Linearization, Bootstrap and Jackknife variance estimation for the ratio estimators under PPS sampling

Yiliu Cao

21158335

STAT 854

University of Waterloo

Department of Acturial and Statistical Science

Abstract

Abstract here

1 Introduction

2 Simple Ratio Estimator

In model-based prediction methods, we assume the finite the population values of outcome and auxiliary information follows a superpopulation model ξ where the survey design is ignorable. Even though the estimators derived from specified superpopulation model are efficient; however, they may generate misleading results when the underlying model is misspecified. In contrast, design-based inference does not rely on any model assumptions, which is more robust to model misspecification. This motivates the model-assisted estimators, which integrate design-based inference through a model-assisted approach. Specifically, an estimator is called model-assisted if it is unbiased under the assumed model and approximately unbiased under the probability sampling design []. This approach is particularly powerful, as it leverages the strengths of model-based estimation while preserving the validity of design-based inference.

Simple ratio estimator is one of the model-assisted estimator. It is derived from the superpopulation model ξ such that $y_i = \beta x_i + \varepsilon_i$, $i = 1, \dots, N$ where ε_i are independent with $\mathbb{E}_\xi[\varepsilon_i] = 0$ and $\text{var}_\xi = x_i\sigma^2$. Equivalently, it can be derived as $\mu_y = \beta\mu_x + \bar{\varepsilon}_N$ where $\bar{\varepsilon}_N = N^{-1}\sum_{i=1}^N \varepsilon_i$. In simple ratio estimators, we assume the sample $\mathbf{S}$ is taken by Simple Random Sampling Without Replacement SRSWOR with survey data $\{y_i, x_i : i \in \mathbf{S}\}$ with μ_x known. It can be derived that the weighted least square estimator of β under ξ is $\widehat{\beta} = \frac{\bar{y}}{\bar{x}}$, and hence the model-based prediction estimator μ_y is $\widehat{\mu}_{y\text{R}} = \widehat{\beta}\mu_x = \frac{\bar{y}}{\bar{x}}\mu_x$, where $\widehat{\mu}_{y\text{R}}$ is called the simple ratio estimator of μ_y. Denote $\mathbb{E}_\xi$ and $\mathbb{E}_p$ as taking expectation over ξ and the probability sampling design respectively, it can be shown that $E_\xi\left(\widehat{\mu}_{y\text{R}} - \mu_y\right) = 0$. This follows the first property of model-assisted estimator. Secondly, it can be shown that $E_p\left(\widehat{\mu}_{y\text{R}}\right) \doteq \mu_y$ under the SRSWOR using linearization (next section), which follows the second property described above.

3 Estimating the variance

This paper will discuss three main methods estimating the variance the estimator, which are linearization, bootstrap and jackknife. They have different algorithm but their goals are both to estimate the variance of the given estimator.

3.1 Linearization

The linearization methods, which is also known as the Taylor series methods, is a technique used in survey sampling and statistics to approximate the variance of a nonlinear estimator by replacing it with a linear approximation. More specifically, it linearizes a complex estimator by Taylor expanding it around its true population value and keep only the first-order term. This transforms the nonlinear estimator into an approximately linear one whose variance is

easier to estimate. Formally, suppose $\hat{\theta} = h(\hat{\mathbf{t}})$, it is defined as

$$\tilde{\theta} = h(\mathbf{t}) + \sum_j \frac{\partial h}{\partial t_j} (\hat{t}_j - t_j)$$

where $\tilde{\theta}$ is the linear approximation of the non-linear estimator $\hat{\theta}$.

We can use linearization to estimate the variance of the simple ratio estimator. Since $\hat{\mu}_{y\mathrm{R}} = \frac{\bar{y}}{\bar{x}}\mu_x$ with μ_x known, its variance can be expressed as $\mathrm{Var}_p(\hat{\mu}_{y\mathrm{R}}) = \mu_x^2 \mathrm{Var}_p(\frac{\bar{y}}{\bar{x}})$. Denote $R = \frac{\mu_y}{\mu_x}$ and $\hat{R} = \frac{\bar{y}}{\bar{x}}$. Using the linearization, we can express $\frac{\bar{y}}{\bar{x}}$ as

$$\hat{R} = \frac{\bar{y}}{\bar{x}} = \frac{\mu_y}{\mu_x} + \frac{1}{\mu_x}(\bar{y} - \mu_y) - \frac{\mu_y}{\mu_x^2}(\bar{x} - \mu_x) + \text{ higher order terms}$$

Since we know that $\bar{y} - \mu_y = O_p\left(\frac{1}{\sqrt{n}}\right)$ and $\bar{x} - \mu_x = O_p\left(\frac{1}{\sqrt{n}}\right)$, it can be further expressed as

$$\hat{R} = \frac{\bar{y}}{\bar{x}} = \frac{\mu_y}{\mu_x} + \frac{1}{\mu_x}(\bar{y} - R\bar{x}) + o_p\left(\frac{1}{\sqrt{n}}\right)$$

Hence the variance of $\mathrm{Var}(\hat{\mu}_{y\mathrm{R}})$ can be written as

$$\begin{aligned}
\mathrm{Var}(\hat{\mu}_{y\mathrm{R}}) &= \mu_x^2 \mathrm{Var}(\frac{\bar{y}}{\bar{x}}) \\
&\doteq \mu_x^2 \frac{1}{\mu_x^2} V_p(\bar{y} - R\bar{x}) \\
&= \left(1 - \frac{n}{N}\right) \frac{1}{n} \left(\sigma_y^2 + R^2\sigma_x^2 - 2R\sigma_{xy}\right)
\end{aligned}$$

where $(1 - \frac{n}{N})\frac{1}{n}$ comes from the SRSWOR.

3.2 Bootstrap

In contrast to linearization which uses the Taylor expansion to directly compute the variance of the simple ratio estimator, bootstrap is another non-parametric method that uses the sample we have, either draw by SRSWOR or other sampling methods, to buuld many re-samples and compute the estimates of a given estimator and then compute the variances. The general standard bootstrap procedures (Wu & Thompson, 2020, pp. 224–225) work as follows:

1. Take bootstrap samples $(X_1^*, \ldots, X_n^*)$ from the original sample $(X_1, \ldots, X_n)$ using SRSWR.
2. Estimate the $\hat{\theta}$ using the bootstrap sample

$$\hat{\theta}^* = h\left(X_1^*, X_2^*, \cdots, X_n^*\right)$$

3. Repeat step 1 and 2 for B times independently to have $(\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*)$.

4. Use the $\{\hat{\theta}_b^*\}_{b=1}^B$ to obtain the bootstrap variance estimator for $\hat{\theta}$ by

$$v_B = \frac{1}{B}\sum_{b=1}^{B}\left(\hat{\theta}_b^* - \bar{\theta}^*\right)^2$$

where $\bar{\theta}^* = B^{-1}\sum_{b=1}^{B}\hat{\theta}_b^*$ or $\bar{\theta}^* = \hat{\theta}$. We can also replace B^{-1} by $(B-1)^{-1}$; however, the difference is ignorable for large B.

We can compute the variance of simple ratio estimator using bootstrap as well. It just replaces $\hat{\theta}$ by $\hat{\mu}_{yR}$ and have $\hat{\mu}_{yR}^* = \frac{\bar{y}^*}{\bar{x}^*}\mu_x$ in step 2. After we have B estimates $\hat{\mu}_{yR}$, $(\hat{\mu}_{yR}^*(1), ...\, \hat{\mu}_{yR}^*(B))$, we can estimate its variance by

$$v_B\left(\hat{\mu}_{yR}\right) = \frac{1}{B}\sum_{b=1}^{B}\left\{\hat{\mu}_{yR}^*(b) - \hat{\mu}_{yR}\right\}^2$$

3.3 Jackknife

The second resampling method is Jackknife, and we will only discuss the delete-1 Jackknife estimator. Compared to Bootstrap, which resamples the entire sample and has a new sample of size exactly n at each simulation, delete-1 Jackknife simply removes one unit every time across all units, resulting in n sub-samples of size $n-1$ (Wu & Thompson, 2020, p. 218). In addition, unlike Bootstrap, Jackknife is only applicable to estimator $\hat{\theta}$ that is the sample mean or a function of sample mean (Shao & Tu, 1995, p. 69); we will firstly discuss the delete-1 Jackknife estimator $\hat{\mu}$ and then extend to some general estimator $\hat{\theta}$.

For any general estimator $\hat{\theta}$, the general Jackknife procedures to compute the estimated variance $v(\hat{\theta})$ works as follows:

1. Compute the delete-1 estimators

$$\hat{\theta}_{-i} = h\left(X_1, \cdots, X_{i-1}, X_{i+1}, \cdots, X_n\right), \quad i = 1, 2, \cdots, n$$

2. Compute the delete-1 jackknife variance estimator by

$$v_J = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\theta}_{-i} - \hat{\theta}_{\cdot}\right)^2$$

where $\hat{\theta}_{\cdot} = n^{-1}\sum_{i=1}^{n}\hat{\theta}_{-i}$

We can extend this idea to simple ratio estimator, which follows as

1. Compute the delete-1 Jackknife estimator

$$\hat{\mu}_{-i} = \frac{\bar{y}_{[-i]}}{\bar{x}_{[-i]}}\mu_x, \quad i = 1, 2, \cdots, n \tag{3}$$

where $\bar{y}_{[-i]} = \frac{1}{n-1}\left(\sum_{j=1}^{n} y_j - y_i\right) = \frac{1}{n-1}\left(n\bar{y} - y_i\right)$ and similarly $\bar{x}_{[-i]} = \frac{1}{n-1}\left(n\bar{x} - x_i\right)$

2. Compute the Jackknife variance estimator as

$$v_J\left(\hat{\mu}_{yR}\right) = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_{-i} - \hat{\mu}_{\cdot}\right)^2 \tag{4}$$

where $\hat{\mu}_{\cdot} = \frac{1}{n}\sum_{i=1}^{n}\hat{\mu}_{-i}$

4 PPS Sampling methods

The main sampling method discussed in this paper is the Probability Proportional to Size (PPS) sampling. In PPS sampling, the size (denoted by z) is positively correlated to the outcome y and is available at the survey design stage. For instance, the previous income (z) is proportion to the expenses today y. An important feature under PPS sampling design is that the first order inclusion probability π_i is proportional to the size variable, i.e., $\pi_i \propto z_i$. Since the inclusion probabilities follows that $\sum_{i=1}^{N}\pi_i = n$, it follows that $\pi_i = n\frac{z_i}{T_z}$, or simply $\pi_i = nz_i$ if z is normalized. This paper will discuss three PPS sampling methods: randomized systematic PPS (RSPPS) sampling, PPS sampling with replacement (PPS-WR), and Poisson sampling.

4.1 Randomized systematic PPS sampling

Randomized systematic PPS sampling is a refined version of systematic PPS sampling. In the systematic PPS sampling, the number of sample n is assumed to be fixed and to be larger than 2. In addition, the size variable is assumed to be normalized, $\sum_{i=1}^{N} z_i = 1$, so that the first-order inclusion probability is given by $\pi_i = nz_i$. It contains two main steps:

1. Construct cumulative inclusion probabilities based on the given order of units in the sampling frame

$$b_0 = 0, b_1 = \pi_1, b_2 = \pi_1 + \pi_2, \ldots, b_N = \pi_1 + \ldots, \pi_N = n$$

2. Generate a random starting point $r \sim \text{Unif}(0,1)$, and select the unit i into the sample $\mathbf{S}$ if

$$b_{i-1} < r + k \leq b_i, \quad k \in \{0, 1, 2, \ldots, n-1\}$$

Systematic PPS sampling implies two key properties. First, it implies the first-order inclusion probability $\pi_i = p(i \in \mathbf{S})$ is indeed equal to nz_i. This can be shown by considering two cases: when the interval $(b_{i-1}, b_i]$ lies entirely within a single unite interval $[l,\ l+1]$, and when $(b_{i-1}, b_i]$ spans across an integer point l. The latter case is more complicated as two possible start r. Nevertheless, in both scenarios, it can be verified that $\pi_i = nz_i$. Second, the second order inclusion probability, $\pi_{ij} = p(i, j \in \mathbf{S})$, may be zero for some pairs (i, j). This occurs because the randomized start r may cause certain pairs to be mutually exclusive in the final sample. For example, if $b_1 = 0.1, b_2 = 0.6$ but $r = 0.5$, then units 1 and 2 cannot be selected together as the r falls outside the range to include both.

To address the limitation of systematic PPS sampling regarding the second-order inclusion probabilities, one possible solution is to introduce an additional step before step 1 and 2,

which is to randomize the order of the N units on the sampling frame. By this preliminary step, it preserves the first-order inclusion probabilities that $\pi_i = nz_i$, but this ensures π_{ij} is strictly positive for all possible pairs (i, j). This refined systematic PPS sampling is referred as randomized systematic PPS sampling.

Due to the strictly positive π_{ij}, this ensures unbiased variance estimation for the Horvitz-Thompson estimator (Wu & Thompson, 2020, p. 74). Let $\hat{T}_{y\mathrm{HT}} = \sum_{i\in\mathbf{S}} \frac{y_i}{\pi_i}$ be the H-T estimator of T_y, then

$$\mathbb{E}(\hat{T}_{y\mathrm{HT}}) = T_y$$

Consequently, the variance of $\hat{T}_{y\mathrm{HT}}$ can be expressed as

$$V\left(\hat{T}_{yHT}\right) = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\pi_{ij} - \pi_i\pi_j\right)\frac{y_i}{\pi_i}\frac{y_j}{\pi_j}$$

and an unbiased estimator of the variance is given by

$$v\left(\hat{T}_{yHT}\right) = \sum_{i\in\mathbf{S}}\sum_{j\in\mathbf{S}}\frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}}\frac{y_i}{\pi_i}\frac{y_j}{\pi_j}$$

The above proof can be found at Appendix.

4.2 PPS sampling with replacement

Another PPS sampling metho is PPS sampling with replacement. In this approach, a unit is selected from the population with probabilities $(z_1, \ldots, z_N)$ such that $\sum_{i=1}^{N} z_i = 1$, and independently repeat it for n times with the same probabilities to have a sample of size n. Unlike randomized systematic PPS sampling, this method directly uses the size variables as selection probabilities rather than the inclusion probabilities. More specifically, by Newton's formula that, it can be shown that $\pi_i = P(i \in \mathbf{S}) = 1 - (1 - z_i)^n \doteq nz_i$.

To estimate the total T_y under PPS sampling with replacement, Hansen-Hurwitz (HH) estimator is used instead of Horitz-Thompson estimator. Formally, $\hat{T}_{y\mathrm{HH}}$ is defined as

$$\hat{T}_{y\mathrm{HH}} = \sum_{i\in\mathbf{S}^*}\frac{y_i}{nz_i} = \frac{1}{n}\sum_{i=1}^{n}\frac{Y_i}{Z_i} = \frac{1}{n}\sum_{i=1}^{n}R_i$$

where $\mathbf{S}^*$ is the sample with the duplicated units. It can be shown that $\hat{T}_{y\mathrm{HH}}$ is an unbiased estimator of T_y such that

$$E\left(\hat{T}_{\mathrm{yHH}}\right) = T_y$$

with theoretical variance and variance estimator as

$$V\left(\hat{T}_{y\mathrm{HH}}\right) = \frac{1}{n}\sum_{i=1}^{N} z_i\left(\frac{y_i}{z_i} - T_y\right)^2$$

$$v\left(\hat{T}_{y\mathrm{HH}}\right) = \frac{1}{n(n-1)}\sum_{i\in\mathbf{S}^*}\left(\frac{y_i}{z_i} - \hat{T}_{y\mathrm{HH}}\right)^2$$

All proofs can be found at Appendix. Although nz_i is not exactly equal to π_i as in the Horvitz-Thompson estimator, Hansen-Hurwitz estimator $\hat{T}_{y\text{HH}}$ can be viewed roughly as the $\hat{T}_{y\text{HT}}$ for a large population. This is because, as population size N increases, the chance to have the duplicated units decreases. Therefore, the variance estimation $v(\hat{T}_{y\text{HH}})$ can be used as an approximation to $v(\hat{T}_{y\text{HT}})$. This approximation is advantageous because the variance estimator $v\left(\hat{T}_{y\text{HH}}\right)$ does not require the second-order inclusion probabilities which is very hard to compute in practice. We will further illustrate this in the next section.

4.3 Poisson Sampling

The last PPS sampling discussed in this paper is called the Poisson Sampling. In this approach, each unit is drawn independently from the population with $\pi_i = nz_i$, following the two key steps below:

1. Generate $r_i \sim \text{Unif}(0, 1)$ for each unit i and select the unit i to the sample if $r_i \leq \pi_i$.
2. Repeat step 1 for $i = 1, \ldots, N$ independently.

Comparing to the previous two methods, Poisson sampling does not produce a fixed number of n; the actual sample size may be either larger or smaller than the set n. However, since the inclusion probabilities π_i are known, the Horvitz-Thompson estimator can be applied to estimate the population total T_y. Nevertheless, as each unit is drawn independently, the variance of the HT estimator under Poisson sampling differs slightly from that under randomized systematic PPS sampling. As usual, $\hat{T}_{y\text{HT}}$ is defined as

$$\hat{T}_{y\text{HT}} = \sum_{i \in \mathbf{S}} \frac{y_i}{\pi_i}$$

which is unbiased for T_y. In addition, the theoretical variance of $\hat{T}_{y\text{HT}}$ is

$$V\left(\hat{T}_{y\text{HT}}\right) = \sum_{i=1}^{N} \frac{1 - \pi_i}{\pi_i} y_i^2$$

and an unbiased estimator of this variance is

$$v\left(\hat{T}_{\text{yHT}}\right) = \sum_{i \in \mathbf{S}} \frac{1 - \pi_i}{\pi_i^2} y_i^2$$

Comparing the three methods, each exhibits distinct features in estimating the population total. Randomized systematic PPS sampling is the most general one as it includes the second-order inclusion probabilities. However, π_{ij} does not have a closed form formula and may be computed by simulation in practice []. In contrast, PPS sampling with replacement and Poisson sampling involve independent draws and does not incorporate π_{ij}. The former selects units based on the size variable, while the latter selects based directly on the inclusion probabilities π_i. This is why the Horvitz-Thompson and Hansen-Huritz estimators are introduced, respectively. In the next section, we will explore how to estimate the variance of the ratio estimator under these three PPS sampling methods using linearization, bootstrap, and jackknife techniques.

5 Generalized Ratio Estimator

If the sample is drawn from either of the three PPS sampling method, we can not simply apply the simple ratio estimator to estimate the μ_y as it simple ratio estimator assumes the sample are drawn usng SRSWOR. To deal with the sample drawn from unequal probability sampling design, we need another ratio estimator called generalized ratio estimator (GRE) to estimate the mean population outcome. Like the simple ratio estimator, we still assumes the generalized ratio estimator works under the simple linear regression model, ξ. However, since the PPS sampling is an unequal probability sampling design, instead of setting $\hat{R} = \frac{\bar{y}}{\bar{x}}$, we need to use Horvitz-Thompson estimator or Hansen-Hurwitz estimator to estimate $\hat{R}$ depends on the specific PPS sampling methods.

5.1 Randomized Systematic PPS Sampling

For the most general unequal probability sampling design such as randomized systematic PPS sampling with $\pi_i = P(i \in \mathbf{S})$ and $\pi_{ij} = P(i, j \in \mathbf{S})$, the HT estimators for μ_y and μ_x are written as

$$\hat{\mu}_{y\text{HT}} = \frac{1}{N}\sum_{i\in\mathbf{S}}\frac{y_i}{\pi_i} = \frac{1}{N}\sum_{i\in\mathbf{S}} d_i y_i \quad \text{and} \quad \hat{\mu}_{x\text{HT}} = \frac{1}{N}\sum_{i\in\mathbf{S}}\frac{x_i}{\pi_i} = \frac{1}{N}\sum_{i\in\mathbf{S}} d_i x_i$$

where $d_i = 1/\pi_i$ is called the basic design weights. Since $\hat{\mu}_{y\text{HT}}$ and $\hat{\mu}_{x\text{HT}}$ are unbiased with respect to μ_y and μ_x (Appendix), denote $\hat{R}$ as

$$\hat{R} = \frac{\hat{\mu}_{y\text{HT}}}{\hat{\mu}_{x\text{HT}}} = \frac{\sum_{i\in\mathbf{S}} d_i y_i}{\sum_{i\in\mathbf{S}} d_i x_i}$$

such that $E_p(\hat{R}) \doteq R = \mu_y/\mu_x$. This implies the generalized ratio estimator as

$$\hat{\mu}_{y\text{GR}} = \hat{R}\mu_x = \left(\frac{\sum_{i\in\mathbf{S}} d_i y_i}{\sum_{i\in\mathbf{S}} d_i x_i}\right)\mu_x$$

where μ_x is known. We can verify that the generalized ratio estimator is still a model-assisted estimator. Under the simple linear regression model, the expectation of HT estimator of y is

$$E_\xi\left(\hat{\mu}_{y\text{HT}}\right) = \frac{1}{N}\sum_{i\in\mathbf{S}}\frac{E_\xi[y_i]}{\pi_i} = \frac{1}{N}\sum_{i\in\mathbf{S}}\frac{\beta x_i}{\pi_i} = \beta\hat{\mu}_{x\text{HT}}$$

Also

$$E_\xi\left(\hat{\mu}_{y\text{GR}}\right) = \frac{\beta\hat{\mu}_{x\text{HT}}}{\hat{\mu}_{x\text{HT}}}\mu_x = \beta\mu_x$$

Since we also know that $E_\xi(\mu_y) = \beta\mu_x$, it implies $E_\xi\left(\hat{\mu}_{y\text{GR}} - \mu_y\right) = 0$.

To prove $\hat{\mu}_{y\text{GR}}$ is approximately unbiased under the sampling design. Using linearization, $\hat{R}$ can be expressed as

$$\hat{R} = \frac{\hat{\mu}_{y\text{HT}}}{\hat{\mu}_{x\text{HT}}} = R + \frac{1}{\mu_x}\left(\hat{\mu}_{y\text{HT}} - R\hat{\mu}_{x\text{HT}}\right) + o_p\left(\frac{1}{\sqrt{n}}\right)$$

It follows that

$$\begin{aligned} E_p(\hat{R}) &= E_p\left(R + \frac{1}{\mu_x}\left(\hat{\mu}_{y\mathrm{HT}} - R\hat{\mu}_{x\mathrm{HT}}\right) + o_p\left(\frac{1}{\sqrt{n}}\right)\right) \\ &= R + \frac{1}{\mu_x}\left(\mu_y - R\mu_x\right) + o_p\left(\frac{1}{\sqrt{n}}\right) \\ &\doteq R \end{aligned}$$

Similarly, the variance of $\hat{R}$ over the sampling desing can be written as

$$\begin{aligned} V_p(\hat{R}) &= V_p\left(R + \frac{1}{\mu_x}\left(\hat{\mu}_{y\mathrm{HT}} - R\hat{\mu}_{x\mathrm{HT}}\right) + o_p\left(\frac{1}{\sqrt{n}}\right)\right) \\ &\doteq \frac{1}{\mu_x^2} V_p\left(\hat{\mu}_{y\mathrm{HT}} - R\hat{\mu}_{x\mathrm{HT}}\right) \\ &= \frac{1}{\mu_x^2} V_p\left(\frac{1}{N}\sum_{i\in\mathbf{S}} d_i y_i - \frac{R}{N}\sum_{i\in\mathbf{S}} d_i x_i\right) \\ &= \frac{1}{\mu_x^2} V_p\left(\frac{1}{N}\sum_{i\in\mathbf{S}} d_i (y_i - R x_i)\right) \\ &= \frac{1}{\mu_x^2} V_p\left(\hat{\mu}_{e\mathrm{HT}}\right) \end{aligned}$$

where $\hat{\mu}_{e\mathrm{HT}} = N^{-1}\sum_{i\in\mathbf{S}} d_i e_i$ and $e_i = y_i - Rx_i$. Since $\hat{\mu}_{y\mathrm{GR}} = \hat{R}\mu_x$, the above results imply that

$$E_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq \mu_y \quad V_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq V_p\left(\hat{\mu}_{e\mathrm{HT}}\right)$$

In the general unequal probability sampling, the variance of $V_p\left(\hat{\mu}_{e\mathrm{HT}}\right)$ can be expressed as

$$V_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq V_p\left(\hat{\mu}_{e\mathrm{HT}}\right) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\pi_{ij} - \pi_i\pi_j\right)\frac{e_i}{\pi_i}\frac{e_j}{\pi_j}$$

and

$$v_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq v_p\left(\hat{\mu}_{e\mathrm{HT}}\right) = \frac{1}{\hat{N}^2}\sum_{i\in\mathbf{S}}\sum_{j\in\mathbf{S}}\frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}}\frac{\hat{e}_i}{\pi_i}\frac{\hat{e}_j}{\pi_j}$$

where $\hat{N} = \sum_{i\in\mathbf{S}}\frac{1}{\pi_i} = \sum_{i\in\mathbf{S}} d_i$ and can be replaced by N if the population size is known. Alternatively, we can also write

$$v_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq v_p\left(\hat{\mu}_{e\mathrm{HT}}\right) = \frac{\mu_x^2}{\hat{\mu}_{x\mathrm{HT}}^2}\frac{1}{\hat{N}^2}\sum_{i\in\mathbf{S}}\sum_{j\in\mathbf{S}}\frac{\pi_{ij} - \pi_i\pi_j}{\pi_{ij}}\frac{\hat{e}_i}{\pi_i}\frac{\hat{e}_j}{\pi_j}$$

5.2 Other PPS Sampling Method

The linearization approach described above is applicable to general unequal probability sampling designs that involve second-order inclusion probabilities, such as randomized systematic

PPS sampling. However, as discussed in Sections 4.2 and 4.3, PPS sampling with replacement and Poisson sampling involve different estimation metrics for the population total and is not necessarily to have second order inclusion probabilities. Accordingly, the variance estimation for the generalized ratio estimator must also be adapted to remain consistent with the underlying sampling design.

In the case of Poisson sampling, each unit is drawn independently. As a result, the cross-product terms present in the Yates–Grundy–Sen form of the variance $V_p(\hat{\mu}_{e\mathrm{HT}})$ in equation x vanishes and is replaced by a simpler expression as shown in equation x and y. On the other hand, in section 4.3, we have seen that $\hat{T}_{y\mathrm{HT}}$ under Poisson sampling is unbiased of T_y. This implies that $\hat{R}$ in equation (x) is still approximately design-unbiased of R. Following the same linearization procedures in section 5.1, the variance of generalized ratio estimator under Poisson sampling becomes

$$V_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq V_p\left(\hat{\mu}_{e\mathrm{HT}}\right) = \frac{1}{N^2}\sum_{i=1}^{N}\frac{1-\pi_i}{\pi_i}e_i^2$$

with the corresponding variance estimator given by

$$v_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq v_p\left(\hat{\mu}_{e\mathrm{HT}}\right) = \frac{1}{\hat{N}^2}\sum_{i\in\mathbf{S}}\frac{1-\pi_i}{\pi_i^2}\hat{e}_i^2$$

Comparing to randomized systematic PPS and Poisson sampling, as discussed in section 4.2, PPS sampling with replacement requires the use of the Hansen–Hurwitz (HH) estimator rather than the Horvitz–Thompson (HT) estimator to estimate the population total. Since both $\hat{\mu}_{y\mathrm{HH}}$ and $\hat{\mu}_{y\mathrm{HH}}$ are unbiased for μ_y and μ_x respectively, the generalized ratio estimator under PPS sampling with replacement is constructed as

$$\hat{\mu}_{y\mathrm{GR}} = \left(\frac{\hat{\mu}_{y\mathrm{HH}}}{\hat{\mu}_{x\mathrm{HH}}}\right)\mu_x = \left(\frac{\sum_{i\in\mathbf{S}}\frac{y_i}{nz_i}}{\sum_{i\in\mathbf{S}}\frac{x_i}{nz_i}}\right)\mu_x$$

Therefore, $\hat{R} = \frac{\hat{\mu}_{y\mathrm{HH}}}{\hat{\mu}_{x\mathrm{HH}}}$ is still approximately design-unbiased for R. Using linearization similarly as before, we can obtain

$$E_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq \mu_y \quad V_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq V_p\left(\hat{\mu}_{e\mathrm{HH}}\right)$$

where $\hat{\mu}_{e\mathrm{HH}} = N^{-1}\sum_{i\in\mathbf{S}} w_i e_i$ with $w_i = 1/nz_i$ and $e_i = y_i - Rx_i$. The theoretical variance of the generalized ratio estimator under this method is then

$$V_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq V_p\left(\hat{\mu}_{e\mathrm{HH}}\right) = \frac{1}{N^2}\frac{1}{n}\sum_{i=1}^{N} z_i\left(\frac{e_i}{z_i} - T_e\right)^2$$

with an unbiased estimator of the variance is given by

$$v_p\left(\hat{\mu}_{y\mathrm{GR}}\right) \doteq v_p\left(\hat{\mu}_{e\mathrm{HH}}\right) = \frac{1}{N^2}\frac{1}{n(n-1)}\sum_{i\in\mathbf{S}^*}\left(\frac{\hat{e}_i}{z_i} - \hat{T}_{e\mathrm{HH}}\right)^2$$

where $\mathbf{S}^*$ denotes the sample with duplicated units.

6 Bootstrap and Jackknife in GRE

The previous section introduced the way to to estimate the variance of generalized ratio estimator using linearization. In addition to that, this paper will also introduce how to estimate the variance using Bootstrap and Jackknife resampling methods.

In the bootstrap approach, suppose we have the sample $\mathbf{S} = (X_1, ..., X_n)$ taken from the population using randomized systematic PPS sampling, the procedure to estimate the variance of $\hat{\mu}_{y\mathrm{GR}}$ using bootstrap works as follows:

1. Let $\mathbf{S}^*$ be the set of units selected including duplicated units from the $\mathbf{S}$ using SRSWR.
2. Obtain the bootstrap sample data $\mathbf{S}^*$ from the original data $\mathbf{S}$ such that $\{(y_i, x_i), i \in \mathbf{S}^*\}$
3. Compute $\hat{\mu}_{yR}$ using the bootstrap sample

$$\hat{\mu}^*_{y\mathrm{GR}} = \left(\frac{\hat{\mu}^*_{y\mathrm{HT}}}{\hat{\mu}^*_{x\mathrm{HT}}}\right)\mu_x$$

   where $\hat{\mu}^*_{y\mathrm{HT}} = \sum_{i\in\mathbf{S}^*} d_i y_i$ and $\hat{\mu}^*_{x\mathrm{HT}} = \sum_{i\in\mathbf{S}^*} d_i x_i$ with $d_i = 1/\pi_i$.
4. Repeat step 1 to 3 for B times to have $(\hat{\mu}^*_{y\mathrm{GR}}(1), ... \hat{\mu}^*_{y\mathrm{GR}}(B))$, and then calculate the bootstrap variance estimator of $\hat{\mu}_{yR}$ by

$$v_B\left(\hat{\mu}_{y\mathrm{GR}}\right) = \frac{1}{B}\sum_{b=1}^{B}\left\{\hat{\mu}^*_{y\mathrm{GR}}(b) - \hat{\mu}_{y\mathrm{GR}}\right\}^2$$

For Poisson sampling, the bootstrap procedure is identical to the one described above, since the Horvitz–Thompson estimator is also used for the population total. However, for PPS sampling with replacement, the procedure is slightly different due to the Hansen–Hurwitz estimator. In particular, Step 3 changes to:

$$\hat{\mu}^*_{y\mathrm{GR}} = \left(\frac{\hat{\mu}^*_{y\mathrm{HH}}}{\hat{\mu}^*_{x\mathrm{HH}}}\right)\mu_x$$

where $\hat{\mu}^*_{y\mathrm{HH}} = \sum_{i\in\mathbf{S}^*} \frac{y_i}{nz_i}$, $\hat{\mu}^*_{x\mathrm{HH}} = \sum_{i\in\mathbf{S}^*} \frac{x_i}{nz_i}$.

The delete-1 Jackknife estimator offers an alternative resampling method for estimating the variance of the generalized ratio estimator (GRE). However, unlike the standard Jackknife described in Section 2.2, the GRE now incorporates design weights through either the Hansen–Hurwitz (HH) or Horvitz–Thompson (HT) estimators. Consequently, when implementing the Jackknife, it is necessary to adjust for the weights within each replicate sample.

For the randomized systematic PPS sampling, it is recommended that the total weight for each Jackknife replicate preserves the original estimated total $\hat{N} = \sum_{i\in\mathbf{S}} d_i$ []. Specifically, if the unit i is excluded in a Jackknife replicate, the replicate weights should be defined as

$$\begin{cases} 0 & j = i \\ \frac{\hat{N}}{\hat{N}-d_i} d_j & j \neq i \end{cases}$$

This implies $\sum_{j\in\mathbf{S}^*} d_j = \sum_{j\in\mathbf{S}^*} \frac{\hat{N}}{\hat{N}-d_i} d_j = \frac{\hat{N}}{\hat{N}-d_i}(\hat{N}-d_i) = \hat{N}$.

The complete Jackknife procedure workd as follows

1. Compute the delete-1 Jackknife estimator

$$\hat{\mu}_{y\mathrm{GR}[-i]} = \frac{\hat{\mu}_{y\mathrm{HT}[-i]}}{\hat{\mu}_{x\mathrm{HT}[-i]}}\mu_x, \quad i = 1, 2, \cdots, n$$

   where

$$\hat{\mu}_{y\mathrm{HT}[-i]} = \frac{\sum_{j\neq i} d_j^{(-i)} y_j}{\sum_{j\neq i} d_j^{(-i)}}, \quad \hat{\mu}_{x\mathrm{HT}[-i]} = \frac{\sum_{j\neq i} d_j^{(-i)} x_j}{\sum_{j\neq i} d_j^{(-i)}}$$

$$\text{s.t.} \quad d_j^{(-i)} = \begin{cases} 0 & j = i \\ \frac{\hat{N}}{\hat{N}-d_i} d_j & j \neq i \end{cases}$$

   where $\hat{N} = \sum_{i\in\mathbf{S}} d_i$, $d_i = \frac{1}{\pi_i}$
2. Compute the Jackknife variance estimator as

$$v_J\left(\hat{\mu}_{y\mathrm{GR}}\right) = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_{y\mathrm{GR}[-i]} - \hat{\mu}_{\cdot}\right)^2$$

   where $\hat{\mu}_{\cdot} = \frac{1}{n}\sum_{i=1}^{n}\hat{\mu}_{y\mathrm{GR}[-i]}$

As with the bootstrap procedure, Poisson sampling follows the same Jackknife algorithm as randomized systematic PPS sampling since both designs are based on first-order inclusion probabilities and HT-type estimators. For PPS sampling with replacement, the Hansen–Hurwitz estimator is used and the Jackknife procedure must be modified accordingly. In this case, Step 1 is adjusted to:

$$\hat{\mu}_{y\mathrm{GR}[-i]} = \frac{\hat{\mu}_{y\mathrm{HH}[-i]}}{\hat{\mu}_{x\mathrm{HH}[-i]}}\mu_x, \quad i = 1, 2, \cdots, n$$

where $d_i = \frac{1}{nz_i}$ are the HH design weights.

7 Simulation

This simulation study is adopted from Changbao and Thompson (2020), where we firstly simulate the finite population data and then apply each of the three PPS sampling methods to generate corresponding samples. The objective of this study is to compare variance estimates obtained by the three proposed methods which are linearization, Bootstrap and Jackknife, under the three PPS sampling design respectively. In addition, this simulation study will also compute the confidence interval of μ_y based on the generalized ratio estimator, and evaluate their performance by comparing them to the true value of μ_y. All the R codes in the simulation can be found at Appendix.

7.1 Simulation design

In this study, the covariates is assumed to be univariate and is generated from the uniform distribution $X_i \sim \text{Unif}(0, 1)$ with size variables generated from an exponential distribution shifted by 0.5, $z \sim \exp(1) + 0.5$. The outcome y_i is generated according to the model

$$y_i = 1 + 2x_i + 2.5z_i + \varepsilon_i$$

where ε_i is an independent error term drawn from a standard normal distribution, $\varepsilon_i \sim \mathcal{N}(0, 1)$. In addition, the population size is set to $N = 5000$, and the sample size $n = 500$. The R codes for generating the population data can be found at Appendix.

Table 1: Summary statistics of the population data for outcome y, covariate x and size z

Variable	Mean	SD	Min	Median	Max
y	5.767	2.714	-0.603	5.257	24.549
x	0.512	0.288	0.000	0.520	1.000
z	1.500	0.999	0.500	1.193	8.988

Table 1 presents the summary statistics for each variable in the population data. It seems that the true population mean μ_y is 5.767 with standard deviation 2.714. In the remainder of this section, we estimate μ_y using the generalized ratio estimator using different methods and compare the results to the true value.

7.2 Simulation metrics

As we discussed in Section 5 and 6, the estimation procedure of variance estimation for linearization, Bootstrap and Jackknife under each PPS sampling method is distinct due to different ways to draw the sample. In this simulation study, we follow the estimation procedures described in those sections precisely to estimate μ_y and its variance using generalized ratio estimator. However, the only exception is in the application of linearization under randomized systematic PPS sampling. As mentioned in section 5.1, the estimation of second order inclusion probabilities π_{ij} is challenging in practice. Although Thompson and Wu (2008) proposed a simulation-based approach to estimate π_{ij}, it requires long time to generate the full matrix of joint inclusion probabilities as the number of simulation is about 1,000,000 []. Therefore, this study will not consider that method; instead, we approximate using the Hansen–Hurwitz estimator for variance estimation under randomized systematic PPS sampling.

To construct the confidence interval of $\hat{\mu}_y$, this paper proceed as follows. For linearization method, we adopt the approach recommended by Wu and Thompson (2020), to build the confidence interval as

$$\left(\hat{\mu}_{y\text{GR}} - Z_{\alpha/2}\left[v_p\left(\hat{\mu}_{y\text{GR}}\right)\right]^{1/2}, \quad \hat{\mu}_{y\text{GR}} + Z_{\alpha/2}\left[v_p\left(\hat{\mu}_{y\text{GR}}\right)\right]^{1/2}\right)$$

where $Z_{\alpha/2}$ is the $1-\alpha/2$ quantile of standard normal distribution, and α is the significance level. For the confidence interval of Bootstrap and Jackknife, this paper will use the Bootstrap percentile approach [shao and Tu, pp. 132-133]. More specifically, if $\alpha = 0.05$, the 95% confidence interval is constructed by taking the 2.5th and 97.5th percentiles of the bootstrap or jackknife resampled estimates.

Furthermore, to obtain a benchmark for the true variance of the generalized ratio estimator under each PPS sampling method, we also employ a Monte Carlo simulation approach to estimate the variance empirically. Specifically, for each PPS sampling method, we generate 10,000 independent samples from the population and compute the generalized ratio estimator for each replicate. The empirical variance is then calculated based on these 10,000 replicates. The R codes for this approach can be found on the Appendix.

7.3 Simulation results

Table 2: Point Estimates and Confidence Intervals

Method	Sampling	Variance	Estimate	CI	Width
Linearization	RSYSPPS	0.211	5.596	(4.70, 6.50)	1.80
Linearization	Poisson	0.182	6.078	(5.24, 6.91)	1.67
Linearization	PPS_WR	0.177	5.732	(4.91, 6.56)	1.65

Table 2 shows the estimated variance of generalized ratio estimator of μ_y using linearization as described in section 5 under the three PPS sampling methods. It seems that the PPS sampling with replacement has the best performance among the three sampling methods as it has the lowest variance and the narrowest confidence interval. More importantly, this approach gives a very close point estimate $\hat{\mu}_y$ to the true population mean (5.732 vs. 5.767). For the rest two methods, the randomized systematic PPS sampling seems to have worst performance which has a estimated variance significantly larger than the other two.

For bootstrap

Table 3: Point Estimates and Confidence Intervals

Method	Sampling	Variance	Estimate	CI	Width
Bootstrap	RSYSPPS	0.200	5.596	(4.84, 6.59)	1.75
Bootstrap	Poisson	0.215	6.078	(5.31, 7.06)	1.75
Bootstrap	PPS_WR	0.192	5.732	(4.97, 6.69)	1.72

For Jackknife

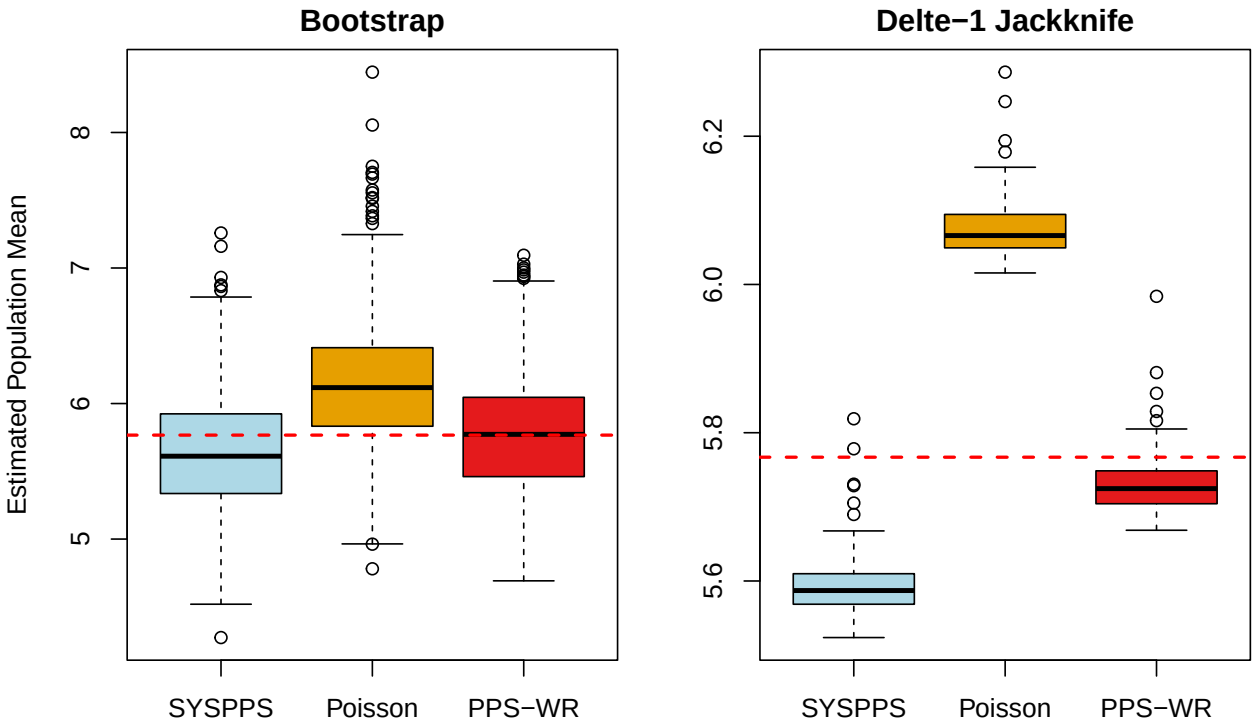

Figure 1: *Booxplot of distribution of the estimated population mean using generalized ratio estimator via Bootstrap and Delete-1 Jackknife under the three PPS sampling method; red dashed line is the true population mean*

Table 4: Point Estimates and Confidence Intervals

Method	Sampling	Variance	Estimate	CI	Width
Jackknife	RSYSPPS	0.217	5.596	(4.68, 6.51)	1.83
Jackknife	Poisson	0.204	6.078	(5.19, 6.96)	1.77
Jackknife	PPS_WR	0.196	5.732	(4.86, 6.60)	1.73

8 Discussion

9 Appendix

9.1 R codes

9.1.1 Generate the population data and draw samples under the three PPS sampling methods

The R codes for generating the population and sample data

```
# Generate the population data
set.seed(854)
N <- 5000
n <- 100

x <- runif(N)
z <- 0.5 + rexp(N)

y <- 1 + 2 * x + 2.5 * z + rnorm(N)
pdata <- cbind(y, x, z)

##############################################################################
# 1. Randomized Systematic PPS Sampling Method
syspps <- function(x, n){
  N = length(x) # The population size
  U = sample(N,N) # Sample the index without replacement
  xx = x[U] # Find the corresponding size values
  z = rep(0,N) # Initial the size variable
  for(i in 1:N) z[i] = n * sum(xx[1:i]) / sum(x) # The grid of b
  r = runif(1)
  s = numeric()
  for(i in 1:N){
    if(z[i] >= r){
      s = c(s, U[i])
      r = r + 1
    }
  }
  return(s[order(s)])
}
set.seed(854)
sam = syspps(z, n) # The n units selected
ys = y[sam] # The y values in S, same as ys=y[sam]
piN = n * z/sum(z) # The inclusion probabilities
# sum(piN) # Check: sum(pi) = n
pis = piN[sam] # pi_i for i in S

# We normalize the z here for the approximate of HT using HH later
```

```r
zs <- z/sum(z)
zs <- zs[sam]

sdata_syspps = pdata[sam,] # The sample data matrix
sdata_syspps <- cbind(sdata_syspps, pis, zs)

##############################################################################
# 2. PPS sampling with replacement
set.seed(854)
z <- z/sum(z) # normalized size variable
sam <- sample(N, n, replace = T, prob = z)
ys <- y[sam] # Values of y in the sample
zs <- z[sam] # Values of the size variable

sdata_wrpps <- pdata[sam,]
sdata_wrpps <- cbind(sdata_wrpps, zs)
sdata_wrpps <- sdata_wrpps[, !(colnames(sdata_wrpps) == "z")]

##############################################################################
# 3. Poisson sampling method
set.seed(854)
piN <- n * z / sum(z) # Inclusion probabilities
sam <- numeric()
for(i in 1:N){
  r = runif(1)
  if(r <= piN[i]) sam = c(sam,i)
}
ys = y[sam]
pis = piN[sam]

sdata_poisson = pdata[sam,]
sdata_poisson <- cbind(sdata_poisson, pis)
##############################################################################
# Optional: Export the population and the sample data
# write.csv(pdata, "./data/pdata.csv", row.names = FALSE)
# write.csv(sdata_syspps, "./data/sdata_syspps.csv", row.names = FALSE)
# write.csv(sdata_wrpps, "./data/sdata_wrpps.csv", row.names = FALSE)
# write.csv(sdata_poisson, "./data/sdata_poisson.csv", row.names = FALSE)
```

9.1.2 R codes for implementing linearization for each PPS sampling method

The R codes for linearization for each PPS sampling method

```r
# Read the generated data
pdata <- read.csv("data/pdata.csv")
sdata_syspps <- read.csv("data/sdata_syspps.csv")
sdata_poisson <- read.csv("data/sdata_poisson.csv")
sdata_wrpps <- read.csv("data/sdata_wrpps.csv")

##############################################################################
# 1. Using HH to estimate HH for RSPPS and PPS sampling with replacement
linVarGR_HH <- function(sdata, mux, N, conf = 0.95){
  y <- sdata$y
  x <- sdata$x
  z <- sdata$zs
  n <- length(y)

  # HH weights & point estimate
  w  <- 1 / (n * z)          # HH weight per draw
  Ty <- sum(w * y)
  Tx <- sum(w * x)
  mu_GR <- (Ty / Tx) * mux
  Rhat  <- Ty / Tx

  # Linearised residuals
  e <- y - Rhat * x

  # HH variance for residual total
  Te_hat <- sum(w * e) # HH total of residuals
  v_lin <- (1 / N^2) *
    (1 / (n * (n - 1))) *
    sum((e / z - Te_hat)^2)

  # Confidence intervals
  z  <- qnorm(1 - (1 - conf)/2)
  se <- sqrt(v_lin)
  ci <- c(mu_GR - z * se,
          mu_GR + z * se)

  return(list(point = mu_GR,
              var = v_lin,
              se = se,
              ci = ci,
              level = conf))
}

##############################################################################
```

```r
# 2. Estimating the variance and confidence interval for Poisson Sampling
# under linearization
linVarGR_poisson <- function(sdata, mux, N, conf = 0.95) {
  y  <- sdata$y
  x  <- sdata$x
  pi <- sdata$pis # first-order inclusion probs
  d  <- 1 / pi    # HT weights
  n  <- length(y)

  # Point estimate
  muY_HT  <- sum(d * y) / N
  muX_HT  <- sum(d * x) / N
  mu_GR   <- (muY_HT / muX_HT) * mux
  Rhat    <- muY_HT / muX_HT

  # Linearised residuals
  e_hat <- y - Rhat * x

  # Poisson HT variance
  v_lin <- (1 / N^2) * sum((1 - pi) / pi^2 * e_hat^2)

  # Confidence intervals
  z   <- qnorm(1 - (1 - conf)/2)
  se  <- sqrt(v_lin)
  ci  <- c(mu_GR - z * se,
           mu_GR + z * se)

  return(list(point = mu_GR,
              var = v_lin,
              se = se,
              ci = ci,
              level = conf))
}

##############################################################################
# Use the functions to estimate the variance for each PPS sampling method
mux <- mean(pdata$x)

# 1. For radomized systematic PPS sampling
vlinear_syspps  <- linVarGR_HH(sdata_syspps,
                               mux = mux,
                               N   = 5000)
# vlinear_syspps
# 2. For Poisson Sampling
```

```r
vlinear_poisson <- linVarGR_poisson(sdata_poisson,
                                    mux = mux,
                                    N   = 5000)
# vlinear_poisson
# 3. For PPS sampling with replacement
vlinear_wrpps  <- linVarGR_HH(sdata_wrpps,
                              mux = mux,
                              N   = 5000)
# vlinear_wrpps
```

9.1.3 R codes for Bootstrap under each PPS sampling Method

R codes for Bootstrap

```r
# Read the generated data
pdata <- read.csv("data/pdata.csv")
sdata_syspps <- read.csv("data/sdata_syspps.csv")
sdata_poisson <- read.csv("data/sdata_poisson.csv")
sdata_wrpps <- read.csv("data/sdata_wrpps.csv")

##############################################################################
# 1. Bootstrap function for randomized systematic PPS and
# Poisson sampling (use pi)
varBootGR <- function(sdata, mux, N, B = 1000, conf = 0.95){
  set.seed(3)
  ys <- sdata$y
  xs <- sdata$x
  pis <- sdata$pis
  n <- length(ys)

  # Hortitz-Thompson weights
  di  <- 1 / pis

  # Point estimate
  muY_HT <- sum(di * ys) / N
  muX_HT <- sum(di * xs) / N
  muY_GR <- (muY_HT / muX_HT) * mux

  muGR_star <- numeric(B)
  for (b in seq_len(B)) {
    bsam <- sample.int(n, n, replace = TRUE)
    yb <- ys[bsam]
    xb <- xs[bsam]
    db <- di[bsam]
```

```r
    muY_HT_star <- sum(db * yb) / N
    muX_HT_star <- sum(db * xb) / N
    muGR_star[b] <- (muY_HT_star / muX_HT_star) * mux
  }

  # Compute the variance of generalized ratio estimator
  vB <- (B - 1) * var(muGR_star) / B

  # Confidence interval
  alpha <- 1 - conf
  ci    <- quantile(muGR_star,
                    probs = c(alpha/2, 1 - alpha/2),
                    names = FALSE)

  return(list(var = vB,
       ci  = ci,
       level = conf,
       point = muY_GR,
       estimates = muGR_star))
}

##############################################################################
# 2. Bootstrap function for PPS sampling with replacement (use z)
varBootGR_wrpps <- function(sdata, mux, N, B = 1000, conf = 0.95){
  set.seed(854)
  ys <- sdata$y
  xs <- sdata$x
  zs <- sdata$zs
  n  <- length(ys)

  # Hansen-Hurwitz weights
  dHH <- 1 / (n * zs)

  # Point estimate
  Ty_HH <- sum(dHH * ys)
  Tx_HH <- sum(dHH * xs)
  muY_GR <- (Ty_HH / Tx_HH) * mux

  muGR_star <- numeric(B)
  for (b in seq_len(B)) {
    bsam <- sample.int(n, n, replace = TRUE)

    yb <- ys[bsam]
    xb <- xs[bsam]
```

```
    zb <- zs[bsam]
    dB <- 1 / (n * zb)

    Ty_star <- sum(dB * yb)
    Tx_star <- sum(dB * xb)
    muGR_star[b] <- (Ty_star / Tx_star) * mux
  }

  # Compute the variance
  vB <- (B - 1) * var(muGR_star) / B

  # Confidence interval of mu_y
  alpha <- 1 - conf
  ci <- quantile(muGR_star,
                 probs = c(alpha/2, 1 - alpha/2),
                 names = FALSE)

  return(list(var = vB,
              ci = ci,
              level = conf,
              point = muY_GR,
              estimates = muGR_star))
}

##############################################################################
mux <- mean(pdata$x)
N <- 5000
B <- 1000

vBoot_syspps <- varBootGR(sdata_syspps, mux, N, B)
# vBoot_syspps
vBoot_poisson <- varBootGR(sdata_poisson, mux, N, B)
# vBoot_poisson
vBoot_wrpps <- varBootGR_wrpps(sdata_wrpps, mux, N, B = 1000)
# vBoot_wrpps
```

9.1.4 R codes for delete-1 Jackknife under each PPS sampling Method

R codes for Delete-1 Jackknife

```
# Read the simulated data
pdata <- read.csv("data/pdata.csv")
sdata_syspps <- read.csv("data/sdata_syspps.csv")
sdata_poisson <- read.csv("data/sdata_poisson.csv")
sdata_wrpps <- read.csv("data/sdata_wrpps.csv")
```

```r
##############################################################################
# 1. The Delete-1 Jackknife variance estimation for H-T estimators
varJackGR <- function(sdata, mux, N, level = 0.95) {
  ys  <- sdata$y
  xs  <- sdata$x
  pis <- sdata$pis

  n    <- length(ys)
  di   <- 1 / pis
  Nhat <- sum(di)

  # Point estimate
  muY_HT <- sum(di * ys) / N
  muX_HT <- sum(di * xs) / N
  muGR   <- (muY_HT / muX_HT) * mux

  muGR_mi <- numeric(n)
  for (i in seq_len(n)) {
    gamma   <- Nhat / (Nhat - di[i])
    di_mi   <- di * gamma
    di_mi[i] <- 0
    muY_HT_mi <- sum(di_mi * ys) / sum(di_mi)
    muX_HT_mi <- sum(di_mi * xs) / sum(di_mi)
    muGR_mi[i] <- (muY_HT_mi / muX_HT_mi) * mux
  }

  # Jackknife variance
  mu_dot <- mean(muGR_mi)
  vJack  <- (n - 1) / n * sum((muGR_mi - mu_dot)^2)
  seJack <- sqrt(vJack)

  # Confidence interval
  alpha <- 1 - level
  z     <- qnorm(1 - alpha/2)
  ci    <- c(lower = muGR - z * seJack,
             upper = muGR + z * seJack)

  return(list(point = muGR,
              var = vJack,
              ci = ci,
              estimates = muGR_mi))
}

##############################################################################
```

```r
# 2. Delete-1 Jackknife estimator using HH estimators
varJackGR_wrpps <- function(sdata, mux, level = 0.95) {
  ys <- sdata$y
  xs <- sdata$x
  zs <- sdata$zs
  n  <- length(ys)

  # HH weights
  z_i <- zs / sum(zs)
  wi  <- 1 / (n * z_i)

  # Point estimate
  muY_HH <- sum(wi * ys) / sum(wi)
  muX_HH <- sum(wi * xs) / sum(wi)
  muY_GR <- (muY_HH / muX_HH) * mux

  muGR_mi <- numeric(n)
  for (i in seq_len(n)) {
    # Replicate weights
    wi <- wi
    wi_mi <- (n / (n - 1)) * wi
    wi_mi[i] <- 0

    muY_HH_mi <- sum(wi_mi * ys) / sum(wi_mi)
    muX_HH_mi <- sum(wi_mi * xs) / sum(wi_mi)
    muGR_mi[i] <- (muY_HH_mi / muX_HH_mi) * mux
  }

  # Jackknife variance
  mu_dot <- mean(muGR_mi)
  vJack  <- (n - 1) / n * sum((muGR_mi - mu_dot)^2)
  seJack <- sqrt(vJack)

  alpha  <- 1 - level
  zcrit  <- qnorm(1 - alpha/2)
  ci     <- c(lower = muY_GR - zcrit * seJack,
              upper = muY_GR + zcrit * seJack)

  return(list(point = muY_GR,
              variance = vJack,
              ci = ci,
              estimates = muGR_mi))
}
```

```r
##############################################################################
mux <- mean(pdata$x)
N   <- 5000

vJack_syspps <- varJackGR(sdata_syspps, mux, N)
# vJack_syspps
vJack_poisson <- varJackGR(sdata_poisson, mux, N)
# vJack_poisson
vJack_wrpps <- varJackGR_wrpps(sdata_wrpps, mux)
# vJack_wrpps
```

Reference

Shao, J., & Tu, D. (1995). *The jackknife and bootstrap* (pp. XVII, 517). Springer New York. https://doi.org/10.1007/978-1-4612-0795-5

Wu, C., & Thompson, M. E. (2020). *Sampling theory and practice*. Springer Cham. https://doi.org/10.1007/978-3-030-44246-0