100, Ls = 5, alloc = c(0.5, 0, 0.5), takeall = 0,
  takeall.adjust = FALSE)
```

The two designs are presented in Table 2. Failing to include a take-all stratum yields a sample size of n = 99, smaller than the target n = 100. In this case, the unrounded sample size for stratum 5 is noadjust$nh.noint[5]= 25.40 for N_5 = 24 units. Note that when n is large or when the target CV is small, it is possible to get several take-all strata.

Table 2
Stratified designs obtained with and without an automatic adjustment for a take-all stratum

	n		1	2	3	4	5
		b_h	118.59	200.92	340.39	576.68	
		N_h	114	116	64	39	24
adjust	100	n_h	13	20	25	18	24
noadjust	99	n_h	13	20	24	18	24

2.4 Adding a take-all stratum

We now consider the data base on N = 284 Swedish municipalities given in the appendix of Särndal, Swensson and Wretman (1992). The following instructions use the geometric method to stratify this population in Ls=5 strata using the variable REV84, the 1984 real estate values. The power allocation with exponent 0.7 and alloc=c(0.35,0.35,0) is used. The R-object of class strata geo contains the stratified design. The command plot(geo) produces the plot presented in Figure 1. It provides a histogram of the

stratification variable with the stratum boundaries and a summary table for the stratified design.

```
> data(Sweden)
> geo <- strata.geo(x = Sweden$REV84, CV = 0.05, Ls = 5,
  alloc = c(0.35, 0.35, 0))
```

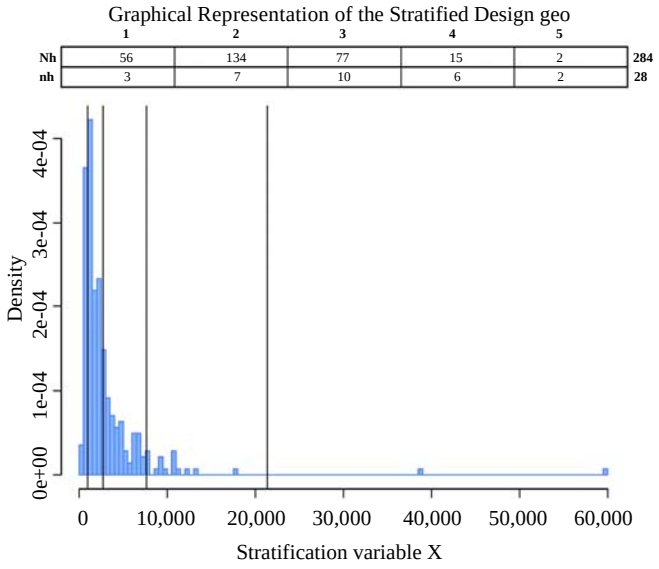

Figure 1 Plot of the R-object geo

Figure 1 shows that the geometric stratification method puts two of the three extreme REV84 values in a take-all stratum. The following Rcode creates cum a stratified design for this population using the cum $\sqrt{f}$ method. The application of this stratification method is awkward since the bins have length {max(REV84)−min(REV84)}/50= 1191. Considering Figure 1 most of the bins have a null frequency; indeed stratum 5 comprises 43 of the 50 bins. This design does not have a take-all stratum. To calculate the sample sizes obtained by requesting a take-all stratum one can use the function strata.bh, with the cum $\sqrt{f}$ boundaries stored in cum$bh, with the command takeall=1. This gives the third sampling plan in Table 3. The fourth sampling plan of Table 3 cum3 is created by setting the sample size in stratum 5 of the cum $\sqrt{f}$ design equal to its population size with the command cum3$nh[5]<−cum3$Nh[5]. The variance of the estimate $\bar{y}_s$ for the variable REV84 using this fourth sampling design is calculated using var.strata.

```
> cum <- strata.cumrootf(x = Sweden$REV84, nclass = 50,
  CV = 0.05, Ls = 5, alloc = c(0.35, 0.35, 0))
> cum2 <- strata.bh(x = Sweden$REV84, bh = cum$bh, CV = 0.05,
  Ls = 5, takeall = 1, alloc = c(0.35, 0.35, 0))
> cum3 <- cum
> cum3$nh[5] <- cum3$Nh[5]
> cum3.var <- var.strata(cum3, y = Sweden$REV84)
```

Table 3
Four stratified designs for the population of Swedish municipalities

Method		1	2	3	4	5	*n*	CV
geometric	N_h	56	134	77	15	2		
	n_h	3	7	10	6	2	28	4.83
cum $\sqrt{f}$	N_h	120	70	52	27	15		
	n_h	7	7	9	8	10	41	4.87
	n_h^{modif1}	2	2	3	2	15	24	4.44
	n_h^{modif2}	7	7	9	8	15	46	2.29

Table 3 highlights that the sampling fraction in the fifth stratum drives the value of *n*. The cum $\sqrt{f}$ design appears to be less efficient than the geometric design since it sampling fraction in stratum 5 is 10/15 = 67%. Requesting a take-all stratum gives a value of *n* comparable to that obtained with the geometric design. The REV84 population has three outliers that were identified in Table 1. The geometric and cum $\sqrt{f}$ stratification methods depend heavily on the maximum X-value; therefore before applying these techniques it might be wise to put the three outliers aside. This is considered in the next section.

The simple *ad hoc* method to arbitrarily change the stratum sample sizes presented in this section can be applied in several situations. For instance, when some strata have samples of size 1, they can be increased to 2 in order to have an unbiased variance estimator.

2.5 Certainty stratum

In a stratified design it might be useful to constrain some units to be sampled, before constructing the strata. The argument certain available in the four strata-function makes this possible. As an example we revisit the comparison of the cum $\sqrt{f}$ and the geometric sampling designs presented in Table 3. The three large municipalities highlighted in Figure 1 are put in a certainty stratum, and the $N = 281$ remaining municipalities are stratified into Ls=4 strata using the two stratification methods. The R-code for constructing these two designs is given below. The command x=sort(Sweden$REV84) orders the municipalities by increasing *REV*84; thus the three large municipalities are entries 282, 283 and 284 of the sorted vector. The two R objects of class strata, geo_cer and cum_cer, each contain an element certain.info that provides information on the certainty stratum.

```
> geo_cer <- strata.geo(x = sort(Sweden$REV84), CV = 0.05,
  Ls = 4, alloc = c(0.35, 0.35, 0), certain = 282:284)
> cum_cer <- strata.cumrootf(x = sort(Sweden$REV84),
  nclass = 50, CV = 0.05, Ls = 4, alloc = c(0.35, 0.35, 0),
  certain = 282:284)
> cum_cer$certain.info

   Nc      meanc
 3.00   38923.67
```

In Table 4, the cum $\sqrt{f}$ design is more efficient that the geometric design. Putting the three large municipalities in a certainty stratum is helpful since the sample sizes in Table 4 are smaller than those of Table 3. The argument `certain` can force any set of units in the sample. It can be used to include units that are extreme for a secondary variable, different from the stratification variable, or that have a history of high volatility.

Table 4
Two stratified designs for the Swedish municipalities constructed with a certainty stratum

Method		1	2	3	4	5	*n*	CV
geometric	N_h	42	116	88	35	3		
	n_h	2	5	7	7	3	24	4.71
cum $\sqrt{f}$	N_h	127	79	46	29	3		
	n_h	3	4	4	5	3	19	4.72

3. Optimization method

The stratification methods introduced in Section 2 do not always give an optimal stratified design, that minimizes the sample size n needed to reach the target CV (or minimizes the CV for a fixed n). This section introduces the function `strata.LH` that allows the determination of optimal designs. The name `LH` stands for Lavallée and Hidiroglou (1988) who pioneered the construction of optimal stratified designs for real life survey populations. In a stratified design with a take-all stratum, the variance of the simple expansion estimator is given by

$$\mathrm{Var}(\bar{y}_s) = \sum_{h=1}^{L-1}\left(\frac{N_h}{N}\right)^2\left(\frac{1}{(n-N_L)a_h} - \frac{1}{N_h}\right)S_{yh}^2,$$

where $\{a_h\}$ is the allocation rule for setting stratum sample sizes. The n that ensures a CV of c is given by

$$n = N_L + \frac{\sum_{h=1}^{L-1} N_h^2 S_{yh}^2 \,/\, (a_h N^2)}{\bar{Y}^2 c^2 + \sum_{h=1}^{L-1} N_h S_{yh}^2 \,/\, N^2}. \qquad (1)$$

In this expression one can write $n = n(b_1,\ldots,b_L)$ to highlight that the value of n depends on the stratum boundaries. The `strata.LH` function tries to find the optimal boundaries b_h that minimize $n(b_1, \ldots, b_{L-1})$. Two minimization algorithms are available, either Sethi's (1963) algorithm as implemented by Lavallée and Hidiroglou (1988) with `algo="Sethi"` or Kozak's (2004) random search algorithm with `algo="Kozak"`. The latter is the default option. This section assumes $Y = X$; it does not distinguish the stratification from the survey variable.

3.1 Sethi (1963) example with the normal distribution

A classical problem is to determine the optimal boundaries for L strata in an infinite population from a known distribution. For instance, Sethi (1963) derived the optimal bounds for the normal and the χ^2_{30} distributions. To obtain approximate solutions, one can run the `strata.LH` function on a large Monte Carlo population simulated from the known distribution, without requesting a take-all stratum. In (1), one has $N_h / N^2 \approx 0$ and the optimal boundaries are the same for any target CV c.

The following R-code simulates populations of size 10^5 from the $N(10, 1)$ and the χ^2_{30} distributions. Observe that *stratification* requires the stratification variable to be non negative, so that it would not work on standard normal deviates. By subtracting 10 from the $N(10, 1)$ boundaries, we get the ones for a $N(0, 1)$. The calculations are done with the `strata.LH` function with the argument `algo="Sethi"` and with `takeall=0`, so that a take-all stratum is not requested.

```
> z <- rnorm(100000, 10)
> zl5 <- strata.LH(x = z, CV = 0.001, Ls = 5,
  alloc = c(0.5, 0, 0.5), takeall = 0, algo = "Sethi")
> zl5$bh - 10

[1] -1.1247340 -0.3480829 0.3297044 1.0979017

> x30 <- rchisq(100000, 30)
> xl5 <- strata.LH(x = x30, CV = 0.01, Ls = 5,
  alloc = c(0.5, 0, 0.5), takeall = 0, algo = "Sethi")
> xl5$bh

[1] 22.82148 28.12303 33.38642 40.20165
```

In Table 5, the agreement between the true bounds reported in Table 8 of Sethi (1963) and the Monte Carlo bounds is quite good. This approach could be used to calculate the optimal stratum boundaries for an arbitrary distribution, see for instance Khan, Nand, and Ahmad (2008).

Table 5
Comparison of Sethi's (1963) optimal stratum boundaries and of the approximate boundaries obtained with *stratification*

		***stratification*'s results**				**Sethi's results**			
	L	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
	2	-0.007				0.00			
b_h	3	-0.531	0.567			-0.55	0.55		
$N(0,1)$	4	-0.883	-0.008	0.864		-0.88	0.00	0.88	
	5	-1.125	-0.348	0.330	1.098	-1.11	-0.34	0.34	1.11
	2	30.674				30.6			
b_h	3	26.535	35.141			26.0	35.0		
χ^2_{30}	4	24.340	30.733	38.179		24.0	30.6	38.0	
	5	22.821	28.123	33.386	40.202	22.0	28.0	33.0	40.0

3.2 Gunning and Horgan (2004) example

In their original proposal, Lavallée and Hidiroglou (1988) always had a take-all stratum for a skewed survey

variable. To show that this was not always mandatory, Gunning and Horgan (2004) derived the optimal stratified designs featuring a take-all stratum for the four populations considered in Table 1. The findings of their Table 7 (with slight corrections due to rounding errors) is reproduced in Table 6. Comparing Tables 1 and 6, one sees that the optimal designs featuring a take-all stratum have n-values larger than 100 for three populations out of four. The optimal design is superior to the geometric design only for the Debtors population. The R-code to run Sethi's algorithm on the Debtors population is given below.

```
> pop1LH <- strata.LH(x = Debtors, CV = 0.0359, Ls = 5,
  alloc = c(0.5, 0, 0.5), takeall = 1, algo = "Sethi")
```

In Table 6, one would expect the optimal designs obtained through an iterative algorithm to have a smaller sample size than the *ad hoc* geometric designs. This fails to occur for three populations. This might be caused by a failure of Sethi's algorithm to find the true minimum value for *n*. To check this, we reran the programs to produce Table 6 with the argument `algo="Kozak"`. The sample sizes n are given in the second column of Table 7. Kozak's algorithm finds a smaller n-value that Sethi's for three of the four populations. This highlights the weakness of Sethi's algorithm for real populations. The second column of Table 7 has n values larger than 100 for two of the four populations. In these cases, the geometric design might be better because a take-all stratum is not required. To check this we reran Kozak's algorithm wihout a take-all stratum, *i.e.*, with `takeall=0`. The results are reported in the third column of Table 7. For the Debtors and the UScolleges populations, taking away the take-all stratum reduces the sample size n. Still, for the UScities population, Kozak's algorithm does worst than the geometric design. It failed to find the true minimum value of n with the default arguments that control its random search. To better understand the results of Table 7, we now present in more details the selection of initial stratum boundaries in `strata.LH` and the parameters that control the random search with `algo="Kozak"`.

Table 6
Optimal stratified designs featuring a take-all stratum obtained with Sethi's algorithm for the 4 populations of Table 1

Population	n	CV		1	2	3	4	5
Debtors	93	0.0359	b_h	349.33	1,190.16	3,482.98	10,322.50	
			N_h	1,856	991	350	146	26
			n_h	13	17	17	20	26
UScities	137	0.0145	b_h	14.72	21.62	35.59	80.47	
			N_h	189	270	336	164	79
			n_h	4	8	16	30	79
UScolleges	107	0.0183	b_h	512.32	869.76	1,577.23	3,668.85	
			N_h	133	180	185	110	69
			n_h	4	6	10	18	69
USbanks	104	0.0107	b_h	99.37	129.60	181.94	317.36	
			N_h	70	66	82	65	74
			n_h	4	4	7	15	74

Table 7
Sample size n for three optimal designs and four populations

Population	`algo=Sethi`	`algo=Kozak`	`algo=Kozak`
	`takeall=1`	`takeall=1`	`takeall=0`
Debtors	93	92	82
UScities	137	114	123
UScolleges	107	107	95
USbanks	104	88	88

3.3 Customization of the algorithms

The default initial stratum boundaries for the two iterative algorithms are the arithmetic starting point of Gunning and Horgan (2007), with $b_h = \min X + (\max X - \min X) \times h/L$, for $h = 1,\ldots,L-1$. In Table 7, this choice is questionable and the geometric stratum boundaries would have been closer to the true optimal boundaries. In `strata.LH`, the argument `initbh=` allows to specify a vector of $L-1$ initial boundary values. The maximum number of iterations can be changed with the `maxiter` element of the `algo.control` argument.

Kozak's algorithm was first proposed in Kozak (2004), see also Kozak and Verma (2006). It uses a random search that selects the $L-1$ stratum boundaries among the sorted values of X, with the duplicates discarded. At one iteration, it randomly picks a number d in the set `{-maxstep,-maxstep+1,…,maxstep}` and one of the $L-1$ boundaries. Then it moves the selected boundary by d positions in the vector of sorted X-values. If (1) is smaller with the new boundary it is kept, otherwise it is discarded and the boundaries are left unchanged at this iteration. The algorithm stops when the boundaries have not been changed for `maxstill` consecutive iterations. The default values are `maxstep=3` and `maxstill=100`. Two consecutive runs of Kozak's algorithm might lead to different designs because of the random nature of this algorithm. The `strata.LH` runs the algorithm `rep` times and the information for each run is contained in the `rep.detail` element of `R`-objects of class strata; the default value is `rep=3`. If the `rep` runs lead to different designs, then the tuning parameters of the algorithm can be changed. One can also use use `rep="change"` which runs the algorithm 27 times with different starting and `maxstep` values. An additional example illustrating an instance where Kozak's algorithm does not reach a global minimum is presented in the Appendix.

With N_u unique X-values, there are approximately $\binom{N_u-1}{L-1}$ possible sets of stratum boundaries. If this number is smaller than `minsol` all the possible sets of strata are tried, rather than carrying out a random search. The default value is `minsol=1000`. The elements `maxstep`, `maxstill`, `minsol` and `rep` belong to the `algo.control` argument. In Table 7, we were unable to improve the geometric stratified design for the UScities population. The command to run

Kozak's algorithm 27 times with various tuning parameters is given below.

```
> data(UScities)
> pop2LHrep <- strata.LH(x = UScities, CV = 0.0145, Ls = 5,
  alloc = c(0.5, 0, 0.5), takeall = 0, algo = "Kozak",
  algo.control = list(rep = "change"))
```

This command takes a few seconds to run and yields a stratifed design with $n = 100$, similar to that presented in Table 1 for the UScities.

3.4 Designs with a predetermined sample size *n*

With Kozak's algorithm it is possible to find the boundaries that minimize the CV of $\bar{y}_s$ for a fixed sample size n rather than minimizing n for a predetermined CV. As an example we revisit the stratified designs of Table 1. The geometric boundaries are used as initial values and the default Kozak algorithm is run. The R-code for the Debtors population is given below.

```
> pop1k <- strata.LH(x = Debtors, initbh = pop1$bh, n = 100,
  Ls = 5, alloc = c(0.5, 0, 0.5), algo = "Kozak")
```

The CVs of the estimator of $\bar{y}_s$ obtained with the optimal stratified designs are 3.12%, 1.43%, 1.72%, and 1.04% for the four populations as compared with 3.59%, 1.45%, 1.83%, and 1.07% in Table 1. Thus the iterative algorithm allowed to reduce the CVs.

4. Stratification with anticipated moments

A difference between the stratification variable X and the survey variable Y can be accounted for by having a model for the conditional distribution of Y given X. In *stratification*, there is a log-linear model where

$$Y = \exp(\alpha)X^{\beta}\exp(\sigma\epsilon),$$

and an heteroscedastic linear model with

$$Y = \alpha + \beta X + \sigma\epsilon X^{\gamma}, \qquad (2)$$

and α, β, and γ are real parameters specified by the user and ϵ is a $N(0, 1)$ random variable. A random replacement model (Rivest 1999) is also available and stratum specific mortality rates (Baillargeon, Rivest and Ferland 2007) can be added to the log-linear model.

Under these models, the anticipated mean of Y for the units classified in stratum h, with $X \in [b_{h-1}, b_h)$ are

$$\bar{Y}_h = \frac{1}{N_h}\sum_{b_{h-1}\le X_i<b_h} E(Y \mid X_i)$$

while the anticipated variance is

$$S^2_{yh} = \frac{1}{N_h}\sum_{b_{h-1}\le X_i<b_h}\{E(Y \mid X_i) - \bar{E}(Y \mid X)_h\}^2 + \frac{1}{N_h}\sum_{b_{h-1}\le X_i<b_h}\mathrm{Var}(Y \mid X_i)$$

where $\bar{E}(Y \mid X)_h$ is the average of the predicted values of Y for the units in stratum h. In `strat.cumrootf`, `strata.geo` and `strata.bh` these expressions are used to evaluate the sampling properties of $\bar{y}_s$ while in `strata.LH`, the minimization of (1) is carried out with anticipated moments. In `strata.LH` the stratum boundaries depend on the model for the relationship between X and Y; they do not for the other `strata` functions.

4.1 An example with the *MU*284 Swedish municipalities

In Section 2.5 two stratified sampling plans were derived for the *MU*284 population with *REV*84 as stratification variable. The R-code that follows investigates the performance of these sampling designs for the variable *RMT*85. The vector `ord` contains the position of the order statistics of the *REV*84 variable; thus `Y[ord]` is the vector of the *RMT*85 variable, ordered by increasing *REV*84 -value.

```
> data(Sweden)
> X <- Sweden$REV84
> Y <- Sweden$RMT85
> ord <- order(X)
> geo_rmt <- var.strata(geo_cer, y = Y[ord])
> cum_rmt <- var.strata(cum_cer, y = Y[ord])
> c(geo_rmt$RRMSE,cum_rmt$RRMSE)

[1] 0.06889558 0.07368794
```

In section 2.4, the CVs of the estimator $\bar{y}_s$ for the stratification variable *REV*84 were less than 5% for the cum$\sqrt{f}$ and the geometric designs. When estimating the mean of *RMT*85, the CVs are larger than 6%. This emphasizes that calculating sample sizes with a stratification variable underestimate the n needed to reach the target CV for a different survey variable. These results are reported in the first two designs of Table 8. Table 8 also shows the optimal design calculated by applying Kozak's algorithm to the *REV*84 variable, assuming $Y = X$.

Following Rivest (2002), a log-linear model is fitted for the relationship between the two variables. As shown in Figure 2, there are outliers and the following R-code estimates the parameters of the log-linear model by discarding the municipalities with extreme X / Y quantiles. The 18 discarded municipalities are represented by a star in Figure 2. The R-code for fitting the model to the non outliers follows.

```
> keep <- (X/Y > quantile(X/Y, 0.03))&(X/Y < quantile(X/Y, 0.97))
> reg <- lm(log(Y)[keep] ~ log(X)[keep])
> coef(reg)

(Intercept) log(X)[keep]
-3.153025   1.058355

> summary(reg)$sigma

[1] 0.25677
```

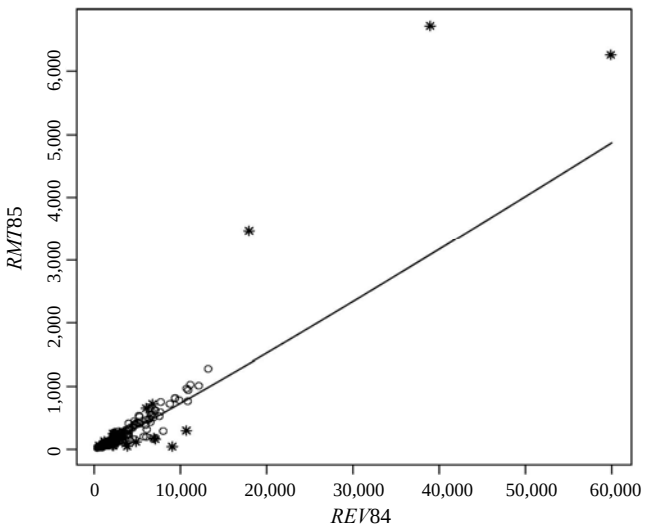

Figure 2 Plot of *RMT*85 by *REV*84 from the data set Sweden

The following code stratifies the *MU*284 population on *REV*84 using the cum $\sqrt{f}$ and the geometric method. The allocation is however carried out with anticipated moments calculated with the log-linear regression model of *RMT*85 on *REV*84. The strata of these two designs are the same as those calculated earlier. The model affects only the anticipated CV. It is not so for the optimal design where the anticipated moments are used in the stratification algorithm. Kozak's algorithm might fail to find the global minimum n value when using anticipated moments; thus we use the bounds calculated with $Y = X$ as starting values.

```
> geo_cer.m <- strata.geo(x = X[ord], CV = 0.05, Ls = 4,
  alloc = c(0.35, 0.35, 0), model = "loglinear",
  certain = (length(X) - 2):length(X), model.control =
  list(beta = 1.058355, sig2 = 0.25677^2))
> geo_cer.var <- var.strata(geo_cer.m, y = Y[ord])
> cum_cer.m <- strata.cumrootf(x = X[ord], nclass = 50,
  CV = 0.05, Ls = 4, alloc = c(0.35, 0.35, 0),
  certain = (length(X) - 2):length(X), model = "loglinear",
  model.control = list(beta = 1.058355, sig2 = 0.25677^2))
> cum_cer.var <- var.strata(cum_cer.m, y = Y[ord])
> LH <- strata.LH(x = X, CV = 0.05, Ls = 5,
  alloc = c(0.35, 0.35, 0), takeall = 1)
> LH.var <- var.strata(LH, y = Y)
> LH_m <- strata.LH(x = X, CV = 0.05, Ls = 5,
  initbh = LH$bh, alloc = c(0.35, 0.35, 0), takeall = 1,
  model = "loglinear", model.control = list(beta = 1.058355,
  sig2 = 0.25677^2))
> LH_m.var <- var.strata(LH_m, y = Y)
```

In Table 8, sample sizes calculated with anticipated moments give CVs smaller than 5% for estimating the mean *RMT*85 variable. The optimal LH design requires a n slightly smaller than the other two. Accounting for $Y \neq X$ when minimizing (1) gives a larger take-all stratum since its size increased from 4 to 5 when using the anticipated moments.

Finally observe that the arguments `model` and `model.control` can be used with `var.strata`. For the geometric design considered in this section, one can get results very similar to those obtained with the argument `y=Y`. As shown below, the model yields a CV of 6.894% as compared with 6.890% obtained with the original *RMT*85-variable. For the cum $\sqrt{f}$ method the model CV is 7.282% as compared to 7.369% found earlier while for the Lavallée Hidiroglou algorithm these two values are 7.080% and 7.110%.

```
> geo_rmt2 <- var.strata(geo_cer, model = "loglinear",
  model.control = list(beta = 1.058355, sig2 = 0.25677^2))
> geo_rmt2$RRMSE

[1] 0.0689368
```

Table 8
Three stratified designs for estimating the mean *RMT*85 with *REV*84 as the stratification variable

Model	Method		1	2	3	4	5	*n* anticip.	CV
$Y = X$	cum $\sqrt{f}$	N_h	127	79	46	29	3		
		n_h	3	4	4	5	3	19	7.37
	geometric	N_h	42	116	88	35	3		
		n_h	2	5	7	7	3	24	6.89
	LH	N_h	120	82	45	33	4		
		n_h	3	4	4	5	4	20	7.11
loglinear	cum $\sqrt{f}$	N_h	127	79	46	29	3		
		n_h	6	8	9	10	3	36	4.78
	geometric	N_h	42	116	88	35	3		
		n_h	3	8	13	13	3	40	4.74
	LH	N_h	121	81	45	32	5		
		n_h	6	7	7	9	5	34	4.90

4.2 Anderson, Kish and Cornell (1976) example with the bivariate normal distribution

Anderson *et al.* (1976) investigated the optimal stratification for Y based on X when (X, Y) has a bivariate normal distribution with correlation ρ. Thus model (2) holds with $\alpha = \gamma = 0$, $\beta = \rho$, and $\sigma^2 = 1 - \rho^2$ where X has a $N(0, 1)$ distribution. To reproduce Anderson *et al.* (1976) results, we generate a population of size $N = 10^5$ from a $N(0, 1)$ distribution and select `model="linear"` (as in Section 3.1 a mean of 10 was used to prevent X from being negative). For a linear model, only Kozak's algorithm works. Given the special nature of the problem, the `maxstep` parameter is set to 20 and only one repetition (`rep=1`) of the algorithm is run. When there is no take-all stratum, the optimal stratum boundaries are independent of the CV, as in Section 3.1. We used CV = 0.01 in the calculations.

```
> x <- rnorm(1e+05, 10)
> bi3a <- strata.LH(x = x, CV = 0.01, Ls = 3, takenone = 0,
  model = "linear",
  model.control = list(beta = 0.25, sig2 = 1 - 0.25^2,
  gamma = 0), algo.control = list(maxstep = 20, rep = 1))
> bi3a$bh - 10

[1] -0.619354 0.604198
```

In Table 9, *stratification*'s results are equal to Anderson's *et al.* (1976) findings up to nearly two decimals. This highlights the flexible nature of the package; it can find the optimal stratified design for any distribution of the stratification variable and for some general models for the conditional distribution of Y given X.

Table 9
Comparison of Anderson *et al.* (1976) optimal stratum boundaries with the approximate boundaries obtained with *stratification*

		***stratification*'s results**				**Anderson *et al.*'s results**			
L	**|ρ|**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
3	0.250	-0.619	0.604			-0.61	0.61		
	0.950	-0.591	0.568			-0.58	0.58		
	0.990	-0.571	0.549			-0.56	0.56		
4	0.250	-0.984	0.004	0.985		-0.98	0.00	0.98	
	0.950	-0.930	0.009	0.942		-0.93	0.00	0.93	
	0.990	-0.902	-0.001	0.895		-0.90	0.00	0.90	
5	0.250	-1.245	-0.377	0.387	1.251	-1.24	-0.38	0.38	1.24
	0.950	-1.187	-0.358	0.372	1.197	-1.19	-0.37	0.37	1.19
	0.990	-1.136	-0.344	0.353	1.144	-1.14	-0.35	0.35	1.14

5. Additional features

Baillargeon and Rivest (2009) considered additional aspects of a stratified design, namely stratum specific anticipated non-response rates and the addition of a take-none stratum with a null sample size. This section discusses briefly how these additional items are handled in *stratification*. Non-response needs to be accounted for when optimizing for n. A take-none stratum makes $\overline{y}_s$ biased; in this case the precision target is specified in terms of a Relative Root Mean Squared Error (RRMSE) rather than a CV. Formula (4.3) of Baillargeon and Rivest (2009) provides a generalization of (1) that includes these two features. This is the formula used for calculating sample sizes in the optimization procedure.

5.1 Non-response

Non-response can be corrected *a posteriori*, by dividing the no non-response stratum sample sizes by the response rates. This is illustrated in the following R-code that considers the MRTS variable, representative of Statistics Canada Monthly Retail Trade Survey. *Post hoc* non-response corrections are implemented in the `var.strata` function with the argument `rh.postcorr=TRUE`. An alternative is to consider response rates when allocating the sample to the strata. They can be specified in a `strata` function with the argument `rh=`. This approach penalizes strata with a high non-response; it typically yields a smaller n value than the *a posteriori* corrections. This is illustrated in the cum $\sqrt{f}$ portion of Table 10. With four strata and response rates of 0.8, 0.8, 0.9, 1, the *a posteriori* correction needs $n = 445$ to reach the target CV for the MRTS variable, as compared with $n = 444$ for an allocation that takes non-response into account.

```
> data(MRTS)
> cum <- strata.cumrootf(x = MRTS, nclass = 500, CV = 0.01,
  Ls = 4, alloc = c(0.5, 0, 0.5))
> cum.var <- var.strata(cum, rh = c(0.8, 0.8, 0.9, 1))
> cum.post <- var.strata(cum, rh = c(0.8, 0.8, 0.9, 1),
  rh.postcorr = TRUE)
> cum_rh <- strata.cumrootf(x = MRTS, nclass = 500, CV = 0.01,
  Ls = 4, alloc = c(0.5, 0, 0.5), rh = c(0.8, 0.8, 0.9, 1))
```

Non-response can also be accounted for when constructing an optimal sampling design, either *a posteriori* or in the stratum construction. These two approaches are implemented for the MRTS population in the following R-code. The higher non-response rates for the small units penalize the first stratum which is smaller when non-response is accounted for in the stratification algorithm, as can be seen in Table 10. Still accounting for non-response in the stratum construction gives a smaller n-value than an *a posteriori* correction. Table 3 of Baillargeon and Rivest (2009) presents additional examples, including both anticipated moments and non-response, of the construction of stratified designs for the MRTS population.

```
> LH <- strata.LH(x = MRTS, CV = 0.01, Ls = 4,
  alloc = c(0.5, 0, 0.5), takeall = 1)
> LH.var <- var.strata(LH, rh = c(0.8, 0.8, 0.9, 1))
> LH.post <- var.strata(LH, rh = c(0.8, 0.8, 0.9, 1),
  rh.postcorr = TRUE)
> LH_rh <- strata.LH(x = MRTS, CV = 0.01, Ls = 4,
  alloc = c(0.5, 0, 0.5), takeall = 1, rh = c(0.8, 0.8, 0.9, 1))
```

Table 10
Two examples of non-response correction: Either *a posteriori* (post) or when constructing the design

Method	**rh**		**1**	**2**	**3**	**4**	***n***	**anticip. CV**
cum $\sqrt{f}$	none	N_h	778	742	355	125		
		n_h	87	90	88	125	390	1.11
		n_h^{post}	109	113	98	125	445	1.00
	given	N_h	778	742	355	125		
		n_h	105	108	106	125	444	1.00
LH	none	N_h	774	675	374	177		
		n_h	77	65	60	177	379	1.11
		n_h^{post}	96	81	67	177	421	1.00
	given	N_h	675	677	449	199		
		n_h	70	69	80	199	418	1.00

5.2 Take-none stratum

A take-none stratum with a null sample size might be advantageous when the population has small units with Y-values close to 0. The precision of $\overline{y}_s$ is then measured by the mean squared error, $\mathrm{Var}(\overline{y}_s) + (T_{0y}/N)^2$, where T_{0y} is

the anticipated Y-total in the take-none stratum. Setting `takenone=1` in the `strata.LH` function constructs an optimal design with a take-none stratum. Baillargeon and Rivest (2009) showed that Sethi's algorithm does not work in this case and that Kozak's algorithm should be used. When a take-none stratum is used, a rough bias correction can be implemented by dividing $\overline{y}_s$ by the proportion of the total of the X variable in the take some strata. Thus the bias penalty in the mean square error might be too stringent and an alternative measure of precision, such as $\text{Var}(\overline{y}_s) + (p \times T_{0y}/N)^2$, could be used in the stratification algorithm where p is a number in $(0, 1)$. This smaller bias penalty can be implemented by setting the argument `bias.penalty` equal to p. The following R-code constructs three optimal stratified designs for the MRTS population, with and without a take-none stratum; the default full bias penalty is compared to a reduced penalty with $p = 0.5$.

```
> data(MRTS)
> notn <- strata.LH(x = MRTS, CV = 0.1, Ls = 3,
  alloc = c(0.5, 0, 0.5))
> tn1 <- strata.LH(x = MRTS, CV = 0.1, Ls = 3,
  alloc = c(0.5, 0, 0.5), takenone = 1)
> tn0.5 <- strata.LH(x = MRTS, CV = 0.1, Ls = 3,
  alloc = c(0.5, 0, 0.5), takenone = 1, bias.penalty = 0.5)
```

The sample sizes n for the three designs are given in Table 11. Including a take-none stratum with a full bias penalty reduces n, from 22 to 16; for this design the take-none stratum accounts for 3% of the total of the X-variable. Reducing the biais penalty to $p = 0.5$ increases the size of the take-none stratum and reduces n. Additional illustrations are given in Table 2 of Baillargeon and Rivest (2009). They show that the size of a take-none stratum typically decreases with the target RRMSE. For the MRTS example, the addition of a take-none stratum diminishes the n-value substantially while for others it does not change the design.

Table 11
Sample sizes for three optimal stratified designs for the MRTS population

takenone	0	1	1
bias.penalty	NA	1	0.5
n	22	16	13
% T_x	0	3	9

6. Conclusion

The R-package *stratification* offers flexible methods for the construction of a stratified sampling design using a univariate stratification variable such as a measure of size in a business survey. Several methods are available to determine the stratum boundaries and the stratum sample sizes. *stratification* allows the investigation of features such as a take-all stratum, a take-none stratum, the extent of the discrepancy between X and Y, and a stratum specific non-response.

Acknowledgements

We are grateful to S. Er, E. Gagnon, M. Kozak, and J. Stardom for constructive comments on the package and to the Canada Research Chair on Statistical Sampling and Data Analysis and the Natural Sciences and Engineering Research Council of Canada for their financial support. This research was supported by U.S. National Science Foundation grant SES-0751671.

7. Appendix

7.1 More details on Kozak's algorithm

As described in Section 3.3 Kozak's algorithm uses a random search. Besides decreasing the optimization criterion, either the n-value or the *RRMSE* of $\overline{y}_s$, *stratification* requires that the take-some strata contain at least `minNh` units and that they have positive sample sizes, for the new boundary to be admissible. The default is `minNh=2`. A non random, Kozak's algorithm is also available with `method="modified"` in the `algo.control` argument. It tries all the possible changes at one iteration and picks the one that gives the largest drop of the optimization criterion. It is slower than Kozak's algorithm without improving the detection of the global minimum of the optimization criterion. Therefore, it will not be discussed any further.

To illustrate the complete enumeration of all possible solutions mentioned in Section 3.3, consider the `USbanks` data set. It contains 357 values, but only 200 unique values. If one wishes to stratify this population in two strata, only $\binom{200-1}{2-1} = 199$ solutions are possible. The following command performs a complete enumeration of the possible solutions:

```
> enum <- strata.LH(x = USbanks, CV = 0.05, Ls = 2,
  alloc = c(0.5, 0, 0.5))
```

These solutions, with their associated optimization criteria value, can be found in `enum$sol.detail`. Only the solutions fulfilling the admissibility constraints mentioned above are included in `enum$sol.detail`.

When running Kozak's algorithm, the initial boundary values might fail to meet the admissibility constraints; the algorithm might not be able to move at all. In such a case, the initial boundaries are replaced by robust ones. The robust boundaries give an empty take-none stratum if such a stratum is requested, take-all strata as small as possible, and take-some strata with approximately the same number of unique X-values.

Consider once again the example of Section 3.2 with the `UScities` data set, where Kozak's algorithm reached a local minimum with the default arguments. With geometric initial boundaries, Kozak's algorithm converges rapidly to what appears to be a global minimum.

```
> LH_init <- strata.LH(x = UScities, initbh = pop2$bh,
  n = 100, Ls = 5, alloc = c(0.5, 0, 0.5), takeall = 0,
  algo.control = list(rep = 1))
> LH_init$iter.detail

      b1    b2    b3   b4         opti   step   iter   run
1   18.5  33.5  59.5  107   0.01444981      0      0     1
2   20.5  33.5  59.5  107   0.01435576      2      2     1
3   19.5  33.5  59.5  107   0.01434272     -1     10     1
4   19.5  33.5  58.0  107   0.01432714     -1     12     1
5   19.5  31.5  58.0  107   0.01431013     -2     13     1
6   19.5  32.5  58.0  107   0.01430163      1     63     1

> LH_init$niter
[1] 163
```

The output element `LH_init$iter.detail` contains information about the initial boundaries and the 5 iterations with a change of boundaries only. A total of 163 iterations were needed for the algorithm to converge. The geometric initial boundaries are very close to the optimal solutions. A local minimum can also be avoided by changing some of the algorithm's parameters. The following R-code allows larger steps (`maxstep=20`) and increases the maximal number of iterations (`maxstill=1000`) and the number of repetitions of the algorithm (`rep=20`).

```
> LH_param <- strata.LH(x = UScities, n = 100, Ls = 5,
  alloc = c(0.5, 0, 0.5), takeall = 0, algo.control =
  list(maxstep = 20, maxstill = 1000, rep = 20))
```

The results for the 20 repetitions are reported in `LH_param$rep.detail` and summarized in Table 12. The solution obtained with the geometric initial boundaries is reached 9 times out of 20.

Table 12
Solutions found by Kozak's algorithm for 20 repetitions

CV	B1	B2	B3	B4	frequency
0.0143	19.50	32.50	58.00	107.00	9
0.0167	16.50	23.50	37.50	78.00	5
0.0167	15.50	22.50	35.50	73.00	6

Figure 3 shows how larger steps help the algorithm to reach the global minimum (CV = 0.0143), compared to a run of the algorithm with the default arguments (dotted lines, CV = 0.0167).

7.2 R package *stratification* summary table

This appendix provides a quick reference for the `R` package *stratification*. Table 13 lists the five functions in *stratification* and their arguments. The following notes complete the table.

(1) According to the general allocation scheme (Hidiroglou and Srinath 1993). The stratum sample sizes are proportional to $N_h^{2q_1}\overline{Y}_h^{2q_2}S_{yh}^{2q_3}$.

(2) The default value of `initbh` is the set of arithmetic starting points of Gunning and Horgan (2007), see Section 3.3. If `takenone=1` and `initbh` is of size `Ls-1`, the initial boundary of the take-none stratum is set to the first percentile of `X`. If this first percentile is equal to the minimum value of `X`, this initial boundary would lead to an empty take-none stratum. In that case, the initial boundary of the take-none stratum is rather set to the second smallest value of `X`.

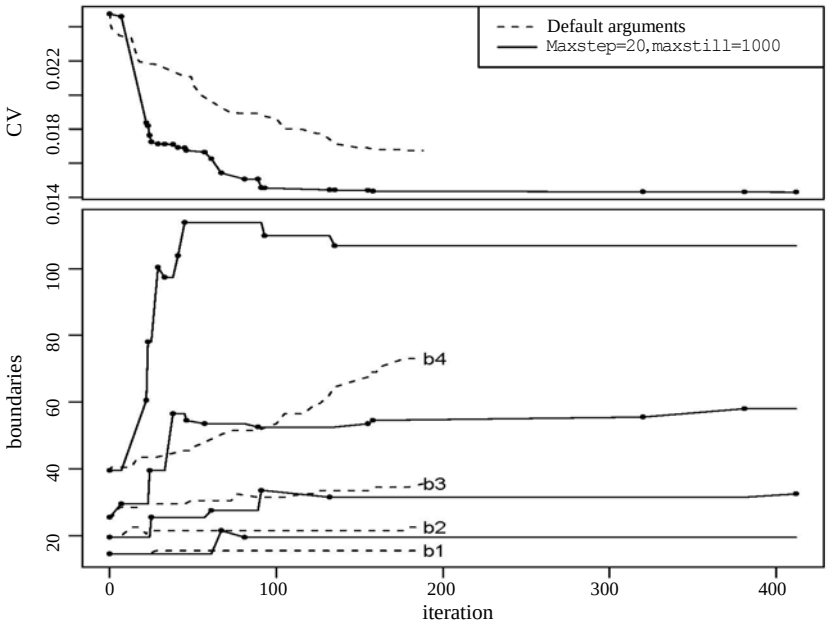

Figure 3 Iterations histories for two runs of Kozak's algorithm

(3) The elements to specify in the `algo.control` argument depend on the algorithm. The following table shows the elements used by each algorithm and their default values. See `help(strata.LH)` for a complete description of every element.

Algorithm	maxiter	method	minNh	maxstep	maxstill	rep	minsol
Sethi	500	-	-	-	-	-	-
Original Kozak	10,000	"original"	2	3	100	3	1,000
Modified Kozak	3,000	"modified"	2	3	-	-	1,000

(4) The elements of the `model.control` argument depend on the model:

- `loglinear` model with mortality:

$$Y = \begin{cases} \exp(\alpha + \texttt{beta}\,\log(X) + \texttt{epsilon}) & \\ \qquad\qquad \text{with probability } p_h & \\ 0 \qquad\qquad \text{with probability } 1-p_h & \end{cases}$$

where $\texttt{epsilon} \sim N(0, \texttt{sig2})$ is independent of X. The parameter p_h is specified through `ph`, `ptakenone` and `pcertain`.

- heteroscedastic `linear` model :

$$Y = \texttt{beta}\,X + \texttt{epsilon}$$

where

$$\texttt{epsilon} \sim N(0, \texttt{sig2}\,X^{\texttt{gamma}}).$$

- `random` replacement model:

$$Y = \begin{cases} X & \text{with probability } 1 - \texttt{epsilon} \\ X\text{new} & \text{with probability } \texttt{epsilon} \end{cases}$$

where Xnew is a random variable independent of X with the same distribution as X.

The following table presents `model.control` default values according to the model.

model	beta	sig2	ph	ptakenone	pcertain	gamma	epsilon
"loglinear"	1	0	rep(1,Ls)	1	1	-	-
"linear"	1	0	-	-	-	0	-
"random"	-	-	-	-	-	-	0

Table 13
R package *stratification* summary table

argument	Strata.cumrootf	Strata.geo	Strata.LH	Strata.bh	Var.strata	description	format	default
x	•	•	•	•		stratification variable	vector	none (x is mandatory)
n	•	•	•	•		target total sample size	scalar	none (n or CV is mandatory)
CV	•	•	•	•		target CV or RRMSE	scalar	none (n or CV is mandatory)
Ls	•	•	•	•		number of sampled strata	scalar	3
alloc	•	•	•	•		allocation specification (1)	list (q1,q2,q3) where qi≥ 0	Neyman (q1=q3=0.5, q2=0)
certain	•	•	•	•		x–indices for units sampled with certainty	vector	NULL (no certainty stratum)
nclass	•					number of bins	scalar	*min*(10*L*, *N*)
bh				•		strata boundaries	vector	none (bh is mandatory)
takeall.adjust				•		indicator of adjustment for take-all strata	True or False	FALSE (no adjustment)
takeall			•	•		number of take-all strata	one of {0, 1,..., Ls − 1}	0
initbh			•			initial strata boundaries (2)	vector	equidistant boundaries
algo			•			algorithm identification	"Kozak" or "Sethi"	"Kozak"
algo.control			•			algorithm's parameters specification (3)	list (maxiter, method, minNh, maxstep, maxstill, rep)	depends on algo
strata					•	stratification scheme	strata object	none (strata is mandatory)
y					•	study variable	vector	NULL (model given instead)
model	•	•	•	•	•	model identification	"none", "loglinear", "linear"* or "random"* →	"none" (*unavailable with Sethi's algo)
model.control	•	•	•	•	•	model's parameter specification (4)	list (beta, sig2, ph, ptakenone, gamma, epsilon)	depends on model, but equivalent to model="none"
rh	•	•	•	•	•	anticipated response rates	scalar or vector	rep(1,Ls) or rh from strata
rh.postcorr					•	indicator of posterior correction for non-response	TRUE or FALSE	FALSE (no correction)
takenone			•	•		number of take-none strata	0 or 1	0
bias.penalty			•	•		penalty for the bias	scalar	1

References

Anderson, D.W., Kish, L. and Cornell, R.G. (1976). Quantifying gains from stratification for optimum and approximately optimum strata using a bivariate normal model. *Journal of the American Statistical Association*, 71, 887-892.

Baillargeon, S., Rivest, L.-P. and Ferland, M. (2007). Stratification en enquêtes entreprises : une revue et quelques avancées. *Proceedings of the Survey Methods Section, Statistical Society of Canada* (www.ssc.ca/survey/documents/SSC2007_S_Baillargeon.pdf).

Baillargeon, S., and Rivest, L.-P. (2009). A general algorithm for univariate stratification. *International Statistical Review*, 77, 331-344.

Cochran, W.G. (1961). Comparison of methods for determining stratum boundaries. *Bulletin of the International Statistical Institute*, 32, 345-358.

Cochran, W.G. (1977). *Sampling Techniques. Third Edition*. New York: John Wiley & Sons, Inc.

Dalenius, T., and Hodges, J.L., Jr. (1959). Minimum variance stratification. *Journal of the American Statistical Association*, 54, 88-101.

Dayal, S. (1985). Allocation of sample using values of auxiliary characteristics. *Journal of Statistical Planning and Inference*, 11, 321-328.

Detlefsen, R.E., and Veum, C.S. (1991). Design issues for the Retail Trade Sample Surveys of the U.S. Bureau of the Census. *Proceedings of the Survey Research Methods Section*, American Statistical Association, 214-219.

Gunning, P., and Horgan, J.M. (2004). A new algorithm for the construction of stratum boundaries in skewed populations. *Survey Methodology*, 30, 159-166.

Gunning, P., and Horgan, J.M. (2007). Improving the Lavallée and Hidiroglou algorithm for stratification of skewed populations. *Journal of Statistical Computation and Simulation*, 77, 277-291.

Hidiroglou, M.A. (1986). The construction of a self-representing stratum of large units in survey design. *The American Statistician*, 40, 27-31.

Hidiroglou, M.A., and Srinath, K.P. (1993). Problems associated with designing subannual business surveys. *Journal of Business and Economic Statistics*, 11, 397-405.

Khan, M.G.M., Nand, N. and Ahmad, N. (2008). Determining the optimum strata boundary points using dynamic programming. *Survey Methodology*, 34, 205-214.

Kozak, M. (2004). Optimal stratification using random search method in agricultural surveys. *Statistics in Transition*, 6, 797-806.

Kozak, M., and Verma, M.R. (2006). Geometric versus optimization approach to stratification: A comparison of efficiency. *Survey Methodology*, 32, 157-163.

Lavallée, P., and Hidiroglou, M. (1988). On the stratification of skewed populations. *Survey Methodology*, 14, 33-43.

McEvoy, R.H. (1956). Variation in bank asset portfolios. *The Journal of Finance*, 11(4), 463-473.

Rivest, L.-P. (1999). Stratum jumpers: Can we avoid them?. *ASA Proceedings of the Section on Survey Research Methods*, American Statistical Association, (Alexandria, VA), 64-72.

Rivest, L.-P. (2002). A generalization of the Lavallée and Hidiroglou algorithm for stratification in business surveys. *Survey Methodology*, 28, 191-198.

Särndal, C.E., Swensson, B. and Wretman, J. (1992). *Model Assisted Survey Sampling*. New York: Springer Verlag.

Sethi, V.K. (1963). A note on the optimum stratification of populations for estimating the population means. *Australian Journal of Statistics*, 5, 20-33.

Sigman, R.S., and Monsour, N.J. (1995). Selecting samples from list frames of businesses. In *Business Survey Methods*, (Eds., B.G. Cox, D.A. Binder, B.N. Chinnappa, A. Christianson, M.L. Colledge and P.S. Kott), 133-152.

Slanta, J., and Krenzke, T. (1996). Applying the Lavallée and Hidiroglou method to obtain stratification boundaries for the Census Bureau's Annual Capital Expenditure Survey. *Survey Methodology*, 22, 65-75.

Sweet, E.M., and Sigman R.S. (1995). Evaluation of model-assisted procedures for stratifying skewed populations using auxiliary data. *U.S. Bureau of the Census* (www.census.gov/srd/papers/pdf/sm95-22.pdf).