Sampling Design and Survey Practice — Lab 1

September 20, 2021

1 Introduction

- We will use R in the lab session.
- Installing R and basic commands: see the attached file. (`lab0.pdf`)

1.1 Random experiments in R

- Use `sample(n)` to generate a random permutation from $1, \cdots, n$.

```
sample(5)

## [1] 5 4 1 2 3
```

- Every time `sample(n)` is executed, it produces a different permutation.
- This means that the experiment is not **reproducible**, and debugging the code may become difficult.
- To overcome this problem, fix the state of the random number generator using `set.seed()` function.

```
set.seed(0)
sample(5)

## [1] 1 4 3 5 2
```

1.2 `sample()` function

- `n`: integer / `x`: vector
- `sample(n)`: select a random permutation from $1, \cdots, n$.
- `sample(x)`: randomly permute `x`.

```
sample(c('A','B','C','D','A'))

## [1] "A" "C" "D" "B" "A"
```

- `sample(x, size=n)`: randomly sample `n` items from `x` without replacement.

```
sample(c('A','B','C','D','A'), 3)

## [1] "C" "A" "A"

sample(10, 3)

## [1]  5 10  2
```

- `sample(x, replace=TRUE)`: a bootstrap sample from `x`.

```
sample(10, replace=TRUE)

##  [1]  6 10  7  9  5  5  9  9  5  5
```

- `sample(x, n, replace=TRUE)`: randomly sample `n` items from `x` with replacement.

```
sample(10, 3, replace=TRUE)

## [1]  2 10  9
```

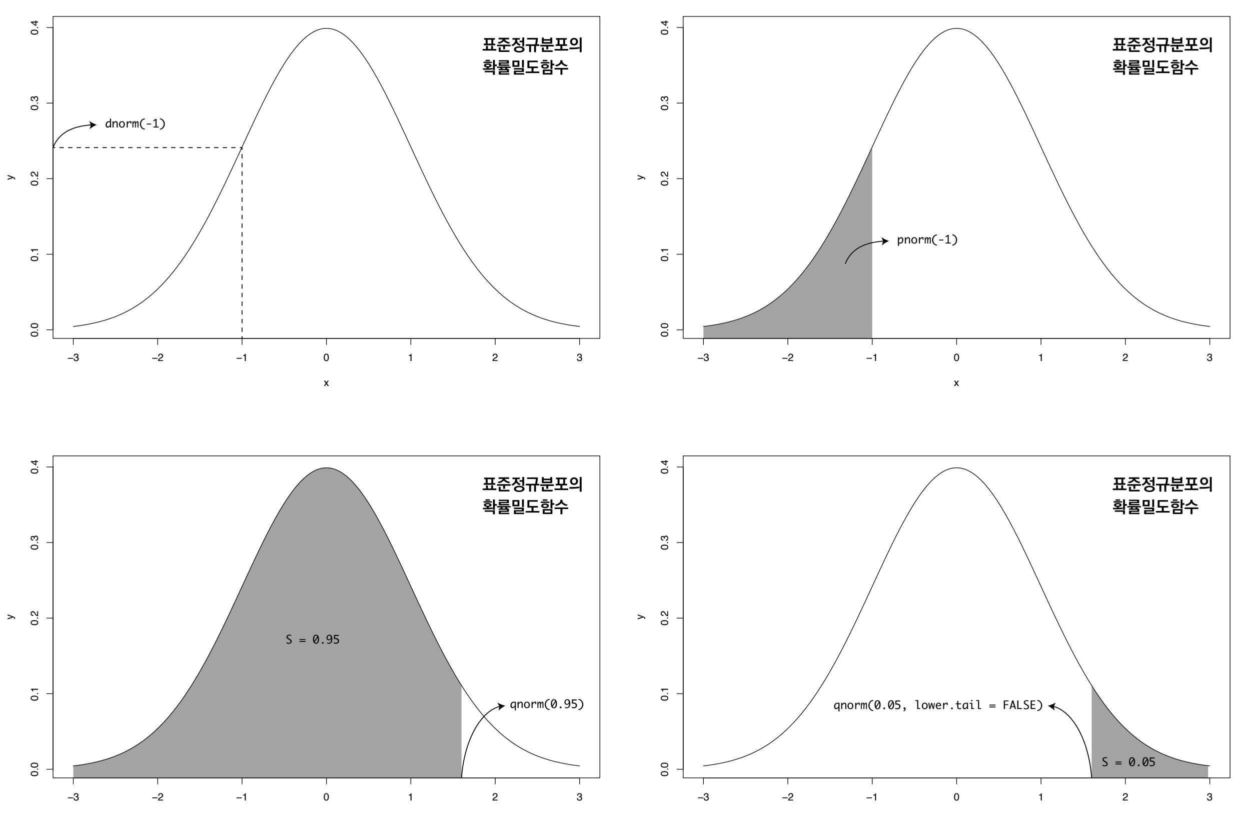
- e.g. Lotto number generator

```
sample(45, 6)

## [1] 44 15 33 20 35  6
```

1.3 Distribution functions

- `d` + (distribution): probability distribution function (pdf)
- `p` + (distribution): cumulative distribution function (cdf)
- `q` + (distribution): quantile function (e.g. `qnorm(0.975) = 1.959964`)
- `r` + (distribution): random number following the distribution



2 Simple random sampling

- We will use the `subway_snu.csv` data, collected from `http://data.seoul.go.kr/dataList/OA-12914/S/1/datasetView.do` (서울 열린데이터 광장, 서울시 지하철호선별 역별 승하차 인원 정보).
- The data set contains number of passengers who got off at the 'Seoul National University station' each day, observed from 2021.3.1 to 2021.8.31. (6 months, 184 days).
- The aim is to estimate the <u>mean of daily number of passengers who got off at the 'Seoul National University station'</u> (μ).

```
subway <- read.csv("subway_snu.csv", header = TRUE)
dim(subway)

## [1] 184   3

N <- 184
head(subway, 7)

##       date number day
## 1 20210301  20689 MON
## 2 20210302  44535 TUE
## 3 20210303  45346 WED
## 4 20210304  44638 THU
## 5 20210305  47902 FRI
## 6 20210306  36507 SAT
## 7 20210307  26463 SUN

mu <- mean(subway$number) # population mean
sigma.sq <- var(subway$number)*(N-1)/N # population variance
c(mean=mu, sd=sqrt(sigma.sq))

##      mean        sd
## 39450.473  8119.895
```

- For example, 20689 people got off at the SNU station on March 1st, 2021 (Monday).
- We *know* that $\mu = 39450.47$. We will use simple random sampling (SRS) to estimate μ and compare the estimated value to the true value of μ.

2.1 Estimation of a population mean

- There are $N = 184$ units in the population.
- Draw $n = 30$ samples $y_1, \cdots, y_n$ using SRS and estimate the population mean μ.
- In SRS, we estimate μ using the sample mean $\bar{y} = \sum_{i=1}^{n} y_i/n$.

```
set.seed(0)
n <- 30
sample.idx <- sample(N,n)
```

```r
sample.subway <- subway[sample.idx,]
y.bar <- mean(sample.subway$number)
y.bar # estimate of the population mean
```

```
## [1] 37744.87
```

- The estimated variance of the $\bar{y}$ is given as

$$\widehat{\text{var}}(\bar{y}) = \frac{N-n}{Nn}s^2, \quad s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2.$$

```r
s.sq <- var(sample.subway$number)
est.var <- (N-n)/(N*n)*s.sq
est.var
```

```
## [1] 2171541
```

- Using the estimated mean and variance, $100(1-\alpha)\%$ confidence interval of μ is given as

$$\left[\bar{y} - z_{\alpha/2}\sqrt{\widehat{\text{var}}(\bar{y})},\, \bar{y} + z_{\alpha/2}\sqrt{\widehat{\text{var}}(\bar{y})}\right].$$

```r
alpha <- 0.05
lower <- y.bar + qnorm(alpha/2)*sqrt(est.var)
upper <- y.bar - qnorm(alpha/2)*sqrt(est.var)
c(lower=lower, upper=upper) # 95% confidence interval
```

```
##    lower    upper
## 34856.63 40633.10
```

- Simulation with 1000 repetitions:

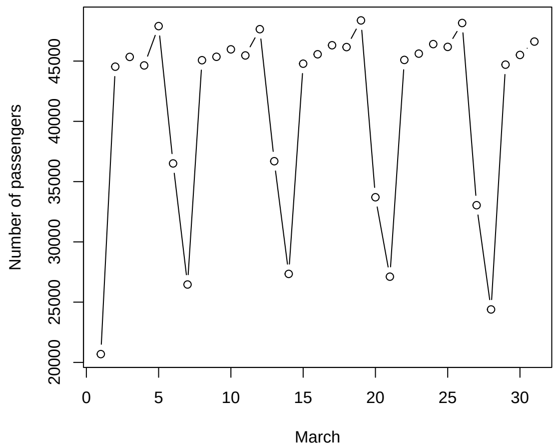
```r
count <- 0
for(i in 1:1000){
  sample.idx <- sample(N,n)
  sample.subway <- subway[sample.idx,]
  y.bar <- mean(sample.subway$number)
  s.sq <- var(sample.subway$number)
  est.var <- (N-n)/(N*n)*s.sq
  lower <- y.bar + qnorm(alpha/2)*sqrt(est.var)
  upper <- y.bar - qnorm(alpha/2)*sqrt(est.var)
  if((lower < mu) & (mu < upper)) count <- count + 1
}
count / 1000
```

```
## [1] 0.935
```

3 Stratified random sampling

- From the plot below, we see that stratified random sampling can be conducted by forming three strata: (1) weekdays (2) Saturday (3) Sunday.

```
plot(1:31, subway$number[1:31], type="b", xlab="March", ylab="Number of passengers")
```



- Define a `group` variable, to be used for stratified random sampling.

```
subway$group <- ifelse(subway$day=="SAT",2,ifelse(subway$day=="SUN",3,1))
N.vector <- table(subway$group)
N.vector

##
##   1   2   3
## 132  26  26
```

- We wish to select $n_1 = 10$, $n_2 = 10$, $n_3 = 10$ units from the weekdays, Saturday, Sunday, respectively.
- To conduct a stratified random sampling, we use the `strata` function in the **sampling** package.

```
n1 <- 10
n2 <- 10
n3 <- 10
n.vector <- c(n1, n2, n3)

# install.packages("sampling")
library(sampling)

set.seed(0)
strata.subway <- sampling::strata(subway, "group", size=n.vector, method="srswor")
strata.subway2 <- getdata(subway, strata.subway)
```

- Using the **survey** package, we can easily estimate μ.

```
# install.packages("survey")
library(survey)
mydesign <- svydesign(ids=~1, strata=~group, data=strata.subway2, fpc=~rep(N.vector,each=10))
res <- svymean(~number, design=mydesign)
res
```

```
##         mean     SE
## number 40318 544.71
```

```
confint(res)
```

```
##           2.5 %   97.5 %
## number 39250.36 41385.58
```

- Same results can be obtained without using the package. For stratified random sampling, estimator of the μ is given as

$$\bar{y}_{\text{st}} = \sum_{j=1}^{L} w_j \bar{y}_{j\cdot}, \quad w_j = \frac{N_j}{N}.$$

```
y.bar.vec <- tapply(strata.subway2$number, strata.subway2$group, mean)
y.bar.vec
```

```
##       1       2       3
## 45035.3 30932.3 25754.1
```

```
w.j <- N.vector/N
y.bar.st <- sum(y.bar.vec*w.j)
y.bar.st
```

```
## [1] 40317.97
```

- The estimated variance of the $\bar{y}_{\text{st}}$ is given as

$$\widehat{\text{var}}(\bar{y}_{\text{st}}) = \sum_{j=1}^{L} w_j^2 \frac{N_j - n_j}{n_j N_j} s_j^2, \quad s_j^2 = \text{sample variance in the } j\text{th stratum.}$$

```
s.sq.vec <- tapply(strata.subway2$number, strata.subway2$group, var)
est.var.st <- sum(w.j^2*(N.vector-n.vector)/(N.vector*n.vector)*s.sq.vec)
est.var.st
```

```
## [1] 296707.3
```

- Using the estimated mean and variance, $100(1-\alpha)\%$ confidence interval of μ is given as

$$\left[\bar{y}_{\text{st}} - z_{\alpha/2}\sqrt{\widehat{\text{var}}(\bar{y}_{\text{st}})},\ \bar{y}_{\text{st}} + z_{\alpha/2}\sqrt{\widehat{\text{var}}(\bar{y}_{\text{st}})}\right].$$

```r
alpha <- 0.05
lower.st <- y.bar.st + qnorm(alpha/2)*sqrt(est.var.st)
upper.st <- y.bar.st - qnorm(alpha/2)*sqrt(est.var.st)
c(lower=lower.st, upper=upper.st) # 95% confidence interval
```

```
##    lower    upper
## 39250.36 41385.58
```

- Observe that the same result is obtained.