Rstudy\_week3

DKLee

2023-11-05

#dplyr 패키지

library(dplyr)

##   
## 다음의 패키지를 부착합니다: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

nrow(mtcars)

## [1] 32

str(mtcars)

## 'data.frame': 32 obs. of 11 variables:  
## $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...  
## $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...  
## $ disp: num 160 160 108 258 360 ...  
## $ hp : num 110 110 93 110 175 105 245 62 95 123 ...  
## $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...  
## $ wt : num 2.62 2.88 2.32 3.21 3.44 ...  
## $ qsec: num 16.5 17 18.6 19.4 17 ...  
## $ vs : num 0 0 1 1 0 1 0 1 1 1 ...  
## $ am : num 1 1 1 0 0 0 0 0 0 0 ...  
## $ gear: num 4 4 4 3 3 3 3 4 4 4 ...  
## $ carb: num 4 4 1 1 2 1 4 2 2 4 ...

# 조건에 맞는 데이터 추출

filter(mtcars, cyl==4)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2

filter(mtcars, cyl >= 6 & mpg > 20)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1

head(select(mtcars, am, gear))

## am gear  
## Mazda RX4 1 4  
## Mazda RX4 Wag 1 4  
## Datsun 710 1 4  
## Hornet 4 Drive 0 3  
## Hornet Sportabout 0 3  
## Valiant 0 3

# 데이터 정렬하기

head(arrange(mtcars, wt))

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1

# wt기준으로 오름차순 정렬.  
head(arrange(mtcars, mpg, desc(wt)))

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Lincoln Continental 10.4 8 460 215 3.00 5.424 17.82 0 0 3 4  
## Cadillac Fleetwood 10.4 8 472 205 2.93 5.250 17.98 0 0 3 4  
## Camaro Z28 13.3 8 350 245 3.73 3.840 15.41 0 0 3 4  
## Duster 360 14.3 8 360 245 3.21 3.570 15.84 0 0 3 4  
## Chrysler Imperial 14.7 8 440 230 3.23 5.345 17.42 0 0 3 4  
## Maserati Bora 15.0 8 301 335 3.54 3.570 14.60 0 1 5 8

# mpg 기준으로 오름차순 정렬 후, 같은 값일경우 wt 기준으로 내림차순 정렬.

# 데이터에 열 추가하기

mutate(mtcars, years = '1974')

## mpg cyl disp hp drat wt qsec vs am gear carb years  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4 1974  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4 1974  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1 1974  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1 1974  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2 1974  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1 1974  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4 1974  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2 1974  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2 1974  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4 1974  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4 1974  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3 1974  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3 1974  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3 1974  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4 1974  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4 1974  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4 1974  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1 1974  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2 1974  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1 1974  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1 1974  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2 1974  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2 1974  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4 1974  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2 1974  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1 1974  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2 1974  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2 1974  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4 1974  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6 1974  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8 1974  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2 1974

mutate(mtcars, mpg\_rank = rank(mpg))

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2  
## mpg\_rank  
## Mazda RX4 19.5  
## Mazda RX4 Wag 19.5  
## Datsun 710 24.5  
## Hornet 4 Drive 21.5  
## Hornet Sportabout 15.0  
## Valiant 14.0  
## Duster 360 4.0  
## Merc 240D 26.0  
## Merc 230 24.5  
## Merc 280 16.5  
## Merc 280C 13.0  
## Merc 450SE 11.0  
## Merc 450SL 12.0  
## Merc 450SLC 7.5  
## Cadillac Fleetwood 1.5  
## Lincoln Continental 1.5  
## Chrysler Imperial 5.0  
## Fiat 128 31.0  
## Honda Civic 29.5  
## Toyota Corolla 32.0  
## Toyota Corona 23.0  
## Dodge Challenger 9.0  
## AMC Javelin 7.5  
## Camaro Z28 3.0  
## Pontiac Firebird 16.5  
## Fiat X1-9 28.0  
## Porsche 914-2 27.0  
## Lotus Europa 29.5  
## Ford Pantera L 10.0  
## Ferrari Dino 18.0  
## Maserati Bora 6.0  
## Volvo 142E 21.5

# mpg의 데이터 순위를 열로 만듦.

# 중복 값 제거하기

distinct(mtcars, cyl)

## cyl  
## Mazda RX4 6  
## Datsun 710 4  
## Hornet Sportabout 8

# cly값이 겹치는 행 첫 행 제외하고 모두 제거.  
distinct(mtcars, gear)

## gear  
## Mazda RX4 4  
## Hornet 4 Drive 3  
## Porsche 914-2 5

distinct(mtcars, cyl, gear)

## cyl gear  
## Mazda RX4 6 4  
## Datsun 710 4 4  
## Hornet 4 Drive 6 3  
## Hornet Sportabout 8 3  
## Toyota Corona 4 3  
## Porsche 914-2 4 5  
## Ford Pantera L 8 5  
## Ferrari Dino 6 5

# 여러 열 지정시 각 열에서 중복값이 제거되는것이 아닌 지정한 열의 모든 값이  
# 동일한 행들이 제거됨.  
# 중복 값을 제거하면 해당 열이 몇 가지 관측치로 구성되어있는지 확인 가능.

# 데이터 요약하기

summarise(mtcars, cyl\_mean = mean(cyl), cyl\_min = min(cyl), cyl\_max = max(cyl))

## cyl\_mean cyl\_min cyl\_max  
## 1 6.1875 4 8

summarise(mtcars, mean(cyl), min(cyl), max(cyl))

## mean(cyl) min(cyl) max(cyl)  
## 1 6.1875 4 8

# 이름 지정해도 되고 안해도 됨.  
  
gr\_cyl = group\_by(mtcars, cyl)  
summarize(gr\_cyl, n())

## # A tibble: 3 × 2  
## cyl `n()`  
## <dbl> <int>  
## 1 4 11  
## 2 6 7  
## 3 8 14

# 그룹별로 요약하기  
  
summarise(gr\_cyl, n\_distinct(gear))

## # A tibble: 3 × 2  
## cyl `n\_distinct(gear)`  
## <dbl> <int>  
## 1 4 3  
## 2 6 3  
## 3 8 2

# cyl열 그룹에서 gear값이 중복인 데이터를 제외한 개수.

# 샘플 추출하기

sample\_n(mtcars, 10)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2

# 개수 기준으로 샘플 데이터 추출.  
sample\_frac(mtcars, 0.2)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1

# 전체 데이터의 0.2를 추출. 비율 기준.

# 파이프 연산자

group\_by(mtcars, cyl) %>% summarise(n())

## # A tibble: 3 × 2  
## cyl `n()`  
## <dbl> <int>  
## 1 4 11  
## 2 6 7  
## 3 8 14

# 새로운 변수에 저장할 필요 없이 데이터와 함수를 연결하여 사용 가능.  
mutate(mtcars, mpg\_rank = rank(mpg)) %>% arrange(mpg\_rank)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Cadillac Fleetwood 10.4 8 472.0 205 2.93 5.250 17.98 0 0 3 4  
## Lincoln Continental 10.4 8 460.0 215 3.00 5.424 17.82 0 0 3 4  
## Camaro Z28 13.3 8 350.0 245 3.73 3.840 15.41 0 0 3 4  
## Duster 360 14.3 8 360.0 245 3.21 3.570 15.84 0 0 3 4  
## Chrysler Imperial 14.7 8 440.0 230 3.23 5.345 17.42 0 0 3 4  
## Maserati Bora 15.0 8 301.0 335 3.54 3.570 14.60 0 1 5 8  
## Merc 450SLC 15.2 8 275.8 180 3.07 3.780 18.00 0 0 3 3  
## AMC Javelin 15.2 8 304.0 150 3.15 3.435 17.30 0 0 3 2  
## Dodge Challenger 15.5 8 318.0 150 2.76 3.520 16.87 0 0 3 2  
## Ford Pantera L 15.8 8 351.0 264 4.22 3.170 14.50 0 1 5 4  
## Merc 450SE 16.4 8 275.8 180 3.07 4.070 17.40 0 0 3 3  
## Merc 450SL 17.3 8 275.8 180 3.07 3.730 17.60 0 0 3 3  
## Merc 280C 17.8 6 167.6 123 3.92 3.440 18.90 1 0 4 4  
## Valiant 18.1 6 225.0 105 2.76 3.460 20.22 1 0 3 1  
## Hornet Sportabout 18.7 8 360.0 175 3.15 3.440 17.02 0 0 3 2  
## Merc 280 19.2 6 167.6 123 3.92 3.440 18.30 1 0 4 4  
## Pontiac Firebird 19.2 8 400.0 175 3.08 3.845 17.05 0 0 3 2  
## Ferrari Dino 19.7 6 145.0 175 3.62 2.770 15.50 0 1 5 6  
## Mazda RX4 21.0 6 160.0 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160.0 110 3.90 2.875 17.02 0 1 4 4  
## Hornet 4 Drive 21.4 6 258.0 110 3.08 3.215 19.44 1 0 3 1  
## Volvo 142E 21.4 4 121.0 109 4.11 2.780 18.60 1 1 4 2  
## Toyota Corona 21.5 4 120.1 97 3.70 2.465 20.01 1 0 3 1  
## Datsun 710 22.8 4 108.0 93 3.85 2.320 18.61 1 1 4 1  
## Merc 230 22.8 4 140.8 95 3.92 3.150 22.90 1 0 4 2  
## Merc 240D 24.4 4 146.7 62 3.69 3.190 20.00 1 0 4 2  
## Porsche 914-2 26.0 4 120.3 91 4.43 2.140 16.70 0 1 5 2  
## Fiat X1-9 27.3 4 79.0 66 4.08 1.935 18.90 1 1 4 1  
## Honda Civic 30.4 4 75.7 52 4.93 1.615 18.52 1 1 4 2  
## Lotus Europa 30.4 4 95.1 113 3.77 1.513 16.90 1 1 5 2  
## Fiat 128 32.4 4 78.7 66 4.08 2.200 19.47 1 1 4 1  
## Toyota Corolla 33.9 4 71.1 65 4.22 1.835 19.90 1 1 4 1  
## mpg\_rank  
## Cadillac Fleetwood 1.5  
## Lincoln Continental 1.5  
## Camaro Z28 3.0  
## Duster 360 4.0  
## Chrysler Imperial 5.0  
## Maserati Bora 6.0  
## Merc 450SLC 7.5  
## AMC Javelin 7.5  
## Dodge Challenger 9.0  
## Ford Pantera L 10.0  
## Merc 450SE 11.0  
## Merc 450SL 12.0  
## Merc 280C 13.0  
## Valiant 14.0  
## Hornet Sportabout 15.0  
## Merc 280 16.5  
## Pontiac Firebird 16.5  
## Ferrari Dino 18.0  
## Mazda RX4 19.5  
## Mazda RX4 Wag 19.5  
## Hornet 4 Drive 21.5  
## Volvo 142E 21.5  
## Toyota Corona 23.0  
## Datsun 710 24.5  
## Merc 230 24.5  
## Merc 240D 26.0  
## Porsche 914-2 27.0  
## Fiat X1-9 28.0  
## Honda Civic 29.5  
## Lotus Europa 29.5  
## Fiat 128 31.0  
## Toyota Corolla 32.0

#mp\_rank = mutate(mtcars, mpg\_rank = rank(mpg))  
#arrange(mp\_rank, mpg\_rank) 과 동일.

library(readxl)  
exdata1 = read\_excel("C:/Rstudy/Sample1.xlsx")

# 선택한 변수만 추출하기

exdata1 %>% select(ID)

## # A tibble: 20 × 1  
## ID  
## <dbl>  
## 1 1  
## 2 2  
## 3 3  
## 4 4  
## 5 5  
## 6 6  
## 7 7  
## 8 8  
## 9 9  
## 10 10  
## 11 11  
## 12 12  
## 13 13  
## 14 14  
## 15 15  
## 16 16  
## 17 17  
## 18 18  
## 19 19  
## 20 20

exdata1 %>% select(ID, AREA, Y21\_CNT)

## # A tibble: 20 × 3  
## ID AREA Y21\_CNT  
## <dbl> <chr> <dbl>  
## 1 1 서울 50  
## 2 2 경기 25  
## 3 3 제주 10  
## 4 4 서울 8  
## 5 5 서울 30  
## 6 6 서울 1  
## 7 7 경기 2  
## 8 8 서울 10  
## 9 9 서울 4  
## 10 10 경기 17  
## 11 11 서울 5  
## 12 12 제주 53  
## 13 13 경기 35  
## 14 14 제주 15  
## 15 15 경기 16  
## 16 16 서울 15  
## 17 17 경기 8  
## 18 18 서울 5  
## 19 19 제주 15  
## 20 20 제주 4

# 주로 열 추출에 사용됨.  
exdata1 %>% select(-AREA, -Y21\_CNT)

## # A tibble: 20 × 11  
## ID SEX AGE CAR\_YN Y21\_AMT Y21F\_AMT Y21O\_CNT Y20\_AMT Y20\_CNT Y20F\_AMT  
## <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 F 50 1 1300000 170000 25 1000000 40 30000  
## 2 2 M 40 1 450000 50000 10 700000 30 150000  
## 3 3 F 28 0 275000 7500 3 500000 7 10000  
## 4 4 M 50 0 2300000 50000 3 2500000 3 80000  
## 5 5 M 27 1 845000 130000 11 760000 28 170000  
## 6 6 F 23 0 42900 0 1 300000 6 195000  
## 7 7 F 56 0 150000 5000 1 130000 2 13000  
## 8 8 F 47 1 650000 45000 6 400000 7 200000  
## 9 9 M 20 0 930000 50000 3 250000 2 127000  
## 10 10 F 38 0 520000 11000 10 550000 16 248000  
## 11 11 M 35 0 150000 10000 3 490000 25 35000  
## 12 12 F 44 1 1150000 270000 37 1150000 48 350000  
## 13 13 F 60 0 550000 120000 10 800000 32 150000  
## 14 14 M 55 1 1050000 300000 5 2900000 18 350000  
## 15 15 F 46 1 600000 105000 4 1000000 20 200000  
## 16 16 F 32 1 530000 380000 7 1000000 43 500000  
## 17 17 M 30 1 250000 70000 6 400000 10 48000  
## 18 18 F 29 1 150000 7000 3 100000 5 25000  
## 19 19 F 27 0 300000 150000 10 320000 27 100000  
## 20 20 M 27 1 130000 38000 2 150000 30 130000  
## # ℹ 1 more variable: Y20O\_CNT <dbl>

# 특정 열 제외도 가능.

# 필요한 데이터만 추출하기

exdata1 %>% filter(AREA == '서울' & Y21\_CNT >= 10)

## # A tibble: 4 × 13  
## ID SEX AGE AREA CAR\_YN Y21\_AMT Y21\_CNT Y21F\_AMT Y21O\_CNT Y20\_AMT  
## <dbl> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 F 50 서울 1 1300000 50 170000 25 1000000  
## 2 5 M 27 서울 1 845000 30 130000 11 760000  
## 3 8 F 47 서울 1 650000 10 45000 6 400000  
## 4 16 F 32 서울 1 530000 15 380000 7 1000000  
## # ℹ 3 more variables: Y20\_CNT <dbl>, Y20F\_AMT <dbl>, Y20O\_CNT <dbl>

# 데이터 정렬하기

exdata1 %>% arrange(AGE)

## # A tibble: 20 × 13  
## ID SEX AGE AREA CAR\_YN Y21\_AMT Y21\_CNT Y21F\_AMT Y21O\_CNT Y20\_AMT  
## <dbl> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 9 M 20 서울 0 930000 4 50000 3 250000  
## 2 6 F 23 서울 0 42900 1 0 1 300000  
## 3 5 M 27 서울 1 845000 30 130000 11 760000  
## 4 19 F 27 제주 0 300000 15 150000 10 320000  
## 5 20 M 27 제주 1 130000 4 38000 2 150000  
## 6 3 F 28 제주 0 275000 10 7500 3 500000  
## 7 18 F 29 서울 1 150000 5 7000 3 100000  
## 8 17 M 30 경기 1 250000 8 70000 6 400000  
## 9 16 F 32 서울 1 530000 15 380000 7 1000000  
## 10 11 M 35 서울 0 150000 5 10000 3 490000  
## 11 10 F 38 경기 0 520000 17 11000 10 550000  
## 12 2 M 40 경기 1 450000 25 50000 10 700000  
## 13 12 F 44 제주 1 1150000 53 270000 37 1150000  
## 14 15 F 46 경기 1 600000 16 105000 4 1000000  
## 15 8 F 47 서울 1 650000 10 45000 6 400000  
## 16 1 F 50 서울 1 1300000 50 170000 25 1000000  
## 17 4 M 50 서울 0 2300000 8 50000 3 2500000  
## 18 14 M 55 제주 1 1050000 15 300000 5 2900000  
## 19 7 F 56 경기 0 150000 2 5000 1 130000  
## 20 13 F 60 경기 0 550000 35 120000 10 800000  
## # ℹ 3 more variables: Y20\_CNT <dbl>, Y20F\_AMT <dbl>, Y20O\_CNT <dbl>

# 기본 오름차순 정렬.  
exdata1 %>% arrange(desc(Y21\_AMT))

## # A tibble: 20 × 13  
## ID SEX AGE AREA CAR\_YN Y21\_AMT Y21\_CNT Y21F\_AMT Y21O\_CNT Y20\_AMT  
## <dbl> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 4 M 50 서울 0 2300000 8 50000 3 2500000  
## 2 1 F 50 서울 1 1300000 50 170000 25 1000000  
## 3 12 F 44 제주 1 1150000 53 270000 37 1150000  
## 4 14 M 55 제주 1 1050000 15 300000 5 2900000  
## 5 9 M 20 서울 0 930000 4 50000 3 250000  
## 6 5 M 27 서울 1 845000 30 130000 11 760000  
## 7 8 F 47 서울 1 650000 10 45000 6 400000  
## 8 15 F 46 경기 1 600000 16 105000 4 1000000  
## 9 13 F 60 경기 0 550000 35 120000 10 800000  
## 10 16 F 32 서울 1 530000 15 380000 7 1000000  
## 11 10 F 38 경기 0 520000 17 11000 10 550000  
## 12 2 M 40 경기 1 450000 25 50000 10 700000  
## 13 19 F 27 제주 0 300000 15 150000 10 320000  
## 14 3 F 28 제주 0 275000 10 7500 3 500000  
## 15 17 M 30 경기 1 250000 8 70000 6 400000  
## 16 7 F 56 경기 0 150000 2 5000 1 130000  
## 17 11 M 35 서울 0 150000 5 10000 3 490000  
## 18 18 F 29 서울 1 150000 5 7000 3 100000  
## 19 20 M 27 제주 1 130000 4 38000 2 150000  
## 20 6 F 23 서울 0 42900 1 0 1 300000  
## # ℹ 3 more variables: Y20\_CNT <dbl>, Y20F\_AMT <dbl>, Y20O\_CNT <dbl>

# 내림차순 정렬 옵션.  
exdata1 %>% arrange(AGE, desc(Y21\_AMT))

## # A tibble: 20 × 13  
## ID SEX AGE AREA CAR\_YN Y21\_AMT Y21\_CNT Y21F\_AMT Y21O\_CNT Y20\_AMT  
## <dbl> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 9 M 20 서울 0 930000 4 50000 3 250000  
## 2 6 F 23 서울 0 42900 1 0 1 300000  
## 3 5 M 27 서울 1 845000 30 130000 11 760000  
## 4 19 F 27 제주 0 300000 15 150000 10 320000  
## 5 20 M 27 제주 1 130000 4 38000 2 150000  
## 6 3 F 28 제주 0 275000 10 7500 3 500000  
## 7 18 F 29 서울 1 150000 5 7000 3 100000  
## 8 17 M 30 경기 1 250000 8 70000 6 400000  
## 9 16 F 32 서울 1 530000 15 380000 7 1000000  
## 10 11 M 35 서울 0 150000 5 10000 3 490000  
## 11 10 F 38 경기 0 520000 17 11000 10 550000  
## 12 2 M 40 경기 1 450000 25 50000 10 700000  
## 13 12 F 44 제주 1 1150000 53 270000 37 1150000  
## 14 15 F 46 경기 1 600000 16 105000 4 1000000  
## 15 8 F 47 서울 1 650000 10 45000 6 400000  
## 16 4 M 50 서울 0 2300000 8 50000 3 2500000  
## 17 1 F 50 서울 1 1300000 50 170000 25 1000000  
## 18 14 M 55 제주 1 1050000 15 300000 5 2900000  
## 19 7 F 56 경기 0 150000 2 5000 1 130000  
## 20 13 F 60 경기 0 550000 35 120000 10 800000  
## # ℹ 3 more variables: Y20\_CNT <dbl>, Y20F\_AMT <dbl>, Y20O\_CNT <dbl>

# 중첩정렬.

# 데이터 요약하기

exdata1 %>% summarise(TOT\_T21\_AMT = sum(Y21\_AMT))

## # A tibble: 1 × 1  
## TOT\_T21\_AMT  
## <dbl>  
## 1 12322900

# 구분 없이 Y21\_AMT의 총 합계 구하기.  
exdata1 %>% group\_by(AREA) %>% summarise(SUM = sum(Y21\_AMT))

## # A tibble: 3 × 2  
## AREA SUM  
## <chr> <dbl>  
## 1 경기 2520000  
## 2 서울 6897900  
## 3 제주 2905000

# AREA를 기준으로 각 지역 별 Y21\_AMT의 합계 구하기.

m\_history = read\_excel("C:/Rstudy/Sample2\_m\_history.xlsx")  
f\_history = read\_excel("C:/Rstudy/Sample3\_f\_history.xlsx")

# 데이터 세로 결합하기

exdata\_bindjoin = bind\_rows(m\_history, f\_history)  
View(exdata\_bindjoin)  
# axis = 0 방향으로 결합.

jeju\_y21 = read\_excel("C:/Rstudy/Sample4\_y21\_history.xlsx")  
jeju\_y20 = read\_excel("C:/Rstudy/Sample5\_y20\_history.xlsx")

# 데이터 가로 결합하기

left\_join(jeju\_y21, jeju\_y20, by = "ID")

## # A tibble: 8 × 8  
## ID SEX AGE AREA AMT21 Y21\_CNT AMT20 Y20\_CNT  
## <dbl> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 1 F 50 서울 1300000 50 100000 40  
## 2 2 M 40 경기 450000 25 700000 30  
## 3 4 M 50 서울 400000 8 125000 3  
## 4 5 M 27 서울 845000 30 760000 28  
## 5 7 F 56 경기 150000 2 130000 2  
## 6 8 F 47 서울 570000 10 400000 7  
## 7 9 M 20 인천 930000 4 NA NA  
## 8 10 F 38 경기 520000 17 550000 16

# 첫 번째 테이블 기준으로 가로 결합하기. right\_join도 있음.  
inner\_join(jeju\_y21, jeju\_y20, by = 'ID')

## # A tibble: 7 × 8  
## ID SEX AGE AREA AMT21 Y21\_CNT AMT20 Y20\_CNT  
## <dbl> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 1 F 50 서울 1300000 50 100000 40  
## 2 2 M 40 경기 450000 25 700000 30  
## 3 4 M 50 서울 400000 8 125000 3  
## 4 5 M 27 서울 845000 30 760000 28  
## 5 7 F 56 경기 150000 2 130000 2  
## 6 8 F 47 서울 570000 10 400000 7  
## 7 10 F 38 경기 520000 17 550000 16

# ID 변수가 동일할 때만 가로로 결합.  
full\_join(jeju\_y21, jeju\_y20, by = 'ID')

## # A tibble: 10 × 8  
## ID SEX AGE AREA AMT21 Y21\_CNT AMT20 Y20\_CNT  
## <dbl> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 1 F 50 서울 1300000 50 100000 40  
## 2 2 M 40 경기 450000 25 700000 30  
## 3 4 M 50 서울 400000 8 125000 3  
## 4 5 M 27 서울 845000 30 760000 28  
## 5 7 F 56 경기 150000 2 130000 2  
## 6 8 F 47 서울 570000 10 400000 7  
## 7 9 M 20 인천 930000 4 NA NA  
## 8 10 F 38 경기 520000 17 550000 16  
## 9 3 <NA> NA <NA> NA NA 50000 5  
## 10 6 <NA> NA <NA> NA NA 300000 6

# ID 기준으로 모든 데이터 가로 결합.

### 데이터 구조 변형하기

#install.packages('reshape2')  
library(reshape2)  
  
head(airquality)

## 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  
## 5 NA NA 14.3 56 5 5  
## 6 28 NA 14.9 66 5 6

names(airquality) = tolower(names(airquality))  
# 변수명 소문자로 바꾸기.

# 열을 행으로 바꾸기

melt\_test = melt(airquality)

## No id variables; using all as measure variables

melt\_test2 = melt(airquality, id.vars = c('month', 'wind'),  
 measure.vars = 'ozone')  
# id.vars : 식별자 변수, measure.vars : 반환하는 값  
head(melt\_test2)

## month wind variable value  
## 1 5 7.4 ozone 41  
## 2 5 8.0 ozone 36  
## 3 5 12.6 ozone 12  
## 4 5 11.5 ozone 18  
## 5 5 14.3 ozone NA  
## 6 5 14.9 ozone 28

# 행을 열로 바꾸기

aq\_melt = melt(airquality, id.vars = c('month','day'), na.rm = FALSE)  
View(aq\_melt)  
  
aq\_dcast = dcast(aq\_melt, month + day ~ variable)  
# 기준 변수 : month, day   
# 데이터 프레임으로 반환.  
head(aq\_dcast)

## month day ozone solar.r wind temp  
## 1 5 1 41 190 7.4 67  
## 2 5 2 36 118 8.0 72  
## 3 5 3 12 149 12.6 74  
## 4 5 4 18 313 11.5 62  
## 5 5 5 NA NA 14.3 56  
## 6 5 6 28 NA 14.9 66

acast(aq\_melt, day ~ month ~ variable)

## , , ozone  
##   
## 5 6 7 8 9  
## 1 41 NA 135 39 96  
## 2 36 NA 49 9 78  
## 3 12 NA 32 16 73  
## 4 18 NA NA 78 91  
## 5 NA NA 64 35 47  
## 6 28 NA 40 66 32  
## 7 23 29 77 122 20  
## 8 19 NA 97 89 23  
## 9 8 71 97 110 21  
## 10 NA 39 85 NA 24  
## 11 7 NA NA NA 44  
## 12 16 NA 10 44 21  
## 13 11 23 27 28 28  
## 14 14 NA NA 65 9  
## 15 18 NA 7 NA 13  
## 16 14 21 48 22 46  
## 17 34 37 35 59 18  
## 18 6 20 61 23 13  
## 19 30 12 79 31 24  
## 20 11 13 63 44 16  
## 21 1 NA 16 21 13  
## 22 11 NA NA 9 23  
## 23 4 NA NA NA 36  
## 24 32 NA 80 45 7  
## 25 NA NA 108 168 14  
## 26 NA NA 20 73 30  
## 27 NA NA 52 NA NA  
## 28 23 NA 82 76 14  
## 29 45 NA 50 118 18  
## 30 115 NA 64 84 20  
## 31 37 NA 59 85 NA  
##   
## , , solar.r  
##   
## 5 6 7 8 9  
## 1 190 286 269 83 167  
## 2 118 287 248 24 197  
## 3 149 242 236 77 183  
## 4 313 186 101 NA 189  
## 5 NA 220 175 NA 95  
## 6 NA 264 314 NA 92  
## 7 299 127 276 255 252  
## 8 99 273 267 229 220  
## 9 19 291 272 207 230  
## 10 194 323 175 222 259  
## 11 NA 259 139 137 236  
## 12 256 250 264 192 259  
## 13 290 148 175 273 238  
## 14 274 332 291 157 24  
## 15 65 322 48 64 112  
## 16 334 191 260 71 237  
## 17 307 284 274 51 224  
## 18 78 37 285 115 27  
## 19 322 120 187 244 238  
## 20 44 137 220 190 201  
## 21 8 150 7 259 238  
## 22 320 59 258 36 14  
## 23 25 91 295 255 139  
## 24 92 250 294 212 49  
## 25 66 135 223 238 20  
## 26 266 127 81 215 193  
## 27 NA 47 82 153 145  
## 28 13 98 213 203 191  
## 29 252 31 275 225 131  
## 30 223 138 253 237 223  
## 31 279 NA 254 188 NA  
##   
## , , wind  
##   
## 5 6 7 8 9  
## 1 7.4 8.6 4.1 6.9 6.9  
## 2 8.0 9.7 9.2 13.8 5.1  
## 3 12.6 16.1 9.2 7.4 2.8  
## 4 11.5 9.2 10.9 6.9 4.6  
## 5 14.3 8.6 4.6 7.4 7.4  
## 6 14.9 14.3 10.9 4.6 15.5  
## 7 8.6 9.7 5.1 4.0 10.9  
## 8 13.8 6.9 6.3 10.3 10.3  
## 9 20.1 13.8 5.7 8.0 10.9  
## 10 8.6 11.5 7.4 8.6 9.7  
## 11 6.9 10.9 8.6 11.5 14.9  
## 12 9.7 9.2 14.3 11.5 15.5  
## 13 9.2 8.0 14.9 11.5 6.3  
## 14 10.9 13.8 14.9 9.7 10.9  
## 15 13.2 11.5 14.3 11.5 11.5  
## 16 11.5 14.9 6.9 10.3 6.9  
## 17 12.0 20.7 10.3 6.3 13.8  
## 18 18.4 9.2 6.3 7.4 10.3  
## 19 11.5 11.5 5.1 10.9 10.3  
## 20 9.7 10.3 11.5 10.3 8.0  
## 21 9.7 6.3 6.9 15.5 12.6  
## 22 16.6 1.7 9.7 14.3 9.2  
## 23 9.7 4.6 11.5 12.6 10.3  
## 24 12.0 6.3 8.6 9.7 10.3  
## 25 16.6 8.0 8.0 3.4 16.6  
## 26 14.9 8.0 8.6 8.0 6.9  
## 27 8.0 10.3 12.0 5.7 13.2  
## 28 12.0 11.5 7.4 9.7 14.3  
## 29 14.9 14.9 7.4 2.3 8.0  
## 30 5.7 8.0 7.4 6.3 11.5  
## 31 7.4 NA 9.2 6.3 NA  
##   
## , , temp  
##   
## 5 6 7 8 9  
## 1 67 78 84 81 91  
## 2 72 74 85 81 92  
## 3 74 67 81 82 93  
## 4 62 84 84 86 93  
## 5 56 85 83 85 87  
## 6 66 79 83 87 84  
## 7 65 82 88 89 80  
## 8 59 87 92 90 78  
## 9 61 90 92 90 75  
## 10 69 87 89 92 73  
## 11 74 93 82 86 81  
## 12 69 92 73 86 76  
## 13 66 82 81 82 77  
## 14 68 80 91 80 71  
## 15 58 79 80 79 71  
## 16 64 77 81 77 78  
## 17 66 72 82 79 67  
## 18 57 65 84 76 76  
## 19 68 73 87 78 68  
## 20 62 76 85 78 82  
## 21 59 77 74 77 64  
## 22 73 76 81 72 71  
## 23 61 76 82 75 81  
## 24 61 76 86 79 69  
## 25 57 75 85 81 63  
## 26 58 78 82 86 70  
## 27 57 73 86 88 77  
## 28 67 80 88 97 75  
## 29 81 77 86 94 76  
## 30 79 83 83 96 68  
## 31 76 NA 81 94 NA

# 기준 변수 : day, month  
# array로 반환.

# 데이터 요약하기

acast(aq\_melt, month ~ variable, mean)

## ozone solar.r wind temp  
## 5 NA NA 11.622581 65.54839  
## 6 NA 190.1667 10.266667 79.10000  
## 7 NA 216.4839 8.941935 83.90323  
## 8 NA NA 8.793548 83.96774  
## 9 NA 167.4333 10.180000 76.90000

# month 기준으로 각 변수별 평균 구하기.  
dcast(aq\_melt, month ~ variable, sum)

## month ozone solar.r wind temp  
## 1 5 NA NA 360.3 2032  
## 2 6 NA 5705 308.0 2373  
## 3 7 NA 6711 277.2 2601  
## 4 8 NA NA 272.6 2603  
## 5 9 NA 5023 305.4 2307

### 데이터 정제하기 - 이상치 및 결측치 처리

# 결측치 확인하기

x = c(1,2,NA,4,5)  
sum(x)

## [1] NA

is.na(x)

## [1] FALSE FALSE TRUE FALSE FALSE

table(is.na(x))

##   
## FALSE TRUE   
## 4 1

# 결측치 제외하기

sum(x, na.rm = TRUE)

## [1] 12

# 결측치 개수 확인하기

sum(is.na(airquality))

## [1] 44

# 전체 데이터의 결측치 개수.  
table(is.na(airquality))

##   
## FALSE TRUE   
## 874 44

colSums(is.na(airquality))

## ozone solar.r wind temp month day   
## 37 7 0 0 0 0

# 변수 별 결측치 개수.

# 결측치 제거하기

na.omit(airquality)

## 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  
## 7 23 299 8.6 65 5 7  
## 8 19 99 13.8 59 5 8  
## 9 8 19 20.1 61 5 9  
## 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  
## 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

# 결측치가 있는 행을 제거함.

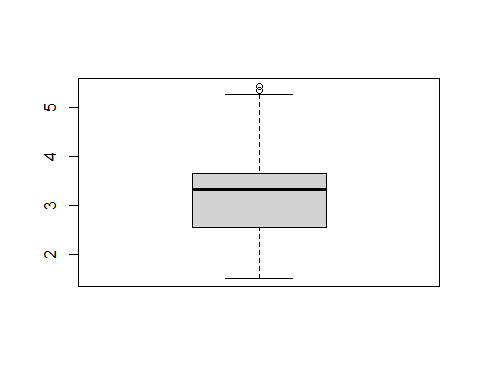
# 결측치 대체하기

airquality[is.na(airquality)] = 0  
colSums(is.na(airquality))

## ozone solar.r wind temp month day   
## 0 0 0 0 0 0

# 이상치 확인하기

boxplot(mtcars$wt)  
boxplot(mtcars$wt)$stats



## [,1]  
## [1,] 1.5130  
## [2,] 2.5425  
## [3,] 3.3250  
## [4,] 3.6500  
## [5,] 5.2500

# boxplot에 그려진 값들을 기술통계량으로 확인.

# 이상치 처리하기

mtcars$wt = ifelse(mtcars$wt > 5.25, NA, mtcars$wt)  
# 조건, 조건이 참일때, 조건이 거짓일때.  
mtcars$wt

## [1] 2.620 2.875 2.320 3.215 3.440 3.460 3.570 3.190 3.150 3.440 3.440 4.070  
## [13] 3.730 3.780 5.250 NA NA 2.200 1.615 1.835 2.465 3.520 3.435 3.840  
## [25] 3.845 1.935 2.140 1.513 3.170 2.770 3.570 2.780