Chapter 6

dplyr을 이용한 데이터 전처리

데이터 전처리

데이터 전처리(Preprocessing) - dplyr 패키지

함수	기능
filter()	행 추출
select()	열(변수) 추출
arrange()	정렬
mutate()	변수 추가
summarise()	통계치 산출
group_by()	집단별로 나누기
left_join()	데이터 합치기(열)
bind_rows()	데이터 합치기(행)

데이터 전처리 filter()

dplyr 패키지 로드 & 데이터 준비

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

##
 id class math english science
1  1     1   50    98     50
2  2     1   60    97     60
3  3     1   45    86     78
4  4     1   30    98     58
5  5     2   25    80     65
6  6     2   50    89     98
7  7     2   80    90     45
8  8     2   90    78     25
9  9     3   20    98     15
10 10    3   50    98     45
```

데이터 전처리 filter()

filter를 이용한 행 추출

```
data1 <- filter(data, class == 1) #class가 1인 학생
data1

data2 <- filter(data, math >= 50) #수학점수가 50이상인 학생
data2

data3 <- filter(data, math >= 80 & english >= 80) #수학, 영어가 80이상인 학생
data3

data4 <- filter(data, math %in% c(50:60) ) #매칭확인
data4

data5 <- filter(data, class %in% c(1,3,5) ) #1 3 5반
```

파이프라인 %>%

파이프라인으로 작업처리 하기

```
ex1 <- filter(data, class != 1) #클래스가 1이아닌 학생
ex1
ex2 <- filter(ex1, math >= 50 ) #수학이 50점 이상인 학생
ex2
result <- filter(ex2, id %% 2 == 0) #id가 짝수인 학생(나머지)
result

#위와 동일한 결과
result <- data %>%
          filter(class != 1 ) %>%
          filter(math >= 50) %>%
          filter(id %% 2 == 0)
```

[참고] 단축키 [Ctrl+Shit+M]으로 %>% 기호 입력

R연산 기호

R에서 연산기호

논리 연산자	기능
<	작다
<=	작거나 같다
>	크다
>=	크거나 같다
==	같다
!=	같지 않다
|	또는
&	그리고
%in%	매칭 확인

산술 연산자	기능
+	더하기
-	빼기
*	곱하기
/	나누기(소수 몫)
^ , **	제곱
%/%	나누기(정수 몫)
%%	나눗셈의 나머지

문제

ggplot2에 있는 mpg 데이터를 이용해 분석 문제를 해결해 보세요.

- Q1. 자동차 배기량에 따라 고속도로 연비가 다른지 알아보려고 합니다. displ(배기량)이 4 이하인 자동차와 5 이상인 자동차 중 어떤 자동차의 hwy(고속도로 연비)가 평균적으로 더 높은지 파이프라인을 이용해서 알아보세요.

- Q2. 자동차 제조 회사에 따라 도시 연비가 다른지 알아보려고 합니다. "audi" 제조년월이 2000년 이상인 데이터의 cty 합계, 평균을 구하세요

- Q3. "chevrolet", "ford", "honda" 자동차의 고속도로 연비 평균을 알아보려고 합니다. 이 회사들의 자동차를 추출한 뒤 hwy 전체 평균을 구해보세요.

데이터 전처리 select()

select를 이용한 열 추출

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

select(data,  id)

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

id	class	english	science
1	2	98	50
2	1	97	60
3	2	86	78
4	1	98	58
5	1	80	65
6	2	89	98

➡

class	english
2	98
1	97
2	86
1	98
1	80
2	89

데이터 전처리 select()

select를 이용한 열 추출 & 가독성 높이기

```
select(data, id)

select(data, id, math, english) #여러 행 셀렉트

select(data, -class) #행 제외

select(data, -class, -english) #여러행 제외

#파이프라인
data %>%
  filter(class == 1) %>%      # class 가 1 인 행 추출
  select(english)             # english 추출
  head(10)                    # 10열만 보기
```

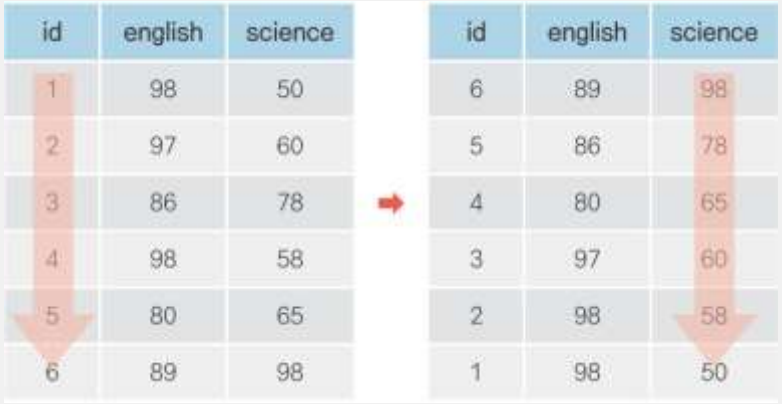
데이터 전처리 arrange()

arrange를 이용한 정렬

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

arrange(data, id) #수학기준 오름차순

   id class math english science
1   9     3   20      98      15
2   5     2   25      80      65
3   4     1   30      98      58
4   3     1   45      86      78
5  12     3   45      85      32
6  13     4   46      98      65
7  14     4   48      87      12
8   1     1   50      98      50
9   6     2   50      89      98
10 10     3   50      98      45
……
```



id	english	science
1	98	50
2	97	60
3	86	78
4	98	58
5	80	65
6	89	98

id	english	science
6	89	98
5	86	78
4	80	65
3	97	60
2	98	58
1	98	50

데이터 전처리 arrange()

arrange를 이용한 내림차순 정렬

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

arrange(data, desc(math)) #수학기준 내림차순

arrange(data, desc(class)) #클래스기준 내림차순

arrange(data, desc(class), id) #클래스기준 내림차순, 아이디 오름차순


#파이프라인
data %>% arrange(class, desc(math)) # 클래스 오름차순 수학 내림차순
```

데이터 전처리 mutate()

mutate()를 이용한 새로운 행 추가

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

mutate(data, total = math + english + science) # 합계변수 추가
```

```
   id class math english science total
1   1     1   50      98      50   198
2   2     1   60      97      60   217
3   3     1   45      86      78   209
4   4     1   30      98      58   186
5   5     2   25      80      65   170
6   6     2   50      89      98   237
7   7     2   80      90      45   215
8   8     2   90      78      25   193
9   9     3   20      98      15   133
10 10     3   50      98      45   193
……
```

id	english	science
1	98	50
2	97	60
3	86	78
4	98	58
5	80	65
6	89	98

➡

id	english	science	total
1	98	50	148
2	97	60	157
3	86	78	164
4	98	58	156
5	80	65	145
6	89	98	187

데이터 전처리 mutate()

mutate()를 이용한 새로운 여러 행 추가

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

mutate(data, total = math + english + science,
             avg = (math + english + science)/3 ) #합계, 평균 추가
```

```
   id class math english science total      avg
1   1     1   50      98      50   198 66.00000
2   2     1   60      97      60   217 72.33333
3   3     1   45      86      78   209 69.66667
4   4     1   30      98      58   186 62.00000
5   5     2   25      80      65   170 56.66667
6   6     2   50      89      98   237 79.00000
7   7     2   80      90      45   215 71.66667
8   8     2   90      78      25   193 64.33333
9   9     3   20      98      15   133 44.33333
10 10     3   50      98      45   193 64.33333
……
```

데이터 전처리 mutate()

mutate()를 이용한 새로운 여러 행 추가

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

# 파생변수로 새로운 데이터 만들기
data1 <- mutate(data, total = math + english + science,
                      avg = (math + english + science)/3 )
data2 <- mutate(data1, result = ifelse(avg >= 60, "pass", "fail" ) )
data2


#파이프라인으로 한번에 쓰기
result <- data %>%
         mutate(total = math + english + science,
                avg = (math + english + science)/3, 2)  %>%
         mutate(result = ifelse(avg > 60, "pass", "fail")) %>%
         arrange( desc(avg) ) %>%
         head(10)
```

문제

ggIplot2에 있는 mpg 데이터를 사용합니다.

- Q1. mpg데이터에서 class(자동차 종류), cty(도시 연비) 변수를 추출해 새로운 데이터를 만들고 class(자동차 종류)가 "suv"인 자동차와 "compact"인 자동차 중 어떤 자동차의 cty(도시 연비)만 추출합니다.

  - 파이프라인으로 한번에 처리하세요.

```
      class cty
1   compact  18
2   compact  21
3   compact  20
4   compact  21
5   compact  16
6   compact  18
7   compact  18
8   compact  18
9   compact  16
10  compact  20
```

- Q2. audi에서 생산한 자동차 중에 중 hwy가 1~5위에 해당하는 자동차의 (제조사, 모델, 년도, hwy)데이터만 출력하세요.

  - 파이프라인으로 한번에 처리하세요.

```
  manufacturer       model year hwy
1         audi          a4 2008  31
2         audi          a4 2008  30
3         audi          a4 1999  29
4         audi          a4 1999  29
5         audi a4 quattro 2008  28
```

문제

gglplot2에 있는 mpg 데이터는 연비를 나타내는 변수가 hwy(고속도로 연비), cty(도시 연비) 두 종류로 분리되어 있습니다.

- Q1. mpg 데이터 복사본을 만들고, cty 와 hwy 를 더한 '합산 연비 변수'를 추가.
- Q2. 앞에서 만든 '합산 연비 변수'를 2 로 나눠 '평균 연비 변수'를 추가.
- Q3. '평균 연비 변수'가 가장 높은 자동차 5순위 데이터를 출력.
- Q4. 원본 데이터를 이용해서 1~3 번 문제에 더하여 avg가 35이상이면 high, 35미만이면 row를 추가하는 파이프라인으로 한번에 처리하는 구문을 완성하세요.

```
  manufacturer      model displ year cyl      trans drv cty hwy fl      class total  avg result
1   volkswagen new beetle   1.9 1999   4 manual(m5)   f  35  44  d subcompact    79 39.5   high
2   volkswagen      jetta   1.9 1999   4 manual(m5)   f  33  44  d    compact    77 38.5   high
3   volkswagen new beetle   1.9 1999   4   auto(l4)   f  29  41  d subcompact    70 35.0   high
4       toyota    corolla   1.8 2008   4 manual(m5)   f  28  37  r    compact    65 32.5    row
5        honda      civic   1.6 1999   4 manual(m5)   f  28  33  r subcompact    61 30.5    row
```

데이터 전처리 summarise() , group_by()

summarise() 집단별로 요약해서 추출하기

전체를 요약하여 보여주고 list형태로 반환 합니다.

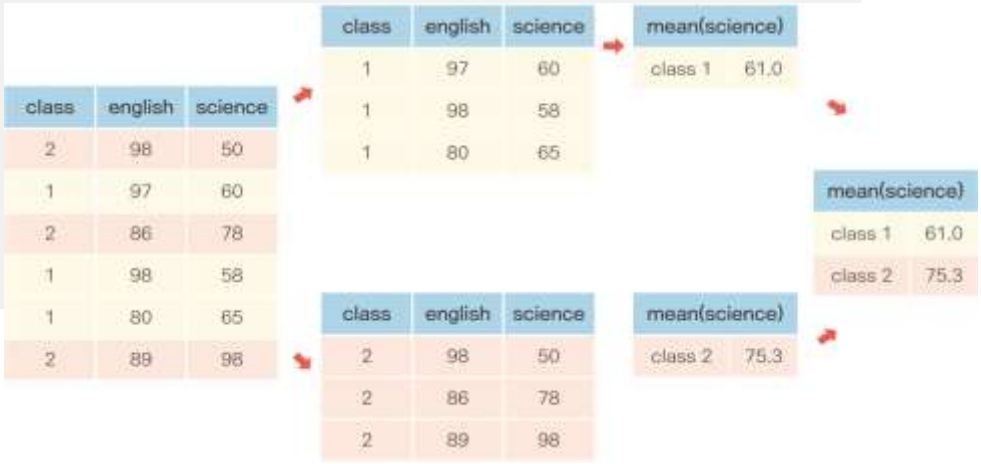
```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

summarise(data, mean_math = mean(math) ) # math로 평균 산출

##
 mean_math
1    57.45

summarise(data, mean_math = mean(math), #평균
                sum_math = sum(math), #합계
                count = n() ) # 빈도수

##
 mean_math  sum_math  count
1      57.45       1149     20
```



데이터 전처리 summarise() , group_by()

자주 사용하는 요약통계량 함수

함수	의미
mean()	평균
sd()	표준편차
sum()	합계
median()	중앙값
min()	최솟값
max()	최댓값
n()	빈도

데이터 전처리 summarise() , group_by()

group_by() 로 요약하기 전에 그룹핑 하기

summarise() 가 전체를 요약할 때 사용하는 반해 group_by() 는 그룹별 데이터를 요약할 때 사용합니다.
즉, 데이터를 지정한 조건에 따라 그룹으로 묶어 주는 역할을 합니다.

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

group_by(data, class) # data를 class별로 그룹핑
summarise(data1, math_sum = sum(math)) # class별 수학점수 합계
```

	class	math_sum
	<int>	<int>
1	1	185
2	2	245
3	3	180
4	4	227
5	5	312

데이터 전처리 summarise() , group_by()

파이프라인으로 한번에 요약통계 구하기

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

# 파이프라인으로 한번에 작성하기
data %>%
  group_by(class) %>%
  summarise(math_avg = mean(math),
            eng_avg = mean(english),
            sci_avg = mean(science),
            total = n() )
```

	class	math_avg	eng_avg	sci_avg	total
	<int>	<dbl>	<dbl>	<dbl>	<int>
1	1	46.2	94.8	61.5	4
2	2	61.2	84.2	58.2	4
3	3	45	86.5	39.2	4
4	4	56.8	84.8	55	4
5	5	78	74.2	83.2	4

데이터 전처리 summarise() , group_by()

group_by() 집단별 집단 처리

```
library(dplyr)
data <- read.csv("csv_exam.csv")
data

# 제조사별 구동방식 도로연비
mpg %>%
  group_by(manufacturer, drv) %>%
  summarise( mean_cty = mean(cty) )
```

```
# A tibble: 22 x 3
# Groups:   manufacturer [15]
   manufacturer drv   mean_cty
   <chr>        <chr>    <dbl>
 1 audi         4         16.8
 2 audi         f         18.9
 3 chevrolet    4         12.5
 4 chevrolet    f         18.8
 5 chevrolet    r         14.1
 6 dodge        4         12
 7 dodge        f         15.8
 8 ford         4         13.3
 9 ford         r         14.8
10 honda        f         24.4
```

데이터 전처리 left_join()

데이터프레임 가로 합치기 left_join()

필요한 데이터프레임을 가로로 합쳐서 리스트형태로 반환합니다. (행이 안맞으면 NA처리됩니다)

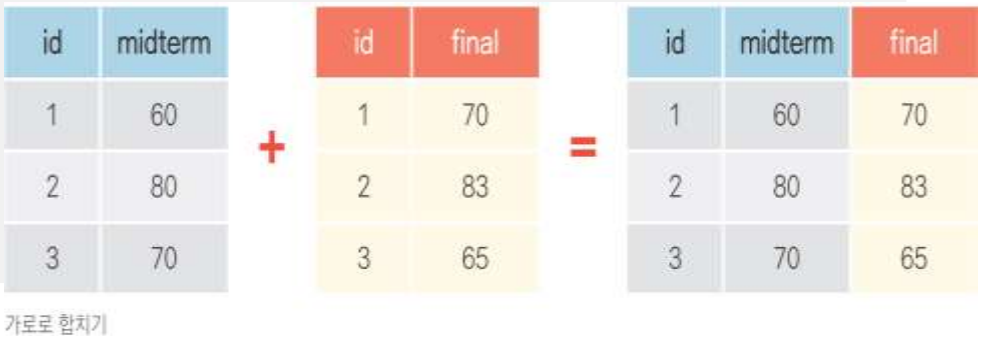
```
library(dplyr)

name = data.frame(class = c(1:5),
                  teacher = c("hong", "kim", "park", "lee", "choi") )

job = data.frame(teacher = c("hong", "kim", "park", "lee", "choi"),
                 job = c("math", "kor", "science", "eng", "society"))


left_join(name, job, by = "teacher") # 열이름을 by로 연결합니다

##
  class teacher subject
1     1    hong    math
2     2     kim     kor
3     3    park science
4     4     lee     eng
5     5    choi society
```



[주의] by에 변수명을 지정할 때 변수명 앞 뒤에 겹따옴표 입력

데이터 전처리 left_join()

데이터프레임 가로 합치기 left_join()

연결할 열이름이 다르다면 아래와 같이 강제 연결할 수 있습니다. (행이 안맞으면 NA처리됩니다)

```
library(dplyr)

name = data.frame(class = c(1:5),
                  teacher = c("hong", "kim", "park", "lee", "choi") )

job = data.frame(ttt = c("hong", "kim", "park", "lee", "choi"),
                 subject = c("math", "kor", "science", "eng", "society"))


left_join(name, job, by = c("teacher"="ttt") )
```

[주의] by에 변수명을 지정할 때 변수명 앞 뒤에 겹따옴표 입력

데이터 전처리 bind_rows()

데이터프레임 세로 합차기bind_rows()

연결할 열이름이 다르다면 아래와 같이 강제 연결할 수 있습니다. (열이 안맞으면 NA처리됩니다)

```
library(dplyr)

a <- data.frame(id = c(1, 2, 3, 4, 5),
                test = c(60, 80, 70, 90, 85))

b <- data.frame(id = c(6, 7, 8, 9, 10),
                test = c(70, 83, 65, 95, 80))

bind_rows(a,b) # 세로로 합치기

##
 id test
1   1   60
2   2   80
3   3   70
4   4   90
5   5   85
6   6   70
7   7   83
8   8   65
9   9   95
10 10   80
```

문제

mpg데이터를 이용합니다.

Q01
회사별로 "suv" 자동차의 도시 및 고속도로 통합 연비 평균을 구해 내림차순으로 정렬하고, 1~10위까지 출력하세요

Q02
mpg데이터의 class는 자동차 특징에 따라 분류된 변수입니다.
class별 cty평균을 구하고 높은 순으로 정렬해 출력하세요.

Q03
mpg데이터의 hwy 평균이 가장 높은 제조사 3곳을 출력하세요.

Q04
어떤 회사가 compact(경차) 를 많이 생산하는지 확인하려 합니다.
class가 compact(경차) 인 제조사별 차종 수를 내림차순 정렬해 출력하세요
힌트: class가 compact인 행 데이터를 먼저 추출

문제

mpg데이터의 if변수는 자동차 연료입니다.
아래 같은 표를 생성하세요.

fl	kind	price
c	CNG	2.35
d	diesel	2.38
e	ethanol	2.11
p	premium	2.76
r	regular	2.22

Q1
위 표를 mpg데이터에 left_join하고, 새로운 데이터를 만들어 냅니다.

그 이후에 파이프라인을 사용해서
model, fl, kind, price 데이터만 추출한 후에 앞부분 10행만 출력하세요

종합 문제

ggplot2에 존재하는 midwest데이터를 사용합니다 (미국 주 437개 지역의 인구 통계를 담고 있는 데이터 입니다)

Q1

midwest데이터를 데이터프레임으로 가져오고 구조를 확인하세요

Q2

popadults 는 해당 지역의 성인 인구, poptotal 은 전체 인구를 나타냅니다. midwest 데이터에
전체 인구 대비 성년 인구 백분율(adult_of_percent)
전체 인구 대비 미성년 인구 백분율(young_of_percent) 변수를 추가하세요.
전체 인구 대비 성년 인구 백분율공식 = (성인인구/전체인구) * 100

Q3

아래 등급표에 따라 grade변수를 추가하고, 미성년인구 백분율이 가장 높은 상위 5개
county(지역), 미성년백분율, grade를 출력하세요.

Q4

popasian은 해당 지역의 아시아인 인구를 나타냅니다. '전체 인구 대비 아시아인 인구 백분율' 변수를 추가하고,
하위 10개 지역의 state(주), county(지역명), 아시아인 인구 백분율을 출력하세요.

	county	young_of_percent	grade
1	ISABELLA	51.50	large
2	MENOMINEE	50.59	large
3	ATHENS	49.32	large
4	MECOSTA	49.06	large
5	MONROE	47.36	large

종합 문제

Q5

popasian은 해당 지역의 아시아인 인구입니다.

전체 인구 대비 아시아인 인구 백분율변수를 추가하고, 하위 10개 지역의 state(주), county(지역명), 아시아인 인구 백분율을 출력하세요.

	state	county	asian_percent
1	WI	MENOMINEE	0.00000000
2	IN	BENTON	0.01059210
3	IN	CARROLL	0.01594981
4	OH	VINTON	0.02703190
5	WI	IRON	0.03250447
6	IL	SCOTT	0.05315379
7	IN	CLAY	0.06071645
8	MI	OSCODA	0.06375925
9	OH	PERRY	0.06654625
10	IL	PIATT	0.07074865

Chapter 6

수고하셨습니다