PR3

PR3\_201821479\_황혜린

2018년 9월 25일

# Factor

## 1. 명목형 자료 만들기

score<-factor(c(3, 2, 3, 4, 3, 1, 1, 3, 2, 2, 1, 1, 5),  
 levels=c(1,2,3,4),  
 labels=c("A", "B", "C", "D"),  
 ordered=T)  
score

## [1] C B C D C A A C B B A A <NA>  
## Levels: A < B < C < D

## 2. 명목형 자료로 변환하기

fac\_char <- c("포도", "키위", "메론", "바나나", "딸기" )  
attributes(fac\_char)

## NULL

fac\_char <- as.factor(fac\_char)  
attributes(fac\_char)

## $levels  
## [1] "딸기" "메론" "바나나" "키위" "포도"   
##   
## $class  
## [1] "factor"

fac\_num = 1:4  
attributes(fac\_num)

## NULL

fac\_num = as.factor(fac\_num)  
attributes(fac\_num)

## $levels  
## [1] "1" "2" "3" "4"  
##   
## $class  
## [1] "factor"

## 3. 팩터형 자료 빈도 파악

table(score)

## score  
## A B C D   
## 4 3 4 1

tb <- table(score)  
tb[tb>3]

## score  
## A C   
## 4 4

## 4. 서수형 자료와 명목형 자료의 차이

score[1] > score[3]

## [1] FALSE

fac\_char[1] > fac\_char[2]

## Warning in Ops.factor(fac\_char[1], fac\_char[2]): '>' not meaningful for  
## factors

## [1] NA

#(1)은 서수형 자료로서 숫자의 크기를 비교하는 수식에서 false라는 값을 나타낼 수 있지만, (2)는 명목형 자료로서 명목형 자료란 분류를 목적으로 하므로 숫자의 크기를 비교하는 것이 의미가 없다. 따라서 "not meaningful for factors"라는 오류가 뜬다.

# Matrix

## 5. matrix 생성

mat <- matrix(1:8, nrow=2, ncol=4, byrow=T)  
  
dim(mat) ; length(mat)

## [1] 2 4

## [1] 8

matrix(1:8, nrow=2, ncol=4, byrow=F)

## [,1] [,2] [,3] [,4]  
## [1,] 1 3 5 7  
## [2,] 2 4 6 8

matrix(1:8, nrow=2)

## [,1] [,2] [,3] [,4]  
## [1,] 1 3 5 7  
## [2,] 2 4 6 8

matrix(1:8, ncol=2)

## [,1] [,2]  
## [1,] 1 5  
## [2,] 2 6  
## [3,] 3 7  
## [4,] 4 8

matrix(1:8, ncol=4, byrow=T)

## [,1] [,2] [,3] [,4]  
## [1,] 1 2 3 4  
## [2,] 5 6 7 8

matrix(1:9, nrow=3, ncol=3, dimnames = (list(c("r1", "r2","r3"), c("c1", "c2","c3"))))

## c1 c2 c3  
## r1 1 4 7  
## r2 2 5 8  
## r3 3 6 9

## 6. matrix 각 차원에 이름 부여

mat

## [,1] [,2] [,3] [,4]  
## [1,] 1 2 3 4  
## [2,] 5 6 7 8

rownames(mat)<-c("행 1","행 2")  
colnames(mat)<-c("열 1","열 2","열 3","열 4")  
mat

## 열 1 열 2 열 3 열 4  
## 행 1 1 2 3 4  
## 행 2 5 6 7 8

## 7. rbind() 와 cbind()를 사용한 매트릭스 생성

x <- 2:4 ;x

## [1] 2 3 4

y <- 9:11 ; y

## [1] 9 10 11

cbind(x, y)

## x y  
## [1,] 2 9  
## [2,] 3 10  
## [3,] 4 11

rbind(x, y)

## [,1] [,2] [,3]  
## x 2 3 4  
## y 9 10 11

## 8. rbind()와 cbind()를 사용한 데이터 추가

mat

## 열 1 열 2 열 3 열 4  
## 행 1 1 2 3 4  
## 행 2 5 6 7 8

cbind(mat,10:11)

## 열 1 열 2 열 3 열 4   
## 행 1 1 2 3 4 10  
## 행 2 5 6 7 8 11

rbind(mat,20:23)

## 열 1 열 2 열 3 열 4  
## 행 1 1 2 3 4  
## 행 2 5 6 7 8  
## 20 21 22 23

## 9. matrix 데이터 접근과 변환

x <- 1:3 ;x

## [1] 1 2 3

y <- 10:12 ; y

## [1] 10 11 12

mat <- cbind(x, y) ; mat

## x y  
## [1,] 1 10  
## [2,] 2 11  
## [3,] 3 12

mat[1,1] <- 100 ; mat

## x y  
## [1,] 100 10  
## [2,] 2 11  
## [3,] 3 12

mat[2,] <- mat[2,] / 4 ; mat

## x y  
## [1,] 100.0 10.00  
## [2,] 0.5 2.75  
## [3,] 3.0 12.00

mat[,2] <- mat[,2] - mat[,1] \* 3 ; mat

## x y  
## [1,] 100.0 -290.00  
## [2,] 0.5 1.25  
## [3,] 3.0 3.00

# list

## 10. 여러 벡터를 이용해 리스트 만들기

str\_vec <- c("korea", "USA", "Japan")  
num\_vec <- c(100, 200, 300, 400, 500)  
mat <- matrix(2:9, 2, 4)  
list\_tot <- list(str\_vec, num\_vec, mat)  
print(list\_tot)

## [[1]]  
## [1] "korea" "USA" "Japan"  
##   
## [[2]]  
## [1] 100 200 300 400 500  
##   
## [[3]]  
## [,1] [,2] [,3] [,4]  
## [1,] 2 4 6 8  
## [2,] 3 5 7 9

names(list\_tot) <- c('str\_vec', 'num\_vec', 'mat')  
print(list\_tot)

## $str\_vec  
## [1] "korea" "USA" "Japan"  
##   
## $num\_vec  
## [1] 100 200 300 400 500  
##   
## $mat  
## [,1] [,2] [,3] [,4]  
## [1,] 2 4 6 8  
## [2,] 3 5 7 9

## 11. list 함수 내에서 성분의 이름 지정하여 리스트 만들기

list\_tot2= list(seq = seq(1,10,2),   
 str = c("토끼", "사자","코끼리", "양"),  
 plus = rep(c('고구마', '감자','옥수수'),2))  
print(list\_tot2)

## $seq  
## [1] 1 3 5 7 9  
##   
## $str  
## [1] "토끼" "사자" "코끼리" "양"   
##   
## $plus  
## [1] "고구마" "감자" "옥수수" "고구마" "감자" "옥수수"

## 12. 데이터의 속성을 확인하는 다양한 함수

class(list\_tot)

## [1] "list"

length(list\_tot)

## [1] 3

attributes(list\_tot)

## $names  
## [1] "str\_vec" "num\_vec" "mat"

str(list\_tot)

## List of 3  
## $ str\_vec: chr [1:3] "korea" "USA" "Japan"  
## $ num\_vec: num [1:5] 100 200 300 400 500  
## $ mat : int [1:2, 1:4] 2 3 4 5 6 7 8 9

## 13. 리스트의 성분에 접근하기

list\_tot2[1]

## $seq  
## [1] 1 3 5 7 9

list\_tot2[3]

## $plus  
## [1] "고구마" "감자" "옥수수" "고구마" "감자" "옥수수"

list\_tot2[1:2]

## $seq  
## [1] 1 3 5 7 9  
##   
## $str  
## [1] "토끼" "사자" "코끼리" "양"

list\_tot2$seq

## [1] 1 3 5 7 9

list\_tot2$str

## [1] "토끼" "사자" "코끼리" "양"

## 14. 리스트의 성분안에 있는 원소에 접근하기

list\_tot[[3]][1]

## [1] 2

list\_tot2$seq[3]

## [1] 5

list\_tot2$str[1:2]

## [1] "토끼" "사자"

## 15. 리스트의 성분이나 원소 조작하기

list\_tot[1] = NULL  
str(list\_tot)

## List of 2  
## $ num\_vec: num [1:5] 100 200 300 400 500  
## $ mat : int [1:2, 1:4] 2 3 4 5 6 7 8 9

list\_tot2$str[1] = "고양이"  
str(list\_tot2)

## List of 3  
## $ seq : num [1:5] 1 3 5 7 9  
## $ str : chr [1:4] "고양이" "사자" "코끼리" "양"  
## $ plus: chr [1:6] "고구마" "감자" "옥수수" "고구마" ...

list\_tot$NEW = 2:5  
str(list\_tot)

## List of 3  
## $ num\_vec: num [1:5] 100 200 300 400 500  
## $ mat : int [1:2, 1:4] 2 3 4 5 6 7 8 9  
## $ NEW : int [1:4] 2 3 4 5

## 16. 리스트의 성분에 하위 리스트 추가하여 계층적으로 리스트 만들기

list\_tot$hierarchy[[1]] = list\_tot2  
str(list\_tot)

## List of 4  
## $ num\_vec : num [1:5] 100 200 300 400 500  
## $ mat : int [1:2, 1:4] 2 3 4 5 6 7 8 9  
## $ NEW : int [1:4] 2 3 4 5  
## $ hierarchy:List of 1  
## ..$ :List of 3  
## .. ..$ seq : num [1:5] 1 3 5 7 9  
## .. ..$ str : chr [1:4] "고양이" "사자" "코끼리" "양"  
## .. ..$ plus: chr [1:6] "고구마" "감자" "옥수수" "고구마" ...

## 17. Array 생성하기

arr <- array(1:18, dim=c(3,3,2),  
 dimnames=list(c("KOR","CHI","JAP"),  
 c("GDP.R", "USD.R", "Cuur.Acc"),  
 c("2011Y", "2012Y")))  
arr

## , , 2011Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 1 4 7  
## CHI 2 5 8  
## JAP 3 6 9  
##   
## , , 2012Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 10 13 16  
## CHI 11 14 17  
## JAP 12 15 18

arr1 <- 1:18  
dim(arr1) <- c(3,3,2)  
dimnames(arr1) <- list(c("KOR","CHI","JAP"),  
 c("GDP.R", "USD.R", "Cuur.Acc"),  
 c("2011Y", "2012Y"))  
arr1

## , , 2011Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 1 4 7  
## CHI 2 5 8  
## JAP 3 6 9  
##   
## , , 2012Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 10 13 16  
## CHI 11 14 17  
## JAP 12 15 18

arr1==arr

## , , 2011Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR TRUE TRUE TRUE  
## CHI TRUE TRUE TRUE  
## JAP TRUE TRUE TRUE  
##   
## , , 2012Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR TRUE TRUE TRUE  
## CHI TRUE TRUE TRUE  
## JAP TRUE TRUE TRUE

## 18. Array 조작방법

arr

## , , 2011Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 1 4 7  
## CHI 2 5 8  
## JAP 3 6 9  
##   
## , , 2012Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 10 13 16  
## CHI 11 14 17  
## JAP 12 15 18

arr[1,,]

## 2011Y 2012Y  
## GDP.R 1 10  
## USD.R 4 13  
## Cuur.Acc 7 16

arr[,-2,]

## , , 2011Y  
##   
## GDP.R Cuur.Acc  
## KOR 1 7  
## CHI 2 8  
## JAP 3 9  
##   
## , , 2012Y  
##   
## GDP.R Cuur.Acc  
## KOR 10 16  
## CHI 11 17  
## JAP 12 18

arr[,,2]

## GDP.R USD.R Cuur.Acc  
## KOR 10 13 16  
## CHI 11 14 17  
## JAP 12 15 18

arr[,,"2012Y"]

## GDP.R USD.R Cuur.Acc  
## KOR 10 13 16  
## CHI 11 14 17  
## JAP 12 15 18

arr[c(T,T,F),,2]

## GDP.R USD.R Cuur.Acc  
## KOR 10 13 16  
## CHI 11 14 17

arr.tmp <- arr  
arr.tmp

## , , 2011Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 1 4 7  
## CHI 2 5 8  
## JAP 3 6 9  
##   
## , , 2012Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 10 13 16  
## CHI 11 14 17  
## JAP 12 15 18

arr.tmp[,,1] <- c(5,6,4)  
arr.tmp

## , , 2011Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 5 5 5  
## CHI 6 6 6  
## JAP 4 4 4  
##   
## , , 2012Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 10 13 16  
## CHI 11 14 17  
## JAP 12 15 18

arr.tmp[,1,2] <- NA  
arr.tmp

## , , 2011Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 5 5 5  
## CHI 6 6 6  
## JAP 4 4 4  
##   
## , , 2012Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR NA 13 16  
## CHI NA 14 17  
## JAP NA 15 18

arr.tmp[is.na(arr.tmp)] <- c(8,5,2)  
arr.tmp

## , , 2011Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 5 5 5  
## CHI 6 6 6  
## JAP 4 4 4  
##   
## , , 2012Y  
##   
## GDP.R USD.R Cuur.Acc  
## KOR 8 13 16  
## CHI 5 14 17  
## JAP 2 15 18

## 19. NA값 다루기

x <- c(1, 2, NA, 4, NA, 5)  
is.na(x)

## [1] FALSE FALSE TRUE FALSE TRUE FALSE

bad <- is.na(x)  
y <- x[!bad]  
mean(y)

## [1] 3

x <- c(1, 2, NA, 4, NA, 5)  
good <- complete.cases(x)  
x[good]

## [1] 1 2 4 5

# PR3 연습문제

## List

## 1-1)

female\_unemployment\_rate <- c(3.7, 3.0, 2.7, 2.5, 3.3, 2.9, 2.6, 2.6, 4.0, 3.7, 3.1, 3.1, 4.2, 3.8, 3.2, 3.0, 4.2, 3.6, 3.3, 3.2, 4.4, 3.7, 3.2, 2.9) #여성실업률  
male\_unemployment\_rate <- c(3.8, 3.4, 3.2, 3.1, 3.8, 3.2, 3.2, 2.9, 3.9, 3.6, 3.4, 3.3, 4.0, 3.8, 3.6, 3.2, 4.3, 3.8, 3.7, 3.2, 4.2, 4.0, 3.7, 3.4) #남성실업률  
unemploy\_rate <- (female\_unemployment\_rate + male\_unemployment\_rate) / 2 #실업률  
unemploy\_rate\_list <- list(female\_unemployment\_rate, male\_unemployment\_rate, unemploy\_rate)  
print(unemploy\_rate\_list) #리스트 출력!!

## [[1]]  
## [1] 3.7 3.0 2.7 2.5 3.3 2.9 2.6 2.6 4.0 3.7 3.1 3.1 4.2 3.8 3.2 3.0 4.2  
## [18] 3.6 3.3 3.2 4.4 3.7 3.2 2.9  
##   
## [[2]]  
## [1] 3.8 3.4 3.2 3.1 3.8 3.2 3.2 2.9 3.9 3.6 3.4 3.3 4.0 3.8 3.6 3.2 4.3  
## [18] 3.8 3.7 3.2 4.2 4.0 3.7 3.4  
##   
## [[3]]  
## [1] 3.75 3.20 2.95 2.80 3.55 3.05 2.90 2.75 3.95 3.65 3.25 3.20 4.10 3.80  
## [15] 3.40 3.10 4.25 3.70 3.50 3.20 4.30 3.85 3.45 3.15

names(unemploy\_rate\_list) <- c('female\_unemployment\_list', 'male\_unemployment\_list', 'unemploy\_rate') #names하고 출력   
print(unemploy\_rate\_list)

## $female\_unemployment\_list  
## [1] 3.7 3.0 2.7 2.5 3.3 2.9 2.6 2.6 4.0 3.7 3.1 3.1 4.2 3.8 3.2 3.0 4.2  
## [18] 3.6 3.3 3.2 4.4 3.7 3.2 2.9  
##   
## $male\_unemployment\_list  
## [1] 3.8 3.4 3.2 3.1 3.8 3.2 3.2 2.9 3.9 3.6 3.4 3.3 4.0 3.8 3.6 3.2 4.3  
## [18] 3.8 3.7 3.2 4.2 4.0 3.7 3.4  
##   
## $unemploy\_rate  
## [1] 3.75 3.20 2.95 2.80 3.55 3.05 2.90 2.75 3.95 3.65 3.25 3.20 4.10 3.80  
## [15] 3.40 3.10 4.25 3.70 3.50 3.20 4.30 3.85 3.45 3.15

str(unemploy\_rate\_list) #데이터의 속성을 한번 더 확인!

## List of 3  
## $ female\_unemployment\_list: num [1:24] 3.7 3 2.7 2.5 3.3 2.9 2.6 2.6 4 3.7 ...  
## $ male\_unemployment\_list : num [1:24] 3.8 3.4 3.2 3.1 3.8 3.2 3.2 2.9 3.9 3.6 ...  
## $ unemploy\_rate : num [1:24] 3.75 3.2 2.95 2.8 3.55 3.05 2.9 2.75 3.95 3.65 ...

## 1-2)

unemploy\_rate\_list[1] = NULL #첫번째 성분을 삭제합니다  
str(unemploy\_rate\_list)

## List of 2  
## $ male\_unemployment\_list: num [1:24] 3.8 3.4 3.2 3.1 3.8 3.2 3.2 2.9 3.9 3.6 ...  
## $ unemploy\_rate : num [1:24] 3.75 3.2 2.95 2.8 3.55 3.05 2.9 2.75 3.95 3.65 ...

unemploy\_rate\_list$female\_unemployment\_list = "startup\_rate"  
str(unemploy\_rate\_list) #startup\_rate라는 원소를 덮어씁니다

## List of 3  
## $ male\_unemployment\_list : num [1:24] 3.8 3.4 3.2 3.1 3.8 3.2 3.2 2.9 3.9 3.6 ...  
## $ unemploy\_rate : num [1:24] 3.75 3.2 2.95 2.8 3.55 3.05 2.9 2.75 3.95 3.65 ...  
## $ female\_unemployment\_list: chr "startup\_rate"

unemploy\_rate\_list$NEW = 1:24  
str(unemploy\_rate\_list) #새로운 1:24라는 성분을 추가합니다

## List of 4  
## $ male\_unemployment\_list : num [1:24] 3.8 3.4 3.2 3.1 3.8 3.2 3.2 2.9 3.9 3.6 ...  
## $ unemploy\_rate : num [1:24] 3.75 3.2 2.95 2.8 3.55 3.05 2.9 2.75 3.95 3.65 ...  
## $ female\_unemployment\_list: chr "startup\_rate"  
## $ NEW : int [1:24] 1 2 3 4 5 6 7 8 9 10 ...

## Array

## 2-1)

female\_unemployment\_arr <- array(female\_unemployment\_rate, dim=c(4,6,1),   
dimnames=list(c("1/4","2/4","3/4","4/4"),  
 c("2012Y", "2013Y", "2014Y", "2015Y", "2016Y", "2017Y"),  
 c("woman\_unemp")))  
  
female\_unemployment\_arr #여성실업률 array

## , , woman\_unemp  
##   
## 2012Y 2013Y 2014Y 2015Y 2016Y 2017Y  
## 1/4 3.7 3.3 4.0 4.2 4.2 4.4  
## 2/4 3.0 2.9 3.7 3.8 3.6 3.7  
## 3/4 2.7 2.6 3.1 3.2 3.3 3.2  
## 4/4 2.5 2.6 3.1 3.0 3.2 2.9

male\_unemployment\_arr <- array(male\_unemployment\_rate, dim=c(4,6,1),   
dimnames=list(c("1/4","2/4","3/4","4/4"),  
 c("2012Y", "2013Y", "2014Y", "2015Y", "2016Y", "2017Y"),  
 c("man\_unemp")))  
  
male\_unemployment\_arr #남성실업률 array

## , , man\_unemp  
##   
## 2012Y 2013Y 2014Y 2015Y 2016Y 2017Y  
## 1/4 3.8 3.8 3.9 4.0 4.3 4.2  
## 2/4 3.4 3.2 3.6 3.8 3.8 4.0  
## 3/4 3.2 3.2 3.4 3.6 3.7 3.7  
## 4/4 3.1 2.9 3.3 3.2 3.2 3.4

unemploy\_rate\_arr <- array(unemploy\_rate, dim=c(4,6,1),   
dimnames=list(c("1/4","2/4","3/4","4/4"),  
 c("2012Y", "2013Y", "2014Y", "2015Y", "2016Y", "2017Y"),  
 c("total\_unemp")))  
  
unemploy\_rate\_arr #총 실업률 array

## , , total\_unemp  
##   
## 2012Y 2013Y 2014Y 2015Y 2016Y 2017Y  
## 1/4 3.75 3.55 3.95 4.1 4.25 4.30  
## 2/4 3.20 3.05 3.65 3.8 3.70 3.85  
## 3/4 2.95 2.90 3.25 3.4 3.50 3.45  
## 4/4 2.80 2.75 3.20 3.1 3.20 3.15

## 2-2)

female\_unemployment\_arr[,-3,]

## 2012Y 2013Y 2015Y 2016Y 2017Y  
## 1/4 3.7 3.3 4.2 4.2 4.4  
## 2/4 3.0 2.9 3.8 3.6 3.7  
## 3/4 2.7 2.6 3.2 3.3 3.2  
## 4/4 2.5 2.6 3.0 3.2 2.9

male\_unemployment\_arr[,-3,]

## 2012Y 2013Y 2015Y 2016Y 2017Y  
## 1/4 3.8 3.8 4.0 4.3 4.2  
## 2/4 3.4 3.2 3.8 3.8 4.0  
## 3/4 3.2 3.2 3.6 3.7 3.7  
## 4/4 3.1 2.9 3.2 3.2 3.4

unemploy\_rate\_arr[,-3,]

## 2012Y 2013Y 2015Y 2016Y 2017Y  
## 1/4 3.75 3.55 4.1 4.25 4.30  
## 2/4 3.20 3.05 3.8 3.70 3.85  
## 3/4 2.95 2.90 3.4 3.50 3.45  
## 4/4 2.80 2.75 3.1 3.20 3.15

#위 3개는 2014년을 제외한 각각의 실업률입니다  
  
unemploy\_rate\_arr #총 실업률만 출력합니다

## , , total\_unemp  
##   
## 2012Y 2013Y 2014Y 2015Y 2016Y 2017Y  
## 1/4 3.75 3.55 3.95 4.1 4.25 4.30  
## 2/4 3.20 3.05 3.65 3.8 3.70 3.85  
## 3/4 2.95 2.90 3.25 3.4 3.50 3.45  
## 4/4 2.80 2.75 3.20 3.1 3.20 3.15

# PR3 도전문제

# 문제

mat <- matrix(c(2,4,-1,3,1,2,1,-3,2,4,-2,2), nrow=3, ncol=4)  
mat #계수 행렬을 만들고 값을 확인해봅니다

## [,1] [,2] [,3] [,4]  
## [1,] 2 3 1 4  
## [2,] 4 1 -3 -2  
## [3,] -1 2 2 2

mat[1,] <- mat[1,] / 2 ; mat #과정2

## [,1] [,2] [,3] [,4]  
## [1,] 1 1.5 0.5 2  
## [2,] 4 1.0 -3.0 -2  
## [3,] -1 2.0 2.0 2

mat[2,] <- mat[2,] - mat[1,] \* 4 ; mat

## [,1] [,2] [,3] [,4]  
## [1,] 1 1.5 0.5 2  
## [2,] 0 -5.0 -5.0 -10  
## [3,] -1 2.0 2.0 2

mat[3,] <- mat[3,] - mat[1,] \* (-1) ; mat #과정3

## [,1] [,2] [,3] [,4]  
## [1,] 1 1.5 0.5 2  
## [2,] 0 -5.0 -5.0 -10  
## [3,] 0 3.5 2.5 4

mat[2,] <- mat[2,] / (-5) ; mat #과정4

## [,1] [,2] [,3] [,4]  
## [1,] 1 1.5 0.5 2  
## [2,] 0 1.0 1.0 2  
## [3,] 0 3.5 2.5 4

mat[1,] <- mat[1,] - mat[2,] \* 1.5 ; mat

## [,1] [,2] [,3] [,4]  
## [1,] 1 0.0 -1.0 -1  
## [2,] 0 1.0 1.0 2  
## [3,] 0 3.5 2.5 4

mat[3,] <- mat[3,] - mat[2,] \* 3.5 ; mat #과정5

## [,1] [,2] [,3] [,4]  
## [1,] 1 0 -1 -1  
## [2,] 0 1 1 2  
## [3,] 0 0 -1 -3

mat[3,] <- mat[3,] / (-1) ; mat #과정6

## [,1] [,2] [,3] [,4]  
## [1,] 1 0 -1 -1  
## [2,] 0 1 1 2  
## [3,] 0 0 1 3

mat[1,] <- mat[1,] - mat[3,] \* (-1) ; mat

## [,1] [,2] [,3] [,4]  
## [1,] 1 0 0 2  
## [2,] 0 1 1 2  
## [3,] 0 0 1 3

mat[2,] <- mat[2,] - mat[3,] \* 1 ; mat #과정7

## [,1] [,2] [,3] [,4]  
## [1,] 1 0 0 2  
## [2,] 0 1 0 -1  
## [3,] 0 0 1 3