Sampling Design and Survey Practice Lab #5

TA - Seungkyu Kim

2022-11-30

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))
```

오늘 실습 시간에는 전체 모집단에서 각 데이터에 unequal probabilities 를 부여하여 (집락) 샘플링을 하는 법에 대해 살펴보고(1,2절), 역으로 표집된 데이터에 unequal probabilities 가정을 이용하여 추정량을 구하는 방법을 살펴본다(3절).

1. Sampling with Unequal Probabilities in One-Stage Cluster

```
data(classes)
classes[1:2,]
```

```
##   class class_size
## 1     1         44
## 2     2         33
```

```
N<-nrow(classes)
set.seed(78065)

# select 5 classes with probability proportional to class size and with replacement
sample_units<-sample(1:N,5,replace=TRUE,prob=classes$class_size)
sample_units
```

```
## [1]  5 14  6 14  6
```

```
mysample<-classes[sample_units,]
mysample
```

```
##      class class_size
## 5        5         76
## 14      14        100
## 6        6         63
## 14.1    14        100
## 6.1      6         63
```

모집단에서 class_size 에 비례하는 확률로(즉, 수업시간에 배운 pps에 해당) 복원추출을 허용하여 표집을 한 것이 sample_units 에 저장되었다. 참고로 sample 함수는 R 기본 패키지에 내장되어있는 함수로, 꼭 샘플 표집 상황이 아니더라도 난수 생성 등을 할 때 유용하게 사용할 수 있다.

```
# calculate ExpectedHits and sampling weights
mysample$ExpectedHits<-5*mysample$class_size/sum(classes$class_size)
mysample$SamplingWeight<-1/mysample$ExpectedHits
mysample$psuid<-row.names(mysample)
mysample
```

```
##      class class_size ExpectedHits SamplingWeight psuid
## 5        5         76    0.5873261       1.702632     5
## 14      14        100    0.7727975       1.294000    14
## 6        6         63    0.4868624       2.053968     6
## 14.1    14        100    0.7727975       1.294000  14.1
## 6.1      6         63    0.4868624       2.053968   6.1
```

```
# check sum of sampling weights
sum(mysample$SamplingWeight)
```

```
## [1] 8.398568
```

ExpectedHits 변수는 표집된 각 class (집락)에 대하여 해당 집락이 표집될 확률을 계산한 것이다. 또한 Sampling-Weight 변수는 위에서 구한 값에 역수를 취한 값인데, 표집된 모든 데이터에 대한 sampling weight의 총합은 기댓값이 N (모집단의 총 집락 수)인 분포를 따르게 된다.

다음은 모집단에서 비복원추출과 pps를 이용하여 표집하는 방법이다.

```
# sampling without replacement
set.seed(330582)
cluster(data=classes, clustername=c("class"), size=5, method="systematic",
        pik=classes$class_size,description=TRUE)
```

```
## Number of selected clusters: 5
## Number of units in the population and number of selected units: 15 5
```

```
##   class ID_unit      Prob
## 1     1       1 0.3400309
## 2     5       5 0.5873261
## 3     8       8 0.3400309
## 4    11      11 0.3554869
## 5    14      14 0.7727975
```

pik는 표집 확률과 관련있다고 보면 되며, method에 “systematic” 옵션은 pik에 주어진 확률을 반영하여 표집하는 방법이라고 생각하면 된다. **?cluster** 를 콘솔에 입력해보면 단순임의표집+복원추출, 단순임의표집+비복원추출 등의 다른 옵션이 있는 것을 확인해 볼 수 있다.

2. Sampling with Unequal Probabilities in Two-Stage Cluster

classes 데이터는 집락이 15개이고, 이 모집단에서 각 집락 당 4명의 학생을 표집하는 이단집락표집을 시행하려 한다.

```
# create data frame classeslong
data(classes)
classeslong<-classes[rep(1:nrow(classes),times=classes$class_size),]
classeslong$studentid <- sequence(classes$class_size)
nrow(classeslong)
```

```
## [1] 647
```

```
table(classeslong$class) # check class sizes
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
##  44  33  26  22  76  63  20  44  54  34  46  24  46 100  15
```

```
head(classeslong)
```

```
##     class class_size studentid
## 1       1         44         1
## 1.1     1         44         2
## 1.2     1         44         3
## 1.3     1         44         4
## 1.4     1         44         5
## 1.5     1         44         6
```

```
# select a two-stage cluster sample, psu: class, ssu: studentid
# number of psus selected: n = 5 (pps systematic)
# number of students selected: m_i = 4 (srs without replacement)
# problist<-list(classes$class_size/647) # same results as next command
problist<-list(classes$class_size/647,4/classeslong$class_size) #selection prob
problist[[1]]  # extract the first object in the list. This is pps, size M_i/M
```

```
##  [1] 0.06800618 0.05100464 0.04018547 0.03400309 0.11746522 0.09737249
##  [7] 0.03091190 0.06800618 0.08346213 0.05255023 0.07109737 0.03709428
## [13] 0.07109737 0.15455951 0.02318393
```

```
problist[[2]][1:5] # first 5 values in second object in list
```

```
## [1] 0.09090909 0.09090909 0.09090909 0.09090909 0.09090909
```

```
# number of psus and ssus
n<-5
numberselect<-list(n,rep(4,n))
numberselect
```

```
## [[1]]
## [1] 5
##
## [[2]]
## [1] 4 4 4 4 4
```

```
# two-stage sampling
set.seed(75745)
tempid<-mstage(classeslong,stage=list("cluster","stratified"),
               varnames=list("class","studentid"),
               size=numberselect, method=list("systematic","srswor"),pik=problist)
```

여기까지가 표집 과정이다. 다음부터는 표집된 데이터에 관련된 여러 정보들을 정리 및 표출하는 작업이다.

```
# get data
sample1<-getdata(classeslong,tempid)[[1]]
# sample 1 contains the ssus of the 5 psus chosen at the first stage
# Prob_ 1 _stage has the first-stage selection probabilities
head(sample1)
```

```
##      class_size studentid class ID_unit Prob_ 1 _stage
## 4.21         22        22     4     125      0.1700155
## 4.20         22        21     4     124      0.1700155
## 4.6          22         7     4     110      0.1700155
## 4            22         1     4     104      0.1700155
## 4.7          22         8     4     111      0.1700155
## 4.8          22         9     4     112      0.1700155
```

```
nrow(sample1)
```

```
## [1] 285
```

```
table(sample1$class) # lists the psus selected in the first stage
```

```
##
##   4   6   9  13  14
##  22  63  54  46 100
```

```
sample2<-getdata(classeslong,tempid)[[2]]
# sample 2 contains the final sample
# Prob_ 2 _stage has the second-stage selection probabilities
# Prob has the final selection probabilities
head(sample2)
```

```
##      class class_size studentid ID_unit Prob_ 2 _stage      Prob
## 4.21     4         22        22     125     0.18181818 0.0309119
## 4.7      4         22         8     111     0.18181818 0.0309119
## 4.5      4         22         6     109     0.18181818 0.0309119
## 4.19     4         22        20     123     0.18181818 0.0309119
## 6.48     6         63        49     250     0.06349206 0.0309119
## 6.53     6         63        54     255     0.06349206 0.0309119
```

```r
nrow(sample2) # sample of 20 ssus altogether
```

```
## [1] 20
```

```r
table(sample2$class) # 4 ssus selected from each psu
```

```
##
##  4  6  9 13 14
##  4  4  4  4  4
```

```r
# calculate final weight = 1/Prob
sample2$finalweight<-1/sample2$Prob
# check that sum of final sampling weights equals population size
sum(sample2$finalweight)
```

```
## [1] 647
```

```r
sample2[,c(1,2,3,6,7)] # print variables from final sample
```

```
##       class class_size studentid      Prob finalweight
## 4.21      4         22        22 0.0309119       32.35
## 4.7       4         22         8 0.0309119       32.35
## 4.5       4         22         6 0.0309119       32.35
## 4.19      4         22        20 0.0309119       32.35
## 6.48      6         63        49 0.0309119       32.35
## 6.53      6         63        54 0.0309119       32.35
## 6.23      6         63        24 0.0309119       32.35
## 6.33      6         63        34 0.0309119       32.35
## 9.50      9         54        51 0.0309119       32.35
## 9.29      9         54        30 0.0309119       32.35
## 9.31      9         54        32 0.0309119       32.35
## 9.36      9         54        37 0.0309119       32.35
## 13.10    13         46        11 0.0309119       32.35
## 13       13         46         1 0.0309119       32.35
## 13.45    13         46        46 0.0309119       32.35
## 13.39    13         46        40 0.0309119       32.35
## 14.4     14        100         5 0.0309119       32.35
## 14.78    14        100        79 0.0309119       32.35
## 14.98    14        100        99 0.0309119       32.35
## 14.63    14        100        64 0.0309119       32.35
```

(Prob 관련해서 수업시간에 설명)

3. Computing Estimates from an Unequal-Probability and with-Replacement Sample

다음은 일단집락표집에서 모합, 모평균을 추정하는 과정이다. 모평균을 구할 때는 비추정량을 사용하였다.

```
studystat <- data.frame(class = c(12, 141, 142, 5, 1),
                        Mi = c(24, 100, 100, 76, 44),
                        tothours=c(75,203,203,191,168))
studystat$wt<-647/(studystat$Mi*5)
sum(studystat$wt) # check weight sum, which estimates N=15 psus
```

```
## [1] 12.62321
```

```
# design for with-replacement sample, no fpc argument
d0604 <- svydesign(id = ~1, weights=~wt, data = studystat)
d0604
```

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

```
# Ratio estimation using Mi as auxiliary variable
ratio0604<-svyratio(~tothours, ~Mi,design = d0604)
ratio0604
```

```
## Ratio estimator: svyratio.survey.design2(~tothours, ~Mi, design = d0604)
## Ratios=
##                Mi
## tothours 2.703268
## SEs=
##                 Mi
## tothours 0.3437741
```

```
confint(ratio0604, level=.95,df=4)
```

```
##                2.5 %   97.5 %
## tothours/Mi 1.748798 3.657738
```

```
# Can also estimate total hours studied for all students in population
svytotal(~tothours,d0604)
```

```
##          total     SE
## tothours  1749 222.42
```

일단집락표집에서 모합을 추정할 때는 표집된 각 집락에서 관측치의 총합을 구한 후, 그들을 가지고 단순임의표집(srs) 처럼 생각하면 된다고 하였는데, 코드를 살펴보면 그 과정이 그대로 반영되었음을 알 수 있다. (특히 svydesign 에서 id=~1 부분이 그렇다.)

다음은 이단집락표집에서의 추정이다.

```
students <- data.frame(class = rep(studystat$class,each=5),
                       popMi = rep(studystat$Mi,each=5),
                       sampmi=rep(5,25),
                       hours=c(2,3,2.5,3,1.5,2.5,2,3,0,0.5,3,0.5,1.5,2,3,1,2.5,3,5,2.5,4,4.5,3,2,5))
# The 'with' function allows us to calculate using variables from a data frame
# without having to type the data frame name for all of them
students$studentwt <- with(students,(647/(popMi*5)) * (popMi/sampmi))
# check the sum of the weights
sum(students$studentwt)
```

```
## [1] 647
```

```
# create the design object
d0606 <- svydesign(id = ~class, weights=~studentwt, data = students)
d0606
```

```
## 1 - level Cluster Sampling design (with replacement)
## With (5) clusters.
## svydesign(id = ~class, weights = ~studentwt, data = students)
```

```
# estimate mean and SE
svymean(~hours,d0606)
```

```
##       mean     SE
## hours  2.5 0.3606
```

```
degf(d0606)
```

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

```
confint(svymean(~hours,d0606),level=.95,df=4) #use t-approximation
```

```
##          2.5 %   97.5 %
## hours 1.498938 3.501062
```

```
# estimate total and SE
svytotal(~hours,d0606)
```

```
##        total     SE
## hours 1617.5 233.28
```

```
confint(svytotal(~hours,d0606),level=.95,df=4)
```

```
##          2.5 %   97.5 %
## hours 969.8132 2265.187
```

'students$studentwt <- with(students,(647/(popMi5)) (popMi/sampmi))' 부분을 보면 2절에서의 Prob와 같이 계산됨을 알 수 있고, 계산식에서 분자, 분모의 popMi 가 소거되어 모든 데이터의 weight가 같아짐을 알 수 있다.