과제3 -4장

```
#1 일표본 t검정 함수
x <- c(2.7, 2.5, 2.5, 2.6, 2.5, 2.3, 3.3, 1.8, 2.7, 2.9, NA,
       2.9, 3.1, 2.4, 2.9, 3.2, 2.5, 3.2, 2.8, 2.6, 2.9, NA,
       1.8, 3.1, 2.6, 2.5, 2.7, 3.1, 2.7, 3.1, 2.7, 3.0, NA,
        2.3, 2.9, 2.5, 3.3, 2.0, 2.2, 2.9, 3.1, 3.2, 2.0,NA)
x <- na.omit(x)
mn =mean(x)
sigma = 0.397
alpha = 0.05
zstat=(mn-2.5)/(sigma/sqrt(40))
pval <- 1-pnorm(zstat)
mu0 <-0
```

```
listresult = list(mn =mean(x),sd = sd(x),
                   zstat=(mn-2.5)/(sigma/sqrt(40)),decision = if(pval < c(alpha)){
  print("Reject H0")
  }else {
    print("Do Not reject H0")},ci =c(mn-(1.96 %*% (sigma/sqrt(40))),mn+(1.96 %*% (sigma/sqrt(40
)))),sigma = 0.397,alpha = 0.05)
```

```
## [1] "Reject H0"
```

```
listresult
```

```
## $mn
## [1] 2.7
##
## $sd
## [1] 0.3967819
##
## $zstat
## [1] 3.186174
##
## $decision
## [1] "Reject H0"
##
## $ci
## [1] 2.576968 2.823032
##
## $sigma
## [1] 0.397
##
## $alpha
## [1] 0.05
```

```
#tips.csv처리
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
TIPS <- read.csv("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩tips.csv")
TIPS$sex= factor(TIPS$sex)
TIPS$smoker= factor(TIPS$smoker)
TIPS$day= factor(TIPS$day)
TIPS$time= factor(TIPS$time)
TIPS$tiprate = TIPS$tip /TIPS$total_bill
write.csv(TIPS, file="TIPS.csv")
```

```
#3.4장코드 실행
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students1.txt",header=T) # 파일 마지막 행에서 [Enter]를 누르지 않은 경우
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students2.txt",header=T) #파일 마지막 행에서 [Enter]를 누른 경우
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : int  100 80 90 95 100
```

```
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students1.txt",header=T,as.is=TRUE)
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : int  100 80 90 95 100
```

```
# 파일을 있는 형태 그대로 읽음. as.is=TRUE 면 문자를 문자로 읽음
```

```
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students3.txt",header=TRUE, stringsAsFactors =FALSE)
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name.   : chr  "강서준," "김도형," "박정원," "이상훈," ...
##  $ korean. : chr  "100," "90," "90," "100," ...
##  $ english.: chr  "90," "100," "95," "85," ...
##  $ math    : int  100 80 90 NA 100
```

```
# 파일을 읽을 때 문장을 요인으로 인식하지 않도록 설정.
```

```
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students3.txt",header=TRUE, sep=',',
stringsAsFactors =FALSE)
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : chr  " 100" " 80" " 90" " NA" ...
```

```
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students3.txt",header=TRUE, sep=',',
as.is=TRUE)
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : chr  " 100" " 80" " 90" " NA" ...
```

```
# 구분 기호는 쉼표(,), 첫 행은 header로 인식하여 파일을 있는 그대로 읽어들이면
# NA로 인해 math 요소가 문장으로 인식됨
```

```
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students3.txt",header=TRUE, sep=',',
as.is=TRUE, na.strings= 'NA')
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : chr  " 100" " 80" " 90" " NA" ...
```

```
# 'NA' 문장을 결측값 NA로 처리하라고 해도 처리가 안됨. 정확한 문장은 NA 앞에 빈 칸이 있어야 하
기 때문
```

```
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students3.txt",header=TRUE, sep=',',
as.is=TRUE, na.strings= ' NA')
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : int  100 80 90 NA 100
```

```
# 'NA'로 정확하게 입력하자 결측값 NA로 처리되면서 math 요소가 모두 숫자로 인식됨
```

```
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students3.txt",header=TRUE, sep=',', as.is=TRUE,strip.white=TRUE)
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : int  100 80 90 NA 100
```

```
# strip.white에서 빈칸을 제거하면 na.string의 기본값이 'NA'로 설정되어 math 요소가 모두 숫자로 인식됨.
```

```
# 첫 행이 header이므로 header 옵션을 지정할 필요가 없음
df <- read.csv("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students.csv")
df
```

```
##     name korean english math
## 1 강서준    100      90  100
## 2 김도형     90     100   80
## 3 박정원     90      95   90
## 4 이상훈    100      85   95
## 5 최건우     85     100  100
```

```
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : int  100 80 90 95 100
```

```
df$name <- as.character(df$name)
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : int  100 80 90 95 100
```

```
df <- read.csv("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students.csv",stringsAsFactors = FALSE
)
str(df)
```

```
## 'data.frame':    5 obs. of  4 variables:
##  $ name   : chr  "강서준" "김도형" "박정원" "이상훈" ...
##  $ korean : int  100 90 90 100 85
##  $ english: int  90 100 95 85 100
##  $ math   : int  100 80 90 95 100
```

```
# 파일을 읽을 때 문장을 요인으로 인식하지 않도록 설정함
```

```
df <- read.table("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students3.txt",header=TRUE, sep=',',
as.is=TRUE)
```

```
write.table(df,file="C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩output.txt")
# 문장에 큰따옴표가 표시됨.
```

```
test <- c(15, 20, 30, NA, 45)
test[test<40]
```

```
## [1] 15 20 30 NA
```

```
test[test%%3!= 0]
```

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

```
test[is.na(test)]
```

```
## [1] NA
```

```
test[!is.na(test)]
```

```
## [1] 15 20 30 45
```

```
test[test%%2==0 & !is.na(test)]
```

```
## [1] 20 30
```

```
DF <- data.frame(
 name = c('길동', '춘향', '철수'),
 age = c(30, 16, 21),
 gender = factor(c('M', 'F','M'))) # 데이터 프레임인 경우
DF
```

```
##   name age gender
## 1 길동  30      M
## 2 춘향  16      F
## 3 철수  21      M
```

```
DF[DF$gender=='F', ] # 성별이 여성인 행 추출
```

```
##   name age gender
## 2 춘향  16      F
```

```
DF[DF$age<30 & DF$gender=='M', ] # 30살 미만의 남성 행 추출
```

```
##   name age gender
## 3 철수  21      M
```

```
x <- 5
if(x%%2 == 0) {
 print('x는 짝수') # 조건식이 참
} else {
 print('x는 홀수') # 조건식이 거짓
}
```

```
## [1] "x는 홀수"
```

```
x <- -1
if(x>0) {
 print('x is a positive value.') # x가 0보다 크면 출력
} else if(x<0) {
 print('x is a negative value.') # 위 조건을 만족하지 않고 x가 0보다 작으면 출력
} else {
 print('x is zero.') # 위 조건을 모두 만족하지 않으면 출력
}
```

```
## [1] "x is a negative value."
```

```
x <- c(-5:5)
options(digits=3)
sqrt(x)
```

```
## Warning in sqrt(x): NaN이 생성되었습니다
```

```
##  [1]  NaN  NaN  NaN  NaN  NaN 0.00 1.00 1.41 1.73 2.00 2.24
```

```
sqrt(ifelse(x>=0, x, NA)) # NaN이 발생하지 않게 음수면 NA로 표시
```

```
##  [1]   NA   NA   NA   NA   NA 0.00 1.00 1.41 1.73 2.00 2.24
```

```
DF <- read.csv("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students.csv")
DF
```

```
##     name korean english math
## 1 강서준    100      90  100
## 2 김도형     90     100   80
## 3 박정원     90      95   90
## 4 이상훈    100      85   95
## 5 최건우     85     100  100
```

```
DF[, 2] = ifelse(DF[, 2]>= 0 & DF[, 2]<= 100, DF[, 2], NA)
DF[, 3] = ifelse(DF[, 3]>= 0 & DF[, 3]<= 100, DF[, 3], NA)
DF[, 4] = ifelse(DF[, 4]>= 0 & DF[, 4]<= 100, DF[, 4], NA)
DF # ifelse 문으로 2~4열 값 중 0~100 외의 값은 NA로 처리함
```

```
##     name korean english math
## 1 강서준    100      90  100
## 2 김도형     90     100   80
## 3 박정원     90      95   90
## 4 이상훈    100      85   95
## 5 최건우     85     100  100
```

```
# repeat 문을 이용해 1부터 10까지 숫자 증가시키기
i <- 1 # i의 시작값은 1
repeat {
 if(i>10) { # i가 10을 넘으면 반복을 중단(break)함
 break
 } else {
 print(i)
 i <- i+1 # i를 1 증가시킴.
 }
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

```
# while 문을 이용해 1부터 10까지 숫자 증가시키기
i <- 1 # i의 시작값은 1임.
while(i <= 10){ # i가 10 이하인 동안에 반복함
 print(i)
 i <- i+1 } #i를 1증가시킴.
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

```
# while 문을 이용해 구구단 2단 만들기
i <- 1
while(i<10) {
 print(paste(2, 'X', i, '=', 2*i))
 i <- i+1
}
```

```
## [1] "2 X 1 = 2"
## [1] "2 X 2 = 4"
## [1] "2 X 3 = 6"
## [1] "2 X 4 = 8"
## [1] "2 X 5 = 10"
## [1] "2 X 6 = 12"
## [1] "2 X 7 = 14"
## [1] "2 X 8 = 16"
## [1] "2 X 9 = 18"
```

```
# for 문을 이용한 1부터 10까지 숫자 증가시키기
for(i in 1:10) {
 print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

```
# for 문을 이용해 구구단 2단 만들기
for(i in 1:9) {
 print(paste(2, 'X', i, '=', 2*i))
}
```

```
## [1] "2 X 1 = 2"
## [1] "2 X 2 = 4"
## [1] "2 X 3 = 6"
## [1] "2 X 4 = 8"
## [1] "2 X 5 = 10"
## [1] "2 X 6 = 12"
## [1] "2 X 7 = 14"
## [1] "2 X 8 = 16"
## [1] "2 X 9 = 18"
```

```
# for 문을 이용해 구구단 2~9단 만들기
for(i in 2:9) {
 for(j in 1:9) {
 print(paste(i, 'X', j, '=', i*j))
 }}
```

```
## [1] "2 X 1 = 2"
## [1] "2 X 2 = 4"
## [1] "2 X 3 = 6"
## [1] "2 X 4 = 8"
## [1] "2 X 5 = 10"
## [1] "2 X 6 = 12"
## [1] "2 X 7 = 14"
## [1] "2 X 8 = 16"
## [1] "2 X 9 = 18"
## [1] "3 X 1 = 3"
## [1] "3 X 2 = 6"
## [1] "3 X 3 = 9"
## [1] "3 X 4 = 12"
## [1] "3 X 5 = 15"
## [1] "3 X 6 = 18"
## [1] "3 X 7 = 21"
## [1] "3 X 8 = 24"
## [1] "3 X 9 = 27"
## [1] "4 X 1 = 4"
## [1] "4 X 2 = 8"
## [1] "4 X 3 = 12"
## [1] "4 X 4 = 16"
## [1] "4 X 5 = 20"
## [1] "4 X 6 = 24"
## [1] "4 X 7 = 28"
## [1] "4 X 8 = 32"
## [1] "4 X 9 = 36"
## [1] "5 X 1 = 5"
## [1] "5 X 2 = 10"
## [1] "5 X 3 = 15"
## [1] "5 X 4 = 20"
## [1] "5 X 5 = 25"
## [1] "5 X 6 = 30"
## [1] "5 X 7 = 35"
## [1] "5 X 8 = 40"
## [1] "5 X 9 = 45"
## [1] "6 X 1 = 6"
## [1] "6 X 2 = 12"
## [1] "6 X 3 = 18"
## [1] "6 X 4 = 24"
## [1] "6 X 5 = 30"
## [1] "6 X 6 = 36"
## [1] "6 X 7 = 42"
## [1] "6 X 8 = 48"
## [1] "6 X 9 = 54"
## [1] "7 X 1 = 7"
## [1] "7 X 2 = 14"
## [1] "7 X 3 = 21"
## [1] "7 X 4 = 28"
## [1] "7 X 5 = 35"
## [1] "7 X 6 = 42"
## [1] "7 X 7 = 49"
## [1] "7 X 8 = 56"
## [1] "7 X 9 = 63"
## [1] "8 X 1 = 8"
## [1] "8 X 2 = 16"
## [1] "8 X 3 = 24"
## [1] "8 X 4 = 32"
## [1] "8 X 5 = 40"
## [1] "8 X 6 = 48"
## [1] "8 X 7 = 56"
## [1] "8 X 8 = 64"
## [1] "8 X 9 = 72"
## [1] "9 X 1 = 9"
## [1] "9 X 2 = 18"
## [1] "9 X 3 = 27"
## [1] "9 X 4 = 36"
## [1] "9 X 5 = 45"
## [1] "9 X 6 = 54"
## [1] "9 X 7 = 63"
## [1] "9 X 8 = 72"
## [1] "9 X 9 = 81"
```

```
# 1부터 10까지의 수 중 짝수만 출력하기
for(i in 1:10) {
 if(i%%2 == 0) {
 print(i)}}
```

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
```

```
# 1부터 10까지의 수 중 소수 출력하기
for(i in 1:10){
  check=0
  for (j in 1:i){
    if(i%%j ==0){
      check = check+1
      }
  }
if(check==2){
  print(i)
}
}
```

```
## [1] 2
## [1] 3
## [1] 5
## [1] 7
```

```
df <- read.csv("C:₩₩Users₩₩User₩₩Desktop₩₩data₩₩Sources₩₩students.csv")
df # 데이터에 100 초과 값과 음수 값이 포함되어 있음
```

```
##     name korean english math
## 1 강서준    100      90  100
## 2 김도형     90     100   80
## 3 박정원     90      95   90
## 4 이상훈    100      85   95
## 5 최건우     85     100  100
```

```
for(i in 2:4){
  df[,i] <- ifelse(df[,i]>=0 & df[,i]<=100,df[,i],NA)
}
df
```

```
##     name korean english math
## 1 강서준    100      90  100
## 2 김도형     90     100   80
## 3 박정원     90      95   90
## 4 이상훈    100      85   95
## 5 최건우     85     100  100
```

```
fact <- function(x) { # 함수의 이름은 fact, 입력은 x
 fa <- 1 # 계승값을 저장할 변수
 while(x>1) { # x가 1보다 큰 동안 반복
 fa <- fa*x # x 값을 fa에 곱한 후 fa에 다시 저장
 x <- x-1 # x 값을 1 감소
 }
 return(fa) # 최종 계산된 fa 반환
}
fact(5) # 5!을 계산한 결과 출력
```

```
## [1] 120
```

```
my.is.na<-function(x) {
  table(is.na(x))
}
```

```
my.is.na(airquality) # 이 결과는 table(is.na(airquality))와 같음.
```

```
##
## FALSE  TRUE
##   874    44
```

```
table(is.na(airquality))
```

```
##
## FALSE  TRUE
##   874    44
```

```
str(airquality) # airquality 데이터의 구조를 살펴봄.
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
# airquality 데이터에서 NA인 것은 TRUE, 아니면 FALSE로 나타냄. 데이터가 많아 head 함수로 추려냄.
head(is.na(airquality))
```

```
##      Ozone Solar.R  Wind  Temp Month   Day
## [1,] FALSE   FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE   FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE   FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE   FALSE FALSE FALSE FALSE FALSE
## [5,]  TRUE    TRUE FALSE FALSE FALSE FALSE
## [6,] FALSE    TRUE FALSE FALSE FALSE FALSE
```

```
table(is.na(airquality)) # NA가 총 44개 있음.
```

```
## 
## FALSE  TRUE 
##   874    44
```

```
table(is.na(airquality$Temp)) # Temp에는 NA가 없음을 확인함.
```

```
## 
## FALSE 
##   153
```

```
table(is.na(airquality$Ozone)) # Ozone에는 NA가 37개 발견됨.
```

```
## 
## FALSE  TRUE 
##   116    37
```

```
mean(airquality$Temp) # NA가 없는 Temp는 평균이 구해짐.
```

```
## [1] 77.9
```

```
mean(airquality$Ozone) # NA가 있는 Ozone은 평균이 NA로 나옴.
```

```
## [1] NA
```

```
air_narm = airquality[!is.na(airquality$Ozone), ] # Ozone 속성에서 NA가 없는 값만 추출함.
air_narm
```

```
##     Ozone Solar.R Wind Temp Month Day
## 1      41     190  7.4   67     5   1
## 2      36     118  8.0   72     5   2
## 3      12     149 12.6   74     5   3
## 4      18     313 11.5   62     5   4
## 6      28      NA 14.9   66     5   6
## 7      23     299  8.6   65     5   7
## 8      19      99 13.8   59     5   8
## 9       8      19 20.1   61     5   9
## 11      7      NA  6.9   74     5  11
## 12     16     256  9.7   69     5  12
## 13     11     290  9.2   66     5  13
## 14     14     274 10.9   68     5  14
## 15     18      65 13.2   58     5  15
## 16     14     334 11.5   64     5  16
## 17     34     307 12.0   66     5  17
## 18      6      78 18.4   57     5  18
## 19     30     322 11.5   68     5  19
## 20     11      44  9.7   62     5  20
## 21      1       8  9.7   59     5  21
## 22     11     320 16.6   73     5  22
## 23      4      25  9.7   61     5  23
## 24     32      92 12.0   61     5  24
## 28     23      13 12.0   67     5  28
## 29     45     252 14.9   81     5  29
## 30    115     223  5.7   79     5  30
## 31     37     279  7.4   76     5  31
## 38     29     127  9.7   82     6   7
## 40     71     291 13.8   90     6   9
## 41     39     323 11.5   87     6  10
## 44     23     148  8.0   82     6  13
## 47     21     191 14.9   77     6  16
## 48     37     284 20.7   72     6  17
## 49     20      37  9.2   65     6  18
## 50     12     120 11.5   73     6  19
## 51     13     137 10.3   76     6  20
## 62    135     269  4.1   84     7   1
## 63     49     248  9.2   85     7   2
## 64     32     236  9.2   81     7   3
## 66     64     175  4.6   83     7   5
## 67     40     314 10.9   83     7   6
## 68     77     276  5.1   88     7   7
## 69     97     267  6.3   92     7   8
## 70     97     272  5.7   92     7   9
## 71     85     175  7.4   89     7  10
## 73     10     264 14.3   73     7  12
## 74     27     175 14.9   81     7  13
## 76      7      48 14.3   80     7  15
## 77     48     260  6.9   81     7  16
## 78     35     274 10.3   82     7  17
## 79     61     285  6.3   84     7  18
## 80     79     187  5.1   87     7  19
## 81     63     220 11.5   85     7  20
## 82     16       7  6.9   74     7  21
## 85     80     294  8.6   86     7  24
## 86    108     223  8.0   85     7  25
## 87     20      81  8.6   82     7  26
## 88     52      82 12.0   86     7  27
## 89     82     213  7.4   88     7  28
## 90     50     275  7.4   86     7  29
## 91     64     253  7.4   83     7  30
## 92     59     254  9.2   81     7  31
## 93     39      83  6.9   81     8   1
## 94      9      24 13.8   81     8   2
## 95     16      77  7.4   82     8   3
## 96     78      NA  6.9   86     8   4
## 97     35      NA  7.4   85     8   5
## 98     66      NA  4.6   87     8   6
## 99    122     255  4.0   89     8   7
## 100    89     229 10.3   90     8   8
## 101   110     207  8.0   90     8   9
## 104    44     192 11.5   86     8  12
## 105    28     273 11.5   82     8  13
## 106    65     157  9.7   80     8  14
## 108    22      71 10.3   77     8  16
## 109    59      51  6.3   79     8  17
## 110    23     115  7.4   76     8  18
## 111    31     244 10.9   78     8  19
## 112    44     190 10.3   78     8  20
## 113    21     259 15.5   77     8  21
## 114     9      36 14.3   72     8  22
## 116    45     212  9.7   79     8  24
## 117   168     238  3.4   81     8  25
## 118    73     215  8.0   86     8  26
## 120    76     203  9.7   97     8  28
## 121   118     225  2.3   94     8  29
## 122    84     237  6.3   96     8  30
## 123    85     188  6.3   94     8  31
## 124    96     167  6.9   91     9   1
## 125    78     197  5.1   92     9   2
## 126    73     183  2.8   93     9   3
## 127    91     189  4.6   93     9   4
## 128    47      95  7.4   87     9   5
## 129    32      92 15.5   84     9   6
## 130    20     252 10.9   80     9   7
## 131    23     220 10.3   78     9   8
## 132    21     230 10.9   75     9   9
## 133    24     259  9.7   73     9  10
## 134    44     236 14.9   81     9  11
## 135    21     259 15.5   76     9  12
## 136    28     238  6.3   77     9  13
## 137     9      24 10.9   71     9  14
## 138    13     112 11.5   71     9  15
## 139    46     237  6.9   78     9  16
## 140    18     224 13.8   67     9  17
## 141    13      27 10.3   76     9  18
## 142    24     238 10.3   68     9  19
## 143    16     201  8.0   82     9  20
## 144    13     238 12.6   64     9  21
## 145    23      14  9.2   71     9  22
## 146    36     139 10.3   81     9  23
## 147     7      49 10.3   69     9  24
## 148    14      20 16.6   63     9  25
## 149    30     193  6.9   70     9  26
## 151    14     191 14.3   75     9  28
## 152    18     131  8.0   76     9  29
## 153    20     223 11.5   68     9  30
```

```
mean(air_narm$Ozone) # 결측값이 제거된 데이터에서는 mean 함수가 정상적으로 동작함.
```

```
## [1] 42.1
```

```
# na.omit 함수를 이용해 결측값 처리하기
air_narm1 = na.omit(airquality)
mean(air_narm1$Ozone)
```

```
## [1] 42.1
```

```
# 함수 속성인 na.rm을 이용해 결측값 처리하기
mean(airquality$Ozone, na.rm = T)
```

```
## [1] 42.1
```

```
table(is.na(airquality))
```

```
##
## FALSE  TRUE
##   874    44
```

```
table(is.na(airquality$Ozone))
```

```
##
## FALSE  TRUE
##   116    37
```

```
table(is.na(airquality$Solar.R))
```

```
##
## FALSE  TRUE
##   146     7
```

```
air_narm = airquality[!is.na(airquality$Ozone) & !is.na(airquality$Solar.R), ]
mean(air_narm$Ozone)
```

```
## [1] 42.1
```

```
patients = data.frame(name = c('환자1', '환자2', '환자3', '환자4', '환자5'), age = c(22, 20, 25
, 30, 27), gender=factor(c('M', 'F', 'M', 'K', 'F')), blood.type = factor(c('A', 'O', 'B', 'AB'
, 'C')))
patients
```

```
##    name age gender blood.type
## 1 환자1  22      M          A
## 2 환자2  20      F          O
## 3 환자3  25      M          B
## 4 환자4  30      K         AB
## 5 환자5  27      F          C
```

```
# 성별에서 이상값 제거
patients_outrm = patients[patients$gender=='M'|patients$gender=='F', ]
patients_outrm
```

```
##    name age gender blood.type
## 1 환자1  22      M          A
## 2 환자2  20      F          O
## 3 환자3  25      M          B
## 5 환자5  27      F          C
```

```
# 성별과 혈액형에서 이상값 제거
patients_outrm1 = patients[(patients$gender == 'M'|patients$gender == 'F') & (patients$blood.ty
pe ==
'A'|patients$blood.type == 'B'|patients$blood.type == 'O'|patients$blood.type == 'AB'), ]
patients_outrm1
```

```
##    name age gender blood.type
## 1 환자1  22      M          A
## 2 환자2  20      F          O
## 3 환자3  25      M          B
```

```
# 이상값이 포함된 환자 데이터
patients = data.frame(name = c('환자1', '환자2', '환자3', '환자4', '환자5'), age = c(22, 20, 25
, 30, 27), gender = c(1, 2, 1, 3, 2), blood.type = c(1, 3, 2, 4, 5))
patients
```

```
##    name age gender blood.type
## 1 환자1  22      1          1
## 2 환자2  20      2          3
## 3 환자3  25      1          2
## 4 환자4  30      3          4
## 5 환자5  27      2          5
```

```
# 성별에 있는 이상값을 결측값으로 변경
patients$gender = ifelse((patients$gender<1|patients$gender>2), NA, patients$gender)
patients
```

```
##    name age gender blood.type
## 1 환자1  22      1          1
## 2 환자2  20      2          3
## 3 환자3  25      1          2
## 4 환자4  30     NA          4
## 5 환자5  27      2          5
```

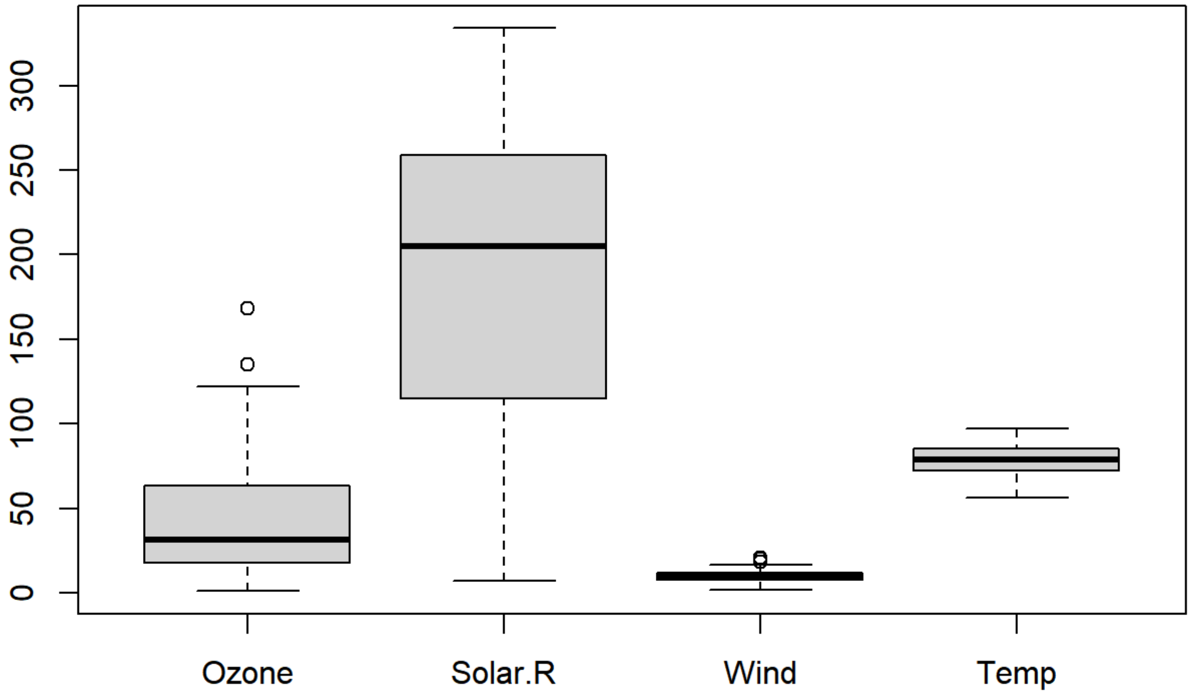
```
# 형액형에 있는 이상값도 결측값으로 변경
patients$blood.type = ifelse((patients$blood.type<1|patients$blood.type>4), NA, patients$blood.
type)
patients
```

```
##    name age gender blood.type
## 1 환자1  22      1          1
## 2 환자2  20      2          3
## 3 환자3  25      1          2
## 4 환자4  30     NA          4
## 5 환자5  27      2         NA
```

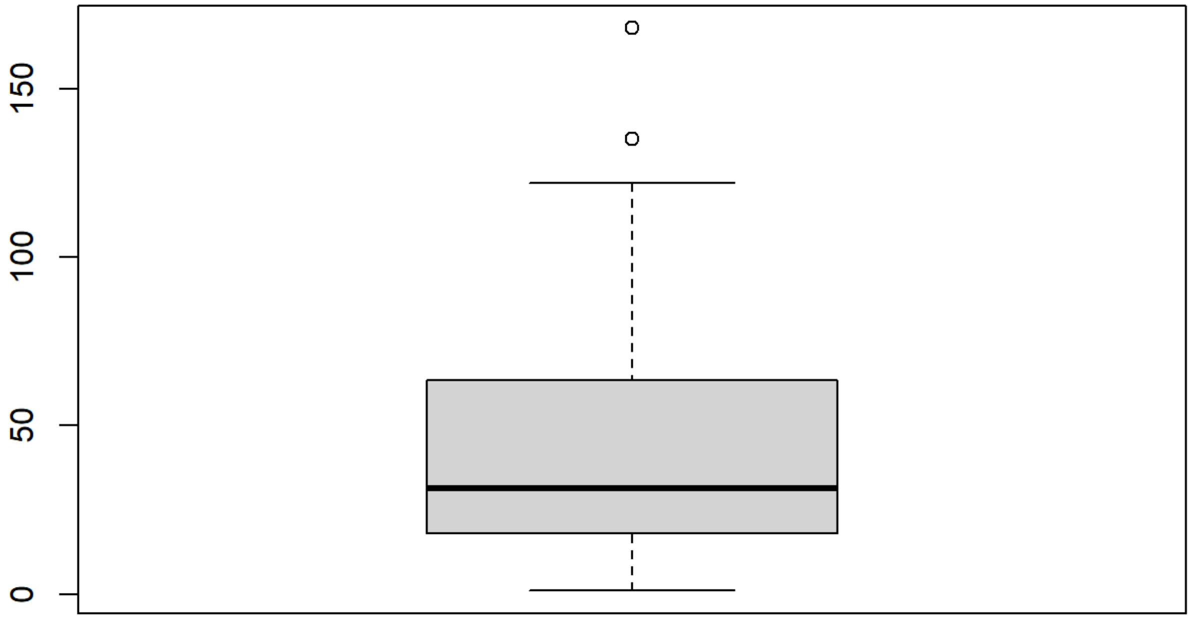
```
# 결측값을 모두 제거
patients[!is.na(patients$gender)&!is.na(patients$blood.type), ]
```

```
##    name age gender blood.type
## 1 환자1  22      1          1
## 2 환자2  20      2          3
## 3 환자3  25      1          2
```

```
boxplot(airquality[, c(1:4)]) # Ozone, Solar.R, Wind, Temp에 대한 boxplot
```



```
boxplot(airquality[, 1])$stats # Ozone의 boxplot 통계값 계산
```



```
##       [,1]
## [1,]   1.0
## [2,]  18.0
## [3,]  31.5
## [4,]  63.5
## [5,] 122.0
```

```
air = airquality # 임시 저장 변수로 airquality 데이터 복사
table(is.na(air$Ozone)) # Ozone의 현재 NA 개수 확인
```

```
## 
## FALSE  TRUE 
##   116    37
```

```
# 이상값을 NA로 변경
air$Ozone = ifelse(air$Ozone<1|air$Ozone>122, NA, air$Ozone)
table(is.na(air$Ozone)) # 이상값 처리 후 NA 개수 확인(2개 증가)
```

```
## 
## FALSE  TRUE 
##   114    39
```

```
# NA 제거
air_narm = air[!is.na(air$Ozone), ]
mean(air_narm$Ozone) # 이상값 두 개 제거로 is.na 함수를 이용한 결과보다 값이 줄어듦
```

```
## [1] 40.2
```