05 dplyr를 이용한 데이터 처리 ¶

학습내용

- dplyr 패키지의 함수를 알아보기

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

In [1]:

```
library(dplyr)
exam <- read.csv("D:₩₩dataset₩₩R_Doit₩₩csv_exam.csv")
exam
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

id	class	math	english	science
1	1	50	98	50
2	1	60	97	60
3	1	45	86	78
4	1	30	98	58
5	2	25	80	65
6	2	50	89	98
7	2	80	90	45
8	2	90	78	25
9	3	20	98	15
10	3	50	98	45
11	3	65	65	65
12	3	45	85	32
13	4	46	98	65
14	4	48	87	12
15	4	75	56	78
16	4	58	98	65
17	5	65	68	98
18	5	80	78	90
19	5	89	68	87
20	5	78	83	58

In [2]:

```
names(exam)
```

'id' 'class' 'math' 'english' 'science'

exam에서 class가 1인 값만 가져오기

- filter(조건)
- exam %>% filter(class==1)

In [3]:

```
exam %>% filter(class==1)
```

id	class	math	english	science
1	1	50	98	50
2	1	60	97	60
3	1	45	86	78
4	1	30	98	58

(ex) 5-1 실습해 보기

- filter를 이용하여 3반인 경우 출력해보기
- filter를 이용 1반이 아닌 경우 행 보기
- filter를 이용 수학점수가 평균 이상인 친구 출력해 보기

In [4]:

```
exam %>% filter(class==1 & math >=50)   # 1반이면서 50점 이상출력
```

id	class	math	english	science
1	1	50	98	50
2	1	60	97	60

In [5]:

```
exam %>% filter(class==1 | math >=50)   # 1반이거나 50점 이상출력
```

id	class	math	english	science
1	1	50	98	50
2	1	60	97	60
3	1	45	86	78
4	1	30	98	58
6	2	50	89	98
7	2	80	90	45
8	2	90	78	25
10	3	50	98	45
11	3	65	65	65
15	4	75	56	78
16	4	58	98	65
17	5	65	68	98
18	5	80	78	90
19	5	89	68	87
20	5	78	83	58

1,3,5반에 해당되는 친구 출력 %in% 이용

- filter (컬럼명 %in% c(1,3,5))

In [6]:

```
exam %>% filter( class %in% c(1,3,5) )
```

id	class	math	english	science
1	1	50	98	50
2	1	60	97	60
3	1	45	86	78
4	1	30	98	58
9	3	20	98	15
10	3	50	98	45
11	3	65	65	65
12	3	45	85	32
17	5	65	68	98
18	5	80	78	90
19	5	89	68	87
20	5	78	83	58

실습과제 5-2 - p133

5-2 필요한 변수만 추출하기

- select(변수명)

In [7]:

```
names(exam)
```

'id' 'class' 'math' 'english' 'science'

In [8]:

```
exam %>% select(english)
```

english
98
97
86
98
80
89
90
78
98
98
65
85
98
87
56
98
68
78
68
83

In [9]:

```
exam %>% select(math, english, science)
```

math	english	science
50	98	50
60	97	60
45	86	78
30	98	58
25	80	65
50	89	98
80	90	45
90	78	25
20	98	15
50	98	45
65	65	65
45	85	32
46	98	65
48	87	12
75	56	78
58	98	65
65	68	98
80	78	90
89	68	87
78	83	58

변수 제외

In [10]:

```
exam %>% select(-english)
```

id	class	math	science
1	1	50	50
2	1	60	60
3	1	45	78
4	1	30	58
5	2	25	65
6	2	50	98
7	2	80	45
8	2	90	25
9	3	20	15
10	3	50	45
11	3	65	65
12	3	45	32
13	4	46	65
14	4	48	12
15	4	75	78
16	4	58	65
17	5	65	98
18	5	80	90
19	5	89	87
20	5	78	58

In [11]:

```
exam %>% select(-math, -english)
```

id	class	science
1	1	50
2	1	60
3	1	78
4	1	58
5	2	65
6	2	98
7	2	45
8	2	25
9	3	15
10	3	45
11	3	65
12	3	32
13	4	65
14	4	12
15	4	78
16	4	65
17	5	98
18	5	90
19	5	87
20	5	58

filter()와 select() 합치기

1반인 친구들의 math, english, science의 점수

In [12]:

```
names(exam)
```

'id' 'class' 'math' 'english' 'science'

In [13]:

```
exam %>% filter(class==1) %>% select(math, english, science)
```

math	english	science
50	98	50
60	97	60
45	86	78
30	98	58

1반의 친구들의 5행만 보기

In [14]:

```
exam %>% filter(class==1) %>% head(5)
```

id	class	math	english	science
1	1	50	98	50
2	1	60	97	60
3	1	45	86	78
4	1	30	98	58

더해보기 p138

5-3 순서대로 정렬해 보기

- arrange(기준컬럼)

In [15]:

```
exam %>% arrange(math)
```

id	class	math	english	science
9	3	20	98	15
5	2	25	80	65
4	1	30	98	58
3	1	45	86	78
12	3	45	85	32
13	4	46	98	65
14	4	48	87	12
1	1	50	98	50
6	2	50	89	98
10	3	50	98	45
16	4	58	98	65
2	1	60	97	60
11	3	65	65	65
17	5	65	68	98
15	4	75	56	78
20	5	78	83	58
7	2	80	90	45
18	5	80	78	90
19	5	89	68	87
8	2	90	78	25

In [16]:

```
exam %>% arrange(math) %>% head
```

id	class	math	english	science
9	3	20	98	15
5	2	25	80	65
4	1	30	98	58
3	1	45	86	78
12	3	45	85	32
13	4	46	98	65

In [17]:

```
### 합계를 구해서 이를 이용하여 정렬하기
exam$total <- exam$math + exam$english + exam$science
head(exam, 10)
```

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

In [18]:

```
exam %>% arrange(desc(total))  # 내림 차순 정렬
```

id	class	math	english	science	total
18	5	80	78	90	248
19	5	89	68	87	244
6	2	50	89	98	237
17	5	65	68	98	231
16	4	58	98	65	221
20	5	78	83	58	219
2	1	60	97	60	217
7	2	80	90	45	215
3	1	45	86	78	209
13	4	46	98	65	209
15	4	75	56	78	209
1	1	50	98	50	198
11	3	65	65	65	195
8	2	90	78	25	193
10	3	50	98	45	193
4	1	30	98	58	186
5	2	25	80	65	170
12	3	45	85	32	162
14	4	48	87	12	147
9	3	20	98	15	133

실습과제 5-3

- mpg 데이터를 이용해서 분석 문제를 해결해 보시오.
- "audi"에서 생산한 자동차 중에 어떤 자동차 모델의 hwy(고속도로 연비)가 높은지 알아보려고 합니다. "audi"에서 생산한 자동차 중 hwy가 1~5위에 해당하는 자동차의 데이터를 출력해보자. (쉽게 배우는 데이터 분석 p141 참조)

5-4 파생변수 추가하기

- mutate(변수명)
- exam %>% mutate(total=math+english+science) %>% head

In [19]:

```
exam %>% mutate(total=math+english+science) %>% head
```

id	class	math	english	science	total
1	1	50	98	50	198
2	1	60	97	60	217
3	1	45	86	78	209
4	1	30	98	58	186
5	2	25	80	65	170
6	2	50	89	98	237

여러개의 변수를 생성하기

In [20]:

```
exam %>% mutate(total=math+english+science,
                mean = (math+english+science)/3 ) %>% head
```

id	class	math	english	science	total	mean
1	1	50	98	50	198	66.00000
2	1	60	97	60	217	72.33333
3	1	45	86	78	209	69.66667
4	1	30	98	58	186	62.00000
5	2	25	80	65	170	56.66667
6	2	50	89	98	237	79.00000

In [21]:

```
summary(exam$total)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  133.0   191.2   209.0   201.8   219.5   248.0 
```

total이 중앙값을 넘었을 경우 pass, 아니면 fail로 하자

In [22]:

```
exam %>% mutate(total_pf=ifelse(total >209, "pass", "fail")) %>% head
```

id	class	math	english	science	total	total_pf
1	1	50	98	50	198	fail
2	1	60	97	60	217	pass
3	1	45	86	78	209	fail
4	1	30	98	58	186	fail
5	2	25	80	65	170	fail
6	2	50	89	98	237	pass

이를 확인 후, total_pf로 정렬하기

In [23]:

```
exam %>% mutate(total_pf=ifelse(total >209, "pass", "fail")) %>% arrange(desc(total_pf))
```

id	class	math	english	science	total	total_pf
2	1	60	97	60	217	pass
6	2	50	89	98	237	pass
7	2	80	90	45	215	pass
16	4	58	98	65	221	pass
17	5	65	68	98	231	pass
18	5	80	78	90	248	pass
19	5	89	68	87	244	pass
20	5	78	83	58	219	pass
1	1	50	98	50	198	fail
3	1	45	86	78	209	fail
4	1	30	98	58	186	fail
5	2	25	80	65	170	fail
8	2	90	78	25	193	fail
9	3	20	98	15	133	fail
10	3	50	98	45	193	fail
11	3	65	65	65	195	fail
12	3	45	85	32	162	fail
13	4	46	98	65	209	fail
14	4	48	87	12	147	fail
15	4	75	56	78	209	fail

5-5 집단별로 요약하기

- group_by()
- summarise()

exam의 math, english, science의 평균을 구해보자

- summarise(변수명 = mean(변수명))

In [24]:

```
exam %>% summarise(mean_math = mean(math))
```

mean_math
57.45

In [25]:

```
exam %>% summarise(mean_math = mean(math),
                   mean_eng = mean(english),
                   mean_sci = mean(science))
```

mean_math	mean_eng	mean_sci
57.45	84.9	59.45

집단별로 분리해서 위의 내용을 출력해 보자.

- 그룹으로 묶고 이에 대한 요약값을 구하기
- 데이터 셋 %>% group_by(변수명) %>% summarise(...)

In [26]:

```
exam %>% group_by(class) %>% summarise(mean_math = mean(math),
                   mean_eng = mean(english),
                   mean_sci = mean(science))
```

class	mean_math	mean_eng	mean_sci
1	46.25	94.75	61.50
2	61.25	84.25	58.25
3	45.00	86.50	39.25
4	56.75	84.75	55.00
5	78.00	74.25	83.25

In [27]:

```
exam %>% group_by(class) %>%
         summarise(mean_math = mean(math),
                   mean_eng = mean(english),
                   mean_sci = mean(science),
                   n=n())
```

class	mean_math	mean_eng	mean_sci	n
1	46.25	94.75	61.50	4
2	61.25	84.25	58.25	4
3	45.00	86.50	39.25	4
4	56.75	84.75	55.00	4
5	78.00	74.25	83.25	4

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

mpg 데이터 셋 불러오기

In [28]:

```
library(ggplot2)
```

In [29]:

```
mpg <- ggplot2::mpg
```

In [30]:

```
head(mpg)
```

manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact

제조사별 구동방식(drv)별 cty(도시연비)의 평균 산출하기

- drv(구동방식)
  - 4 사륜 구동
  - f 전륜 구동
  - r 후륜 구동

In [31]:

```
mpg %>% group_by(manufacturer, drv) %>%
     summarise(mean_cty = mean(cty)) %>%
     head(10)
```

manufacturer	drv	mean_cty
audi	4	16.81818
audi	f	18.85714
chevrolet	4	12.50000
chevrolet	f	18.80000
chevrolet	r	14.10000
dodge	4	12.00000
dodge	f	15.81818
ford	4	13.30769
ford	r	14.75000
honda	f	24.44444

실습과제 5-4

- 회사별로 분리한다.(group_by)
- class에서 suv 추출한다.(행) - filter()
- total(통합 연비) 파생변수 생성 - mutate()
- total(통합 연비) 의 평균 mean_total를 산출 - summarise()
- arrange() 내림 차순의 정렬
- 1~5위까지 : head()

In [32]:

```
mpg %>%
  group_by(manufacturer) %>%
  filter(class=="suv") %>%
  mutate(tot = (cty + hwy)/2) %>%
  summarise(mean_tot = mean(tot)) %>%
  arrange(desc(mean_tot)) %>% head(5)
```

manufacturer	mean_tot
subaru	21.91667
toyota	16.31250
nissan	15.87500
mercury	15.62500
jeep	15.56250

더해보기 p150

5-6 데이터 합치기

- dplyr:left_join()
- bind_rows()

In [33]:

```
kor <- data.frame(id=c(1,2,3,4,5),
                  kor=c(80,80,70,90,100))
math <- data.frame(id=c(1,2,3,4,5),
                   math=c(100,80,90,70,60))
eng <- data.frame(id=c(1,2,3,4,6),
                  eng=c(100,70,50,60,90))
```

- left_join(데이터셋1, 데이터셋2, by='컬럼명')

In [34]:

```
total <- left_join(kor, math, by='id')
total
```

id	kor	math
1	80	100
2	80	80
3	70	90
4	90	70
5	100	60

In [35]:

```
total_all <- left_join(total, eng, by='id')
total_all
```

id	kor	math	eng
1	80	100	100
2	80	80	70
3	70	90	50
4	90	70	60
5	100	60	NA

In [36]:

```
total_all2 <- left_join(eng, total, by='id')
total_all2
```

id	eng	kor	math
1	100	80	100
2	70	80	80
3	50	70	90
4	60	90	70
6	90	NA	NA

세로(행) 합치기, 행 추가

In [37]:

```
kor1 <- data.frame(id=c(1,2,3,4,5),
                   kor=c(80,80,70,90,100))
kor2 <- data.frame(id=c(6,7,8,9,10),
                   kor=c(80,80,70,90,100))
```

In [38]:

```
kor_all <- bind_rows(kor1, kor2)
```

In [39]:

```
kor_all
```

id	kor
1	80
2	80
3	70
4	90
5	100
6	80
7	80
8	70
9	90
10	100

변수가 kor로 동일하여 가능, 하지만 다를 경우는 어떻게 해야 하나? rename()를 이용하여 동일하게 한 이후에 합치기 가능