Data Exploration

데이터 파악에 사용되는 기초 함수

- 함수 종류

함수	기능
head()	데이터 목록에서 앞부분의 일부 데이터 출력
tail()	데이터 목록에서 뒷부분의 일부 데이터 출력
View()	별도의 뷰어 창에 데이터 표시
dim()	데이터의 차원 출력
str()	데이터의 구조와 속성 출력
summary()	요약 기초 통계량 출력

- 테스트 데이터 준비

```
exam <- read.csv("data-files/csv_exam.csv")
```

데이터 파악에 사용되는 기초 함수

- 목록 데이터의 앞부분 일부 또는 뒷부분 일부를 출력하는 함수

```
> head(exam)
  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
> head(exam, n = 3)
  id class math english science
1  1     1   50      98      50
2  2     1   60      97      60
3  3     1   45      86      78
```

```
> tail(exam)
   id class math english science
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
> tail(exam, n = 3)
   id class math english science
18 18     5   80      78      90
19 19     5   89      68      87
20 20     5   78      83      58
```

데이터 파악에 사용되는 기초 함수

- 데이터의 행과 열 정보 출력

```
> dim(exam)
[1] 20  5
```

- 데이터 구조 및 속성 출력

```
> str(exam)
'data.frame':   20 obs. of  5 variables:
 $ id     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ class  : int  1 1 1 1 2 2 2 2 3 3 ...
 $ math   : int  50 60 45 30 25 50 80 90 20 50 ...
 $ english: int  98 97 86 98 80 89 90 78 98 98 ...
 $ science: int  50 60 78 58 65 98 45 25 15 45 ...
```

데이터 파악에 사용되는 기초 함수

- 요약 기초 통계량 출력

```
> summary(exam)
       id             class        math           english         science
 Min.   : 1.00   Min.   :1   Min.   :20.00   Min.   :56.0   Min.   :12.00
 1st Qu.: 5.75   1st Qu.:2   1st Qu.:45.75   1st Qu.:78.0   1st Qu.:45.00
 Median :10.50   Median :3   Median :54.00   Median :86.5   Median :62.50
 Mean   :10.50   Mean   :3   Mean   :57.45   Mean   :84.9   Mean   :59.45
 3rd Qu.:15.25   3rd Qu.:4   3rd Qu.:75.75   3rd Qu.:98.0   3rd Qu.:78.00
 Max.   :20.00   Max.   :5   Max.   :90.00   Max.   :98.0   Max.   :98.00
```

데이터 파악에 사용되는 기초 함수

- ggplot2 패키지에 포함된 mpg 데이터셋으로 실습

```
this_mpg <- as.data.frame(ggplot2::mpg)


head(this_mpg)
tail(this_mpg)
View(this_mpg)
dim(this_mpg)
str(this_mpg)
class(this_mpg)
summary(this_mpg)
```

결과 생략

dplyr

- 데이터 처리에 특화된 패키지

- R로 작성된 plyr 패키지가 처리 속도가 느린 문제점을 노출하지만 dplyr는 C++로 작성되어 처리 속도가 매우 빠름

- 주요 함수

함수명	내용	유사함수
filter()	지정한 조건식에 맞는 데이터 추출	subset()
select()	열의 추출	data[, c('Year', 'Month')]
mutate()	열 추가	transform()
arrange()	정렬	order(), sort()
summarise()	집계	aggregate()
group_by	그룹화	-

dplyr

- 패키지 설치

```
install.packages(c('dplyr', 'hflights'))
library(dplyr)
library(hflights) # 테스트 데이터 셋 패키지
```

- 데이터 래핑
  - tbl_df 함수를 사용해서 처리 효율성 향상된 데이터 프레임 래퍼 생성

```
dim(hflights)

# data frame wrapping
hflights_df <- tbl_df(hflights)
hflights_df
```

dplyr

- 필터링 → filter() 함수 사용

```
filter(hflights_df, Month == 1 & DayofMonth == 1)
filter(hflights_df, Month == 1 | Month == 2)
```

- 정렬 → arrange() 함수 사용

```
arrange(hflights_df, ArrDelay, Year, Month)
arrange(hflights_df, desc(ArrDelay), Month, Year)
```

- 선택적 컬럼 구성 → select() 함수 사용

```
select(hflights_df, Year, Month, DayOfWeek)
select(hflights_df, Year:DayOfWeek)
select(hflights_df, -(Year:DayOfWeek))
```

dplyr

- 계산된 컬럼, 컬럼 추가 → mutate() 함수 사용

```
mutate(hflights_df,
       gain = ArrDelay - DepDelay,
       gain_per_hour = gain/(AirTime/60))

# transform과 비교
transform(hflights,
          gain = ArrDelay - DepDelay,
          gain_per_hour = gain/(AirTime/60))
```

- 집계 → summarise() 함수 사용

```
summarise(hflights_df, delay = mean(DepDelay, na.rm = TRUE))
```

dplyr

- 그룹 생성 → group_by() 함수 사용

```
planes <- group_by(hflights_df, TailNum)
delay <- summarise(planes, count = n(),
                   dist = mean(Distance, na.rm = T),
                   delay = mean(ArrDelay, na.rm = T))
delay <- filter(delay, count > 20, dist < 2000)

delay

ggplot(delay, aes(x = dist, y = delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  geom_smooth()
```

dplyr

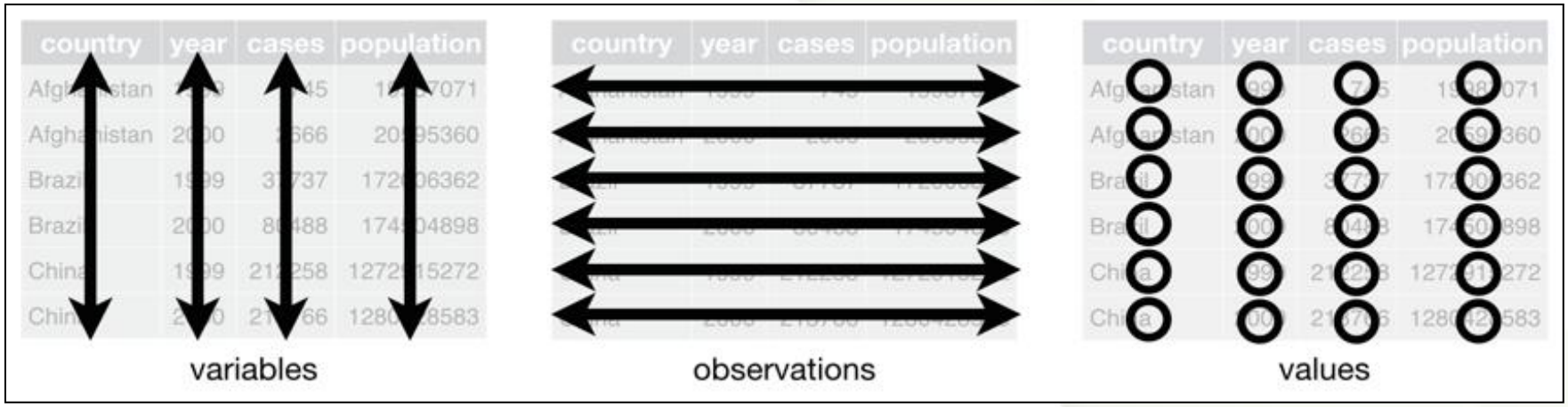
- 함수 체인
  - 연속적인 함수 호출을 단순하게 작성

```
a1 <- group_by(hflights, Year, Month, DayofMonth)
a2 <- select(a1, Year:DayofMonth, ArrDelay, DepDelay)
a3 <- summarise(a2, arr = mean(ArrDelay, na.rm = TRUE),
                    dep = mean(DepDelay, na.rm = TRUE))
a4 <- filter(a3, arr > 30 | dep > 30)
a4

# function chain
hflights_df %>%
  group_by(Year, Month, DayofMonth) %>%
  summarise(arr = mean(ArrDelay, na.rm = TRUE),
            dep = mean(DepDelay, na.rm = TRUE)) %>%
  filter(arr > 30 | dep > 30)
```

tidyr

- R의 함수들을 적용할 때 좋은 데이터 구조
  - 데이터 셋의 각 변수는 자신만의 컬럼으로 구성된다
  - 데이터 셋의 각 관측치는 행으로 구성된다
  - 데이터 셋의 각 값은 셀에 표현된다



tidyr

- 데이터프레임 구조 변경을 처리하는 패키지
  - reshape2를 데이터프레임 전용으로 축소한 패키지

- 주요 함수

함수명	내용	유사함수
spread()	Wide to Long 변환	dcast()
gather()	Long to Wide 변환	melt()
separate()	하나의 컬럼을 여러 컬럼으로 분리	colsplit()
unite()	여러 개의 컬럼을 하나의 컬럼으로 병합	dcast()

- 패키지 설치

```
install.packages("tidyr")
library("tidyr")
library(devtools)
install_github("garrettgman/DSR")
library(DSR)
```

tidyr

- table1 ~ table6의 예제 데이터셋 구조 확인

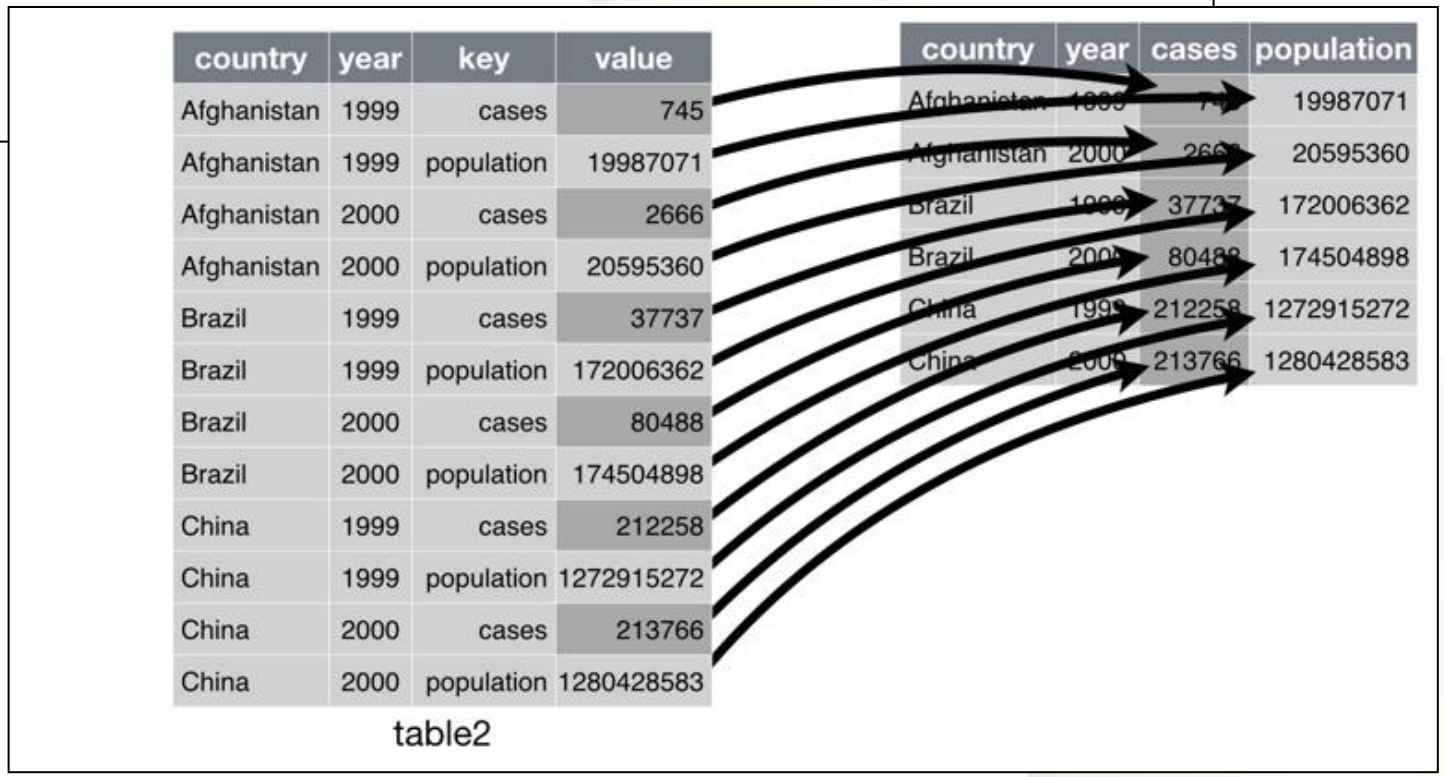
```
table1
table2
table3
table4
table5
table6
```

tidyr

- spread() 함수
  - key : value 쌍의 컬럼을 간결한 컬럼으로 변경
  - Row 데이터를 Column 데이터로 변형

```
tidy <- spread(table2, key, value)
tidy

[결과 생략]
```



country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

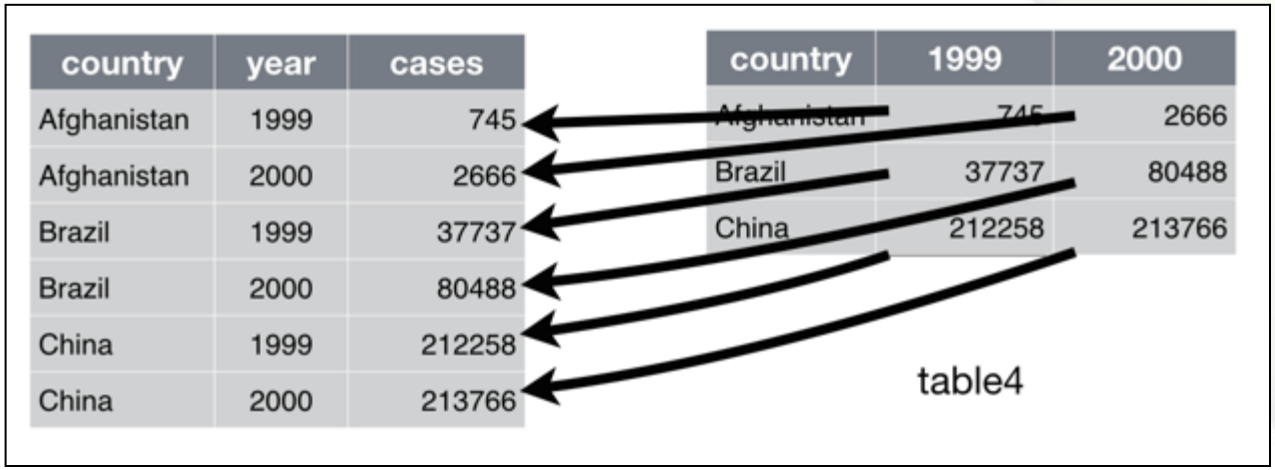
country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

tidyr

- gather() 함수
  - spread() 함수와 반대 방향의 데이터 변환
  - Column 데이터를 Row 데이터로 변환

```
gathered_messy <- gather(table4, year, case, 2:3)
gathered_messy

[결과 생략]
```



tidyr

- separate() 함수 사용
  - 하나의 컬럼을 여러 개의 컬럼으로 분할

```
seperated_messy <- separate(table3,
                            rate,
                            into = c("cases", "population"),
                            sep = "/")
seperated_messy

[ 결과 생략 ]

seperated_messy2 <- separate(table3,
                             year,
                             into = c("century", "year"),
                             sep = 2)
seperated_messy2

[ 결과 생략 ]
```

tidyr

- unite() 함수 사용
  - 여러 개의 컬럼을 하나의 컬럼으로 병합

```
united <- unite(table6,
                year,
                century,
                year,
                sep = "")
united

[ 결과 생략 ]
```

컬럼 이름 바꾸기 (변수 이름 바꾸기)

- rename() 함수 사용

```
df_raw <- data.frame(var1 = c(1, 2, 1),
                     var2 = c(2, 3, 2))
df_raw

df_new <- df_raw # 복사본 만들기

df_new2 <- rename(df_new, v1 = var1, v2 = var2)
df_new2
```

```
  var1 var2
1    1    2
2    2    3
3    1    2
```

```
  v1 v2
1  1  2
2  2  3
3  1  2
```

- colnames() 함수 사용

```
colnames(df_raw) <- c("vx", 'vy')
df_raw
```

```
  vx vy
1  1  2
2  2  3
3  1  2
```

컬럼 이름 변경 (변수 이름 변경)

- 연습

```
copied_mpg <- mpg
renamed_mpg <- rename(copied_mpg, city = cty, highway = hwy)
renamed_mpg

cnames <- colnames(copied_mpg)
cnames[8:9] <- c('city', 'highway')
cnames

colnames(copied_mpg) <- cnames
copied_mpg
```

결과 생략

컬럼 추가하기 (파생 변수 만들기)

- 연습

```
> df <- data.frame(var1 = c(4, 3, 8), var2 = c(2, 6, 1))
> df
  var1 var2
1    4    2
2    3    6
3  8    1

> df$var_sum <- df$var1 + df$var2
> df
  var1 var2 var_sum
1    4    2       6
2    3    6       9
3  8    1       9

> df$var_mean <- df$var_sum / 2
> df
  var1 var2 var_sum var_mean
1    4    2       6      3.0
2    3    6       9      4.5
3    8    1       9      4.5
```

컬럼 추가하기 (파생 변수 만들기)

- 연습2

```
mpg$mean <- (mpg$cty + mpg$hwy) / 2
mpg

summary(mpg$mean)
hist(mpg$mean)

mpg$test <- ifelse(mpg$mean >= 20, "pass", "fail")
mpg
table(mpg$test)
qplot(mpg$test)

mpg$grade <- ifelse(mpg$mean >= 30, "A",
                    ifelse(mpg$mean >= 25, "B",
                           ifelse(mpg$mean >= 20, "C", "D")))
mpg
table(mpg$grade)
qplot(mpg$grade)
```

결과 생략

조건에 맞는 데이터 추출

- filter() 함수와 조건식 사용

```
exam <- read.csv("data-files/csv_exam.csv")
exam

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

exam %>% filter(math > 50)
exam %>% filter(math < 50)
exam %>% filter(english >= 80)
exam %>% filter(english <= 80)

exam %>% filter(class == 1 & math > 50)
exam %>% filter(class == 2 & english >= 80)

exam %>% filter(math >= 90 | english >= 90)
exam %>% filter(class == 1 | class == 2 | class == 3)
```

결과 생략

조건에 맞는 데이터 추출

- filter() 함수와 조건식 사용 연습

[mpg 데이터 셋에서]

배기량(displ) 4 이하인 자동차와 5이상인 자동차의 고속도로 연비(hwy) 평균 비교

제조사(manufacturer) audi와 toyoto의 도시 연비(cty) 평균 비교

제조사(manufacturer) chevrolet, ford, honda의 모든 자동차의 고속도로 평균 연비(hwy)

필요한 변수만 추출하기

- select() 함수 사용

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

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

exam %>%
  filter(class == 1) %>%
  select(english)

exam %>%
  select(id, math) %>%
  head(10)
```

결과 생략

필요한 변수만 추출하기

- select() 함수 사용 연습

[mpg 데이서 셋에서]

class, cty, hwy 컬럼의 데이터 중 앞부분 10개 조회

앞의 결과에서 자동차 종류(class)가 suv 와 compact인 자동차의 도시 연비(cty) 평균 비교

순서대로 정렬하기

- arrange() 함수 사용

```
> exam %>% arrange(math)
   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
… 이하 생략

> exam %>% arrange(desc(math))
   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
… 이하 생략
```

```
> exam %>% arrange(class, math)
   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
… 이하 생략
```

순서대로 정렬하기

- arrange() 함수 사용 연습

[mpg 데이서 셋에서]

audi에서 생산한 자동차 중에 hwy가 1 ~ 5위에 해당하는 자동차의 데이터 출력

파생 변수 추가하기 (컬럼 추가)

- mutate() 함수 사용

```
> exam %>%
+   mutate(total = math + english + science) %>%
+   head(5)
  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

> exam %>%
+   mutate(total = math + english + science,
+          mean = total / 3) %>%
+   head(5)
  id class math english science total     mean
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
```

파생 변수 추가하기 (컬럼 추가)

- mutate() 함수 사용 연습

[mpg 데이서 셋에서]

mpg 데이터셋의 복사본 만들기

복사본을 대상으로 cty + hwy로 계산된 컬럼 추가하기 (total)

변경된 데이터셋을 대상으로 연비 평균을 저장하는 컬럼 추가하기 (mean)

연비 평균을 기준으로 내림 차순 정렬하고 상위 3개 데이터 출력하기

위 작업을 하나의 연속된 구문으로 작성하기

집단 별로 요약하기

- summaries() 함수 사용

```
> exam %>%
+   summarise(mean_math = mean(math))
  mean_math
1     57.45

> exam %>%
+   group_by(class) %>%
+   summarise(mean_math = mean(math))
# A tibble: 5 x 2
  class mean_math
  <int>     <dbl>
1     1      46.2
2     2      61.2
3     3      45
4     4      56.8
5     5      78
```

집단 별로 요약하기

- summaries() 함수 사용 (계속)

```
> exam %>%
+   group_by(class) %>%
+   summarise(mean_math = mean(math),
+             sum_math = sum(math),
+             median_math = median(math),
+             n = n())
# A tibble: 5 x 5
  class mean_math sum_math median_math     n
  <int>     <dbl>    <int>       <dbl> <int>
1     1      46.2      185        47.5     4
2     2      61.2      245        65       4
3     3      45        180        47.5     4
4     4      56.8      227        53       4
5     5      78        312        79       4
```

집단 별로 요약하기

- 자주 사용하는 요약 통계 함수

함수	기능	함수	기능
mean()	평균	min()	최소값
sd()	표준편차	max()	최대값
sum()	합계	n()	빈도
median()	중앙값		

집단 별로 요약하기

- summaries() 함수 사용 (계속)

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

# A tibble: 5 x 2
  manufacturer mean_tot
  <chr>            <dbl>
1 subaru            21.9
2 toyota            16.3
3 nissan            15.9
4 mercury           15.6
5 jeep              15.6
```

집단 별로 요약하기

- summaries() 함수 사용 연습

[mpg 데이서 셋에서]

class 별 cty 평균 구하기

앞의 결과를 cty 평균 기준 내림차순 정렬

고속도로 연비(hwy)가 가장 높은 3개의 회사 조회

회사별 compact 차종 수를 내림차순으로 정렬해서 출력

데이터 합치기

- left_join() 함수 사용

```
test1 <- data.frame(id = c(1, 2, 3, 4, 5),
                    midterm = c(60, 80, 70, 90, 85))
test2 <- data.frame(id = c(1, 2, 3, 4, 5),
                    final = c(70, 83, 65, 95, 80))

test1
test2

total <- left_join(test1, test2, by = 'id')
total

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

결과 생략

데이터 합치기

- bind_rows() 함수 사용

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

group_all <- bind_rows(group_a, group_b)
group_all


# 결합하는 두 데이터프레임의 컬럼명이 같은 경우와 다른 경우의 차이는?
```

```
결과 생략
```

데이터 합치기

- left_join(), bind_rows() 함수 사용 연습

[mpg 데이서 셋에서]

먼저 아래 코드를 실행해서 데이터 프레임을 만드세요

```
    fuel <- data.frame(fl = c('c', 'd', 'e', 'p', 'r'),
                       price_fl = c(2.35, 2.38, 2.11, 2.76, 2.22),
                       stringsAsFactors = FALSE)
```

this_mpg라는 이름으로 mpg의 복사본을 만드세요

this_mpg 데이터셋과 fuel 데이터셋을 fl 컬럼을 기준으로 병합하세요

model, fl, price_fl 컬럼을 조회하세요

결과 생략

결측치 정제하기

- is.na() 함수로 결측치 찾기

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

```
> table(is.na(df))

FALSE  TRUE
    8     2
> table(is.na(df$sex))

FALSE  TRUE
    4     1
> table(is.na(df$score))

FALSE  TRUE
    4     1
```

결측치 정제하기

- na.omit() 함수 사용

```
> df_nomiss <- df %>% filter(!is.na(score) & !is.na(sex))
> df_nomiss
  sex score
1   M     5
2   F     4
3   M     4

> df_nomiss2 <- na.omit(df)
> df_nomiss2
  sex score
1   M     5
2   F     4
4   M     4
```

결측치 정제하기

- na.rm 전달인자 사용

```
> mean(df$score)
[1] NA
> mean(df$score, na.rm = TRUE)
[1] 4
> sum(df$score, na.rm = TRUE)
[1] 16

> exam <- read.csv("data-files/csv_exam.csv")
> exam[c(3, 8, 15), "math"] <- NA

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

  mean_math
1  NA

> exam %>%
+   summarise(mean_math = mean(math, na.rm = TRUE))
  mean_math
1  55.23529
```

결측치 대체하기

- 평균 값으로 결측치 대체

```
> exam <- read.csv("data-files/csv_exam.csv")
> math_mean <- mean(exam$math, na.rm = TRUE)
[1] 57.45

> exam$math <- ifelse(is.na(exam$math), math_mean, exam$math)
> table(is.na(exam$math))
FALSE
   20
```

- 연습

```
[ mpg 데이터셋 ]

먼저 아래와 같이 데이터를 변경하세요

    this_mpg <- as.data.frame(ggplot2::mpg)
    this_mpg[c(65, 124, 131, 153, 212), "hwy"] <- NA

drv 컬럼과 hwy 컬럼에 NA 존재 여부 및 개수 확인

결측치를 제외하고 drv별로 hwy 평균 계산
```

이상치(Outlier) 정제하기

- 존재할 수 없는 값

```
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)

outlier$sex <- ifelse(outlier$sex %in% c(1, 2), outlier$sex, NA)
outlier

outlier$score <- ifelse(outlier$score %in% 1:5, outlier$score, NA)
outlier

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

결과 생략

이상치(Outlier) 정제하기

- 극단적인 값

```
boxplot(mpg$hwy, horizontal = TRUE)
hwy_range <- boxplot(mpg$hwy)$stats
hwy_range

mpg$hwy <- ifelse(mpg$hwy < hwy_range[1, 1] | mpg$hwy > hwy_range[5, 1],
NA, mpg$hwy)
table(is.na(mpg$hwy))

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

결과 생략

이상치(Outlier) 정제하기

- 이상치 제거하기 연습

[mpg 데이서 셋에서]

먼저 아래 코드를 실행해서 데이터 프레임을 만드세요

```
mpg <- as.data.frame(ggplot2::mpg)
mpg[c(10, 14, 58, 93), "drv"] <- "k";
mpg[c(29, 43, 129, 203), "cty"] <- c(3, 4, 39, 42)
```

drv에 이상치가 있는지 확인하고 이상치를 결측 처리 하세요

cty에 이상치가 있는지 확인하고 이상치를 결측 처리 하세요

drv, cty의 결측치 값을 제외하고 drv별로 cty의 평균을 구하세요

결과 생략

시각화 (Visualization)

- 별도의 시각화 문서 참고

변수 사이의 관계 표현하기

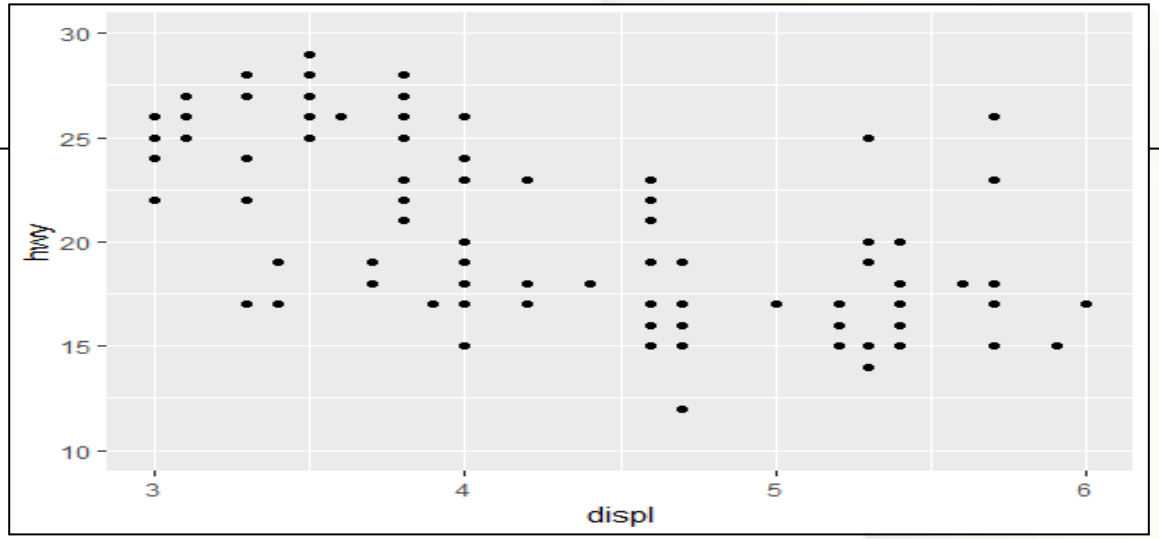
- 산점도 그래프 사용

```
ggplot(data = mpg, aes(x = displ, y = hwy))

ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point()

ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  xlim(3, 6)

ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  xlim(3, 6) +
  ylim(10, 30)
```



변수 사이의 관계 표현하기

- 산점도 그래프 연습

[ggplot2 패키지에 포함된 mpg 데이터 셋 사용]

x축은 cty, y축은 hwy로 산점도 그래프 만들기

[ggplot2 패키지에 포함된 midwest 데이터 셋 사용]

x축은 poptotal, y축은 popasian으로 산점도 그래프 만들기 (단, 전체 인구는 50만 이하, 아시아 인구는 1만 명 이하 지역만 산점도에 표시)

결과 생략

집단 사이의 차이 표현하기

- 막대 그래프 사용

```
mpg <- ggplot2::mpg

df_mpg <- ggplot2::mpg %>%
  group_by(drv) %>%
  summarise(mean_hwy = mean(hwy))
df_mpg

ggplot(data = df_mpg, aes(x = drv, y = mean_hwy)) + geom_col()

ggplot(data = df_mpg, aes(x = reorder(drv, -mean_hwy), y = mean_hwy)) +
  geom_col()

ggplot(data = mpg, aes(x = drv)) + geom_bar()


ggplot(data = mpg, aes(x = hwy)) + geom_bar()
```

결과 생략

집단 사이의 차이 표현하기

- 막대 그래프 연습

[ggplot2 패키지에 포함된 mpg 데이터 셋 사용]

suv 차종을 대상으로 cty가 가장 높은 회사 다섯 곳을 막대 그래프로 표현하세요

자동차 종류별 빈도를 표현한 막대 그래프를 표현하세요

결과 생략

시계열 데이터 표현하기

- 선 그래프 사용

```
ggplot(data = economics, aes(x = date, y = unemploy)) + geom_line()
```

- 선 그래프 연습

```
[ ggplot2 패키지의 economics 데이터 셋 사용 ]

시간에 따른 개인 저축률(psavert)의 변화를 나타낸 그래프를 만드세요
```

```
결과 생략
```

집단 사이의 분포 차이 표현하기

- 상자 그림 사용

```
ggplot(data = mpg, aes(x = drv, y = hwy)) + geom_boxplot()
```

- 상자 그림 연습

[ggplot2 패키지의 mpg 데이터 셋 사용]

자동차 종류(class)가 ‘compact’, ‘subcompact’, ‘suv’인 자동자의
도시연비(cty)를 표시하는 상자 그림을 만드세요

결과 생략