히든:그레이스

Start! 데이터분석 R

Profile

김영우

㈜히든그레이스 데이터분석팀장

데이터분석 교육(R, SPSS, AMOS, 데이터분석방법론)

연구방법론 및 통계분석 대학원 출강(2013~현재)

데이터 저널리즘 언론매체 '데이터 저널(datajournal.kr)' 대표

오마이뉴스 시민기자 - '데이터로 보는 뉴스' 연재

한국데이터베이스진흥원 빅데이터 분석 전문가 과정 수료

저서『쉽게 배우는 R 데이터 분석, 이지스퍼블리싱』

Profile

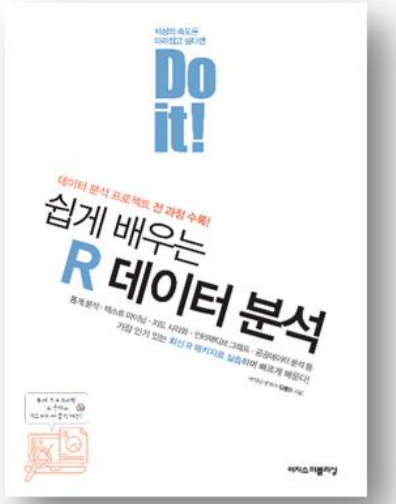
김영우

㈜히든그레이스 데이터분석팀장

facebook.com/groups/datacommunity

stats7445@gmail.com

010-5461-7445



1.안녕 R? - 친해지기

손맛 느껴보기

패키지 설치

```
install.packages("dplyr")
install.packages("ggplot2")
```

패키지 로드

```
library(dplyr)
library(ggplot2)
```

mpg 데이터 설명 보기

```
?mpg
```

데이터 검토

```
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl      trans   drv   cty   hwy    fl
##          <chr> <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>
## 1         audi    a4   1.8  1999     4   auto(l5)     f    18    29     p
## 2         audi    a4   1.8  1999     4 manual(m5)     f    21    29     p
## 3         audi    a4   2.0  2008     4 manual(m6)     f    20    31     p
## 4         audi    a4   2.0  2008     4   auto(av)     f    21    30     p
## 5         audi    a4   2.8  1999     6   auto(l5)     f    16    26     p
## 6         audi    a4   2.8  1999     6 manual(m5)     f    18    26     p
##     class
##     <chr>
## 1 compact
## 2 compact
## 3 compact
## 4 compact
## 5 compact
## 6 compact
```

```r
dim(mpg)
```

```
## [1] 234  11
```

```r
str(mpg)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    234 obs. of  11 variables:
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...
```

```r
summary(mpg)
```

```
##  manufacturer          model               displ           year
##  Length:234         Length:234         Min.   :1.600   Min.   :1999
##  Class :character   Class :character   1st Qu.:2.400   1st Qu.:1999
##  Mode  :character   Mode  :character   Median :3.300   Median :2004
##                                        Mean   :3.472   Mean   :2004
##                                        3rd Qu.:4.600   3rd Qu.:2008
##                                        Max.   :7.000   Max.   :2008
##       cyl           trans               drv                 cty
##  Min.   :4.000   Length:234         Length:234         Min.   : 9.00
##  1st Qu.:4.000   Class :character   Class :character   1st Qu.:14.00
##  Median :6.000   Mode  :character   Mode  :character   Median :17.00
##  Mean   :5.889                                         Mean   :16.86
##  3rd Qu.:8.000                                         3rd Qu.:19.00
##  Max.   :8.000                                         Max.   :35.00
##       hwy             fl                 class
##  Min.   :12.00   Length:234         Length:234
##  1st Qu.:18.00   Class :character   Class :character
##  Median :24.00   Mode  :character   Mode  :character
##  Mean   :23.44
##  3rd Qu.:27.00
##  Max.   :44.00
```

```
View(mpg)
```

데이터 분석

- 1.회사별 평균 연비 높은순 정렬

```
mpg %>%
  group_by(manufacturer) %>%
  summarise(mean.hwy=mean(hwy)) %>%
  arrange(desc(mean.hwy))
```

```
## # A tibble: 15 x 2
##    manufacturer mean.hwy
##           <chr>    <dbl>
## 1         honda 32.55556
## 2    volkswagen 29.22222
## 3       hyundai 26.85714
## 4          audi 26.44444
## 5       pontiac 26.40000
## 6        subaru 25.57143
## 7        toyota 24.91176
## 8        nissan 24.61538
## 9     chevrolet 21.89474
## 10         ford 19.36000
## 11      mercury 18.00000
## 12        dodge 17.94595
## 13         jeep 17.62500
```

```
## 14      lincoln 17.00000
## 15   land rover 16.50000
```

데이터 분석

- 2.포드 연비 높은순 정렬

```
mpg %>%
  filter(manufacturer=="ford") %>%
  group_by(model) %>%
  arrange(desc(hwy))
```

```
## Source: local data frame [25 x 11]
## Groups: model [4]
##
##    manufacturer           model displ  year   cyl      trans   drv   cty
##           <chr>           <chr> <dbl> <int> <int>      <chr> <chr> <int>
## 1          ford         mustang   3.8  1999     6 manual(m5)     r    18
## 2          ford         mustang   4.0  2008     6 manual(m5)     r    17
## 3          ford         mustang   3.8  1999     6   auto(l4)     r    18
## 4          ford         mustang   4.0  2008     6   auto(l5)     r    16
## 5          ford         mustang   4.6  2008     8 manual(m5)     r    15
## 6          ford         mustang   4.6  1999     8 manual(m5)     r    15
## 7          ford         mustang   4.6  2008     8   auto(l5)     r    15
## 8          ford         mustang   4.6  1999     8   auto(l4)     r    15
## 9          ford         mustang   5.4  2008     8 manual(m6)     r    14
## 10         ford    explorer 4wd   4.0  1999     6 manual(m5)     4    15
## 11         ford    explorer 4wd   4.0  2008     6   auto(l5)     4    13
## 12         ford    explorer 4wd   4.6  2008     8   auto(l6)     4    13
```

```
## 13         ford  expedition 2wd   5.4  2008     8   auto(l6)     r    12
## 14         ford  expedition 2wd   4.6  1999     8   auto(l4)     r    11
## 15         ford  expedition 2wd   5.4  1999     8   auto(l4)     r    11
## 16         ford    explorer 4wd   4.0  1999     6   auto(l5)     4    14
## 17         ford    explorer 4wd   4.0  1999     6   auto(l5)     4    14
## 18         ford    explorer 4wd   5.0  1999     8   auto(l4)     4    13
## 19         ford f150 pickup 4wd   4.2  1999     6   auto(l4)     4    14
## 20         ford f150 pickup 4wd   4.2  1999     6 manual(m5)     4    14
## 21         ford f150 pickup 4wd   4.6  2008     8   auto(l4)     4    13
## 22         ford f150 pickup 4wd   5.4  2008     8   auto(l4)     4    13
## 23         ford f150 pickup 4wd   4.6  1999     8 manual(m5)     4    13
## 24         ford f150 pickup 4wd   4.6  1999     8   auto(l4)     4    13
## 25         ford f150 pickup 4wd   5.4  1999     8   auto(l4)     4    11
##      hwy    fl      class
##    <int> <chr>      <chr>
## 1     26     r subcompact
## 2     26     r subcompact
## 3     25     r subcompact
## 4     24     r subcompact
## 5     23     r subcompact
## 6     22     r subcompact
## 7     22     r subcompact
## 8     21     r subcompact
## 9     20     p subcompact
## 10    19     r        suv
## 11    19     r        suv
```

```
## 12    19     r         suv
## 13    18     r         suv
## 14    17     r         suv
## 15    17     r         suv
## 16    17     r         suv
## 17    17     r         suv
## 18    17     r         suv
## 19    17     r      pickup
## 20    17     r      pickup
## 21    17     r      pickup
## 22    17     r      pickup
## 23    16     r      pickup
## 24    16     r      pickup
## 25    15     r      pickup
```

데이터 분석

- 3.배기량이 연비에 미치는 영향 회귀분석

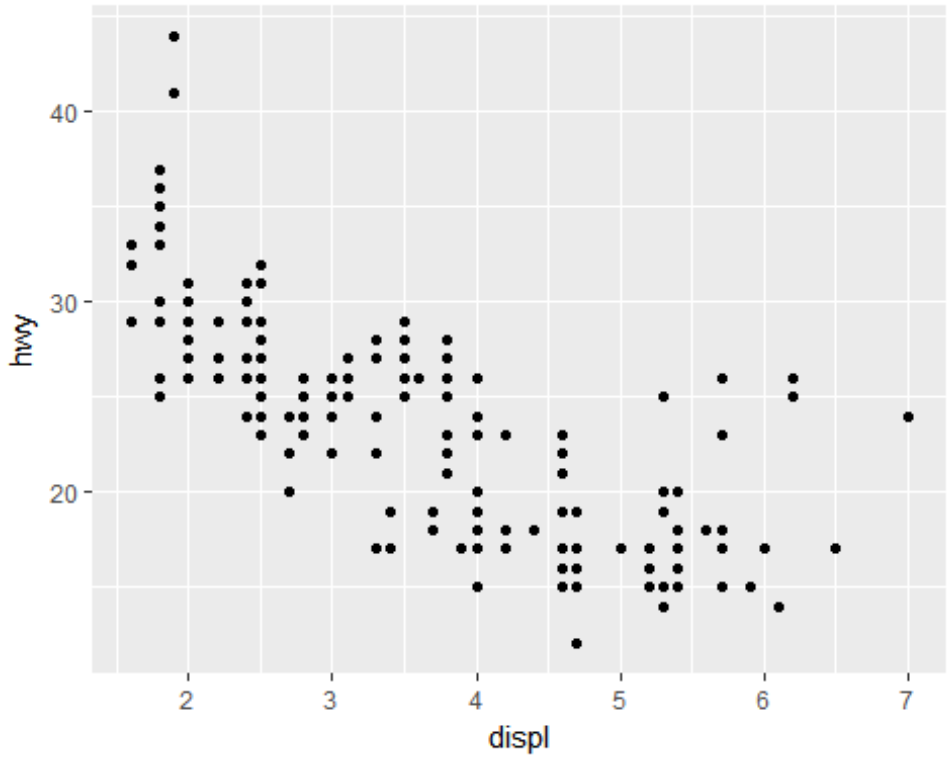
```
lm.mpg <- lm(data=mpg, hwy ~ displ)  # 회귀분석
summary(lm.mpg)                      # 결과 출력

##
## Call:
## lm(formula = hwy ~ displ, data = mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1039 -2.1646 -0.2242  2.0589 15.0105
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.6977     0.7204   49.55   <2e-16 ***
## displ        -3.5306     0.1945  -18.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.836 on 232 degrees of freedom
## Multiple R-squared:  0.5868, Adjusted R-squared:  0.585
## F-statistic: 329.5 on 1 and 232 DF,  p-value: < 2.2e-16
```

그래프 만들기

- 배기량과 연비 관계 그래프

```
qplot(data = mpg, x = displ, y = hwy)
```



2.연장 챙기기

변수

사용할 패키지 로드

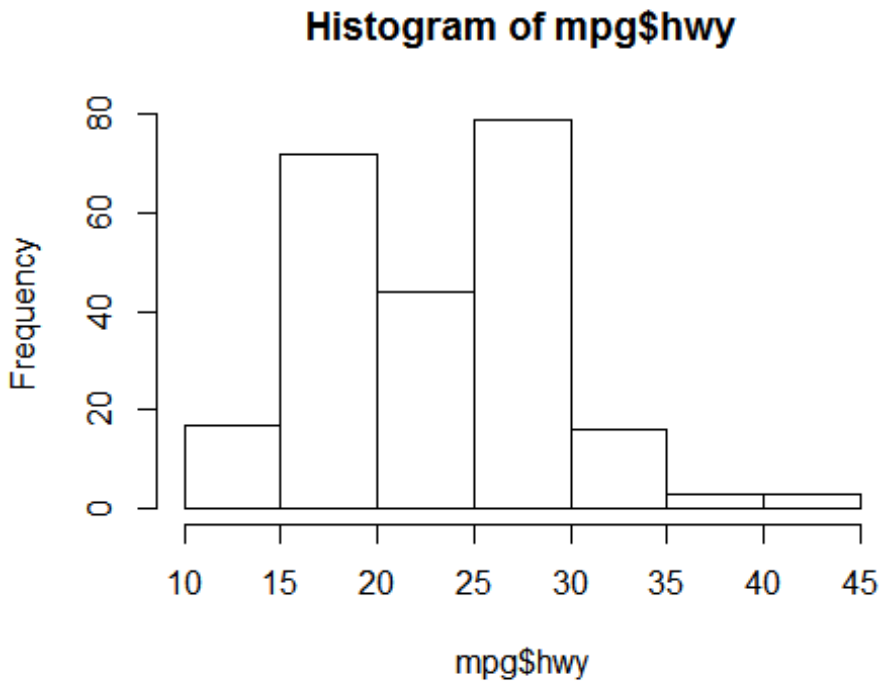
```
library(ggplot2)
```

mpg 데이터의 변수 다뤄보기

```
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl      trans   drv   cty   hwy    fl
##          <chr> <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>
## 1         audi    a4   1.8  1999     4   auto(l5)     f    18    29     p
## 2         audi    a4   1.8  1999     4 manual(m5)     f    21    29     p
## 3         audi    a4   2.0  2008     4 manual(m6)     f    20    31     p
## 4         audi    a4   2.0  2008     4   auto(av)     f    21    30     p
## 5         audi    a4   2.8  1999     6   auto(l5)     f    16    26     p
## 6         audi    a4   2.8  1999     6 manual(m5)     f    18    26     p
##     class
##     <chr>
## 1 compact
## 2 compact
## 3 compact
## 4 compact
## 5 compact
## 6 compact
```

```r
mean(mpg$hwy)
## [1] 23.44017
max(mpg$hwy)
## [1] 44
min(mpg$hwy)
## [1] 12
hist(mpg$hwy)
```



변수 만들기

```
a <- 1
a
```

```
## [1] 1
```

```
b <- 2
b
```

```
## [1] 2
```

```
c <- 3
c
```

```
## [1] 3
```

```
ab <- 3.5
ab
```

```
## [1] 3.5
```

변수로 연산하기

```
a+b
```

```
## [1] 3
```

```
a+b+c
```

```
## [1] 6
```

```
4/b
```

```
## [1] 2
```

```
5*b
```

```
## [1] 10
```

연속값 변수 만들기

```
d <- c(1,2,3,4,5)
d
```

```
## [1] 1 2 3 4 5
```

```
e <- c(1:5)
e
```

```
## [1] 1 2 3 4 5
```

```
f <- seq(1, 5)
f
```

```
## [1] 1 2 3 4 5
```

```
g <- seq(1, 10, by=2)  # 1~10 까지 2 씩 증가
g
```

```
## [1] 1 3 5 7 9
```

연속값 변수로 연산하기

```
d
```

```
## [1] 1 2 3 4 5
```

```
d+2
```

```
## [1] 3 4 5 6 7
```

```
d
```

```
## [1] 1 2 3 4 5
```

```
e
```

```
## [1] 1 2 3 4 5
```

```
d+e
```

```
## [1]  2  4  6  8 10
```

문자 변수 만들기

```
a2 <- "a"
a2
```

```
## [1] "a"
```

```
b2 <- "text"
b2
```

```
## [1] "text"
```

```
c2 <- "Hello world!"
c2
```

```
## [1] "Hello world!"
```

연속 문자 변수 만들기

```
d2 <- c("a", "b", "c")
d2
```

```
## [1] "a" "b" "c"
```

```
e2 <- c("Hello!", "World", "is", "good!")
e2
```

```
## [1] "Hello!" "World"  "is"     "good!"
```

문자 변수는 연산 불가

```
b2+2
```

```
## Error in b2 + 2: non-numeric argument to binary operator
```

```
a2+b2
```

```
## Error in a2 + b2: non-numeric argument to binary operator
```

함수

```
a <- c(1,2,3)
a
```

```
## [1] 1 2 3
```

```
mean(a)  # 평균
```

```
## [1] 2
```

```
max(a)   # 최대값
```

```
## [1] 3
```

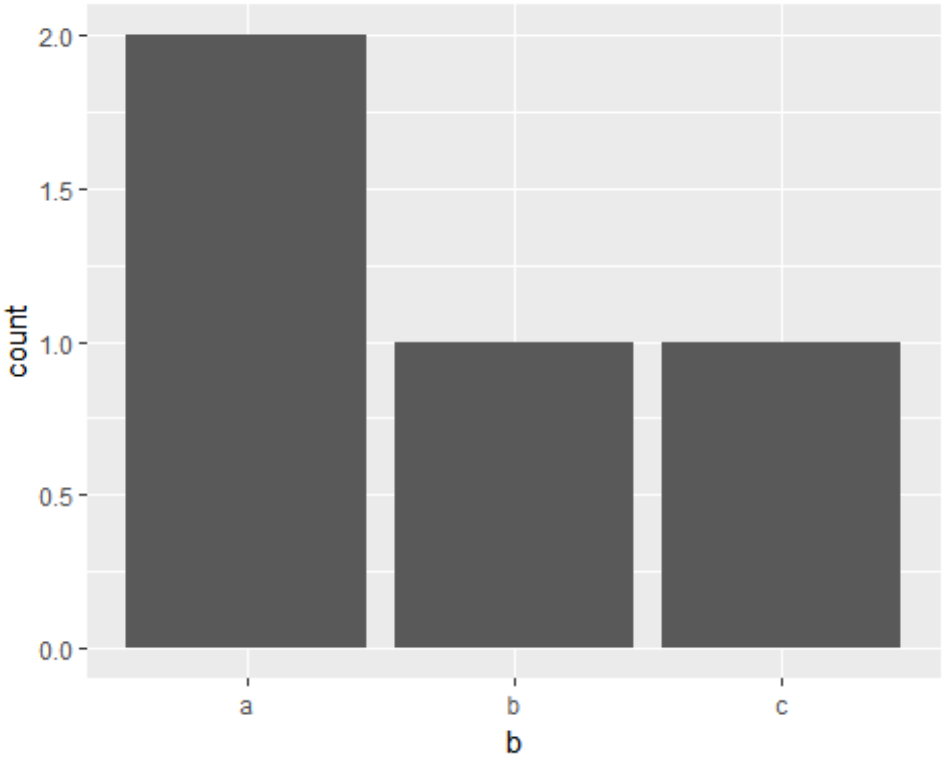
```
min(a)   # 최소값
```

```
## [1] 1
```

```r
b <- c("a","a","b","c")
b
```

```
## [1] "a" "a" "b" "c"
```

```r
qplot(b)  # 빈도 그래프 만들기
```



문자 처리 함수

```
e2
```

```
## [1] "Hello!" "World"  "is"     "good!"
```

문자 처리 함수

```r
paste(e2, collapse = " ")  # 빈칸 구분자로 문자 붙이기
```

```
## [1] "Hello! World is good!"
```

문자 처리 함수

```
e2_paste <- paste(e2, collapse = " ")  # 함수 출력 결과로 변수 만들기
e2_paste
```

```
## [1] "Hello! World is good!"
```

문자 처리 함수

```
e3_paste <- paste(e2, collapse = ",")
e3_paste
```

```
## [1] "Hello!,World,is,good!"
```

패키지 이용하기

- 스크립트 저장하기
- Workspace 저장하기
- Rstudio 재실행

패키지 이용하기

패키지 로드 안하면 함수 사용 불가

```
b
```

```
## [1] "a" "a" "b" "c"
```

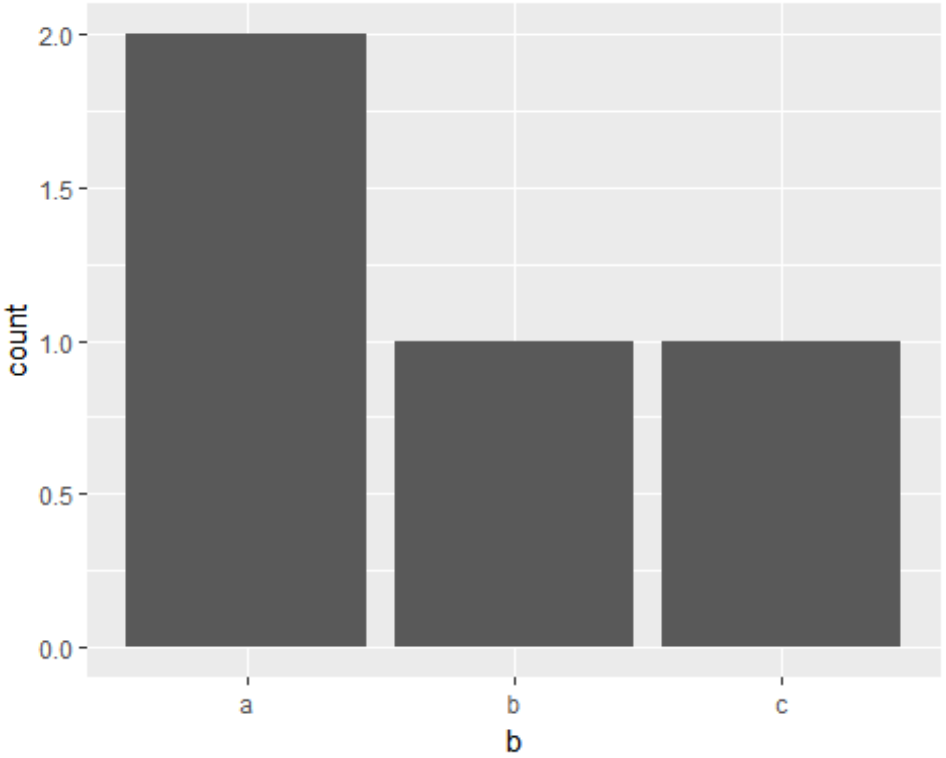
```
qplot(b)
```

```
## Error in qplot(b): could not find function "qplot"
```

```r
head(mpg)
```

```
## Error in head(mpg): object 'mpg' not found
```

```r
library(ggplot2)
qplot(b)
```



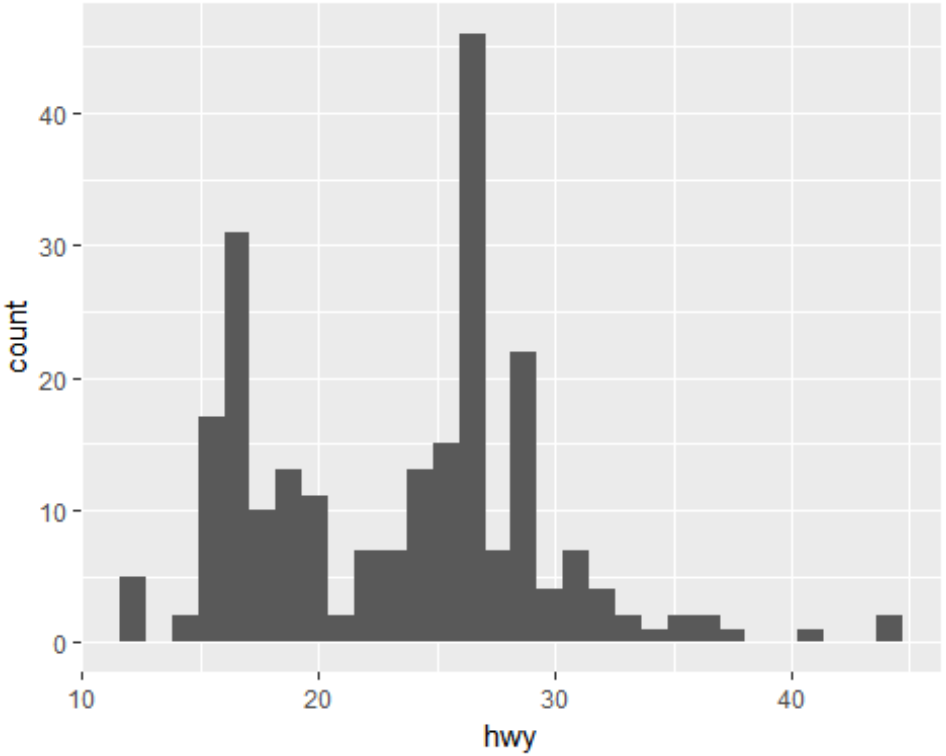

```r
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl      trans   drv   cty   hwy    fl
##          <chr> <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>
## 1         audi    a4   1.8  1999     4   auto(l5)     f    18    29     p
## 2         audi    a4   1.8  1999     4 manual(m5)     f    21    29     p
## 3         audi    a4   2.0  2008     4 manual(m6)     f    20    31     p
```

```
## 4         audi    a4   2.0  2008     4   auto(av)     f    21    30     p
## 5         audi    a4   2.8  1999     6   auto(l5)     f    16    26     p
## 6         audi    a4   2.8  1999     6 manual(m5)     f    18    26     p
##     class
##     <chr>
## 1 compact
## 2 compact
## 3 compact
## 4 compact
## 5 compact
## 6 compact
```

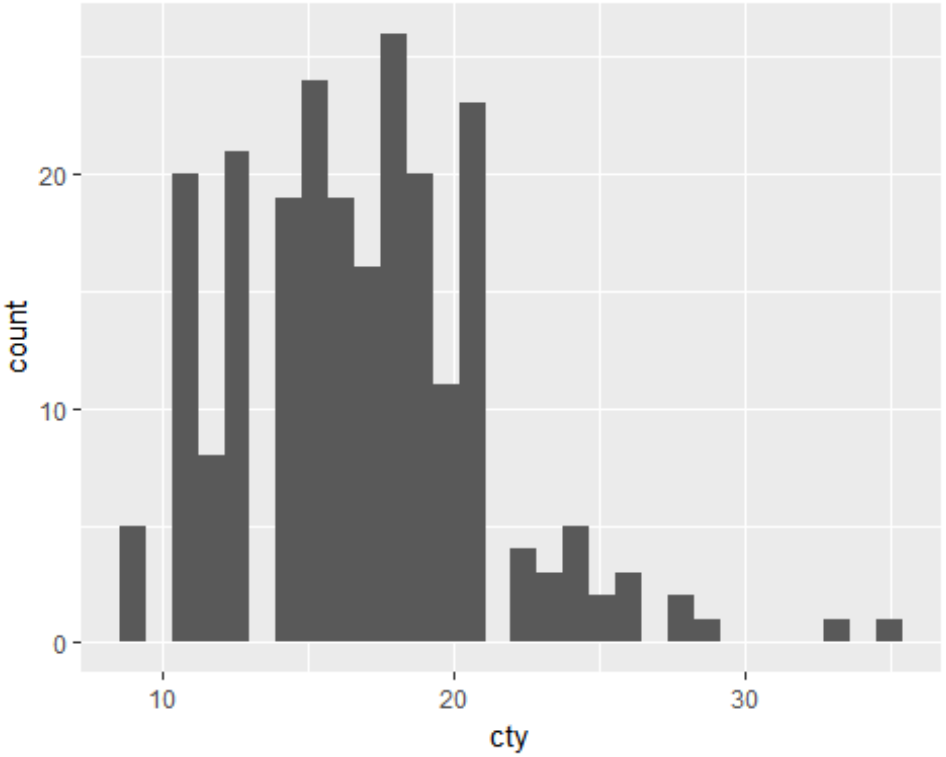
함수 파라미터(parameter) 지정하기

```
# 'x =' x 축
qplot(data = mpg, x = hwy)
```



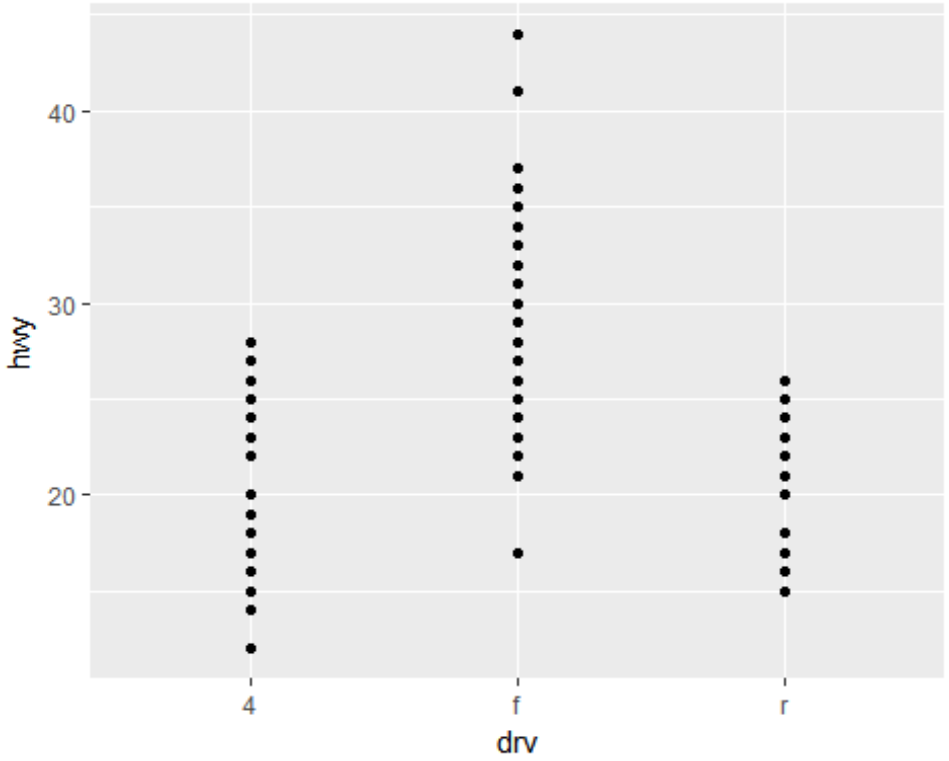
함수 파라미터(parameter) 지정하기

```
# 'x =' x 축
qplot(data = mpg, x = cty)
```



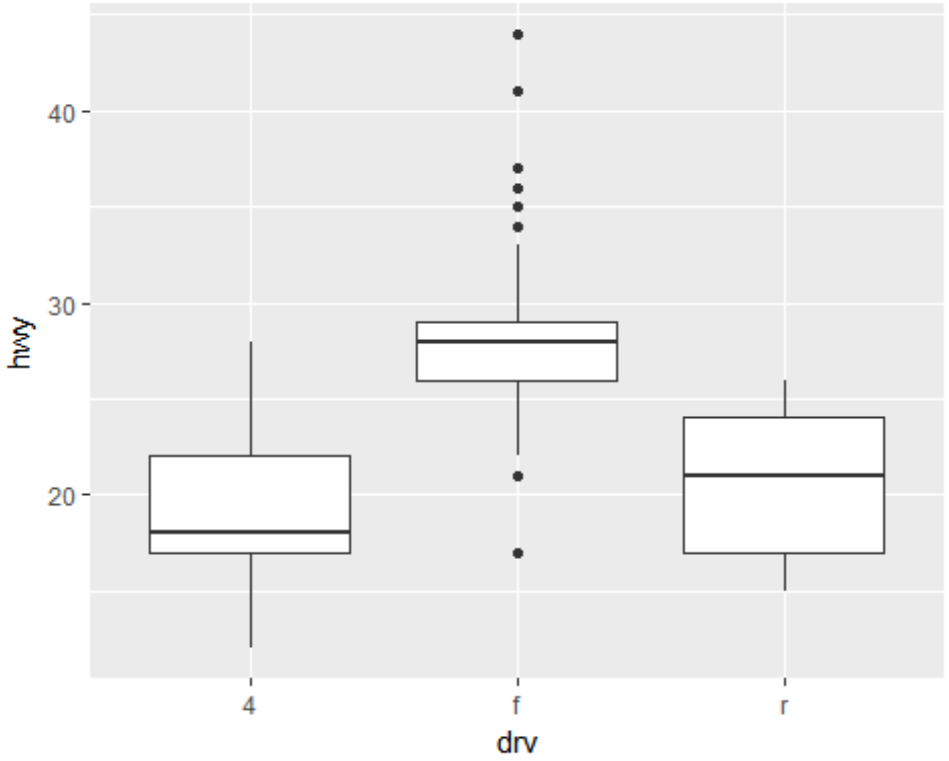
함수 파라미터(parameter) 지정하기

```
# 'geom = ' 그래프 형식
qplot(data = mpg, y = hwy, x = drv, geom = "point")
```



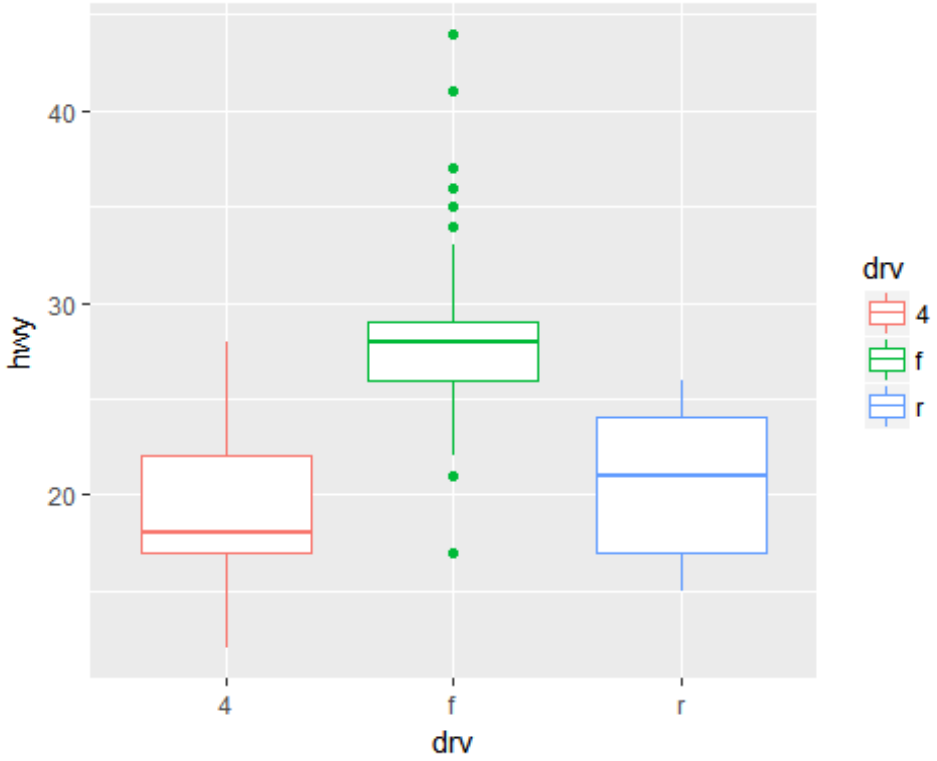
함수 파라미터(parameter) 지정하기

```
# 'geom = ' 그래프 형식
qplot(data = mpg, y = hwy, x = drv, geom = "boxplot")
```



함수 파라미터(parameter) 지정하기

```
# 'colour = ' 색깔 구분
qplot(data = mpg, y = hwy, x = drv, geom = "boxplot", colour = drv)
```



함수 사용법이 궁금할 땐 help

```
?qplot
```

3.데이터의 세계로

데이터 프레임 만들기

```
history <- c(90, 80, 60, 70)  # 역사점수 생성
history
```

```
## [1] 90 80 60 70
```

```
math <- c(50, 60, 100, 20)    # 수학점수 생성
math
```

```
## [1]  50  60 100  20
```

```r
# 변수 합해서 데이터프레임 만들기
df_midterm <- data.frame(history, math)
df_midterm
```

```
##   history math
## 1      90   50
## 2      80   60
## 3      60  100
## 4      70   20
```

```r
# 반 추가하기
class <- c(1, 1, 2, 2)
class
```

```
## [1] 1 1 2 2
```

```r
# 반 추가하기
class <- c(1, 1, 2, 2)
class
```

```
## [1] 1 1 2 2
```

```r
df_midterm <- data.frame(history, math, class)
df_midterm
```

```
##   history math class
## 1      90   50     1
## 2      80   60     1
## 3      60  100     2
## 4      70   20     2
```

```r
mean(df_midterm$history)
```

```
## [1] 75
```

```r
mean(df_midterm$math)
```

```
## [1] 57.5
```

외부 데이터 불러오기

엑셀 데이터 불러오기

```
# readxl 패키지 설치
install.packages("readxl")

# readxl 패키지 로드
library(readxl)
```

```r
# 엑셀 파일 불러오기
df_finalexam <- read_excel("finalexam.xlsx", sheet = 1, col_names = T)
```

[주의] Working directory에 불러올 파일이 있어야 함

```
df_finalexam
```

```
## # A tibble: 20 x 5
##       id class  math history english
##    <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 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
```

```r
mean(df_finalexam$math)
```

```
## [1] 57.45
```

```r
mean(df_finalexam$history)
```

```
## [1] 84.9
```

```r
mean(df_finalexam$english)
```

```
## [1] 59.45
```

col_names 파라미터

```
# 첫번째 행 변수명으로 로드
read_excel("finalexam.xlsx", sheet = 1, col_names = T)
```

```
## # A tibble: 20 x 5
##       id class  math history english
##    <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 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
```

```r
# 첫번째 행 변수명으로 로드 X
read_excel("finalexam.xlsx", sheet = 1, col_names = F)
```

```
## # A tibble: 21 x 5
##    X__1  X__2  X__3    X__4    X__5
##    <chr> <chr> <chr>   <chr>   <chr>
## 1     id class  math history english
## 2      1     1    50      98      50
## 3      2     1    60      97      60
## 4      3     1    45      86      78
## 5      4     1    30      98      58
## 6      5     2    25      80      65
## 7      6     2    50      89      98
## 8      7     2    80      90      45
## 9      8     2    90      78      25
## 10     9     3    20      98      15
## 11    10     3    50      98      45
## 12    11     3    65      65      65
## 13    12     3    45      85      32
## 14    13     4    46      98      65
## 15    14     4    48      87      12
## 16    15     4    75      56      78
## 17    16     4    58      98      65
## 18    17     5    65      68      98
## 19    18     5    80      78      90
```

```
## 20     19      5     89       68       87
## 21     20      5     78       83       58
```

csv 파일 불러오기

- 범용 데이터 형식
- 값 사이를 쉼표(,)로 구분
- 용량 작음, 다양한 소프트웨어에서 사용

```
read.csv("csv_exam.csv", header = T)
```

```
##    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
```

csv로 저장하기

```
#csv 로 저장
write.csv(df_finalexam, file = "output_newdata.csv")
```

rda 파일 이용하기

- R 전용 데이터 파일
- 용량 작고 빠름

```
# rda 로 저장 - r data 파일
save(df_finalexam, file = "output_newdata.rda")

# Environment 데이터 지우기
rm(df_finalexam)

# rda 불러오기
load("output_newdata.rda")
df_finalexam
```

```
## # A tibble: 20 x 5
##       id class  math history english
##    <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 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
```

목차

3.그래프 만들기

- 산점도
- 막대 그래프
- 선 그래프
- 상자그림

4.데이터 정제하기

- 빠진 데이터를 찾아라! - 결측치 정제하기
- 이상한 데이터를 찾아라! - 이상치 정제하기

데이터 갖고 놀기 #1 - 기초

1. 데이터 파악하기

함수	기능
head()	Raw 데이터 앞 부분 출력
tail()	Raw 데이터 뒷 부분 출력
View()	Raw 데이터 뷰어 창에서 확인
dim()	데이터 차원 출력
str()	데이터 속성 출력
summary()	변수의 요약통계량 출력

데이터 준비하기

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

head() - 데이터 앞부분 확인하기

```
head(exam)      # 앞에서부터 6 행까지 출력
```

```
##   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, 10)  # 앞에서부터 10 행까지 출력
```

```
##    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
```

tail() - 데이터 뒷부분 확인하기

```
tail(exam)      # 뒤에서부터 6 행까지 출력
```

```
##    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, 10)  # 뒤에서부터 10 행까지 출력
```

```
##    id class math english science
## 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
```

View() - 뷰어 창에서 데이터 확인하기

```
# 데이터 뷰어 창에서 exam 데이터 확인
View(exam)
```

[유의] View()에서 맨 앞의 V는 대문자

dim() - 몇 행 몇 열로 구성되는지 알아보기

```
dim(exam)  # 행, 열 출력
```

```
## [1] 20  5
```

str() - 속성 파악하기

```
str(exam)  # 데이터 속성 확인

## '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 ...
```

summary() - 요약통계량 산출하기

```
summary(exam)  # 요약통계량 출력
```

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

mpg 데이터 파악하기

```
# ggplo2 의 mpg 데이터를 데이터 프레임 형태로 가져오기
mpg <- as.data.frame(ggplot2::mpg)
```

mpg 데이터 파악하기

```
head(mpg)    # Raw 데이터 앞부분 확인
```

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

```
tail(mpg)    # Raw 데이터 뒷부분 확인
```

```
##     manufacturer  model displ year cyl      trans drv cty hwy fl   class
## 229   volkswagen passat   1.8 1999   4   auto(l5)   f  18  29  p midsize
## 230   volkswagen passat   2.0 2008   4   auto(s6)   f  19  28  p midsize
## 231   volkswagen passat   2.0 2008   4 manual(m6)   f  21  29  p midsize
## 232   volkswagen passat   2.8 1999   6   auto(l5)   f  16  26  p midsize
## 233   volkswagen passat   2.8 1999   6 manual(m5)   f  18  26  p midsize
## 234   volkswagen passat   3.6 2008   6   auto(s6)   f  17  26  p midsize
```

```r
View(mpg)    # Raw 데이터 뷰어 창 확인

dim(mpg)     # 행, 열 출력
```

```
## [1] 234  11
```

```r
str(mpg)     # 데이터 속성 확인
```

```
## 'data.frame':    234 obs. of  11 variables:
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...
```

```r
summary(mpg)  # 요약통계량 출력
```

```
##  manufacturer          model               displ            year
##  Length:234         Length:234         Min.   :1.600   Min.   :1999
##  Class :character   Class :character   1st Qu.:2.400   1st Qu.:1999
##  Mode  :character   Mode  :character   Median :3.300   Median :2004
##                                        Mean   :3.472   Mean   :2004
##                                        3rd Qu.:4.600   3rd Qu.:2008
##                                        Max.   :7.000   Max.   :2008
##       cyl            trans               drv                 cty
##  Min.   :4.000   Length:234         Length:234         Min.   : 9.00
##  1st Qu.:4.000   Class :character   Class :character   1st Qu.:14.00
##  Median :6.000   Mode  :character   Mode  :character   Median :17.00
##  Mean   :5.889                                         Mean   :16.86
##  3rd Qu.:8.000                                         3rd Qu.:19.00
##  Max.   :8.000                                         Max.   :35.00
##       hwy             fl                 class
##  Min.   :12.00   Length:234         Length:234
##  1st Qu.:18.00   Class :character   Class :character
##  Median :24.00   Mode  :character   Mode  :character
##  Mean   :23.44
##  3rd Qu.:27.00
##  Max.   :44.00
```

2. 데이터 수정하기 - 변수명 바꾸기

dplyr 패키지 설치 & 로드

```
install.packages("dplyr")  # dplyr 설치
library(dplyr)             # dplyr 로드
```

데이터 프레임 생성

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

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

1. 데이터 프레임 복사본 만들기

```
df_new <- df_raw  # 복사본 생성
df_new            # 출력
```

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

2. 변수명 바꾸기

```
df_new <- rename(df_new, v2 = var2)  # var2 를 v2 로 수정
df_new
```

```
##   var1 v2
## 1    1  2
## 2    2  3
## 3    1  2
```

[유의] rename()에 '새 변수명 = 기존 변수명' 순서로 입력

수정 전후 비교

```
df_raw
```

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

```
df_new
```

```
##   var1 v2
## 1    1  2
## 2    2  3
## 3    1  2
```

1분 퀴즈

mpg 데이터의 cty 변수는 도시 연비를 의미하고 hwy 변수는 고속도로 연비를 의미합니다. mpg 데이터를 이용해서 아래 문제를 해결해보세요.

- 1.mpg 데이터의 복사본을 생성하세요.
- 2.cty는 city로, hwy는 highway로 변수명을 수정하세요.
- 3.데이터 일부를 출력해서 변수명이 바뀐 것을 확인해보세요.

최종 출력 결과물

```
##   manufacturer model displ year cyl      trans drv city highway fl   class
## 1         audi    a4   1.8 1999   4   auto(l5)   f   18      29  p compact
## 2         audi    a4   1.8 1999   4 manual(m5)   f   21      29  p compact
## 3         audi    a4   2.0 2008   4 manual(m6)   f   20      31  p compact
## 4         audi    a4   2.0 2008   4   auto(av)   f   21      30  p compact
## 5         audi    a4   2.8 1999   6   auto(l5)   f   16      26  p compact
## 6         audi    a4   2.8 1999   6 manual(m5)   f   18      26  p compact
```

1분 퀴즈

mpg 데이터의 cty 변수는 도시 연비를 의미하고 hwy 변수는 고속도로 연비를 의미합니다. mpg 데이터를 이용해서 아래 문제를 해결해보세요.

- 1.mpg 데이터의 복사본을 생성하세요.
- 2.cty는 city로, hwy는 highway로 변수명을 수정하세요.
- 3.데이터 일부를 출력해서 변수명이 바뀐 것을 확인해보세요.

```
mpg_new <- mpg                                  # 복사본 생성
mpg_new <- rename(mpg_new, city = cty)          # cty 를 city 로 수정
mpg_new <- rename(mpg_new, highway = hwy)       # hwy 를 highway 로 수정
head(mpg_new)                                   # 데이터 일부 출력
```

3. 데이터 수정하기 - 파생변수 만들기

변수 조합하기

데이터 프레임 생성

```
df <- df_raw
df
```

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

파생변수 생성

```
df$var_sum <- df$var1 + df$var2  # var_sum 파생변수 생성
df
```

```
##   var1 var2 var_sum
## 1    1    2       3
## 2    2    3       5
## 3    1    2       3
```

```r
df$var_mean <- (df$var1 + df$var2)/2  # var_mean 파생변수 생성
df
```

```
##   var1 var2 var_sum var_mean
## 1    1    2       3      1.5
## 2    2    3       5      2.5
## 3    1    2       3      1.5
```

mpg 통합 연비 변수 만들기

```
mpg$total <- (mpg$cty + mpg$hwy)/2  # 통합 연비 변수 생성

head(mpg)

##   manufacturer model displ year cyl      trans drv cty hwy fl   class
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact
## 4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact
## 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact
## 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact
##   total
## 1  23.5
## 2  25.0
## 3  25.5
## 4  25.5
## 5  21.0
## 6  22.0

mean(mpg$total)

## [1] 20.14957
```

조건문 활용하기 - 고연비 합격 판정 변수 만들기

기준값 정하기

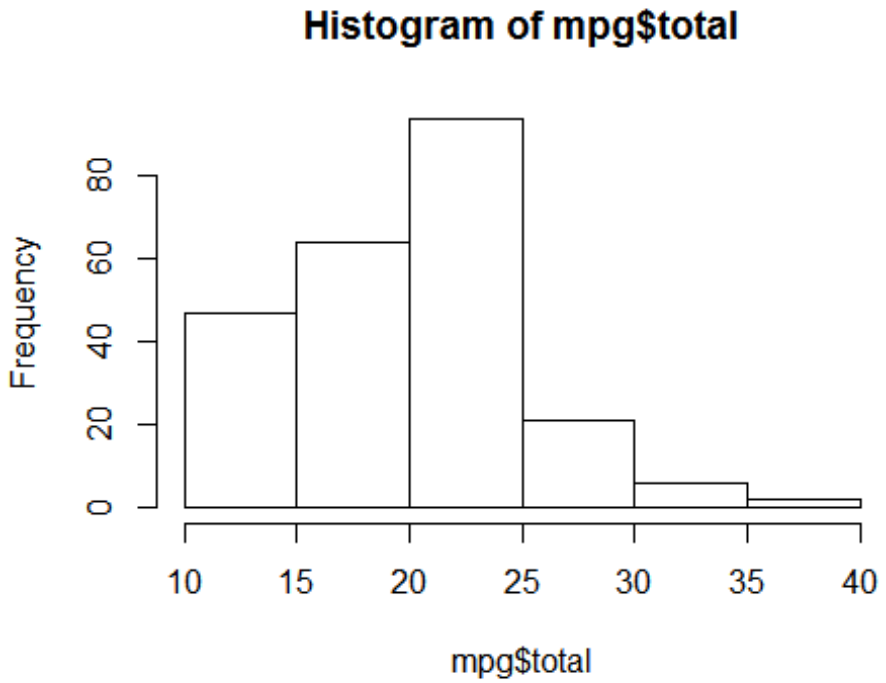
```
summary(mpg$total)  # 요약통계량 산출
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.50   15.50   20.50   20.15   23.50   39.50
```

조건문 활용하기 - 고연비 합격 판정 변수 만들기

기준값 정하기

```
hist(mpg$total)      # 히스토그램 생성
```



조건문으로 합격 판정 변수 만들기

```
# 20 이상이면 pass, 그렇지 않으면 fail 부여
mpg$test <- ifelse(mpg$total >= 20, "pass", "fail")

head(mpg, 20) # 데이터 확인
```

```
##    manufacturer                model displ year cyl      trans drv cty hwy
## 1          audi                   a4   1.8 1999   4   auto(l5)   f  18  29
## 2          audi                   a4   1.8 1999   4 manual(m5)   f  21  29
## 3          audi                   a4   2.0 2008   4 manual(m6)   f  20  31
## 4          audi                   a4   2.0 2008   4   auto(av)   f  21  30
## 5          audi                   a4   2.8 1999   6   auto(l5)   f  16  26
## 6          audi                   a4   2.8 1999   6 manual(m5)   f  18  26
## 7          audi                   a4   3.1 2008   6   auto(av)   f  18  27
## 8          audi          a4 quattro   1.8 1999   4 manual(m5)   4  18  26
## 9          audi          a4 quattro   1.8 1999   4   auto(l5)   4  16  25
## 10         audi          a4 quattro   2.0 2008   4 manual(m6)   4  20  28
## 11         audi          a4 quattro   2.0 2008   4   auto(s6)   4  19  27
## 12         audi          a4 quattro   2.8 1999   6   auto(l5)   4  15  25
## 13         audi          a4 quattro   2.8 1999   6 manual(m5)   4  17  25
## 14         audi          a4 quattro   3.1 2008   6   auto(s6)   4  17  25
## 15         audi          a4 quattro   3.1 2008   6 manual(m6)   4  15  25
## 16         audi          a6 quattro   2.8 1999   6   auto(l5)   4  15  24
## 17         audi          a6 quattro   3.1 2008   6   auto(s6)   4  17  25
```

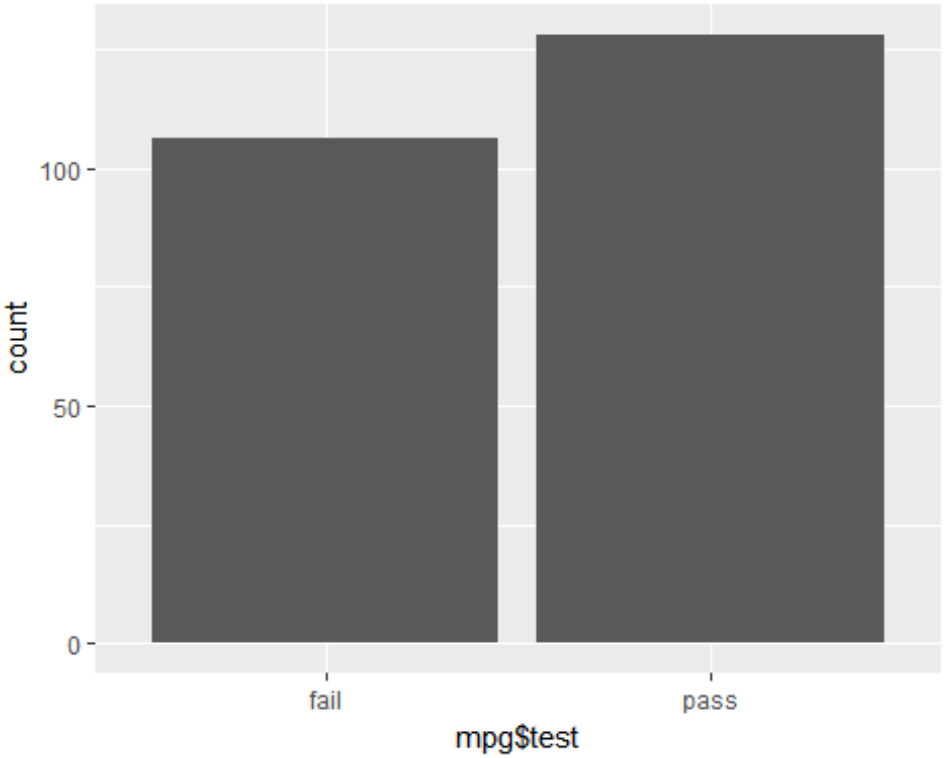
```
## 18         audi          a6 quattro   4.2 2008   8   auto(s6)   4  16  23
## 19    chevrolet c1500 suburban 2wd   5.3 2008   8   auto(l4)   r  14  20
## 20    chevrolet c1500 suburban 2wd   5.3 2008   8   auto(l4)   r  11  15
##    fl   class total test
## 1   p compact  23.5 pass
## 2   p compact  25.0 pass
## 3   p compact  25.5 pass
## 4   p compact  25.5 pass
## 5   p compact  21.0 pass
## 6   p compact  22.0 pass
## 7   p compact  22.5 pass
## 8   p compact  22.0 pass
## 9   p compact  20.5 pass
## 10  p compact  24.0 pass
## 11  p compact  23.0 pass
## 12  p compact  20.0 pass
## 13  p compact  21.0 pass
## 14  p compact  21.0 pass
## 15  p compact  20.0 pass
## 16  p midsize  19.5 fail
## 17  p midsize  21.0 pass
## 18  p midsize  19.5 fail
## 19  r     suv  17.0 fail
## 20  e     suv  13.0 fail
```

빈도표, 막대 그래프로 합격 판정 살펴보기

```
table(mpg$test)  # 연비 합격 빈도표 생성
```

```
##
## fail pass
##  106  128
```

```r
library(ggplot2)  # ggplot2 로드
qplot(mpg$test)   # 연비 합격 빈도 막대 그래프 생성
```



중첩 조건문 활용하기 - 연비 등급 변수 만들기

등급	total 기준
A	30 이상
B	20~29
C	20 미만

```r
# total 을 기준으로 A, B, C 등급 부여
mpg$grade <- ifelse(mpg$total >= 30, "A",
                    ifelse(mpg$total >= 20, "B", "C"))
```

[유의] ifelse()가 두 번 반복되므로 열리는 괄호와 닫히는 괄호가 각각 두 개, 쉼표도 각각 두 개

```r
head(mpg, 20)  # 데이터 확인
```

```
##    manufacturer                model displ year cyl      trans drv cty hwy
## 1          audi                   a4   1.8 1999   4   auto(l5)   f  18  29
## 2          audi                   a4   1.8 1999   4 manual(m5)   f  21  29
## 3          audi                   a4   2.0 2008   4 manual(m6)   f  20  31
## 4          audi                   a4   2.0 2008   4   auto(av)   f  21  30
## 5          audi                   a4   2.8 1999   6   auto(l5)   f  16  26
## 6          audi                   a4   2.8 1999   6 manual(m5)   f  18  26
## 7          audi                   a4   3.1 2008   6   auto(av)   f  18  27
## 8          audi            a4 quattro   1.8 1999   4 manual(m5)   4  18  26
## 9          audi            a4 quattro   1.8 1999   4   auto(l5)   4  16  25
## 10         audi            a4 quattro   2.0 2008   4 manual(m6)   4  20  28
## 11         audi            a4 quattro   2.0 2008   4   auto(s6)   4  19  27
## 12         audi            a4 quattro   2.8 1999   6   auto(l5)   4  15  25
## 13         audi            a4 quattro   2.8 1999   6 manual(m5)   4  17  25
## 14         audi            a4 quattro   3.1 2008   6   auto(s6)   4  17  25
## 15         audi            a4 quattro   3.1 2008   6 manual(m6)   4  15  25
## 16         audi            a6 quattro   2.8 1999   6   auto(l5)   4  15  24
## 17         audi            a6 quattro   3.1 2008   6   auto(s6)   4  17  25
## 18         audi            a6 quattro   4.2 2008   8   auto(s6)   4  16  23
## 19    chevrolet c1500 suburban 2wd   5.3 2008   8   auto(l4)   r  14  20
## 20    chevrolet c1500 suburban 2wd   5.3 2008   8   auto(l4)   r  11  15
##    fl   class total test grade
## 1   p compact  23.5 pass     B
## 2   p compact  25.0 pass     B
```

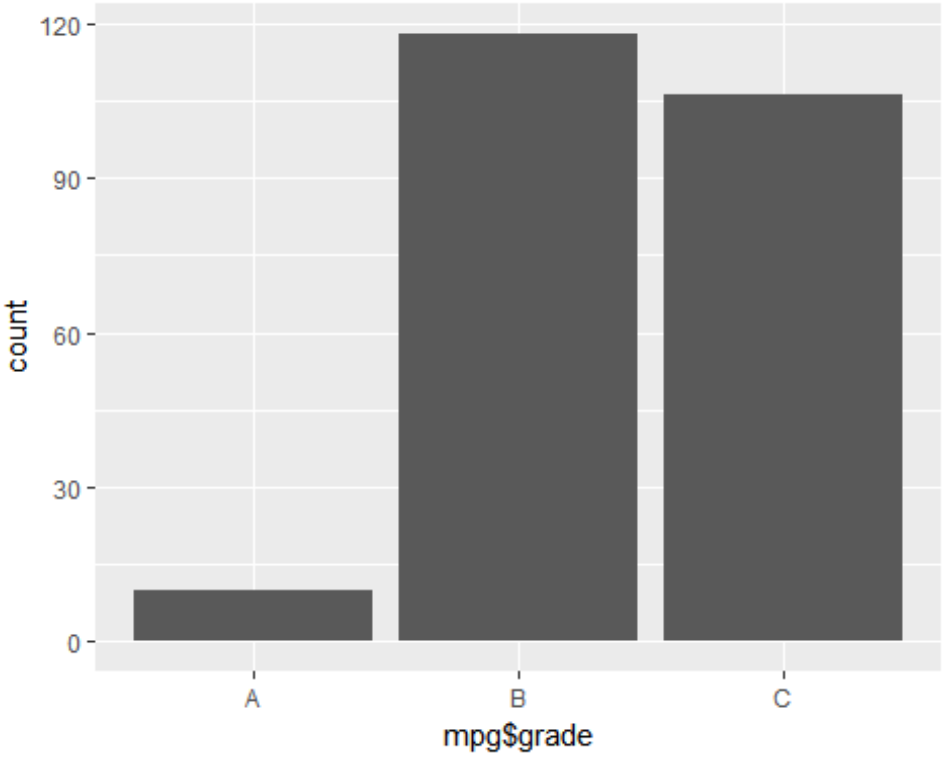
```
## 3   p compact  25.5 pass     B
## 4   p compact  25.5 pass     B
## 5   p compact  21.0 pass     B
## 6   p compact  22.0 pass     B
## 7   p compact  22.5 pass     B
## 8   p compact  22.0 pass     B
## 9   p compact  20.5 pass     B
## 10  p compact  24.0 pass     B
## 11  p compact  23.0 pass     B
## 12  p compact  20.0 pass     B
## 13  p compact  21.0 pass     B
## 14  p compact  21.0 pass     B
## 15  p compact  20.0 pass     B
## 16  p midsize  19.5 fail     C
## 17  p midsize  21.0 pass     B
## 18  p midsize  19.5 fail     C
## 19  r     suv  17.0 fail     C
## 20  e     suv  13.0 fail     C
```

빈도표, 막대 그래프로 연비 등급 살펴보기

```
table(mpg$grade) # 등급 빈도표 생성
```

```
##
##   A   B   C
##  10 118 106
```

```
qplot(mpg$grade) # 등급 빈도 막대 그래프 생성
```



[정리하기]

```r
# 1. 데이터, 패키지 준비
mpg <- as.data.frame(ggplot2::mpg)  # 데이터 가져오기
library(dplyr)                      # dplyr 로드
library(ggplot2)                    # ggplot2 로드

# 2. 데이터 파악
head(mpg)    # Raw 데이터 앞부분
tail(mpg)    # Raw 데이터 뒷부분
View(mpg)    # Raw 데이터 뷰어 창
dim(mpg)     # 차원
str(mpg)     # 속성
summary(mpg) # 요약통계량
```

```r
# 3. 변수명 수정
mpg <- rename(mpg, company = manufacturer)  # 변수명 수정

# 4. 파생변수 생성
mpg$total <- (mpg$cty + mpg$hwy)/2                    # 변수 조합
mpg$test <- ifelse(mpg$total >= 20, "pass", "fail")  # 조건문 활용

# 5. 빈도 확인
table(mpg$test)  # 빈도표 출력
qplot(mpg$test)  # 막대 그래프 생성
```

[분석 도전]

ggplot2 패키지에는 미국 동북중부 437개 지역의 인구통계 정보를 담은 midwest라는 데이터가 포함되어 있습니다. midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

[분석 도전]

midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

- 1.ggplot2의 midwest 데이터를 데이터 프레임 형태로 불러와서 데이터의 특성을 파악하세요.

[분석 도전]

midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

- 1.ggplot2의 midwest 데이터를 데이터 프레임 형태로 불러와서 데이터의 특성을 파악하세요.

```
midwest <- as.data.frame(ggplot2::midwest)
head(midwest)
tail(midwest)
View(midwest)
dim(midwest)
str(midwest)
summary(midwest)
```

[분석 도전]

midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

- 2.poptotal(전체 인구) 변수를 total로, popasian(아시아 인구) 변수를 asian으로 변수명을 수정하세요.

[분석 도전]

midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

- 2.poptotal(전체 인구) 변수를 total로, popasian(아시아 인구) 변수를 asian으로 변수명을 수정하세요.

```
library(dplyr)
midwest <- rename(midwest, total = poptotal)
midwest <- rename(midwest, asian = popasian)
```

[분석 도전]

midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

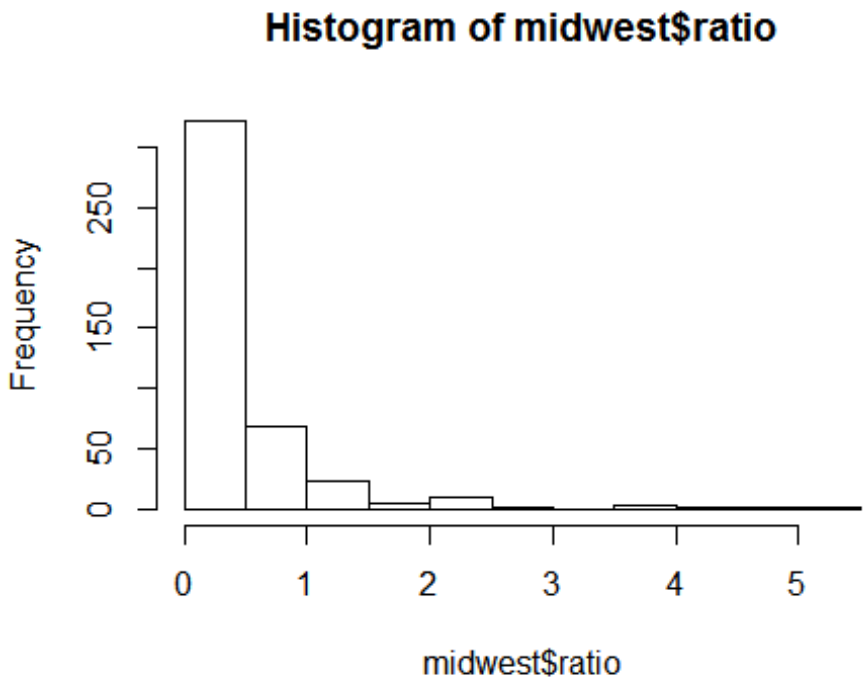
- 3.total, asian 변수를 이용해서 '전체 인구 대비 아시아 인구 백분율' 파생변수를 만들고, 히스토그램을 그려서 도시들이 어떻게 분포하는지 살펴보세요.

[분석 도전]

midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

- 3.total, asian 변수를 이용해서 '전체 인구 대비 아시아 인구 백분율' 파생변수를 만들고, 히스토그램을 그려서 도시들이 어떻게 분포하는지 살펴보세요.

```
midwest$ratio <- midwest$asian/midwest$total*100
hist(midwest$ratio)
```


Histogram of midwest$ratio
Frequency
0
50
150
250
0
1
2
3
4
5
midwest$ratio

[분석 도전]

midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

- 4.아시아 인구 백분율 전체 평균을 초과하면 "large", 그 외에는 "small"을 부여하는 파생변수를 만들어 보세요.

[분석 도전]

midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

- 4.아시아 인구 백분율 전체 평균을 초과하면 "large", 그 외에는 "small"을 부여하는 파생변수를 만들어 보세요.

```
mean(midwest$ratio)
```

```
## [1] 0.4872462
```

```
midwest$group <- ifelse(midwest$ratio > 0.4872, "large", "small")
```

[분석 도전]

midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

- 5."large"와 "small"에 해당하는 지역이 얼마나 되는지 빈도표와 빈도 막대 그래프를 만들어 확인해보세요.

[분석 도전]

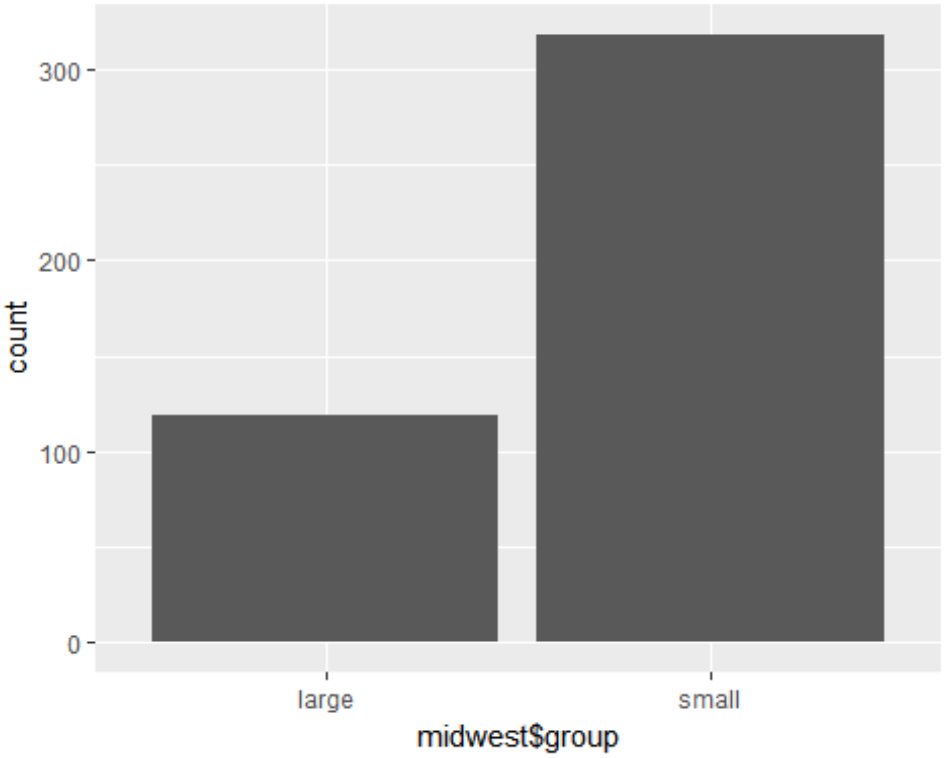
midwest 데이터를 사용해서 데이터 분석 문제를 해결해보세요.

- 5."large"와 "small"에 해당하는 지역이 얼마나 되는지 빈도표와 빈도 막대 그래프를 만들어 확인해보세요.

```
table(midwest$group)
```

```
##
## large small
##   119   318
```

```
library(ggplot2)
qplot(midwest$group)
```


300
200
100
0
count
large
small
midwest$group

데이터 갖고 놀기 #2 - 고급지게!

데이터 전처리(Preprocessing, Manipulation, Handling, Wrangling, Munging)

- 분석에 적합하게 데이터를 가공하는 작업
  - 조건에 맞는 데이터만 추출
  - 필요한 변수만 추출
  - 순서대로 정렬
  - 파생변수 추가
  - 집단별로 요약
  - 데이터 합치기

데이터 갖고 놀기 #2 - 고급지게!

dplyr 패키지

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

1. 조건에 맞는 데이터만 추출하기 - filter()

dplyr 패키지 로드

```
library(dplyr)
```

조건에 맞는 행 추출

```
# exam 에서 class 가 1 인 경우만 추출하여 출력
exam %>% filter(class == 1)
```

```
##   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
```

[참고] 단축키 [Ctrl+Shit+M]으로 %>% 기호 입력

```r
# 2 반인 경우만 추출
exam %>% filter(class == 2)
```

```
##   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
```

```r
# 1 반이 아닌 경우
exam %>% filter(class != 1)
```

```
##    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
## 14 18     5   80      78      90
## 15 19     5   89      68      87
## 16 20     5   78      83      58
```

```r
# 3 반이 아닌 경우
exam %>% filter(class != 3)
```

```
##    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  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
## 14 18     5   80      78      90
## 15 19     5   89      68      87
## 16 20     5   78      83      58
```

초과, 미만, 이상, 이하 조건 걸기

```
# 수학점수가 50 점을 초과한 경우
exam %>% filter(math > 50)
```

```
##    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
```

```r
# 수학점수가 50 점 미만인 경우
exam %>% filter(math < 50)
```

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

```r
# 영어점수가 80 점 이상인 경우
exam %>% filter(english >= 80)
```

```
##    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
## 13 16     4   58      98      65
## 14 20     5   78      83      58
```

```r
# 영어점수가 80 점 이하인 경우
exam %>% filter(english <= 80)
```

```
##   id class math english science
## 1  5     2   25      80      65
## 2  8     2   90      78      25
## 3 11     3   65      65      65
## 4 15     4   75      56      78
## 5 17     5   65      68      98
## 6 18     5   80      78      90
## 7 19     5   89      68      87
```

여러 조건을 충족하는 행 추출하기

```
# 1 반 이면서 수학점수가 50 점 이상인 경우
exam %>% filter(class == 1 & math >= 50)
```

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

```r
# 2 반 이면서 영어점수가 80 점 이상인 경우
exam %>% filter(class == 2 & english >= 80)
```

```
##   id class math english science
## 1  5     2   25      80      65
## 2  6     2   50      89      98
## 3  7     2   80      90      45
```

여러 조건 중 하나 이상 충족하는 행 추출하기

```
# 수학점수가 90 점 이상이거나 영어점수가 90 점 이상인 경우
exam %>% filter(math >= 90 | english >= 90)
```

```
##   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
```

```r
# 영어점수가 90 점 미만이거나 과학점수가 50 점 미만인 경우
exam %>% filter(english < 90 | science < 50)
```

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

목록에 해당되는 행 추출하기

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

```
##    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
## 12 20     5   78      83      58
```

%in% 기호 이용하기

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

```
##    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
## 12 20     5   78      83      58
```

추출한 행으로 데이터 만들기

- 1반, 2반 새 데이터 생성해서 각 반 수학점수 평균 구하기

```
class1 <- exam %>% filter(class == 1)  # class 가 1 인 행 추출, class1 에 할당
class2 <- exam %>% filter(class == 2)  # class 가 2 인 행 추출, class2 에 할당

mean(class1$math)                      # 1 반 수학점수 평균 구하기
## [1] 46.25
mean(class2$math)                      # 2 반 수학점수 평균 구하기
## [1] 61.25
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.배기량(displ)이 4 이하인 자동차와 5 이상인 자동차 중 어떤 자동차의 고속도로 연비가 더 높은가요?

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.배기량(displ)이 4 이하인 자동차와 5 이상인 자동차 중 어떤 자동차의 고속도로 연비가 더 높은가요?

```
mpg_a <- mpg %>% filter(displ <= 4)  # displ 4 이하 추출
mpg_b <- mpg %>% filter(displ >= 5)  # displ 5 이상 추출

mean(mpg_a$hwy)  # displ 4 이하 hwy 평균
## [1] 25.96319
mean(mpg_b$hwy)  # displ 5 이상 hwy 평균
## [1] 18.07895
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 2.audi와 toyota 중 어느 회사 자동차의 도시 연비가 더 높은가요?(힌트: 문자열을 입력할 때는 앞 뒤에 겹따옴표를 붙여야 합니다)

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 2.audi와 toyota 중 어느 회사 자동차의 도시 연비가 더 높은가요?(힌트: 문자열을 입력할 때는 앞 뒤에 겹따옴표를 붙여야 합니다)

```
mpg_audi <- mpg %>% filter(manufacturer == "audi")       # audi 추출
mpg_toyota <- mpg %>% filter(manufacturer == "toyota")  # toyota 추출

mean(mpg_audi$cty)    # audi 의 cty 평균
## [1] 17.61111
mean(mpg_toyota$cty)  # toyota 의 cty 평균
## [1] 18.52941
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 3.chevrolet, ford, honda 자동차만 추출해서 새로운 변수에 할당하고 고속도로 연비 평균을 구하세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 3.chevrolet, ford, honda 자동차만 추출해서 새로운 변수에 할당하고 고속도로 연비 평균을 구하세요.

```
# class 가 chevrolet, ford, honda 에 해당되면 추출, mpg_new 에 할당
mpg_new <- mpg %>% filter(manufacturer %in% c("chevrolet", "ford", "honda"))
mean(mpg_new$hwy)
```

```
## [1] 22.50943
```

R에서 사용하는 기호들

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

R에서 사용하는 기호들

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

2. 필요한 변수만 추출하기 - select()

```
exam %>% select(math)  # math 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
```

```r
exam %>% select(english)  # english 추출
```

```
##    english
## 1       98
## 2       97
## 3       86
## 4       98
## 5       80
## 6       89
## 7       90
## 8       78
## 9       98
## 10      98
## 11      65
## 12      85
## 13      98
## 14      87
## 15      56
## 16      98
## 17      68
## 18      78
## 19      68
## 20      83
```

여러 변수 추출하기

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

변수 제외하기

```
exam %>% select(-math)  # math 제외
```

```
##    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
```

```r
exam %>% select(-math, -english)  # math, english 제외
```

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

filter()와 select() 조합하기

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

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

dplyr Tip - 줄 바꿔서 가독성 높은 코드 만들기

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

dplyr Tip - 일부만 출력하기

```
exam %>%
  select(id, math) %>%  # id, math 추출
  head                  # 앞부분 6 행까지 추출
```

```
##   id math
## 1  1   50
## 2  2   60
## 3  3   45
## 4  4   30
## 5  5   25
## 6  6   50
```

dplyr Tip - 일부만 출력하기

```
exam %>%
  select(id, math) %>%  # id, math 추출
  head(10)              # 앞부분 10 행까지 추출
```

```
##    id math
## 1   1   50
## 2   2   60
## 3   3   45
## 4   4   30
## 5   5   25
## 6   6   50
## 7   7   80
## 8   8   90
## 9   9   20
## 10 10   50
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.class, cty 변수를 추출하여 새로운 데이터를 만들고, 데이터 일부를 출력해서 확인하세요.
- 2.class가 suv인 자동차와 compact인 자동차 중 어떤 자동차의 도시 연비가 더 높은지 알아보세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.class, cty 변수를 추출하여 새로운 데이터를 만들고, 데이터 일부를 출력해서 확인하세요.
- 2.class가 suv인 자동차와 compact인 자동차 중 어떤 자동차의 도시 연비가 더 높은지 알아보세요.

1번 정답

```
df <- mpg %>% select(class, cty)  # class, cty 변수 추출
head(df)                          # df 일부 출력
```

```
##     class cty
## 1 compact  18
## 2 compact  21
## 3 compact  20
## 4 compact  21
## 5 compact  16
## 6 compact  18
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.class, cty 변수를 추출하여 새로운 데이터를 만들고, 데이터 일부를 출력해서 확인하세요.
- 2.class가 suv인 자동차와 compact인 자동차 중 어떤 자동차의 도시 연비가 더 높은지 알아보세요.

2번 정답

```
df_suv <- df %>% filter(class == "suv")          # class 가 suv 인 행 추출
df_compact <- df %>% filter(class == "compact")  # class 가 compact 인 행 추출

mean(df_suv$cty)                                 # suv 의 cty 평균
## [1] 13.5
mean(df_compact$cty)                             # compact 의 cty 평균
## [1] 20.12766
```

3. 순서대로 정렬하기 - arrange()

오름차순으로 정렬하기

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

내림차순으로 정렬하기

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

정렬 기준 변수 여러개 지정

```
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
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- audi에서 생산한 자동차 중 고속도로 연비를 기준으로 1~5위에 해당하는 자동차의 데이터 출력하세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- audi에서 생산한 자동차 중 고속도로 연비를 기준으로 1~5위에 해당하는 자동차의 데이터 출력하세요.

```
mpg %>% filter(manufacturer == "audi") %>%  # audi 추출
  arrange(desc(hwy)) %>%                    # hwy 내림차순 정렬
  head(5)                                   # 5 행까지 출력
```

```
##   manufacturer      model displ year cyl      trans drv cty hwy fl   class
## 1         audi         a4   2.0 2008   4 manual(m6)   f  20  31  p compact
## 2         audi         a4   2.0 2008   4   auto(av)   f  21  30  p compact
## 3         audi         a4   1.8 1999   4   auto(l5)   f  18  29  p compact
## 4         audi         a4   1.8 1999   4 manual(m5)   f  21  29  p compact
## 5         audi a4 quattro   2.0 2008   4 manual(m6)   4  20  28  p compact
##   total test grade
## 1  25.5 pass     B
## 2  25.5 pass     B
## 3  23.5 pass     B
## 4  25.0 pass     B
## 5  24.0 pass     B
```

4. 파생변수 추가하기 - mutate()

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

```
##   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
```

여러 파생변수 동시에 추가

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

```
##   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
## 6  6     2   50      89      98   237 79.00000
```

조건문 활용하기

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

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

추가한 변수 활용하기

```
exam %>%
  mutate(total = math + english + science) %>%  # 총합 변수 추가
  arrange(total) %>%                            # 총합 변수 기준 정렬
  head                                          # 일부 추출
```

```
##   id class math english science total
## 1  9     3   20      98      15   133
## 2 14     4   48      87      12   147
## 3 12     3   45      85      32   162
## 4  5     2   25      80      65   170
## 5  4     1   30      98      58   186
## 6  8     2   90      78      25   193
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.mpg 데이터에 도시 연비와 고속도로 연비를 더한 합산 변수를 만들어 추가하세요.
- 2.앞에서 만든 합산 변수를 2로 나눠 평균 변수를 만드세요.
- 3.평균 변수를 기준으로 정렬하고, 가장 높은 자동차 3종의 데이터를 출력하세요.
- 4.위의 모든 과정은 하나의 dplyr 구문으로 만들어야 합니다.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.mpg 데이터에 도시 연비와 고속도로 연비를 더한 합산 변수를 만들어 추가하세요.
- 2.앞에서 만든 합산 변수를 2로 나눠 평균 변수를 만드세요.
- 3.평균 변수를 기준으로 정렬하고, 가장 높은 자동차 3종의 데이터를 출력하세요.
- 4.위의 모든 과정은 하나의 dplyr 구문으로 만들어야 합니다.

```
mpg %>%
  mutate(total = cty + hwy,      # 합산 변수 생성
         mean = total / 2) %>%  # 평균 변수 생성
  arrange(desc(mean)) %>%        # 평균 변수 내림차순 정렬
  head(3)                        # 상위 3 종 추출
```

```
##   manufacturer      model displ year cyl      trans drv cty hwy fl
## 1   volkswagen new beetle   1.9 1999   4 manual(m5)   f  35  44  d
## 2   volkswagen      jetta   1.9 1999   4 manual(m5)   f  33  44  d
## 3   volkswagen new beetle   1.9 1999   4   auto(l4)   f  29  41  d
##        class total test grade mean
## 1 subcompact    79 pass     A 39.5
```

```
## 2    compact    77 pass     A 38.5
## 3 subcompact    70 pass     A 35.0
```

5. 집단별로 요약하기 - group_by(), summarise()

요약통계량 구하기

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

##   mean_math
## 1     57.45
```

집단별 요약통계량 구하기

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

```
## # A tibble: 5 x 2
##   class mean_math
##   <int>     <dbl>
## 1     1     46.25
## 2     2     61.25
## 3     3     45.00
## 4     4     56.75
## 5     5     78.00
```

여러 요약통계량 한 번에 구하기

```
exam %>%
  group_by(class) %>%                    # class 별로 분리
  summarise(mean_math = mean(math),      # 수학점수 평균
            sum_math = sum(math),        # 수학점수 총합
            median_math = median(math), # 수학점수 중앙값
            n = n())                     # 수학점수 사례수
```

```
## # A tibble: 5 x 5
##   class mean_math sum_math median_math     n
##   <int>     <dbl>    <int>       <dbl> <int>
## 1     1     46.25      185        47.5     4
## 2     2     61.25      245        65.0     4
## 3     3     45.00      180        47.5     4
## 4     4     56.75      227        53.0     4
## 5     5     78.00      312        79.0     4
```

자주 사용하는 요약통계량 함수

함수	의미
mean()	평균
sd()	표준편차
sum()	총합
median()	중앙값
min()	최소값
max()	최대값
n()	사례수

각 집단별로 다시 집단 나누기

```
mpg %>%
  group_by(manufacturer, drv) %>%      # 회사별, 구방방식별 분리
  summarise(mean_cty = mean(cty)) %>%  # cty 평균 산출
  head(10)                             # 일부 출력
```

```
## Source: local data frame [10 x 3]
## Groups: manufacturer [5]
##
##    manufacturer   drv mean_cty
##           <chr> <chr>    <dbl>
## 1          audi     4 16.81818
## 2          audi     f 18.85714
## 3     chevrolet     4 12.50000
## 4     chevrolet     f 18.80000
## 5     chevrolet     r 14.10000
## 6         dodge     4 12.00000
## 7         dodge     f 15.81818
## 8          ford     4 13.30769
## 9          ford     r 14.75000
## 10        honda     f 24.44444
```

dplyr 조합하기

- 문제) 회사별로 suv 자동차의 도시 및 고속도로 통합 연비 평균을 구해 내림차순으로 정렬하고 1~5위까지 출력하라

분석 절차 생각해보기

절차	기능	dplyr 함수
1	회사별로 분리	group_by()
2	suv 추출	filter()
3	통합 연비 변수 생성	mutate()
4	통합 연비 평균 산출	summarise()
5	내림차순 정렬	arrange()
6	1~5위까지 출력	head()

dplyr 조합하기

- 문제) 회사별로 suv 자동차의 도시 및 고속도로 통합 연비 평균을 구해 내림차순으로 정렬하고 1~5위까지 출력하라

```
mpg %>%
  group_by(manufacturer) %>%            # 회사별로 분리
  filter(class == "suv") %>%            # suv 추출
  mutate(tot = (cty+hwy)/2) %>%         # 통합 연비 변수 생성
  summarise(mean_tot = mean(tot)) %>%   # 통합 연비 평균 산출
  arrange(desc(mean_tot)) %>%           # 내림차순 정렬
  head(5)                               # 1~5 위까지 출력
```

```
## # A tibble: 5 x 2
##   manufacturer mean_tot
##          <chr>    <dbl>
## 1       subaru 21.91667
## 2       toyota 16.31250
## 3       nissan 15.87500
## 4      mercury 15.62500
## 5         jeep 15.56250
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.회사별 평균 도시 연비(cty)를 내림차순으로 정렬하고, 도시 연비가 가장 높은 회사 세 곳을 출력하세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.회사별 평균 도시 연비(cty)를 내림차순으로 정렬하고, 도시 연비가 가장 높은 회사 세 곳을 출력하세요.

```
mpg %>%
  group_by(manufacturer) %>%            # 회사별 분리
  summarise(mean_cty = mean(cty)) %>%   # cty 평균 산출
  arrange(desc(mean_cty)) %>%           # 내림차순 정렬
  head(3)
```

```
## # A tibble: 3 x 2
##   manufacturer mean_cty
##          <chr>    <dbl>
## 1        honda 24.44444
## 2   volkswagen 20.92593
## 3       subaru 19.28571
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 2.각 회사에서 class가 경차(compact)인 자동차를 몇 종류 생산했는지, 차종 수가 내림차순으로 정렬된 표를 만들어보세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 2.각 회사에서 class가 경차(compact)인 자동차를 몇 종류 생산했는지, 차종 수가 내림차순으로 정렬된 표를 만들어보세요.

```
mpg %>%
  filter(class == "compact") %>%  # compact 추출
  group_by(manufacturer) %>%      # manufacturer 별 분리
  summarise(count = n()) %>%      # 사례수 산출
  arrange(desc(count))            # 내림차순 정렬
```

```
## # A tibble: 5 x 2
##   manufacturer count
##          <chr> <int>
## 1         audi    15
## 2   volkswagen    14
## 3       toyota    12
## 4       subaru     4
## 5       nissan     2
```

6. 데이터 합치기

가로로 합치기 - 열 추가: left_join()

세로로 합치기 - 행 추가: bind_rows()

가로로 합치기 - 열 추가: left_join()

데이터 생성

```r
# 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
```

```
##   id midterm
## 1  1      60
## 2  2      80
## 3  3      70
## 4  4      90
## 5  5      85
```

```
test2  # test2 출력
```

```
##   id final
## 1  1    70
## 2  2    83
## 3  3    65
## 4  4    95
## 5  5    80
```

id 기준으로 합치기

```
total <- left_join(test1, test2, by = "id")  # id 기준으로 합쳐서 total 에 할당
total                                        # exam 출력
```

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

[주의] by에 변수명을 지정할 때 변수명 앞 뒤에 겹따옴표 입력

응용 - 다른 데이터 활용해서 변수 추가하기

담임교사 명단 데이터 생성

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

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

class 기준 합치기

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

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

세로로 합치기 - 행 추가: bind_rows()

데이터 생성

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

# 학생 6~10 번 시험 데이터 생성
group_b <- data.frame(id = c(6, 7, 8, 9, 10),
                      test = c(70, 83, 65, 95, 80))
```

```r
group_a  # group_a 출력
```

```
##   id test
## 1  1   60
## 2  2   80
## 3  3   70
## 4  4   90
## 5  5   85
```

```r
group_b  # group_b 출력
```

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

세로로 합치기

```
group_all <- bind_rows(group_a, group_b)  # 데이터 합쳐서 group_all 에 할당
group_all                                 # group_all 출력
```

```
##    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
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

fl(fuel)	연료 종류	가격(갤런당 USD)
c	CNG	2.35
d	diesel	2.38
e	ethanol E85	2.11
p	premium	2.76
r	regular	2.22

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

fuel 데이터 프레임 생성

```
# fuel 데이터 생성
fuel <- data.frame(fl = c("c", "d", "e", "p", "r"),
                   price_fl = c(2.35, 2.38, 2.11, 2.76, 2.22),
                   stringsAsFactors = F)
fuel
```

```
##   fl price_fl
## 1  c     2.35
## 2  d     2.38
## 3  e     2.11
## 4  p     2.76
## 5  r     2.22
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- fuel 데이터를 이용해서 mpg 데이터에 '연료 가격' 변수를 추가하세요. 데이터에서 model, fl, 연료 가격 변수만 추출하고 앞부분 10행을 출력해서 변수가 잘 추가됐는지 확인해보세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- fuel 데이터를 이용해서 mpg 데이터에 '연료 가격' 변수를 추가하세요. 데이터에서 model, fl, 연료 가격 변수만 추출하고 앞부분 10행을 출력해서 변수가 잘 추가됐는지 확인해보세요.

```
mpg <- left_join(mpg, fuel, by = "fl")  # mpg 에 연료 가격 변수 추가

mpg %>%
  select(model, fl, price_fl) %>%       # model, fl, price_fl 추출
  head(10)                              # 앞부분 10 행 출력
```

```
##          model fl price_fl
## 1           a4  p     2.76
## 2           a4  p     2.76
## 3           a4  p     2.76
## 4           a4  p     2.76
## 5           a4  p     2.76
## 6           a4  p     2.76
## 7           a4  p     2.76
## 8  a4 quattro  p     2.76
## 9  a4 quattro  p     2.76
## 10 a4 quattro  p     2.76
```

[분석 도전]

ggplot2 패키지의 midwest 데이터를 새로 불러와서 데이터 분석 문제를 해결해보세요.

- 1.popadults는 해당 지역의 성인 인구, poptotal은 전체 인구를 나타냅니다. midwest 데이터에 '전체 인구 대비 미성년 인구 백분율' 변수를 추가하세요.

[분석 도전]

ggplot2 패키지의 midwest 데이터를 새로 불러와서 데이터 분석 문제를 해결해보세요.

- 1.popadults는 해당 지역의 성인 인구, poptotal은 전체 인구를 나타냅니다. midwest 데이터에 '전체 인구 대비 미성년 인구 백분율' 변수를 추가하세요.

```
# midwest 불러오기
midwest <- as.data.frame(ggplot2::midwest)

# midwest 에 백분율 변수 추가
midwest <- midwest %>%
  mutate(ratio_child = (poptotal-popadults)/poptotal*100)
```

[분석 도전]

ggplot2 패키지의 midwest 데이터를 새로 불러와서 데이터 분석 문제를 해결해보세요.

- 2.미성년 인구 백분율이 가장 높은 상위 5개 지역(county)의 미성년 인구 백분율을 출력하세요.

[분석 도전]

ggplot2 패키지의 midwest 데이터를 새로 불러와서 데이터 분석 문제를 해결해보세요.

- 2.미성년 인구 백분율이 가장 높은 상위 5개 지역(county)의 미성년 인구 백분율을 출력하세요.

```
midwest %>%
  arrange(desc(ratio_child)) %>%   # ratio_child 내림차순 정렬
  select(county, ratio_child) %>%  # county, ratio_child 추출
  head(5)                          # 상위 5 행 출력
```

```
##      county ratio_child
## 1  ISABELLA    51.50117
## 2 MENOMINEE    50.59126
## 3    ATHENS    49.32073
## 4   MECOSTA    49.05918
## 5    MONROE    47.35818
```

[분석 도전]

ggplot2 패키지의 midwest 데이터를 새로 불러와서 데이터 분석 문제를 해결해보세요.

- 3.아래 기준에 따라 미성년 비율 등급 변수를 추가하고, 각 등급에 몇 개의 지역이 있는지 알아보세요.

분류	기준
large	40% 이상
middle	30% ~ 40% 미만
small	30% 미만

[분석 도전]

ggplot2 패키지의 midwest 데이터를 새로 불러와서 데이터 분석 문제를 해결해보세요.

- 3.아래 기준에 따라 미성년 비율 등급 변수를 추가하고, 각 등급에 몇 개의 지역이 있는지 알아보세요.

분류	기준
large	40% 이상
middle	30% ~ 40% 미만
small	30% 미만

```
# midwest 에 grade 변수 추가
midwest <- midwest %>%
  mutate(grade = ifelse(ratio_child >= 40, "large",
                        ifelse(ratio_child >= 30, "middle", "small")))

# 미성년 비율 등급 빈도표
table(midwest$grade)
```

```
##
##  large middle  small
##     32    396      9
```

[분석 도전]

ggplot2 패키지의 midwest 데이터를 새로 불러와서 데이터 분석 문제를 해결해보세요.

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

[분석 도전]

ggplot2 패키지의 midwest 데이터를 새로 불러와서 데이터 분석 문제를 해결해보세요.

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

```
midwest %>%
  mutate(ratio_asian = (popasian/poptotal)*100) %>%  # 백분율 변수 추가
  arrange(ratio_asian) %>%                           # 내림차순 정렬
  select(state, county, ratio_asian) %>%             # 변수 추출
  head(10)                                           # 상위 10 행 출력
```

```
##    state    county ratio_asian
## 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
```

그래프 만들기

- 산점도
- 막대 그래프
- 선 그래프
- 상자그림

ggplot2 로드

```
library(ggplot2)
```

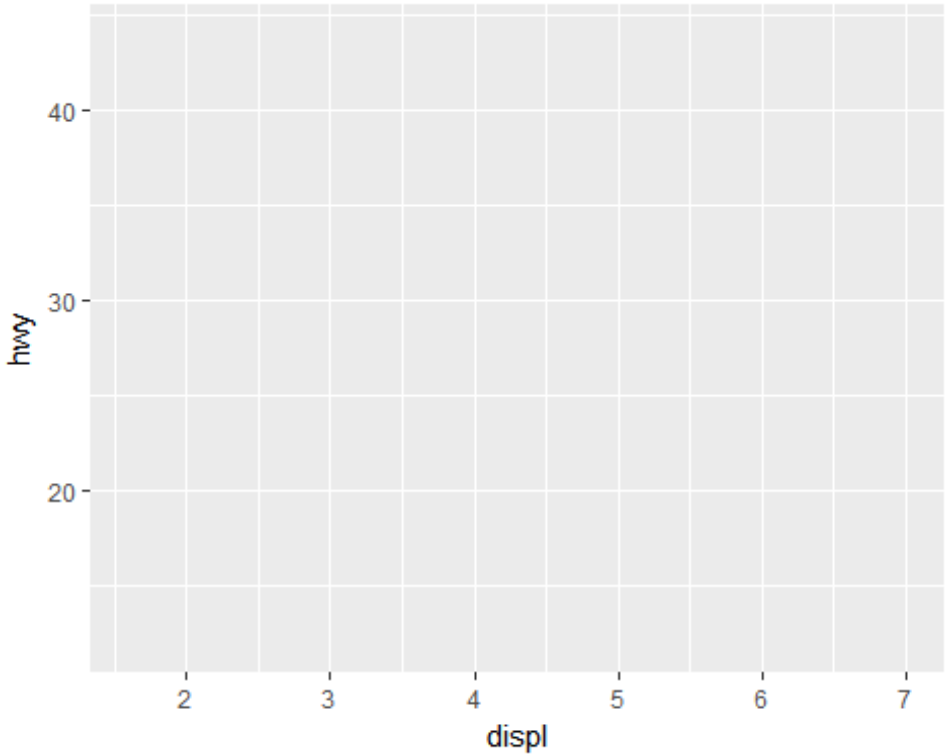
ggplot() 레이어 구조

- Step1. 배경 설정(축)
- Step2. 그래프 모양 추가(점, 막대, 선)
- Step3. 추가 옵션 설정 추가(범위, 색, 표식)

1.산점도

x축, y축 모두 연속변수

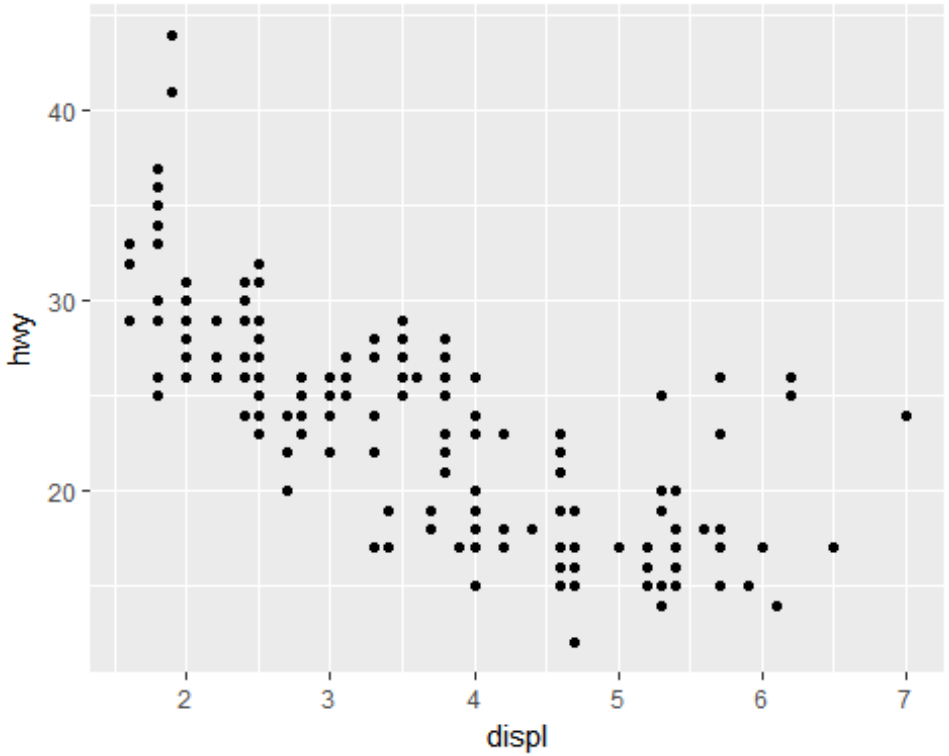
```
ggplot(data = mpg, aes(x = displ, y = hwy))
```



1.산점도

x축, y축 모두 연속변수

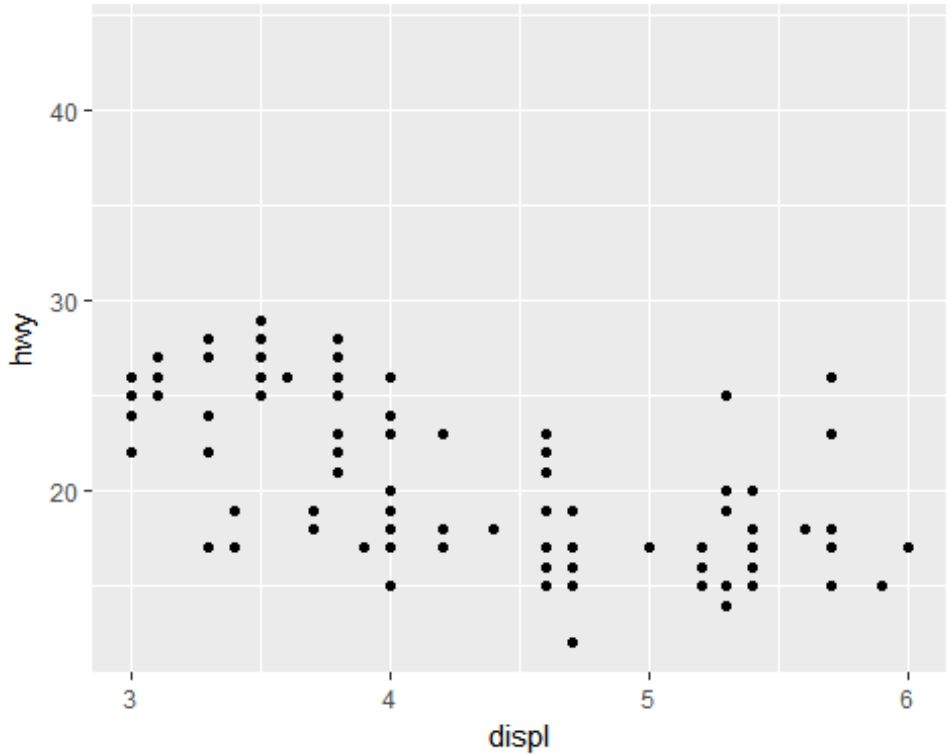
```
ggplot(data = mpg, aes(x = displ, y = hwy)) + geom_point()
```



1.산점도

축 범위 변경 - x축

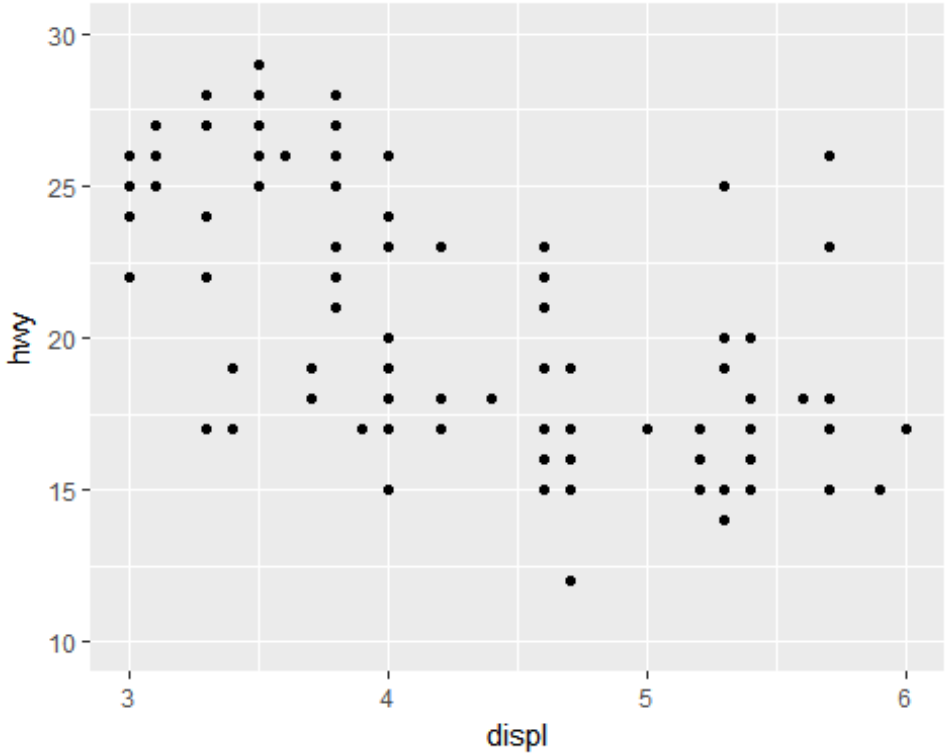
```
ggplot(data = mpg, aes(x = displ, y = hwy)) + geom_point() + xlim(3, 6)
```



1.산점도

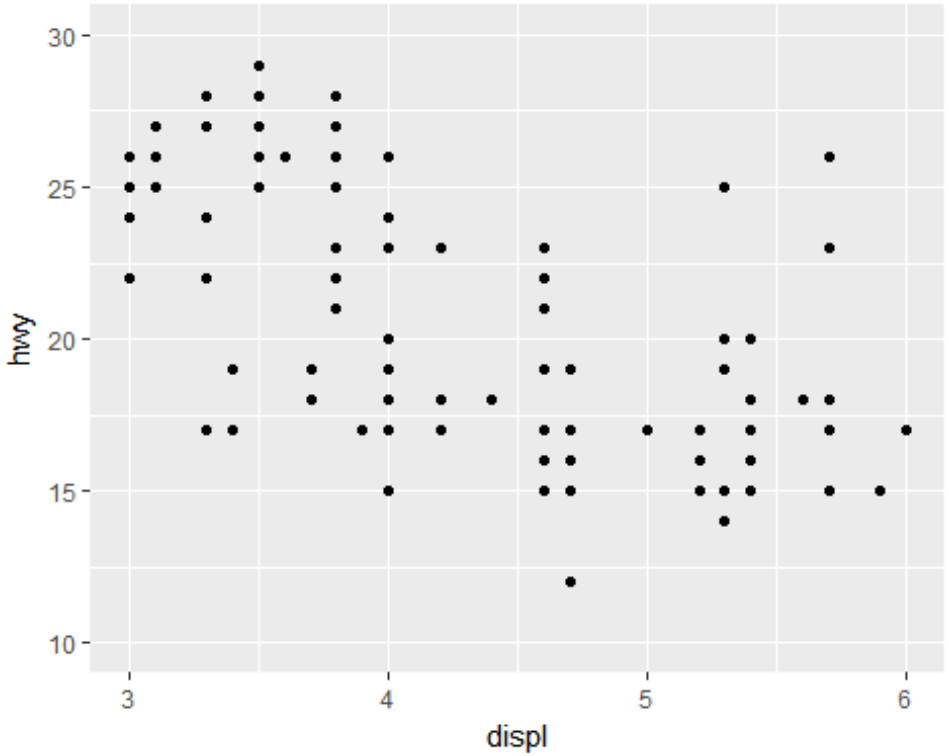
축 범위 변경 - x축, y축

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



ggplot2 코드 가독성 높이기

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



2.막대 그래프

(1)평균 막대 그래프

x축 범주변수, y축 연속변수

- Step1. 평균표 생성
- Step2. 그래프 생성

2.막대 그래프

(1)평균 막대 그래프

Step1. 집단별 평균표 만들기

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

df_mpg
```

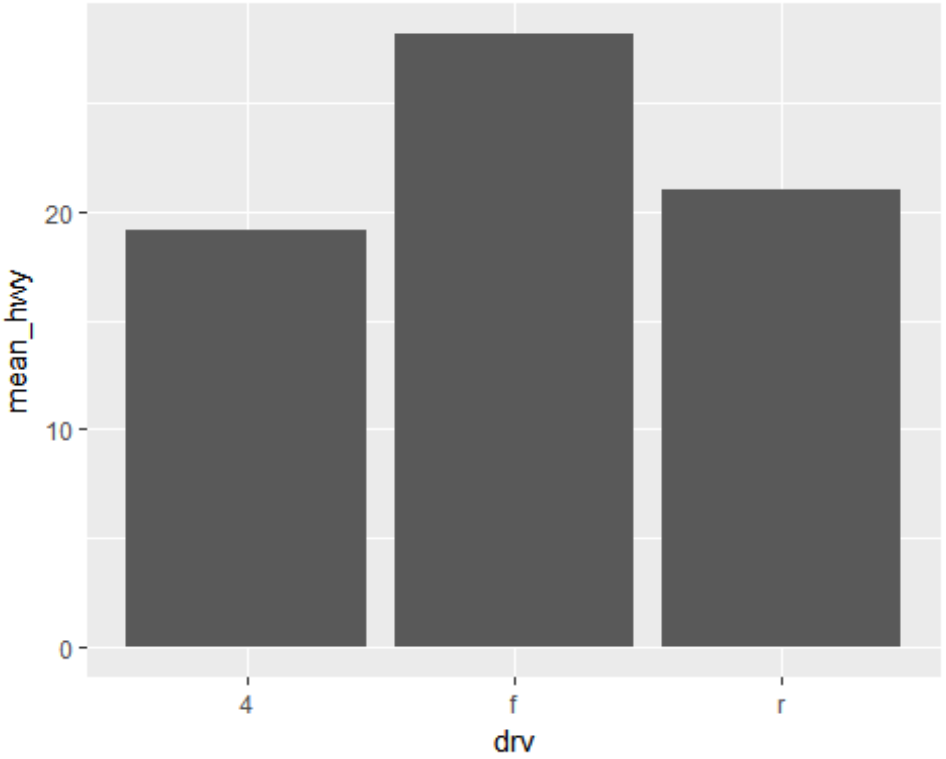
```
## # A tibble: 3 x 2
##     drv mean_hwy
##   <chr>    <dbl>
## 1     4 19.17476
## 2     f 28.16038
## 3     r 21.00000
```

2.막대 그래프

(1)평균 막대 그래프

Step2. 그래프 생성

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

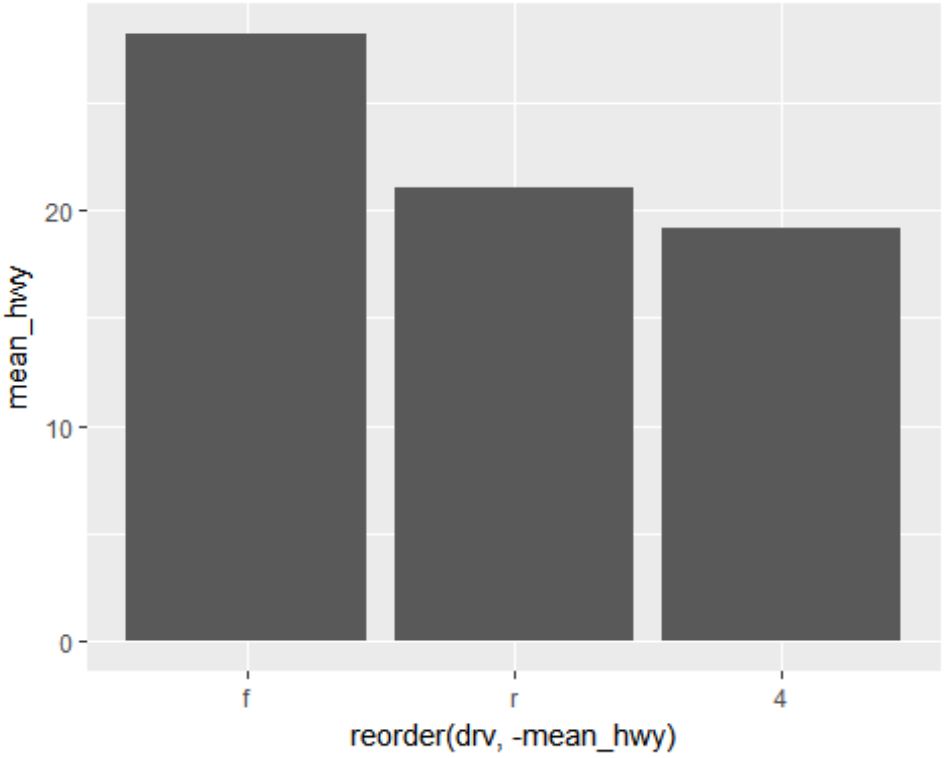


2.막대 그래프

(1)평균 막대 그래프

Step2. 그래프 생성 - 내림차순 정렬

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

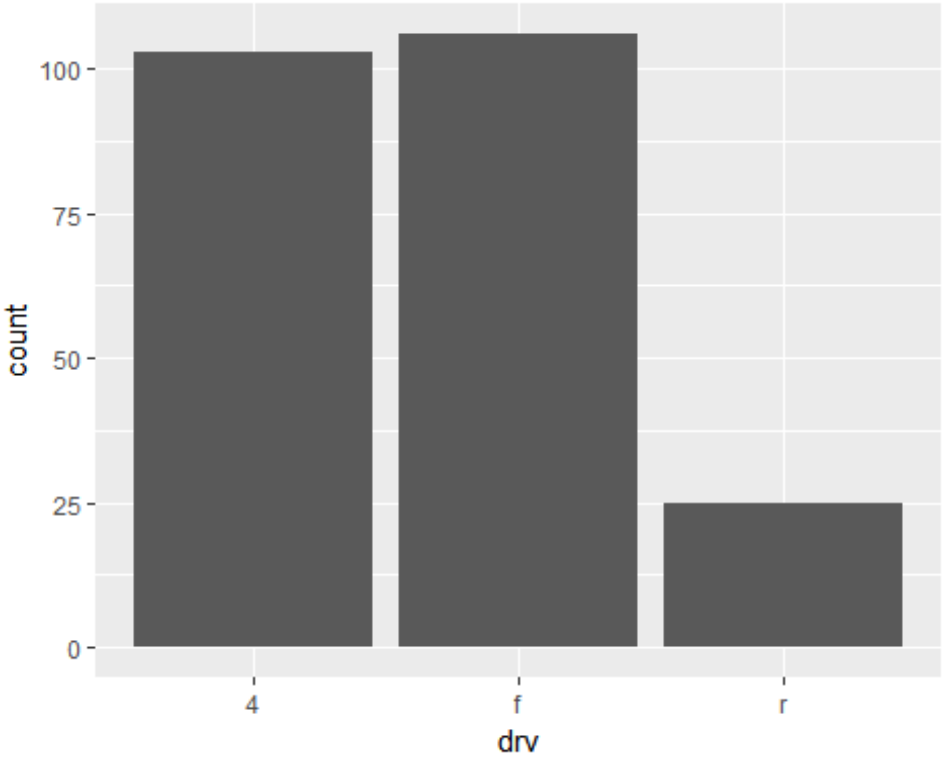


2.막대 그래프

(2)빈도 막대 그래프

x축 범주변수, y축 빈도

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

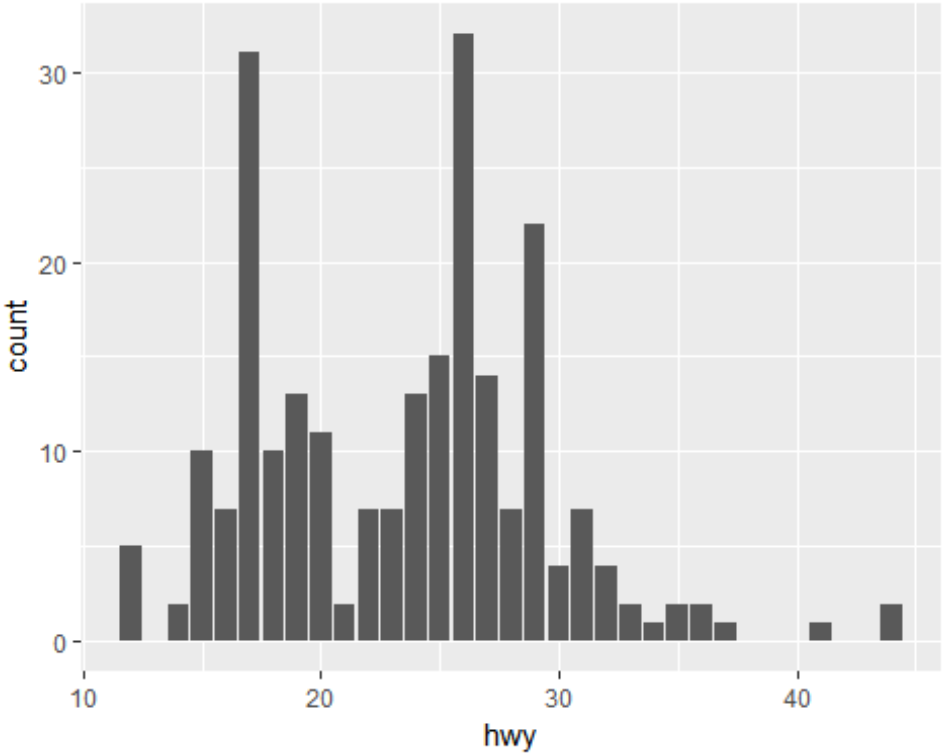


2.막대 그래프

(2)빈도 막대 그래프

x축 연속변수, y축 빈도

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

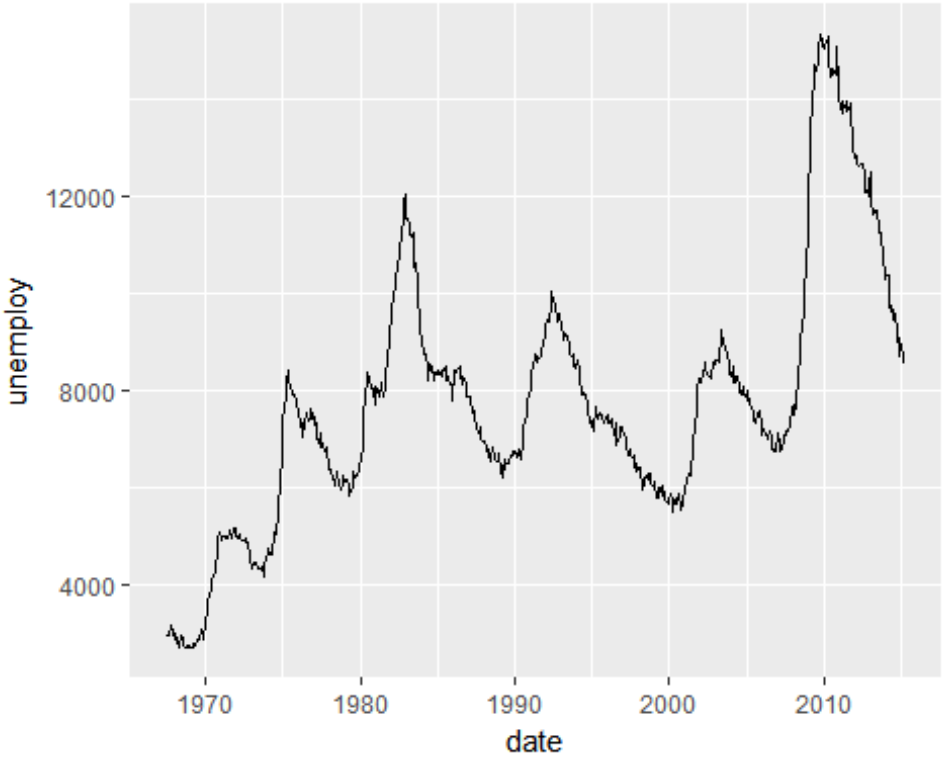


3.선 그래프

시계열 그래프

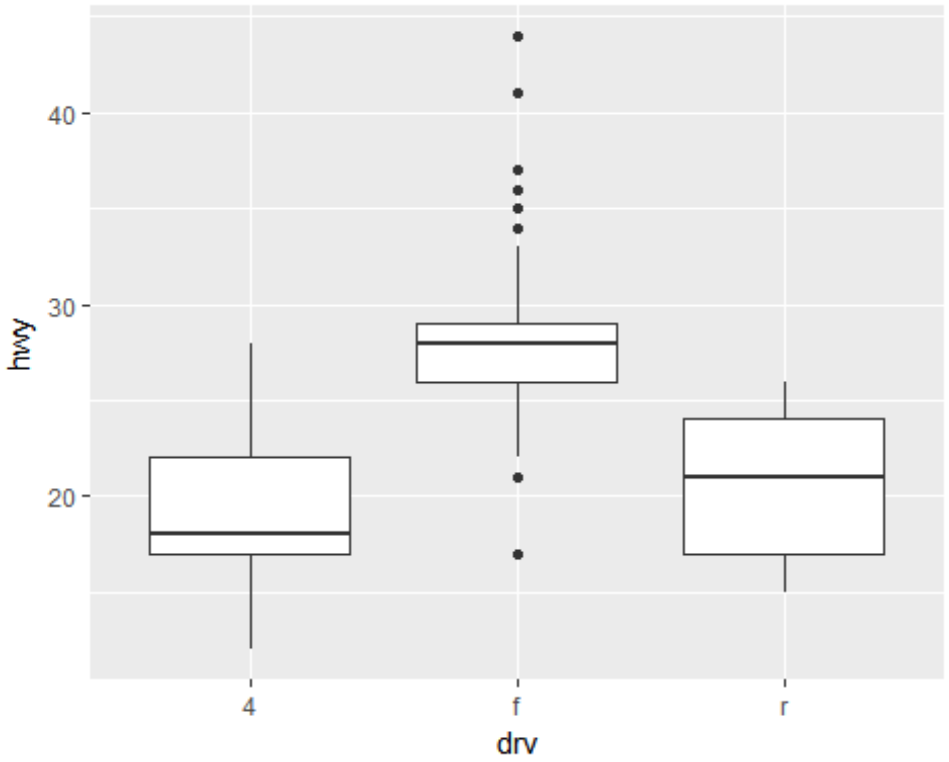
x축(시간), y축 모두 연속변수

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



4.상자그림

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



4.상자그림

상자 그림	값	설명
상자 아래 세로선	아래 수염	하위 0~25% 내에 해당하는 값(극단치 제외)
상자 밑면	1사분위수(Q1)	하위 25% 위치 값
상자 내 굵은 선	2사분위수(Q2)	하위 50% 위치 값(중앙값)
상자 윗면	3사분위수(Q3)	하위 75% 위치 값
상자 위 세로선	윗수염	하위 75~100% 내에 해당하는 값(극단치 제외)
상자 밖 점 표식	극단치	Q1, Q3 밖 1.5 IQR을 벗어난 값

[참고] 1.5 IQR: 사분위 범위(Q1~Q3간 거리)의 1.5배

1분 퀴즈

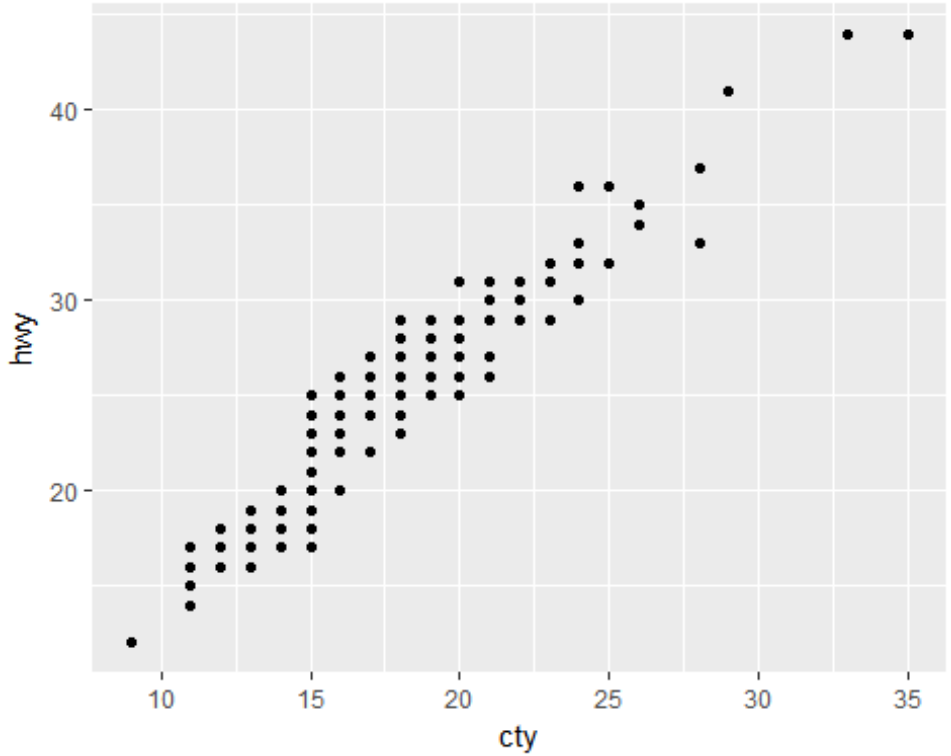
- mpg 데이터를 사용해서 x축은 도시 연비, y축은 고속도로 연비로 된 산점도를 만드세요.

1분 퀴즈

- mpg 데이터를 사용해서 x축은 도시 연비, y축은 고속도로 연비로 된 산점도를 만드세요.

정답

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



1분 퀴즈

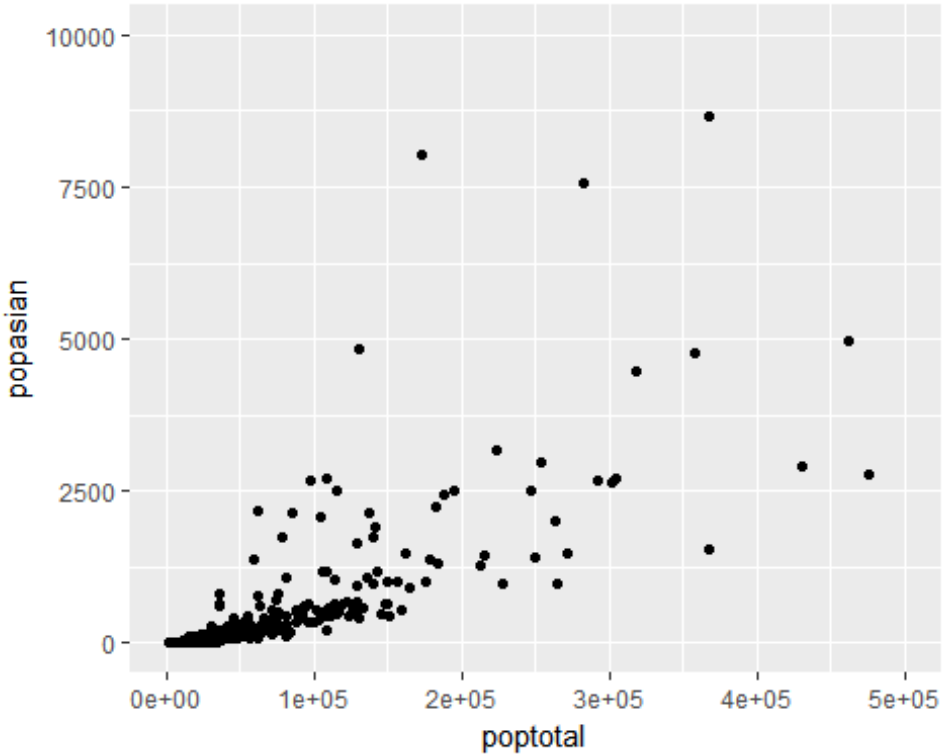
- midwest 데이터를 이용해서 x축은 전체 인구, y축은 아시아인 인구로 된 산점도를 만드세요. 전체 인구 50만명 이하, 아시아인 인구 1만명 이하의 지역만 표시되게 설정하세요.

1분 퀴즈

- midwest 데이터를 이용해서 x축은 전체 인구, y축은 아시아인 인구로 된 산점도를 만드세요. 전체 인구 50만명 이하, 아시아인 인구 1만명 이하의 지역만 표시되게 설정하세요.

정답

```
ggplot(data = midwest, aes(x = poptotal, y = popasian)) +
  geom_point() +
  xlim(0, 500000) +
  ylim(0, 10000)
```



[참고] options(scipen = 99)

1분 퀴즈

- mpg 데이터를 이용해서 suv 차종의 평균 도시 연비가 가장 높은 회사 다섯 곳의 평균 도시 연비 막대 그래프를 만드세요. 연비가 높은 순으로 정렬하세요.

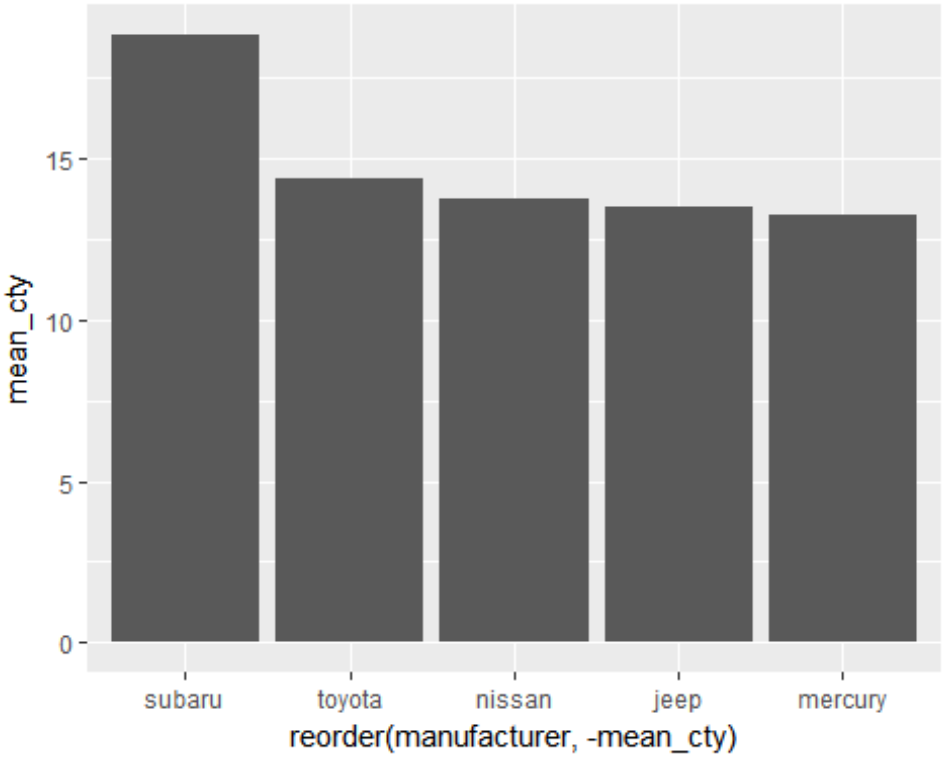
1분 퀴즈

- mpg 데이터를 이용해서 suv 차종의 평균 도시 연비가 가장 높은 회사 다섯 곳의 평균 도시 연비 막대 그래프를 만드세요. 연비가 높은 순으로 정렬하세요.

정답

```
# 평균 표 생성
df <- mpg %>%
  filter(class == "suv") %>%
  group_by(manufacturer) %>%
  summarise(mean_cty = mean(cty)) %>%
  arrange(desc(mean_cty)) %>%
  head(5)

# 그래프 생성
ggplot(data = df, aes(x = reorder(manufacturer, -mean_cty), y = mean_cty)) + geom_col()
```


15
10
5
0
mean_cty
subaru
toyota
nissan
jeep
mercury
reorder(manufacturer, -mean_cty)

1분 퀴즈

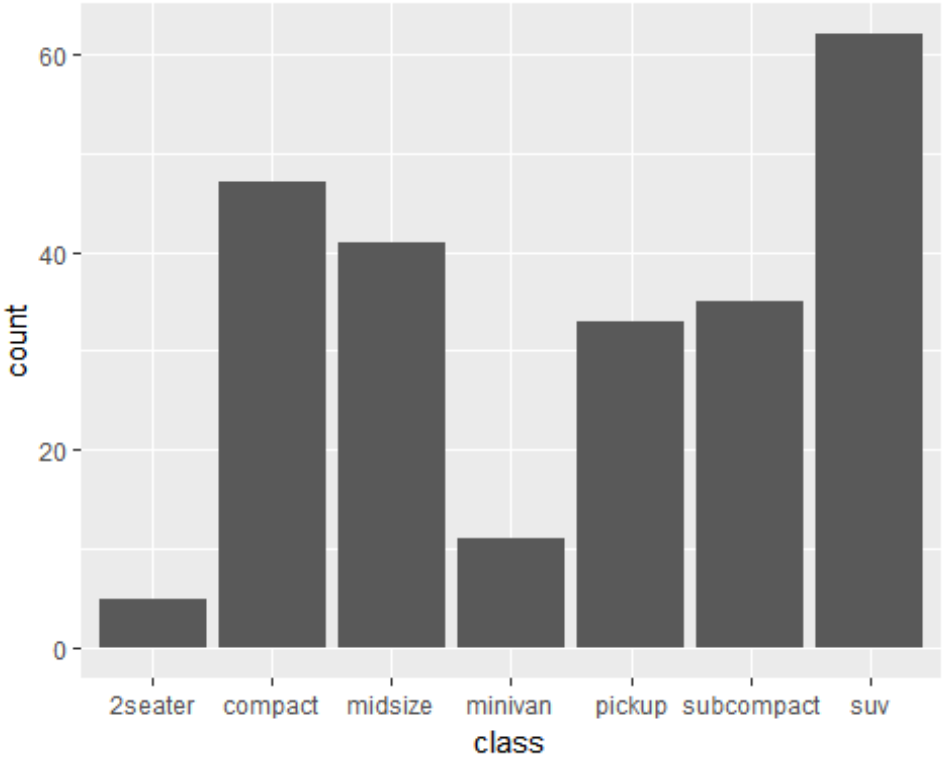
- mpg 데이터를 이용해서 자동차 종류별 빈도 막대 그래프를 만드세요.

1분 퀴즈

- 자동차 종류별 빈도 막대 그래프를 만드세요.

정답

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



1분 퀴즈

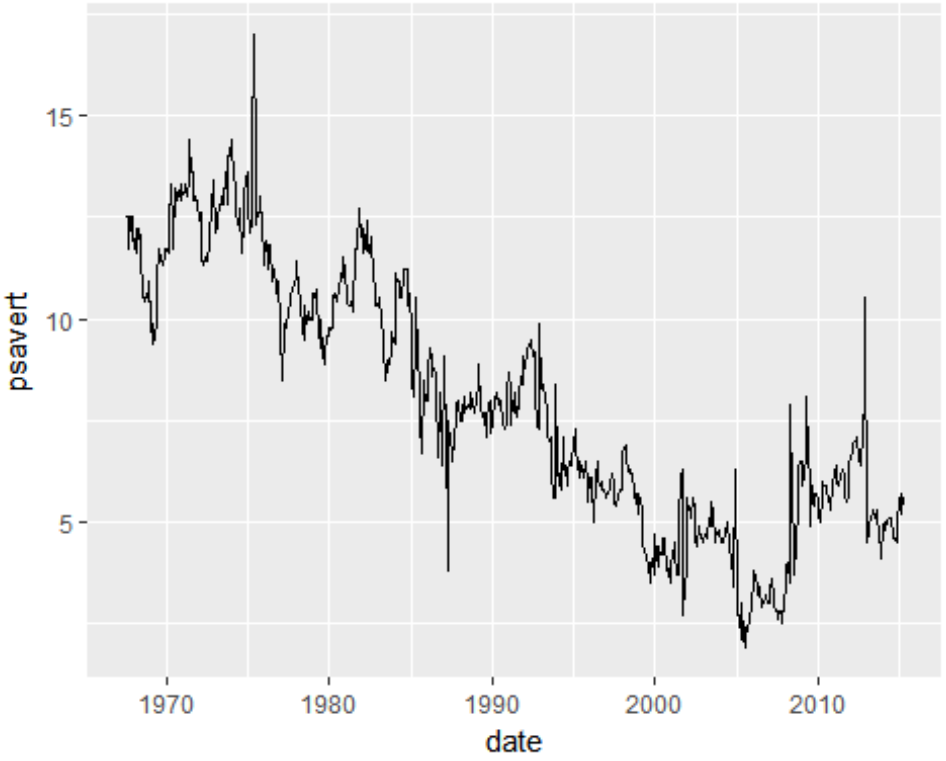
- economics 데이터를 이용해서 개인 저축률(psavert) 변수가 시간에 따라 어떻게 변해왔는지 나타낸 시계열 그래프를 만드세요.

1분 퀴즈

- economics 데이터를 이용해서 개인 저축률(psavert) 변수가 시간에 따라 어떻게 변해왔는지 나타낸 시계열 그래프를 만드세요.

정답

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



1분 퀴즈

- mpg 데이터를 이용해서 자동차 종류(class)가 compact, subcompact, suv인 자동차의 도시 연비(cty)가 어떻게 다른지 나타낸 상자 그림을 만드세요.

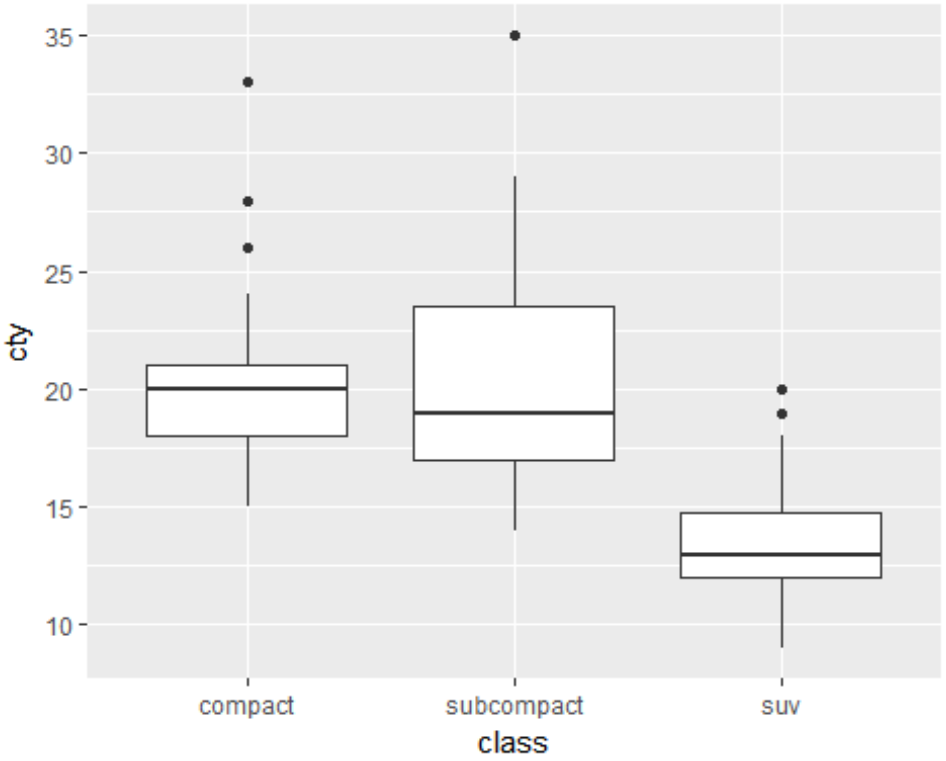
1분 퀴즈

- mpg 데이터를 이용해서 자동차 종류(class)가 compact, subcompact, suv인 자동차의 도시 연비(cty)가 어떻게 다른지 나타낸 상자 그림을 만드세요.

정답

```
class_mpg <- mpg %>%
  filter(class %in% c("compact", "subcompact", "suv"))

ggplot(data = class_mpg, aes(x = class, y = cty)) + geom_boxplot()
```


35
30
25
20
15
10
cty
compact
subcompact
suv
class

데이터 정제하기

1.빠진 데이터를 찾아라! - 결측치 정제하기

결측치(Missing Value) - 누락된 값, 비어있는 값

- 함수 적용 불가, 분석 결과 왜곡
- 제거 후 분석 실시

결측치 만들기

결측치 표기 - 대문자 NA

```
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
```

[유의] NA 앞 뒤에 겹따옴표 없음

결측치 확인하기

```
is.na(df)         # 결측치 확인

##        sex score
## [1,] FALSE FALSE
## [2,] FALSE FALSE
## [3,]  TRUE FALSE
## [4,] FALSE FALSE
## [5,] FALSE  TRUE

table(is.na(df))  # 결측치 빈도 출력

##
## FALSE  TRUE
##     8     2
```

변수별로 결측치 확인하기

```
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
```

결측치 포함된 상태로 분석

```
mean(df$score)  # 평균 산출
```

```
## [1] NA
```

```
sum(df$score)   # 합계 산출
```

```
## [1] NA
```

결측치 제거하기

결측치 있는 행 제거하기

```
df %>% filter(is.na(score))   # score 가 NA 인 데이터만 출력

##   sex score
## 1   F    NA

df %>% filter(!is.na(score))  # score 가 NA 아닌 데이터만 출력

##    sex score
## 1    M     5
## 2    F     4
## 3 <NA>     3
## 4    M     4
```

결측치 제외한 데이터로 분석하기

```
df_nomiss <- df %>% filter(!is.na(score))  # 결측치 제외된 데이터 생성
df_nomiss
```

```
##    sex score
## 1    M     5
## 2    F     4
## 3 <NA>     3
## 4    M     4
```

```
mean(df_nomiss$score)                      # score 평균 산출
```

```
## [1] 4
```

```
sum(df_nomiss$score)                       # score 합계 산출
```

```
## [1] 16
```

여러 변수 결측치 동시에 제거하기

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

```
##   sex score
## 1   M     5
## 2   F     4
## 3   M     4
```

결측치가 하나라도 있으면 제거하기

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

```
##   sex score
## 1   M     5
## 2   F     4
## 4   M     4
```

- 분석에 필요한 데이터까지 손실 될 가능성 유의
- ex) 성별-소득 관계 분석하는데 지역 결측치까지 제거

함수의 결측치 제외 기능 이용하기 - 'na.rm = T'

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

```
## [1] 4
```

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

```
## [1] 16
```

summarise()에서 na.rm = T 사용하기

준비하기

```
exam <- read.csv("csv_exam.csv")            # 데이터 불러오기
exam[c(3, 8, 15), "math"] <- NA             # 3, 8, 15 행의 math 에 NA 할당
```

na.rm = T 적용

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

##   mean_math
## 1        NA

exam %>% summarise(mean_math = mean(math, na.rm = T))  # 결측치 제외하고 평균 산출

##   mean_math
## 1  55.23529
```

다른 함수들에 적용

```
exam %>% summarise(mean_math = mean(math, na.rm = T),       # 평균 산출
                   sum_math = sum(math, na.rm = T),         # 총합 산출
                   median_math = median(math, na.rm = T))  # 중앙값 산출
```

```
##   mean_math sum_math median_math
## 1  55.23529      939          50
```

결측치 대체하기

- 결측치 많을 경우 모두 제외하면 데이터 손실 큼
- 대안: 다른 값 채워넣기

결측치 대체법(Imputation)

- 대표값(평균, 최빈값 등)으로 일괄 대체
- 통계분석기법 적용, 예측값 추정해서 대체

평균값으로 결측치 대체하기

평균 구하기

```
mean(exam$math, na.rm = T) # 결측치 제외하고 math 평균 산출
```

```
## [1] 55.23529
```

평균으로 대체하기

```
exam$math <- ifelse(is.na(exam$math), 55, exam$math)  # math 가 NA 면 55 로 대체
table(is.na(exam$math))                               # 결측치 빈도표 생성
```

```
##
## FALSE
##    20
```

```
mean(exam$math)                                       # math 평균 산출
```

```
## [1] 55.2
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 먼저 mpg 데이터를 불러와서 몇 개의 값을 결측치로 만들겠습니다.

```
mpg <- as.data.frame(ggplot2::mpg)        # mpg 데이터 가져오기
mpg[c(65,124,131,153,212), "hwy"] <- NA   # NA 할당하기
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

1. hwy에 결측치가 몇 개 있는지 알아보세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

1. hwy에 결측치가 몇 개 있는지 알아보세요.

```
table(is.na(mpg$hwy))  # hwy 결측치 빈도표 생성
```

```
##
## FALSE  TRUE
##   229     5
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

2. hwy 결측치를 제외하고, 구동방식(drv)별 hwy 평균을 구하세요. 하나의 dplyr 구문으로 만들어야 합니다.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

2. hwy 결측치를 제외하고, 구동방식(drv)별 hwy 평균을 구하세요. 하나의 dplyr 구문으로 만들어야 합니다.

```
mpg %>%
  filter(!is.na(hwy)) %>%
  group_by(drv) %>%
  summarise(mean_hwy = mean(hwy))
```

```
## # A tibble: 3 x 2
##     drv mean_hwy
##   <chr>    <dbl>
## 1     4 19.24242
## 2     f 28.20000
## 3     r 21.00000
```

2.이상한 데이터를 찾아라! - 이상치 정제하기

이상치(Outlier) - 정상범주에서 크게 벗어난 값

- 이상치 포함시 분석 결과 왜곡
- 결측 처리 후 제외하고 분석

이상치 종류	예	해결 방법
존재할 수 없는 값	성별 변수에 3	결측 처리
극단적인 값	몸무게 변수에 200	정상범위 기준 정해서 결측 처리

이상치 제거하기 - (1)존재할 수 없는 값

- 논리적으로 존재할 수 없으므로 바로 결측 처리

이상치 포함된 데이터 생성 - sex 3, score 6

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

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

이상치 확인하기

```
table(outlier$sex)
```

```
##
## 1 2 3
## 3 2 1
```

```
table(outlier$score)
```

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

결측 처리하기 - sex

```r
# sex 가 3 이면 NA 할당
outlier$sex <- ifelse(outlier$sex == 3, NA, outlier$sex)
outlier
```

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

결측 처리하기 - score

```
# sex 가 1~5 아니면 NA 할당
outlier$score <- ifelse(outlier$score > 5, NA, outlier$score)
outlier
```

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

결측치 제외하고 분석

```
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          4
## 2     2          3
```

이상치 제거하기 - (2)극단적인 값

- 정상범위 기준 정해서 벗어나면 결측 처리

판단 기준	예
논리적 판단	성인 몸무게 40kg~150kg 벗어나면 극단치
통계적 판단	상하위 0.3% 극단치 또는 상자그림 1.5 IQR 벗어나면 극단치

상자그림으로 극단치 기준 정하기

상자그림 생성

```
mpg <- as.data.frame(ggplot2::mpg)
boxplot(mpg$hwy)
```

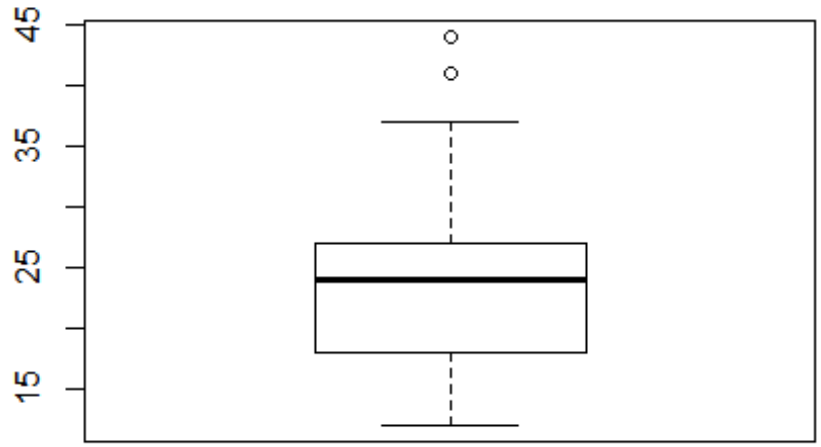


그림	값	의미
직사각형 밑면	1사분위수(Q1)	하위 25% 지점에 위치하는 값
직사각형내 가로선	2분위수(Q2, 중앙값)	하위 50% 지점에 위치하는 값
직사각형 윗면	3사분위수(Q3)	하위 75% 지점에 위치하는 값
직사각형 밖 가로선	1.5 IQR 내 최대값	Q1, Q3 밖 1.5 IQR 내에서 가장 큰 값
점 표식	극단치	1.5 IQR을 벗어난 값

상자그림 통계치 출력

```
boxplot(mpg$hwy)$stats  # 상자그림 통계치 출력
```

```
##      [,1]
## [1,]   12
## [2,]   18
## [3,]   24
## [4,]   27
## [5,]   37
## attr(,"class")
##         1
## "integer"
```

결측 처리하기

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

```
##
## FALSE  TRUE
##   231     3
```

결측치 제외하고 분석하기

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

```
## # A tibble: 3 x 2
##     drv mean_hwy
##   <chr>    <dbl>
## 1     4 19.17476
## 2     f 27.72816
## 3     r 21.00000
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 자동차 구동방식(drv)은 사륜구동(4), 전륜구동(f), 후륜구동(r)으로 분류됩니다.
- mpg 데이터를 불러와서 몇개 행의 drv 변수에 존재할 수 없는 값을 할당하겠습니다.
- 몇개 행의 cty 변수에 극단적으로 크거나 작은 값을 할당하겠습니다.

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

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.drv 변수에 이상치가 있는지 확인하세요. `%in%` 기호를 활용해서 drv 변수의 이상치를 결측 처리한 후 이상치가 사라졌는지 확인하세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 1.drv 변수에 이상치가 있는지 확인하세요. `%in%` 기호를 활용해서 drv 변수의 이상치를 결측 처리한 후 이상치가 사라졌는지 확인하세요.

```
# 이상치 확인
table(mpg$drv)
```

```
##
##   4   f   k   r
## 100 106   4  24
```

```
# drv 가 f, r, 4 인 경우 기존 값 할당, 그 외 NA 할당
mpg$drv <- ifelse(mpg$drv %in% c("f", "r", "4"), mpg$drv, NA)

# 이상치 확인
table(mpg$drv)
```

```
##
##   4   f   r
## 100 106  24
```

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

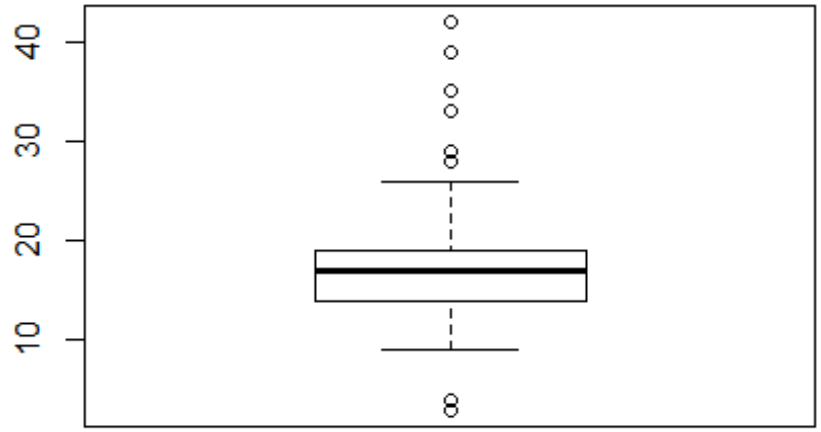
- 2.상자그림을 생성해서 cty 변수에 이상치가 있는지 확인하세요. 상자그림의 통계치를 이용해서 정상범위를 벗어난 값을 결측 처리한 후 다시 상자그림을 생성해서 이상치가 제외됐는지 확인해세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 2.상자그림을 생성해서 cty 변수에 이상치가 있는지 확인하세요. 상자그림의 통계치를 이용해서 정상범위를 벗어난 값을 결측 처리한 후 다시 상자그림을 생성해서 이상치가 제외됐는지 확인해세요.

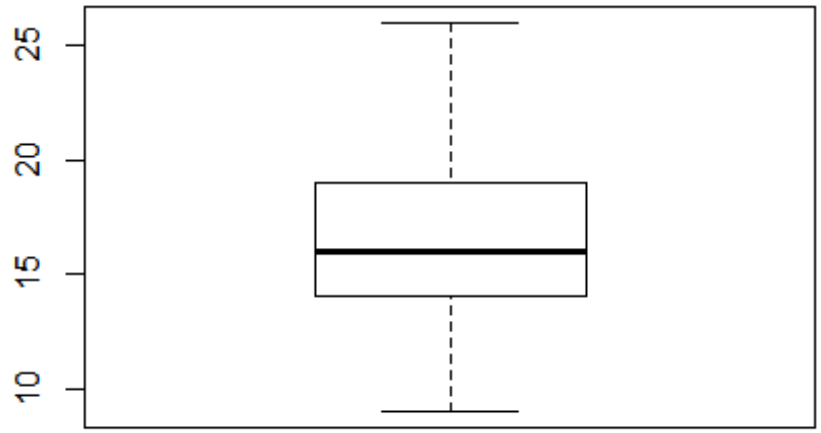
```
# 상자그림 생성 및 통계치 산출
boxplot(mpg$cty)$stats
```




```
##      [,1]
## [1,]    9
## [2,]   14
## [3,]   17
## [4,]   19
## [5,]   26
```

```
# 9~26 벗어나면 NA  할당
mpg$cty <- ifelse(mpg$cty < 9 | mpg$cty > 26, NA, mpg$cty)
```

```r
# 상자그림 생성
boxplot(mpg$cty)
```



1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 3.이상치가 제외된 데이터로 구동방식별 cty 평균을 구하세요.

1분 퀴즈

mpg 데이터를 이용해서 분석 문제를 해결해보세요.

- 3.이상치가 제외된 데이터로 구동방식별 cty 평균을 구하세요.

```
# 구동방식별 cty 평균
mpg %>%
  filter(!is.na(drv) & !is.na(cty)) %>%
  group_by(drv) %>%
  summarise(mean_hwy = mean(cty))
```

```
## # A tibble: 3 x 2
##     drv mean_hwy
##   <chr>    <dbl>
## 1     4 14.24742
## 2     f 19.47000
## 3     r 13.95833
```

R 기본 문법으로 데이터 추출하기

```
exam[,]  # 조건 없이 전체 데이터 출력
```

```
##    id class math english science
## 1   1     1   50      98      50
## 2   2     1   60      97      60
## 3   3     1   55      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   55      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   55      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
```

행 번호로 행 추출하기

```
exam[1,]  # 1 행 추출
```

```
##   id class math english science
## 1  1     1   50      98      50
```

```
exam[2,]  # 2 행 추출
```

```
##   id class math english science
## 2  2     1   60      97      60
```

조건을 충족하는 행 추출하기

```
exam[exam$class == 1,]  # class 가 1 인 행 추출
```

```
##   id class math english science
## 1  1     1   50      98      50
## 2  2     1   60      97      60
## 3  3     1   55      86      78
## 4  4     1   30      98      58
```

```
exam[exam$math >= 80,]  # 수학점수가 80 점 이상인 행 추출
```

```
##    id class math english science
## 7   7     2   80      90      45
## 18 18     5   80      78      90
## 19 19     5   89      68      87
```

여러 조건 동시 충족

```
# 1 반 이면서 수학점수가 50 점 이상
exam[exam$class == 1 & exam$math >= 50,]
```

```
##   id class math english science
## 1  1     1   50      98      50
## 2  2     1   60      97      60
## 3  3     1   55      86      78
```

```
# 영어점수가 90 점 미만이거나 과학점수가 50 점 미만
exam[exam$english < 90 | exam$science < 50,]
```

```
##    id class math english science
## 3   3     1   55      86      78
## 5   5     2   25      80      65
## 6   6     2   50      89      98
## 7   7     2   80      90      45
## 8   8     2   55      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
## 14 14     4   48      87      12
## 15 15     4   55      56      78
## 17 17     5   65      68      98
```

```
## 18 18     5   80      78      90
## 19 19     5   89      68      87
## 20 20     5   78      83      58
```

열 번호로 변수 추출하기

```
exam[,1]  # 첫 번째 열 추출
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```
exam[,2]  # 두 번째 열 추출
```

```
##  [1] 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5
```

```
exam[,3]  # 세 번째 열 추출
```

```
##  [1] 50 60 55 30 25 50 80 55 20 50 65 45 46 48 55 58 65 80 89 78
```

변수명으로 변수 추출하기

```
exam[, "class"]  # class 변수 추출
```

```
##  [1] 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5
```

```
exam[, "math"]   # math 변수 추출
```

```
##  [1] 50 60 55 30 25 50 80 55 20 50 65 45 46 48 55 58 65 80 89 78
```

```
exam[,c("class", "math", "english")]  # class, math, english 변수 추출
```

```
##    class math english
## 1      1   50      98
## 2      1   60      97
## 3      1   55      86
## 4      1   30      98
## 5      2   25      80
## 6      2   50      89
## 7      2   80      90
## 8      2   55      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    55       56
## 16      4    58       98
## 17      5    65       68
## 18      5    80       78
## 19      5    89       68
## 20      5    78       83
```

[유의] 변수명 양쪽에 겹따옴표(") 입력

행, 변수 동시 추출하기

```
# 행, 변수 모두 인덱스
exam[1,3]
```

```
## [1] 50
```

```
# 행은 인덱스, 열은 변수명
exam[5, "english"]
```

```
## [1] 80
```

```
# 행은 부등호 조건, 열은 변수명
exam[exam$math >= 50, "english"]
```

```
##  [1] 98 97 86 89 90 78 98 65 56 98 68 78 68 83
```

```
# 행은 부등호 조건, 열은 변수명
exam[exam$math >= 50, c("english", "science")]
```

```
##    english science
## 1       98      50
## 2       97      60
## 3       86      78
## 6       89      98
## 7       90      45
## 8       78      25
```

```
## 10       98       45
## 11       65       65
## 15       56       78
## 16       98       65
## 17       68       98
## 18       78       90
## 19       68       87
## 20       83       58
```

dplyr과 R 기본 문법 비교

문제) 수학점수 50 이상, 영어점수 80 이상인 학생들을 대상으로 각 반의 전과목 총평균을 구하라

기본 문법

```
exam$tot <- (exam$math + exam$english + exam$science)/3
aggregate(data=exam[exam$math >= 50 & exam$english >= 80,], tot~class, mean)
```

dplyr

```
exam %>%
  filter(math >= 50 & english >= 80) %>%
  mutate(tot = (math + english + science)/3) %>%
  group_by(class) %>%
  summarise(mean = mean(tot))
```

변수 타입(type) 이해하기

변수타입마다 가능한 분석 방법, 적용 가능한 기능 다름

numeric

- 연속변수(Continuous variable) 또는 양적변수(Quantitative variable)
- 숫자 크기가 강도를 의미 – 산술 가능
- ex) 키, 몸무게, 소득

factor

- 범주변수(Categorical variable) 또는 명목변수(Nominal variable)
- 숫자가 분류 의미 – 산술 불가
- ex) 성별(1 남자, 2 여자), 지역(1 서울, 2 대전, 3 대구, 4 부산)

변수 타입 간 차이

```
var1 <- c(1,2,3,1,2)           # numeric 변수 생성
var2 <- factor(c(1,2,3,1,2))   # factor 변수 생성


var1  # numeric 변수 출력
```

```
## [1] 1 2 3 1 2
```

```
var2  # factor 변수 출력
```

```
## [1] 1 2 3 1 2
## Levels: 1 2 3
```

factor 변수는 연산이 안된다

```
var1+2  # numeric 변수로 연산
```

```
## [1] 3 4 5 3 4
```

```
var2+2  # factor 변수로 연산
```

```
## Warning in Ops.factor(var2, 2): '+' not meaningful for factors
```

```
## [1] NA NA NA NA NA
```

변수 타입 확인하기

```
class(var1)
```

```
## [1] "numeric"
```

```
class(var2)
```

```
## [1] "factor"
```

factor 변수의 구성 범주 확인하기

```
levels(var1)
```

```
## NULL
```

```
levels(var2)
```

```
## [1] "1" "2" "3"
```

문자로 구성된 factor 변수

```r
var3 <- c("a", "b", "b", "c")          # 문자 변수 생성
var4 <- factor(c("a", "b", "b", "c")) # 문자로 된 factor 변수 생성
```

```
# 출력 결과 비교
var3
```

```
## [1] "a" "b" "b" "c"
```

```
var4
```

```
## [1] a b b c
## Levels: a b c
```

```r
# 타입 확인
class(var3)
```

```
## [1] "character"
```

```r
class(var4)
```

```
## [1] "factor"
```

함수마다 적용 가능한 변수 타입이 다르다

```
mean(var1)
```

```
## [1] 1.8
```

```
mean(var2)
```

```
## Warning in mean.default(var2): argument is not numeric or logical:
## returning NA
```

```
## [1] NA
```

변수 타입 바꾸기

```
var2 <- as.numeric(var2)  # numeric 타입으로 변환

mean(var2)    # 함수 재적용

## [1] 1.8

class(var2)   # 타입 확인

## [1] "numeric"

levels(var2)  # 범주 확인

## NULL
```

1분 퀴즈

mpg 데이터를 이용해서 아래 문제를 해결해보세요. drv 변수는 자동차의 구동방식을 나타냅니다(f:Front Wheel Drive(전륜구동), r:Rear Wheel Drive(후륜구동), 4:Four Wheel Drive(4륜구동)).

- 1.drv 변수의 타입을 확인해보세요.
- 2.drv 변수를 `as.factor()`를 이용해서 factor 타입으로 변환한 후 다시 타입을 확인해보세요.
- 3.drv가 어떤 범주로 구성되는지 확인해보세요.

1분 퀴즈

mpg 데이터를 이용해서 아래 문제를 해결해보세요. drv 변수는 자동차의 구동방식을 나타냅니다(f:Front Wheel Drive(전륜구동), r:Rear Wheel Drive(후륜구동), 4:Four Wheel Drive(4륜구동)).

- 1.drv 변수의 타입을 확인해보세요.
- 2.drv 변수를 `as.factor()`를 이용해서 factor 타입으로 변환한 후 다시 타입을 확인해보세요.
- 3.drv가 어떤 범주로 구성되는지 확인해보세요.

```
class(mpg$drv)                  # 타입 확인
## [1] "character"
mpg$drv <- as.factor(mpg$drv)  # factor 로 변환
class(mpg$drv)                  # 타입 확인
## [1] "factor"
levels(mpg$drv)                 # 범주 확인
## [1] "4" "f" "r"
```

변환 함수

함수	기능
as.numeric()	숫자로 변환
as.factor()	factor로 변환
as.character()	문자로 변환
as.Date()	날짜로 변환
as.data.frame()	데이터 프레임으로 변환

변수 타입들

Type	형태	특징
numeric	숫자	연산 가능
character	문자	데이터 앞 뒤에 겹따옴표가 표시됨
factor	범주	숫자/문자 형태, 숫자 형태라도 연산 불가, 인자(Levels)를 지님
logical	논리	참/거짓을 의미, 대문자 TRUE/FALSE 또는 T/F로 표시됨

Data Frame 외 데이터 구조

matrix

- 행과 열로 구성된 데이터 셋
- 행렬 연산시 사용
- data frame과 달리 행, 열 모두 같은 변수 타입으로만 구성

```
x <- c(1:12)
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

```
# 2 열 행렬
a <- matrix(x, ncol = 2)
a
```

```
##      [,1] [,2]
## [1,]    1    7
## [2,]    2    8
## [3,]    3    9
## [4,]    4   10
## [5,]    5   11
## [6,]    6   12
```

```r
# 2 행 행렬
a<- matrix(x, nrow = 2)
a
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    3    5    7    9   11
## [2,]    2    4    6    8   10   12
```

```r
# 연산
a + 2
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    3    5    7    9   11   13
## [2,]    4    6    8   10   12   14
```

array

- 차원이 여러개인 행열
- 3차원 이상의 행렬 표현 가능

```
# 1~20 으로 2 행 x 5 열 x 2 차원
array(1:20, dim=c(2, 5, 2))
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]    2    4    6    8   10
##
## , , 2
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   11   13   15   17   19
## [2,]   12   14   16   18   20
```

```r
# 1~30 으로 2 행 x 5 열 x 3 차원
array(1:30, dim=c(2, 5, 3))
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]    2    4    6    8   10
##
## , , 2
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   11   13   15   17   19
## [2,]   12   14   16   18   20
##
## , , 3
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   21   23   25   27   29
## [2,]   22   24   26   28   30
```

list

- 여러 유형의 데이터를 하나로 묶은 데이터 유형
- 벡터, 행렬, 데이터 프레임 등 혼합 구성 가능
- 여러 유형의 데이터 통합 활용시 사용

```r
# 여러 유형의 데이터 생성
x1 <- c(1, 2, 3, 4) # Vector
x2 <- matrix(c(1:6), ncol = 2) # Matrix
x3 <- array(1:20, c(2,5,2)) # Array
x4 <- data.frame(a=c(1,2,3), b=c("a","b","c")) # Data frame
```

```r
# 리스트 생성
x.list <- list(f1 = x1, f2 = x2, f3 = x3, f4 = x4)
x.list
```

```
## $f1
## [1] 1 2 3 4
##
## $f2
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
##
## $f3
## , , 1
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]    2    4    6    8   10
##
## , , 2
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   11   13   15   17   19
## [2,]   12   14   16   18   20
##
```

```
##
## $f4
##   a b
## 1 1 a
## 2 2 b
## 3 3 c
```

```r
# 리스트 요소 접근
x.list$f2[2,]  # f2 요소의 2 행
```

```
## [1] 2 5
```

```r
x.list$f4$a    # f4 요소의 변수 a
```

```
## [1] 1 2 3
```

준비는 모두 끝났다. 실전투입!

실전 분석 미션1 "한국인의 삶을 파악하라!" - 한국복지패널 데이터

분석 목표

- 분석1: 성별에 따른 소득
- 분석2: 나이와 소득의 관계
- 분석3: 연령대에 따른 소득
- 분석4: 연령대 및 성별에 따른 소득

준비하기

foreign 패키지 설치

```
install.packages("foreign")
```

패키지 로드

```
library(foreign)
library(dplyr)
library(ggplot2)
```

데이터 불러오기

```
# 복지패널데이터 로드
raw_welfare <- read.spss("data_spss_Koweps2014.sav", to.data.frame = T)

# 데이터 copy
welfare <- raw_welfare
```

데이터 검토

```
dim(welfare)
str(welfare)
head(welfare)
summary(welfare)
View(welfare)
```

변수명

```
welfare <- rename(welfare,
                  sex = h0901_4,     # 성별
                  birth = h0901_5,   # 태어난 연도
                  income = h09_din)  # 소득
```

분석1: 성별에 따른 소득

절차

1.변수 검토 및 정제 - 성별

- 1-1.변수 검토, 수정
- 1-2.정제 - 이상치 확인 및 결측처리

2.변수 검토 및 정제 - 소득

- 2-1.변수 검토, 수정
- 2-2.정제 - 이상치 확인 및 결측처리

3.성별 소득 평균 분석

- 성별 소득 평균표 생성
- 그래프 생성

1.변수 검토 및 정제- 성별

1-1.변수 검토, 수정

```
class(welfare$sex)
```

```
## [1] "numeric"
```

```
summary(welfare$sex)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.309   2.000   2.000
```

```
table(welfare$sex)
```

```
##
##    1    2
## 4873 2175
```

1-2.정제 - 이상치 확인 및 결측처리

- 성별 이상치 : 모름/무응답=9

```
# 이상치 확인
table(welfare$sex)
```

```
##
##    1    2
## 4873 2175
```

```
# 이상치 결측 처리
welfare$sex <- ifelse(welfare$sex == 9, NA, welfare$sex)

# 결측치 확인
table(is.na(welfare$sex))
```

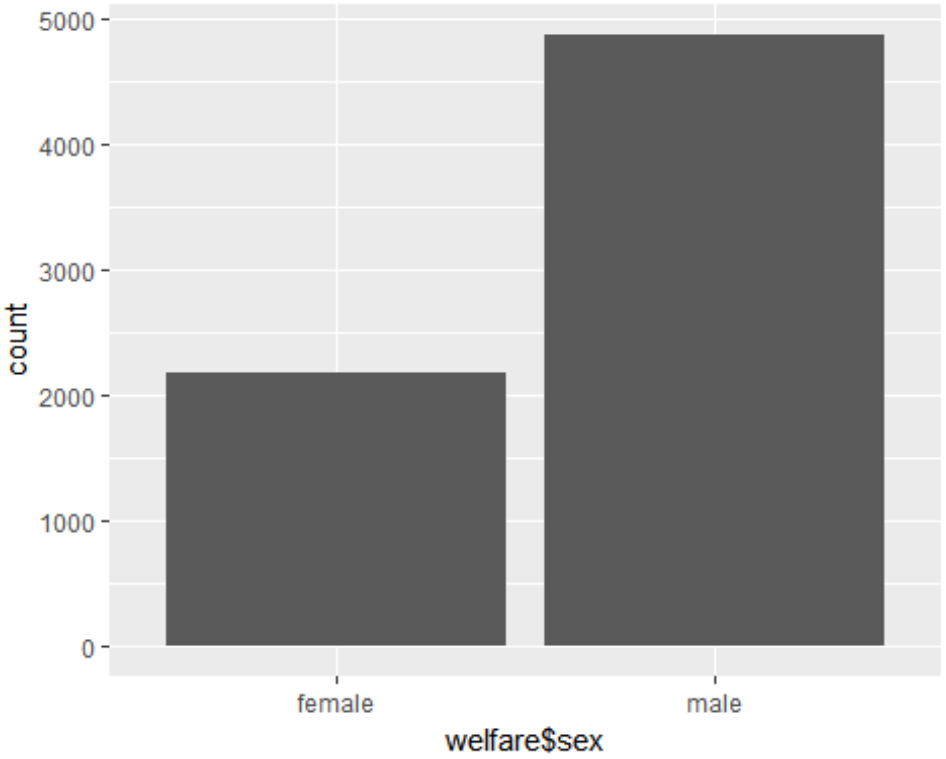
```
##
## FALSE
##  7048
```

변수 값 변경

```
# 항목 이름 부여
welfare$sex <- ifelse(welfare$sex == 1, "male", "female")
table(welfare$sex)
```

```
##
## female   male
##   2175   4873
```

```
qplot(welfare$sex)
```


5000
4000
3000
2000
1000
0
count
female
male
welfare$sex

2.변수 검토 및 정제- 소득

2-1.변수 검토, 수정

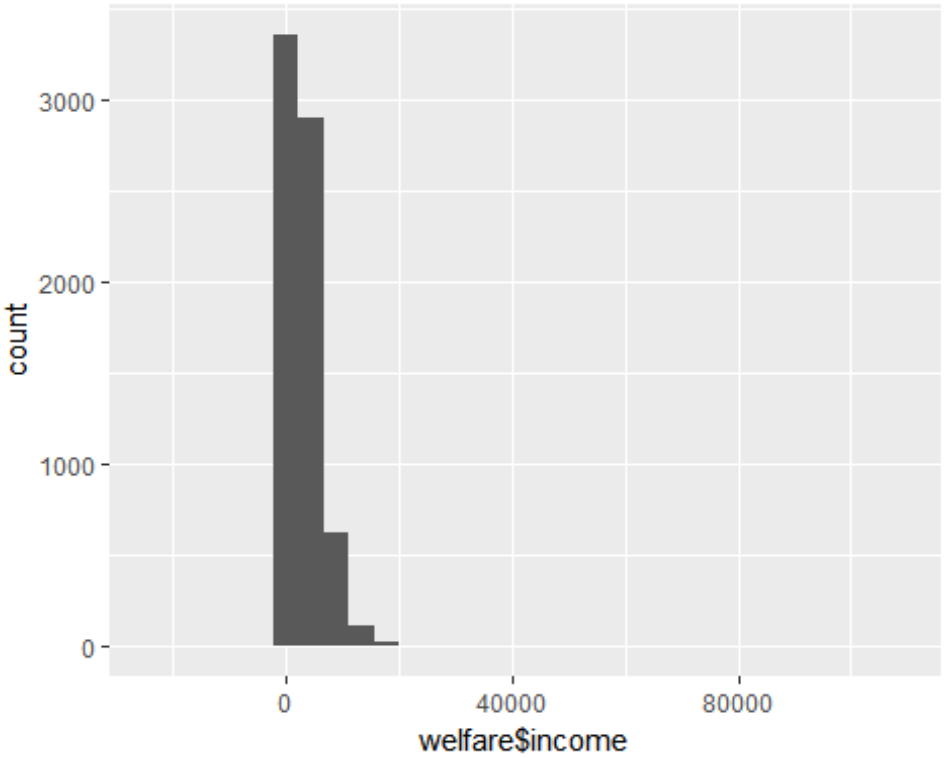
```
class(welfare$income)
```

```
## [1] "numeric"
```

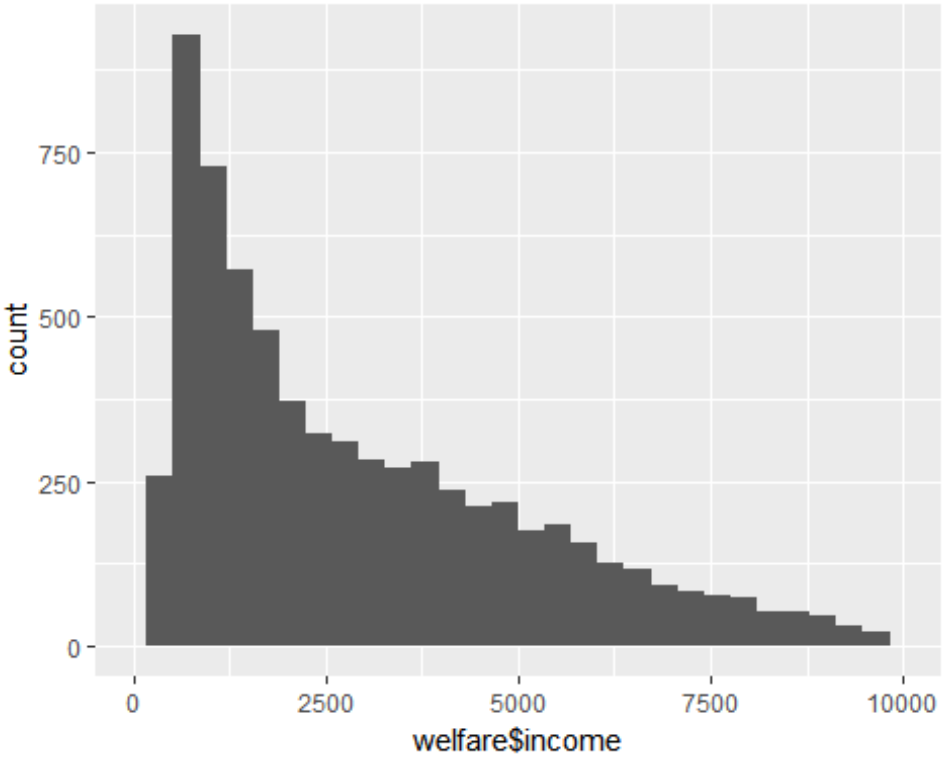
```
summary(welfare$income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -20516    1108    2404    3336    4642  108888
```

```
qplot(welfare$income)
```



```r
qplot(welfare$income) + xlim(0, 10000)  # x 축 설정
```



2-2.정제 - 이상치 확인 및 결측처리

- 소득 이상치 : 모름/무응답 없음

```
table(is.na(welfare$income))
```

```
##
## FALSE
##  7048
```

3.성별 소득 평균 분석

성별 소득 평균표 생성

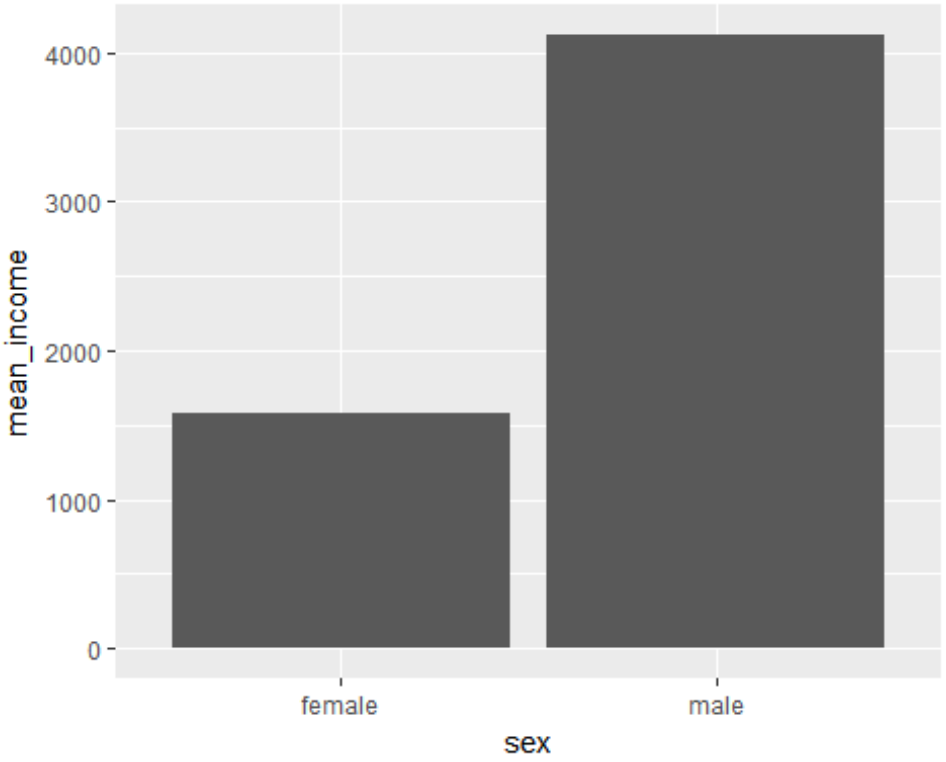
```
sex_income <- welfare %>%
  group_by(sex) %>%
  summarise(mean_income = mean(income))

sex_income
```

```
## # A tibble: 2 x 2
##      sex mean_income
##    <chr>       <dbl>
## 1 female    1581.255
## 2   male    4118.903
```

그래프 생성

```
ggplot(data = sex_income, aes(x = sex, y = mean_income)) + geom_col()
```



분석2: 나이와 소득의 관계

절차

1.변수 검토 및 정제 - 나이

- 1-1.태어난 연도 변수 검토
- 1-2.정제 - 이상치 확인 및 결측처리
- 1-3.나이 변수 생성

2.변수 검토 및 정제 - 소득

- 앞에서 완료됨

3.나이별 소득 평균 분석

- 나이별 소득 평균표 생성
- 그래프 생성

1.변수 검토 및 정제- 나이

1-1.태어난 연도 변수 검토

1.변수 검토 및 정제- 나이

1-1.태어난 연도 변수 검토

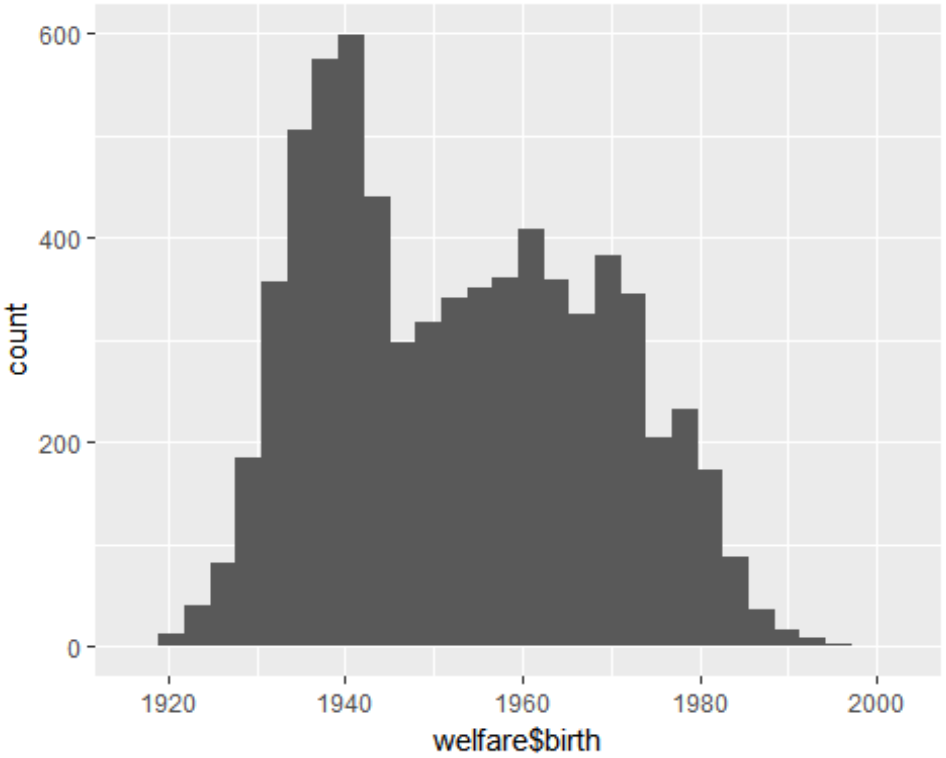
```
class(welfare$birth)
```

```
## [1] "numeric"
```

```
summary(welfare$birth)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1918    1940    1952    1953    1966    2002
```

```
qplot(welfare$birth)
```


600
400
200
0
count
1920
1940
1960
1980
2000
welfare$birth

1-2.정제 - 이상치 확인 및 결측처리

- 태어난 연도 이상치 : 모름/무응답=9999
  - (1)이상치 확인, 결측처리
  - (2)결측치 확인

1-2.정제 - 이상치 확인 및 결측처리

- 태어난 연도 이상치 : 모름/무응답=9999

```
# 이상치 확인
summary(welfare$birth)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1918    1940    1952    1953    1966    2002
```

```
# 이상치 결측처리
welfare$birth <- ifelse(welfare$birth == 9999, NA, welfare$birth)

# 결측치 확인
table(is.na(welfare$birth))
```

```
##
## FALSE
##  7048
```

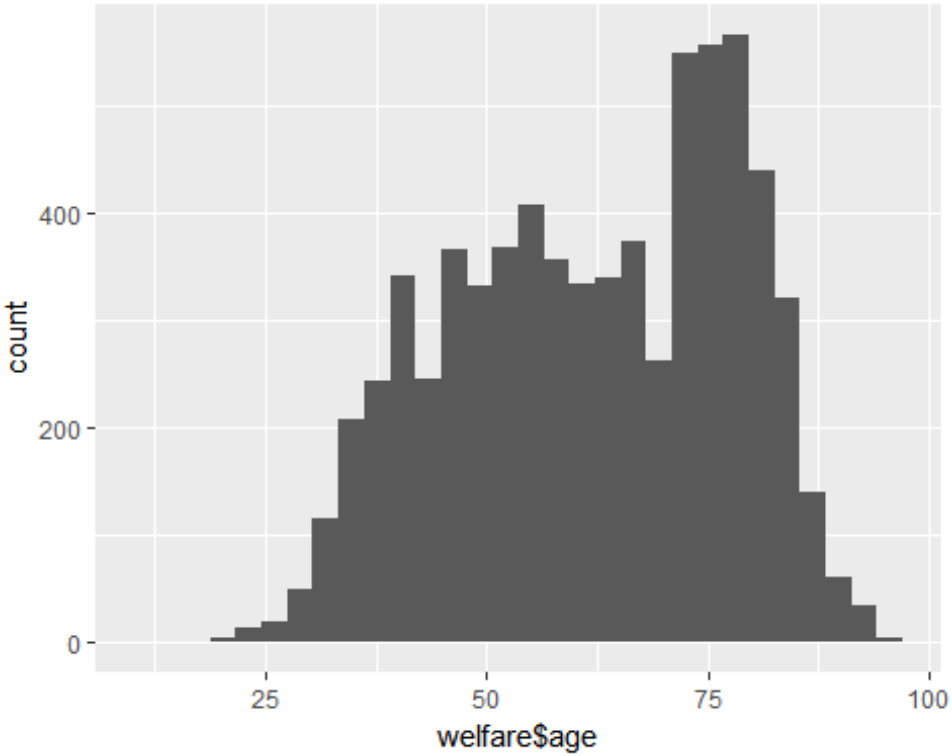
1-3.나이 변수 생성

1-3.나이 변수 생성

```
welfare$age <- 2014-welfare$birth+1
summary(welfare$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.00   49.00   63.00   62.01   75.00   97.00
```

```
qplot(welfare$age)
```



2.변수 검토 및 정제- (2)소득

- 앞에서 완료됨

3.나이별 소득 평균 분석

나이별 소득 평균표 생성

3.나이별 소득 평균 분석

나이별 소득 평균표 생성

```
age_income <- welfare %>%
  group_by(age) %>%
  summarise(mean_income = mean(income))

age_income
```

```
## # A tibble: 79 x 2
##      age mean_income
##    <dbl>       <dbl>
## 1     13    252.0000
## 2     20   1094.9000
## 3     21   2117.6000
## 4     22   2656.0000
## 5     23   1748.2500
## 6     24   5429.6000
## 7     25   2310.4000
## 8     26   5273.3714
## 9     27   3394.9800
## 10    28   3061.2222
## 11    29   6700.5000
## 12    30   3829.3478
```

```
## 13    31   4631.0200
## 14    32   4120.4977
## 15    33   4602.2392
## 16    34   4890.4436
## 17    35   6498.3254
## 18    36   5183.6235
## 19    37   5245.9724
## 20    38   5339.9528
## 21    39   4935.6589
## 22    40   5451.0591
## 23    41   5600.6653
## 24    42   5028.2800
## 25    43   5611.1456
## 26    44   5915.0214
## 27    45   4902.6651
## 28    46   5151.2195
## 29    47   4536.9596
## 30    48   5095.5735
## 31    49   5410.0626
## 32    50   5025.2876
## 33    51   4442.5982
## 34    52   5274.7871
## 35    53   4968.7415
## 36    54   4796.8304
## 37    55   4810.4013
## 38    56   4764.4008
```

```
## 39    57   4923.5692
## 40    58   4136.8440
## 41    59   4286.6234
## 42    60   4261.5271
## 43    61   3471.1073
## 44    62   3729.9290
## 45    63   3783.8135
## 46    64   2702.1936
## 47    65   3225.8067
## 48    66   3010.4055
## 49    67   2599.9184
## 50    68   2480.3602
## 51    69   2541.8000
## 52    70   2450.2841
## 53    71   2225.1972
## 54    72   1929.1064
## 55    73   1791.0443
## 56    74   1823.9989
## 57    75   1840.8915
## 58    76   1619.7357
## 59    77   1472.0482
## 60    78   1594.6102
## 61    79   1393.5847
## 62    80   1286.4084
## 63    81   1307.8524
## 64    82   1262.2224
```

```
## 65    83   1294.1739
## 66    84   1027.0374
## 67    85   1169.5734
## 68    86   1278.5464
## 69    87   1158.0653
## 70    88   1022.4857
## 71    89   1150.9250
## 72    90    974.3478
## 73    91   1270.5857
## 74    92    713.3385
## 75    93    696.6308
## 76    94   1545.5250
## 77    95    828.5000
## 78    96   2041.0000
## 79    97   1109.0000
```

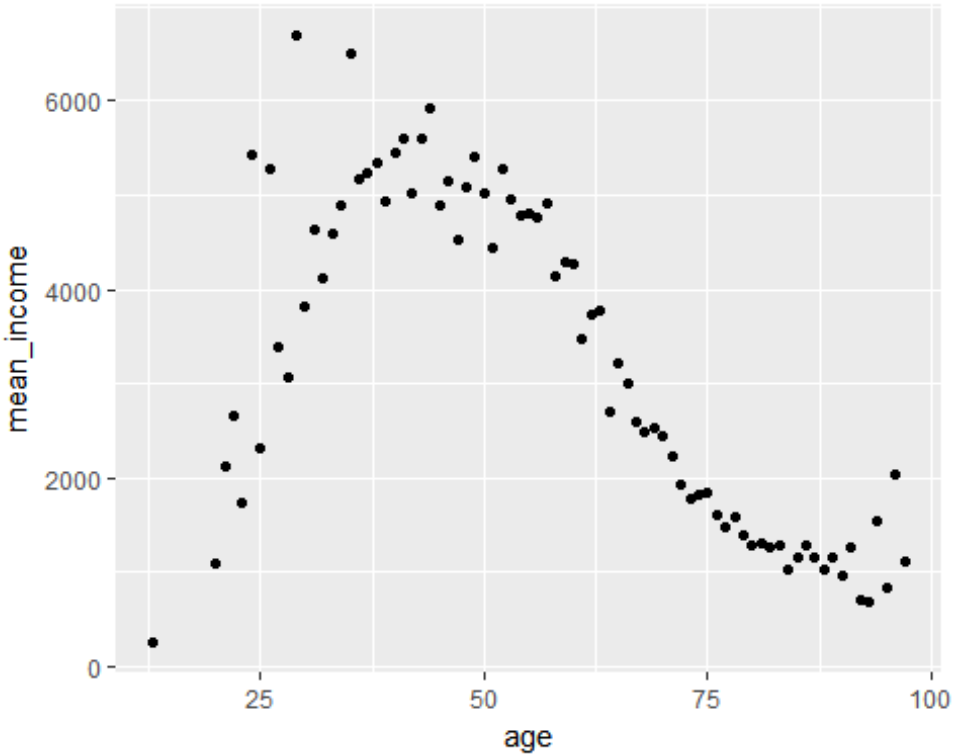
3.나이별 소득 평균 분석

그래프 생성 - 산점도

3.나이별 소득 평균 분석

그래프 생성 - 산점도

```
ggplot(data = age_income, aes(x = age, y = mean_income)) + geom_point()
```



분석3: 연령대에 따른 소득

절차

1.변수 검토 및 정제 - 연령대

- 1-1.연령대 변수 생성

2.변수 검토 및 정제 - 소득

- 앞에서 완료됨

3.연령대별 소득 평균 분석

- 연령대별 소득 평균표 생성
- 그래프 생성

1.변수 검토 및 정제 - 연령대

1-1.연령대 변수 생성

범주	기준
초년	30세 미만
중년	30~59세
노년	60세 이상

1.변수 검토 및 정제 - 연령대

1-1.연령대 변수 생성

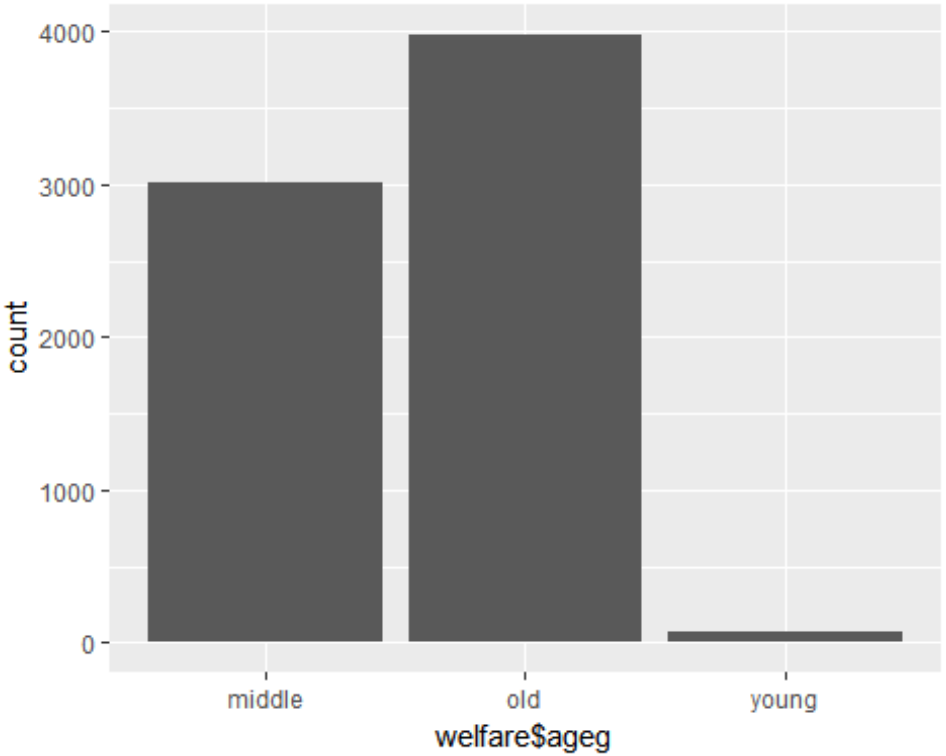
범주	기준
초년	30세 미만
중년	30~59세
노년	60세 이상

```
welfare <- welfare %>%
  mutate(ageg = ifelse(age < 30, "young",
                       ifelse(age <= 59, "middle", "old")))


table(welfare$ageg)
```

```
##
## middle    old  young
##   3004   3979     65
```

```
qplot(welfare$ageg)
```


4000
3000
2000
1000
0
count
middle
old
young
welfare$ageg

2.변수 검토 및 정제 - 소득

- 앞에서 완료됨

3.연령대별 소득 평균 분석

연령대별 소득 평균표 생성

- 초년 빈도 적으므로 제외

3.연령대별 소득 평균 분석

연령대별 소득 평균표 생성

- 초년 빈도 적으므로 제외

```
welfare_income <- welfare %>%
  filter(ageg != "young") %>%
  group_by(ageg) %>%
  summarise(mean_income = mean(income))

welfare_income
```

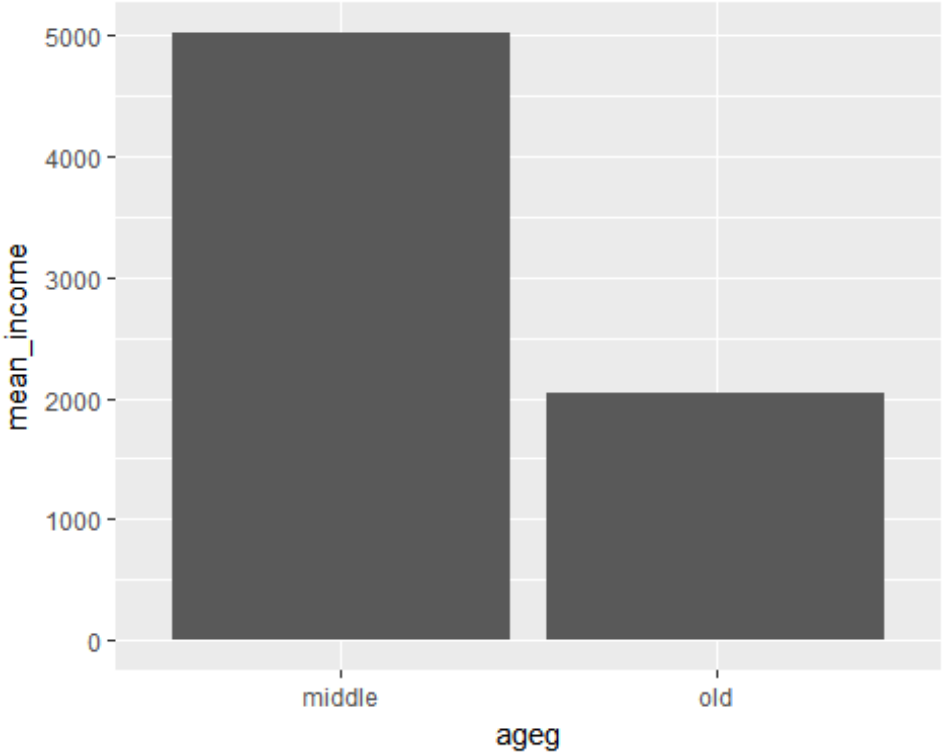
```
## # A tibble: 2 x 2
##     ageg mean_income
##    <chr>       <dbl>
## 1 middle    5017.822
## 2    old    2049.348
```

3.연령대별 소득 평균 분석

그래프 만들기

그래프 만들기

```
ggplot(data = welfare_income, aes(x = ageg, y = mean_income)) + geom_col()
```



분석4: 연령대 및 성별에 따른 소득

절차

1.연령대 및 성별 소득 평균표 생성

2.그래프 만들기

1.연령대 및 성별 소득 평균표 생성

- 초년 제외

1.연령대 및 성별 소득 평균표 생성

- 초년 제외

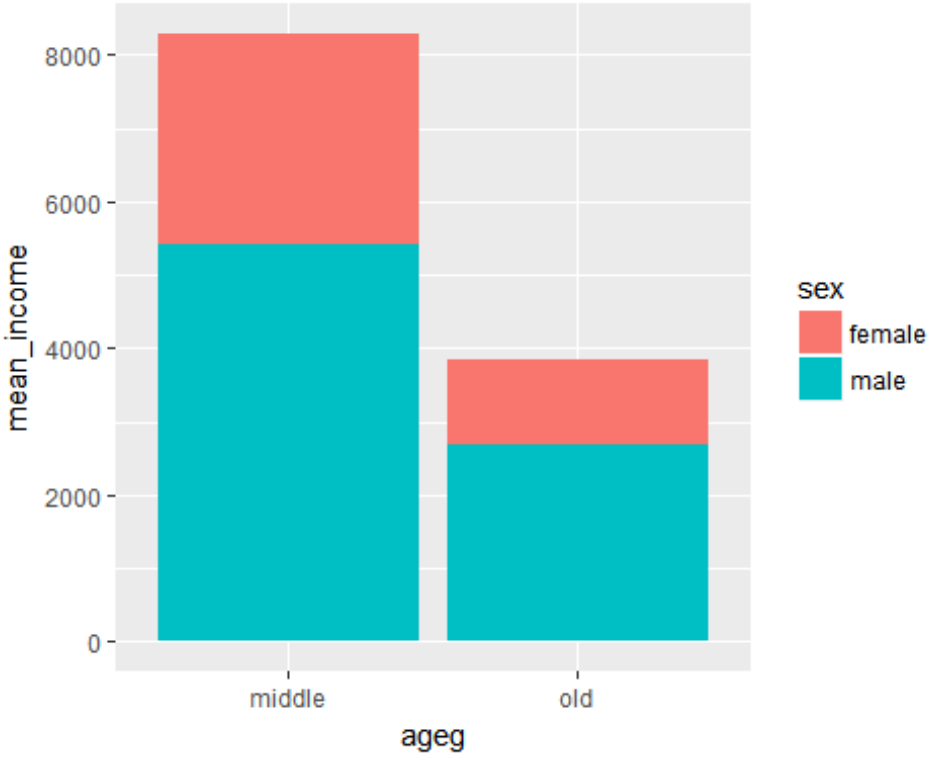
```
sex_income <- welfare %>%
  filter(ageg != "young") %>%
  group_by(ageg, sex) %>%
  summarise(mean_income = mean(income))

sex_income
```

```
## Source: local data frame [4 x 3]
## Groups: ageg [?]
##
##     ageg    sex mean_income
##    <chr>  <chr>       <dbl>
## 1 middle female    2868.804
## 2 middle   male    5420.444
## 3    old female    1179.192
## 4    old   male    2674.162
```

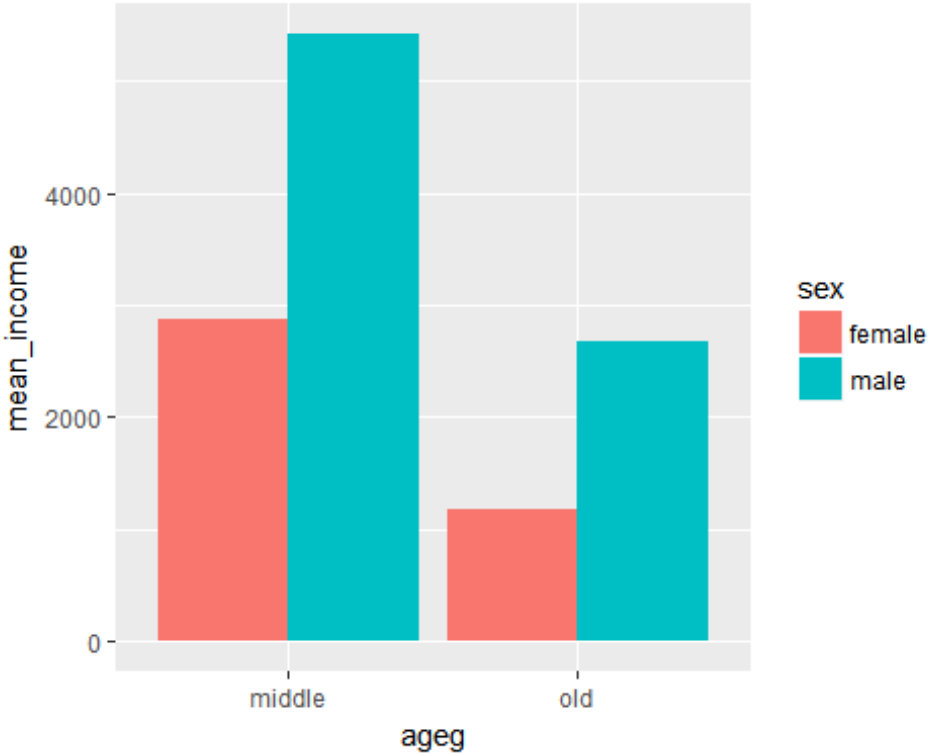
2.그래프 만들기

```
ggplot(data = sex_income, aes(x = ageg, y = mean_income, fill = sex)) +
  geom_col()
```



2.그래프 만들기

```
ggplot(data = sex_income, aes(x = ageg, y = mean_income, fill = sex)) +
  geom_col(position = "dodge")  # position 변경(기본값 = "stack")
```



준비는 모두 끝났다. 실전투입!

실전 분석 미션2 "20~30대의 연애행각을 파악하라!" - 연애실태조사 데이터

분석 목표

- 분석1 : 어장 관리 경험에 따른 연애 횟수
- 분석2 : 어장 피해 경험에 따른 연애 횟수
- 분석3 : 외도 경험에 따른 연애 횟수
- 분석4 : 연애 횟수와 외도 횟수의 관계
- 분석5 : 연애 횟수와 연애 만족도의 관계
- 분석6 : 처음 키스한 나이와 연애 횟수의 관계

준비하기

패키지 로드

```
library(dplyr)
library(ggplot2)
library(readxl)
```

데이터 로드

```
raw_date <- read_excel("data_xlsx_lovesurvey.xlsx",
                       sheet = 1,
                       col_names = T)

date <- raw_date
```

데이터 확인

```
dim(date)
str(date)
head(date)
summary(date)
View(date)
```

분석1 : 어장 관리 경험에 따른 연애 횟수

절차

1.어장 관리 경험 변수 검토 및 정제

2.연애횟수 변수 검토 및 정제

3.어장 관리 경험에 따른 연애 횟수 분석

- 어장 관리 경험별 연애 횟수 평균표 생성
- 그래프 만들기

1.어장 관리 경험 변수 검토 및 정제

```
# 어장 관리 경험 변수 검토
class(date$fishing)
```

```
## [1] "numeric"
```

```
table(date$fishing)
```

```
##
##  1  2
## 27 31
```

```
# 결측치 확인
table(is.na(date$fishing))
```

```
##
## FALSE  TRUE
##    58    36
```

```r
# 항목 이름 부여
date <- date %>%
  mutate(fishing = ifelse(fishing == 1, "yes", "no"))

table(date$fishing)
```

```
##
##  no yes
##  31  27
```

2.연애횟수 변수 검토 및 정제

```
# 연애횟수 변수 검토
class(date$num_love)
```

```
## [1] "numeric"
```

```
summary(date$num_love)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   2.000   2.805   3.000  10.000      17
```

```
table(date$num_love)
```

```
##
##  0  1  2  3  4  5  6  7  8 10
##  8 15 17 18  5  4  5  1  2  2
```

```
# 결측치 확인
table(is.na(date$num_love))
```

```
##
## FALSE  TRUE
##    77    17
```

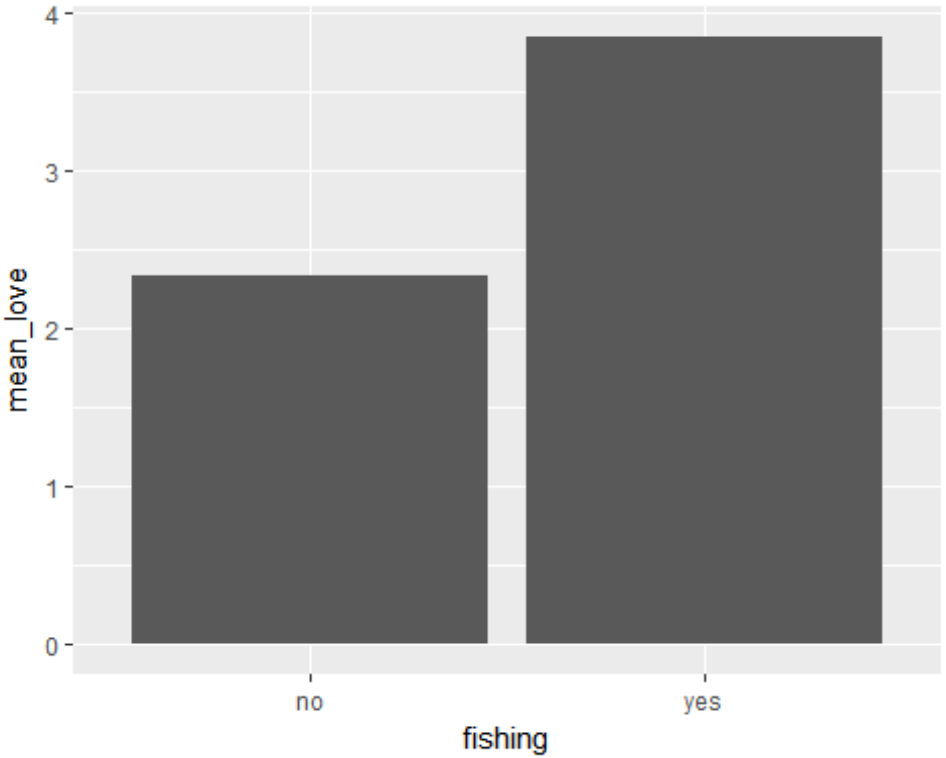
3.어장 관리 경험에 따른 연애 횟수 분석

```
date_num <- date %>%
  filter(!is.na(fishing) & !is.na(num_love)) %>%
  group_by(fishing) %>%
  summarise(mean_love = mean(num_love))

date_num
```

```
## # A tibble: 2 x 2
##   fishing mean_love
##     <chr>     <dbl>
## 1      no  2.333333
## 2     yes  3.851852
```

```
# 그래프 만들기
ggplot(data = date_num, aes(x = fishing, y = mean_love)) + geom_col()
```


4
3
2
1
0
mean_love
no
yes
fishing

‘통계, 히든:그레이스에서 답을 찾다

데이터분석강의

공공데이터분석

논문통계분석

논문통계강의

빅데이터분석

HBBN

HBBN

세상의 히든카드를 Joker로 만들다

서비스 소개

데이터 분석

예측분석 모델링

데이터분석 교육

논문통계

세상의 히든카드를 Joker로 만들다

Contact

전화

010-5461-7445

메일

stats7445@gmail.com

김영우, Kim Young Woo

데이터분석팀장

www.hidden-analysis.co.kr

010.5461.7445

stats7445@gmail.com

김성은, Kim Sung Eun

대표이사

www.hidden-analysis.co.kr

010.3558.8121

admin@hiddenjgrace.com

세상의 히든카드를 Joker로 만들다

찾아오는 길

대중교통 이용시

지하철

강남역 신분당선 하차 후 강남역 5번 출구, 도보 1분거리

강남역 하차

간선버스 140, 400, 402, 407, 420, 440, 441, 462, 470, 471, 541, 542

광역버스 M4403, M6427, M7412, M7426, 9404, 9408

강남역 도씨에빌 II 하차

직행버스 1311, 5300, 5300-1, 6800

광역버스 9500, 9501, 9802

찾는 방법

(STEP 1) 파리바게뜨와 아리따움 화장품 가게를 지난다.

(STEP 2) CU편의점과 GUGUS 옆 모모안경점 사이에 도씨에빌 2 건물 출입구를 찾는다.

(STEP 3) 엘레베이터 18층을 누른다.

(STEP 4) 오른쪽 끝에 있는 1801호 (유리문)으로 온다.

감사합니다.