WeDataLab
데이터컨설팅 데이터아키텍처

3. dplyr 패키지를 이용한 데이터 전처리

dplyr 로 가공하기

dplyr 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)

dplyr 로 가공하기

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

# exam에서 class가 1인 경우만 추출해 출력

exam %>% filter(class == 1)

exam %>% filter(class !=1) # 1반이 아닌경우

exam %>% filter(math > 50) # 수학 점수가 50점을 초과한 경우

exam %>% filter(english >=80) # 영어 점수가 80 이상인

# 1반이면서 수학 점수가 50 이상인 경우
exam %>% filter(class == 1 & math >= 50)

# 수학 점수가 90점 이상이거나 영어 점수가 90점 이상인
exam %>% filter(math >= 90 | english >= 90)

# 1, 3, 5반에 해당하면 추출
exam %>% filter(class == 1 | class == 3 | class == 5
exam %>% filter(class %in% c(1,3,5))

class1 <- exam %>% filter(class == 1) # class가 1인 행 추출, class1에 할당

mean(class1$math) # 1반 수학 점수 평균 구하기
```

	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
11	11	3	65	65	65
12	12	3	45	85	32
13	13	4	46	98	65
14	14	4	48	87	12
15	15	4	75	56	78
16	16	4	58	98	65
17	17	5	65	68	98
18	18	5	80	78	90
19	19	5	89	68	87
20	20	5	78	83	58

dplyr 로 가공하기

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

# exam에서 class가 1인 경우만 추출해 출력

exam %>% filter(class == 1)

exam %>% filter(class !=1) # 1반이 아닌경우

exam %>% filter(math > 50) # 수학 점수가 50점을 초과한 경우

exam %>% filter(english >=80) # 영어 점수가 80 이상인 경우

# 1반이면서 수학 점수가 50 이상인 경우
exam %>% filter(class == 1 & math >= 50)

# 수학 점수가 90점 이상이거나 영어 점수가 90점 이상
exam %>% filter(math >= 90 | english >= 90)

# 1, 3, 5반에 해당하면 추출
exam %>% filter(class == 1 | class == 3 | class == 5)
exam %>% filter(class %in% c(1,3,5))

class1 <- exam %>% filter(class == 1) # class가 1인 행 추출, class1에 할당

mean(class1$math) # 1반 수학 점수 평균 구하기
```

%>% 파이프 연산자
Ctrl + Shift + M

```
  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
```

dplyr 로 가공하기

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

# exam에서 class가 1인 경우만 추출해 출력

exam %>% filter(class == 1)

exam %>% filter(class !=1) # 1반이 아닌경우

exam %>% filter(math > 50) # 수학 점수가 50점을 초과

exam %>% filter(english >=80) # 영어 점수가 80 이상

# 1반이면서 수학 점수가 50 이상인 경우
exam %>% filter(class == 1 & math >= 50)

# 수학 점수가 90점 이상이거나 영어 점수가 90점 이상
exam %>% filter(math >= 90 | english >= 90)

# 1, 3, 5반에 해당하면 추출
exam %>% filter(class == 1 | class == 3 | class ==
exam %>% filter(class %in% c(1,3,5))
```

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

```
class1 <- exam %>% filter(class == 1) # class가 1인 행 추출, class1에 할당

mean(class1$math) # 1반 수학 점수 평균 구하기
```

dplyr 로 가공하기

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

# exam에서 class가 1인 경우만 추출해 출력

exam %>% filter(class == 1)

exam %>% filter(class !=1) # 1반이 아닌경우

exam %>% filter(math > 50) # 수학 점수가 50점을 초과한 경우

exam %>% filter(english >=80) # 영어 점수가 80 이상

# 1반이면서 수학 점수가 50 이상인 경우
exam %>% filter(class == 1 & math >= 50)

# 수학 점수가 90점 이상이거나 영어 점수가 90점 이상
exam %>% filter(math >= 90 | english >= 90)

# 1, 3, 5반에 해당하면 추출
exam %>% filter(class == 1 | class == 3 | class ==
exam %>% filter(class %in% c(1,3,5))
```

```
   id class math english science
1   2     1   60      97      60
2   7     2   80      90      45
3   8     2   90      78      25
4  11     3   65      65      65
5  15     4   75      56      78
6  16     4   58      98      65
7  17     5   65      68      98
8  18     5   80      78      90
9  19     5   89      68      87
10 20     5   78      83      58
```

```
class1 <- exam %>% filter(class == 1) # class가 1인 행 추출, class1에 할당

mean(class1$math) # 1반 수학 점수 평균 구하기
```

dplyr 로 가공하기

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

# exam에서 class가 1인 경우만 추출해 출력

exam %>% filter(class == 1)

exam %>% filter(class !=1) # 1반이 아닌경우

exam %>% filter(math > 50) # 수학 점수가 50점을 초과

exam %>% filter(english >=80) # 영어 점수가 80 이상

# 1반이면서 수학 점수가 50 이상인 경우
exam %>% filter(class == 1 & math >= 50)

# 수학 점수가 90점 이상이거나 영어 점수가 90점 이상
exam %>% filter(math >= 90 | english >= 90)

# 1, 3, 5반에 해당하면 추출
exam %>% filter(class == 1 | class == 3 | class == 5)
exam %>% filter(class %in% c(1,3,5))
```

```
   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   9     3   20      98      15
9  10     3   50      98      45
10 12     3   45      85      32
11 13     4   46      98      65
12 14     4   48      87      12
```

```
class1 <- exam %>% filter(class == 1) # class가 1인 행 추출, class1에 할당

mean(class1$math) # 1반 수학 점수 평균 구하기
```

dplyr 로 가공하기

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

# exam에서 class가 1인 경우만 추출해 출력

exam %>% filter(class == 1)

exam %>% filter(class !=1) # 1반이 아닌경우

exam %>% filter(math > 50) # 수학 점수가 50점을 초과한 경우

exam %>% filter(english >=80) # 영어 점수가 80 이상인 경우

# 1반이면서 수학 점수가 50 이상인 경우
exam %>% filter(class == 1 & math >= 50)

# 수학 점수가 90점 이상이거나 영어 점수가 90점 이상
exam %>% filter(math >= 90 | english >= 90)

# 1, 3, 5반에 해당하면 추출
exam %>% filter(class == 1 | class == 3 | class == 5)
exam %>% filter(class %in% c(1,3,5))

class1 <- exam %>% filter(class == 1) # class가 1인 행 추출, class1에 할당

mean(class1$math) # 1반 수학 점수 평균 구하기
```

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

dplyr 로 가공하기

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

# exam에서 class가 1인 경우만 추출해 출력

exam %>% filter(class == 1)

exam %>% filter(class !=1) # 1반이 아닌경우

exam %>% filter(math > 50) # 수학 점수가 50점을 초과한 경우

exam %>% filter(english >=80) # 영어 점수가 80 이상인 경우

# 1반이면서 수학 점수가 50 이상인 경우
exam %>% filter(class == 1 & math >= 50)

# 수학 점수가 90점 이상이거나 영어 점수가 90점 이상인
exam %>% filter(math >= 90 | english >= 90)

# 1, 3, 5반에 해당하면 추출
exam %>% filter(class == 1 | class == 3 | class == 5)
exam %>% filter(class %in% c(1,3,5))

class1 <- exam %>% filter(class == 1) # class가 1인 행 추출, class1에 할당

mean(class1$math) # 1반 수학 점수 평균 구하기
```

```
  id class math english science
1  1     1   50      98      50
2  2     1   60      97      60
3  4     1   30      98      58
4  7     2   80      90      45
5  8     2   90      78      25
6  9     3   20      98      15
7 10     3   50      98      45
8 13     4   46      98      65
9 16     4   58      98      65
```

dplyr 로 가공하기

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

# exam에서 class가 1인 경우만 추출해 출력

exam %>% filter(class == 1)

exam %>% filter(class !=1) # 1반이 아닌경우

exam %>% filter(math > 50) # 수학 점수가 50점을 초과한 경우

exam %>% filter(english >=80) # 영어 점수가 80 이상인

# 1반이면서 수학 점수가 50 이상인 경우
exam %>% filter(class == 1 & math >= 50)

# 수학 점수가 90점 이상이거나 영어 점수가 90점 이상인
exam %>% filter(math >= 90 | english >= 90)

# 1, 3, 5반에 해당하면 추출
exam %>% filter(class == 1 | class == 3 | class == 5)
exam %>% filter(class %in% c(1,3,5))

class1 <- exam %>% filter(class == 1) # class가 1인 행 추출, class1에 할당

mean(class1$math) # 1반 수학 점수 평균 구하기
```

	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	9	3	20	98	15
6	10	3	50	98	45
7	11	3	65	65	65
8	12	3	45	85	32
9	17	5	65	68	98
10	18	5	80	78	90
11	19	5	89	68	87

dplyr 로 가공하기

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

# exam에서 class가 1인 경우만 추출해 출력

exam %>% filter(class == 1)

exam %>% filter(class !=1) # 1반이 아닌경우

exam %>% filter(math > 50) # 수학 점수가 50점을 초과한 경우

exam %>% filter(english >=80) # 영어 점수가 80 이상인 경우

# 1반이면서 수학 점수가 50 이상인 경우
exam %>% filter(class == 1 & math >= 50)

# 수학 점수가 90점 이상이거나 영어 점수가 90점 이상인 경우
exam %>% filter(math >= 90 | english >= 90)

# 1, 3, 5반에 해당하면 추출
exam %>% filter(class == 1 | class == 3 | class == 5)
exam %>% filter(class %in% c(1,3,5))

class1 <- exam %>% filter(class == 1) # class가 1인 행 추출, class1에 할당

mean(class1$math) # 1반 수학 점수 평균 구하기
```

```
[1] 46.25
```

dplyr 로 가공하기

```
exam %>% select(math) # math 추출

exam$math

exam %>% select(class, math, english) # class, math, english 변수 추출

exam %>% select(-math) # math 제외

# class가 1인 행만 추출한 다음 english 추출
exam %>% filter(class == 1) %>% select(english)

exam %>%
  filter(class == 1) %>% # class가 1인 행 추출
  select(english) # english 추출
```

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

dplyr 로 가공하기

```
exam %>% select(math) # math 추출

exam$math

exam %>% select(class, math, english) # class, math, english 변수 추출

exam %>% select(-math) # math 제외

# class가 1인 행만 추출한 다음 english 추출
exam %>% filter(class == 1) %>% select(english)

exam %>%
  filter(class == 1) %>% # class가 1인 행 추출
  select(english) # english 추출
```

```
 [1] 50 60 45 30 25 50 80 90 20 50 65 45 46 48 75 58 65 80 89 78
```

dplyr 로 가공하기

```
exam %>% select(math) # math 추출

exam$math

exam %>% select(class, math, english) # class, math, english 변수 추출

exam %>% select(-math) # math 제외

# class가 1인 행만 추출한 다음 english 추출
exam %>% filter(class == 1) %>% select(english)

exam %>%
  filter(class == 1) %>% # class가 1인 행 추출
  select(english) # english 추출
```

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

dplyr 로 가공하기

```
exam %>% select(math) # math 추출

exam$math

exam %>% select(class, math, english) # class, math, english 변수 추출

exam %>% select(-math) # math 제외

# class가 1인 행만 추출한 다음 english 추출
exam %>% filter(class == 1) %>% select(english)

exam %>%
  filter(class == 1) %>% # class가 1인 행 추출
  select(english) # english 추출
```

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

dplyr 로 가공하기

```
exam %>% select(math) # math 추출

exam$math

exam %>% select(class, math, english) # class, math, english 변수 추출

exam %>% select(-math) # math 제외

# class가 1인 행만 추출한 다음 english 추출
exam %>% filter(class == 1) %>% select(english)

exam %>%
  filter(class == 1) %>% # class가 1인 행 추출
  select(english) # english 추출
```

```
  english
1      98
2      97
3      86
4      98
```

dplyr 로 가공하기

```
exam %>% select(math) # math 추출

exam$math

exam %>% select(class, math, english) # class, math, english 변수 추출

exam %>% select(-math) # math 제외

# class가 1인 행만 추출한 다음 english 추출
exam %>% filter(class == 1) %>% select(english)

exam %>%
  filter(class == 1) %>% # class가 1인 행 추출
  select(english) # english 추출
```

가독성을 위해서,

%>%(파이프 연산자)에서 줄을 바꾼다.

Enter를 치면 알아서 들여쓰기가 된다.

dplyr 로 가공하기

```
exam %>% arrange(math) # math 오름차순 정렬

exam %>% arrange(desc(math)) # math 내림차순 정렬

exam %>% arrange(class, math) # class 및 math 오름차순 정렬
```

	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
11	16	4	58	98	65
12	2	1	60	97	60
13	11	3	65	65	65
14	17	5	65	68	98
15	15	4	75	56	78
16	20	5	78	83	58
17	7	2	80	90	45
18	18	5	80	78	90
19	19	5	89	68	87
20	8	2	90	78	25

dplyr 로 가공하기

```
exam %>% arrange(math) # math 오름차순 정렬

exam %>% arrange(desc(math)) # math 내림차순 정렬

exam %>% arrange(class, math) # class 및 math 오름차순 정렬
```

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

dplyr 로 가공하기

```
exam %>% arrange(math) # math 오름차순 정렬

exam %>% arrange(desc(math)) # math 내림차순 정렬

exam %>% arrange(class, math) # class 및 math 오름차순 정렬
```

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

WeDataLab
데이터컨설팅 데이터아키텍처

4. 파생변수 추가하기 & 집단별로 요약하기

4. 파생변수 추가하기

```
exam %>%
  mutate(total = math + english + science) %>% # 종합 변수 추가
  head # 일부 추출

exam %>%
  mutate(total = math + english + science, # 종합 변수 추가
         mean = (math + english + science) / 3) %>% # 총평균 변수 추가
  head

exam %>%
  mutate(test = ifelse(science >= 60,"pass","fail"))
  head

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

mutate(data.frame, new_variable = 수식)

Make New Variables

4. 파생변수 추가하기

```
exam %>%
  mutate(total = math + english + science) %>% # 종합 변수 추가
  head # 일부 추출
```

```
exam %>%
  mutate(total = math + english + science, # 종합 변수 추가
         mean = (math + english + science) / 3) %>% # 총평균 변수 추가
  head
```

```
exam %>%
  mutate(test = ifelse(science >= 60,"pass","fail")) %>%
  head
```

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

4. 파생변수 추가하기

```
exam %>%
  mutate(total = math + english + science) %>% # 종합 변수 추가
  head # 일부 추출

exam %>%
  mutate(total = math + english + science, # 종합 변수 추가
         mean = (math + english + science) / 3) %>% # 총평균 변수 추가
  head
```

```
exam %>%
  mutate(test = ifelse(science >= 60,"pass","fail")) %>%
  head
```

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

id	class	math	english	science	test
1	1	50	98	50	fail
2	1	60	97	60	pass
3	1	45	86	78	pass
4	1	30	98	58	fail
5	2	25	80	65	pass
6	2	50	89	98	pass

4. 파생변수 추가하기 & 집단별로 요약하기

```
exam %>%
  mutate(total = math + english + science) %>% # 종합 변수 추가
  head # 일부 추출

exam %>%
  mutate(total = math + english + science, # 종합 변수 추가
         mean = (math + english + science) / 3) %>% # 총평균 변수 추가
  head

exam %>%
  mutate(test = ifelse(science >= 60,"pass","fail")) %>%
  head

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

id	class	math	english	science	total
9	3	20	98	15	133
14	4	48	87	12	147
12	3	45	85	32	162
5	2	25	80	65	170
4	1	30	98	58	186
8	2	90	78	25	193

4. 파생변수 추가하기 & 집단별로 요약하기

요약 통계량 함수

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

4. 집단별로 요약하기

```
exam %>% summarise(mean_math = mean(math)) # math 평균 산출
```

```
exam %>%
  group_by(class) %>% # class별로 분리
  summarise(mean_math = mean(math)) # math 평균 산출

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

mpg %>%
  group_by(manufacturer) %>% # 회사별로 분리
  filter(class == "suv") %>% # suv 추출
  mutate(tot = (cty + hwy)/2) %>% # 통합 연비 변수
  summarise(mean_tot = mean(tot)) %>% #통합 연비 평균 산출
  arrange(desc(mean_tot)) %>% # 내림차순 정렬
  head(5) # 일부 출력
```

mean_math
57.45

summarise(data.frame, functions...) 수치형 값에 대한 "요약" 통계량을 계산하여 출력한다.

4. 집단별로 요약하기

summarise( ):

- summarise() is typically used on grouped data created by group_by().
- The output will have one row for each group.
- **summarise(data.frame, functions...)**
- 수치형 값에 대한 "요약" 통계량을 계산하여 출력한다.
- Center: mean(), median()
- Spread: sd(), IQR(), mad()
- Range: min(), max(), quantile()
- Position: first(), last(), nth(),

4. 집단별로 요약하기

```
exam %>% summarise(mean_math = mean(math)) # math 평균 산출
```

```
exam %>%
  group_by(class) %>% # class별로 분리
  summarise(mean_math = mean(math)) # math 평균 산출
```

```
exam %>%
  group_by(class) %>% # class별로 분리
  summarise(mean_math = mean(math), # math 평균
            sum_math = sum(math), # math 합계
            median_math = median(math), # math 중앙
            n = n()) # 학생수
```

```
mpg %>%
  group_by(manufacturer) %>% # 회사별로 분리
  filter(class == "suv") %>% # suv 추출
  mutate(tot = (cty + hwy)/2) %>% # 통합 연비 변수
  summarise(mean_tot = mean(tot)) %>% #통합 연비 평
  arrange(desc(mean_tot)) %>% # 내림차순 정렬
  head(5) # 일부 출력
```

class <int>	measn_math <dbl>
1	46.2
2	61.2
3	45.0
4	56.8
5	78.0

4. 집단별로 요약하기

```
exam %>% summarise(mean_math = mean(math)) # math 평균 산출

exam %>%
  group_by(class) %>% # class별로 분리
  summarise(mean_math = mean(math)) # math 평균 산출

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

mpg %>%
  group_by(manufacturer) %>% # 회사별로 분리
  filter(class == "suv") %>% # suv 추출
  mutate(tot = (cty + hwy)/2) %>% # 통합 연비
  summarise(mean_tot = mean(tot)) %>% #통합 연
  arrange(desc(mean_tot)) %>% # 내림차순 정렬
  head(5) # 일부 출력
```

class	mean_math	sum_math	median_math	n
<int>	<dbl>	<int>	<dbl>	<int>
1	46.2	185	47.5	4
2	61.2	245	65.0	4
3	45.0	180	47.5	4
4	56.8	227	53.0	4
5	78.0	312	79.0	4

4. 집단별로 요약하기

```
exam %>% summarise(mean_math = mean(math)) # math 평균 산출

exam %>%
  group_by(class) %>% # class별로 분리
  summarise(mean_math = mean(math)) # math 평균 산출

exam %>%
  group_by(class) %>% # class별로 분리
  summarise(mean_math = mean(math), # math 평균
            sum_math = sum(math), # math 합계
            median_math = median(math), # mat
            n = n()) # 학생수
```

```
mpg %>%
  group_by(manufacturer) %>% # 회사별로 분리
  filter(class == "suv") %>% # suv 추출
  mutate(tot = (cty + hwy)/2) %>% # 통합 연비
  summarise(mean_tot = mean(tot)) %>% #통합 연
  arrange(desc(mean_tot)) %>% # 내림차순 정렬
  head(5) # 일부 출력
```

manufacturer	mean_tot
<chr>	<dbl>
subaru	21.9
toyota	16.3
nissan	15.9
mercury	15.6
jeep	15.6

WeDataLab
데이터 컨설팅 데이터 아카데미

데이터 프레임 합치기 & 정제하기

11. 데이터프레임 합치기

left_join(df1, df2, by='...')

```
# 중간고사 데이터 생성
test1 <- data.frame(id = c(1,2,3,4,5),
                    midterm = c(60,80,70,90,85))


# 기말고사 데이터 생성
test2 <- data.frame(id = c(1,2,3,4,5),
                    final = c(70, 83, 65, 95, 80))

test1 # test1 출력
test2 # test2 출력

# id를 기준으로 합쳐 total에 할당
total <- left_join(test1, test2, by = "id")
total # total 출력

name <- data.frame(class = c(1,2,3,4,5),
                   teacher = c("kim","lee","park","choi","ju
name

exam_new <- left_join(exam, name, by = "class")
exam_new
```

```
test1 # test1 출력
id midterm
 1       60
 2       80
 3       70
 4       90
 5       85
test2 # test2 출력
id final
 1    70
 2    83
 3    65
 4    95
 5    80
```

11. 데이터 프레임 합치기

```
# 중간고사 데이터 생성
test1 <- data.frame(id = c(1,2,3,4,5),
                    midterm = c(60,80,70,90,85))


# 기말고사 데이터 생성
test2 <- data.frame(id = c(1,2,3,4,5),
                    final = c(70, 83, 65, 95, 80))

test1 # test1 출력
test2 # test2 출력

# id를 기준으로 합쳐 total에 할당
total <- left_join(test1, test2, by = "id")
total # total 출력

name <- data.frame(class = c(1,2,3,4,5),
                   teacher = c("kim","lee","park","choi","jung"))
name

exam_new <- left_join(exam, name, by = "class")
exam_new
```

```
  id midterm final
1  1      60    70
2  2      80    83
3  3      70    65
4  4      90    95
5  5      85    80
```

11. 데이터 프레임 합치기

```
# 중간고사 데이터 생성
test1 <- data.frame(id = c(1,2,3,4,5),
                    midterm = c(60,80,70,90,85))


# 기말고사 데이터 생성
test2 <- data.frame(id = c(1,2,3,4,5),
                    final = c(70, 83, 65, 95, 80))

test1 # test1 출력
test2 # test2 출력

# id를 기준으로 합쳐 total에 할당
total <- left_join(test1, test2, by = "id")
total # total 출력
```

```
name <- data.frame(class = c(1,2,3,4,5),
                   teacher = c("kim","lee","park","choi","jung"))
name
```

```
exam_new <- left_join(exam, name, by = "class")
exam_new
```

class	teacher
1	kim
2	lee
3	park
4	choi
5	jung

11. 데이터 프레임 합치기

```
# 중간고사 데이터 생성
test1 <- data.frame(id = c(1,2,3,4,5),
                    midterm = c(60,80,70,90,85))


# 기말고사 데이터 생성
test2 <- data.frame(id = c(1,2,3,4,5),
                    final = c(70, 83, 65, 95, 80))

test1 # test1 출력
test2 # test2 출력

# id를 기준으로 합쳐 total에 할당
total <- left_join(test1, test2, by = "id")
total # total 출력

name <- data.frame(class = c(1,2,3,4,5),
                   teacher = c("kim","lee","park","choi","jung"))
name

exam_new <- left_join(exam, name, by = "class")
exam_new
```

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

11. 데이터 프레임 합치기

```
# 학생 1~5번 시험 데이터 생성
group_a <- data.frame(id = c(1,2,3,4,5),
                      test = c(60,80,70,90,85))
# 학생 6~10번 시험 데이터 생성
group_b <- data.frame(id = c(1,2,3,4,5),
                      test = c(70,83,65,95,80))
group_a
group_b
```

```
> group_a
  id test
1  1   60
2  2   80
3  3   70
4  4   90
5  5   85
> group_b
  id test
1  1   70
2  2   83
3  3   65
4  4   95
5  5   80
```

```
group_all <- bind_rows(group_a, group_b)
group_all
```

11. 데이터 프레임 합치기

```
# 학생 1~5번 시험 데이터 생성
group_a <- data.frame(id = c(1,2,3,4,5),
                      test = c(60,80,70,90,85))
# 학생 6~10번 시험 데이터 생성
group_b <- data.frame(id = c(1,2,3,4,5),
                      test = c(70,83,65,95,80))
group_a
group_b

group_all <- bind_rows(group_a, group_b)
group_all
```

```
  id test
1  1   60
2  2   80
3  3   70
4  4   90
5  5   85
1  1   70
2  2   83
3  3   65
4  4   95
5  5   80
```

WeDataLab
데이터컨설팅 데이터아키텍처

결측치, 이상치

데이터 정제하기 [결측치]

```r
df <- data.frame(sex = c("M","F", NA,"M","F"),
                 score = c(5,4,3,4,NA))
df

is.na(df) # 결측치 확인

table(is.na(df)) # 결측치 빈도 출력

table(is.na(df$sex)) # sex 결측치 빈도 출력
table(is.na(df$score)) # score 결측치 빈도 출력

mean(df$score) # 평균 산출
sum(df$score) # 합계 산출

df %>% filter(is.na(score)) # score가 NA인 데이터만 출력
df %>% filter(!is.na(score)) # score 결측치 제거

df_nomiss <- df %>% filter(!is.na(score)) # score 결측치 제거
mean(df_nomiss$score) # score 평균 산출
sum(df_nomiss$score) # score 합계 산출
```

```
> df <- data.frame(sex = c("M","F", NA,"M","F"),
+                  score = c(5,4,3,4,NA))
> df
   sex score
1    M     5
2    F     4
3 <NA>     3
4    M     4
5    F    NA
```

데이터 정제하기 [결측치]

```r
df <- data.frame(sex = c("M","F", NA,"M","F"),
                 score = c(5,4,3,4,NA))
df

is.na(df) # 결측치 확인

table(is.na(df)) # 결측치 빈도 출력

table(is.na(df$sex)) # sex 결측치 빈도 출력
table(is.na(df$score)) # score 결측치 빈도 출력

mean(df$score) # 평균 산출
sum(df$score) # 합계 산출

df %>% filter(is.na(score)) # score가 NA인 데이터만 출력
df %>% filter(!is.na(score)) # score 결측치 제거

df_nomiss <- df %>% filter(!is.na(score)) # score 결측치 제거
mean(df_nomiss$score) # score 평균 산출
sum(df_nomiss$score) # score 합계 산출
```

```
> is.na(df) # 결측치 확인
        sex score
[1,] FALSE FALSE
[2,] FALSE FALSE
[3,]  TRUE FALSE
[4,] FALSE FALSE
[5,] FALSE  TRUE
```

데이터 정제하기 [결측치]

```
df <- data.frame(sex = c("M","F", NA,"M","F"),
                 score = c(5,4,3,4,NA))
df

is.na(df) # 결측치 확인

table(is.na(df)) # 결측치 빈도 출력

table(is.na(df$sex)) # sex 결측치 빈도 출력
table(is.na(df$score)) # score 결측치 빈도 출력

mean(df$score) # 평균 산출
sum(df$score) # 합계 산출

df %>% filter(is.na(score)) # score가 NA인 데이터만 출력
df %>% filter(!is.na(score)) # score 결측치 제거

df_nomiss <- df %>% filter(!is.na(score)) # score 결측치 제거
mean(df_nomiss$score) # score 평균 산출
sum(df_nomiss$score) # score 합계 산출
```

```
> table(is.na(df)) # 결측치 빈도 출력

FALSE  TRUE
    8     2
```

데이터 정제하기 [결측치]

```
df <- data.frame(sex = c("M","F", NA,"M","F"),
                 score = c(5,4,3,4,NA))
df

is.na(df) # 결측치 확인

table(is.na(df)) # 결측치 빈도 출력

table(is.na(df$sex)) # sex 결측치 빈도 출력
table(is.na(df$score)) # score 결측치 빈도 출력

mean(df$score) # 평균 산출
sum(df$score) # 합계 산출

df %>% filter(is.na(score)) # score가 NA인 데이터만 출력
df %>% filter(!is.na(score)) # score 결측치 제거

df_nomiss <- df %>% filter(!is.na(score)) # score 결측치 제거
mean(df_nomiss$score) # score 평균 산출
sum(df_nomiss$score) # score 합계 산출
```

```
> table(is.na(df$sex)) #sex 결측치 빈도 출력

FALSE  TRUE
    4     1
> table(is.na(df$score)) #score 결측치 빈도 출력

FALSE  TRUE
    4     1
```

데이터 정제하기 [결측치]

```r
df <- data.frame(sex = c("M","F", NA,"M","F"),
                 score = c(5,4,3,4,NA))
df

is.na(df) # 결측치 확인

table(is.na(df)) # 결측치 빈도 출력

table(is.na(df$sex)) # sex 결측치 빈도 출력
table(is.na(df$score)) # score 결측치 빈도 출력

mean(df$score) # 평균 산출
sum(df$score) # 합계 산출

df %>% filter(is.na(score)) # score가 NA인 데이터만 출력
df %>% filter(!is.na(score)) # score 결측치 제거

df_nomiss <- df %>% filter(!is.na(score)) # score 결측치 제거
mean(df_nomiss$score) # score 평균 산출
sum(df_nomiss$score) # score 합계 산출
```

```
> mean(df$score) # 평균 산출
[1] NA
> sum(df$score) # 합계 산출
[1] NA
```

데이터 정제하기 [결측치]

```r
df <- data.frame(sex = c("M","F", NA,"M","F"),
                 score = c(5,4,3,4,NA))
df

is.na(df) # 결측치 확인

table(is.na(df)) # 결측치 빈도 출력

table(is.na(df$sex)) # sex 결측치 빈도 출력
table(is.na(df$score)) # score 결측치 빈도 출력

mean(df$score) # 평균 산출
sum(df$score) # 합계 산출

df %>% filter(is.na(score)) # score가 NA인 데이터만 출력
df %>% filter(!is.na(score)) # score 결측치 제거

df_nomiss <- df %>% filter(!is.na(score)) # score 결측치 제거
mean(df_nomiss$score) # score 평균 산출
sum(df_nomiss$score) # score 합계 산출
```

```
df %>% filter
sex score
  F    NA
df %>% filter
 sex score
   M     5
   F     4
<NA>     3
   M     4
```

데이터 정제하기 [결측치]

```
df <- data.frame(sex = c("M","F", NA,"M","F"),
                 score = c(5,4,3,4,NA))
df

is.na(df) # 결측치 확인

table(is.na(df)) # 결측치 빈도 출력

table(is.na(df$sex)) # sex 결측치 빈도 출력
table(is.na(df$score)) # score 결측치 빈도 출력

mean(df$score) # 평균 산출
sum(df$score) # 합계 산출

df %>% filter(is.na(score)) # score가 NA인 데이터만 출력
df %>% filter(!is.na(score)) # score 결측치 제거

df_nomiss <- df %>% filter(!is.na(score)) # score 결측치 제거
mean(df_nomiss$score) # score 평균 산출
sum(df_nomiss$score) # score 합계 산출
```

```
> df_nomiss <- df %>% filter(!is.na(score)) # score 결측치 제거
> mean(df_nomiss$score) # score 평균 산출
[1] 4
> sum(df_nomiss$score) # score 합계 산출
[1] 16
```

데이터 정제하기 [결측치]

```
df_nomiss <- df %>% filter(!is.na(score)&!is.na(sex)) # score, sex 결측치 제거
df_nomiss
```

```
df_nomiss2 <- na.omit(df) # 모든 변수에 결측치 없는 데이터 추출
df_nomiss2

mean(df$score, na.rm = T) # 결측치 제외하고 평균 산출
sum(df$score, na.rm = T) # 결측치 제외하고 합계 산출
```

```
sex score
  M     5
  F     4
  M     4
```

데이터 정제하기 [결측치]

```
df_nomiss <- df %>% filter(!is.na(score)&!is.na(sex)) # score, sex 결측치 제거
df_nomiss
```

```
df_nomiss2 <- na.omit(df) # 모든 변수에 결측치 없는 데이터 추출
df_nomiss2
```

```
mean(df$score, na.rm = T) # 결측치 제외하고 평균 산출
sum(df$score, na.rm = T) # 결측치 제외하고 합계 산출
```

sex	score
M	5
F	4
M	4

데이터 정제하기 [결측치]

```r
df_nomiss <- df %>% filter(!is.na(score)&!is.na(sex)) # score, sex 결측치 제거
df_nomiss

df_nomiss2 <- na.omit(df) # 모든 변수에 결측치 없는 데이터 추출
df_nomiss2

mean(df$score, na.rm = T) # 결측치 제외하고 평균 산출
sum(df$score, na.rm = T) # 결측치 제외하고 합계 산출
```

```
> mean(df$score, na.rm = T) # 결측치 제외하고 평균 산출
[1] 4
> sum(df$score, na.rm = T) # 결측치 제외하고 합계 산출
[1] 16
```

데이터 정제하기 [이상치]

```r
outlier <- data.frame(sex = c(1,2,1,3,2,1),
                      score = c(5,4,3,4,2,6))
outlier

table(outlier$sex)
table(outlier$score)

# sex가 3이면 NA 할당
outlier$sex <- ifelse(outlier$sex == 3, NA, outlier$sex)
outlier

# score가 5보다 크면 NA 할당
outlier$score <- ifelse(outlier$score > 5, NA, outlier$score)
outlier

outlier %>%
  filter(!is.na(sex) & !is.na(score)) %>%
  group_by(sex) %>%
  summarise(mean_score = mean(score))
```

sex	score
1	5
2	4
1	3
3	4
2	2
1	6

데이터 정제하기 [이상치]

```r
outlier <- data.frame(sex = c(1,2,1,3,2,1),
                      score = c(5,4,3,4,2,6))
outlier

table(outlier$sex)
table(outlier$score)

# sex가 3이면 NA 할당
outlier$sex <- ifelse(outlier$sex == 3, NA, outlier$sex)
outlier

# score가 5보다 크면 NA 할당
outlier$score <- ifelse(outlier$score > 5, NA, outlier$score)
outlier

outlier %>%
  filter(!is.na(sex) & !is.na(score)) %>%
  group_by(sex) %>%
  summarise(mean_score = mean(score))
```

```
> table(outlier$sex)

1 2 3
3 2 1
> table(outlier$score)

2 3 4 5 6
1 1 2 1 1
```

데이터 정제하기 [이상치]

```r
outlier <- data.frame(sex = c(1,2,1,3,2,1),
                      score = c(5,4,3,4,2,6))
outlier

table(outlier$sex)
table(outlier$score)

# sex가 3이면 NA 할당
outlier$sex <- ifelse(outlier$sex == 3, NA, outlier$sex)
outlier

# score가 5보다 크면 NA 할당
outlier$score <- ifelse(outlier$score > 5, NA, outlier$score)
outlier

outlier %>%
  filter(!is.na(sex) & !is.na(score)) %>%
  group_by(sex) %>%
  summarise(mean_score = mean(score))
```

sex	score
1	5
2	4
1	3
NA	4
2	2
1	6

데이터 정제하기 [이상치]

```r
outlier <- data.frame(sex = c(1,2,1,3,2,1),
                      score = c(5,4,3,4,2,6))
outlier

table(outlier$sex)
table(outlier$score)

# sex가 3이면 NA 할당
outlier$sex <- ifelse(outlier$sex == 3, NA, outlier$sex)
outlier

# score가 5보다 크면 NA 할당
outlier$score <- ifelse(outlier$score > 5, NA, outlier$score)
outlier

outlier %>%
  filter(!is.na(sex) & !is.na(score)) %>%
  group_by(sex) %>%
  summarise(mean_score = mean(score))
```

sex	score
1	5
2	4
1	3
NA	4
2	2
1	NA

데이터 정제하기 [이상치]

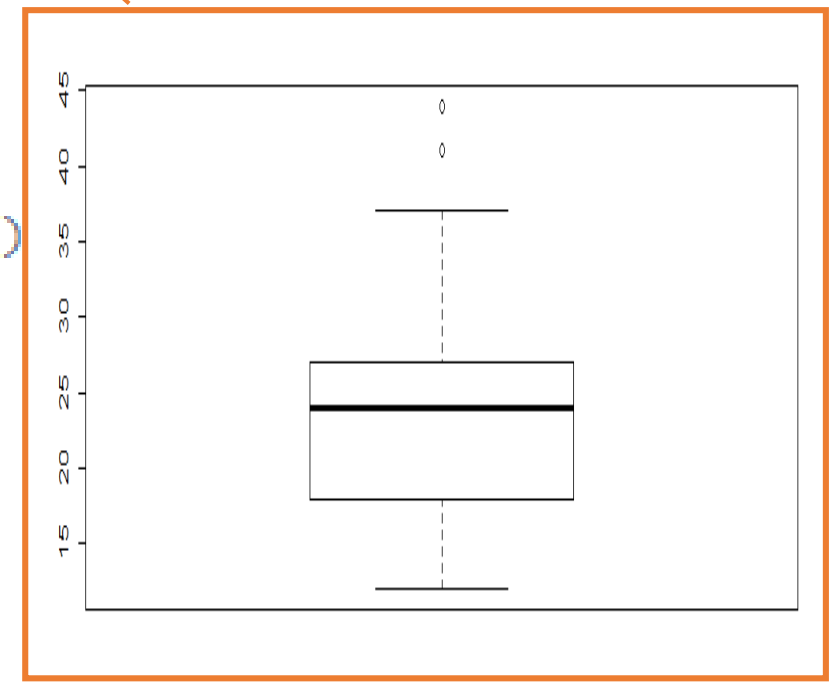
```
outlier <- data.frame(sex = c(1,2,1,3,2,1),
                      score = c(5,4,3,4,2,6))
outlier

table(outlier$sex)
table(outlier$score)

# sex가 3이면 NA 할당
outlier$sex <- ifelse(outlier$sex == 3, NA, outlier$sex)
outlier

# score가 5보다 크면 NA 할당
outlier$score <- ifelse(outlier$score > 5, NA, outlier$score)
outlier
```

```
outlier %>%
  filter(!is.na(sex) & !is.na(score)) %>%
  group_by(sex) %>%
  summarise(mean_score = mean(score))
```

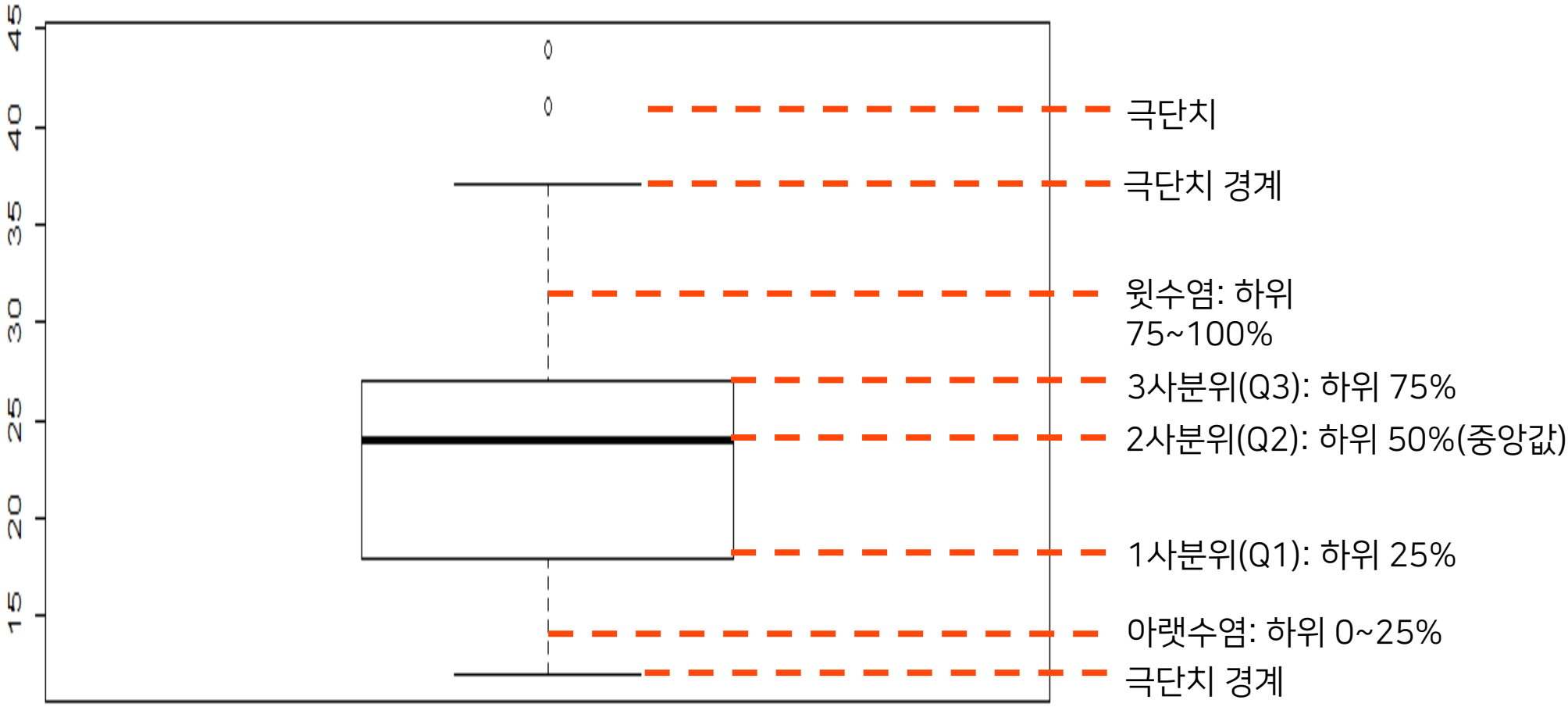
```
A tibble: 2 x 2
    sex mean_score
  <dbl>      <dbl>
1  1.00       4.00
2  2.00       3.00
```

데이터 정제하기 [이상치]

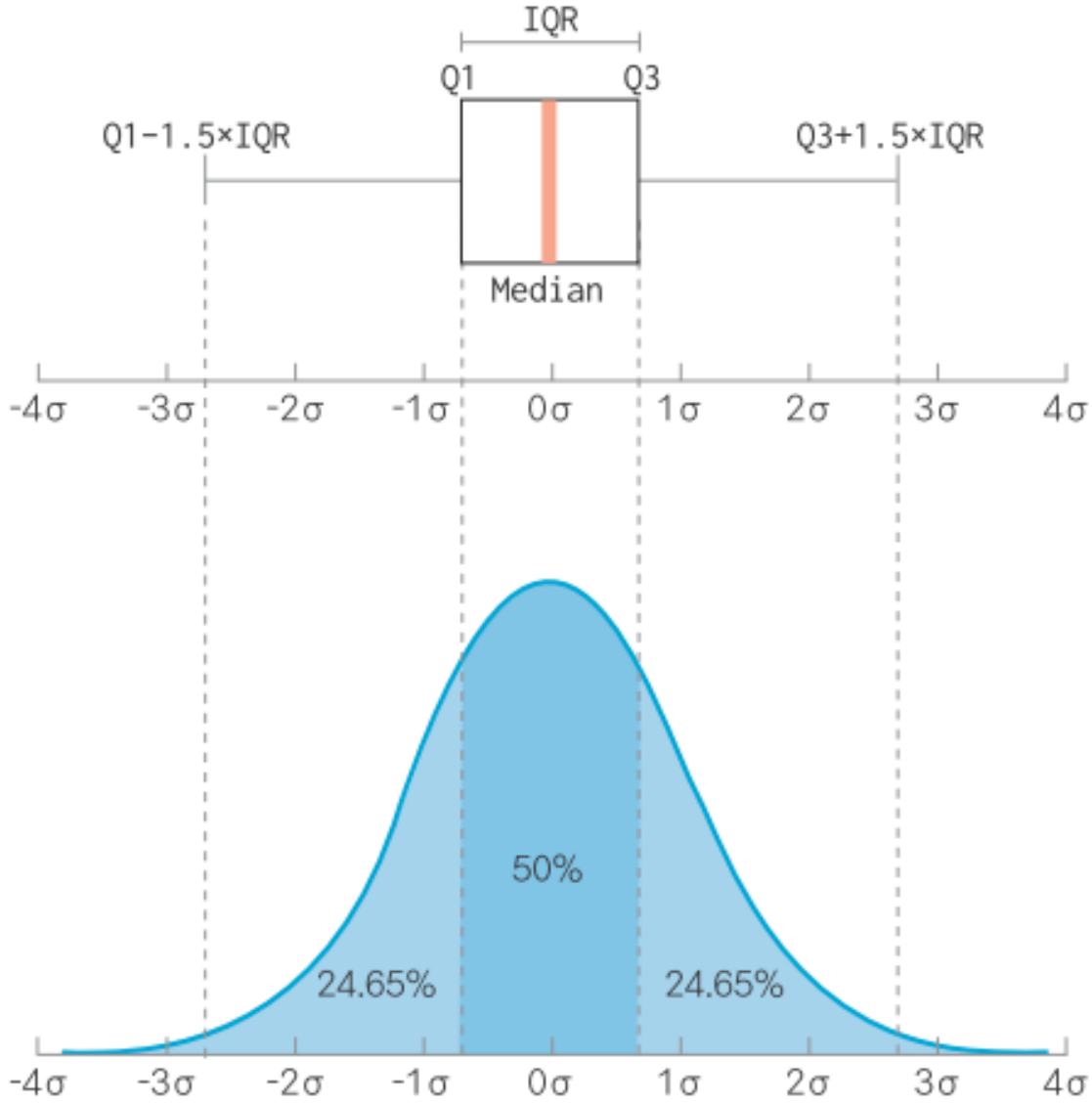
```r
boxplot(mpg$hwy) # 상자 그림 통계치 출력

# 12~37 벗어나면 NA 할당
mpg$hwy <- ifelse(mpg$hwy < 12 | mpg$hwy > 37, NA, mpg$hwy)
table(is.na(mpg$hwy))

mpg %>%
  group_by(drv) %>%
  summarise(mean_hwy = mean(hwy, na.rm = T))
```



데이터 정제하기 [이상치]



데이터 정제하기 [이상치]



데이터 정제하기 [이상치]

```r
boxplot(mpg$hwy) # 상자 그림 통계치 출력
```

```r
# 12~37 벗어나면 NA 할당
mpg$hwy <- ifelse(mpg$hwy < 12 | mpg$hwy > 37, NA, mpg$hwy)
table(is.na(mpg$hwy))
```

```r
mpg %>%
  group_by(drv) %>%
  summarise(mean_hwy = mean(hwy, na.rm = T))
```

```
FALSE  TRUE
  231     3
```

데이터 정제하기 [이상치]

```
boxplot(mpg$hwy) # 상자 그림 통계치 출력

# 12~37 벗어나면 NA 할당
mpg$hwy <- ifelse(mpg$hwy < 12 | mpg$hwy > 37, NA, mpg$hwy)
table(is.na(mpg$hwy))
```

```
mpg %>%
  group_by(drv) %>%
  summarise(mean_hwy = mean(hwy, na.rm = T))
```

```
A tibble: 3 x 2
drv   mean_hwy
<chr>    <dbl>
4         19.2
f         27.7
r         21.0
```