Sampling Design and Survey Practice Lab #4

TA - Seungkyu Kim

2022-11-23

Install and load packages

```
name_pkg <- c("survey", "sampling", "SDAResources")
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))
```

1. 일단집락표집 (One-Stage Cluster Sampling) - 집락 당 ssu 크기 같을 때

gpa 데이터는 총 100개의 집락 중 5개의 집락에 대해 일단집락표집을 시행한 데이터이다.

```
data(gpa)
# define one-stage cluster design
# note that id is suite instead of individual student as we take an SRS of suites
dgpa<-svydesign(id=~suite,weights=~wt,fpc=~rep(100,20),data=gpa)
dgpa
```

```
## 1 - level Cluster Sampling design
## With (5) clusters.
## svydesign(id = ~suite, weights = ~wt, fpc = ~rep(100, 20), data = gpa)
```

표집된 데이터들의 sampling weight는 100/5 = 20 으로 되어있고, svydesign 명령어 적용 시 fpc 값은 전부 100으로 설정했음을 알 수 있다. (모집단의 크기(총 원소수)가 다닌 총 집락수로 설정한 것을 알 수 있다.)

또한 이전에 단순임의표집이나 층화임의표집을 하였을때는 id = ~1 로 지정하였는데, 이번에는 suite 변수를 따라가게끔 지정이 되어있다. 즉, survey 명령어에서 id 옵션은 집락표집을 위한 것임을 알 수 있다. (임의표집에서 id = ~1 은 모든 샘플을 1번 집락, 즉 하나의 집락으로 간주한다는 뜻이고 이는 집락표집을 시행하지 않았음을 의미하게 된다.)

```
# estimate mean and se
gpamean<-svymean(~gpa,dgpa)
gpamean
```

```
##      mean     SE
## gpa 2.826 0.1637
```

```
degf(dgpa)
```

```
## [1] 4
```

```
# n=5, t-approximation is suggested for CI
confint(gpamean,level=.95,df=4) # use t-approximation
```

```
##        2.5 %   97.5 %
## gpa 2.371593 3.280407
```

```
# confint(gpamean,level=.95) # uses normal approximation, if desired (for large n)
# estimate total and se (if desired)
gpatotal<-svytotal(~gpa,dgpa)
gpatotal
```

```
##      total     SE
## gpa 1130.4 65.466
```

```
confint(gpatotal,level=.95,df=4)
```

```
##        2.5 %   97.5 %
## gpa 948.6374 1312.163
```

나머지는 기존에 해왔던 것들과 크게 다르지 않다.

아래는 수업시간에 배웠던 공식을 사용하여 추정량의 표준편차를 직접 구하는 과정이다.

```
# you can also calculate SEs by direct formula
suitesum<-tapply(gpa$gpa,gpa$suite,sum)  #sum gpa for each suite
# variability comes from among the suites
st2<-var(suitesum)
st2
```

```
## [1] 2.25568
```

```
# SE of t-hat, formula (5.3) of SDA
vthat <-100^2*(1-5/100)*st2/5
sqrt(vthat)
```

```
## [1] 65.46596
```

```
# SE of ybar, formula (5.6) of SDA
sqrt(vthat)/(4*100)
```

```
## [1] 0.1636649
```

2. 일단집락표집 (One-Stage Cluster Sampling) - 집락 당 ssu 크기 다를 때

algebra 데이터는 187개 class(집락의 역할을 한다) 중 12개 class에 대해 표집을 시행한 데이터이다.

```
data(algebra)
colnames(algebra)
```

```
## [1] "class" "Mi"    "score"
```

이번엔 gpa 데이터와 다르게 각 집락 당 ssu의 갯수가 다르기 때문에, Mi 변수에 이에 대한 정보를 추가로 정리해둔 것을 알 수 있다.

```
nrow(algebra)
```

```
## [1] 299
```

```
algebra$sampwt<-rep(187/12,299)
# define one-stage cluster design
dalg<-svydesign(id=~class,weights=~sampwt,fpc=~rep(187,299), data=algebra)
dalg
```

```
## 1 - level Cluster Sampling design
## With (12) clusters.
## svydesign(id = ~class, weights = ~sampwt, fpc = ~rep(187, 299),
##     data = algebra)
```

```
# estimate mean and se
svymean(~score,dalg)
```

```
##         mean     SE
## score 62.569 1.4916
```

```
# n=12, t-distribution is suggested for CI
degf(dalg)
```

```
## [1] 11
```

```
confint(svymean(~score,dalg),level=.95,df=11) #use t-approximation
```

```
##          2.5 %  97.5 %
## score 59.28562 65.8515
```

```
# estimate total and se if desired
svytotal(~score,dalg)
```

```
##        total    SE
## score 291533 19893
```

```
confint(svytotal(~score,dalg),level=.95,df=11)
```

```
##          2.5 %   97.5 %
## score 247749.4 335316.6
```

3. 이단집락표집 (Two-Stage Cluster Sampling)

coots 데이터는 184개의 coot nests 에서 egg 2개씩을 조사한 데이터이다. 즉 nest가 집락이 되고, 그 안에 들어있는 egg가 ssu가 된다. 그리고 그 중 2개씩만을 조사하였으니 이단집락표집이 된다.

```
data(coots)
nrow(coots) #368
```

```
## [1] 368
```

```
colnames(coots)
```

```
## [1] "clutch"  "csize"   "length"  "breadth" "volume"  "tmt"
```

여기서 clutch 변수는 집락의 index, csize 변수는 각 집락에 속해있는 ssu의 갯수 (즉, 각 nest에 들어있던 egg의 갯수) 를 의미한다.

```
coots$ssu<-rep(1:2,184) # index of ssu
coots$relwt<-coots$csize/2
head(coots)
```

```
##   clutch csize length breadth    volume tmt ssu relwt
## 1      1    13  44.30   31.10 3.7957569   1   1   6.5
## 2      1    13  45.90   32.70 3.9328497   1   2   6.5
## 3      2    13  49.20   34.40 4.2156036   1   1   6.5
## 4      2    13  48.70   32.70 4.1727621   1   2   6.5
## 5      3     6  51.05   34.25 0.9317646   0   1   3.0
## 6      3     6  49.35   34.40 0.9007362   0   2   3.0
```

```
dcoots<-svydesign(id=~clutch+ssu,weights=~relwt,data=coots)
dcoots
```

```
## 2 - level Cluster Sampling design (with replacement)
## With (184, 368) clusters.
## svydesign(id = ~clutch + ssu, weights = ~relwt, data = coots)
```

```
svymean(~volume,dcoots)  #ratio estimator
```

```
##          mean    SE
## volume 2.4908 0.061
```

```
confint(svymean(~volume,dcoots),level=.95,df=183)
```

```
##           2.5 %   97.5 %
## volume 2.370423 2.611134
```

id 옵션에 clutch+ssu 를 입력함으로써 이단집락표집으로 간주시킬 수 있게 된다.

coots 데이터의 특징으로는 총 집락의 개수가 unknown 이라는 것이다. 따라서 svydesign 명령어 적용 시 fpc 옵션을 입력하지 않았음을 알 수 있고, 이는 집락 표집 시 복원추출을 사용하였다는 가정 하에 분석을 진행하였다는 뜻이 된다.

아래는 schools 데이터를 이용하여 이단집락표집 분석을 시행하는 과정이다. 이 때는 전체 school (집락) 수가 75임이 알려져 있다. 복원추출과 비복원추출을 가정하였을 때 각각의 코드가 담겨져 있으니 차이점을 확인해보기 바란다.

```
### With-replacement variance

data(schools)
head(schools)
```

```
##   schoolid gender math reading mathlevel readlevel  Mi finalwt
## 1        9      F   42      42         2         2 163  61.125
## 2        9      F   29      30         1         1 163  61.125
## 3        9      M   31      25         1         1 163  61.125
## 4        9      F   22      33         1         2 163  61.125
## 5        9      M   35      36         1         2 163  61.125
## 6        9      F   30      17         1         1 163  61.125
```

```
# calculate with-replacement variance; no fpc argument
# include psu variable in id; include weights
dschools<-svydesign(id=~schoolid,weights=~finalwt,data=schools)
# dschools tells you this is treated as a with-replacement sample
dschools
```

```
## 1 - level Cluster Sampling design (with replacement)
## With (10) clusters.
## svydesign(id = ~schoolid, weights = ~finalwt, data = schools)
```

```
mathmean<-svymean(~math,dschools)
mathmean
```

```
##        mean     SE
## math 33.123 1.7599
```

```
degf(dschools)
```

```
## [1] 9
```

```
# use t distribution for confidence intervals because there are only 10 psus
confint(mathmean,df=degf(dschools))
```

```
##         2.5 %  97.5 %
## math 29.14179 37.1041
```

```
# estimate proportion and total number of students with mathlevel=2
svymean(~factor(mathlevel),dschools)
```

```
##                         mean     SE
## factor(mathlevel)1 0.71231 0.0542
## factor(mathlevel)2 0.28769 0.0542
```

```
svytotal(~factor(mathlevel),dschools)
```

```
##                        total      SE
## factor(mathlevel)1 12303.4 2244.14
## factor(mathlevel)2  4969.1  676.26
```

```
### Without-replacement variance

# create a variable giving each student an id number
schools$studentid<-1:(nrow(schools))
# calculate without-replacement variance
# specify both stages of the sample in the id argument
# give both sets of population sizes in the fpc argument
# do not include the weight argument
dschoolwor<-svydesign(id=~schoolid+studentid,fpc=~rep(75,nrow(schools))+Mi,
                      data=schools)
dschoolwor
```

```
## 2 - level Cluster Sampling design
## With (10, 200) clusters.
## svydesign(id = ~schoolid + studentid, fpc = ~rep(75, nrow(schools)) +
##     Mi, data = schools)
```

```
mathmeanwor<-svymean(~math,dschoolwor)
mathmeanwor
```

```
##        mean     SE
## math 33.123 1.6605
```

```
confint(mathmeanwor,df=degf(dschoolwor))
```

```
##         2.5 %   97.5 %
## math 29.36667 36.87923
```

```
# estimate proportion and total number of students with mathlevel=2
svymean(~factor(mathlevel),dschoolwor)
```

```
##                         mean     SE
## factor(mathlevel)1 0.71231 0.0516
## factor(mathlevel)2 0.28769 0.0516
```

```
svytotal(~factor(mathlevel),dschoolwor)
```

```
##                        total      SE
## factor(mathlevel)1 12303.4 2097.83
## factor(mathlevel)2  4969.1  657.69
```