Sampling Design and Survey Practice — Lab 3

November 10th, 2021

1 Cluster Sampling

- We will use the `seoul_covid` data set (`http://data.seoul.go.kr/dataList/OA-20279/S/1/datasetView.do`). The data consists of all confirmed cases of COVID-19 in Seoul from January 2020 to October 2021.

```
seoul <- read.csv("seoul_covid.csv", header = TRUE)
head(seoul)
```

```
##       확진일     지역 y cluster
## 1 2021.10.31   강서구 0 2021.10
## 2 2021.10.31 영등포구 0 2021.10
## 3 2021.10.31   타시도 0 2021.10
## 4 2021.10.31 동대문구 0 2021.10
## 5 2021.10.31   구로구 0 2021.10
## 6 2021.10.30   타시도 0 2021.10
```

- We wish to estimate total number of confirmed cases in Gwanak-gu from Jan. 2020 to Oct. 2021, i.e. $\tau = \sum_{i=1}^{N} \sum_{j=1}^{m_i} y_{ij} = \sum_{i=1}^{N} y_{i\cdot}$, where we use the conventional notations from the lecture note.
- We will use the cluster sampling, where we regard each month as a cluster. Thus, the data set consists of $N = 22$ clusters.
- Choose $n = 11$ clusters by SRS, and observe **all** elements in each selected cluster.

✓ What we know:

- $N = 22$, $n = 11$
- m_i (number of elements in cluster i)

2020.1.	2020.2.	2020.3.	2020.4.	2020.5.	2020.6.	2020.7.	2020.8.	2020.9.	2020.10	2020.11
7	80	391	156	229	459	281	2415	1306	733	2904
2020.12	2021.1.	2021.2.	2021.3.	2021.4.	2021.5.	2021.6.	2021.7.	2021.8.	2021.9.	2021.10
10432	4878	4060	3897	5803	6030	6258	14504	15193	21383	18840

- $M = 120239$ (the number of elements in the population)
- $\bar{M} = 5465.409$ (the average cluster size of the population)

```
N <- length(unique(seoul$cluster))
n <- 11
m_i <- table(seoul$cluster)
M <- sum(table(seoul$cluster))
M_bar <- mean(table(seoul$cluster))
```

1.1 Estimating Population Mean and Population Total

- Sample 11 clusters using SRS. Note that `seoul_cluster` dataframe has the information of N, but does not know M and $\bar{M}$.

```
seoul$fpc <- rep(N, M)
set.seed(0)
index <- sample(unique(seoul$cluster),n)
seoul_cluster <- seoul[seoul$cluster %in% index,]
```

- Estimation of population mean μ:

$$\bar{y}_{\text{cl}} = \frac{\sum_{i=1}^{n} y_{i\cdot}}{\sum_{i=1}^{n} m_i},$$
$$\widehat{\text{Var}}(\bar{y}_{\text{cl}}) = \frac{N-n}{Nn\bar{m}^2} s_c^2$$

  Since $\bar{M}$ is unknown, $\bar{m} = \frac{1}{n}\sum_{i=1}^{n} m_i$ is used instead in estimating the variance of $\bar{y}_{\text{cl}}$.
- `id` argument must be specified using a variable that uniquely identifies each cluster.
- `fpc` argument is the number of clusters instead of the number of elements in the population, i.e. N.

```
library(survey)
onestage <- svydesign(id = ~cluster, data = seoul_cluster, fpc = ~fpc)
summary(onestage)
```

```
## 1 - level Cluster Sampling design
## With (11) clusters.
## svydesign(id = ~cluster, data = seoul_cluster, fpc = ~fpc)
## Probabilities:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.5     0.5     0.5     0.5     0.5     0.5
## Population size (PSUs): 22
## Data variables:
## [1] "확진일"  "지역"    "y"       "cluster" "fpc"
```

```
mu_est <- svymean(~y, design=onestage)
mu_est
```

```
##       mean     SE
## y 0.056507 0.0043
```

```
confint(mu_est)
```

```
##        2.5 %     97.5 %
## y 0.04812718 0.06488614
```

```
mean(seoul$y) # true mean
```

```
## [1] 0.0523707
```

- Estimation of population total τ when M is known:

$$\hat{\tau} = M\bar{y}_{\text{cl}},$$
$$\widehat{\text{Var}}(\hat{\tau}) = M^2\widehat{\text{Var}}(\bar{y}_{\text{cl}})$$

```
total_est2 <- mu_est[1] * M
total_est2
```

```
##        y
## 6794.305
```

```
confint(mu_est) * M
```

```
##      2.5 %   97.5 %
## y 5786.764 7801.845
```

```
sum(seoul$y) # true total
```

```
## [1] 6297
```

- Estimation of population total τ when M is unknown:

$$\hat{\tau} = N\bar{y}_{\text{t}} = \frac{N}{n}\sum_{i=1}^{n} y_{i\cdot},$$
$$\widehat{\text{Var}}(\hat{\tau}) = N^2\widehat{\text{Var}}(\bar{y}_{\text{t}}) = N^2\left(\frac{N-n}{Nn}\right)s_t^2$$

```
total_est1 <- svytotal(~y, design=onestage)
total_est1
```

```
##   total     SE
## y  9288 2217.8
```

```
confint(total_est1)
```

```
##      2.5 %   97.5 %
## y 4941.247 13634.75
```

```
sum(seoul$y) # true total
```

```
## [1] 6297
```

1.2 Cluster Sampling with Probability Proportional to Size (PPS)

- Suppose we sample each cluster (month) with probability proportional to its size (number of cases in that month). That is, the probability of including the cluster i in the sample is given as $\pi_i = n\frac{m_i}{M}$.

```r
set.seed(0)
cluster_size <- m_i[unique(seoul$cluster)]
seoul$pi <- n*cluster_size[seoul$cluster]/M
index <- sample(unique(seoul$cluster),n,prob=cluster_size)
seoul_cluster_pps <- seoul[seoul$cluster %in% index,]
```

- We indicate the use of pps using `probs` variable.

```r
onestage_pps <- svydesign(id = ~cluster, data = seoul_cluster_pps, probs = ~pi)
svymean(~y, design = onestage_pps) |> confint()
```

```
##        2.5 %    97.5 %
## y 0.03914709 0.0656562
```

```r
svytotal(~y, design = onestage_pps) |> confint()
```

```
##      2.5 %   97.5 %
## y 4707.007 7894.436
```

2 Two-Stage Cluster Sampling

- Instead of inspecting **all** elements in the cluster, we will **sample** elements from each cluster.
- Let the primary sampling unit be the month, and the secondary sampling unit be the date.
- Two-stage cluster sampling design requires variables to identify the primary and secondary sampling units, the total number of clusters in the population (N), and the total number of elements in each selected cluster (M_i).
- `id` is given as a **formula** of variables that uniquely identify each primary sampling unit and secondary sampling unit.
- `fpc1` and `fpc2` indicate the number of clusters in the population (`fpc1`= N), and the total dates in each sampled month (`fpc2`= M_i), respectively.

```r
colnames(seoul)[5] <- "fpc1"
ssu_length <- tapply(seoul$확진일, seoul$cluster, \(cl) length(unique(cl)))
seoul$fpc2 <- ssu_length[seoul$cluster]
set.seed(1)
#primary sampling
index1 <- sample(unique(seoul$cluster),n)
seoul_ps <- seoul[seoul$cluster %in% index1,]
#secondary sampling
m <- 15 # m_1 = ... = m_11 = 15
index2 <- tapply(seoul_ps$확진일, seoul_ps$cluster, \(cl) sample(unique(cl), m))
seoul_ss <- seoul_ps[seoul_ps$확진일 %in% unlist(index2),]
```

```r
twostage <- svydesign(id = ~cluster + 확진일, data = seoul_ss, fpc = ~fpc1 + fpc2)
summary(twostage)
```

```
## 2 - level Cluster Sampling design
## With (11, 165) clusters.
## svydesign(id = ~cluster + 확진일, data = seoul_ss, fpc = ~fpc1 +
##     fpc2)
## Probabilities:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2419  0.2419  0.2419  0.2454  0.2500  0.3261
## Population size (PSUs): 22
## Data variables:
## [1] "확진일"  "지역"    "y"       "cluster" "fpc1"    "pi"      "fpc2"
```

- Estimation of τ:

$$\hat{\tau} = \frac{N}{n} \sum_{i=1}^{n} M_i \bar{y}_i$$

```r
total_est <- svytotal(~y, design = twostage)
total_est
```

```
##    total     SE
## y 9523.6 2130.1
```

```r
confint(total_est)
```

```
##     2.5 %   97.5 %
## y 5348.65 13698.55
```

```r
sum(seoul$y) # true total
```

```
## [1] 6297
```

- Estimation of μ when M is unknown:

$$\hat{\mu} = \frac{\sum_{i=1}^{n} M_i \bar{y}_i}{\sum_{i=1}^{n} M_i} \quad (\text{✗})$$

- Here, we are measuring mean number of positive people in Gwanak-gu per day (secondary sampling unit). However, using the function **svymean**, mean number of positive people in Gwanak-gu per case is estimated.

```r
mu_est1 <- svymean(~y, design = twostage)
mu_est1
```

```
##       mean     SE
## y 0.053596 0.0045
```

```r
confint(mu_est1)
```

```
##        2.5 %     97.5 %
## y 0.04484944 0.06234222
```

```
mean(tapply(seoul$y, seoul$확진일, sum)) # true mean
```

```
## [1] 10.23902
```

- Estimation of μ when M is known:

$$\bar{y} = \frac{\hat{\tau}}{M}$$

```
M <- length(unique(seoul$확진일))
mu_est2 <- total_est[[1]]/M
mu_est2
```

```
## [1] 15.48553
```

```
confint(total_est)/M
```

```
##      2.5 %   97.5 %
## y 8.696991 22.27407
```

```
mean(tapply(seoul$y, seoul$확진일, sum)) # true mean
```

```
## [1] 10.23902
```