Sampling Design and Survey Practice Lab #1

TA - Seungkyu Kim

2022-09-28

0. Packages and data setting

해당 패키지들과 데이터는 앞으로도 계속 쓰일 예정으로, 아래 코드는 매번 돌리고 시작한다고 보면 된다.

Install and load packages

```
name_pkg <- c("survey", "foreign")
name_pkg <- unique(name_pkg)

bool_nopkg <- !name_pkg %in% rownames(installed.packages())
if (sum(bool_nopkg) > 0) {
  install.packages(name_pkg[bool_nopkg])
}
invisible(lapply(name_pkg, library, character.only = T))
```

```
## Loading required package: grid

## Loading required package: Matrix

## Loading required package: survival

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart
```

Load data - API

API는 Academic Performance Index의 줄임말로, api 데이터셋은 California에 위치한 모든 학교에 대한 API 지수 정보가 담겨있다.

```
?api
```

위 코드를 입력하면 api 데이터셋의 모든 변수에 대한 정보를 열람할 수 있다. 이번 시간에 쓰일 몇가지 변수들에 대한 설명을 해보자면,

- enroll : 해당 학교에 등록된 학생 수
- api00, api99 : 2000년, 1999년의 API 지수
- stype : 학교 구분. 초등(E), 중등(M), 고등(H) 세 가지의 label이 존재.
- emer : emergency certification를 가진 교사의 비율 (우리나라로 치면 기간제 교사)

등이 있다.

```
data(api)
```

위 코드를 돌리면 apipop, apisrs, apistrat, apiclus1, apiclus2 데이터셋을 불러오게 된다. apipop 데이터에는 모든 학교(6194 obsevations)에 대한 정보가 담겨있고, apisrs와 apistrat는 각각 다른 방식으로 200개 학교를 샘플링한 데이터셋이다. 여기서 srs는 단순임의표집(simple random sampling), strat는 층화임의표집(stratified random sampling)을 의미한다. 이번 시간에는 apisrs, apistrat 두 데이터만 사용한다.

```
sum(apisrs$stype == 'E'); sum(apisrs$stype == 'M'); sum(apisrs$stype == 'H')
```

```
## [1] 142
```

```
## [1] 33
```

```
## [1] 25
```

```
sum(apistrat$stype == 'E'); sum(apistrat$stype == 'M'); sum(apistrat$stype == 'H')
```

```
## [1] 100
```

```
## [1] 50
```

```
## [1] 50
```

위 코드를 통해 층화임의표집을 시행한 데이터(apistrat)는 초등,중등,고등학교에서 각각 100,50,50개의 계획된 샘플 수를 사용하였음을 알 수 있다.

Load data - CHIS

뒷부분(7페이지)에서 해당 데이터를 사용할 때 설명하겠다.

1. 단순임의표집 (Simple random sampling)

```
srs_design <- svydesign(id=~1, fpc=~fpc, data=apisrs) #fpc : finite population correction
srs_design
```

```
## Independent Sampling design
## svydesign(id = ~1, fpc = ~fpc, data = apisrs)
```

```
svytotal(~enroll, srs_design)
```

```
##          total     SE
## enroll 3621074 169520
```

```
svymean(~enroll, srs_design)
```

```
##          mean     SE
## enroll 584.61 27.368
```

위 코드는 apisrs가 단순임의표집 된 데이터임을 가정했을 때 모합과 모평균을 추정하는 코드이다. apisrs 데이터에는 fpc라는 변수가 있는데,

```
summary(apisrs$fpc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6194    6194    6194    6194    6194    6194
```

보시다시피 전부 6194로 입력되어있다. 단순임의표집에서 모합이나 모분산을 추정하려면 population size N을 알아야하는데, 이 정보가 입력되어 있는거라고 보면 된다.

fpc는 finite population correction을 의미하는데, 그냥 수업시간에 배운 단순임의표집 공식을 적용하는 변수라고 보면 된다. 왜 저런 이름을 가지는지 궁금한 학생은 https://www.statisticshowto.com/finite-population-correction-factor/ 을 참고하기 바란다.

강의노트에 있는 $s.e.(\bar{Y}) = \sqrt{\frac{\sigma^2}{n}\frac{N-n}{N-1}}$ 공식을 실제로 계산해보면,

```
N = 6194; n = 200; s2 = var(apisrs$enroll)
se = sqrt((s2 / n) * ((N-n) / N)) # Note that N instead of N-1
se
```

```
## [1] 27.36837
```

으로 svymean 함수 부분에서의 SE와 같은 값으로 계산됨을 확인할 수 있다. 다만 N-1 대신에 N으로 나눠야 같은 값이 나오는데 왜 이렇게 처리했는지는 의문이다. N이 충분히 크면 N으로 나눠도 근사가 가능하다.

(복원추출)

```
nofpc <- svydesign(id=~1, weights=~pw, data=apisrs)
nofpc
```

```
## Independent Sampling design (with replacement)
## svydesign(id = ~1, weights = ~pw, data = apisrs)
```

```
svytotal(~enroll, nofpc)
```

```
##          total     SE
## enroll 3621074 172325
```

```
svymean(~enroll, nofpc)
```

```
##          mean     SE
## enroll 584.61 27.821
```

위 코드는 단순임의표집 시 복원추출을 하였다는 가정하에 모합과 모평균을 추정하는 코드이다.

```
summary(apisrs$pw)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   30.97   30.97   30.97   30.97   30.97   30.97
```

복원추출 가정 시에는 apisrs 데이터에서 pw라는 변수를 사용하는데, $30.97 = 6194/200$이다. 왜 이런 값을 입력해야하는지 자세한 설명은 뒤에서 설명하겠다.

```
se_nofpc = sqrt(s2 / n)
se_nofpc
```

```
## [1] 27.82121
```

$s.e.(\bar{Y}) = \sqrt{\frac{\sigma^2}{n}}$를 계산해보면 역시 모평균 추정에서의 SE 결과값과 같은 값이 나옴을 알 수 있다.

(추가예제)

1999년과 2000년 간의 API 지수 변동(평균치)에 대한 추정을 해보자.

```
means <- svymean(~api00+api99, srs_design)
means
```

```
##         mean     SE
## api00 656.58 9.2497
## api99 624.68 9.5003
```

```
# The next two lines give the same result.

svycontrast(means, c(api00=1, api99=-1))
```

```
##          contrast     SE
## contrast     31.9 2.0905
```

```
svycontrast(means, quote(api00-api99))
```

```
##          nlcon     SE
## contrast  31.9 2.0905
```

update 함수를 통해 svydesign 함수에서 나온 결과물(여기선 srs_design)에 새로운 변수들을 추가할 수 있다.

```
srs_design <- update(srs_design, apidiff=api00-api99)
srs_design <- update(srs_design, apidct=apidiff/api99)
svymean(~apidiff+apidct, srs_design)
```

```
##              mean     SE
## apidiff 31.900000 2.0905
## apidct   0.056087 0.0041
```

2. 층화임의표집 (Stratified sampling)

이번에는 층화임의표집된 데이터를 사용한다는 가정 하에 모합과 모평균을 추정해보자. 여기선 apistrat 데이터를 사용하고, stype 변수에 의해 층(초등/중등/고등학교)이 나뉘어져 있다고 가정한다.

```
strat_design <- svydesign(id=~1, strata=~stype, fpc=~fpc, data=apistrat)
strat_design
```

```
## Stratified Independent Sampling design
## svydesign(id = ~1, strata = ~stype, fpc = ~fpc, data = apistrat)
```

```
svytotal(~enroll, strat_design)
```

```
##          total     SE
## enroll 3687178 114642
```

```
svymean(~enroll, strat_design)
```

```
##          mean     SE
## enroll 595.28 18.509
```

18.509는 단순임의표집에서 모평균 추정 시 SE 였던 27.368보다 작은 값이다. 즉, 할수만 있다면 층화표집이 단순표집보다 더 효율이 좋다.

[단순임의표집 때와 마찬가지로 svymean(~enroll, strat_design) 결과의 SE 값과 실제 계산값이 같은지 비교해보자.]

전과 마찬가지로 fpc 변수를 확인해보면, 층마다 4421, 1018, 755라는 값으로 입력되어 있음을 알 수 있다. 이는 각 층의 population size N_1, N_2, N_3를 의미하게 된다.

```
summary(as.factor(apistrat$fpc))
```

```
##  755 1018 4421
##   50   50  100
```

```
svytotal(~stype, strat_design)
```

```
##        total SE
## stypeE  4421  0
## stypeH   755  0
## stypeM  1018  0
```

svytotal(~stype, strat_design) 명령을 통해 전체 population 에 'E', 'M', 'H' 가 몇개가 있는지 추정해볼 수 있는데, fpc 변수에서 각 층마다 모집단의 크기에 대한 정보를 제공했으므로 당연히 실제값이 그대로 나오게 된다.

3. Sampling weights and replicate weights

Sampling weights

Sampling weights are the number of individuals in the population each respondent in the sample is representing.

예를 들어 50,000,000명의 인구 중 5,000명을 단순임의표집하여 설문조사를 시행한다면, 모든 샘플 1개 당 10,000명의 정보를 담고 있다고 생각할 수 있을 것이다. 이 때 모든 인구 개개인(indivisual)에 해당하는 sampling weight를 10000이라고 정의한다.

층화임의표집된 데이터인 apistrat에서 각 샘플의 sampling weight를 보면,

```
summary(apistrat[apistrat$stype == 'E',]$pw)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   44.21   44.21   44.21   44.21   44.21   44.21
```

```
summary(apistrat[apistrat$stype == 'M',]$pw)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.36   20.36   20.36   20.36   20.36   20.36
```

```
summary(apistrat[apistrat$stype == 'H',]$pw)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    15.1    15.1    15.1    15.1    15.1    15.1
```

초등학교에서 표집된 샘플은 $44.21 = 4421/100$, 중학교에서 표집된 샘플은 $20.36 = 1018/50$, 고등학교에서 표집된 샘플은 $15.1 = 755/50$의 값이 주어져있다. 층화표집되었을 경우 각 층마다 다른 sampling weight를 가짐을 알 수 있고, 각 값이 어떻게 계산되었는지는 쉽게 알 수 있을것이다.

다음 데이터는 California Health Interview Survey(CHIS)에서 2005년에 조사한 보건 데이터이다. 기관에서는 각 조사대상의 특징(소득, 거주지역, 인종 등)을 고려하여 각 샘플마다 sampling weight를 부여한다. 이는 데이터의 rakedw0 변수를 통해 확인할 수 있다.

```
chis_adult <- read.dta('adult.dta')
summary(chis_adult$rakedw0)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     4.0   183.0   402.0   613.4   773.0  8705.0
```

replicate weight

- BRR : 첨부파일 참조
- Bootstrap : 표집된 샘플에서 같은 size만큼 복원 추출을 통해 새로운 샘플을 여러 개 만든다. (첨부파일 그림 참조)

```
strat_design_2 <- svydesign(id=~1, strata=~stype, weights=~pw, data=apistrat)
# BRR is not applicable to sampling design with fpc applied.

boot_design <- as.svrepdesign(strat_design, type='bootstrap', replicates=100)
BRR_design <- as.svrepdesign(strat_design_2, type='BRR')

boot_design
```

```
## Call: as.svrepdesign.default(strat_design, type = "bootstrap", replicates = 100)
## Survey bootstrap with 100 replicates.
```

```
BRR_design
```

```
## Call: as.svrepdesign.default(strat_design_2, type = "BRR")
## Balanced Repeated Replicates with 104 replicates.
```

```
svymean(~enroll, strat_design)
```

```
##          mean     SE
## enroll 595.28 18.509
```

```
svymean(~enroll, boot_design)
```

```
##          mean     SE
## enroll 595.28 18.538
```

```
svymean(~enroll, BRR_design)
```

```
##          mean     SE
## enroll 595.28 19.675
```

chis 데이터에선 조사기관에서 80개의 replicate weights를 추가로 부여하였다 (데이터에 대한 자세한 설명은 첨부파일 참조). rakedw1 ~ rakedw80 변수 혹은 chis_adult[,420:499] 를 통해 확인해 볼 수 있다. 이를 이용하면 BRR이나 bootstrap 같이 표집된 샘플에서 데이터를 복제하지 않아도 variance의 추정이 가능해진다.

```
chis <- svrepdesign(variables=chis_adult[,1:418],
                    repweights=chis_adult[,420:499],
                    weights=chis_adult[,419], combined.weights=TRUE,
                    type='other', scale=1, rscales=1)
```

4. 부-모집단에서의 추정 (Estimates in subpopulations)

전체 교사 중 emergency certification을 받은 교사의 비율을 emer 변수에 저장하였다고 언급한 바 있다. 아래 코드는 기존에 층화임의표집된 설문결과(strat_design)로부터 이 비율이 20%를 넘는 부-모집단, 0%인 부-모집단에 대한 몇 가지 추정치를 구하는 과정이다.

```
emerg_high <- subset(strat_design, emer>20)
emerg_low <- subset(strat_design, emer==0)

svymean(~api00+api99, emerg_high)
```

```
##         mean     SE
## api00 558.52 21.708
## api99 523.99 21.584
```

```
svymean(~api00+api99, emerg_low)
```

```
##         mean     SE
## api00 749.09 17.516
## api99 720.07 19.061
```

```
svytotal(~enroll, emerg_high)
```

```
##         total     SE
## enroll 762132 128674
```

```
svytotal(~enroll, emerg_low)
```

```
##         total    SE
## enroll 461690 75813
```

[이제 단순임의표집된 설문결과(srs_design)로부터 두 부-모집단에서의 api00 값의 모평균을 추정해보고, 결과의 SE 값과 수업시간에 배운 공식을 이용하여 직접 계산한 결과를 비교해보자.]

svyby 명령어를 사용하면 각 층(stratum)을 부-모집단으로 취급하여 모든 층에 대한 추정치를 한번에 계산하여 출력해준다.

```
bys <- svyby(~bmi_p, ~srsex+racehpr, svymean, design=chis, keep.names=FALSE)
print(bys, digits=3)
```

```
##     srsex                          racehpr bmi_p     se
## 1    MALE                           LATINO  28.2 0.1447
## 2  FEMALE                           LATINO  27.5 0.1443
## 3    MALE                 PACIFIC ISLANDER  29.7 0.7055
## 4  FEMALE                 PACIFIC ISLANDER  27.8 0.9746
## 5    MALE AMERICAN INDIAN/ALASKAN NATIVE  28.8 0.5461
## 6  FEMALE AMERICAN INDIAN/ALASKAN NATIVE  27.0 0.4212
## 7    MALE                             ASIAN  24.9 0.1406
## 8  FEMALE                             ASIAN  23.0 0.1112
## 9    MALE                 AFRICAN AMERICAN  28.0 0.2663
```

```
## 10 FEMALE               AFRICAN AMERICAN  28.4 0.2417
## 11   MALE                          WHITE  27.0 0.0598
## 12 FEMALE                          WHITE  25.6 0.0680
## 13   MALE     OTHER SINGLE/MULTIPLE RACE  26.9 0.3742
## 14 FEMALE     OTHER SINGLE/MULTIPLE RACE  26.7 0.3158
```