Implementing Ramdom Sampling Plans

Cheng Peng

Contents

1 Introduction **1**

2 Creating a meaningful and operational stratification variable **1**
2.1 Description of Existing 2-Digit NAICS Codes 2
2.2 Combining Categories 3
2.3 Loan Default Rates By Industry 4
2.4 Study Population 5

3 Sampling Plans **5**
3.1 Simple Random Sampling 5
3.2 Systematic sampling 6
3.3 Stratified Sampling 7

4 Performance Analysis of Random Samples **7**

5 Assignment **8**

1 Introduction

In this note, I will introduce the steps for taking random samples from the study population. The Bank load data set is treated as a population. We will use this data set as a population to implement various sampling plans.

The original data set was split into 9 subsets that are stored on GitHub. We first load these data sets to R and then combine them as a single data set.

```
loan01 = read.csv("https://pengdsci.github.io/datasets/SBAloan/w06-SBAnational01.csv", header = TRUE)[,
loan02 = read.csv("https://pengdsci.github.io/datasets/SBAloan/w06-SBAnational02.csv", header = TRUE)[,
loan03 = read.csv("https://pengdsci.github.io/datasets/SBAloan/w06-SBAnational03.csv", header = TRUE)[,
loan04 = read.csv("https://pengdsci.github.io/datasets/SBAloan/w06-SBAnational04.csv", header = TRUE)[,
loan05 = read.csv("https://pengdsci.github.io/datasets/SBAloan/w06-SBAnational05.csv", header = TRUE)[,
loan06 = read.csv("https://pengdsci.github.io/datasets/SBAloan/w06-SBAnational06.csv", header = TRUE)[,
loan07 = read.csv("https://pengdsci.github.io/datasets/SBAloan/w06-SBAnational07.csv", header = TRUE)[,
loan08 = read.csv("https://pengdsci.github.io/datasets/SBAloan/w06-SBAnational08.csv", header = TRUE)[,
loan09 = read.csv("https://pengdsci.github.io/datasets/SBAloan/w06-SBAnational09.csv", header = TRUE)[,
bankLoan = rbind(loan01, loan02, loan03, loan04, loan05, loan06, loan07, loan08, loan09)
```

2 Creating a meaningful and operational stratification variable

A stratification variable is a categorical variable that can be used to stratify the population based on its values. Each value (or category) defines a subpopulation. When sampling the population, we can take one

random sub-sample from each sub-population and then combine these sub-samples to define a random sample of the population.

There are different ways of defining a stratification variable. For example, we can discretize a numerical variable, use an existing categorical variable, or modify an existing categorical variable by combining some of the categories in a meaningful way, etc. In this note, I use the North American Industry Classification System (NAICS) as an example to show you how to modify an existing categorical variable to define a stratification variable for sampling purposes.

2.1 Description of Existing 2-Digit NAICS Codes

We first summarized the distribution of the existing NAICS in the following table.

```
naics =as.character(bankLoan$NAICS)  # make a character vector
N=length(naics)                      # find the size of the data.
f.table = -sort(-table(naics))       # sort the vector in descending order
n = length(f.table)                  # find the number of distinct industries
n.0 = sum(f.table < 900)             # industry with less than 0.1% of the population size
# A note of length of R variable: the latest version of R has upper bound
# the maximum length of variable names from 256 characters to a whopping 10,000.
# We should try our best to give meaningful names to R variables.
kable(cbind(Population.size = N, Number.of.Industries=n, Sub.Pop.less.900 = n.0))
```

Population.size	Number.of.Industries	Sub.Pop.less.900
899164	1312	1140

I posted an article that used this data for the case study. One of the tables https://github.com/pengdsci/STA490/blob/main/w06/w06-NAICS-Categories.jpg in the article listed categories based on the first digits of NAICS code. The other related table gives the loan default rate in the corresponding industries https://github.com/pengdsci/STA490/blob/main/w06/w06-NAICS-Default-Rates.jpg. You can download these two tables to your local drive and include them in your R Markdown document if you want to practice and reproduce this report.

For the convenience of referring to these tables, I include these two tables in this document.

Next, we explore the frequency distribution of the 2-digit NAICS codes and decide the potential combinations of categories with a small size.

```
NAICS.2.digits = substr(bankLoan$NAICS, 1, 2)   # extract the first two digits of the NAICS code
bankLoan$NAICS2Digit = NAICS.2.digits           # add the above two-digit variable the loan data
ftable = table(bankLoan$NAICS2Digit)
kable(t(ftable))
```

0	11	21	22	23	31	32	33	42	44	45	48	49	51	52	53	54	55	56	61	62	71	72	81	92
201948	8005	1851	663	66646	61809	79368	82848	74874	84737	72514	20310	22211	13794	96136	38170	257	32685	42555	36646	40760	72618	29

Several patterns you observe from the above table:

- 201948 businesses do not have a NAICS code. Since I will use the 2-digit NAICS code to stratify the population. This variable will be included in the study population that will be defined soon.
- Several categories (21, 22, 49, 55, 92) have relatively small sizes. Since categories 48 and 49 are both transportation and warehouse industries, we will combine the two as indicated in the above two tables.

Description of the first two digits of NAICS.

Sector	Description
11	Agriculture, forestry, fishing and hunting
21	Mining, quarrying, and oil and gas extraction
22	Utilities
23	Construction
31–33	Manufacturing
42	Wholesale trade
44–45	Retail trade
48–49	Transportation and warehousing
51	Information
52	Finance and insurance
53	Real estate and rental and leasing
54	Professional, scientific, and technical services
55	Management of companies and enterprises
56	Administrative and support and waste management and remediation services
61	Educational services
62	Health care and social assistance
71	Arts, entertainment, and recreation
72	Accommodation and food services
81	Other services (except public administration)
92	Public administration

Figure 1: List of all industries using the first two digits of the NAICS code

- As we can see from the above two tables, several industries have different codes. We will combine these codes. In other words, we need to modify the 2-digit code to define the final stratification for stratified sampling.

2.2 Combining Categories

We now combine categories suggested in the above NAICS tables. Before we combine the NAICS codes, we present an example to illustrate how to combine categories using R.

```
cate.vec0=c(1,4,3,6,7,3,6,5,4,6,4,5,8,9,4,3,4,7,3)
# vector of category labels
cate.vec=c(1,4,3,6,7,3,6,5,4,6,4,5,8,9,4,3,4,7,3)
# a copy of the vector of category labels
labs.2.collapse = c(1,6,7)  # define a vector to store categories {1,6,7}
logic.vec=cate.vec %in% labs.2.collapse
# TRUE/FALSE ==> match not no-match
cate.vec[logic.vec] = 99  # if matches (i.e., 1, 5, 7), the value
                          # will be replaced by 99
matx=rbind(cate.vec0=cate.vec0, cate.vec=cate.vec)  # check the results
colnames(matx) = 1:length(cate.vec)
# next kable() function requires a column names
kable(matx)
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
cate.vec0	1	4	3	6	7	3	6	5	4	6	4	5	8	9	4	3	4	7	3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
cate.vec	99	4	3	99	99	3	99	5	4	99	4	5	8	9	4	3	4	99	3

We now combine the actual 2-digit NAICS codes

```
cate.31.33=c("31","32","33") # combining categories 31, 32, and 33
cate.48.49 = c("48", "49")
cate.44.45 = c("44", "45")
NAICS2Digit0 = bankLoan$NAICS2Digit  # extract the 2-digit NAICS
NAICS2Digit =  bankLoan$NAICS2Digit  # extract the 2-digit NAICS-copy
## combining 31,32,and 33
logic.31.33=NAICS2Digit %in% cate.31.33 # identify the three categories.
NAICS2Digit[logic.31.33] = 313   # replace 31, 32, 33 with 313
## combining 44 and 45
logic.44.45=NAICS2Digit %in% cate.44.45 # identify the three categories.
NAICS2Digit[logic.44.45] = 445
## combining 48 and 49
logic.48.49=NAICS2Digit %in% cate.48.49  # identify the three categories.
NAICS2Digit[logic.48.49] = 489
bankLoan$strNAICS = NAICS2Digit
```

2.3 Loan Default Rates By Industry

We now find the loan default rates by industry defined by the stratification variable strNAICS. The loan default status can be defined by the variable MIS_Status.

```
x.table = table(bankLoan$strNAICS, bankLoan$MIS_Status)
no.lab = x.table[,1]  # first column consists of unknown default label
default = x.table[,2]
no.default = x.table[,3]
default.rate = round(100*default/(default+no.default),1)
default.status.rate = cbind(no.lab = no.lab,
                            default = default,
                            no.default = no.default,
                            default.rate=default.rate)
kable(default.status.rate)
```

	no.lab	default	no.default	default.rate
0	281	16799	184868	8.3
11	10	812	8183	9.0
21	0	157	1694	8.5
22	1	94	568	14.2
23	154	15463	51029	23.3
313	126	10438	57465	15.4
42	70	9480	39193	19.5
445	276	28868	98107	22.7
489	123	5939	16469	26.5
51	17	2821	8541	24.8
52	26	2692	6778	28.4
53	44	3904	9684	28.7
54	248	12957	54965	19.1
55	1	26	230	10.2
56	156	7661	24868	23.6

	no.lab	default	no.default	default.rate
61	24	1552	4849	24.2
62	102	5736	49528	10.4
71	24	3013	11603	20.6
72	89	14882	52629	22.0
81	223	14229	58166	19.7
92	2	35	192	15.4

2.4 Study Population

Based on the above frequency distribution of the modified 2-digit NAICS codes (the 3-digit codes are combined categories). We use the following inclusion rule to define the study population: excluding small-size categories 20, 21, 55, 92, and unclassified business with NAICS code 0.

```
del.categories = c("0", "21", "22", "55", "92")
# categories to be deleted in
# the original population
del.obs.status = !(bankLoan$strNAICS %in% del.categories)
# deletion status. ! negation operator
study.pop = bankLoan[del.obs.status,]  # excluding the categories
kable(t(table(study.pop$strNAICS))) # Checking correctness operation
```

11	23	313	42	445	489	51	52	53	54	56	61	62	71	72	81
9005	66646	68029	48743	127251	22531	11379	9496	13632	68170	32685	6425	55366	14640	67600	72618

So we have defined our study population!

3 Sampling Plans

In this section, we are implementing three sampling plans. In each sampling plan, we select 4000 observations in the corresponding samples.

3.1 Simple Random Sampling

We define a sampling list and add it to the study population.

```
study.pop$sampling.frame = 1:length(study.pop$GrAppv)
# sampling list
# names(study.pop)
# checking the sampling list variable
sampled.list = sample(1:length(study.pop$GrAppv), 4000)
# sampling the list
SRS.sample = study.pop[sampled.list,]
# extract the sampling units (observations)
## dimension check
dimension.SRS = dim(SRS.sample)
names(dimension.SRS) = c("Size", "Var.count")
kable(t(dimension.SRS))   # checking the sample size
```

Industry default rates (first two digit NAICS codes).

2 digit code	Description	Default rate (%)
21	Mining, quarrying, and oil and gas extraction	8
11	Agriculture, forestry, fishing and hunting	9
55	Management of companies and enterprises	10
62	Health care and social assistance	10
22	Utilities	14
92	Public administration	15
54	Professional, scientific, and technical services	19
42	Wholesale trade	19
31–33	Manufacturing	19, 16, 14
81	Other services (except public administration)	20
71	Arts, entertainment, and recreation	21
72	Accommodation and food services	22
44–45	Retail trade	22, 23
23	Construction	23
56	Administrative/support & waste management/remediation Service	24
61	Educational services	24
51	Information	25
48–49	Transportation and warehousing	27, 23
52	Finance and insurance	28
53	Real estate and rental and leasing	29

Figure 2: List of all industries using the first two digits of the NAICS code and the corresponding loan default rates

Size	Var.count
4000	30

3.2 Systematic sampling

```
jump.size = dim(study.pop)[1]%/%4000
# find the jump size in the systematic sampling
# jump.size
rand.starting.pt=sample(1:jump.size,1) # find the random starting value
sampling.id = seq(rand.starting.pt, dim(study.pop)[1], jump.size)  # sampling IDs
#length(sampling.id)
sys.sample=study.pop[sampling.id,]
# extract the sampling units of systematic samples
sys.Sample.dim = dim(sys.sample)
names(sys.Sample.dim) = c("Size", "Var.count")
kable(t(sys.Sample.dim))
```

Size	Var.count
4013	30

Because the jump size involves rounding error and the population is large, the actual systematic sample size is slightly different from the target size. In this report, I used the integer part of the actual jump size. The actual systematic sampling size is slightly bigger than the target size. We can take away some records random from the systematic sample to make the size to be equal to the target size.

3.3 Stratified Sampling

We take an SRS from each stratum. The sample size should be approximately proportional to the size of the corresponding stratum.

First, we calculate the SRS size for each stratum and then take the SRS from the corresponding stratum.

```
freq.table = table(study.pop$strNAICS)  # frequency table of strNAICS
rel.freq = freq.table/sum(freq.table)   # relative frequency
strata.size = round(rel.freq*4000)      # strata size allocation
strata.names=names(strata.size)         # extract strNAICS names for accuracy checking
```

```
kable(t(strata.size))  # make a nice-looking table using kable().
```

11	23	313	42	445	489	51	52	53	54	56	61	62	71	72	81
52	384	392	281	733	130	66	55	79	393	188	37	319	84	390	418

In the following code chunk, we take stratified samples.

```
strata.sample = study.pop[1,]    # create a reference data frame
strata.sample$add.id = 1   # add a temporary ID to because in the loop
                           # i =2 testing a single iteration
for (i in 1:length(strata.names)){
   ith.strata.names = strata.names[i]   # extract data frame names
   ith.strata.size = strata.size[i]     # allocated stratum size
   # The following code identifies observations to be selected
   ith.sampling.id = which(study.pop$strNAICS==ith.strata.names)
   ith.strata = study.pop[ith.sampling.id,]  # i-th stratified population
   ith.strata$add.id = 1:dim(ith.strata)[1]  # add sampling list/frame
   # The following code generates a subset of random ID
   ith.sampling.id = sample(1:dim(ith.strata)[1], ith.strata.size)
   ## Create a selection status -- pay attention to the operator: %in%
   ith.sample =ith.strata[ith.strata$add.id %in%ith.sampling.id,]
   ## dim(ith.sample)          $ check the sample
   strata.sample = rbind(strata.sample, ith.sample)  # stack all data frame!
 }
 # dim(strata.sample)
 strat.sample.final = strata.sample[-1,]  # drop the temporary stratum ID
 #kable(head(strat.sample.final))         # accuracy check!
```

4 Performance Analysis of Random Samples

Next week, we perform a comparative analysis of the three random samples using the baseline default rates.

5 Assignment

In this assignment, You need to take three random samples as I did in this note:

1. Simple random sample (SRS).
2. Systematic random sample: you need to choose a random starting number.
3. Stratified sample: you need a stratification variable to split the population.
4. Cluster sampling: you need to (a) define a new vector with unique zip codes, (b)take a random sample of zip codes, (c) include all loans in the randomly selected zip codes to get the cluster sample.