Marcin Kozak

University of Information Technology and Management in Rzeszow

ON EQUAL-PRECISION STRATIFICATION IN DOMAINS SUBJECT TO FIXED SAMPLE SIZE

Introduction

Stratified sampling is one of the most common sampling designs in surveys in economic official statistics, such as those conducted by the Central Statistical Office of Poland. Independent sampling in domains is of special interest for practical reasons; for instance, in Polish economic surveys voivodeships constitute the domains, and in many surveys estimation is required for both the whole country and each voivodeship. This paper deals with optimizing stratification when estimation for domains is of interest in addition to estimation for the whole population.

Consider a population U subdivided into D non-overlapping domains U_d,

$$U = \bigcup_{d=1}^{D} U_d .$$

We aim to stratify each domain U_d, $d = 1, \dots , D$; the constraints for stratification are related to the whole population, namely sample size n from the whole population is fixed and equal to

$$n = \sum_{d=1}^{D} n_d ,$$

where n_d is the sample size from the dth domain. The quantities n_d, $d = 1, \dots , D$, which are to be determined, depend on stratification points provided for the corresponding domains. Stratification is to be done with the aim of optimizing a common value of precision of estimation (that is, minimizing the variance or, equivalently, the coefficient of variation of the estimator of a population characteristic considered) in the domains. The problem of fixed-precision stratification of a population subdivided into domains was considered by Lednicki and Wieczorkowski (2003); however, they studied stratification orientated towards minimizing a final sample size from a population subject to fixed precision of estimation in the domains of the population. Kozak and Zieliński (2005) presented formulas for sample allocation between domains and strata in the domains under fixed-precision approach; they did not study, however, stratification in such a situation.

Following the approach the most common from the practical point of view, we will consider constructing a so-called take-all stratum (Hidiroglou, 1986; Lavallée and Hidiroglou, 1988) in each domain; all the elements from the take-all stratum are taken to the sample with probability 1. The take-all stratum approach is effective for a positively skewed stratification variable, which is the case in many surveys (e.g. Hidiroglou, 1986; Lavallée and Hidiroglou, 1988; Lednicki and Wieczorkowski, 2003). Let us also assume that a stratification variable is equal to the survey variable that we aim to study, an often assumption in the stratification theory and practice.

We will use the following notation. H_d is the number of strata to be constructed in the dth domain; N_d is the size of the dth domain; N_{dh} is the size of the hth stratum in the dth domain; $W_{dh} = N_{dh} / \sum_{h=1}^{H_d} N_{dh}$; and S_{dh}^2 is the population variance of the variable X restricted to the hth stratum of the dth domain.

Under the conventional unbiased estimation and the take-all stratum approach, the coefficient of variation (for simplicity denoted as δ) of the estimator $\hat{t}_d$ of the population total of the variable X in the dth domain is equal to:

$$cv(\hat{t}_d) = \delta_d = \left(\sum_{h=1}^{H_d} \sum_{k=1}^{N_{dh}} X_{dhk} \right)^{-1} \sqrt{\sum_{h=1}^{H_d - 1} S_{dh}^2 W_{dh}^2 \left(\frac{1}{n_{dh}} - \frac{1}{N_{dh}} \right)} \qquad (1)$$

Our aim is to stratify each domain U_d, $d = 1, \dots, D$, so as to minimize the common value of δ_d, say δ, subject to the fixed sample size n from the whole population. The aim of this paper is to present and compare two algorithms for such stratification.

1. Optimum stratification

The problem under consideration can be formulated in the following way. Find D sets of strata boundaries $\mathbf{a}_d = \left(a_1, \ldots, a_{H_d - 1}\right)^T$, each referring to the corresponding dth domain, that minimize the common value δ of the coefficients of variation δ_d of the estimator considered in the domains, i.e.:

$$\delta_d = \delta \text{ for all } d = 1, \ldots, D, \tag{2}$$

subject to:

$$n = \sum_{d=1}^{D} n_d\ , \tag{3}$$

n being fixed at the outset. In other words, we aim to minimize δ subject to the constraints (2) and (3). The vector of strata boundaries $\mathbf{a}_d$ unequivocally divides the dth domain into H_d non-overlapping strata, providing lower and upper boundaries of the strata on the range from min(X) to max(X), X being the stratification variable (for details see, e.g., Lednicki and Wieczorkowski, 2003).

A general description of stratification can be found in multiplicity of papers; see e.g., Dalenius and Hodges (1959), Lavallée and Hidiroglou (1988), Rivest (2002), Lednicki and Wieczorkowski (2003), or Kozak and Verma (2006). For a particular domain, stratification we consider aims at minimizing the variance of the estimator studied (this approach is equivalent to minimizing the coefficient of variation (1)) subject to given sample size n_d. Such stratification can be treated as the optimization problem in which the formula (1) stands for the optimization function and strata boundaries are the parameters sought. For our purpose, one may apply any optimization function that starts the optimization process based on initial strata boundaries; for example, it may be the *optim* function available in R language (R Development Core Team, 2006) that uses the simplex method of Nelder and Mead (1965) (this function was used by Lednicki and Wieczorkowski, 2003), the random search method (Kozak, 2004. Kozak and Verma, 2006), recently found efficient by Baillargeon et al. (2007) and Baillargeon and Rivest (2009) and implemented in the R package *stratification* (Baillargeon and Rivest 2007), genetic algorithm (Keskintürk and Er, 2007) (Kozak, 2013 proved that random search method is a little more efficient than Keskintürk and Er's genetic algorithm) or any other similar method.

To the best of the author's knowledge, an algorithm that might be applied to the problem under consideration (i.e., minimizing δ subject to (2) and (3)) has not yet been proposed in the literature. The problem we must deal with is that we do not know the domain sample sizes n_d so we are not able to perform the stratification in the domains. This issue begs the question, how to decide about the sample sizes that are required to fulfill the constraint (2). This is not troublesome in the approach of Lednicki and Wieczorkowski (2003) since they minimize sample size n subject to given δ_d, $d = 1, \dots , D$; for this reason, they can perform stratification in each domain independently. This is, unfortunately, not the case in our situation.

If the population is small, one can stratify all the domains at the same time; then, the set of parameters searched for would comprise strata boundaries from all the domains. Otherwise, i.e., when the domains are too large to be stratified simultaneously, such an approach is unlikely to be applied. Below we present two alternative algorithms; both of them aim to minimize δ subject to (2) and (3), but the approaches to do it are different.

Algorithm 1

1. In each dth domain, construct H_d strata via any approximate stratification method (e.g. Dalenius and Hodges, 1959; Eckman, 1959; Gunning and Horgan, 2004). Since we consider the take-all stratum approach, change the last stratum boundary in such a way that the last (take-all) stratum comprises five elements.
2. Determine initial values of n_d via the formula

$$n_d = nN_d \left(\sum_{d=1}^{D} N_d \right)^{-1}, \quad d = 1, \dots , D.$$

3. Using any optimization approach to stratification (see, e.g., Lednicki and Wieczorkowski, 2003; Kozak, 2004; Kozak and Verma, 2006, and the citations in these papers), stratify each domain independently so as to minimize δ_d subject to sample size n_d. Take the approximate strata boundaries (obtained in step 1) as the initial values in the optimization. As a result, a vector $\boldsymbol{\delta} = (\delta_1, \dots , \delta_D)^T$ of the domain coefficients of variation (1) and a vector $\mathbf{n} = (n_1, \dots , n_D)^T$ of the domain sample sizes are determined.
4. Choose a domain for which δ_d is minimal; let it be the domain $d_{\min}$. Let a domain for which δ_d is maximal be $d_{\max}$. Determine

$$n_d = n_d - p \text{ if } d = d_{\min},$$
$$n_d = n_d + p \text{ if } d = d_{\max},$$
$$n_d = n_d \text{ otherwise.}$$

The value of parameter p should be chosen based on the domain sizes and sample size n assumed. (Detailed information on parameter p is provided later).

5. Perform steps 3 and 4 until at least one of the stopping criteria given below is fulfilled. In step 3, instead of the approximate strata boundaries, as initial parameters to perform the optimization take the strata boundaries obtained in a previous iteration; as the sample sizes n_d take the sample sizes obtained in a previous iteration.

The stopping criteria used in step 5 are as follows:

(i) $R = \delta_{d_{\min}} / \delta_{d_{\max}} > \varepsilon$, ε being a value fixed at the outset such that $\varepsilon < 1$;

(ii) $R_{i+1} < R_i$, R_i and R_{i+1} being the R value in the ith and $(i + 1)$th iteration, respectively; if this stopping criterion is fulfilled, the result of the ith step should be taken as the final result.

The value of parameter p from step 4 of the algorithm has an influence on a size of changes of δ_d values and on time of executing the algorithm. A large p value will likely result in noticeable differences between δ_d values obtained, whereas a small p value will make the algorithm time-consuming. On the contrary, the greater the ε value, the smaller differences between δ_d values may be obtained and the more time the algorithm needs. Therefore, the algorithm may be executed several times; in subsequent executions, the p value should be noticeably smaller and ε noticeably greater. For instance, the first p value may be equal to $0.05n$, the second, to $0.01n$, and the third, to $0.002n$; the first ε value may be then equal to 0.80, the second, to 0.90, and the third, to 0.99. (Certainly, these are just exemplary p and ε values; they should be chosen on the basis of the domain sizes and n assumed. For instance, the greater n assumed, the greater the last ε value may be [providing less differences among final δ_d]). In each new execution of the algorithm, steps 1 and 2 should be omitted; instead, the strata boundaries and domain sample sizes obtained in the previous execution should be used in the first iteration (step 3).

Worth noting is that, although for various n_d the optimum strata boundaries in the dth domain may be different, it is likely that strata boundaries will not be changed in some iteration(s) of the algorithm, especially when p is small. Using the strata boundaries obtained in a previous iteration as the initial parameters makes optimization in a particular iteration takes less time. Therefore, the first execution of the third step, that is, with approximate strata boundaries as the initial parameters in optimization, should take most time; next iterations would likely be noticeably shorter.

If the domains are very large (for instance, they comprise hundreds of thousands of elements), each domain should be saved in a different file on a computer disc. Then, in a particular iteration (besides the first execution of the third step, in which the computer opens and works on every domain separately) we need to open and work only on two files independently (i.e. once we finish stratifying one domain, we close the corresponding file and open the second one to work on the second domain considered in this iteration).

Note that we have considered estimation of the population total. The algorithm, however, works for any stratification problem, irrespective of a parameter to be estimated, an estimator to be used, as well as a stratification approach employed (provided that it works based on initial parameters). Moreover, even though we present the algorithm for univariate stratification, it applies also for multivariate stratification; the only difference is that in step 3 we would apply a multivariate stratification algorithm instead of a univariate one.

In this paper, we have used random search (Kozak, 2004; Kozak and Verma, 2006) and Nelder and Mead's (1965) optimization methods for stratification; both of them provided the same results.

Algorithm 2

This algorithm uses Lavallée and Hidiroglou's (1988) algorithm for stratification, which is orientated towards minimization of sample size subject to fixed precision of estimation. The algorithm is as follows.

1. Perform steps 1 and 2 from algorithm 1. Set $c = c_0$ as the initial value of the target CV for each domain; let n_T denote the target total sample size.
2. Using any optimization approach to stratification (see e.g. Lavallée-Hidiroglou 1988; Lednicki and Wieczorkowski, 2003; Kozak, 2004; Kozak and Verma, 2006), stratify each domain independently so as to minimize n_d subject to fixed c. Take the approximate strata boundaries (obtained in step 1) as the initial values in the optimization. As a result, a vector $\mathbf{n} = (n_1, \dots , n_D)^T$ of the domain sample sizes is determined. Thus, we have $n = \sum_{d=1}^{D} n_d$, n_d being the sample size obtained for the dth domain.
3. If $|n - n_T| < q$, stop.
4. If $|n - n_T| < |n_{\text{old}} - n_T|$, n_{old} being n from the previous iteration, stop and take results from the previous iteration as final results (this step applies only for iterations other than the first one).
5. If $n > n_T$, increase the target CV as $c_{\text{new}}=c_{\text{old}} + \varepsilon$ for some $\varepsilon >0$; if $n<n_T$, $c_{\text{new}}=c_{\text{old}} - \varepsilon$. Go back to step 2.

The parameters q from step 3 and ε from step 5 need to be commented on. If we choose too small a q value, especially when the population and the target overall sample size are large, then it is possible that the algorithm would be executed for a very long time. On the other hand, too large a q value may cause the final overall sample size n be too far from the target sample size n_T to be accepted. The ε value is directly related with the q value: the higher the ε value, the higher the q value has to be chosen, and vice versa. The ε being too small will result in unnecessarily long execution of the algorithm; on the other hand, too large the ε value might cause the algorithm not able to reach the CV that is related to the target n_T.

Step 4 is to assure that the algorithm will not follow into a loop in which in one iteration ε is decreased and in the subsequent iteration it is increased, after which it would have the same value as in the previous iteration.

Alternatively, similarly to algorithm 1, algorithm 2 may be applied gradually, where in the subsequent executions q and ε are decreased. Most of the general, non-technical comments given on algorithm 1 apply also to algorithm 2.

2. Numerical example

To present the use of the algorithms, we generated a population of size N = 100 000 subdivided into four domains of sizes 13 000, 50 000, 7000, and 30 000, respectively. The following formula was applied to generate the positively skewed stratification variable X_d in the dth domain:

$$X_d = \left[\exp\left(Z_d\right)\right],$$

where Z_d is the realization of an $N\left(10, \sigma_d^2\right)$ random variable (i.e., the random variable normally distributed with the mean 10 and standard deviation σ_d); the function [.] stands for rounding to integers (to simulate an often situation in survey practice). The standard deviations σ_d chosen were 0.4, 0.4, 0.8, and 0.6, respectively. Numbers of strata H_d constructed in the respective domains were 5, 8, 4, and 8; the sample size from the whole population was assumed to be n = 5000.

Algorithm 1 was executed three times; in the first execution, its parameters were p = 250 and ε = 0.80; in the second execution, p = 50 and ε = 0.92; finally, in the third execution, p = 10 and ε = 0.995.

The first execution needed nine iterations (Table 1), which led to the ratio R = 0.868: this stopped execution of the algorithm. In each iteration strata boundaries in both stratified domains changed. The algorithm was executed once more, with the use of the domain strata boundaries and sample sizes obtained in the previous execution and with p = 80 and ε = 0.92. The results are presented in

Table 2. This execution needed just two iterations to fulfill the first stopping rule as it led to R = 0.936 > 0.92. Again, in each iteration the strata boundaries in the stratified domains changed. Finally, the third execution needed eleven iterations (Table 3), providing final domain precisions 0.004056, 0.004063, 0.004059, and 0.004063, respectively. The ratio obtained in the last iteration was R = 0.998, which fulfilled the first stopping criterion. In most iterations of this execution one or both strata boundary sets did not changed. Altogether, algorithm 1 needed 23 iterations so a single stratification algorithm (that is, for one domain) was carried out 46 times; however, there were 36 effective iterations (the effective iteration being one in which strata boundaries changed).

Table 1

Results of the first execution of algorithm 1, i.e. for p = 250 and ε = 0.80

δ_1	δ_2	δ_3	δ_4	n_1	n_2	n_3	n_4	R
0.007610	0.002283	0.012505	0.002992	650	2500	350	1500	0.183
0.007610	0.002427	0.008488	0.002992	650	2250	600	1500	0.286
0.007610	0.002596	0.006430	0.002992	650	2000	850	1500	0.341
0.006325	0.002798	0.006430	0.002992	900	1750	850	1500	0.435
0.006325	0.003047	0.005181	0.002992	900	1500	1100	1500	0.473
0.005467	0.003047	0.005181	0.003324	1150	1500	1100	1250	0.557
0.004836	0.003365	0.005181	0.003324	1400	1250	1100	1250	0.642
0.004836	0.003365	0.004309	0.003767	1400	1250	1350	1000	0.696
0.004340	0.003795	0.004309	0.003767	1650	1000	1350	1000	0.868

Note:
The subsequent columns contain precision of estimation δ_d and sample sizes n_d corresponding to the dth domain, and ratio $R = \delta_{d_{\min}} / \delta_{d_{\max}}$.

Table 2

Results of the second execution of algorithm 1, namely for p = 50 and ε = 0.92

δ_1	δ_2	δ_3	δ_4	n_1	n_2	n_3	n_4	R
0.004253	0.003795	0.004309	0.003865	1700	1000	1350	950	0.881
0.004253	0.003900	0.004167	0.003865	1700	950	1400	950	0.909
0.004169	0.003900	0.004167	0.003963	1750	950	1400	900	0.936

Note:
The subsequent columns contain precision of estimation δ_d and sample sizes n_d corresponding to the dth domain, ratio $R = \delta_{d_{\min}} / \delta_{d_{\max}}$.

Like algorithm 1, algorithm 2 was executed three times. In the first execution its parameters were: c_0 = 0.01 (as the initial target CV), ε = 0.0005 and q = 500; in the second execution, ε = 0.0001 and q = 25; and in the third execution, ε = 0.00005 and q = 5.

Table 3

Results of the third execution of algorithm 1, namely for $p = 10$ and $\varepsilon = 0.995$

δ_1	δ_2	δ_3	δ_4	n_1	n_2	n_3	n_4	R
0.004152	0.003922	0.004167	0.003963	1760	940	1400	900	0.941
0.004152	0.003945	0.004139	0.003963	1760	930	1410	900	0.950
0.004136	0.003968	0.004139	0.003963	1770	920	1410	900	0.958
0.004136	0.003968	0.004112	0.003987	1770	920	1420	890	0.959
0.004120	0.003991	0.004112	0.003987	1780	910	1420	890	0.968
0.004104	0.003991	0.004112	0.004012	1790	910	1420	880	0.971
0.004104	0.004015	0.004085	0.004012	1790	900	1430	880	0.978
0.004088	0.004015	0.004085	0.004037	1800	900	1430	870	0.982
0.004072	0.004039	0.004085	0.004037	1810	890	1430	870	0.988
0.004072	0.004039	0.004059	0.004063	1810	890	1440	860	0.992
0.004056	0.004063	0.004059	0.004063	1820	880	1440	860	0.998

Note:
The subsequent columns contain precision of estimation δ_d and sample sizes n_d corresponding to the dth domain, and ratio $R = \delta_{d_{\min}} / \delta_{d_{\max}}$.

The results are presented in Table 4. The first execution needed 13 iterations; it leaded to CV = 0.40 and sample sizes from domains n = $(1490, 968, 1550, 944)^T$, which sum up to the overall sample size equal n = 4952. Interestingly, the second execution of the algorithm led to a smaller sample size, n = 4913, than that from previous execution in spite of a smaller CV obtained (CV = 0.39). This execution was stopped based on the condition from step 4. The last, third execution led to CV = 0.00385 and n = 5001; it took just one iteration. Altogether, algorithm 2 needed 16 iterations so a single stratification algorithm (that is, for one domain) was carried out 64 times; however, there were 39 effective iterations.

Table 4

Results of execution of algorithm 2

CV	n_1	n_2	n_3	n_4	n	E
0.0100	375	163	497	158	1193	1
0.0095	408	180	534	175	1297	1
0.0090	446	199	575	194	1414	1
0.0085	488	222	621	216	1547	1
0.0080	537	250	672	243	1702	1
0.0075	593	284	730	275	1882	1
0.0070	660	325	797	315	2097	1
0.0065	738	373	875	364	2350	1
0.0060	833	436	968	425	2662	1
0.0055	948	517	1078	504	3047	1
0.0050	1089	623	1206	608	3526	1
0.0045	1264	767	1363	747	4141	1
0.0040	1490	968	1550	944	4952	1
0.0039	1543	914	1593	863	4913	2
0.0038*	1597	956	1638	902	5093	2
0.00385	1570	934	1615	882	5001	3

* This iteration has been rejected because provided the worse results than the previous one.

Note:
The subsequent columns contain precision of estimation CV = δ_d, sample sizes n_d corresponding to the dth domain, and the number of execution (E).

Conclusions

The paper presents two algorithms useful in stratifying domains into which a population of study is subdivided, when a sample size from the whole population is fixed. Insofar as could be determined this problem has not yet been posed in the statistical literature. We have presented the application of both algorithms for a particular population subdivided into four domains. The algorithms work for any stratification algorithm that is of optimization type (for some details see Kozak and Verma, 2006), that is, that treats stratification as the optimization problem posed as an optimization function to be minimized under specified constraints, and in which one uses some initial points to start the optimization process. The choice of a stratification method has no bearing on the algorithms' work, as what is going on within the stratification algorithm has no meaning for what is going on within the main algorithm. What has the meaning is the result obtained via the stratification algorithm, that is, the stratification points obtained in a particular execution of the stratification algorithm.

From the comparison of the algorithms we are not able to clearly claim which of them is better. The results obtained via algorithm 2 are better in the sense that it provided a lower CV (which was the function to be minimized) than algorithm 1. However, when there are more than just four domains, algorithm 1 might work faster than algorithm 2 as it is constructed in such a way that in one iteration it works only with the domains for which the results are the worst. It was not observed in the present study. The other aspect of the study is that both algorithms worked with a numerical optimization that might provide local minima − had it not been the case, both algorithms should have provided the same results and just times of their execution would have differed. This shows that there is still a need for an algorithm for optimum stratification that provides globally optimum strata. Such an algorithm would be especially important in official statistics, especially when the survey budget is limited.

It is to be noted that practice shows that with algorithm 2 there may be problems with obtaining a final overall sample size that indeed is equal or even close to the target *n*. This may happen especially when the population is divided into a large number of domains comprising many elements.

These results should be treated as a starting point for further research. A further comparison of the two approaches is necessary, based on simulation studies for a large number of populations as well as various situations; also, a comparison based on real life data might provide useful information on the behavior of the methods. The work on the algorithms presented in this paper has not ended, and it is possible that modifying them might lead to more efficient stratification.

References

Baillargeon S. and Rivest L.P. (2007), Stratification: Stratification of Survey Populations, R package version 1.0.

Baillargeon S., Rivest L.P. and Ferland M. (2007), Stratification en enquetes entreprises: Une revue et quelques avancees. Proceedings of the Survey Methods Section, Statistical Society of Canada.

Baillargeon S. and Rivest L.-P. (2009), A General Algorithm for Univariate Stratification, „International Statistical Review", Vol. 77, pp. 331-344.

Dalenius T. and Hodges J.L. (1959), Minimum Variance Stratification, „Journal of the American Statistical Association", Vol. 54, pp. 88-101.

Eckman G. (1959), An Approximation Useful in Univariate Stratification, „Annals of Mathematical Statistics", Vol. 30, pp. 219-229.

Gunning P. and Horgan J.M. (2004), A Simple Algorithm for Stratifying Skewed Populations, „Survey Methodology", Vol. 30, pp. 159-166.

Hidiroglou M. (1986), The Construction of a Self-Representing Stratum of Large Units in Survey Design, „The American Statistician", Vol. 40, pp. 27-31.

Keskintürk T. and Er Ş. (2007), A Genetic Algorithm Approach to Determine Stratum Boundaries and Sample Sizes of Each Stratum in Stratified Sampling, „Computational Statistics and Data Analysis", Vol. 52, pp. 53-67.

Kozak M. (2004), Optimal Stratification Using Random Search Method in Agricultural Surveys, „Statistics in Transition", Vol. 6(5), pp. 797-806.

Kozak M. (2014), Comparison of Random Search Method and Genetic Algorithm for Stratification, „Communications in Statistics: Simulation and Computation", Vol. 43, pp. 249-253.

Kozak M. and Zieliński A. (2005), Sample Allocation between Domains and Strata, „International Journal of Applied Mathematics & Statistics", Vol. 3, pp. 19-40.

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

Lavallée P. and Hidiroglou M. (1988), On the Stratification of Skewed Populations, „Survey Methodology", Vol. 14, pp. 33-43.

Lednicki B. and Wieczorkowski R. (2003), Optimal Stratification and Sample Allocation between Subpopulations and Strata, „Statistics in Transition", Vol. 6, pp. 287-306.

Nelder J.A. and Mead R. (1965), A Simplex Method for Function Minimization, „Computer Journal", Vol. 7, pp. 308-313.

R Development Core Team (2006), R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org.

Rivest L.P. (2002), A Generalization of Lavallee and Hidiroglou Algorithm for Stratification in Business Surveys, „Survey Methodology", Vol. 28, pp. 207-214.

ON EQUAL-PRECISION STRATIFICATION IN DOMAINS SUBJECT TO FIXED SAMPLE SIZE

Summary

Stratified sampling is one of the most common sampling designs in economic surveys of official statistics. Independent sampling in domains is of special interest for practical reasons; for instance, in Polish economic surveys voivodeships constitute the domains, and in many surveys estimation is required for both the whole country and each voivodeship. The objective of the paper is to present two algorithms for stratification in domains, a population under study is subdivided into, orientated towards minimizing a common value of the coefficients of variation of an estimator considered in the domains, subject to fixed sample size from the whole population. An application of the algorithms and their comparison is presented for an artificial population comprising four domains.