Group Project Report:

Analyzing Stroke Patient Data In Relation to Population Proportion and Blood Glucose Level

STAT 344: Sample Survey
University of British Columbia
Professor: Dr.Wu Lang

Group Members:

Name	Student Number	Contribution
Edward Bai (Leader)	89979660	Part I, Introduction
Xiaoran Fan	12259511	Coding
Zhuoran(Serena) Feng	89939391	Coding
Minghao Wang	56536469	Part I, Section 2-3: writeup Section 4: assisted writeup
Yolanda Wu	47088331	Part I, Section 2-4: assisted writeup Section 4.1: writeup Part II: writeup

PART I - Sample Data Analysis

1 - Introduction: Background and Objective

Unveiling Insights: Exploring the Stroke Prediction Dataset

In the ever-evolving landscape of healthcare, the quest for innovative approaches to predict and prevent life-altering conditions remains paramount. Among these conditions, strokes stand out as one of the leading causes of mortality and long-term disability globally. Recognizing the potential of data-driven insights, researchers have turned their attention to datasets designed to unravel the complexities surrounding stroke prediction.

The "Stroke Prediction Dataset" emerges as a valuable repository, offering a multifaceted glimpse into the factors contributing to stroke occurrence. Through meticulous curation and thoughtful inclusion of diverse variables, this dataset presents an opportunity to dissect the intricate interplay between medical, demographic, and lifestyle factors in the context of stroke risk.

This research embarks on a journey through the expansive landscape of the Stroke Prediction Dataset. As per the World Health Organization (WHO), stroke ranks as the second most common global cause of death, accounting for around 11% of all fatalities. The dataset we have chosen is employed for forecasting the likelihood of a patient experiencing a stroke, relying on input variables such as gender, age, various medical conditions, and smoking habits. Those variables will support us in building a better model to contain the successful estimation. Each entry in the dataset contains pertinent patient information.

Our exploration extends beyond the binary realm of prediction, aiming to discern subtle nuances in risk factors, identify high-risk cohorts, and contribute to the evolving narrative of personalized medicine. As we navigate through the data, we aspire to bridge the gap between raw information and clinical relevance, fostering a deeper comprehension of the intricate tapestry that is stroke estimation and average glucose level estimation.

2 - Data Collection and Summary

2.1 - Data Selection

The dataset is collected online from Kaggle. Kaggle serves as an online platform for data science competitions and a community for data scientists and machine learning professionals,

operating under the umbrella of Google LLC. It is an open data source website. The “Stroke Prediction Dataset” has 5110 observations and 12 attributes, and was last updated in the year of 2021.

The attributes are:

- **id** (quantitative) - unique identifier
- **gender** (categorical) - “Male”, “Female” or “Other”
- **age** (quantitative) - age of patient
- **hypertension** (binomial) - 0 if the patient doesn't have hypertension and 1 if the patient has hypertension
- **heart_disease** (binomial) - 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
- **ever_married** (binomial) - "No" or "Yes"
- **work_type** (categorical) - "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
- **Residence_type** (binomial) - "Rural" or "Urban"
- **avg_glucose_level** (quantitative) - average glucose level in blood
- **bmi** (quantitative) - body mass index
- **smoking_status** (categorical) - "formerly smoked", "never smoked", "smokes" or "Unknown"
- **stroke** (binomial) - 1 if the patient had a stroke or 0 if not

2.2 - Population and Parameter of Interest

While for the purpose of this project, we have treated the entire dataset as the population of interest, we expect the results of our analysis to provide useful insights for the global population regarding stroke awareness and prevention. This generalization is based on the fact that no geographical/population information is provided for the dataset’s source, hence our conclusions will not address specific regions and/or subpopulations.

We will center our attention on two key parameters within our population. First, we will explore the “stroke” attribute, where a value of 1 signifies a patient who experienced a stroke and 0 indicates no stroke. Second, we will explore the “average_glucose_level” attribute, offering insight into each patient’s average glucose level in blood. Hence, our primary parameters of interest are the average glucose level in the blood and the proportion of stroke events.

Given our group had access to the dataset where samples were drawn from, we were able to extract the population parameters for further analysis, where:

- The population average blood glucose level = 106.1477 mg/dL, and
- The population stroke incidence rate = 0.04872798 = 4.87%.

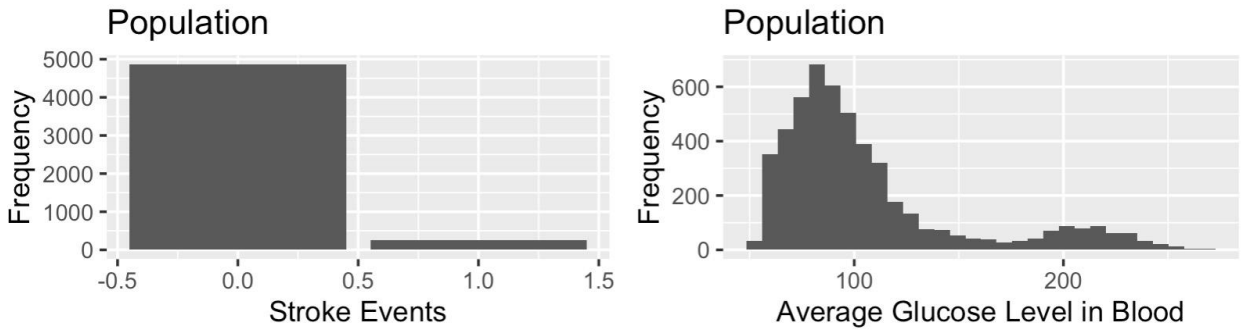

Figure 1. *Distributions of population parameters of interest (left: proportion parameter; right: quantitative parameter).*

2.3 - Sample Size

Calculating the appropriate sample size is possible using the population size, a Z-score of 1.96, assuming worst-case p = 0.5, and selecting an acceptable margin of error. However, it is worth noting that the sample size calculations are typically performed prior to data collection. In our specific situation, we have already treated the entire dataset as the population of interest. Hence, we have opted to set our sample size at 350, which represents less than 10% of the population, enabling us to leverage the central limit theorem.

2.4 - Sampling Methods

In this project, we will employ two different techniques. Simple Random Sampling (SRS) and Stratified Random Sampling.

A simple random sample refers to a randomly chosen subset extracted from a population. This approach ensures that every member of the population has an equal chance of being selected. It is one of the simplest of all probability sampling techniques due to the fact it only involves a single random selection and demands minimal prior knowledge about the population. Since it is random, any research conducted with this sample is expected to have strong internal and external validity.

Stratified sampling on the other hand is a suitable method when our objective is to ensure that particular characteristics are fairly reflected in the sample. This approach involves dividing the population into H mutually exclusive and exhaustive subsets or strata. Within each stratum, a simple random sample is taken. The outcomes from each stratum are then combined to estimate the overall population characteristics. This approach allows us to create individual samples from sub-populations namely strata to make inferences about the whole population's characteristics.

We have selected “work_type” for stratification based on the potential for greater variability compared to other categorical variables. Work types often reflect different lifestyles,

stress levels, and physical activities, all of which can significantly affect both glucose levels and the occurrence of stroke events. Examining health indicators after stratification by work type could yield a more refined and insightful understanding of the data. Therefore, it allows for the inclusion of these factors in the analysis, offering a more comprehensive understanding of their impact on health outcomes. For additional verification, we set seed(1), obtained a sample, and calculated the standard error for all categorical attributes based on the sample, such as 'gender,' 'hypertension,' 'heart_disease,' 'ever_married,' 'work_type,' 'Residence_type,' and 'smoking_status.' Based on our sample, we identified that the 'work_type' attribute exhibits the lowest standard error in estimating the average glucose level in the blood, solidifying our choice to utilize it as the primary stratification variable. It's important to highlight that sample variation may significantly influence this outcome, leading to potentially different standard errors with other samples. Hence, the decision regarding which strata to select may carry uncertainty due to the variability in the calculated standard error from the sample.

While the 'work_type' attribute doesn't yield the smallest standard error in estimating the proportion of stroke events, we still chose to employ it for our stratification. This decision is based on its ability to offer the smallest standard error in estimating the average glucose level in the blood, which contains more information.

As students from non-medical backgrounds, we acknowledge our lack of education in the healthcare domain. Hence, we have avoided using professional medical attributes such as ‘hypertension’ when assigning stratum and instead proposed more understandable attributes supported by computing and comparing SEs. Based on all these reasons discussed, we will leverage the ‘work_type’ attribute for stratification in our later analysis.

3 - Data Analysis

3.1 - The estimated average glucose level in blood

For SRS (Simple Random Sampling), the estimated average of glucose level in blood is 106.514 mg/dL with a SE of 2.334 and having a 95% confidence interval of (101.939, 111.089). The formulas used are as follows:

$$\bar{y}_{SRS} = \frac{1}{n} \cdot \sum_{i=1}^{n} y_i$$

$$SE[\bar{y}_{SRS}] = \sqrt{\left(1 - \frac{n}{N}\right) \times \frac{s_s^2}{n}}$$

$$95\%\ CI = \bar{y}_{SRS} \pm 1.96 \times SE[\bar{y}_{SRS}]$$

As for stratified random sampling, the estimated average of glucose level in blood is 104.872 mg/dL with SE of 2.123 and having a 95% confidence interval of (100.711, 109.033). The formulas used are as follows:

$$\bar{y}_{STR} = \sum_{h=1}^{H} \left(\frac{N_h}{N}\right) \bar{y}_{S_h}$$

$$SE[\bar{y}_{STR}] = \sqrt{\sum_{h=1}^{H} \left(\frac{N_h}{N}\right)^2 \left(1 - \frac{n_h}{N_h}\right) \frac{s^2_{y_h}}{n_h}}$$

$$95\%\ CI\ = \bar{y}_{SRS} \pm 1.96 \times SE[\bar{y}_{SRS}]$$

Where:

- y_i is the average blood glucose level for the *i*th observation in the sample,
- n is the sample size ($n = 350$),
- N is the population size ($N = 5510$), and
- $s^2_{y_h}$ is the sample variance of the average blood glucose level of the *h*th stratum.

Although the standard error of stratified random sampling is smaller than the standard error of SRS, the resulting confidence intervals appear to be similar. When compared to the expected average glucose level, both estimates seem to exceed the normal blood glucose concentration range of 70-100 mg/dL slightly as suggested by the World Health Organization. This is reflective of the population parameter of 106.1477 mg/dL, which also falls slightly above the average blood glucose concentration range.

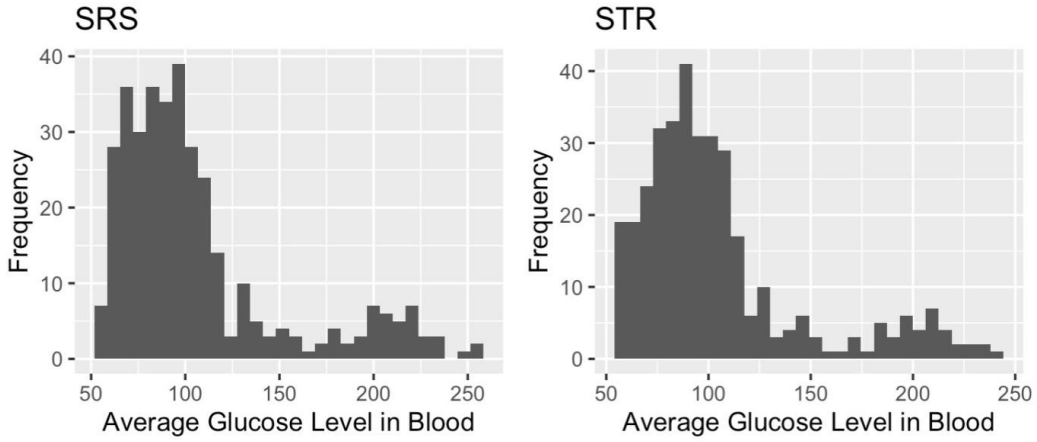

Figure 2. *Sample distributions of average blood glucose level (quantitative variable) obtained via two sampling methods (left: Simple Random Sample; right: Stratified Sample).*

3.2 - The estimated proportion of stroke events

For SRS (Simple Random Sampling), the estimated proportion of stroke events is 0.040 with SE of 0.0101 and having a 95% confidence interval of (0.0202, 0.0598). The formulas are as follows:

$$\widehat{p}_{SRS} = \frac{1}{n} \cdot \sum_{i=1}^{n} y_i$$

$$SE[\widehat{p}_{SRS}] = \sqrt{\left(1 - \frac{n}{N}\right) \times \frac{\widehat{p}(1-\widehat{p})}{n}}$$

$$95\%\ CI\ = \widehat{p}_{SRS} \pm 1.96 \times SE[\widehat{p}_{SRS}]$$

As for stratified sampling, the estimated proportion of stroke events is 0.0486 with SE of 0.0110 and having a 95% confidence interval of (0.0270, 0.0703). The formulas are as follows:

$$\widehat{p}_{STR} = \sum_{h=1}^{H} \left(\frac{N_h}{N}\right) \widehat{p}_{S_h}$$

$$SE[\widehat{p}_{STR}] = \sqrt{\sum_{h=1}^{H} \left(\frac{N_h}{N}\right)^2 \left(1 - \frac{n_h}{N_h}\right) \frac{s^2_{p_h}}{n_h}}$$

$$95\%\ CI\ = \widehat{p}_{STR} \pm 1.96 \times SE[\widehat{p}_{STR}]$$

Where:

- y_i is the binary variable indicating presence (1) or absence (0) of stroke event for the *i*th observation in the sample,
- n is the sample size ($n = 350$),
- N is the population size ($N = 5510$), and
- $s^2_{p_h}$ is the sample variance of the proportion in the *h*th stratum.

The standard error of both the stratification sampling method and the simple random sampling method is extremely similar when estimating the proportion of stroke events.

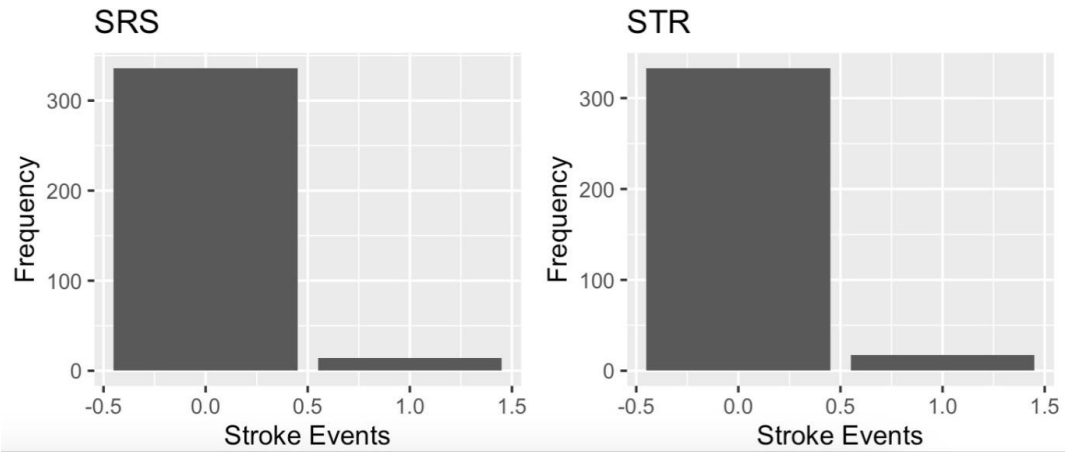

Figure 3. *Sample distributions of stroke events (binary variable) obtained via two sampling methods (left: Simple Random Sample; right: Stratified Sample).*

3.3 - Comparing the Advantages and Disadvantages of Sampling Methods Employed

Analyzing the methods employed in sampling techniques is crucial for obtaining reliable and unbiased results in research. One commonly used method is Simple Random Sampling, valued for its simplicity and minimal requirement of a basic understanding of the Stroke Prediction Dataset. The straightforward execution of this method makes it accessible to researchers with varying levels of expertise, reducing the likelihood of errors in implementation. Moreover, Simple Random Sampling effectively minimizes bias in this case (the estimated average of glucose level in blood in SRS is closer to the population's true value than that in STR), ensuring that each member of the population has an equal chance of being included in the sample.

On the other hand, Stratified Sampling emerges as a more favored approach in many cases due to its ability to provide a more accurate representation of subgroups within a population. Despite its advantages, this method comes with its own set of challenges. It is inherently more complex and time-consuming to implement compared to Simple Random Sampling. However, the additional effort is often justified by the benefits it offers. Stratified Sampling involves dividing the population into distinct strata based on certain characteristics. This division ensures that each subgroup is adequately represented in the sample, thereby reducing the potential for bias. The result is a more comprehensive and nuanced view of the population's diversity. This becomes particularly important when studying a heterogeneous population with distinct subgroups, as it enables researchers to draw conclusions that are more applicable to different segments of the population.

In a specific scenario, when estimating both the average glucose level in blood and the proportion of stroke events, it was observed that both Stratified Sampling and Simple Random Sampling methods yielded comparable standard errors. This indicates that, in this particular

context, the two methods demonstrated similar levels of precision in their estimations. Therefore, the choice between these methods may depend on the specific goals of the study, the characteristics of the population under investigation, and the available resources.

4 - Summary and Discussion

We chose to use the stratification method since it provided us with the lowest standard error. Based on our computed results, we determined the average blood glucose level to be 104.872 mg/dL, with a standard error of 2.123 and a 95% confidence interval ranging from 100.711 to 109.033. On the other hand, the estimated proportion of stroke events is 0.0486, with a standard error of 0.0110 and a 95% confidence interval from 0.0270 to 0.0703. It is noteworthy that the standard errors for both the average blood glucose level and the proportion of stroke events are relatively low, which indicates a consistent level of precision across our estimations.

In this case, all of the 95% confidence intervals computed in our analysis above successfully captured the true population parameter. When compared to the true population average (106.1477) and true proportion (0.04872798), our findings show that the simple random sampling (SRS) method for the average blood glucose level yielded results closer to the true population average. In contrast, the stratified random sampling for the proportion of stroke events aligned more closely with the true population proportion. This observation underscores the effectiveness of different sampling methods in capturing various aspects of the population.

Elevated blood glucose levels present substantial health hazards, such as potential harm to blood vessels, nerves, and vital organs, along with an elevated risk of heart disease and stroke. Conversely, lower blood glucose levels can manifest symptoms like dizziness, confusion, and in severe cases, unconsciousness. Therefore, by considering the given average as a reference point against their personal blood glucose levels, individuals can take proactive steps to reduce potential health risks.

4.1 - Limitations

- *Limitations within the setup*
    1. Population Size - Our study was based on a dataset with 5110 observations, which, while substantial, might not fully represent the global population. Generalizing our results to a larger population should be done with caution.
    2. Sample Size Selection - The sample size was chosen arbitrarily, with the primary consideration being adherence to the Central Limit Theorem. This approach, while statistically valid, may not capture all nuances of a more diversified sampling strategy.
- *Limitations within the methodology and analysis*

1. Stratification Criteria - The selection of attributes for stratification was not informed by professional medical input. This lack of expertise could have influenced our stratification choices, potentially affecting the representativeness of our sample.

- *Limitations within the results*
    1. Applicability of Insights - The data source's lack of specific regional or demographic details limits the applicability of our insights. This means our findings might not accurately reflect variations that occur in different regions or among different population groups.

PART II - Research Article Summary

"The Emperor's New Tests" (Perlman and Wu) critiques a series of contemporary statistical methods that claim to outperform the traditional likelihood ratio test (LRT), and argues the LRT remains an effective method for conducting multiparameter hypothesis testing. While some individuals view these new tests as powerful solutions for data analysis, the authors caution readers that these tests may not be as robust as suggested. In particular, the paper indicates these tests may introduce "unwarranted and inappropriate inferences" as they do not account for the researchers' intentions or the context of the data (Perlman and Wu). Further, adopting such potentially misleading approaches can obscure the need for a clear specification of an alternative hypothesis, which is crucial for the interpretation of any significance test. The title of this research article is therefore akin to the emperor's new clothes, which are invisible to those unfit for their positions. Ultimately, statistical methods are tools that require careful handling, critical approaches, and a thorough understanding of their limitations. It is only with this knowledge, combined with the context of the problem, that an appropriate statistical tool can be selected to maximize its potential to provide accurate insights. Another key takeaway from this paper is that one should be skeptical of nouveau methods that promise easy answers without clear context or consideration of the underlying assumptions, which serves as a reminder for both statisticians and readers from other industries.

5 - Appendix

5.1 - Data Source

Fedesoriano. “Stroke Prediction Dataset.” *Kaggle*, 26 Jan. 2021, www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset.

5.2 - Additional References

Perlman, Michael D., and Lang Wu. “The emperor’s new tests.” *Statistical Science*, vol. 14, no. 4, 1999, https://doi.org/10.1214/ss/1009212517.

Riley, Leanne. “Indicator Metadata Registry Details.” *World Health Organization*, World Health Organization, www.who.int/data/gho/indicator-metadata-registry/imr-details/2380#:~:text=The%20expected%20values%20for%20normal,and%20monitoring%20glycemia%20are%20recommended, Accessed 9 Nov. 2023.

5.3 - Codes

```
# Read in data
stroke_data <- read.csv("healthcare-dataset-stroke-data.csv", header=T)
# pop size
N <- nrow(stroke_data)
N
# 5110
n <- 350
# n is assigned to be 350, which is smaller than 10% of pop. size. Then assume
# that we can use CLT.
############################## Parameter from pop ###################################
avg_pop <- mean(stroke_data$avg_glucose_level)
prop_pop <- mean(stroke_data$stroke)
################################## SRS ######################################
set.seed(1)
SRS_index <- sample.int(N, n, replace = FALSE)
stroke_srs <- stroke_data[SRS_index, ]

# estimator 1: average of average glucose level in blood
avg_srs <- mean(stroke_srs$avg_glucose_level)
# SE
avg_se_srs <- sqrt((1 - n/N)*var(stroke_srs$avg_glucose_level)/n)

# estimator 2: prop. of stroke event
prop_srs <- mean(stroke_srs$stroke)
# SE
prop_se_srs <- sqrt((1 - n/N)*prop_srs*(1 - prop_srs)/n)
################################## STR ######################################
##### step1: decide how to stratify
#### method 1: based on `gender`#####
attach(stroke_data)
# check if there are missing values in variable `avg_glucose_level`
sum(is.na(avg_glucose_level))  # 0
# check if there are missing values in variable `gender`
sum(is.na(gender))  # 0
N.h.gender <- tapply(avg_glucose_level, gender, length) # population size for different gender
N.h.gender
genders <- names(N.h.gender) # name of the genders
genders
N <- sum(N.h.gender) # population size is 5110
```

```
detach(stroke_data)

set.seed(1)
n.h.gender <- round((N.h.gender/N) * n)
STR.sample.gender <- NULL
for (i in 1: length(genders))
{
  row.indices <- which(stroke_data$gender == genders[i])
  sample.indices <- sample(row.indices, n.h.gender[i], replace = F)
  STR.sample.gender <- rbind(STR.sample.gender, stroke_data[sample.indices, ])
}

## estimate 1
avg.h.gender <- tapply(STR.sample.gender$avg_glucose_level,
                       STR.sample.gender$gender, mean)
# manually added avg of "Other" level, because the pop. size of "Other" is too
# small(only 1), so there is not any unit from "Other" level.
avg.h.gender <- c(avg.h.gender, 0)
names(avg.h.gender) <- c("Female", "Male", "Other")

avg.var.h.gender <- tapply(STR.sample.gender$avg_glucose_level,
                           STR.sample.gender$gender, var)
avg.var.h.gender <- c(avg.var.h.gender, 0)
names(avg.var.h.gender) <- c("Female", "Male", "Other")

avg.se.h.gender <- sqrt((1 - n.h.gender/N.h.gender) * avg.var.h.gender/n.h.gender)
# assign the se of "Other" is 0, because var of "Other" is 0, nh of "Other" is
# also 0, 0/0 cannot be calculated, so we just need to manually assign the value
avg.se.h.gender["Other"] <- 0

avg.str.gender <- sum(N.h.gender / N * avg.h.gender)
avg.se.str.gender <- sqrt(sum((N.h.gender / N)^2 * avg.se.h.gender^2))

## estimate 2
prop.h.gender <- tapply(STR.sample.gender$stroke,
                        STR.sample.gender$gender, mean)
prop.h.gender <- c(prop.h.gender, 0)
names(prop.h.gender) <- c("Female", "Male", "Other")

# prop.var.h.gender <- tapply(STR.sample.gender$avg_glucose_level,
```

```
# STR.sample.gender$gender, var)

prop.se.h.gender<- sqrt((1-n.h.gender/N.h.gender) * prop.h.gender*(1-prop.h.gender)/n.h.gender)
prop.se.h.gender["Other"] <- 0

prop.str.gender <- sum(N.h.gender / N * prop.h.gender)
prop.se.str.gender <- sqrt(sum((N.h.gender / N)^2 * prop.se.h.gender^2))

avg_compare <- data.frame(avg = c(avg.str.gender),
                          SE = c(avg.se.str.gender))
prop_compare <- data.frame(prop = c(prop.str.gender),
                           SE = c(prop.se.str.gender))
avg_compare
prop_compare

#### method 2: based on `hypertension` ####
attach(stroke_data)
N.h.hyper <- tapply(avg_glucose_level, hypertension, length)
N.h.hyper
hyper <- names(N.h.hyper)
hyper
N <- sum(N.h.hyper) # population size is 5110
detach(stroke_data)

set.seed(1)
n.h.hyper <- round((N.h.hyper/N) * n)
STR.sample.hyper <- NULL
for (i in 1: length(hyper))
{
  row.indices <- which(stroke_data$hypertension == hyper[i])
  sample.indices <- sample(row.indices, n.h.hyper[i], replace = F)
  STR.sample.hyper <- rbind(STR.sample.hyper, stroke_data[sample.indices, ])
}

## estimate 1
avg.h.hyper <- tapply(STR.sample.hyper$avg_glucose_level,
                      STR.sample.hyper$hypertension, mean)

avg.var.h.hyper <- tapply(STR.sample.hyper$avg_glucose_level,
                          STR.sample.hyper$hypertension, var)
```

```
avg.se.h.hyper <- sqrt((1 - n.h.hyper/N.h.hyper) * avg.var.h.hyper/n.h.hyper)

avg.str.hyper <- sum(N.h.hyper / N * avg.h.hyper)
avg.se.str.hyper <- sqrt(sum((N.h.hyper / N)^2 * avg.se.h.hyper^2))

## estimate 2
prop.h.hyper <- tapply(STR.sample.hyper$stroke,
                       STR.sample.hyper$hypertension, mean)

prop.se.h.hyper <- sqrt((1 - n.h.hyper/N.h.hyper) * prop.h.hyper*(1-prop.h.hyper)/n.h.hyper)

prop.str.hyper <- sum(N.h.hyper / N * prop.h.hyper)
prop.se.str.hyper <- sqrt(sum((N.h.hyper / N)^2 * prop.se.h.hyper^2))

avg_compare <- rbind(avg_compare, data.frame(avg = avg.str.hyper,
                           SE = avg.se.str.hyper))
prop_compare <- rbind(prop_compare, data.frame(prop = prop.str.hyper,
                            SE = prop.se.str.hyper))
avg_compare
prop_compare

#### method 3: based on `heart_disease` ####
attach(stroke_data)
# check if there are missing values in variable `heart_disease`
sum(is.na(heart_disease))  # 0
N.h.heart <- tapply(avg_glucose_level, heart_disease, length)
N.h.heart
heart <- names(N.h.heart)
heart
N <- sum(N.h.heart) # population size is 5110
detach(stroke_data)

set.seed(1)
n.h.heart <- round((N.h.heart/N) * n)
STR.sample.heart <- NULL
for (i in 1: length(heart))
{
  row.indices <- which(stroke_data$heart_disease == heart[i])
  sample.indices <- sample(row.indices, n.h.heart[i], replace = F)
```

```
  STR.sample.heart <- rbind(STR.sample.heart, stroke_data[sample.indices, ])
}

## estimate 1
avg.h.heart <- tapply(STR.sample.heart$avg_glucose_level,
                      STR.sample.heart$heart_disease, mean)

avg.var.h.heart <- tapply(STR.sample.heart$avg_glucose_level,
                          STR.sample.heart$heart_disease, var)

avg.se.h.heart <- sqrt((1 - n.h.heart/N.h.heart) * avg.var.h.heart/n.h.heart)

avg.str.heart <- sum(N.h.heart / N * avg.h.heart)
avg.se.str.heart <- sqrt(sum((N.h.heart / N)^2 * avg.se.h.heart^2))

## estimate 2
prop.h.heart <- tapply(STR.sample.heart$stroke,
                       STR.sample.heart$heart_disease, mean)

prop.se.h.heart <- sqrt((1 - n.h.heart/N.h.heart) * prop.h.heart*(1-prop.h.heart)/n.h.heart)

prop.str.heart <- sum(N.h.heart / N * prop.h.heart)
prop.se.str.heart <- sqrt(sum((N.h.heart / N)^2 * prop.se.h.heart^2))

avg_compare <- rbind(avg_compare, data.frame(avg = avg.str.heart,
                                             SE = avg.se.str.heart))
prop_compare <- rbind(prop_compare, data.frame(prop = prop.str.heart,
                                               SE = prop.se.str.heart))
avg_compare
prop_compare

#### method 4: based on `ever_married` ####
attach(stroke_data)
# check if there are missing values in variable `ever_married`
sum(is.na(ever_married))  # 0
N.h.married <- tapply(avg_glucose_level, ever_married, length)
N.h.married
married <- names(N.h.married)
married
N <- sum(N.h.married) # population size is 5110
```

```
detach(stroke_data)

set.seed(1)
n.h.married <- round((N.h.married/N) * n)
STR.sample.married <- NULL
for (i in 1: length(married))
{
  row.indices <- which(stroke_data$ever_married == married[i])
  sample.indices <- sample(row.indices, n.h.married[i], replace = F)
  STR.sample.married <- rbind(STR.sample.married, stroke_data[sample.indices, ])
}

## estimate 1
avg.h.married <- tapply(STR.sample.married$avg_glucose_level,
                        STR.sample.married$ever_married, mean)

avg.var.h.married <- tapply(STR.sample.married$avg_glucose_level,
                            STR.sample.married$ever_married, var)

avg.se.h.married <- sqrt((1 - n.h.married/N.h.married) * avg.var.h.married/n.h.married)

avg.str.married <- sum(N.h.married / N * avg.h.married)
avg.se.str.married <- sqrt(sum((N.h.married / N)^2 * avg.se.h.married^2))

## estimate 2
prop.h.married <- tapply(STR.sample.married$stroke,
                         STR.sample.married$ever_married, mean)

prop.se.h.married<-sqrt((1-n.h.married/N.h.married)*prop.h.married*(1-prop.h.married)/n.h.mar
ried)

prop.str.married <- sum(N.h.married / N * prop.h.married)
prop.se.str.married <- sqrt(sum((N.h.married / N)^2 * prop.se.h.married^2))

avg_compare <- rbind(avg_compare, data.frame(avg = avg.str.married,
                                             SE = avg.se.str.married))
prop_compare <- rbind(prop_compare, data.frame(prop = prop.str.married,
                                               SE = prop.se.str.married))
avg_compare
prop_compare
```

```
#### method 5: based on `work_type` ####
attach(stroke_data)
# check if there are missing values in variable `work_type`
sum(is.na(work_type))  # 0
N.h.work <- tapply(avg_glucose_level, work_type, length)
N.h.work
work <- names(N.h.work)
work
N <- sum(N.h.work) # population size is 5110
detach(stroke_data)

set.seed(1)
n.h.work <- round((N.h.work/N) * n)
STR.sample.work <- NULL
for (i in 1: length(work))
{
 row.indices <- which(stroke_data$work_type == work[i])
 sample.indices <- sample(row.indices, n.h.work[i], replace = F)
 STR.sample.work <- rbind(STR.sample.work, stroke_data[sample.indices, ])
}

## estimate 1
avg.h.work <- tapply(STR.sample.work$avg_glucose_level,
                     STR.sample.work$work_type, mean)

avg.var.h.work <- tapply(STR.sample.work$avg_glucose_level,
                         STR.sample.work$work_type, var)

avg.se.h.work <- sqrt((1 - n.h.work/N.h.work) * avg.var.h.work/n.h.work)

avg.str.work <- sum(N.h.work / N * avg.h.work)
avg.se.str.work <- sqrt(sum((N.h.work / N)^2 * avg.se.h.work^2))

## estimate 2
prop.h.work <- tapply(STR.sample.work$stroke,
                      STR.sample.work$work_type, mean)

prop.se.h.work <- sqrt((1 - n.h.work/N.h.work) * prop.h.work*(1-prop.h.work)/n.h.work)
```

```
prop.str.work <- sum(N.h.work / N * prop.h.work)
prop.se.str.work <- sqrt(sum((N.h.work / N)^2 * prop.se.h.work^2))

avg_compare <- rbind(avg_compare, data.frame(avg = avg.str.work,
                                    SE = avg.se.str.work))
prop_compare <- rbind(prop_compare, data.frame(prop = prop.str.work,
                                     SE = prop.se.str.work))
avg_compare
prop_compare

#### method 6: based on `Residence_type` ####
attach(stroke_data)
# check if there are missing values in variable `Residence_type`
sum(is.na(Residence_type))  # 0
N.h.res <- tapply(avg_glucose_level, Residence_type, length)
N.h.res
res <- names(N.h.res)
res
N <- sum(N.h.res) # population size is 5110
detach(stroke_data)

set.seed(1)
n.h.res <- round((N.h.res/N) * n)
STR.sample.res <- NULL
for (i in 1: length(res))
{
  row.indices <- which(stroke_data$Residence_type == res[i])
  sample.indices <- sample(row.indices, n.h.res[i], replace = F)
  STR.sample.res <- rbind(STR.sample.res, stroke_data[sample.indices, ])
}

## estimate 1
avg.h.res <- tapply(STR.sample.res$avg_glucose_level,
                    STR.sample.res$Residence_type, mean)

avg.var.h.res <- tapply(STR.sample.res$avg_glucose_level,
                        STR.sample.res$Residence_type, var)

avg.se.h.res <- sqrt((1 - n.h.res/N.h.res) * avg.var.h.res/n.h.res)
```

```
avg.str.res <- sum(N.h.res / N * avg.h.res)
avg.se.str.res <- sqrt(sum((N.h.res / N)^2 * avg.se.h.res^2))

## estimate 2
prop.h.res <- tapply(STR.sample.res$stroke,
                     STR.sample.res$Residence_type, mean)

prop.se.h.res <- sqrt((1 - n.h.res/N.h.res) * prop.h.res*(1-prop.h.res)/n.h.res)

prop.str.res <- sum(N.h.res / N * prop.h.res)
prop.se.str.res <- sqrt(sum((N.h.res / N)^2 * prop.se.h.res^2))

avg_compare <- rbind(avg_compare, data.frame(avg = avg.str.res,
                                     SE = avg.se.str.res))
prop_compare <- rbind(prop_compare, data.frame(prop = prop.str.res,
                                       SE = prop.se.str.res))
avg_compare
prop_compare

#### method 7: based on `smoking_status` ####
attach(stroke_data)
# check if there are missing values in variable `smoking_status`
sum(is.na(smoking_status))  # 0
N.h.smoking <- tapply(avg_glucose_level, smoking_status, length)
N.h.smoking
smoking <- names(N.h.smoking)
smoking
N <- sum(N.h.smoking) # population size is 5110
detach(stroke_data)

set.seed(1)
n.h.smoking <- round((N.h.smoking/N) * n)
STR.sample.smoking <- NULL
for (i in 1: length(smoking))
{
  row.indices <- which(stroke_data$smoking_status == smoking[i])
  sample.indices <- sample(row.indices, n.h.smoking[i], replace = F)
  STR.sample.smoking <- rbind(STR.sample.smoking, stroke_data[sample.indices, ])
}
```

```
## estimate 1
avg.h.smoking <- tapply(STR.sample.smoking$avg_glucose_level,
                STR.sample.smoking$smoking_status, mean)

avg.var.h.smoking <- tapply(STR.sample.smoking$avg_glucose_level,
                  STR.sample.smoking$smoking_status, var)

avg.se.h.smoking <- sqrt((1 - n.h.smoking/N.h.smoking) * avg.var.h.smoking/n.h.smoking)

avg.str.smoking <- sum(N.h.smoking / N * avg.h.smoking)
avg.se.str.smoking <- sqrt(sum((N.h.smoking / N)^2 * avg.se.h.smoking^2))

## estimate 2
prop.h.smoking <- tapply(STR.sample.smoking$stroke,
                 STR.sample.smoking$smoking_status, mean)

prop.se.h.smoking<-sqrt((1-n.h.smoking/N.h.smoking)*prop.h.smoking*(1-prop.h.smoking)/n.h.
smoking)

prop.str.smoking <- sum(N.h.smoking / N * prop.h.smoking)
prop.se.str.smoking <- sqrt(sum((N.h.smoking / N)^2 * prop.se.h.smoking^2))

avg_compare <- rbind(avg_compare, data.frame(avg = avg.str.smoking,
                                SE = avg.se.str.smoking))
prop_compare <- rbind(prop_compare, data.frame(prop = prop.str.smoking,
                                  SE = prop.se.str.smoking))
avg_compare
prop_compare


rownames(avg_compare) <- c("str_gender", "str_hypertension",
                    "str_heart_disease", "str_ever_married",
                    "str_work_type", "str_Residence_type",
                    "str_smoking_status")

rownames(prop_compare) <- c("str_gender", "str_hypertension",
                    "str_heart_disease", "str_ever_married",
                    "str_work_type", "str_Residence_type",
                    "str_smoking_status")
avg_compare
```

```
# SE of `work_type` is the smallest.
prop_compare
# SE of `ever_married` is the smallest.

############################ Plots ########################################
library(ggplot2)

### Pop distribution
pop_stroke_plot <- ggplot(stroke_data) +
                  geom_bar(aes(x = stroke)) +
                  xlab("Stroke Events") +
                  ylab("Frequency") +
                  ggtitle("Population")

pop_avg_plot <- ggplot(stroke_data) +
                geom_histogram(aes(x = avg_glucose_level)) +
                xlab("Average Glucose Level in Blood") +
                ylab("Frequency") +
                ggtitle("Population")

### SRS distribution
srs_stroke_plot <- ggplot(stroke_srs) +
                  geom_bar(aes(x = stroke)) +
                  xlab("Stroke Events") +
                  ylab("Frequency") +
                  ggtitle("SRS")

srs_avg_plot <- ggplot(stroke_srs) +
                geom_histogram(aes(x = avg_glucose_level)) +
                xlab("Average Glucose Level in Blood") +
                ylab("Frequency") +
                ggtitle("SRS")

### STR distribution
str_stroke_plot <- ggplot(STR.sample.work) +
                  geom_bar(aes(x = stroke)) +
                  xlab("Stroke Events") +
                  ylab("Frequency") +
                  ggtitle("STR")
```

```
str_avg_plot <- ggplot(STR.sample.work) +
                geom_histogram(aes(x = avg_glucose_level)) +
                xlab("Average Glucose Level in Blood") +
                ylab("Frequency")+
                ggtitle("STR")

grid.arrange(pop_avg_plot,
             srs_avg_plot,
             str_avg_plot,
             pop_stroke_plot,
             srs_stroke_plot,
             str_stroke_plot,
             ncol = 3,
             nrow = 2)

############################ Compare SRS with STR ##############################
library(dplyr)
srs_str_avg <- data.frame(avg = c(avg_srs, avg.str.work),
                          SE = c(avg_se_srs, avg.se.str.work))
# added the 95% CI
srs_str_avg <- mutate(srs_str_avg, lower_CI = avg - 1.96*SE,
                                   upper_CI = avg + 1.96*SE)
rownames(srs_str_avg) <- c("SRS", "STR")

srs_str_prop <- data.frame(prop = c(prop_srs, prop.str.work),
                           SE = c(prop_se_srs, prop.se.str.work))
# added the 95% CI
srs_str_prop <- mutate(srs_str_prop, lower_CI = prop - 1.96*SE,
                                     upper_CI = prop + 1.96*SE)
rownames(srs_str_prop) <- c("SRS", "STR")

srs_str_avg
srs_str_prop

avg_pop
prop_pop
```