데이터 전처리 : 원하는 형태로 데이터 가공

조건에 맞는 데이터(행) 추출하기

```
#install.packages("dplyr")
library(dplyr)
exam <- read.csv("csv_exam.csv")
exam

#filter
exam %>%  filter(math > 50)
exam %>%  filter(class==1 & english > 50)
exam %>%  filter(math>=90 | english>=90)
exam %>%  filter(class==1 | class==2 | class==3) # 목록에 해당하는 행 추출
exam %>% filter(class %in% c(1,2,3)) # %in% 이용

class1 <- exam %>% filter(class==1) # 추출한 행으로 데이터 만들기

mean(class1$math)
```

데이터 전처리 : 원하는 형태로 데이터 가공

필요한 변수 추출하기

```
#select
exam %>%  select(math)
exam %>%  select(class, math, english) # 여러 변수 추출
exam %>%  select(-english) # 특정 변수 제거

# dplyr 함수 조합
#class가 1인 행을 추출한 후 english 추출
exam %>%  filter(class==1) %>%  select(english)
# 일부 변수를 앞부분 6행까지만 출력
exam %>%  select(id, math) %>%  head()
```

데이터 전처리 : 원하는 형태로 데이터 가공

데이터 정렬하기

```
#arrange
exam %>% arrange(math) #오름차순 정렬
exam %>% arrange(desc(math)) # 내림차순 정렬
exam %>% arrange(class, math) # 두 변수 정렬
```

파생변수 추가하기

```
#mutate
# 총합을 구한 후 total 변수 추가
exam %>%  mutate(total=math+english+science) %>%  head(10)

# 추가한 total로 정렬
exam %>%   mutate(total=math+english+science) %>%
  arrange(total) %>%  head(10)

# 조건에 맞는 데이터 변수 추가
exam %>%  mutate(test=ifelse(science>=60, "pass", "fail")) %>%   head()
```

데이터 전처리 : 원하는 형태로 데이터 가공

```
그룹 나누기
#group_by

exam %>%  group_by(class)

그룹별로 요약하기
#summarise
exam %>% summarise(mean_math=mean(math))

exam %>%
  group_by(class) %>%
  summarise(mean_math=mean(math), # math평균
            sum_math=sum(math), # math 합계
            median_math=median(math), # math 중앙값
            n=n()) # 학생수
```

데이터 전처리 : 원하는 형태로 데이터 가공

```
데이터 합치기(가로)
#left_join
test1 <- data.frame(id=c(1,2,3,4,5),
                    midterm=c(60,80,75,90,85))
test2 <- data.frame(id=c(1,2,3,4,5),
                    final=c(70,83,65,95,80))
total <- left_join(test1, test2, by="id") # id 기준으로 합쳐 total에 할당

# 다른 데이터 활용해 변수 추가
name <- data.frame(class=c(1,2,3,4,5),
                   teacher=c("kim","lee","park","choi","jang"))
exam_new <- left_join(exam, name, by="class") # class 기준으로 병합


데이터 합치기(세로)
#bind_rows
group_a <- data.frame(id=c(1,2,3,4,5),
                      test=c(60,70,80,90,85))
group_b <- data.frame(id=c(6,7,8,9,10),
                      test=c(70,83,65,95,80))
group_all <- bind_rows(group_a, group_b)
```

문제[2]

[ggplot2 팩키지의 mpg 데이터를 이용]

1. 자동차 배기량에 따라 고속도로 연비가 다른지 알아보려고 한다. displ(배기량)이 4 이하인 자동차와 5 이상인 자동차 중 어떤 자동차의 hwy(고속도로 연비)가 평균적으로 더 높은지 비교하시오.

2. “chevrolet", "ford", "honda" 자동차의 고속도로 연비 평균을 알아보려고 한다. 이 회사들의 자동차를 추출한 뒤 hwy 전체 평균을 구하시오.

3. mpg 데이터는 11 개 변수로 구성되어 있다. 이 중 일부만 추출해서 분석에 활용하려고 한다. mpg 데이터에서 class(자동차 종류), cty(도시 연비) 변수를 추출해 새로운 데이터를 만드시오. 새로 만든 데이터의 일부를 출력해서 두 변수로만 구성되어 있는지 확인하시오.

문제[2]

[ggplot2 팩키지의 mpg 데이터를 이용]

4. "audi"에서 생산한 자동차 중에 어떤 자동차 모델의 hwy(고속도로 연비)가 높은지 알아보려고 한다. "audi"에서 생산한 자동차 중 hwy가 1~5위에 해당하는 자동차의 데이터를 출력하시오.

5. mpg 데이터는 연비를 나타내는 변수가 hwy(고속도로 연비), cty(도시 연비) 두 종류로 분리되어 있다. 두 변수를 각각 활용하는 대신 하나의 통합 연비 변수를 만들어 분석하려고 한다.
   1. mpg 데이터 복사본을 만들고, cty 와 hwy 를 더한 '합산 연비 변수'를 추가하시오.
   2. 앞에서 만든 '합산 연비 변수'를 2 로 나눠 '평균 연비 변수'를 추가하시오.
   3. '평균 연비 변수'가 가장 높은 자동차 3 종의 데이터를 출력하시오.
   4. 1~3 번 문제를 해결할 수 있는 하나로 연결된 dplyr 구문을 만들어 출력하시오. (데이터는 복사본 대신 mpg 원본을 이용)

6. mpg 데이터의 class 는 "suv", "compact" 등 자동차를 특징에 따라 일곱 종류로 분류한 변수이다. 어떤 차종의 연비가 높은지 비교해보려고 한다. class 별 cty 평균을 구해보시오.