Estimation of Total Electricity Consumption by Survey Sampling Methods

NASR Rodrigue, NGUYEN Hoai-Nam

June 2024

1 Introduction

In this project, we want to estimate the total electricity consumption by statistical methods in the survey sampling framework. By drawing one sample and stimulating it many times, we want to find the most efficient sampling design in terms of design effect and estimator in terms of their coefficient of variation (CV). We consider three sampling designs: Simple random sampling without replacement (SRSWOR), Bernoulli sampling (BE), and Stratified simple ransom sampling without replacement (STSRSWOR). The following estimators will be used to estimate the total electricity consumption after choosing sampling designs: Horvitz-Thompson estimator, Post-Stratified estimator, Ratio estimator, and Regression estimator. Some estimators require strata which will be given by a defined categorical variable available in the data set.
However, in the second part of this project, we will study the basic stratification method called "Cumulative root frequency", that is used in the function of the package $stratification$, that we are studying using the article by Sophie Baillargeon and Louis-Paul Rivest(2011). This foundation method uses one numerical variable to self-construct the strata by finding its optimal segments or boundaries that satisfy the minimization criteria. The purpose is therefore to divide the population into mutually exclusive sub-populations and use the stratified design to determine stratum sample sizes and calculate the precision of the simple expansion estimator of the survey variable.

2 Part 1

2.1 Data presentation

The population we consider is made of the 34997 "communes" of the French Metropolitan in 2021. The data source is Open Data GRD (Gestionnaire de Réseaux de Distribution). However, each "commune" is observed with a different sector, so in total we have 149245 observations. So the code of "communes" is not an identifier and we add a new variable called "id" to distinguish observations. We also add an auxiliary variable accounting for the total electricity consumption for each "commune" and each sector in 2020 and another one accounting for the total "Point De Livraison" (PDL) which is a unique number attached to the household's electric meter.
After some rectifications, we arrive to the dataset with the following variables:

- Identifier
- The code of communes.
- Its name.
- The electricity consumption measured in MWh in 2021.
- The electricity consumption measured in MWh in 2020.
- The number of "Point de Livraison".
- Name of big sectors with 5 categories: "Agriculture", "Industrie", "Résidentiel", "Teritaire" and "Secteur Inconnu".

We aim to estimate the total electricity consumption in France in 2021. We assume that the total electricity consumption in France in 2021 is unknown (variable Y) and we want to estimate it in the disposal of a sample.
The following table displays the number of observations in our dataset, the total electricity consumption in France in 2021 (Total Y), its variance, and its coefficient of variation (CV) over the whole population:

Number of observations	Total Y	Variance	CV
149 245	435 375 997	570 850 501	819%

Table 1: Summary statistics

It is then remarked that the variation of electrical consumption in France in 2021 to its average is large.

2.2 SRSWOR and BE sampling design

In this section, to estimate the total quantity of electricity consumption in France in 2021, we will implement two basic sampling designs:

- The simple random sample without replacement (SRSWOR) with a sample size of 20000
- The Bernouill sampling (BE) with the probability of being selected $\pi = 20000/149245$

The choice of the sample size of 20000 not only guarantees that the ratio between the sample size and the population is reasonable (closely 14% in our case) but also meets our ambition to estimate the total electricity consumption in France in 2021 with a margin of error of 500000 (MWh) under basic sampling designs. Throughout this section, the Horvitz-Thompson (HT) estimator is used. It's theoretically a p-unbiased estimator of the survey variable given the absence of the non-sampling errors. Illustrations of this property cannot be done based on one sample drawn for a given survey design but require therefore drawing several samples using Monte-Carlo simulations. The results from this practice also permit to comparison of the precision between designs using the coefficient of variation (CV). The smaller the coefficient of variation, the more precise the estimator. Comparing two sampling designs (SRSWOR and BE), it is expected that BE is less efficient than the SRSWOR because provided that the CV of the target variable is non-negative, the design-variance of an estimator under BE design is always larger than one under the SRSWOR design.
Firstly, we display, in the following table, the true variance of the HT estimator of the survey variable calculated from the following formula under two sampling designs over the population. Denote N as the population size, n as the sample size, S^2_{yU} is the variance of survey variable under the population setting (U), and $Var(\hat{y}_\pi)$ as the variance of HT estimator of the target variable. Under SRSWOR design:

$$Var(\hat{y}_\pi) = N^2(\frac{1}{n} - \frac{1}{N})S^2_{yU}$$

Under BE design:

$$Var(\hat{y}_\pi) = \frac{1}{\pi}\sum_{i \in U} y_i^2$$

Design	SRSWOR (n = 20000)	BE (π = 20 000 / 149 245)
True variance	5.505×10^{14}	5.587×10^{14}

Table 2: HT estimator of the total electricity consumption

As table 2 shows, the true variance of the HT estimator of the total electricity consumption under the SRSWOR design is smaller than one under the BE design ($5.505 \times 10^{14} < 5.587 \times 10^{14}$). It aligns with our expectations from a theoretical perspective.
Moving now to the drawing of a sample following two sample designs presented (SRSWOR and BE), we want to estimate the total electricity consumption using the obtained sample. The sample size of the SRSWOR design is still as before

of 20 000 units and the probability of being selected for BE design remains fixed as π = 20 000 / 149 245. The following table shows 4 settings corresponding to 2 sampling designs with their 2 cases with and without 1000 Monte Carlo simulations. We display the estimated consumption as well as its CV.

Design	1000 simulations	Estimated consumption	CV
SRSWOR	No	470 403 312	8.02%
SRSWOR	Yes	435 896 577	5.45%
BE	No	396 581 543	3.94%
BE	Yes	434 546 171	5.52%

Table 3: The designs settings

From one sample drawn, the SRSWOR design performs worse than the BE design (8.02% > 3.94%). The reason might come from drawing one sample and our sample is not representative of the characteristics of the whole population. Therefore, the comparison cannot be conducted with results generated from one random draw but multiple drawings. Indeed, with 1000 Monte Carlo simulations, the SRSWOR performs slightly better than the BE design in terms of precision by producing a CV of 5.45% while the latter one is of 5.52%. This result is as expected.

We display two histograms of estimated electricity consumption under two sampling designs (SRSWOR and BE) to reconfirm the better performance of SRSWOR than BE. The histogram under the SRSWOR design is more symmetric and its jump between bins is less dramatic than the BE design. Moreover, its distribution looks more similar to the Normal distribution than the BE design because of the gradual descendant tails.

2.3 Stratified SRSWOR design

In this section, we will implement and display the result from estimating the survey variable (the electricity consumption in France in 2021) from a stratified simple random sampling without replacement (STSRSWOR) for different allocation methods (Proportional allocation and Neyman allocation). In reality, computation of the true standard deviation and then deducing the Neyman allocation are not feasible. One way to approximate these quantities is to use a proxy to the survey variable, a so-called auxiliary variable that is highly correlated with the variable of interest. Instead of stratifying on the survey variable, we stratify on the auxiliary variable. We examine two cases where one auxiliary variable is highly correlated with the survey variable and the other is poorly correlated to compare the power of correlation. Moreover, the comparison between these stratified designs and the SRSWOR design using simulations is necessarily implemented.

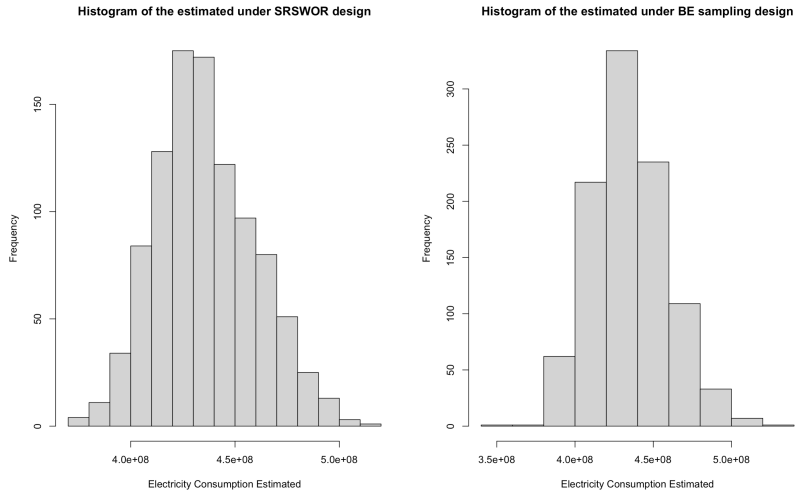

Figure 1: Histograms of the estimators

The following table displays the correlation between two auxiliary variables (electricity consumption in France in 2020 and Total points in France in 2021) with the survey variable (electricity consumption in France in 2021). The electricity consumption in two consecutive years is highly linearly correlated while the other auxiliary variable is poorly correlated with the survey variable.

Variable	Correlation
Consumption 2020	0.99
Total point 2021	0.39

Table 4: Correlations with the survey variable

We consider here 4 settings :

- (A) The STSRSWOR design with proportional allocation
- (B) The STSRSWOR design with Neyman allocation and without auxiliary variable
- (C) The STSRWOR design with Neyman allocation with a highly correlated auxiliary variable as a proxy
- (D) The STSRWOR design with Neyman allocation with a poorly correlated auxiliary variable as a proxy

Under different allocation rules (Proportional and Neyman allocation) and in the presence or not of auxiliary variables, the sample size in each stratum is

computed differently. We display the population size in each stratum (N_h) as well as its sample size (n_h) in the following table:

Settings		1	2	3	4	5
	N_h	32144	25881	34997	21360	34863
(A)	n_h	4308	3468	4690	2862	4672
(B)	n_h	144	10160	3662	479	5555
(C)	n_h	162	10116	3763	462	5497
(D)	n_h	49	181	16811	264	2695

Table 5: Population size and sample size within each setting

The proportional allocation does not take into account the variability inside the stratum so every stratum's sample size is proportional to its population size. On the other hand, the Neyman allocation considers this feature to minimize the target variance. From the above table, it is observed that the settings (B) and (C) - one without an auxiliary variable and another one with a highly correlated variable share a similar distribution of sample size throughout the stratum with a high quantity in the second strata. Meanwhile, the setting (D) with a poorly correlated variable sets a large quantity on the strata 3. This observation could be explained by the fact that setting C approximates the survey variable by the same one but in the previous year while setting D approximates a completely different variable in terms of magnitude as well as characteristics. The following results display the performance of estimating the total electricity consumption when drawing one sample (i.e. without Monte-Carlo simulations). We observe that the STSRSWOR with Neyman allocation outperforms one with proportional allocation in the scenarios without the auxiliary variable and with the highly correlated auxiliary variable while estimating the survey variable with a poorly correlated auxiliary variable, the performance is the worst.

Design	Allocation	Auxiliary variable	Estimated consumption	CV
STSRSWOR	Proportional	No	452 023 239	7.94%
STSRSWOR	Neyman	No	432 178 223	3.66%
STSRSWOR	Neyman	Consumption 2020 (highly correlated)	408 323 673	3.36%
STSRSWOR	Neyman	Total point 2021 (poorly correlated)	459 293 758	8.68%

Table 6: Results from estimating the total of electricity consumption under 4 STSRSWOR designs without Monte-Carlo simulations

To compare the precision of the estimators for the different allocations, we need to compare the results from the Monte-Carlo simulations (1000 samples). The

same conclusion is conducted as in the case without Monte-Carlo simulations with the out-performance of STSRSWOR design with Neyman allocation in the presence of a highly correlated auxiliary variable in comparison with other scenarios and the worst case is when estimating the survey variable with the poorly correlated auxiliary variable. The stratification also upholds its strengths in comparison with the non-stratification case because except for the poorly correlated auxiliary variable case, all stratified simple random (either with proportional allocation or Neyman allocation) leads to a gain of precision to simple random sampling without replacement with the following descending order of precision: STSRSWOR with Neyman allocation on Highly correlated auxiliary variable, STSRSWOR with Neyman allocation on survey variable, STSRSWOR with proportional allocation, SRSWOR (as $3.84\% < 3.93\% < 5.41\% < 5.45\%$)

Design	Allocation	Auxiliary variable	Estimated consumption	CV
STSRSWOR	Proportional	No	435 535 234	5.41%
STSRSWOR	Neyman	No	435 784 402	3.93%
STSRSWOR	Neyman	Consumption 2020 (highly correlated)	435 243 402	3.84%
STSRSWOR	Neyman	Total point 2021 (poorly correlated)	438 61 2930	22.17%
SRSWOR	No	No	435 896 577	5.45%

Table 7: Results from estimating the total of electricity consumption under 4 STSRSWOR designs with Monte-Carlo simulations and one from SRSWOR

2.4 Post-stratified, Ratio and Regression Estimators

Previous sections work with Horvitz-Thompson estimator but from now on, we will consider three new estimators: Post-stratified estimator, Ratio estimator, and Regression estimator. Similar to the previous study, we will examine how well these estimators behave in estimating the total electricity consumption by drawing one sample and then compare their precision based on CV by drawing 1000 Multi Carlo simulations.
Firstly, we use the stratification variable *Sectors* with its 5 categories to perform post-stratification. The following table displays the population size of each stratum, the total, and the variance of the survey variable for each category of the auxiliary variable *Sectors*. Sector "Résidentiel" occupies the most of the population size and consumes the most electricity consumption, but the high variability between observations belongs to sector "Industrie".

For the ratio and regression estimators, two auxiliary variables (Electricity consumption in 2020 and Total point "de livraison") are used. The following figures demonstrate the relationship between each auxiliary variable and the survey

	Agriculture	Industrie	Résidentiel	Secteur Inconnu	Tertiaire
N_h	32 144	25 881	34 997	21 360	34 863
Total of Y	7 121 985	116 061 022	169 589 754	3 283 791	139 319 446
Variance of Y	414 620.4	2 067 576 921	268 602 710	4 604 340	617 991 044

Table 8: Some summary statistics by category of the auxiliary variable *Sectors*

variable. On the one hand, the ratio estimator with Electricity consumption in 2020 is appropriate given the linear relationship between this variable and the survey variable, the zero intercept, and as shown before their high coefficient of correlation. On the other hand, the ratio estimator with the Total point "de livraison" is not such a case because the relationship between this poorly correlated variable and the survey variable is not clearly linear. The same observation could be drawn for the regression estimator.

While the electricity consumption in 2020 is linearly correlated to the electricity consumption in 2021, the form of dependence between the total point "de livraison" and our survey variable is unclear.

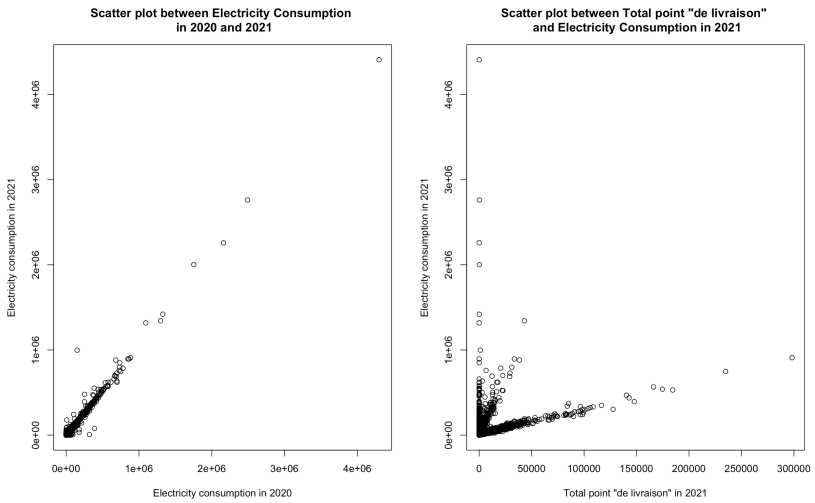

Figure 2: Scatter plots between the different variables.

The following table displays the results, from drawing one sample, of estimating total electricity consumption based on 3 new estimator methods in the same sampling design SRSWOR. We also compare it with the expansion estimator Horvitz-Thompson in the same design.

With Multi-Carlo simulations, the same table is produced to compare the precision of estimators.

Design	Estimator	Auxiliary variable	Estimated consumption	CV
SRSWOR	Post-stratified	Sectors	468 760 427	7.98%
SRSWOR	Ratio	Consumption 2020 (highly correlated)	453 414 061	0.29%
SRSWOR	Ratio	Total point 2021 (poorly correlated)	501 819 675	8.63 %
SRSWOR	Regression	Consumption 2020 (highly correlated)	432 292 798	0.29%
SRSWOR	Regression	Total point 2021 (poorly correlated)	471 446 584	7.78%
SRSWOR	Horvitz-Thompson	No	470 403 312	8.02%

Table 9: Results of estimating total electricity consumption based on 3 new estimator methods in the same sampling design SRSWOR, without Monte-Carlo simulations

Design	Estimator	Auxiliary variable	Estimated consumption	CV
SRSWOR	Post-stratified	Sectors	435 919 673	5.44%
SRSWOR	Ratio	Consumption 2020 (highly correlated)	457 156 835	0.70%
SRSWOR	Ratio	Total point 2021 (poorly correlated)	462 984 150	6.23%
SRSWOR	Regression	Consumption 2020 (highly correlated)	435 352 544	0.72%
SRSWOR	Regression	Total point 2021 (poorly correlated)	436 507 285	4.96%
SRSWOR	Horvitz-Thompson	No	435 896 577	5.45%

Table 10: Results of estimating total electricity consumption based on 3 new estimator methods in the same sampling design SRSWOR, with Monte-Carlo simulations

In summary, both results from one sample or 1000 simulations draw the same conclusion on the out-performance of the ratio estimator and the regression estimator with the auxiliary variable highly correlated with the survey variable. These two estimators generate roughly the same CV which are largely smaller than other estimators or the same estimator but with the poorly correlated auxiliary variable. An adjustment from auxiliary variables gives, therefore, more precision in the estimation. Moreover, in comparison with other settings studied so far, these two settings perform the best, regardless of biasedness or asymptotically unbiasedness properties.

3 Part 2: New stratification methods

Unlike the first part, we already have the sectors that divide the population into mutually exclusive subpopulations, so-called strata. We just simply found the sampling design and performed the estimation on the pre-defined strata levels and observed CV to evaluate its performance.
But in this part, we present new methods to construct the strata by one numerical variable without requiring any available categorical variable. We will stratify the population by using the stratification variable X which is assumed to be positive, known for all the units of the population, and 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 $h = 1, .., L$ such that $b_0 = minX$ and $b_L = maxX + 1$, where $minX$ and $maxX$ are respectively the minimum and the maximum values of the stratification variable.
We will focus on the case where the assumption that the survey variable Y is the same as the stratification variable X is made (i.e. Y = X). However, this assumption is not realistic, and using the same variable to stratify a population and to evaluate the precision of survey estimates might underestimate their variances. One solution is to allocate the sample to the strata based on the anticipated moments of Y knowing that X is in $[b_{h-1}, b_h)$ (i.e. Y $\neq$ X). Nevertheless, given the scope of our study, we only focus on the basic stratification method called "Cumulative root frequency" because it's the foundation in this field and any further methods are its developed versions.

3.1 Basic stratification methods

There are two elementary stratification methods, the cumulative root frequency method of Dalenuis and Hodges (1959) and the geometric method of Gunning and Horgan (2004). The two methods are roughly the same, except that the geometric method sets the stratum boundaries before carrying out the stratum sample size calculations as the cumulative root frequency method. Therefore, we enter in detail only the cumulative root frequency.

3.1.1 Overview of the cumulative root frequency method

The best characteristic for strata construction is the target variable's frequency distribution. Let y_0, y_L be the smallest and largest values of y in the population. The problem is to find intermediate stratum boundaries y_1, y_2, ..., y_{L-1} such that the variance of the estimate $\bar{y}_{st}$, denoted $V(\bar{y}_{st})$ is a minimum where $\bar{y}_{st}$ is the estimate of $\bar{y}$ in the stratified design.

$$V(\bar{y}_{st}) = \frac{1}{n}(\sum_{h=1}^{L} W_h S_h)^2 - \frac{1}{N}\sum_{h=1}^{L} W_h {S_h}^2$$

If the finite population correction is ignored, it is sufficient to minimize $\sum W_h S_h$.

If $f(y)$ is the frequency function of y,

$$W_h = \int_{y_{h-1}}^{y_h} f(t)dt, \quad \frac{\partial W_h}{\partial y_h} = f(y_h)$$

Differentiating $\sum W_h S_h$ over y_h gives equations that are ill-adapted to practical computation. A quick approximate method, due to Dalenius and Hodges (1959), is presented for minimizing $\sum W_h S_h$. Let

$$Z(u) = \int_{y_0}^{u} \sqrt{f(t)}dt$$

If the strata are numerous and narrow, f(y) should be approximately constant within a given stratum. Hence,

$$W_h = \int_{y_{h-1}}^{y_h} f(t)dt = f_h(y_h - y_{h-1})$$

$$S_h = \frac{1}{\sqrt{12}}(y_h - y_{h-1})$$

$$Z_h - Z_{h-1} = \int_{y_{h-1}}^{y_h} \sqrt{f(t)}dt = \sqrt{f_h}(y_h - y_{h-1})$$

where f_h is the "constant" value of f_y in stratum h. By substituting these approximations, we find

$$\sqrt{12}\sum_{h=1}^{L} W_h S_h = \sum_{h=1}^{L}[\sqrt{f_h}(y_h - y_{h-1})]^2 = \sum_{h=1}^{L}(Z_h - Z_{h-1})^2$$

Since $(Z_L - Z_0)$ is fixed, it is easy to verify that the sum on the right is minimized by making $(Z_h - Z_{h-1})$ constant.
Given $f(y)$, the rule is to form the cumulative of $\sqrt{f(y)}$ and choose the y_h so that they create equal intervals on the cum $\sqrt{f(y)}$ scale.
Numerically, for the cum $\sqrt{f(y)}$, the bins are found using multiple steps. A first approximation is done, and then another consists of adapting the approximated values to the sample values. Let us go further into the details:

3.1.2 First approximation

Let us consider that the population is represented by a density function $f(x)$. We have that:

$$Z(u) = \int_{y_0}^{u} \sqrt{f(t)}dt$$

When $u \longrightarrow \infty$, $Z(u)$ approaches an upper bound H. For each $h \in \{1, ..., L-1\}$, the respective roots $y'_1, ..., y'_h, ..., y'_{L-1}$ of the equation

$$Z(u) = \frac{h}{L}H$$

are taken as the (first) approximations, for large L, to the points $y_1, ..., y_h, ..., y_{L-1}$. For example, for the function $f(x) = e^{-x}$, for L=2 strata, the first approximation y'_1 is given by the root of the equation

$$\int_0^u \sqrt{e^{-x}} dt = \frac{1}{2} \times 2$$

$$\longrightarrow 2(1 - e^{-u/2}) = 1$$

Solving this equation for u gives us $x'_1 = u = 1.386$.

3.1.3 Adapting the values

We assume that the density $f(x)$ is stratified into L strata. Two consecutive strata are specified by y_{h-1}, y_h, y_{h+1}. Let us define :

$$I_p(u) = \int_{y_0}^u t^p f(t) dt$$

Based on Dalenuis and Hodges (1959), the set $[y_h]$ of cutting points satisfying the relation

$$\frac{\sigma_h^2 + (y_h - \mu_h)^2}{\sigma_h} = \frac{\sigma_{h+1}^2 + (y_h - \mu_{h+1})^2}{\sigma_{h+1}} \tag{1}$$

corresponds to the minimum variance stratification (MVS), with μ_h being the conditional mean and σ_h^2 the conditional variance. Then we also define

$$J_{ph} = I_p(y_h) - I_p(y_{h-1})$$

μ_h and σ_h^2 can be expressed as functions of $I_p(y_h)$ and so as functions of J_{ph}:

$$\mu_h = \frac{I_1(y_h) - I_1(y_{h-1})}{I_0(y_h) - I_0(y_{h-1})} = \frac{J_{1h}}{J_{0h}}$$

and

$$\sigma_h^2 = \frac{I_2(y_h) - I_2(y_{h-1})}{I_0(y_h) - I_0(y_{h-1})} - \mu_h^2 = \frac{J_{0h} J_{2h} - J_{1h}^2}{J_{0h}^2}$$

Now we substitute these values in (1) and we get the following relation:

$$\frac{J_{2h} - 2y_h J_{1h} + y_h^2 J_{0h}}{\sqrt{J_{0h} J_{2h} - J_{1h}^2}} - \frac{J_{2,h+1} - 2y_h J_{1,h+1} + y_h^2 J_{0,h+1}}{\sqrt{J_{0,h+1} J_{2,h+1} - J_{1,h+1}^2}} = 0 \tag{2}$$

Let us note $A_h = \frac{J_{2h} - 2y_h J_{1h} + y_h^2 J_{0h}}{\sqrt{J_{0h} J_{2h} - J_{1h}^2}}$ and $B_h = \frac{J_{2,h+1} - 2y_h J_{1,h+1} + y_h^2 J_{0,h+1}}{\sqrt{J_{0,h+1} J_{2,h+1} - J_{1,h+1}^2}}$, so expression (2) will be written

$$A_h - B_h = \Delta_h = 0 \tag{3}$$

So that the set of $[y_h]$ satisfying equation (3) corresponds to MVS.
Generally, we do not expect all of those calculated values to satisfy the equation

(3). Thus, we have to adjust these values we obtained to satisfy the condition (3).
Let us consider a rectangular distribution $f(x) = 1$. With no loss of generality, we may assume $0 \leq x \leq b$. For this distribution, we have for L strata, and three consecutive strata, with indices g, h, and i, and for a point y_h that specifies new A'_h and B'_h, we have:

$$\begin{array}{l} \frac{\delta A'_h}{\delta y'_g} = \frac{\delta B'_h}{\delta y'_h} = -\frac{2}{\sqrt{3}} \\ \frac{\delta A'_h}{\delta y'_h} = \frac{\delta B'_h}{\delta y'_i} = \frac{2}{\sqrt{3}} \end{array} \tag{4}$$

and all the other expressions of the type $\delta A'_h/\delta y'_i, \delta B'_h/\delta y'_g$ etc. are equal to zero. And in the same way, we derive analogous expressions for $\delta\Delta_h/\delta y'_h$, etc.
Now let us use these values: we have a set $[x'_h]$ with corresponding Δ'_h-values. We want to find a specific set $[y''_h]$ with $|\Delta''_h| < |\Delta'_h|$.
Using the mean value theorem we get:

$$\Delta''_h = \Delta'_h + (y''_g - y'_g)\frac{\delta\Delta'_h}{\delta y'_g} + (y''_h - y'_h)\frac{\delta\Delta''_k}{\delta y'_k} + (y''_i - y'_i)\frac{\delta\Delta''_h}{\delta y'_i},$$

$$\text{for } h = 1, ..., g, h, i, ..., L-1$$

Note here that $y''_0 = y'_0$ and $y''_L = y'_L$. Then we solve this system for y''_h.
We will skip the details of the calculation and give the computation necessary for the new set:

$$\begin{array}{l} y''_1 = y'_1 - \frac{\sqrt{3}}{2L}[(L-1)\Delta'_1 + (L-2)\Delta'_2 + ... + 2\Delta'_{L-2} + \Delta'_{L-1}] \\ y''_2 = y'_2 - \frac{\sqrt{3}}{2L}[(L-2)\Delta'_1 + 2(L-2)\Delta'_2 + ... + 4\Delta'_{L-2} + 2\Delta'_{L-1}] \\ ... \\ y''_h = y'_h - \frac{\sqrt{3}}{2L}[\sum_{i\leq h} i*(L-h)\Delta'_i + \sum_{i>h} h*(L-i)\Delta'_i] \\ ... \\ y''_{L-1} = y'_{L-1} - \frac{\sqrt{3}}{2L}[\Delta'_1 + 2\Delta'_2 + ... + (L-2)\Delta'_{L-2} + (L-1)\Delta'_{L-1}] \end{array} \tag{5}$$

If necessary, the procedure may be repeated to give a third set $[y'''_h]$ etc. As a final step in the procedure, we now pretend that the above-discussed method for the rectangular case holds reasonably well also for other cases.

Let us now get back to our example with the function e^{-x}. We have for $L = 2$ strata, $y'_1 = 1.386$. And so we compute $A'_1 = 2.276, B'_1 = 2$ which gives us the result

$$\Delta'_1 = 0.276$$

Since Δ'_1 is significantly different from 0, we have to adjust y'_1. We thus compute

$$y''_1 = y'_1 - \frac{\sqrt{3}}{4}\Delta'_1 = 1.27$$

We then evaluate again this value:

$$A'' = 2.016, B_1'' = 2.0007 \text{ which gives us } \Delta_1'' = 0.015$$

Now, Δ_1'' is closer to 0 than Δ_1', and so y_1'' should now be closer to the best point y_1 than y_1' was.

3.1.4 Implementation in the package

All *strata*-functions use Hidiroglou and Shrinath's (1993) rule to allocate the n units in the sample to the strata. The stratum sample sizes are proportional to $N_h^{2q_1}$, $\bar{Y}_h^{2q_2}$, $S_y h^{2q_3}$, where N_h is the size of stratum h, and $\bar{Y}_h$ and $S_y h^2$ are the anticipated or estimated mean and variance of Y in stratum h. We will consider only Neyman allocation which corresponds to $(q_1, q_2, q_3) = (0.5, 0, 0.5)$.
We will use the function *strata.cumrootf* in package *stratification*. Its arguments are

- x: The population vector of the stratification variable
- n: A numeric: the target sample size. It has no default value. The argument n or the argument CV must be input.
- CV: a target CV for the estimates or a predetermined sample size n
- $nclass$: the number of bins of equal size for the x-variable
- Ls: the number of strata
- $alloc$: an allocation rule
- (other arguments ...)

This algorithm pools the $nclass$ bins into Ls strata in such a way that the sums of square roots of the bin frequencies are approximately equal for the Ls strata. In the case of the take-all stratum, the algorithm will automatically identify it and reallocate the rest units for the rest strata using the Neyman allocation. This adjustment is important to ensure a sample size specified in the arguments of the function. Additionally, in a stratified design, it might be useful to constrain some units to be sampled, before construction of the strata. These units are usually extreme values or outliers. This is the so-called certainty stratum.

3.1.5 Application using *strata.cumrootf* (cum $\sqrt{f}$ method)

We will apply the function *strata.cumrootf* to our data in this part. To do so, since these approaches consider that Y=X, we will be using the total consumption of electricity from 2020 as the variable to calculate the boundaries of the strata. We will fix the number of strata to 5, and nclass to 1000, and the allocation rule being Neyman's allocation.
We then proceed to the choice of the CV or the n. In the first part, we used a

sample size of 20000, and we got a CV of approximately 3%. So first, we can fix the target CV of 3% and find the sample size to obtain this objective.
Note that CV* is the Anticipated CV obtained when applying a stratified design to a survey variable.

Model		1	2	3	4	5	n	CV*
Y = X	N_h	134716	10293	3380	787	69		
	n_h	181	55	71	81	64	452	50.5%

Table 11: Stratification results with a target CV

We can see that for a target CV of 3%, the corresponding sample size according to the function is 452. this is very low compared to what we got in the first part. Practically, we get an anticipated CV of 50,5%, which is high compared to our goal.
Now, we will change our strategy so that we predetermine the sample size $n = 20000$, and we find the corresponding CV.

Model		1	2	3	4	5	CV	CV*
Y = X	N_h	134716	10293	3380	787	69		
	n_h	12081	3683	3380	787	69n	0.26%	5.83%

Table 12: Stratification results with a predetermined sample size

Now using $n = 20000$, we have a better CV of 0.26%, with a corresponding anticipated CV of 5,83%.
Calculating sample sizes with a stratification variable underestimates the n needed to reach the target CV for a different survey variable. Using the same variable to stratify a population and to evaluate the precision of survey estimates might underestimate their variances (Y=X) and this assumption (Y=X) is not realistic. So we can allocate the sample to the strata based on the anticipated moments of Y (Y $\neq$ X) knowing that X is in $[x_{h1}, x_h)$ as proposed by Dayal (1985) and Sigman and Monsour (1995). 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. Solutions can be using a log-linear model or a heteroscedastic linear model.

4 Conclusion

In conclusion, we estimated using various methods the total electricity consumption in the French metropolitan area using various sampling methods:
In the first place, we used two auxiliary variables: The consumption in 2020 (high correlation with the survey variable) and the number of points (low correlation with the survey variable).
We observed that results from a single sample or 1,000 simulations consistently indicate that the ratio estimator and the regression estimator outperform others when the auxiliary variable is highly correlated with the survey variable. These two estimators produce comparable coefficients of variation (CV), which are significantly smaller than those of other estimators or the same estimators using a poorly correlated auxiliary variable. Thus, incorporating auxiliary variables enhances the precision of the estimates.
Then, we studied the package stratification, more specifically the cumulative root frequency method, which is implemented in the function $strata.cumrootf$ in the package $stratification$. This method is used to calculate the best strata boundaries that minimize the variance within the strata.
We finally applied the function $strata.cumrootf$ to the data, using the 2020 consumption to calculate boundaries. However, after finding the boundaries and the optimal sample size or target CV, and since the method supposes that Y=X, we saw that using the same variable to stratify a population and to evaluate the precision of survey estimates might underestimate their variances. Then other options might be used, according to the article we studied, but we will not go into their details in this project.

5 Annexes

```
#########################################################
#################### Part 1 #############################
#########################################################
# Import required libraries
library(survey)
library(sampling)
library(dplyr)

##### ---------------------------------------------------
##### 1.1. Data presentation
##### ---------------------------------------------------

# Data Import
energy_21 <- read.csv("survey-2021.csv", sep = ';')
energy_20 <- read.csv("survey-2020.csv", sep = ';')

# Rename the target variable
names(energy_20)[names(energy_20) == "Consommation..MWh."] <- "CONSO"
names(energy_21)[names(energy_21) == "Consommation..MWh."] <- "CONSO"

# Drop duplicates
energy_20 <- energy_20[!duplicated(energy_20), ]
energy_21 <- energy_21[!duplicated(energy_21), ]

# Data preparation
energy_20 <- energy_20 %>%
  group_by(Code.Commune, Libell .Commune,
         Libell .Grand.Secteur, Fili re) %>%
  summarise(Consumption_2020 = sum(CONSO))

energy_21 <- energy_21 %>%
  group_by(Code.Commune, Libell .Commune,
         Libell .Grand.Secteur, Fili re) %>%
  summarise(Consumption_2021 = sum(CONSO),
            Total_point_2021 = sum(Nombre.de.points))

energy_df <- merge(energy_21,
                   energy_20,
                   by = c('Code.Commune','Libell .Commune',
                        'Libell .Grand.Secteur', 'Fili re'),
                   all.x = TRUE)

energy_df <- energy_df %>% mutate_all(~replace(., is.na(.), 0))
```

```r
energy_df <- energy_df[energy_df$Filiere == 'Electricite',]

# Add identifiers
energy_df <- energy_df %>% mutate(id = row_number())

# Correlation
cor(energy_df$Consumption_2021, energy_df$Consumption_2020)
cor(energy_df$Consumption_2021, energy_df$Total_point_2021)

# Look at the data details
attributes(energy_df)
dim(energy_df)
N = dim(energy_df)[1]

##### ------------------------------------------------------------------------
##### 1.2. SRSWOR AND BE SAMPLING DESIGNS
##### ------------------------------------------------------------------------

# Our variable of interest is Consumption_2021
# Our parameter of interest is Total(Consumption_2021)

# Compute the true (population) total, the variance and the average
# of variable of interest
total_CONSO = sum(energy_df$Consumption_2021)
var_CONSO = var(energy_df$Consumption_2021)
CV_CONSO = sqrt(var_CONSO)/mean(energy_df$Consumption_2021)

# Sample size:
sqrt(N**2 * (1-n/N) * 1/n * var_CONSO)*1.96 = 50 000 000
n = 20000
pi <- n/N

# Set seed to replicate results later on
set.seed(2024)

## 1.2.1 The SRSWOR Design: Draw a sample.
## ---------------------------

includ_indic_srswor <- srswor(n = n, N = N)

# Implement the the sampling procedure
srswor_sample <- svydesign(id = ~id,
                           weights = rep(N/n, n),
                           fpc = rep(N, n),
                           data = energy_df[includ_indic_srswor == 1, ])
```

```r
# Estimate the total from the sample
estimated_srswor <- svytotal(~ Consumption_2021, srswor_sample)
estimated_srswor

# Variance
var_estimated_srswor <- SE(estimated_srswor) ^ 2
# Coefficient of variation
coefvar_estimated_srswor <- SE(estimated_srswor) / estimated_srswor[1]

## 1.2.2 The SRSWOR Design: Monte Carlo simulations
## ------------------------

# Initialization of the simulations
n_simu <- 1000

# Create data structures that will hold some values of interest
# vector that will carry the estimated means computed at each simulation
estim_srswor <- matrix(data = 1,
                       nrow = n_simu,
                       ncol = 1)
# vector that will carry the estimated variances computed at each simulation
var_estim_srswor <- matrix(data = 1,
                           nrow = n_simu,
                           ncol = 1)
# Simulate
for (i in 1:n_simu) {

  # Create a sample of indicators following SRSWOR
  includ_indic_srswor <- srswor(n = 20000, N = 149245)

  # Draw the sample from the population
  srswor_sample <- svydesign(id = ~id,
                             weights = rep(N/n, n),
                             fpc = rep(N, n),
                             data = energy_df[includ_indic_srswor == 1, ])

  # Compute estimator of Y
  estimated_srswor <- svytotal(~Consumption_2021, srswor_sample)

  # Save the estimate and its variance in the previously created vectors
  estim_srswor[i] <- estimated_srswor[1]
  var_estim_srswor[i] <- SE(estimated_srswor) ^ 2
}

# Monte Carlo Empirical Mean & Variance
mean(estim_srswor)
```

```r
# Standard deviation
sqrt(var(estim_srswor))

# Coefficient of variation
sqrt(var(estim_srswor)) / mean(estim_srswor)

# Histogram of the results
hist(estim_srswor,
     xlab = 'Electricity Consumption Estimated',
     main = "Histogram of the estimated under SRSWOR design"))

## 1.2.3. The BERNOULLI Design: Draw one sample
## ----------------------------

# Create a function BE, that generates a random vector of inclusion indicators
# according to a Bernoulli sampling design

BE <- function(N, pi) {

  # @param N, integer: the population size
  # @param pi, in (0,1): the inclusion probability

  # @return y: a vector of size N of inclusion indicators (1 = in sample,
  # 0 otherwise)

  # Generate N random values between 0 and 1 from Uniform distribution
  x = runif(N)
  # Create the inclusion indicators following the condition
  y = as.numeric(x < pi)
  return(y)
}

# Now we adapt the code to draw:
# (a) one sample with proportion n/N
includ_indic_be <- BE(149245, pi = 20000/149245)
be_sample <- energy_df[includ_indic_be == 1,]

# (b) 1000 samples with the same proportion n/N
for (i in 1:n_simu) {
  includ_indic_be <- BE(149245, pi = 20000/149245)
  be_sample <- energy_df[includ_indic_be == 1,]
}

# Create function to compute the HT estimator of the total and its variance
# estimate, following the BE design
```

```
BE_svytotal <- function(sample, Y_name, pi) {

  #' @param sample, dataset: the sample from which to compute the estimators
  #' @param Y_name, string: name of the variable of interest
  #' @param pi: the inclusion probability of the BE design

  #' @return c(Y, Var(Y)): a vector containing the estimate of Y
  #' and its variance

  #' Fetch the variable of interest from the dataset
  y_variable = sample[[Y_name]]
  #' HT estimator of the total Y
  y_be_estim = sum(y_variable) / pi
  var_estim_y_be = sum(y_variable ^ 2) * ((1 - pi) / pi^2)

  return(c(y_be_estim, var_estim_y_be))

}

# Implement this function
estimated_y_be <- BE_svytotal(be_sample, "Consumption_2021", 20000/149245)
total_estimated_y_be <- estimated_y_be[1]
var_estimated_y_be <- estimated_y_be[2]
sqrt(var_estimated_y_be) / total_estimated_y_be

## 1.2.4. The BERNOULLI Design: Monte Carlo simulations
## ----------------------------

# Simulate 1000 BE samples, and compute the Monte Carlo empirical
# means & variance

# vector that will carry the estimated means computed at each simulation
total_y_estim_be <- matrix(data = 1,
                           nrow = n_simu,
                           ncol = 1)

# vector that will carry the estimated variances computed at each simulation
var_y_estim_be <- matrix(data = 1,
                         nrow = n_simu,
                         ncol = 1)
for (i in 1:n_simu) {

  #' Same as previously ...
  includ_indic_be <- BE(149245, pi = 20000/149245)
  be_sample <- energy_df[includ_indic_be == 1, ]
```

```
  estimated_y_be <- BE_svytotal(be_sample, "Consumption_2021", 20000/149245)
  total_y_estim_be[i] <- estimated_y_be[1]
  var_y_estim_be[i] <- estimated_y_be[2]
}

# Monte Carlo empirical Mean and Variance
mean(total_y_estim_be)

# standard deviation
sqrt(var(total_y_estim_be))

# coefficient of variation
sqrt(var(total_y_estim_be)) / mean(total_y_estim_be)

# Histogram
hist(total_y_estim_be,
     xlab = 'Electricity Consumption Estimated',
     main = "Histogram of the estimated under BE sampling design'))

#####-------------------------------------------------------------------
##### 1.3: STRATIFIED SRSWOR SAMPLING DESIGN
#####-------------------------------------------------------------------

## 1.3.1. Drawing a sample with a STSRSWOR sampling design and estimation
##-------------------------------------------------------------------

# Our stratum are sectors: 5 big sectors
unique(energy_df$Libell  .Grand.Secteur)

# We use two auxiliary information,
# First auxiliary variable: Consumption in 2020
# Second auxiliary variable: Total point in 2021

# Count number of observation in each sector
bysector <- energy_df %>%
  group_by(Libell  .Grand.Secteur) %>%
  summarise(Count_obs_2021 = n()
            )
bysector <- as.data.frame(bysector)

## 1.3.1.1. Proportional allocation
##-------------------------------

# Size of the samples subsamples
bysector$Prop_alloc <- round(n * bysector$Count_obs_2021 / N)
```

```
# Population size for each stratum
N1 = bysector$Count_obs_2021[1]
N2 = bysector$Count_obs_2021[2]
N3 = bysector$Count_obs_2021[3]
N4 = bysector$Count_obs_2021[4]
N5 = bysector$Count_obs_2021[5]

# Sample size for each stratum
n1 = bysector$Prop_alloc[1]
n2 = bysector$Prop_alloc[2]
n3 = bysector$Prop_alloc[3]
n4 = bysector$Prop_alloc[4]
n5 = bysector$Prop_alloc[5]

# Stratify
stratified_prop <- strata(data = energy_df,
                          stratanames = "Libell  .Grand.Secteur",
                          size = c(n1, n2, n3, n4, n5),
                          method = "srswor")

# Extract the data from the above returned `strata` object
stratified_prop_data <- getdata(data = energy_df,
                                m = stratified_prop)

# Strata sample size
table(stratified_prop_data$Stratum)

# Initialize weights
weights_stsrswor <- c(rep(N1 / n1, n1),
                      rep(N2 / n2, n2),
                      rep(N3 / n3, n3),
                      rep(N4 / n4, n4),
                      rep(N5 / n5, n5))

# Implement the sampling design
stsrswor_sample <- svydesign(id = ~ stratified_prop_data$id,
                             strata = ~ stratified_prop_data$Libell  .Grand.Secteur,
                             weights = weights_stsrswor,
                             fpc = 1 / weights_stsrswor)

# Compute the total estimate and CV
estimated_ybar_stsrswor_prop <- svytotal(~ stratified_prop_data$Consumption_2021,
estimated_ybar_stsrswor_prop[1]
SE(estimated_ybar_stsrswor_prop)/estimated_ybar_stsrswor_prop[1]

## 1.3.1.2. Neyman Allocation
```

```
##-----------------------------

#-Compute-the-standard-deviation-in-each-stratum
sd_conso-<-tapply(energy_df$Consumption_2021,
--------------------energy_df$Libell..Grand.Secteur,
--------------------sd)

#-Compute-the-allocation-size-following-the-Neyman-formula
allocation_size_neyman-<-round(n-*-N-*-sd_conso-/-sum(N-*-sd_conso))

#-Stratify
stratified_neyman-<-strata(data-=-energy_df,
---------------------------stratanames-=-"Libell..Grand.Secteur",
---------------------------size-=-allocation_size_neyman,
---------------------------method-=-"srswor")

#-Exctract-the-data
stratified_neymn_data-<-getdata(data-=-energy_df,
-----------------------------------m-=-stratified_neyman)
#-Strata-sample-size
table(stratified_neymn_data$Stratum)

#-Initialize-weights-following-Neyman
weights_neyman-<-c(rep(N1-/-allocation_size_neyman[1],
---------------------------------allocation_size_neyman[1]),
--------------------rep(N2-/-allocation_size_neyman[2],
---------------------------------allocation_size_neyman[2]),
--------------------rep(N3-/-allocation_size_neyman[3],
---------------------------------allocation_size_neyman[3]),
--------------------rep(N4-/-allocation_size_neyman[4],
---------------------------------allocation_size_neyman[4]),
--------------------rep(N5-/-allocation_size_neyman[5],
---------------------------------allocation_size_neyman[5]))

#-Implement-sampling-design
stsrswor_sample_neyman-<-svydesign(id-=-~stratified_neymn_data$id,
-----------------------strata-=-~stratified_neymn_data$Libell..Grand.Secteur,
-----------------------weights-=-weights_neyman,
-----------------------fpc-=-1-/-weights_neyman)

#-Compute-total-estimate-and-CV
estimated_ybar_stsrswor_neyman<-svytotal(~stratified_neymn_data$Consumption_2021
---------------------------------------stsrswor_sample_neyman)
estimated_ybar_stsrswor_neyman[1]
SE(estimated_ybar_stsrswor_neyman)/estimated_ybar_stsrswor_neyman[1]
```

```
## 1.3.1.3. Neyman Allocation with Consumption_2020
##-------------------------------

# Compute standard deviation of the variable Consumption_2020
sd_conso_2020 <- tapply(energy_df$Consumption_2020,
                        energy_df$Libell .Grand.Secteur,
                        sd)

# Compute the allocation size
allocation_size_neyman_conso2020<-round(n * N * sd_conso_2020 /
                                        sum(N * sd_conso_2020))

# Stratify
stratified_neyman_conso2020 <- strata(data = energy_df,
                                      stratanames = "Libell .Grand.Secteur",
                                      size = allocation_size_neyman_conso2020,
                                      method = "srswor")
# Extract the data
stratified_neyman_conso2020_data <- getdata(data = energy_df,
                                            m= stratified_neyman_conso2020)
# Strata sample size
table(stratified_neyman_conso2020_data$Stratum)

# Initialize weights
weights_neyman_conso2020<- c(rep(N1 / allocation_size_neyman_conso2020[1],
                                 allocation_size_neyman_conso2020[1]),
                             rep(N2 / allocation_size_neyman_conso2020[2],
                                 allocation_size_neyman_conso2020[2]),
                             rep(N3 / allocation_size_neyman_conso2020[3],
                                 allocation_size_neyman_conso2020[3]),
                             rep(N4 / allocation_size_neyman_conso2020[4],
                                 allocation_size_neyman_conso2020[4]),
                             rep(N5 / allocation_size_neyman_conso2020[5],
                                 allocation_size_neyman_conso2020[5]))

# Implement sampling design
stsrswor_sample_neyman_conso2020 <- svydesign(
  id = ~ stratified_neyman_conso2020_data$id,
  strata =~stratified_neyman_conso2020_data$Libell .Grand.Secteu,
  weights = weights_neyman_conso2020,
  fpc = 1 / weights_neyman_conso2020
  )
# Compute the total estimates and CV
estimated_y_stsrswor_neyman_conso2020 <- svytotal(
  ~stratified_neyman_conso2020_data$Consumption_2021,
  stsrswor_sample_neyman_conso2020
```

```
------------------------)
estimated_y_stsrswor_neyman_conso2020[1]
SE(estimated_y_stsrswor_neyman_conso2020)/
------------------------estimated_y_stsrswor_neyman_conso2020[1]

## 1.3.1.4. Neyman Allocation with Total_point_2021
##-------------------------------

# Compute standard deviation of Total_point_2021
sd_point_2021 <- tapply(energy_df$Total_point_2021,
------------------------energy_df$Libell  .Grand.Secteur,
------------------------sd)

# Compute the allocation size
allocation_size_neyman_point_2021 <- round(n * N * sd_point_2021 /
------------------------------------------------------sum(N * sd_point_2021))

# Stratify
stratified_neyman_point_2021 <- strata(data = energy_df,
--------------------------------------stratanames = "Libell  .Grand.Secteur",
--------------------------------------size = allocation_size_neyman_point_2021,
--------------------------------------method = "srswor")
# Extract the data
stratified_neyman_point_2021_data <- getdata(data = energy_df,
-------------------------------------------m = stratified_neyman_point_2021)
# Strata sample size
table(stratified_neyman_point_2021_data$Stratum)

# Initialize weights
weights_neyman_point_2021 <- c(
----rep(N1 / allocation_size_neyman_point_2021[1],
--------allocation_size_neyman_point_2021[1]),
----rep(N2 / allocation_size_neyman_point_2021[2],
--------allocation_size_neyman_point_2021[2]),
----rep(N3 / allocation_size_neyman_point_2021[3],
--------allocation_size_neyman_point_2021[3]),
----rep(N4 / allocation_size_neyman_point_2021[4],
--------allocation_size_neyman_point_2021[4]),
----rep(N5 / allocation_size_neyman_point_2021[5],
--------allocation_size_neyman_point_2021[5]))

# Implement sampling design
stsrswor_sample_neyman_point_2021 <- svydesign(
------------id = ~ stratified_neyman_point_2021_data$id,
------------strata = ~ stratified_neyman_point_2021_data$Libell  .Grand.Secteur,
------------weights = weights_neyman_point_2021,
```

```
            fpc = 1 / weights_neyman_point_2021
            )
# Compute the total estimates and CV
estimated_y_stsrswor_neyman_point_2021 <- svytotal(
                        ~stratified_neyman_point_2021_data$Consumption_2021,
                        stsrswor_sample_neyman_point_2021)
estimated_y_stsrswor_neyman_point_2021[1]
SE(estimated_y_stsrswor_neyman_point_2021)/
            estimated_y_stsrswor_neyman_point_2021[1]

## 1.3.2. Monte Carlo Simulations
##-------------------------------------------------------------------------
# Create data structures that will hold some values of interest
# vector that will carry the estimated means computed at each simulation
total_y_estim_stsrswor <- matrix(data = 1,
                                 nrow = n_simu,
                                 ncol = 4)
# vector that will carry the estimated variances computed at each simulation
var_y_estim_stsrswor <- matrix(data = 1,
                               nrow = n_simu,
                               ncol = 4)

# Simulate
for (i in 1:n_simu) {

  ### Stratify
  ### Proportional allocation
  stratified_prop = strata(data = energy_df,
                           stratanames = "Libell  .Grand.Secteur",
                           size = c(n1, n2, n3, n4, n5),
                           method = "srswor")
  stratified_prop_data = getdata(data = energy_df,
                                 m = stratified_prop)
  ### Neyman allocation
  stratified_neyman = strata(data = energy_df,
                             stratanames = "Libell  .Grand.Secteur",
                             size = allocation_size_neyman,
                             method = "srswor")
  stratified_neyman_data = getdata(data = energy_df,
                                   m = stratified_neyman)

  ### Neyman allocation with Consumption_2020
  stratified_neyman_conso2020 = strata(data = energy_df,
                                       stratanames = "Libell  .Grand.Secteur",
                                       size = allocation_size_neyman_conso2020,
                                       method = "srswor")
```

```
--stratified_neyman_conso2020_data=getdata(data=energy_df,
------------------------------------m=stratified_neyman_conso2020)

--#Neyman consumption with Total_point_2021
--stratified_neyman_point2021=strata(data=energy_df,
---------------------------------------stratanames="Libell .Grand.Secteur",
---------------------------------------size=allocation_size_neyman_point_2021,
---------------------------------------method="srswor")
--stratified_neyman_point2021_data=getdata(data=energy_df,
-------------------------------------------m=stratified_neyman_point2021)

--#Implement sampling designs and compute estimates
--#Proportional allocation
--prop_sample=svydesign(id=~stratified_prop_data$id,
-------------------------strata=~stratified_prop_data$Libell .Grand.Secteur
-------------------------weights=weights_stsrswor,
-------------------------fpc=1/weights_stsrswor)

--#Compute the total estimate and variance
--estimated_y_stsrswor_prop=svytotal(~stratified_prop_data$Consumption_2021,
-----------------------------------prop_sample)
--total_y_estim_stsrswor[i,1]<-estimated_y_stsrswor_prop[1]
--var_y_estim_stsrswor[i,1]<-SE(estimated_y_stsrswor_prop)^2


--#Neyman allocation
--neyman_sample=svydesign(id=~stratified_neyman_data$id,
--------------------strata=~stratified_neyman_data$Libell .Grand.Secteur,
--------------------weights=weights_neyman,
--------------------fpc=1/weights_neyman)

--#Compute total estimate and its variance
--estimated_y_stsrswor_neyman=svytotal(
-----------------------~stratified_neyman_data$Consumption_2021,
-----------------------neyman_sample)
--total_y_estim_stsrswor[i,2]<-estimated_y_stsrswor_neyman[1]
--var_y_estim_stsrswor[i,2]<-SE(estimated_y_stsrswor_neyman)^2

--#Neyman allocation with Consumption_2020
--neymanconso2020_sample=svydesign(
------------id=~stratified_neyman_conso2020_data$id,
------------strata=~stratified_neyman_conso2020_data$Libell .Grand.Secteur,
------------weights=weights_neyman_conso2020,
------------fpc=1/weights_neyman_conso2020)

--#Compute the total estimates and its variance
```

```
  estimated_y_stsrswor_neymanconso2020 = svytotal(
    ~ stratified_neyman_conso2020_data$Consumption_2021,
    neymanconso2020_sample)
  total_y_estim_stsrswor[i, 3] <- estimated_y_stsrswor_neymanconso2020[1]
  var_y_estim_stsrswor[i, 3] <- SE(estimated_y_stsrswor_neymanconso2020)^2

  # Neyman  allocation with Total_point_2021
  neymanpoint2021_sample = svydesign(
    id = ~ stratified_neyman_point2021_data$id,
    strata = ~ stratified_neyman_point2021_data$Libell  .Grand.Secteur,
    weights = weights_neyman_point_2021,
    fpc = 1 / weights_neyman_point_2021)

  # Compute the total estimates and its variance
  estimated_y_stsrswor_neymanpoint2021 = svytotal(
    ~ stratified_neyman_point2021_data$Consumption_2021,
    neymanpoint2021_sample)
  total_y_estim_stsrswor[i, 4] <- estimated_y_stsrswor_neymanpoint2021[1]
  var_y_estim_stsrswor[i, 4] <- SE(estimated_y_stsrswor_neymanpoint2021)^2
}

# Monte Carlo Empirical Mean & Variance
colnames(total_y_estim_stsrswor) <- c("Proportional", "Neyman",
                                      "Neyman-for-conso-2020",
                                      "Neyman-for-point-2021")
colMeans(total_y_estim_stsrswor)

# Standard deviation
sqrt(var(total_y_estim_stsrswor[, "Proportional"]))
sqrt(var(total_y_estim_stsrswor[, "Neyman"]))
sqrt(var(total_y_estim_stsrswor[, "Neyman-for-conso-2020"]))
sqrt(var(total_y_estim_stsrswor[, "Neyman-for-point-2021"]))

# Coefficient of variation
sqrt(var(total_y_estim_stsrswor[, "Proportional"])) /
                        mean(total_y_estim_stsrswor[, "Proportional"])
sqrt(var(total_y_estim_stsrswor[, "Neyman"])) /
                        mean(total_y_estim_stsrswor[, "Neyman"])
sqrt(var(total_y_estim_stsrswor[, "Neyman-for-conso-2020"])) /
                        mean(total_y_estim_stsrswor[, "Neyman-for-conso-2020"])
sqrt(var(total_y_estim_stsrswor[, "Neyman-for-point-2021"])) /
                        mean(total_y_estim_stsrswor[, "Neyman-for-point-2021"])

#####-------------------------------------------------------
##### 2.4. Post-stratified, Ratio and Regression Estimators
#####-------------------------------------------------------
```

```
## 2.4.1. Post-stratified
## ----------------------

# Stratification variable
table(energy_df$Libell .Grand.Secteur)

# Total and variance of survey variable by sectors
by(energy_df$Consumption_2021,energy_df$Libell .Grand.Secteur,sum)
by(energy_df$Consumption_2021,energy_df$Libell .Grand.Secteur,var)

# Draw one sample
si.rec <- srswor(n, N)
ech.si <- svydesign(id=~id,
--------------------weights=rep(N/n,n),
--------------------fpc=rep(n/N,n),
--------------------data=energy_df[si.rec==1,]
)

# Post-strata table
tot.pop <- table(Libell .Grand.Secteur=energy_df$Libell .Grand.Secteur)
tot.pop

# Post-stratified object
ech.post <- postStratify(ech.si,
-------------------------~Libell .Grand.Secteur,
-------------------------tot.pop)

# Estimateur post-stratifi -du total
est.post <- svytotal(~Consumption_2021,
--------------------ech.post)
est.post

# Interpret the sampling weigths:
table(energy_df[si.rec==1,]$Libell .Grand.Secteur)
table(round(1/ech.post$prob,3))

# With simulations
I <- 1000
est.post<-matrix(1,I,1)
for (i in 1:I)
{si.rec <- srswor(n,N)
ech.si <- svydesign(id=~id,
--------------------weights=rep(N/n,n),
--------------------fpc=rep(n/N,n),
--------------------data=energy_df[si.rec==1,])
```

```
ech.post <- postStratify(ech.si,
                         ~Libell  .Grand.Secteur,
                         tot.pop)
est.post[i] <- svytotal(~Consumption_2021,
                        ech.post)[1]
}

# Monte Carlo Empirical Mean
mean(est.post)

# Monte Carlo Empirical CV
sd(est.post)/mean(est.post)

## 2.4.2. Ratio estimator
##-----------------------

# Scatter plot between auxiliary variable and survey variable
par(mfrow = c(1, 2))
plot(energy_df$Consumption_2020, energy_df$Consumption_2021,
     xlab = 'Electricity consumption in 2020',
     ylab = 'Electricity consumption in 2021',
     main = 'Scatter plot between Electricity Consumption \n in 2020 and 2021')
plot(energy_df$Total_point_2021, energy_df$Consumption_2021,
     xlab = 'Total point "de livraison" in 2021',
     ylab = 'Electricity consumption in 2021',
     main = 'Scatter plot between Total point "de livraison" \n and Electricity
par(mfrow = c(1, 1))

# Draw one sample
# Estimateur du R pour auxi Consumption 2021
R.est_conso2020 <-svyratio(~Consumption_2021,
                           ~Consumption_2020,
                           ech.si)
R.est_conso2020
est.ratio_conso2020 <- predict(R.est_conso2020, total = total_CONSO)
est.ratio_conso2020
est.ratio_conso2020$se/est.ratio_conso2020$total

# Estimateur du R pour auxi Total point 2021
R.est_point2021 <-svyratio(~Consumption_2021,
                           ~Total_point_2021,
                           ech.si)
R.est_point2021
est.ratio_point2021<-predict(R.est_point2021, total = total_CONSO)
est.ratio_point2021$se/est.ratio_point2021$total
```

```
# With simulations
est.ratio<- matrix(1,1000,2)

for (i in 1:1000)
{si.rec <- srswor(n,N)
ech.si <- svydesign(id=~id,
--------------------weights=rep(N/n,n),
--------------------fpc=rep(n/N,n),
--------------------data=energy_df[which(si.rec==1),])

# Estimateur du R pour auxi Consumption 2021
est.R<-svyratio(~Consumption_2021,
----------------~Consumption_2020,
----------------ech.si)
est.ratio[i,1] <- predict(est.R, total= total_CONSO)$total

# Estimateur du R pour auxi Total point 2021
est.R_point2020 <-svyratio(~Consumption_2021,
--------------------------~Total_point_2021,
--------------------------ech.si)
est.ratio[i,2] <- predict(est.R_point2020, total= total_CONSO)$total
}

# Monte Carlo Empirical Mean & Variance
colnames(est.ratio) <- c("Consumption_2020","Total_point_2021")
colMeans(est.ratio)

# Coefficient of variation
sqrt(var(est.ratio[,"Consumption_2020"])) / mean(est.ratio[,
-------------------------------------------------"Consumption_2020"])
sqrt(var(est.ratio[,"Total_point_2021"])) / mean(est.ratio[,
-----------------------------------------------------"Total_point_2021"])

## 2.4.2. Regression estimator
##-------------------------

# Draw one sample
si.rec <- srswor(n,N)
ech.si <- svydesign(id=~id,
--------------------weights=rep(N/n,n),
--------------------fpc=rep(n/N,n),
--------------------data=energy_df[si.rec==1,])

# Auxiliary variable total: Consumption_2020
```

```
N_auxi_conso_2020 <-length(energy_df$Consumption_2020)
N_auxi_conso_2020
t.x0_conso_2020 <- sum(energy_df$Consumption_2020)

# Regression estimator with Consumption_2020
ech.si.cal_conso_2020 <- calibrate(ech.si,
------------------------~Consumption_2020,
------------------------c(N_auxi_conso_2020,
-------------------------t.x0_conso_2020))
res_regr_conso_2020 <- svytotal(~Consumption_2021,
--------------------------------ech.si.cal_conso_2020)
res_regr_conso_2020[1]
SE(res_regr_conso_2020)/res_regr_conso_2020[1]

# Auxiliary variable total: Total_point_2021
N_auxi_point_2021 <-length(energy_df$Total_point_2021)
N_auxi_point_2021
t.x0_conso_2021 <- sum(energy_df$Total_point_2021)

# Regression estimator with Total_point_2021
ech.si.cal_point_2021 <- calibrate(ech.si,
-----------------------------------~Total_point_2021,
-----------------------------------c(N_auxi_point_2021,
------------------------------------t.x0_conso_2021))
res_regr_point_2021 <- svytotal(~Consumption_2021,
--------------------------------ech.si.cal_point_2021)
res_regr_point_2021[1]
SE(res_regr_point_2021)/res_regr_point_2021[1]


# With simulations
est.slr<- matrix(1,1000,2)

for (i in 1:1000)
{si.rec <- srswor(n,N)

ech.si <- svydesign(id=~id,
--------------------weights=rep(N/n,n),
--------------------fpc=rep(n/N,n),
--------------------data=energy_df[which(si.rec==1),])

# Auxiliary variable: Consumption_2020
ech.si.cal_conso_2020 <- calibrate(ech.si,
-----------------------------------~Consumption_2020,
-----------------------------------c(length(energy_df$Consumption_2020),
------------------------------------sum(energy_df$Consumption_2020)))
```

```
est.slr[i,1] <- svytotal(~Consumption_2021,
                         ech.si.cal_conso_2020)

# Auxiliary variable: Total_point_2021
ech.si.cal_point_2021 <- calibrate(ech.si,
                                   ~Total_point_2021,
                                   c(length(energy_df$Total_point_2021),
                                   sum(energy_df$Total_point_2021)))
est.slr[i,2] <- svytotal(~Consumption_2021,
                         ech.si.cal_point_2021)

}

# Monte Carlo empirical mean
colnames(est.slr) <- c("Consumption_2020", "Total_point_2021")
colMeans(est.slr)

# Coefficient of variation
sqrt(var(est.slr[,"Consumption_2020"])) / mean(est.slr[,"Consumption_2020"])
sqrt(var(est.slr[,"Total_point_2021"])) / mean(est.slr[,"Total_point_2021"])

################################################################################
############################## Part 2 ##########################################
################################################################################

library(stratification)

# Perform the stratification on Consumption_2020
# Neyman with an anticipated CV of 3%
cum <- strata.cumrootf(x=energy_df$Consumption_2020,
                       CV=0.03,
                       model="none",
                       nclass=1000,
                       Ls=5)
print(cum$CV)

# Evaluate the design on the survey variable
ord <- order(energy_df$Consumption_2020)
var.strata(cum, y = energy_df$Consumption_2021[ord])$RRMSE

# Neyman with n = 20000
cum_n <- strata.cumrootf(x=energy_df$Consumption_2020, n=20000, model="none", ncl

# Evaluate the design on the survey variable
var.strata(cum_n, y = energy_df$Consumption_2021[ord])$RRMSE
```