목차

참고도서

Do it! 쉽게 배우는 R 데이터 분석 – 김영우 (2017), 이지스퍼블리싱

R 시각화와 통계자료분석 1– 나종화 (2016), 자유아카데미

1차시 빠르게 훑기

```r
# 1. 실수 값을 갖는 변수와 관련 함수
a <- 1
var1 <- c(1, 2, 5, 7, 8)
var2 <- c(1:5)
var3 <- seq(1, 10, by = 2)
mean(var2)

# 2. 문자 변수와 관련 함수
str3 <- "Hello World!"
str5 <- c("Hello!", "World", "is", "good!")
paste(str5, collapse = ",")              # 쉼표를 구분자로 str4 의 단어들 하나로 합치기

# 3. 패키지
install.packages("ggplot2")              # ggplot2 패키지 설치
library(ggplot2)                         # ggplot2 패키지 로드

# 4. 변수 타입
var <- c(1,2,3,1,2)                      # numeric 변수 만들기
var <- factor(c(1,2,3,1,2))              # factor 변수 만들기
var <- factor(c("a", "b", "b", "c"))     # 문자로 구성된 factor 변수 만들기
class(var)                               # 변수 타입 확인하기
levels(var)                              # factor 변수의 구성 범주 확인
var <- as.numeric(var)                   # factor 타입을 numeric 타입으로 변환하기
```

2차시 빠르게 훑기

```r
# 1.변수 만들기, 데이터 프레임 만들기
english <- c(90, 80, 60, 70)  # 영어 점수 변수 생성
math <- c(50, 60, 100, 20)    # 수학 점수 변수 생성
data.frame(english, math)     # 데이터 프레임 생성

# 2. 외부 데이터 이용하기

# 엑셀 파일
library(readxl)                                  # readxl 패키지 로드
df_exam <- read_excel("excel_exam.xlsx")         # 엑셀 파일 불러오기

# CSV 파일
df_csv_exam <- read.csv("csv_exam.csv")          # CSV 파일 불러오기
write.csv(df_midterm, file = "df_midterm.csv")   # CSV 파일로 저장하기

# Rda 파일
load("df_midterm.rda")                           # Rda 파일 불러오기
save(df_midterm, file = "df_midterm.rda")        # Rda 파일로 저장하기
```

3차시 빠르게 훑기

```r
# 1.데이터 준비, 패키지 준비
mpg <- as.data.frame(ggplot2::mpg)  # 데이터 불러오기
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 뷰어창에서 확인
dim(mpg)      # 차원
str(mpg)      # 속성
summary(mpg)  # 요약 통계량
# 3.변수명 수정
mpg <- rename(mpg, company = manufacturer)

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

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

```r
# 1.데이터 추출하기
exam[1,]                                  # 행 번호로 행 추출
exam[exam$class == 1,]                    # 조건을 충족하는 행 추출
exam[exam$class == 1 & exam$math >= 50,]  # 여러 조건을 충족하는 행 추출
```

```r
exam[,1]                                  # 열 번호로 변수 추출
exam[, "class"]                           # 변수명으로 변수 추출
exam[,c("class", "math", "english")]      # 변수명으로 여러 변수 추출
exam[1,3]                                 # 행, 변수 동시 추출 - 인덱스
exam[exam$math >= 50, "english"]          # 행, 변수 동시 추출 - 조건문, 변수명
```

```r
# 2.데이터 구조
a <- 1                                  # 벡터 만들기
b <- "hello"

x1 <- data.frame(var1 = c(1,2,3),       # 데이터 프레임 만들기
                 var2 = c("a","b","c"))

x2 <- matrix(c(1:12), ncol = 2)         # 매트릭스 만들기

x3 <- array(1:20, dim=c(2, 5, 2))       # 어레이 만들기

x4 <- list(f1 = a,                      # 리스트 만들기
           f2 = x1,
           f3 = x2,
           f4 = x3)

# 3.리스트 활용하기
x <- boxplot(mpg$cty)  # 상자 그림 만들기
x$stats[,1]            # 요약 통계량 추출
```

4차시 빠르게 훑기

```r
# 1.조건에 맞는 데이터만 추출하기
exam %>% filter(english >= 80)

# 여러 조건 동시 충족
exam %>% filter(class == 1 & math >= 50)

# 여러 조건 중 하나 이상 충족
exam %>% filter(math >= 90 | english >= 90)
exam %>% filter(class %in% c(1,3,5))


# 2.필요한 변수만 추출하기
exam %>% select(math)
exam %>% select(class, math, english)


# 3.함수 조합하기, 일부만 출력하기
exam %>%
  select(id, math) %>%
  head(10)
```

```r
# 4.순서대로 정렬하기
exam %>% arrange(math)          # 오름차순 정렬
exam %>% arrange(desc(math))    # 내림차순 정렬
exam %>% arrange(class, math)  # 여러 변수 기준 오름차순 정렬

# 5.파생변수 추가하기
exam %>% mutate(total = math + english + science)

# 여러 파생변수 한 번에 추가하기
exam %>%
  mutate(total = math + english + science,
         mean = (math + english + science)/3)

# mutate()에 ifelse() 적용하기
exam %>% mutate(test = ifelse(science >= 60, "pass", "fail"))

# 추가한 변수를 dplyr 코드에 바로 활용하기
exam %>%
  mutate(total = math + english + science) %>%
  arrange(total)
```

```r
# 6.집단별로 요약하기
exam %>%
  group_by(class) %>%
  summarize(mean_math = mean(math))

# 각 집단별로 다시 집단 나누기
mpg %>%
  group_by(manufacturer, drv) %>%
  summarize(mean ctv = mean(ctv))


# 7.데이터 합치기
# 가로로 합치기
total <- left_join(test1, test2, by = "id")

# 세로로 합치기
group_all <- bind_rows(group_a, group_b)
```

5차시 빠르게 훑기- R 내장 그래픽 함수

```
par(mfrow = c(2, 3))

plot(mpg$displ, mpg$hwy, xlim=c(3, 6), ylim=c(10, 30))
barplot(df_mpg$mean_hwy, names.arg = df_mpg$drv)
title(ylab = "평균 hwy")
barplot(freqDrv)
title(ylab = "Frequency")
hist(mpg$hwy, breaks = 50)
plot(economics$date, economics$unemploy, type = "l")
boxplot(mpg$drv, mpg$hwy)
```

5차시 빠르게 훑기 - ggplot2 패키지

```
# 1.산점도
ggplot(data = mpg, aes(x = displ, y = hwy)) + geom_point()

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

# 2.평균 막대 그래프

# 1 단계.평균표 만들기
df_mpg <- mpg %>%
  group_by(drv) %>%
  summarise(mean_hwy = mean(hwy))

# 2 단계.그래프 생성하기, 크기순 정렬하기
ggplot(data = df_mpg, aes(x = reorder(drv, -mean_hwy), y = mean_hwy)) + geom_col()

# 3.빈도 막대 그래프
ggplot(data = mpg, aes(x = drv)) + geom_bar()
```

```r
# 4.선 그래프
ggplot(data = economics, aes(x = date, y = unemploy)) + geom_line()

# 5.상자 그림
ggplot(data = mpg, aes(x = drv, y = hwy)) + geom_boxplot()
```

R 통계 분석 기초 – 1차시

2020-7-29 (수)

구혜민

MCM GI Convergence Lab

이번 차시 목표

R 이해하기

R 설치하고 프로젝트 만들기

R 언어의 구성요소 파악하기

- 변수
- 함수
- 패키지
- 변수타입, 데이터구조

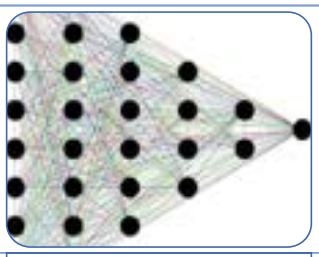
R이란?

- 데이터를 분석하는데 사용되는 소프트웨어

※ R을 활용한 다양한 데이터 분석 예시

기초 통계 분석

- 데이터의 특성을 살펴보는 기초 통계 분석
- 가설 검정에 사용되는 고급 통계 분석 기법



머신 러닝 모델링

- 머신 러닝 (Machine Learning)은 다량의 데이터를 이용해 특정 변수를 예측하는 예측 모형을 만드는 기법
- 랜덤 포레스트, SVM, 딥러닝 등 최신 ML 알고리즘 쉽게 활용 가능

텍스트 마이닝

소셜 네트워크 분석

지도 시각화

등등...

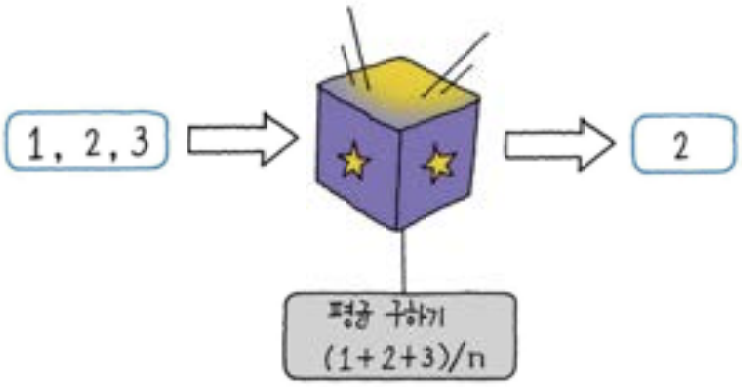
R이 강력한 이유

- 무료로 사용할 수 있는 오픈 소스!
  - SAS, SPSS처럼 전통적인 주요 데이터 분석 소프트웨어들은 대부분 유료
  - R은 무료임에도 상용 툴 못지않은 기능을 갖추고 있음
  - 소규모 또는 개인 사용자들도 손쉽게 데이터 분석 기술을 활용하도록 하여 데이터 분석 기술의 장벽을 낮춤
  - 패키지 공유 사이트인 CRAN에 1만 개가 넘는 패키지 무료 공개,
    이 외에도 GitHub, FTP 등을 통해 최신 분석 기법을 갖춘 패키지가 빠르게 업로드
- 다양한 교육 재료
  - 사용자가 많은 만큼 책, 온라인 강의, 온라인 문서 등 다양한 교육 컨텐츠
- 다양한 그래프를 구현할 수 있다!
  - 멋진 그래프를 만들 수 있는 다양한 기능 제공 https://www.r-graph-gallery.com/
  - 코드 몇 줄로 학술 논문이나 출판물에 사용할 수 있을 정도의 고품질 그래프를 만들 수 있음
  - 데이터 분석부터 그래프 작업까지 하나의 도구로 완성할 수 있어 효율적

R, R Studio 설치하기

- R 과 함께 데이터 분석을 용이하게 해주는 R 통합 개발 환경인 R Studio를 함께 설치한다.
  - https://www.r-project.org/
  - https://rstudio.com/
- 주의점: R 설치 경로에 한국어가 포함되지 않도록 한다.
- 설치가 끝나면 R Studio를 실행한다.

3. 데이터 분석을 위한 연장 챙기기



03-1. 변하는 수, '변수' 이해하기

변수(Variable)

- 다양한 값을 지니고 있는 하나의 속성
- 변수는 데이터 분석의 대상

변수			상수
소득	성별	학점	국적
1,000만 원	남자	3.8	대한민국
2,000만 원	남자	4.2	대한민국
3,000만 원	여자	2.6	대한민국
4,000만 원	여자	4.5	대한민국

변수 만들기


```
a <- 1
a

## [1] 1

b <- 2
b

## [1] 2

c <- 3
c

## [1] 3

d <- 3.5
d

## [1] 3.5
```

변수로 연산하기

```
a+b

## [1] 3

a+b+c

## [1] 6

4/b

## [1] 2

5*b

## [1] 10
```

여러 값으로 구성된 변수 만들기

c()

```
var1 <- c(1, 2, 5, 7, 8)     # 숫자 다섯 개로 구성된 var1 생성
var1

## [1] 1 2 5 7 8

var2 <- c(1:5)               # 1~5 까지 연속값으로 var2 생성
var2

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

seq()

```
var3 <- seq(1, 5)            # 1~5 까지 연속값으로 var3 생성
var3

## [1] 1 2 3 4 5

var4 <- seq(1, 10, by = 2)   # 1~10 까지 2 간격 연속값으로 var4 생성
var4

## [1] 1 3 5 7 9

var5 <- seq(1, 10, by = 3)   # 1~10 까지 3 간격 연속값으로 var5 생성
var5

## [1]  1  4  7 10
```

연속값 변수로 연산하기

```
var1
```

```
## [1] 1 2 5 7 8
```

```
var1+2
```

```
## [1]  3  4  7  9 10
```

```
var1
```

```
## [1] 1 2 5 7 8
```

```
var2
```

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

```
var1+var2
```

```
## [1]  2  4  8 11 13
```

문자로 된 변수 만들기

```
str1 <- "a"
str1
```

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

```
str2 <- "text"
str2
```

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

```
str3 <- "Hello World!"
str3
```

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

연속 문자 변수 만들기

```
str4 <- c("a", "b", "c")
str4
```

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

```
str5 <- c("Hello!", "World", "is", "good!")
str5
```

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

문자로 된 변수로는 연산할 수 없다

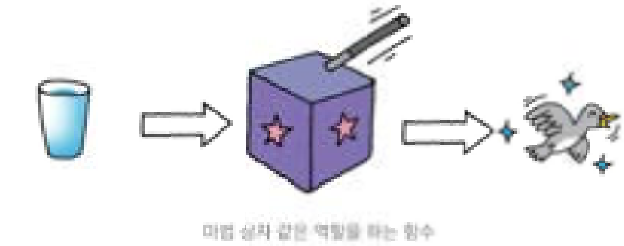
```
str1+2
```

```
## Error in str1 + 2: non-numeric argument to binary operator
```

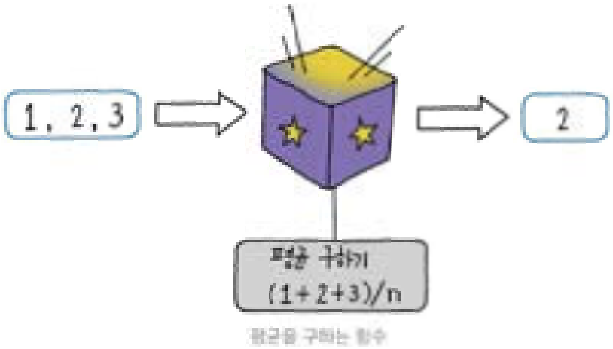
03-2. 마술 상자 같은 '함수' 이해하기

함수

- 값을 넣으면 특정한 기능을 수행해 처음과 다른 값이 출력됨



마법 상자 같은 역할을 하는 함수



평균을 구하는 함수

숫자를 다루는 함수 이용하기

```
# 변수 만들기
x <- c(1, 2, 3)
x
```

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

```
# 함수 적용하기
mean(x)
```

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

```
max(x)
```

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

```
min(x)
```

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

문자를 다루는 함수 이용하기

```
str5
```

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

```
paste(str5, collapse = ",")  # 쉼표를 구분자로 str4의 단어들 하나로 합치기
```

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

함수의 옵션 설정하기 - 파라미터

```
paste(str5, collapse = " ")
```

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

함수의 결과물로 새 변수 만들기

```
x_mean <- mean(x)
x_mean
```

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

```
str5_paste <- paste(str5, collapse = " ")
str5_paste
```

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

사용자 정의 함수 만들기

- 함수명 <- function(파라미터) {실행문}
- 함수 호출 방법: 함수명(파라미터)

```
# 예시: 가중평균을 구하는 함수
weighted.mean <- function(x, weight=rep(1, length(x))) {
  sum(x*weight)/sum(weight)
}
weighted.mean(1:3) # 함수 이용하기
```

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

```
weighted.mean(1:3, 3:1)
```

```
## [1] 1.666667
```

한개의 결과 내보내기

return() 함수를 사용

```
# 예시: 표준오차 구하는 함수
std.error <- function(x) {
  v <- var(x)
  n <- length(x)
  se <- sqrt(v/n)
  return(se)  # return 함수: 한 개의 결과만 내보냄
}
std.error(c(1:10))
```

```
## [1] 0.9574271
```

여러 개의 결과를 리스트로 내보내기

list() 함수를 사용

```
basic.stats <- function(x) {
  n <- length(x)
  m <- mean(x)
  med <- median(x)
  s <- sd(x)
  list(n=n, mean=m, median=med, std=s)
  # list() 함수: 여러 개의 결과만 내보냄
}
basic.stats(c(1:10))
```

```
## $n
## [1] 10
##
## $mean
## [1] 5.5
##
## $median
## [1] 5.5
##
## $std
## [1] 3.02765
```

다른 방식

```
basic.stats <- function(x) {
  stats <- list()
  stats$n <- length(x)
  stats$mean <- mean(x)
  stats$med <- median(x)
  stats$std <- sd(x)
  stats  # 마지막 문장을 결과로 보냄
}
```

벡터 형태로 결과 내보내기

```
# 예시: 표준오차 구하는 함수
basic.stats <- function(x) {
  n <- length(x)
  m <- mean(x)
  med <- median(x)
  s <- sd(x)
  out <- c(n, m, med, s)  # 벡터로 저장
  names(out) <- c("n", "mean", "median", "std")  # 이름 부여
  round(out, 4)  # out 을 출력하되 소수점 4 자리까지만 출력
}
basic.stats(c(1:10))
```

```
##       n    mean  median     std
## 10.0000  5.5000  5.5000  3.0277
```

R 내장 함수 수정하여 사용하기

R에서 제공되는 대부분의 함수는 그 소스가 제공되므로, 필요 시에는 목적에 맞게 수정하여 새로운 이름으로 저장하여 사용할 수 있다. 이 경우, fix() 함수를 이용한다.

```
fix(factorial)  # factorial 함수를 수정
```

R 프로그램 실행하기

R 콘솔창에서 외부에서 작성된 R 프로그램을 실행하는 경우 source() 함수를 이용한다.

```
source("program.txt")
```

예: program.txt 파일의 내용
c <- c(1:5)
y <- seq(10, 50, 10)
mean.x <- mean(x)
mean.y <- mean(y)
std.error <- function(x) {
v <- var(x)
n <- length(x)
se <- sqrt(v/n)
return(se) # return 함수: 한 개의 결과만 내보냄
}
std.error(c(1:10))

함수와 같이 사용되는 제어문

조건문

```
# if... else 문
x <- 2
if (x>=0) sqrt(x) else y <- abs(x)

## [1] 1.414214

# ifelse 문
x <- -2
ifelse(x>=0, sqrt(x), abs(x))

## [1] 2

# switch 문
x<- -2
switch (x>=0, sqrt(x), abs(x))
```

반복문

```
# for 문
x <- 0
for (i in 1:10) {
  x[i] <- i^2
}
x

##  [1]   1   4   9  16  25  36  49  64  81 100

# while 문
x <- 0
i <- 1
while(i<11) {
  x[i] <- i^2
  i <- i+1
}
x

##  [1]   1   4   9  16  25  36  49  64  81 100
```

03-3. 함수 꾸러미, '패키지' 이해하기

패키지(packages)

- 함수가 여러 개 들어 있는 꾸러미
- 하나의 패키지 안에 다양한 함수가 들어있음
- 함수를 사용하려면 패키지 설치 먼저 해야함

ggplot2 패키지 설치하기, 로드하기

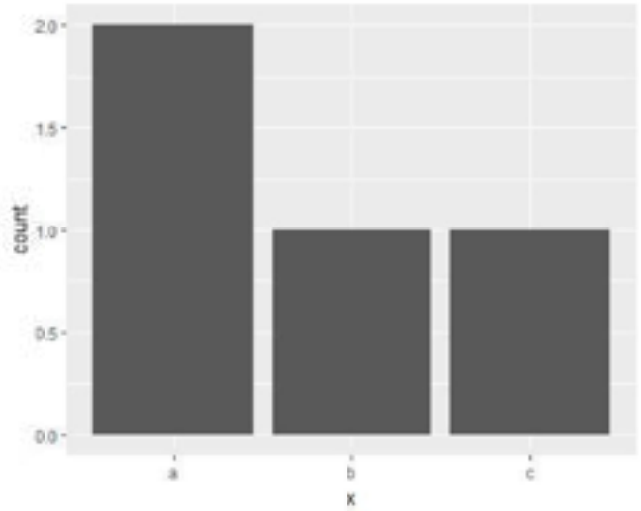
```
install.packages("ggplot2")  # ggplot2 패키지 설치
library(ggplot2)             # ggplot2 패키지 로드
```

함수 사용하기

```
# 여러 문자로 구성된 변수 생성
x <- c("a", "a", "b", "c")
x
```

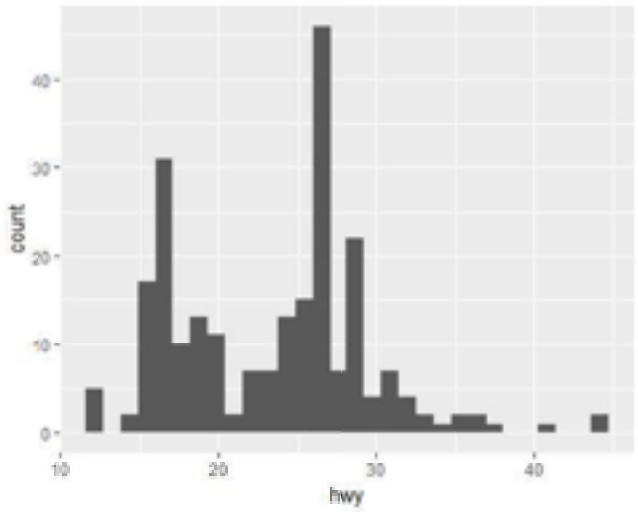
```
## [1] "a" "a" "b" "c"
```

```
# 빈도 그래프 출력
qplot(x)
```



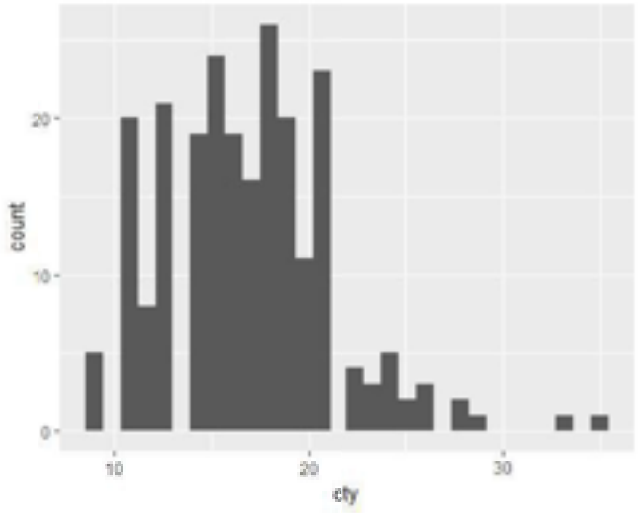
ggplot2의 mpg 데이터로 그래프 만들기

```
# data에 mpg, x축에 hwy 변수 지정하여 그래프 생성
qplot(data = mpg, x = hwy)
```

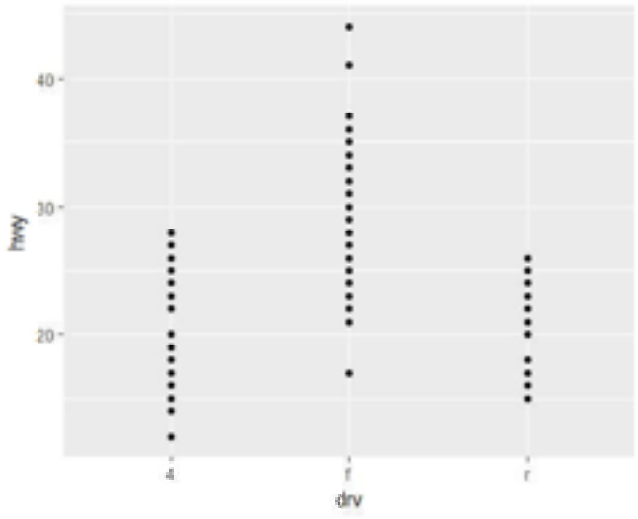


qplot() 파라미터 바꿔보기

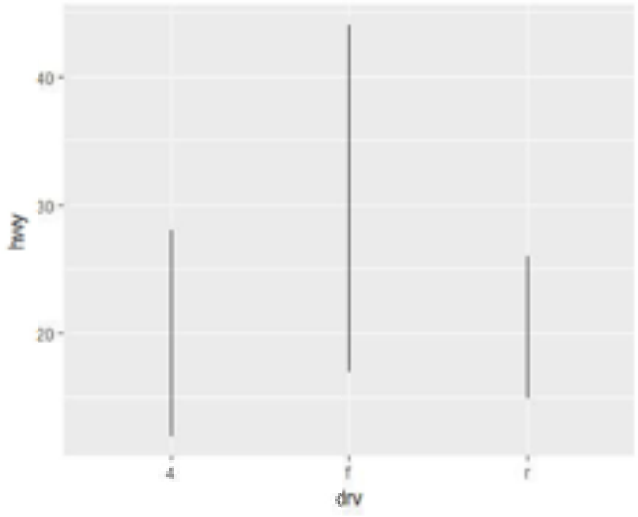
```
# x축 cty
qplot(data = mpg, x = cty)
```



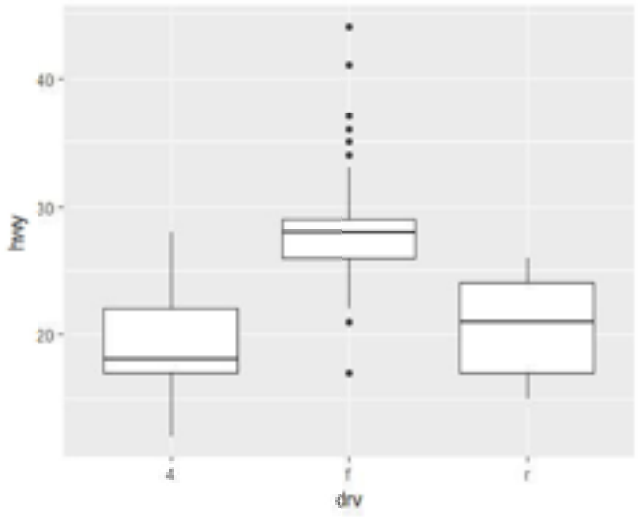
```r
# x 축 drv, y 축 hwy
qplot(data = mpg, x = drv, y = hwy)
```



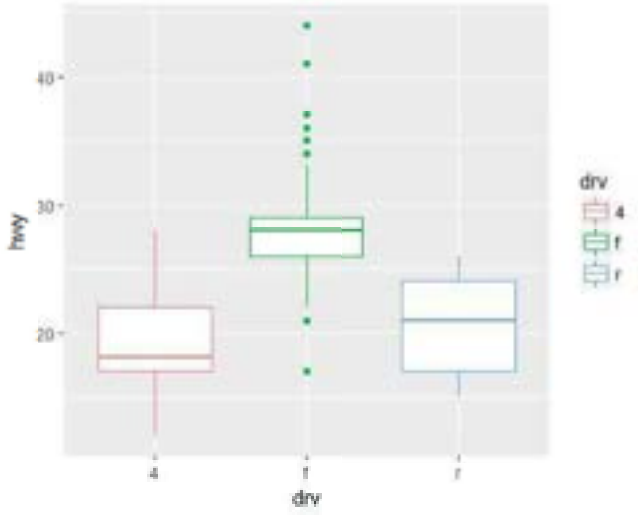

```r
# x 축 drv, y 축 hwy, 선 그래프 형태
qplot(data = mpg, x = drv, y = hwy, geom = "line")
```



```r
# x 축 drv, y 축 hwy, 상자 그림 형태
qplot(data = mpg, x = drv, y = hwy, geom = "boxplot")
```




```r
# x 축 drv, y 축 hwy, 상자 그림 형태, drv 별 색 표현
qplot(data = mpg, x = drv, y = hwy, geom = "boxplot", colour = drv)
```



함수의 기능이 궁금할 땐 Help 함수를 활용해 보세요

```
?qplot
```

혼자서 해보기

Q1. 시험 점수 변수 만들고 출력하기

다섯 명의 학생이 시험을 봤습니다. 학생 다섯 명의 시험 점수를 담고 있는 변수를 만들어 출력해 보세요. 각 학생의 시험 점수는 다음과 같습니다.

`80, 60, 70, 50, 90`

Q2. 전체 평균 구하기

앞 문제에서 만든 변수를 이용해서 이 학생들의 전체 평균 점수를 구해보세요.

Q3. 전체 평균 변수 만들고 출력하기

전체 평균 점수를 담고 있는 새 변수를 만들어 출력해 보세요. 앞 문제를 풀 때 사용한 코드를 응용하면 됩니다.

정답

Q1. 시험 점수 변수 만들고 출력하기

```
score <- c(80, 60, 70, 50, 90)
score
```

```
## [1] 80 60 70 50 90
```

Q2. 전체 평균 구하기

```
mean(score)
```

```
## [1] 70
```

Q3. 전체 평균 변수 만들고 출력하기

```
mean_score <- mean(score)
mean_score
```

```
## [1] 70
```

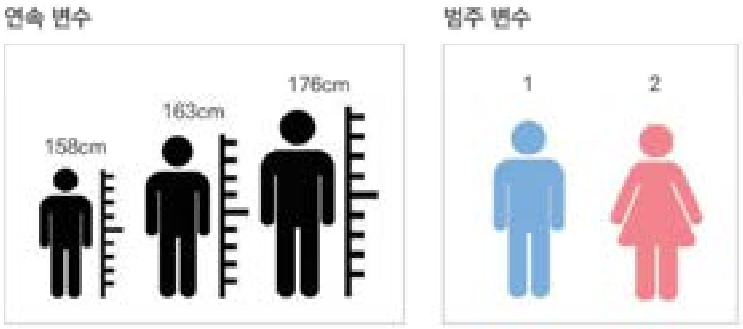
심화

15-2. 변수 타입

변수에는 여러 가지 타입(Type, 속성)이 있음

- 함수에 따라 적용 가능한 변수 타입 다름
- 분석 전에 변수 타입이 무엇인지 확인 필요
- 함수 실행했을 때 오류 발생 또는 예상과 다른 결과가 출력되면 변수 타입 확인 후 함수에 맞게 변경

변수의 종류



- **1. 연속 변수(Continuous Variable) - Numeric 타입**
  - 값이 연속적이고 크기를 의미
  - 더하기 빼기, 평균 구하기 등 산술 가능
  - ex) 키, 몸무게, 소득
- **2. 범주 변수(Categorical Variable) - Factor 타입**
  - 값이 대상을 분류하는 의미를 지님
  - 산술 불가능
  - ex) 성별, 거주지

변수	Data Type	예
연속 변수	Numeric	키(..., 151, 152, ...), 몸무게(..., 58, 59, ...)
범주 변수	Factor	성별(1, 2), 지역(1, 2, 3, 4)

변수 타입 간 차이 알아보기

```
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
```

```
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

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

```
var3
```

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

```
var4
```

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

```
class(var3)
```

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

```
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
```

변환 함수(Coercion Function)

함수	기능
as.numeric()	numeric으로 변환
as.factor()	factor로 변환
as.character()	character로 변환
as.Date()	Date로 변환
as.data.frame()	Data Frame으로 변환

혼자서 해보기

`mpg` 데이터의 `drv` 변수는 자동차의 구동 방식을 나타냅니다. `mpg` 데이터를 이용해 아래 문제를 해결해 보세요.

- Q1. `drv` 변수의 타입을 확인해 보세요.
- Q2. `drv` 변수를 `as.factor()`를 이용해 `factor` 타입으로 변환한 후 다시 타입을 확인해 보세요.
- Q3. `drv`가 어떤 범주로 구성되는지 확인해 보세요.

정답

```
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"
```

R 통계 분석 기초 – 2차시

2020-8-6 (목)

구혜민
MCM GI Convergence Lab

이번 차시 목표

데이터 프레임 만들어 보기
외부 데이터 불러오고 저장하기

4. 데이터 프레임의 세계로!

이름	영어 점수	수학 점수
김지훈	90	50
이유진	80	60
박동현	60	100
김민지	70	20

04-1. 데이터는 어떻게 생겼나? - 데이터 프레임 이해하기

데이터 프레임

이름	영어 점수	수학 점수
김지훈	90	50
이유진	80	60
박동현	60	100
김민지	70	20

데이터 프레임

- '열'은 속성
- '행'은 한 사람의 정보

04-2. 데이터 프레임 만들기 - 시험 성적 데이터를 만들어 보자!

데이터 입력해 데이터 프레임 만들기

```
english <- c(90, 80, 60, 70)  # 영어 점수 변수 생성
english
```

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

```
math <- c(50, 60, 100, 20)    # 수학 점수 변수 생성
math
```

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

```
# english, math 로 데이터 프레임 생성해서 df_midterm 에 할당
df_midterm <- data.frame(english, math)
df_midterm
```

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

```
class <- c(1, 1, 2, 2)
class
```

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

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

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

```
mean(df_midterm$english)  # df_midterm 의 english 로 평균 산출
```

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

```
mean(df_midterm$math)     # df_midterm 의 math 로 평균 산술
```

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

[참고] 같은 함수를 열마다 적용하고 싶다면 `sapply(DATA FRAME, FUNCTION)` 을 이용한다.

데이터 프레임 한 번에 만들기

```
df_midterm <- data.frame(english = c(90, 80, 60, 70),
                         math = c(50, 60, 100, 20),
                         class = c(1, 1, 2, 2))
df_midterm
```

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

혼자서 해보기

Q1. `data.frame()`과 `c()`를 조합해서 표의 내용을 데이터 프레임으로 만들어 출력해보세요.

fruit	price	volume
사과	1800	24
딸기	1500	38
수박	3000	13

Q2. 앞에서 만든 데이터 프레임을 이용해서 과일 가격 평균, 판매량 평균을 구해보세요.

정답

Q1. `data.frame()`과 `c()`를 조합해서 표의 내용을 데이터 프레임으로 만들어 출력해보세요.

```
# 데이터 프레임 만들기
sales <- data.frame(fruit = c("사과", "딸기", "수박"),
                    price = c(1800, 1500, 3000),
                    volume = c(24, 38, 13))

# 데이터 프레임 출력하기
sales
```

```
##   fruit price volume
## 1  사과  1800     24
## 2  딸기  1500     38
## 3  수박  3000     13
```

Q2. 앞에서 만든 데이터 프레임을 이용해서 과일 가격 평균, 판매량 평균을 구해보세요.

```
mean(sales$price)   # 가격 평균
```

```
## [1] 2100
```

```
mean(sales$volume)  # 판매량 평균
```

```
## [1] 25
```

04-3. 외부 데이터 이용하기 - 축적된 시험 성적 데이터를 불러오자!

엑셀 파일 불러오기

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

# readxl 패키지 로드
library(readxl)
```

```r
df_exam <- read_excel("excel_exam.xlsx")  # 엑셀 파일을 불러와서 df_exam 에 할당
df_exam                                   # 출력
```

```
## # A tibble: 20 x 5
##       id class  math english science
##    <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_exam$english)
```

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

```r
mean(df_exam$science)
```

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

getwd()와 paste()를 활용한 직접 경로 지정

```
df_exam <- read_excel(paste(getwd(), "/excel_exam.xlsx", sep = ""))
```

엑셀 파일 첫 번째 행이 변수명이 아니라면?

```
df_exam_novar <- read_excel("excel_exam_novar.xlsx", col_names = F)
df_exam_novar
```

엑셀 파일에 시트가 여러 개 있다면?

```
df_exam_sheet <- read_excel("excel_exam_sheet.xlsx", sheet = 3)
df_exam_sheet
```

csv 파일 불러오기

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

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

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

데이터 프레임을 CSV 파일로 저장하기

```
df_midterm <- data.frame(english = c(90, 80, 60, 70),
                         math = c(50, 60, 100, 20),
                         class = c(1, 1, 2, 2))
df_midterm
```

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

```
write.csv(df_midterm, file = "df_midterm.csv")
write.csv(df_midterm, file = "df_midterm2.csv", row.names = F)
```

RData 파일 활용하기

- R 전용 데이터 파일
- 용량 작고 빠름

데이터 프레임을 RData 파일로 저장하기

```
save(df_midterm, file = "df_midterm.rda")
```

RData 불러오기

```
rm(df_midterm)

df_midterm

## Error in eval(expr, envir, enclos): object 'df_midterm' not found

load("df_midterm.rda")

df_midterm

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

다른 파일을 불러올 때와 차이점

- 엑셀, CSV는 파일을 불러와 새 변수에 할당해서 활용
- rda는 불러오면 저장한 데이터 프레임이 자동 생성됨. 할당 없이 바로 활용

```
# 엑셀 파일 불러와 df_exam 에 할당하기
df_exam <- read_excel("excel_exam.xlsx")

# csv 파일 불러와 df_csv_exam 에 할당하기
df_csv_exam <- read.csv("csv_exam.csv")

# Rda 파일 불러오기
load("df_midterm.rda")
```

R 통계 분석 기초 – 3차시

2020-8-19 (수)

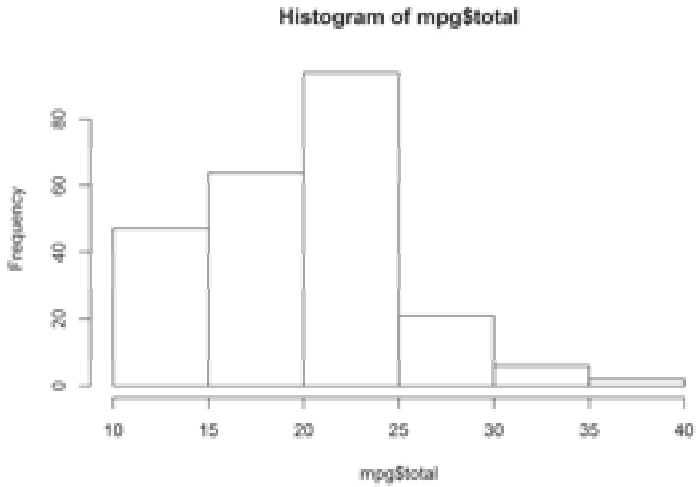
구혜민
MCM GI Convergence Lab

이번 차시 목표

데이터 파악하기, 수정하기
R 내장함수로 데이터 추출하기
(심화) 자료구조

5. 데이터 분석 기초!

데이터 파악하기, 다루기 쉽게 수정하기



05-1. 데이터 파악하기

함수	기능
head()	데이터 앞부분 출력
tail()	데이터 뒷부분 출력
View()	뷰어 창에서 데이터 확인
dim()	데이터 차원 출력
str()	데이터 속성 출력
summary()	요약통계량 출력

exam 데이터 파악하기

데이터 준비

```
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

```
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

```
# ggplo2의 mpg 데이터를 데이터 프레임 형태로 불러오기
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
```

```
View(mpg)    # Raw 데이터 뷰어 창 확인

dim(mpg)     # 행, 열 출력
```

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

```
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

```r
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)  # var2를 v2로 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
```

혼자서 해보기

mpg 데이터의 변수명은 긴 단어를 짧게 줄인 축약어로 되어있습니다. cty 변수는 도시 연비, hwy 변수는 고속도로 연비를 의미합니다. 변수명을 이해하기 쉬운 단어로 바꾸려고 합니다. mpg 데이터를 이용해서 아래 문제를 해결해 보세요.

- Q1. ggplot2 패키지의 mpg 데이터를 사용할 수 있도록 불러온 뒤 복사본을 만드세요.
- Q2. 복사본 데이터를 이용해서 cty는 city로, hwy는 highway로 변수명을 수정하세요.
- Q3. 데이터 일부를 출력해서 변수명이 바뀌었는지 확인해 보세요. 아래와 같은 결과물이 출력되어야 합니다.

```
##   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
```

정답

Q1. ggplot2 패키지의 mpg 데이터를 사용할 수 있도록 불러온 뒤 복사본을 만드세요.

```
mpg <- as.data.frame(ggplot2::mpg)          # mpg 데이터 불러오기
mpg_new <- mpg                              # 복사본 만들기
```

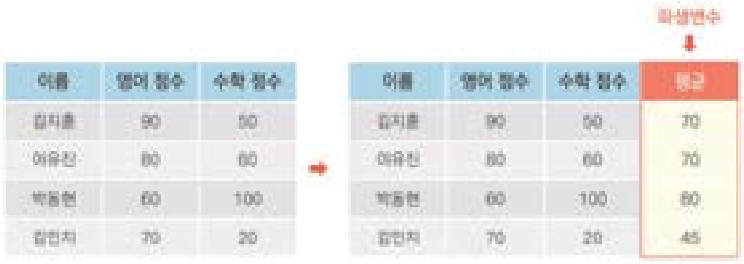
Q2. 복사본 데이터를 이용해서 cty는 city로, hwy는 highway로 변수명을 수정하세요.

```
mpg_new <- rename(mpg_new, city = cty)      # cty 를 city 로 수정
mpg_new <- rename(mpg_new, highway = hwy)   # hwy 를 highway 로 수정
```

Q3. 데이터 일부를 출력해서 변수명이 바뀌었는지 확인해 보세요. 아래와 같은 결과물이 출력되어야 합니다.

```
head(mpg_new)                               # 데이터 일부 출력

##   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
```

05-3. 파생변수 만들기



변수 조합해 파생변수 만들기

데이터 프레임 생성

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

##   var1 var2
## 1    4    2
## 2    3    6
## 3    8    1
```

파생변수 생성

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

```
##   var1 var2 var_sum
## 1    4    2       6
## 2    3    6       9
## 3    8    1       9
```

파생변수 생성

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

```
##   var1 var2 var_sum var_mean
## 1    4    2       6      3.0
## 2    3    6       9      4.5
## 3    8    1       9      4.5
```

[참고] `df$var_mean <- rowMeans(df[,c("var1", "var2")])` 으로도 같은 결과 생성 가능

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
```

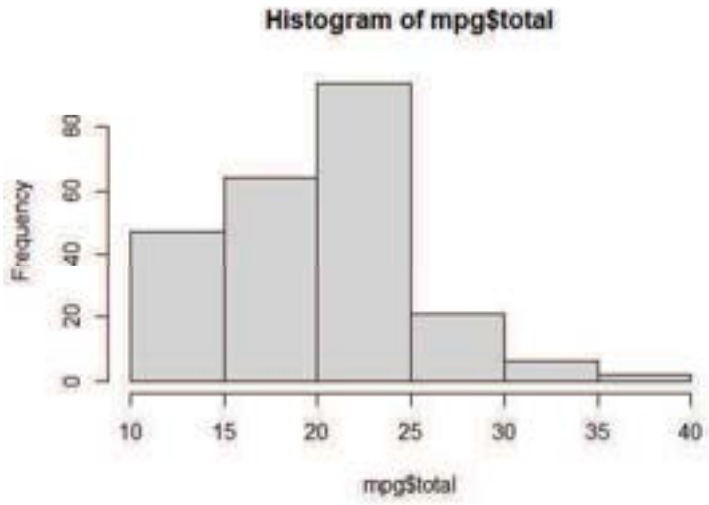
조건문을 활용해 파생변수 만들기

1.기준값 정하기

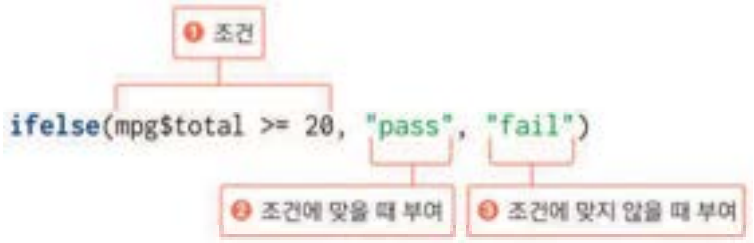
```
summary(mpg$total)  # 요약 통계량 산출
```

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

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



2. 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
```

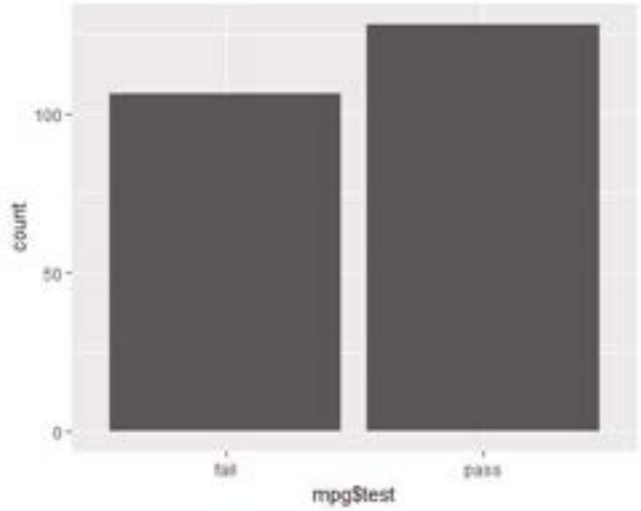
```
## 4   p compact  25.5 pass
## 5   p compact  21.0 pass
## 6   p compact  22.0 pass
## 7   p compact  22.5 pass
## 8   p compact  22.0 pass
## 9   p compact  20.5 pass
## 10  p compact  24.0 pass
## 11  p compact  23.0 pass
## 12  p compact  20.0 pass
## 13  p compact  21.0 pass
## 14  p compact  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
```

3. 빈도표로 합격 판정 자동차 수 살펴보기

```
table(mpg$test)  # 연비 합격 빈도표 생성

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

4. 막대 그래프로 빈도 표현하기

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

head(mpg, 20)  # 데이터 확인
```

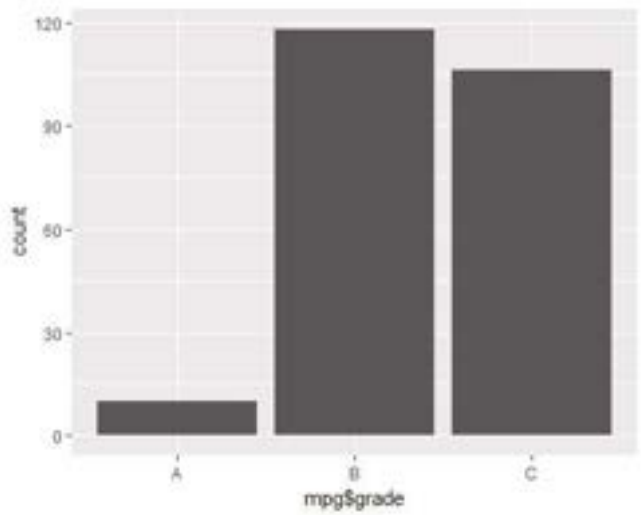
```
##    manufacturer                model displ year cyl      trans drv cty hwy
## 1          audi                   a4   1.8 1999   4   auto(l5)   f  18  29
## 2          audi                   a4   1.8 1999   4 manual(m5)   f  21  29
## 3          audi                   a4   2.0 2008   4 manual(m6)   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
```

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

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
```



원하는 만큼 범주 만들기

```
# A, B, C, D 등급 부여
mpg$grade2 <- ifelse(mpg$total >= 30, "A",
                     ifelse(mpg$total >= 25, "B",
                            ifelse(mpg$total >= 20, "C", "D")))
```

분석 도전!

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

- 문제 1. `ggplot2` 의 `midwest` 데이터를 데이터 프레임 형태로 불러와서 데이터의 특성을 파악하세요.
- 문제 2. `poptotal`(전체 인구)을 `total` 로, `popasian`(아시아 인구)을 `asian` 으로 변수명을 수정하세요.
- 문제 3. `total`, `asian` 변수를 이용해 '전체 인구 대비 아시아 인구 백분율' 파생변수를 만들고, 히스토그램을 만들어 도시들이 어떻게 분포하는지 살펴보세요.
- 문제 4. 아시아 인구 백분율 전체 평균을 구하고, 평균을 초과하면 `"large"`, 그 외에는 `"small"`을 부여하는 파생변수를 만들어 보세요.
- 문제 5. `"large"`와 `"small"`에 해당하는 지역이 얼마나 되는지, 빈도표와 빈도 막대 그래프를 만들어 확인해 보세요.

분석 도전 정답

문제1. `ggplot2`의 `midwest` 데이터를 데이터 프레임 형태로 불러와서 데이터의 특성을 파악하세요.

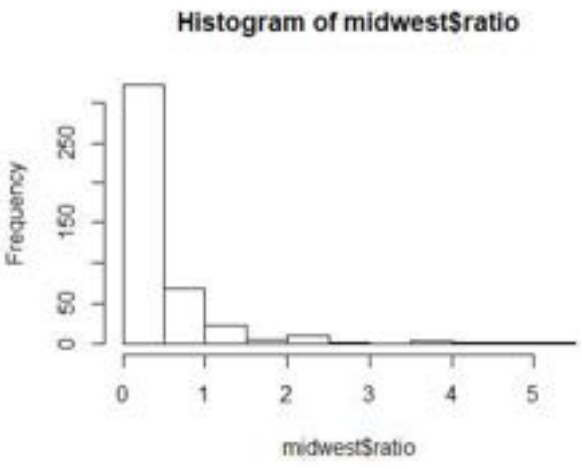
```
midwest <- as.data.frame(ggplot2::midwest)
head(midwest)
tail(midwest)
View(midwest)
dim(midwest)
str(midwest)
summary(midwest)
```

문제2. poptotal(전체 인구)을 total로, popasian(아시아 인구)을 asian으로 변수명을 수정하세요.

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

문제3. total, asian 변수를 이용해 '전체 인구 대비 아시아 인구 백분율' 파생변수를 만들고, 히스토그램을 만들어 도시들이 어떻게 분포하는지 살펴보세요.

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



문제4. 아시아 인구 백분율 전체 평균을 구하고, 평균을 초과하면 `"large"`, 그 외에는 `"small"`을 부여하는 파생변수를 만들어 보세요.

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

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

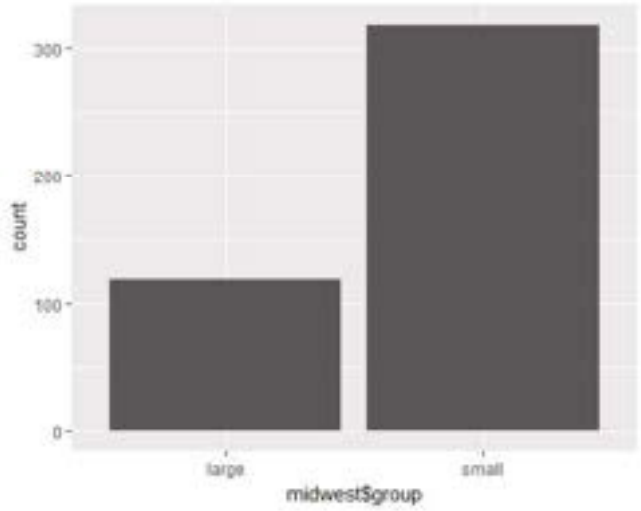
```
midwest$group <- ifelse(midwest$ratio > mean(midwest$ratio), "large", "small")
```

문제5. `"large"`와 `"small"`에 해당하는 지역이 얼마나 되는지, 빈도표와 빈도 막대 그래프를 만들어 확인해 보세요.

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

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

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



15-1. R 내장 함수로 데이터 추출하기

행 번호로 행 추출하기

데이터 준비하기

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

행 번호로 행 추출하기

대괄호안 쉼표 기준, 왼쪽에 행 번호(인덱스) 입력

- 인덱스(Index) : 데이터의 위치 또는 순서를 의미하는 값
- 인덱싱(Indexing) : 인덱스를 이용해 데이터를 추출하는 작업

```
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   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
```

```
exam[1,]  # 1행 추출

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

exam[2,]  # 2행 추출

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

여러 행 한꺼번에 추출하기

```
exam[c(1:4),]  # 1~4 행 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

exam[sample(1:10, 4),]  # 랜덤 행추출
##    id class math english science
## 7   7     2   80      90      45
## 8   8     2   90      78      25
## 5   5     2   25      80      65
## 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   45      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
## 8   8     2   90      78      25
## 18 18     5   80      78      90
## 19 19     5   89      68      87
```

[참고] 조건 벡터 (TRUE/FALSE) 에 의한 추출은 행 번호를 지정하는 것보다 처리가 느림

[참고] which(조건) 을 통해 조건 벡터의 인덱스 벡터를 얻을 수 있음

```r
# 1반 이면서 수학점수가 50점 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
```

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

```
##    id class math english science
## 3   3     1   45      86      78
## 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
## 14 14     4   48      87      12
## 15 15     4   75      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

대괄호안 쉼표 오른쪽에 조건을 입력

```r
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
```

```r
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
```

```r
exam[,3]  # 세 번째 열 추출
```

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

[질문] 열 번호로 변수를 추출할 때의 문제점은? (HINT: 데이터 프레임의 열 배치 순서는 고정적일까?)

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 45 30 25 50 80 90 20 50 65 45 46 48 75 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   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
```

행, 변수 동시 추출하기

```
# 행, 변수 모두 인덱스
exam[1,3]
```

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

```
# 행 인덱스, 열 변수명
exam[5, "english"]
```

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

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

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

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

```
##    english science
## 1       98      50
## 2       97      60
## 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
```

15-3. 데이터 구조 (심화)

- 데이터 프레임 외에도 다양한 데이터 구조가 있음
- 데이터 구조에 따라 활용 방법 다름

데이터 구조	차원	특징
벡터(Vactor)	1차원	한 가지 변수 타입으로 구성
데이터 프레임(Data Frame)	2차원	다양한 변수 타입으로 구성
매트릭스(Matrix)	2차원	한 가지 변수 타입으로 구성
어레이(Array)	다차원	2차원 이상의 매트릭스
리스트(List)	다차원	서로 다른 데이터 구조 포함

데이터 구조 비교하기

1. 벡터(Vactor)

- 하나 또는 여러 개의 값으로 구성된 데이터 구조
- 여러 타입을 섞을 수 없고, 한 가지 타입으로만 구성 가능

```
# 벡터 만들기
a <- 1
a

## [1] 1

b <- "hello"
b

## [1] "hello"

# 데이터 구조 확인
class(a)

## [1] "numeric"

class(b)

## [1] "character"
```

2. 데이터 프레임(Data Frame)

- 행과 열로 구성된 2차원 데이터 구조
- 다양한 변수 타입으로 구성 가능

```
# 데이터 프레임 만들기
x1 <- data.frame(var1 = c(1,2,3),
                 var2 = c("a","b","c"))
x1
```

```
##   var1 var2
## 1    1    a
## 2    2    b
## 3    3    c
```

```
# 데이터 구조 확인
class(x1)
```

```
## [1] "data.frame"
```

3. 매트릭스(Matrix)

- 행과 열로 구성된 2차원 데이터 구조
- 한 가지 타입으로만 구성 가능

```
# 매트릭스 만들기 - 1~12로 2열
x2 <- matrix(c(1:12), ncol = 2)
x2
```

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

```
# 데이터 구조 확인
class(x2)
```

```
## [1] "matrix"
```

4. 어레이(Array)

- 2차원 이상으로 구성된 매트릭스
- 한 가지 타입으로만 구성 가능

```
# array 만들기 - 1~20 으로 2 행 x 5 열 x 2 차원
x3 <- array(1:20, dim = c(2, 5, 2))
x3
```

```
## , , 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
```

5. 리스트(List)

- 모든 데이터 구조를 포함하는 데이터 구조
- 여러 데이터 구조를 합해 하나의 리스트로 구성 가능

```
# 리스트 생성 - 앞에서 생성한 데이터 구조 활용
x4 <- list(f1 = a,    # 벡터
           f2 = x1,   # 데이터 프레임
           f3 = x2,   # 매트릭스
           f4 = x3)   # 어레이
x4
```

```
## $f1
## [1] 1
##
## $f2
##   var1 var2
## 1    1    a
## 2    2    b
## 3    3    c
##
## $f3
##      [,1] [,2]
## [1,]    1    7
## [2,]    2    8
```

```
## [3,]    3    9
## [4,]    4   10
## [5,]    5   11
## [6,]    6   12
##
## $f4
## , , 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
```

```
# 데이터 구조 확인
class(x4)
```

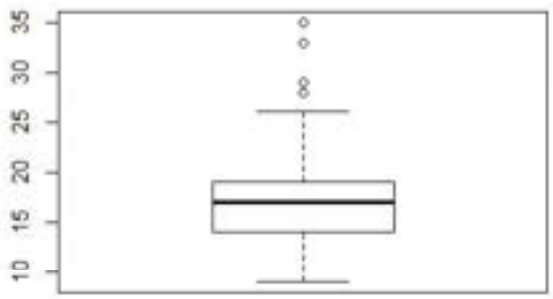
```
## [1] "list"
```

리스트 활용

- 함수의 결과물이 리스트 형태로 반환되는 경우 많음
- 리스트를 활용하면 함수의 결과물에서 특정 값을 추출 가능

boxplot() 출력 결과물에서 값 추출하기

```
mpg <- ggplot2::mpg
x <- boxplot(mpg$cty)
```



```
x
```

```
## $stats
##      [,1]
## [1,]    9
## [2,]   14
## [3,]   17
## [4,]   19
## [5,]   26
## attr(,"class")
##         1
## "integer"
##
## $n
## [1] 234
##
## $conf
##          [,1]
## [1,] 16.48356
## [2,] 17.51644
##
## $out
## [1] 28 28 33 35 29
##
## $group
## [1] 1 1 1 1 1
##
## $names
## [1] "1"
```

```
x$stats[,1]     # 요약 통계량 추출
```

```
## [1]  9 14 17 19 26
```

```
x$stats[3,1]    # 중앙값 추출
```

```
## [1] 17
```

```
x$stats[2,1]    # 1분위수 추출
```

```
## [1] 14
```

R 통계 분석 기초 – 4차시

2020-8-26 (수)
구혜민
MCM GI Convergence Lab

이번 차시 목표

데이터(행) 추출하기
변수(열) 추출하기
정렬하기
파생변수 추가하기
집단별로 요약하기
데이터 합치기

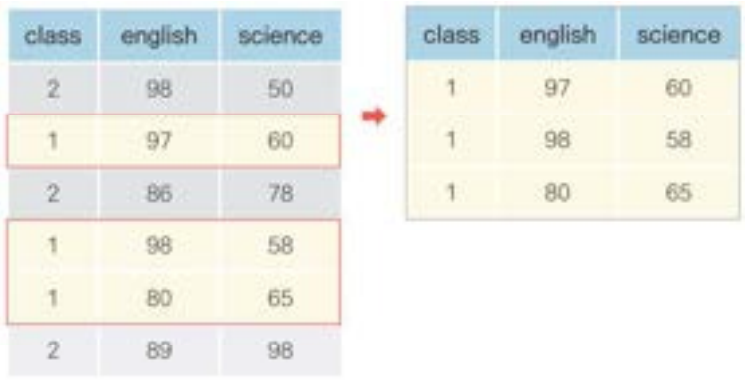
6. 자유자재로 데이터 가공하기

06-1. 데이터 전처리 - 원하는 형태로 데이터 가공하기

데이터 전처리(Preprocessing) - dplyr 패키지

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

06-2. 조건에 맞는 데이터만 추출하기



class	english	science
2	98	50
1	97	60
2	86	78
1	98	58
1	80	65
2	89	98

class	english	science
1	97	60
1	98	58
1	80	65

조건에 맞는 데이터만 추출하기

dplyr 패키지 로드 & 데이터 준비

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

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

초과, 미만, 이상, 이하 조건 걸기

```
# 수학 점수가 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
```

```
# 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
```

여러 조건을 충족하는 행 추출하기

```
# 1반 이면서 수학 점수가 50점 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
```

여러 조건 중 하나 이상 충족하는 행 추출하기

```
# 수학 점수가 90점 이상이거나 영어점수가 90점 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
```

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 해당하면 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

```
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 수학 점수 평균 구하기
```

```
## [1] 46.25
```

```
mean(class2$math)                      # 2 반 수학 점수 평균 구하기
```

```
## [1] 61.25
```

R에서 사용하는 기호들

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

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

혼자서 해보기

`mpg` 데이터를 이용해 분석 문제를 해결해 보세요.

- Q1. 자동차 배기량에 따라 고속도로 연비가 다른지 알아보려고 합니다. `displ`(배기량)이 `4` 이하인 자동차와 `5` 이상인 자동차 중 어떤 자동차의 `hwy`(고속도로 연비)가 평균적으로 더 높은지 알아보세요.
- Q2. 자동차 제조 회사에 따라 도시 연비가 다른지 알아보려고 합니다. `"audi"`와 `"toyota"` 중 어느 `manufacturer`(자동차 제조 회사)의 `cty`(도시 연비)가 평균적으로 더 높은지 알아보세요.
- Q3. `"chevrolet"`, `"ford"`, `"honda"` 자동차의 고속도로 연비 평균을 알아보려고 합니다. 이 회사들의 자동차를 추출한 뒤 `hwy` 전체 평균을 구해보세요.

힌트

Q1. 특정 조건에 해당하는 데이터를 추출해서 평균을 구하면 해결할 수 있는 문제입니다. `filter()`를 이용해 `displ` 변수가 특정 값을 지닌 행을 추출해 새로운 변수에 할당한 다음 평균을 구해보세요.

Q2. 앞 문제와 동일한 절차로 해결하면 됩니다. 단, 변수의 값이 숫자가 아니라 문자라는 점이 다릅니다.

Q3. '여러 조건 중 하나 이상 충족'하면 추출하도록 `filter()` 함수를 구성해보세요. 이렇게 추출한 데이터로 평균을 구하면 됩니다. `%in%`를 이용하면 코드를 짧게 만들 수 있습니다.

정답

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

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

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
```

Q2. 자동차 제조 회사에 따라 도시 연비가 다른지 알아보려고 합니다. `"audi"`와 `"toyota"` 중 어느 `manufacturer`(자동차 제조 회사)의 `cty`(도시 연비)가 평균적으로 더 높은지 알아보세요.

```
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
```

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

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

## [1] 22.50943
```

06-3. 필요한 변수만 추출하기

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

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

```
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
```

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
```

dplyr 함수 조합하기

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

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

가독성 있게 줄 바꾸기

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

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
```

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
```

혼자서 해보기

`mpg` 데이터를 이용해서 분석 문제를 해결해보세요.

- Q1. `mpg` 데이터는 11 개 변수로 구성되어 있습니다. 이 중 일부만 추출해서 분석에 활용하려고 합니다. `mpg` 데이터에서 `class`(자동차 종류), `cty`(도시 연비) 변수를 추출해 새로운 데이터를 만드세요. 새로 만든 데이터의 일부를 출력해서 두 변수로만 구성되어 있는지 확인하세요.
- Q2. 자동차 종류에 따라 도시 연비가 다른지 알아보려고 합니다. 앞에서 추출한 데이터를 이용해서 `class`(자동차 종류)가 `"suv"`인 자동차와 `"compact"`인 자동차 중 어떤 자동차의 `cty`(도시 연비)가 더 높은지 알아보세요.

힌트

Q1. `select()`로 변수를 추출해서 새로운 데이터를 만들어 보세요.

Q2. `filter()`로 조건에 해당하는 데이터를 추출한 뒤 평균을 구하면 해결할 수 있습니다.

정답

Q1. `mpg` 데이터는 11개 변수로 구성되어 있습니다. 이 중 일부만 추출해서 분석에 활용하려고 합니다. `mpg` 데이터에서 `class`(자동차 종류), `cty`(도시 연비) 변수를 추출해 새로운 데이터를 만드세요. 새로 만든 데이터의 일부를 출력해서 두 변수로만 구성되어 있는지 확인하세요.

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

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
```

Q2. 자동차 종류에 따라 도시 연비가 다른지 알아보려고 합니다. 앞에서 추출한 데이터를 이용해서 `class`(자동차 종류)가 `"suv"`인 자동차와 `"compact"`인 자동차 중 어떤 자동차의 `cty`(도시 연비)가 더 높은지 알아보세요.

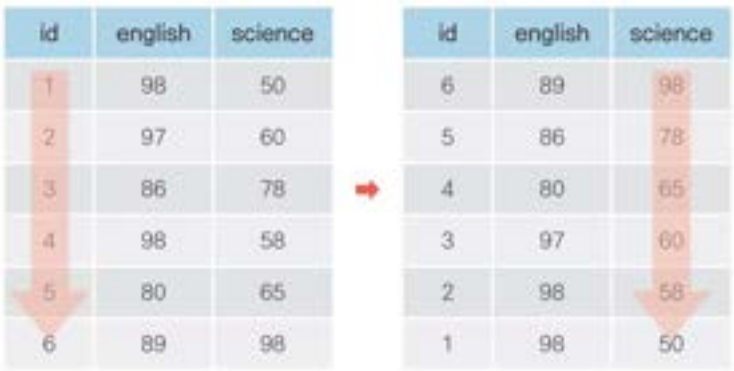
```
df_suv <- df %>% filter(class == "suv")          # class 가 suv 인 행 추출
df_compact <- df %>% filter(class == "compact")  # class 가 compact 인 행 추출

mean(df_suv$cty)                                 # suv 의 cty 평균
```

```
## [1] 13.5
```

```
mean(df_compact$cty)                             # compact 의 cty 평균
```

```
## [1] 20.12766
```

06-4. 순서대로 정렬하기



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

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

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
```

혼자서 해보기

`mpg` 데이터를 이용해서 분석 문제를 해결해보세요.

- `"audi"`에서 생산한 자동차 중에 어떤 자동차 모델의 `hwy`(고속도로 연비)가 높은지 알아보려고 합니다. `"audi"`에서 생산한 자동차 중 `hwy`가 1~5위에 해당하는 자동차의 데이터를 출력하세요.

힌트

`filter()`를 이용해 `"audi"`에서 생산한 자동차만 추출하고, `arrange()`로 `hwy`를 내림차순 정렬하면 됩니다. `head()`를 이용하면 이 중 특정 순위에 해당하는 자동차만 출력할 수 있습니다.

정답

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

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
```

06-5. 파생변수 추가하기

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

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

```r
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 한 번에 추가하기

```r
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
```

`mutate()`에 `ifelse()` 적용하기

```
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 변수를 `dplyr` 코드에 바로 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
```

혼자서 해보기

`mpg` 데이터를 이용해서 분석 문제를 해결해보세요.

`mpg` 데이터는 연비를 나타내는 변수가 `hwy`(고속도로 연비), `cty`(도시 연비) 두 종류로 분리되어 있습니다. 두 변수를 각각 활용하는 대신 하나의 통합 연비 변수를 만들어 분석하려고 합니다.

- Q1. `mpg` 데이터 복사본을 만들고, `cty` 와 `hwy` 를 더한 '합산 연비 변수'를 추가하세요.
- Q2. 앞에서 만든 '합산 연비 변수'를 2 로 나눠 '평균 연비 변수'를 추가세요.
- Q3. '평균 연비 변수'가 가장 높은 자동차 3 종의 데이터를 출력하세요.
- Q4. 1~3 번 문제를 해결할 수 있는 하나로 연결된 `dplyr` 구문을 만들어 출력하세요. 데이터는 복사본 대신 `mpg` 원본을 이용하세요.

힌트

Q1. `mutate()`를 적용한 결과를 `<-`를 이용해 데이터 프레임에 할당하는 형태로 코드를 작성하면 기존 데이터 프레임에 변수가 추가됩니다.

Q3. `arrange()`와 `head()`를 조합하면 됩니다.

Q4. 앞에서 만든 코드들을 `%>%`를 이용해 연결하면 됩니다. 변수를 추가하는 작업을 하나의 `mutate()` 구성하면 코드를 더 간결하게 만들 수 있습니다.

정답

Q1. `mpg` 데이터 복사본을 만들고, `cty`와 `hwy`를 더한 '합산 연비 변수'를 추가하세요.

```
mpg <- as.data.frame(ggplot2::mpg)                # mpg 데이터 불러오기
mpg_new <- mpg                                    # 복사본 만들기

mpg_new <- mpg_new %>% mutate(total = cty + hwy)  # 합산 변수 만들기
```

Q2. 앞에서 만든 '합산 연비 변수'를 2로 나눠 '평균 연비 변수'를 추가세요.

```
mpg_new <- mpg_new %>% mutate(mean = total/2)     # 평균 변수 만들기
```

Q3. '평균 연비 변수'가 가장 높은 자동차 3종의 데이터를 출력하세요.

```
mpg_new %>%
  arrange(desc(mean)) %>%  # 내림차순 정렬
  head(3)                  # 상위 3 행 출력
```

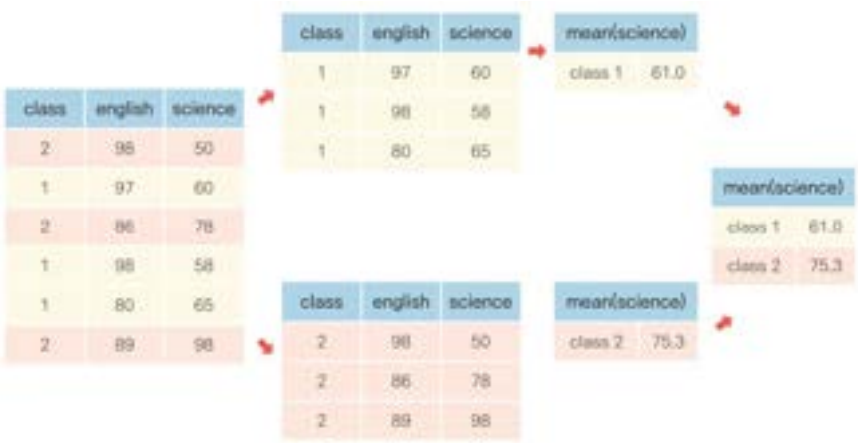
```
##   manufacturer      model displ year cyl      trans drv cty hwy fl
## 1   volkswagen new beetle   1.9 1999   4 manual(m5)   f  35  44  d
## 2   volkswagen      jetta   1.9 1999   4 manual(m5)   f  33  44  d
## 3   volkswagen new beetle   1.9 1999   4   auto(l4)   f  29  41  d
##        class total mean
## 1 subcompact    79 39.5
## 2    compact    77 38.5
## 3 subcompact    70 35.0
```

Q4. 1~3번 문제를 해결할 수 있는 하나로 연결된 `dplyr` 구문을 만들어 출력하세요. 데이터는 복사본 대신 `mpg` 원본을 이용하세요.

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

```
##   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 mean
## 1 subcompact    79 39.5
## 2    compact    77 38.5
## 3 subcompact    70 35.0
```

06-6. 집단별로 요약하기



집단별로 요약하기

통계치 산출하기

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

##   mean math
## 1     57.45
```

집단별로 통계치 산출하기

```
exam %>%
  group_by(class) %>%                # class 별로 분리
  summarize(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 산출하기

```
exam %>%
  group_by(class) %>%                    # class 별로 분리
  summarize(mean_math = mean(math),      # math 평균
            sum_math = sum(math),        # math 합계
            median_math = median(math), # math 중앙값
            n = n())                     # 학생 수
```

```
## # 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()	합계
median()	중앙값
min()	최솟값
max()	최댓값
n()	빈도

각 집단별로 부집단 나누기

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

```
## # A tibble: 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,/ 1~5위 출력하기

분석 절차 생각해보기

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

dplyr 조합하기

```
mpg %>%
  group_by(manufacturer) %>%           # 회사별로 분리
  filter(class == "suv") %>%           # suv 추출
  mutate(tot = (cty+hwy)/2) %>%        # 통합 연비 변수 생성
  summarize(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
```

혼자서 해보기

`mpg` 데이터를 이용해서 분석 문제를 해결해 보세요.

- Q1. `mpg` 데이터의 `class` 는 `"suv"`, `"compact"` 등 자동차를 특징에 따라 일곱 종류로 분류한 변수입니다. 어떤 차종의 연비가 높은지 비교해보려고 합니다. `class` 별 `cty` 평균을 구해보세요.
- Q2. 앞 문제의 출력 결과는 `class` 값 알파벳 순으로 정렬되어 있습니다. 어떤 차종의 도시 연비가 높은지 쉽게 알아볼 수 있도록 `cty` 평균이 높은 순으로 정렬해 출력하세요.
- Q3. 어떤 회사 자동차의 `hwy`(고속도로 연비)가 가장 높은지 알아보려고 합니다. `hwy` 평균이 가장 높은 회사 세 곳을 출력하세요.
- Q4. 어떤 회사에서 `"compact"`(경차) 차종을 가장 많이 생산하는지 알아보려고 합니다. 각 회사별 `"compact"` 차종 수를 내림차순으로 정렬해 출력하세요.

힌트

Q1. `group_by()`를 이용해 `class` 별로 나눈 뒤 `summarize()`를 이용해 `cty` 평균을 구하면 됩니다.

Q2. 앞에서 만든 코드를 `%>%`로 연결하고 내림차순으로 정렬하는 코드를 추가하면 됩니다.

Q3. 2번 문제와 같은 절차로 코드를 구성하고, 일부만 출력하도록 `head()`를 추가하면 됩니다.

Q4. `filter()`를 이용해 `"compact"` 차종만 남긴 후 회사별 자동차 수를 구하면 됩니다. 자동차 수는 데이터가 몇 행으로 구성되는지 빈도를 구하면 알 수 있습니다. 빈도는 `n()`을 이용해 구할 수 있습니다.

정답

Q1. `mpg` 데이터의 `class`는 `"suv"`, `"compact"` 등 자동차를 특징에 따라 일곱 종류로 분류한 변수입니다. 어떤 차종의 연비가 높은지 비교해보려고 합니다. `class`별 `cty` 평균을 구해보세요.

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

mpg %>%
  group_by(class) %>%               # class 별 분리
  summarize(mean_cty = mean(cty))   # cty 평균 구하기
```

```
## # A tibble: 7 x 2
##        class mean_cty
##        <chr>    <dbl>
## 1    2seater 15.40000
## 2    compact 20.12766
## 3    midsize 18.75610
## 4    minivan 15.81818
## 5     pickup 13.00000
## 6 subcompact 20.37143
## 7        suv 13.50000
```

Q2. 앞 문제의 출력 결과는 `class` 값 알파벳 순으로 정렬되어 있습니다. 어떤 차종의 도시 연비가 높은지 쉽게 알아볼 수 있도록 `cty` 평균이 높은 순으로 정렬해 출력하세요.

```
mpg %>%
  group_by(class) %>%                  # class 별 분리
  summarize(mean_cty = mean(cty)) %>%  # cty 평균 구하기
  arrange(desc(mean_cty))              # 내림차순 정렬하기
```

```
## # A tibble: 7 x 2
##        class mean_cty
##        <chr>    <dbl>
## 1 subcompact 20.37143
## 2    compact 20.12766
## 3    midsize 18.75610
## 4    minivan 15.81818
## 5    2seater 15.40000
## 6        suv 13.50000
## 7     pickup 13.00000
```

Q3. 어떤 회사 자동차의 `hwy`(고속도로 연비)가 가장 높은지 알아보려고 합니다. `hwy` 평균이 가장 높은 회사 세 곳을 출력하세요.

```
mpg %>%
  group_by(manufacturer) %>%            # manufacturer 별 분리
  summarize(mean_hwy = mean(hwy)) %>%   # hwy 평균 구하기
  arrange(desc(mean_hwy)) %>%           # 내림차순 정렬하기
  head(3)                               # 상위 3 행 출력
```

```
## # A tibble: 3 x 2
##   manufacturer mean_hwy
##          <chr>    <dbl>
## 1        honda 32.55556
## 2   volkswagen 29.22222
## 3      hyundai 26.85714
```

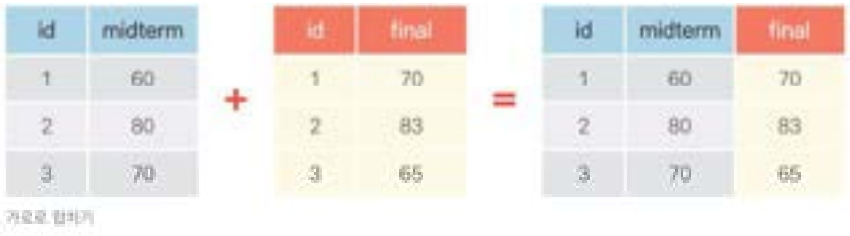
Q4. 어떤 회사에서 `"compact"`(경차) 차종을 가장 많이 생산하는지 알아보려고 합니다. 각 회사별 `"compact"` 차종 수를 내림차순으로 정렬해 출력하세요.

```
mpg %>%
  filter(class == "compact") %>%  # compact 추출
  group_by(manufacturer) %>%      # manufacturer 별 분리
  summarize(count = n()) %>%      # 빈도 구하기
  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
```

06-7. 데이터 합치기

가로로 합치기



id	midterm
1	60
2	80
3	70

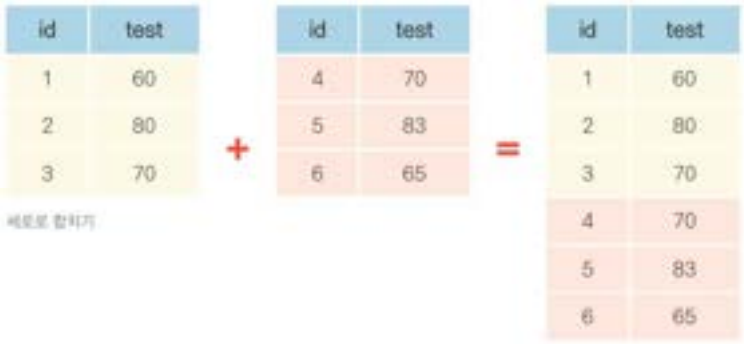
+

id	final
1	70
2	83
3	65

=

id	midterm	final
1	60	70
2	80	83
3	70	65

세로로 합치기



id	test
1	60
2	80
3	70

+

id	test
4	70
5	83
6	65

=

id	test
1	60
2	80
3	70
4	70
5	83
6	65

가로로 합치기

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 합쳐 total에 할당
total                                        # total 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

세로로 합치기

데이터 생성

```
# 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))
group_a  # group_a 출력
```

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

```
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
```

혼자서 해보기

`mpg` 데이터를 이용해서 분석 문제를 해결해 보세요.

`mpg` 데이터의 `fl` 변수는 자동차에 사용하는 연료(fuel)를 의미합니다. 아래는 자동차 연료별 가격을 나타낸 표입니다.

fl	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

우선 이 정보를 이용해서 연료와 가격으로 구성된 데이터 프레임을 만들어 보세요.

```
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  # 출력
```

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

- Q1. `mpg` 데이터에는 연료 종류를 나타낸 `fl` 변수는 있지만 연료 가격을 나타낸 변수는 없습니다. 위에서 만든 `fuel` 데이터를 이용해서 `mpg` 데이터에 `price_fl`(연료 가격) 변수를 추가하세요.
- Q2. 연료 가격 변수가 잘 추가됐는지 확인하기 위해서 `model`, `fl`, `price_fl` 변수를 추출해 앞부분 5 행을 출력해 보세요.

힌트

Q1. `left_join()`을 이용해서 `mpg` 데이터에 `fuel` 데이터를 합치면 됩니다. 두 데이터에 공통으로 들어있는 변수를 기준으로 삼아야 합니다.

Q2. `select()`와 `head()`를 조합하면 됩니다.

정답

Q1. `mpg` 데이터에는 연료 종류를 나타낸 `fl` 변수는 있지만 연료 가격을 나타낸 변수는 없습니다. 위에서 만든 `fuel` 데이터를 이용해서 `mpg` 데이터에 `price_fl`(연료 가격) 변수를 추가하세요.

```
mpg <- as.data.frame(ggplot2::mpg)      # mpg 데이터 불러오기
mpg <- left_join(mpg, fuel, by = "fl")  # mpg에 연료 가격 변수 추가
```

Q2. 연료 가격 변수가 잘 추가됐는지 확인하기 위해서 `model`, `fl`, `price_fl` 변수를 추출해 앞부분 5행을 출력해 보세요.

```
mpg %>%
  select(model, fl, price_fl) %>%       # model, fl, price_fl 추출
  head(5)
```

```
##   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
```

R 통계 분석 기초 – 5차시

2020-9-2 (수)

구혜민

MCM GI Convergence Lab

이번 차시 목표

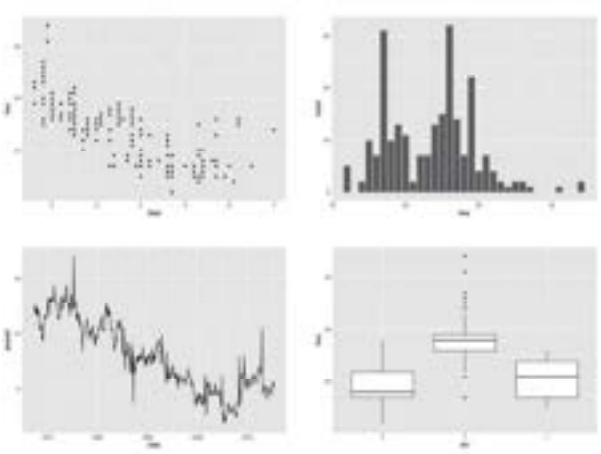
그래프 만들기

- ggplot2 패키지 vs R 내장 그래픽 함수

R로 기초통계 분석하기 1

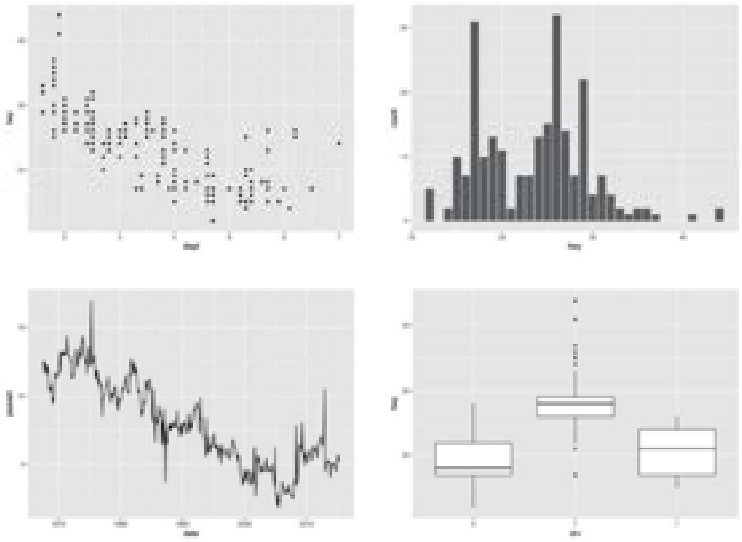
- 가설검정
- 상관분석

08. 그래프 만들기

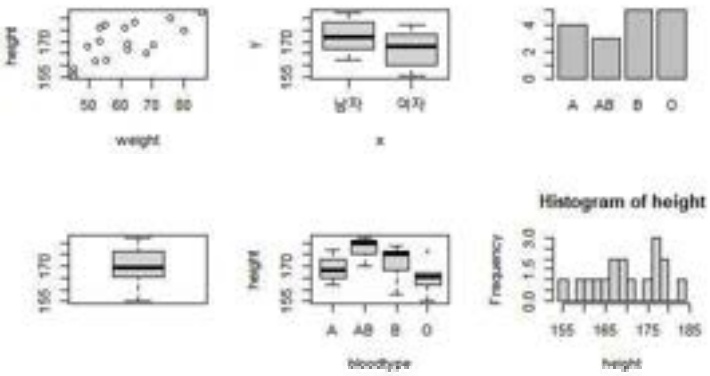


08-1. R로 만들 수 있는 그래프 살펴보기

완성도 높은 그래프를 만들 수 있는 ggplot2 패키지



빠르게 데이터를 탐색하기 위한 R 내장 그래픽 함수



R 내장 그래픽 함수 이용하기

- 빠르게 데이터를 탐색하는 용도로 사용
- 고수준(high-level) 함수: 완성된 플롯 또는 초기 플롯을 제공
  - 예) plot(), barplot(), hist(), boxplot() 등
  - plot() : 어떤 형식의 변수를 넣어도 그에 맞춰 알아서 그래프를 그려주는 generic 함수
  - 공통 옵션: add(=F), type="l/c/p/b/o/h/s", main, sub, xlab, ylab, xlim, ylim 등
  - 선택적 옵션: lty=0~6, lwd(=1), pch=0~25, cex, col, bg, tck(±,0) 등
- 저수준(low-level) 함수: 고수준 함수로 작성된 플롯에 내용을 추가
  - 예) points(), lines(), abline(), text(), axis(), title(), grid(), legend() 등

ggplot2 레이어 구조 이해하기

ggplot2 레이어 구조

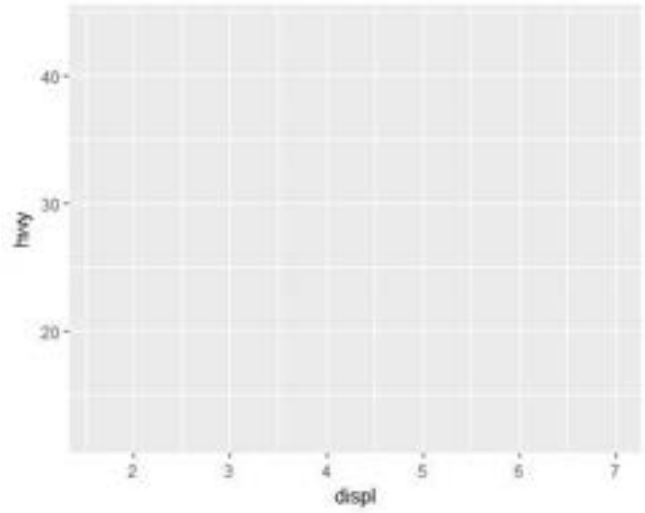
ggplot 함수 구조

ggplot2 로드

```
library(ggplot2)
```

1. 배경 설정하기

```
# x 축 displ, y 축 hwy 로 지정해 배경 생성
ggplot(data = mpg, aes(x = displ, y = hwy))
```



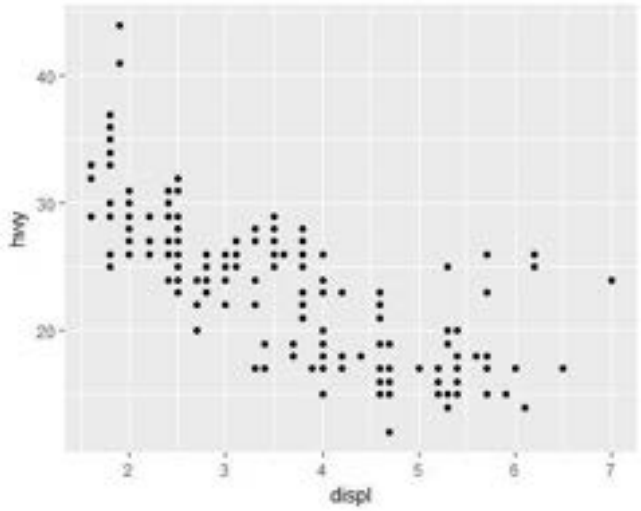
08-2. 산점도 - 변수 간 관계 표현하기

산점도 만들기

- 산점도(Scatter Plot) : 데이터를 x축과 y축에 점으로 표현한 그래프
- 나이와 소득처럼, 연속 값으로 된 두 변수의 관계를 표현할 때 사용

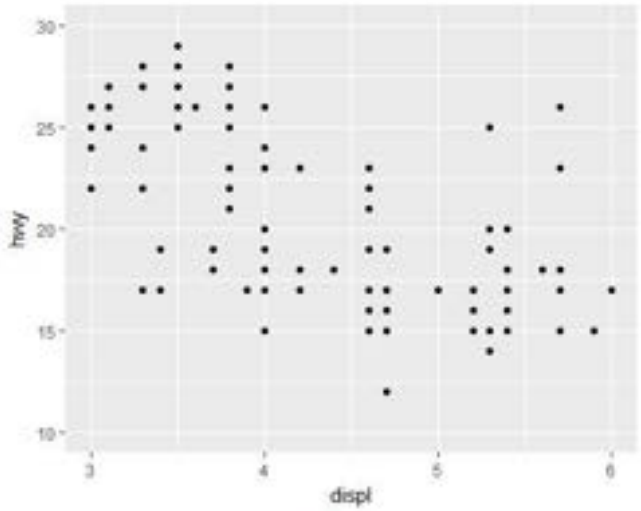
2. 그래프 추가하기

```
# 배경에 산점도 추가
ggplot(data = mpg, aes(x = displ, y = hwy)) + geom_point()
```



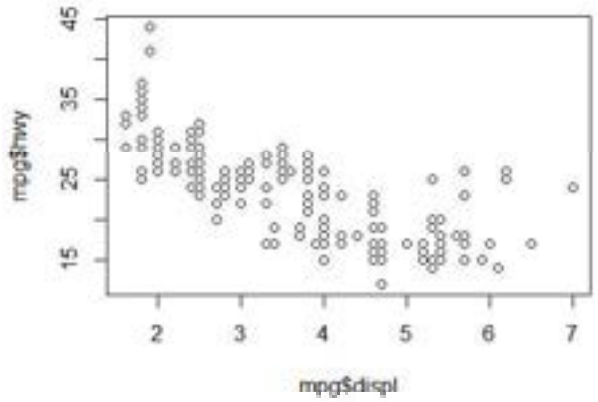
3. 축 범위를 조정하는 설정 추가하기

```
# x 축 범위 3~6, y 축 범위 10~30 으로 지정
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  xlim(3, 6) +
  ylim(10, 30)
```



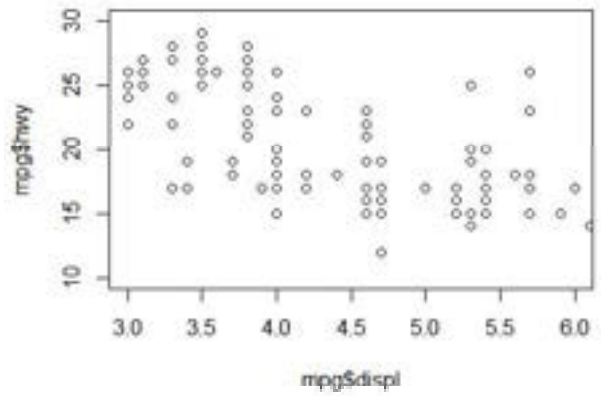
1'. plot()으로 산점도 그래프 그리기

```
plot(mpg$displ, mpg$hwy) # plot(mpg$hwy ~ mpg$displ)
```



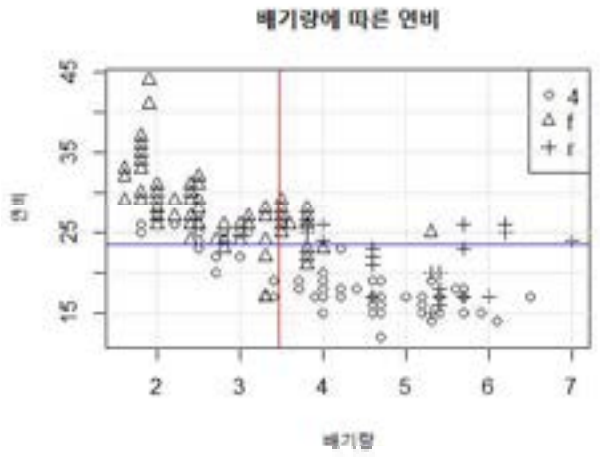
2'. 축 범위를 조정하는 설정 추가하기

```
plot(mpg$displ, mpg$hwy, xlim = c(3, 6), ylim = c(10, 30))
```



3'. plot()에 옵션 추가하기

```
# Levels 별 그래프 보기
mpg$drv <- as.factor(mpg$drv)
plot(mpg$displ, mpg$hwy, pch = as.integer(mpg$drv), ann = F)
legend("topright", levels(mpg$drv), pch = c(1:3))  # Legend 추가

# title 추가
title(main = "배기량에 따른 연비")
title(xlab = "배기량")
title(ylab = "연비")

grid()  # grid 추가

# 수직선 or 수평선 추가
displMean = mean(mpg$displ)
abline(v = displMean, col = "red")
hwyMean = mean(mpg$hwy)
abline(h = hwyMean, col = "blue")
```



08-3. 막대 그래프 - 집단 간 차이 표현하기

- 막대 그래프(Bar Chart) : 데이터의 크기를 막대의 길이로 표현한 그래프
- 성별 소득 차이처럼 집단 간 차이를 표현할 때 주로 사용

막대 그래프 1 - 평균 막대 그래프 만들기

- 각 집단의 평균값을 막대 길이로 표현한 그래프

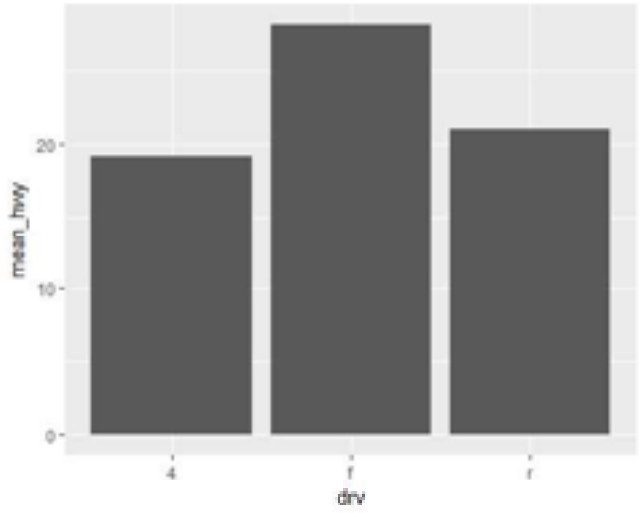
1. 집단별 평균표 만들기

```
library(dplyr)

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
```

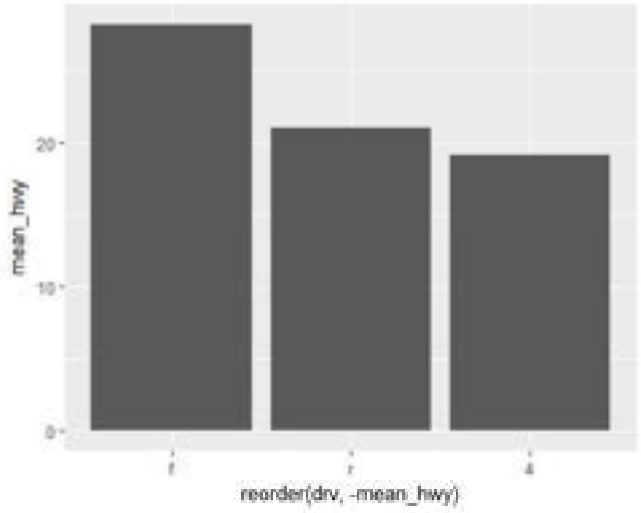
2. 그래프 생성하기

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



3. 크기 순으로 정렬하기

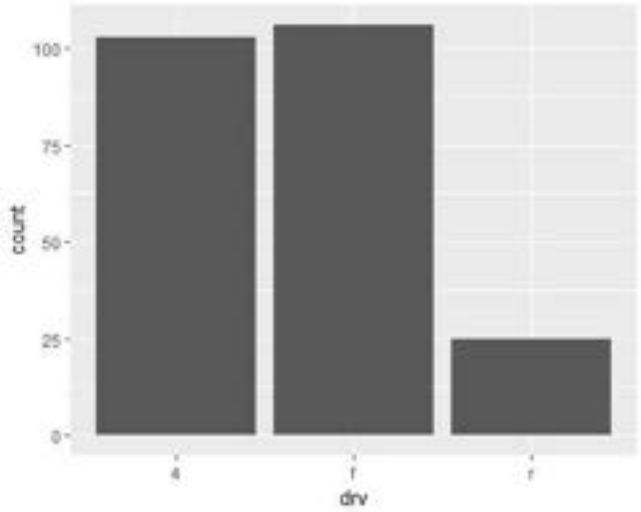
```
ggplot(data = df_mpg, aes(x = reorder(drv, -mean_hwy), y = mean_hwy)) + geom_col()
```



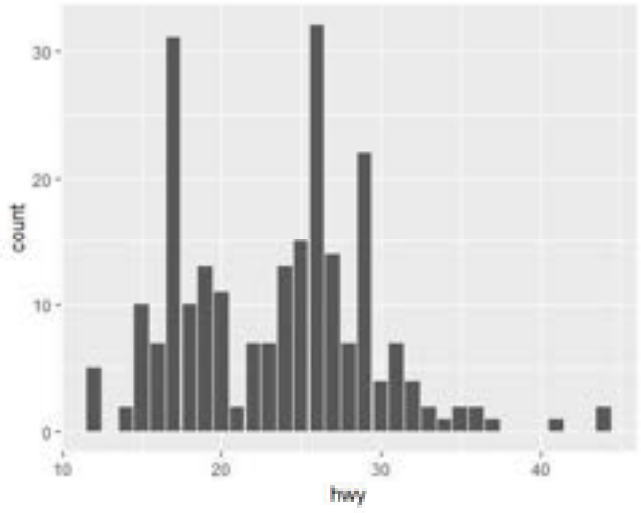
막대 그래프 2 - 빈도 막대 그래프

- 값의 개수(빈도)로 막대의 길이를 표현한 그래프

```
# x축 범주 변수, y축 빈도
ggplot(data = mpg, aes(x = drv)) + geom_bar()
```

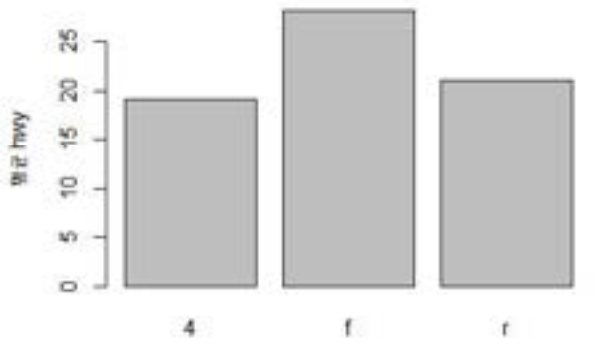



```
# x축 연속 변수, y축 빈도
ggplot(data = mpg, aes(x = hwy)) + geom_bar()
```



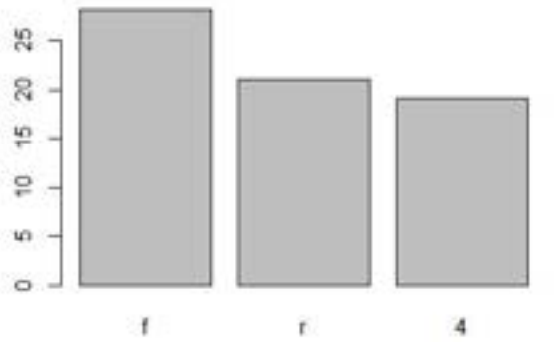
2'. barplot()으로 평균 막대 그래프 그리기

```
barplot(df_mpg$mean_hwy, names.arg = df_mpg$drv)
title(ylab = "평균 hwy")
```



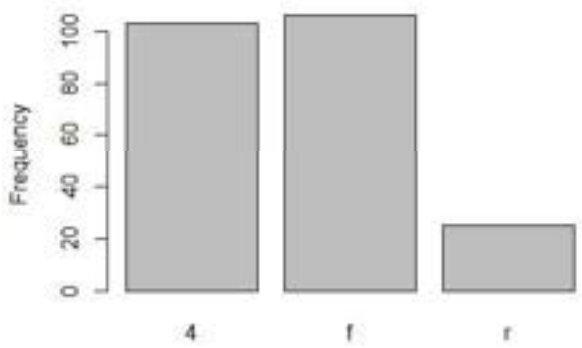
3'. 크기 순으로 정렬하기

```
arr_df_mpg =  arrange(df_mpg, desc(mean_hwy)) # dplyr 함수
barplot(arr_df_mpg$mean_hwy, names.arg = arr_df_mpg$drv)
```



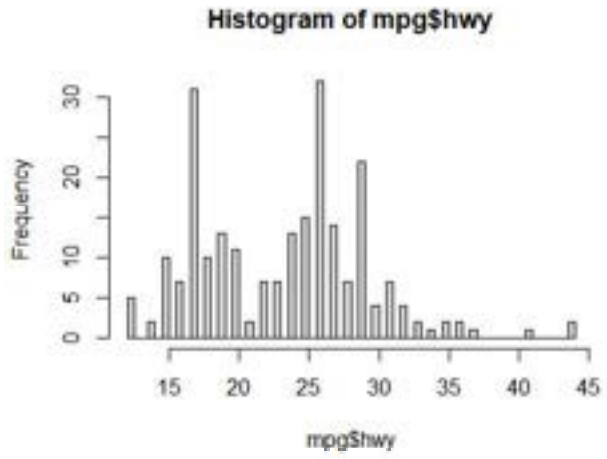
barplot()으로 빈도 막대 그래프 그리기

```
# x 축 범주 변수, y 축 빈도
freqDrv = table(mpg$drv)
barplot(freqDrv)
title(ylab = "Frequency")
```



hist()로 빈도 막대 그래프 그리기

```
# x 축 연속 변수, y 축 빈도
hist(mpg$hwy, breaks = 50)
```



geom_col() vs geom_bar()

- 평균 막대 그래프 : 데이터를 요약한 평균표를 먼저 만든 후 평균표를 이용해 그래프 생성 - geom_col()
- 빈도 막대 그래프 : 별도로 표를 만들지 않고 원자료를 이용해 바로 그래프 생성 - geom_bar()

barplot() vs hist()

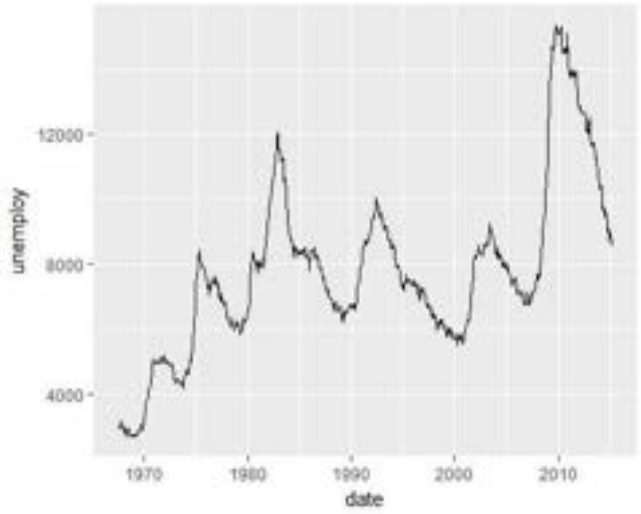
- x축 범주 변수 - barplot()
- x축 연속 변수 - hist()

08-4. 선 그래프 - 시간에 따라 달라지는 데이터 표현하기

- 선 그래프(Line Chart) : 데이터를 선으로 표현한 그래프
- 시계열 그래프(Time Series Chart) : 일정 시간 간격을 두고 나열된 시계열 데이터(Time Series Data)를 선으로 표현한 그래프. 환율, 주가지수 등 경제 지표가 시간에 따라 어떻게 변하는지 표현할 때 활용

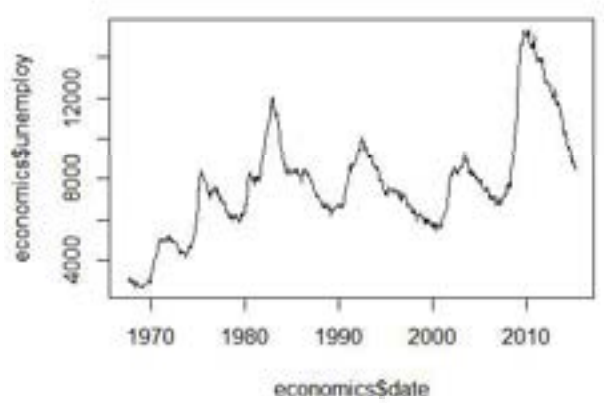
시계열 그래프 만들기

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



plot()으로 시계열 그래프 그리기

```
plot(economics$date, economics$unemploy, type = "l")
```

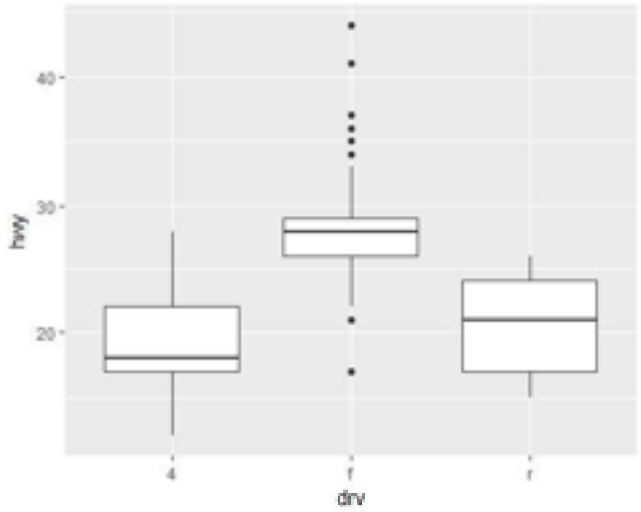


08-5. 상자 그림 - 집단 간 분포 차이 표현하기

- 상자 그림(Box Plot) : 데이터의 분포(퍼져 있는 형태)를 직사각형 상자 모양으로 표현한 그래프
- 분포를 알 수 있기 때문에 평균만 볼 때보다 데이터의 특성을 좀 더 자세히 이해할 수 있음

상자 그림 만들기

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

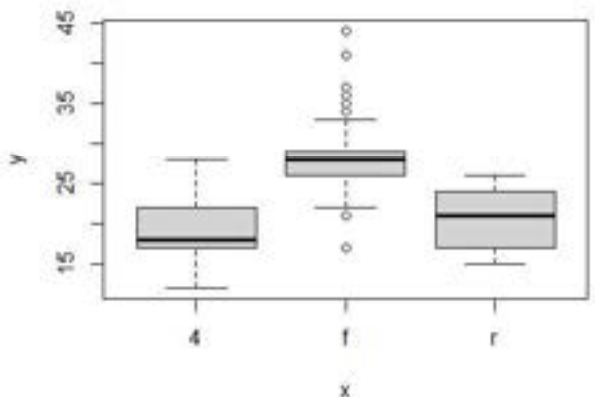


상자 그림	값	설명
상자 아래 세로선	아래 수염	하위 0~25% 내에 해당하는 값
상자 밑면	1사분위수(Q1)	하위 25% 위치 값
상자 내 굵은 선	2사분위수(Q2)	하위 50% 위치 값(중앙값)
상자 윗면	3사분위수(Q3)	하위 75% 위치 값
상자 위 세로선	윗수염	하위 75~100% 내에 해당하는 값
상자 밖 점 표식	극단치	Q1, Q3 밖 1.5 IQR을 벗어난 값

참고 1.5 IQR: 사분위 범위(Q1~Q3간 거리)의 1.5배

plot()으로 상자 그림 만들기

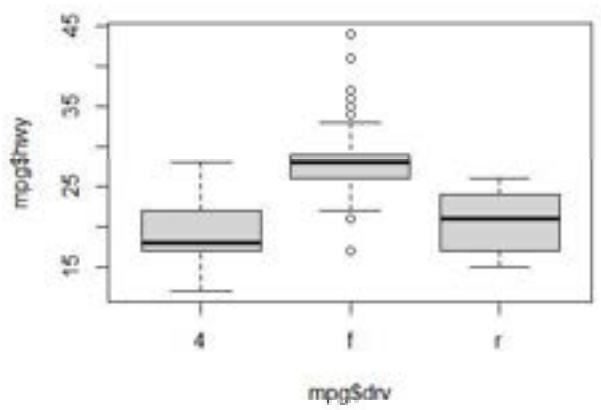
```
mpg$drv <- as.factor(mpg$drv)
plot(mpg$drv, mpg$hwy)
```



[주의] x축 변수가 factor형이 아니라면 as.factor()을 사용해서 factor형으로 변환해야 한다.

boxplot()으로 상자 그림 만들기

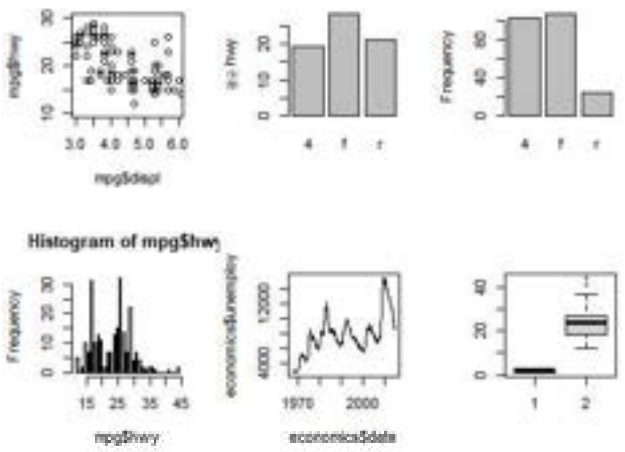
```
boxplot(mpg$hwy ~ mpg$drv)
```



[유의] x축 변수가 character형이더라도 내부적으로 factor형으로 변환해 준다.

한 화면에 여러 그래프 그리기(R 내장 그래픽 전용)

```
par(mfrow = c(2, 3))

plot(mpg$displ, mpg$hwy, xlim=c(3, 6), ylim=c(10, 30))
barplot(df_mpg$mean_hwy, names.arg = df_mpg$drv)
title(ylab = "평균 hwy")
barplot(freqDrv)
title(ylab = "Frequency")
hist(mpg$hwy, breaks = 50)
plot(economics$date, economics$unemploy, type = "l")
boxplot(mpg$drv, mpg$hwy)
```



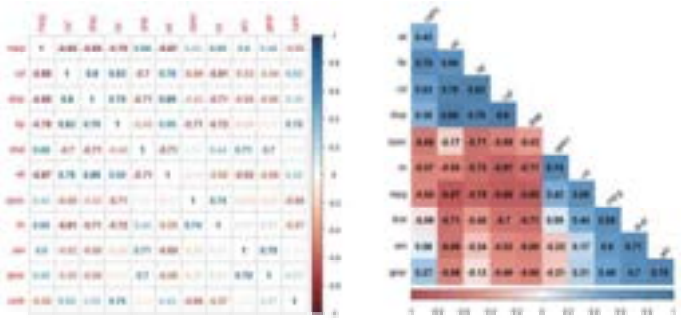
앞에서 다룬 ggplot2 함수들

값	내용
geom_point()	산점도
geom_col()	막대 그래프 - 요약표
geom_bar()	막대 그래프 - 원자료
geom_line()	선 그래프
geom_boxplot()	상자 그림

앞에서다룬 R 내장 그래픽 고수준 함수들

값	내용
plot()	generic (산점도, 선 그래프, 상자 그림)
barplot()	막대 그래프 - 범주형 x축 자료
hist()	막대 그래프 - 연속형 x축 자료
boxplot()	상자 그림

13. 통계 분석 기법을 이용한 가설 검정



R에서 제공하는 확률분포 관련 함수

- 다양한 확률분포에 대해 확률밀도(probability density), 누적확률(cumulative density), 분위수(quantile), 난수(random number)를 제공
- stats 패키지에서 제공하는 분포명 앞에 d, p, q, r 적용

```
# 표준정규분포의 밀도함수값, 누적확률, 분위수, 난수 생성 예
dnorm(0)      # 확률밀도

## [1] 0.3989423

pnorm(1.96)   # 누적확률

## [1] 0.9750021

qnorm(0.975)  # 분위수

## [1] 1.959964

rnorm(5)      # 난수 생성

## [1] -0.97303301  0.07410485  1.01483598 -0.50707063  0.45932758

set.seed(100) # 난수 발생 시 초기값을 고정하여 결과의 임의성 없앰
rnorm(5)

## [1] -0.50219235  0.13153117 -0.07891709  0.88678481  0.11697127
```

R 함수	분포명	모수
norm	정규	mean, sd
exp	지수	1/mean
gamma	감마	shape, 1/scale
pois	포아송	lambda
weibull	와이블	shape
cauchy	코시	location, scale
beta	베타	shape1, shape2
t	student's t	df
f	Fisher's F	df1, df2
chisq	카이제곱	df

R 함수	분포명	모수
binom	이항	size, prob
geom	기하	prob
hyper	초기하	m, n, k
logis	로지스틱	location, scale
lnorm	로그정규	mean, sd
nbinom	음이항	
unif	균일	min, max
wilcox	윌콕슨 순위합 통계량	
signrank	윌콕슨 부호순위 통계량	

13-1. 통계적 가설 검정이란?

기술 통계와 추론 통계

- **기술 통계(Descriptive statistics)**
  - 데이터를 요약해 설명하는 통계 기법
  - ex) 사람들이 받는 월급을 집계해 전체 월급 평균 구하기
- **추론 통계(Inferential statistics)**
  - 단순히 숫자를 요약하는 것을 넘어 어떤 값이 발생할 확률을 계산하는 통계 기법
  - ex) 수집된 데이터에서 성별에 따라 월급에 차이가 있는 것으로 나타났을 때, 이런 차이가 우연히 발생할 확률을 계산

- **추론 통계(Inferential statistics)**
  - 이런 차이가 우연히 나타날 확률이 작다
    - -> 성별에 따른 월급 차이가 통계적으로 유의하다(statistically significant)고 결론
  - 이런 차이가 우연히 나타날 확률이 크다
    - -> 성별에 따른 월급 차이가 통계적으로 유의하지 않다(not statistically significant)고 결론
  - 기술 통계 분석에서 집단 간 차이가 있는 것으로 나타났더라도 이는 우연에 의한 차이일 수 있음
    - 데이터를 이용해 신뢰할 수 있는 결론을 내리려면 유의확률을 계산하는 통계적 가설 검정 절차를 거쳐야 함

통계적 가설 검정

- **통계적 가설 검정(Statistical hypothesis test)**
  - 유의확률을 이용해 가설을 검정하는 방법
- **유의확률(Significance probability, p-value)**
  - 실제로는 집단 간 차이가 없는데 우연히 차이가 있는 데이터가 추출될 확률
  - 분석 결과 유의확률이 크게 나타났다면
    - '집단 간 차이가 통계적으로 유의하지 않다'고 해석
    - 실제로 차이가 없더라도 우연에 의해 이 정도의 차이가 관찰될 가능성이 크다는 의미
  - 분석 결과 유의확률이 작게 나타났다면
    - '집단 간 차이가 통계적으로 유의하다'고 해석
    - 실제로 차이가 없는데 우연히 이 정도의 차이가 관찰될 가능성이 작다, 우연이라고 보기 힘들다는 의미

13-3. 상관분석 - 두 변수의 관계성 분석

상관분석(Correlation Analysis)

- 두 연속 변수가 서로 선형 상관관계에 있는지 검정하는 통계 분석 기법
- 상관계수(Correlation Coefficient)
  - 두 변수가 얼마나 선형으로 관련되어 있는지, 선형 연관성의 정도를 나타내는 값
  - -1~1 사이의 값을 지니고 절댓값이 1에 가까울수록 선형 연관성이 크다는 의미
  - 상관계수가 양수면 양의 상관 관계, 음수면 음의 상관 관계

$$r = \frac{COV(X,Y)}{\sigma(X)\sigma(Y)}$$

- 가설 검정) 귀무가설 $H_0: r = 0$ 하에서 $t = \frac{\sqrt{n-2}r}{\sqrt{1-r^2}} \sim t(n-2)$

실업자 수와 개인 소비 지출의 상관관계

데이터 준비

```
economics <- as.data.frame(ggplot2::economics)
```

상관분석

```
cor.test(economics$unemploy, economics$pce)
```

```
##
##  Pearson's product-moment correlation
##
## data:  economics$unemploy and economics$pce
## t = 18.605, df = 572, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5603164 0.6625460
## sample estimates:
##       cor
## 0.6139997
```

상관행렬 히트맵 만들기

- 상관행렬(Correlation Matrix)
  - 여러 변수 간 상관계수를 행렬로 나타낸 표
  - 어떤 변수끼리 관련이 크고 작은지 파악할 수 있음

데이터 준비

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

상관행렬 만들기

```
car_cor <- cor(mtcars)  # 상관행렬 생성
round(car_cor, 2)       # 소수점 셋째 자리에서 반올림해서 출력
```

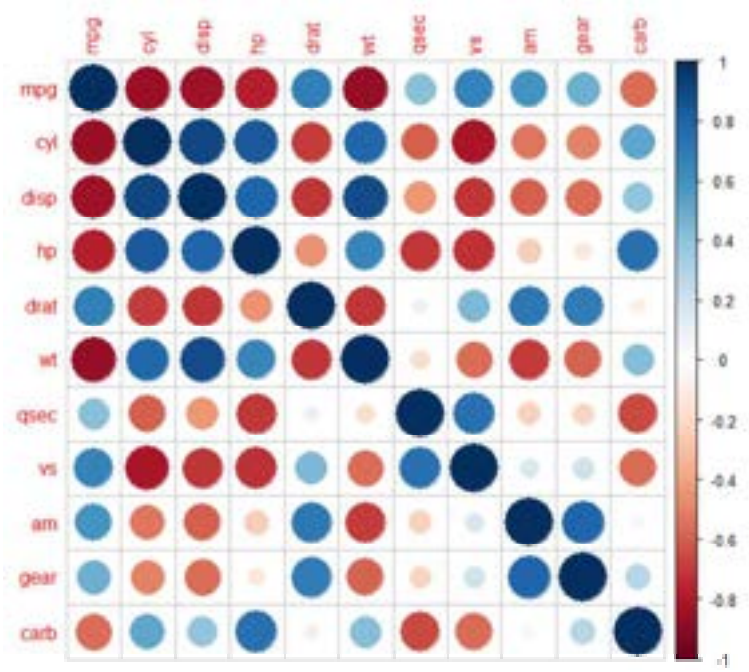
```
##        mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
## mpg   1.00 -0.85 -0.85 -0.78  0.68 -0.87  0.42  0.66  0.60  0.48 -0.55
## cyl  -0.85  1.00  0.90  0.83 -0.70  0.78 -0.59 -0.81 -0.52 -0.49  0.53
## disp -0.85  0.90  1.00  0.79 -0.71  0.89 -0.43 -0.71 -0.59 -0.56  0.39
## hp   -0.78  0.83  0.79  1.00 -0.45  0.66 -0.71 -0.72 -0.24 -0.13  0.75
## drat  0.68 -0.70 -0.71 -0.45  1.00 -0.71  0.09  0.44  0.71  0.70 -0.09
## wt   -0.87  0.78  0.89  0.66 -0.71  1.00 -0.17 -0.55 -0.69 -0.58  0.43
## qsec  0.42 -0.59 -0.43 -0.71  0.09 -0.17  1.00  0.74 -0.23 -0.21 -0.66
## vs    0.66 -0.81 -0.71 -0.72  0.44 -0.55  0.74  1.00  0.17  0.21 -0.57
## am    0.60 -0.52 -0.59 -0.24  0.71 -0.69 -0.23  0.17  1.00  0.79  0.06
## gear  0.48 -0.49 -0.56 -0.13  0.70 -0.58 -0.21  0.21  0.79  1.00  0.27
## carb -0.55  0.53  0.39  0.75 -0.09  0.43 -0.66 -0.57  0.06  0.27  1.00
```

상관행렬 히트맵 만들기

- 히트맵(heat map) : 값의 크기를 색깔로 표현한 그래프

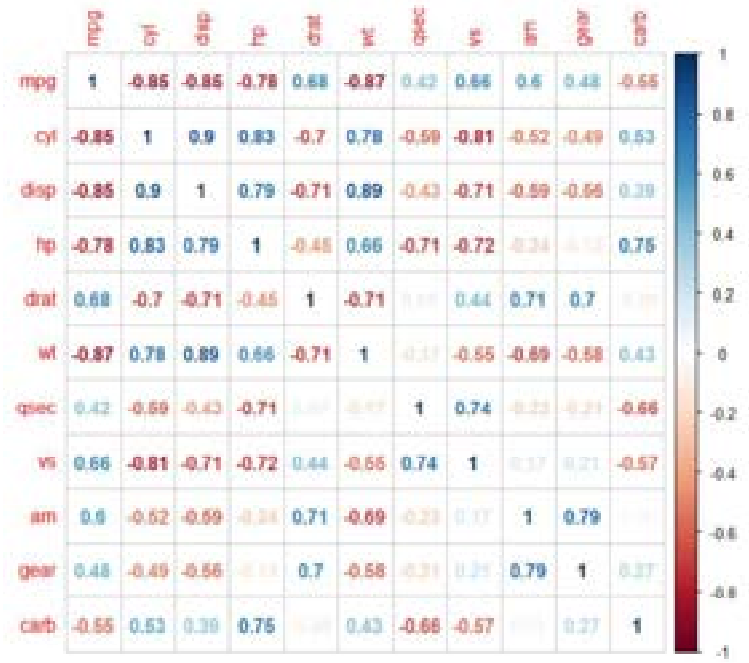
```
install.packages("corrplot")
library(corrplot)
```

```
corrplot(car_cor)
```



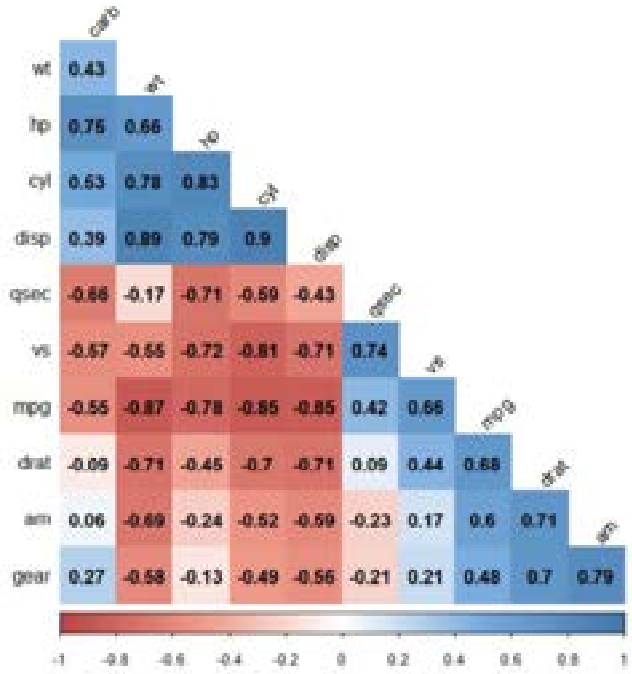
원 대신 상관계수 표시

```
corrplot(car_cor, method = "number")
```



다양한 파라미터 지정하기

```
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

corrplot(car_cor,
         method = "color",        # 색깔로 표현
         col = col(200),          # 색상 200 개 선정
         type = "lower",          # 왼쪽 아래 행렬만 표시
         order = "hclust",        # 유사한 상관계수끼리 군집화
         addCoef.col = "black",   # 상관계수 색깔
         tl.col = "black",        # 변수명 색깔
         tl.srt = 45,             # 변수명 45 도 기울임
         diag = F)                # 대각 행렬 제외
```



R 통계 분석 기초 – 6차시

2020-9-9 (수)

구혜민

MCM GI Convergence Lab

이번 차시 목표

R로 기초통계 분석하기 1 – 연속형 자료의 평균에 대한 검정

- 모집단의 전제 조건 확인 – 정규성 검정 (Normality test)
- t-test
- 비모수적 가설검정

R로 기초통계 분석하기 2 – 다수의 집단에서 평균 비교

- 분산분석 (ANOVA)
- 사후 분석 (Post-Hoc Analysis)

R로 기초통계 분석하기 3 – 범주형 자료의 적합도, 독립성, 동질성 검정

- 카이제곱 검정
- Fisher's exact test

R로 기초통계 분석하기 4 – 두 연속 변수 사이의 선형 관계성 분석

- 상관분석

R shiny로 구현된 Logistic Analysis 소개

R로 기초통계 분석하기 1 – 연속형 자료의 평균에 대한 검정

연속형 변수로 이루어진 모집단의 평균에 대하여 가설 (H_0)을 세우고 이를 검정

집단의 개수가 1개이거나 2개인 경우, 조건에 따라 다음과 같은 검정을 진행한다.

모집단이 정규분포를 따르는 경우

한 개의 집단으로부터 평균 추정 – one-sample t-test

두 개의 집단으로부터 평균 비교 – two-sample t-test

모집단이 정규분포를 따르지 않거나 샘플 수가 적은 경우

비모수적 방법 사용

한 개의 집단/Paired data로부터 평균 추정 – Wilcoxon Signed-Rank Test

두 개의 서로 독립인 집단으로부터 평균 비교 – Wilcoxon Rank-Sum Test (Mann-Whitney U Test)

표 1: 모수적/비모수적 평균 검정 방법 간 대응 관계

	모수적 방법	비모수적 방법
1개의 집단	One-sample t-test	Wilcoxon Signed-Rank Test
2개의 집단	two-sample t-test	Wilcoxon Rank-Sum Test (Mann-Whitney U Test)

모집단의 전제 조건 확인 – 정규성 검정 (Normality test)

t-test는 모집단이 정규 분포를 따르는 것을 전제로 하며 데이터 값 자체를 이용하여 검정을 한다.

비모수적 방법은 모수에 대한 확률분포를 가정하지 않으며 데이터의 순위를 이용하여 검정을 한다.

- 자료의 정규성이 보장된다면 t-test를 이용하는 것이 더욱 정확하지만
- 자료가 정규분포를 따르지 않거나 샘플수가 적을 때는 비모수적 방법을 이용하는 것이 적합하다.

정규성 검정 (Normality test) 방법

Shapiro-Wilk normality test를 통한 가설 검정

- 귀무가설 H_0: 주어진 샘플의 분포는 정규분포를 따른다

Normal Q-Q plot을 통한 시각적 확인

R에서 Shapiro-Wilk normality test 하기

shapiro.test() 를 통해 샤피로-윌크(Shapiro-Wilk) 검정을 수행한다.

귀무가설 H_0: 주어진 샘플의 분포는 정규분포를 따른다

```
library(MASS)
attach(Pima.tr)
shapiro.test(bmi)

##
##  Shapiro-Wilk normality test
##
## data:  bmi
## W = 0.99104, p-value = 0.2523
```

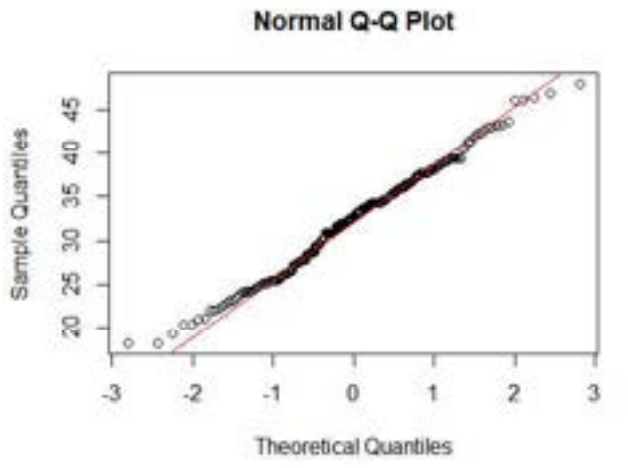
[해석] 유의수준 5%에서 귀무가설을 기각할 수 없다. 데이터가 정규분포를 따른다고 할 수 있다.

R에서 Normal Q-Q plot 그리기

qqnorm()은 정규 분위수-분위수 플롯(Normal Quantile-Quantile plot)을 그려주는 함수이다.

qqline()은 normal Q-Q plot의 기준선 (y=x)을 그려주는 함수이다.

```
qqnorm(bmi)
qqline(bmi, col="red")
```



t-test

모집단이 정규 분포를 따른다는 전제 하에, 모집단의 평균에 대한 가설 (H_0)을 세우고 이를 검정

- 한 개의 집단으로부터 평균 추정 – one-sample t-test
- 두 개의 집단으로부터 평균 비교 – two-sample t-test

One-sample t-test

정규분포를 따르는 모집단으로부터 추출한 샘플로부터 추정한 모집단의 평균이 유의한지 확인

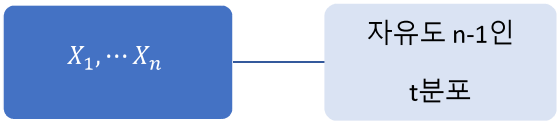
- 귀무가설 $H_0: \mu = \mu_0$
- 대립가설 $H_1: \mu \neq \mu_0$ (양측 검정), $H_1: \mu > \mu_0$ 또는 $\mu < \mu_0$ (단측 검정)

Two-sample t-test

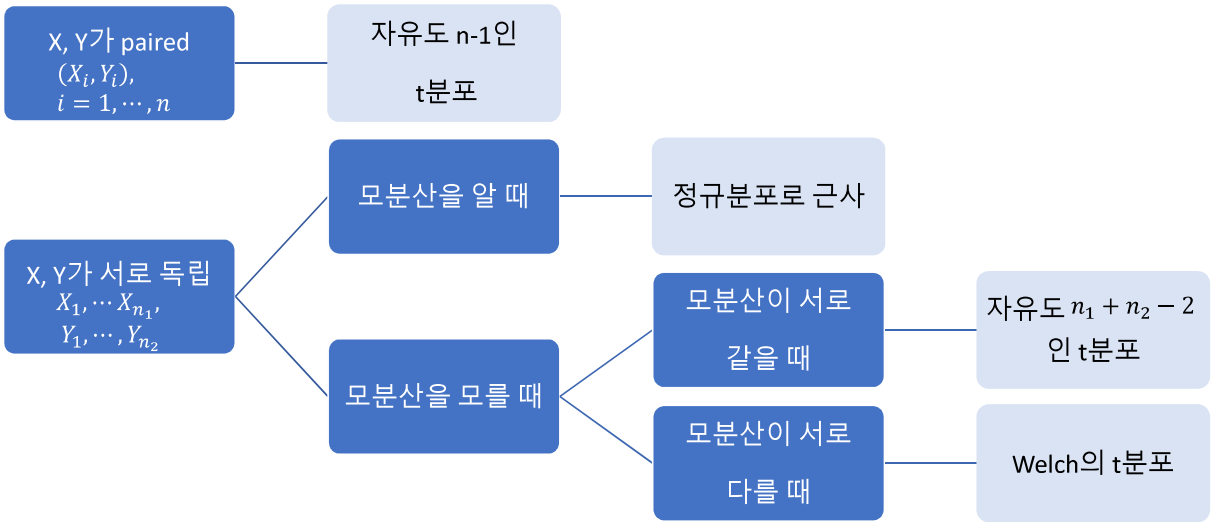
정규분포를 따르는 두 집단의 평균에 통계적으로 유의한 차이가 있는지 확인

- 두 모집단 X, Y의 평균의 차를 비교
- 귀무가설 $H_0: \delta = \mu_x - \mu_y = 0$
- X, Y가 pair로 존재할 경우 – paired t-test (one-sample t-test와 본질적으로 같음)
- X, Y가 서로 독립일 경우 – two-sample t-test
  - 서로 독립일 경우, 모분산을 모른다는 가정 하에, 두 모분산이 같냐 다르냐에 따라 구분

One-sample t-test



Two-sample t-test



R에서 One-sample t-test 하기

t.test(sample, mu=value)를 통해 one-sample t-test를 수행한다.

귀무가설 $H_0: \mu = \mu_0$ (디폴트=0)

대립가설 $H_1: \mu \neq \mu_0$ (양측 검정), $H_1: \mu > \mu_0$ 또는 $\mu < \mu_0$ (단측 검정)

- (디폴트="two.sided") alternative="two.sided"/"greater"/"less"

```
t.test(bmi, mu=30)
```

```
##
##  One Sample t-test
##
## data:  bmi
## t = 5.3291, df = 199, p-value = 2.661e-07
## alternative hypothesis: true mean is not equal to 30
## 95 percent confidence interval:
##  31.45521 33.16479
## sample estimates:
## mean of x
##     32.31
```

```
bmi.ttest <- t.test(bmi, mu=30)  # t.test 결과값 저장한 뒤 확인
names(bmi.ttest)
```

```
##  [1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
##  [6] "null.value"  "stderr"      "alternative" "method"      "data.name"
```

```
bmi.ttest$p.value
```

```
## [1] 2.661441e-07
```

[해석] 유의수준 0.01 하에서 bmi의 평균이 30과 같다고 할 수 없다.

```r
t.test(bmi, mu=30, alternative="greater")  # 대립가설 옵션

##
##  One Sample t-test
##
## data:  bmi
## t = 5.3291, df = 199, p-value = 1.331e-07
## alternative hypothesis: true mean is greater than 30
## 95 percent confidence interval:
##  31.59367      Inf
## sample estimates:
## mean of x
##     32.31

t.test(bmi, mu=30, alternative="less")  # 대립가설 옵션

##
##  One Sample t-test
##
## data:  bmi
## t = 5.3291, df = 199, p-value = 1
## alternative hypothesis: true mean is less than 30
## 95 percent confidence interval:
##      -Inf 33.02633
## sample estimates:
## mean of x
##     32.31
```

[해석] 대립가설이 "greater"일 때 귀무가설을 기각했으므로, 평균이 30보다 크다.

등분산성 검정

two-sample t-test에서 두 집단 X, Y 가 서로 독립인 경우, 분산이 같을 때와 다를 때에 따라 다른 t-test를 수행한다.

- 분산이 같을 경우: Pooled variance에 의한 t-test
- 분산이 다를 경우: Welch t-test

등분산성의 검정을 위해 분산의 비에 대해 F-test를 수행한다.

귀무가설 $H_0: \sigma_x^2 = \sigma_y^2$

R에서 등분산성 검정하기

var.test(sample1 ~ sample2)를 통해 등분산성을 검정한다.

R에서 등분산성 검정하기

var.test(sample1 ~ sample2)를 통해 등분산성을 검정한다.

귀무가설 $H_0: \sigma_x^2 = \sigma_y^2$

```
var.test(bmi ~ type)
```

```
##
##  F test to compare two variances
##
## data:  bmi by type
## F = 1.7595, num df = 131, denom df = 67, p-value = 0.01115
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.140466 2.637564
## sample estimates:
## ratio of variances
##            1.75945
```

[해석] 유의수준 0.05 하에서 type별 분산이 같다고 할 수 없다.

R에서 Two-sample t-test 하기

t.test(sample1, sample2)를 통해 two-sample t-test를 수행한다.

귀무가설 $H_0: \delta = \mu_x - \mu_y = 0$

var.equal=TRUE 를 설정하여 Pooled variance에 의한 t-test가 가능하다. (Default=FALSE)

```
t.test(bmi ~ type)  # 두 집단의 분산이 다를 때
```

```
##
##  Welch Two Sample t-test
##
## data:  bmi by type
## t = -4.512, df = 171.46, p-value = 1.188e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.224615 -2.044547
## sample estimates:
##  mean in group No mean in group Yes
##          31.07424          34.70882
```

[해석] 귀무가설을 기각하여 type별 bmi가 다르다고 결론 내릴 수 있다.

R에서 Paired t-test 하기

t.test(sample1, sample2, paired=TRUE)를 통해 paired t-test를 수행한다.

귀무가설 $H_0: \delta = \mu_x - \mu_y = 0$

```
data(anorexia)

FT <- subset(anorexia, Treat=='FT')
head(FT)
```

```
##    Treat Prewt Postwt
## 56    FT  83.8   95.2
## 57    FT  83.3   94.3
## 58    FT  86.0   91.5
## 59    FT  82.5   91.9
## 60    FT  86.7  100.3
## 61    FT  79.6   76.7
```

```
shapiro.test(FT$Prewt - FT$Postwt)  # 정규성 검정
```

```
##
##  Shapiro-Wilk normality test
##
## data:  FT$Prewt - FT$Postwt
## W = 0.95358, p-value = 0.5156
```

[해석] FT 치료를 받은 집단의 데이터는 정규성을 띄므로 paired t-test를 진행한다.

```
t.test(FT$Prewt, FT$Postwt, paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  FT$Prewt and FT$Postwt
## t = -4.1849, df = 16, p-value = 0.0007003
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -10.94471  -3.58470
## sample estimates:
## mean of the differences
##              -7.264706
```

[해석] 치료 이전과 이후 평균차가 0이 아니므로 치료 효과가 있다고 볼 수 있다.

비모수적 가설검정

비모수적 방법은 모수에 대한 확률분포를 가정하지 않으며 데이터의 순위(합)를 이용하여 검정을 한다.

- 자료의 정규성이 보장된다면 t-test를 이용하는 것이 더욱 정확하지만
- 자료가 정규분포를 따르지 않거나 샘플수가 적을 때는 비모수적 방법을 이용하는 것이 적합하다.

One-sample t-test, Paired t-test ↔ Wilcoxon Signed-Rank test

Two-sample t-test ↔ Wilcoxon Signed-Sum test (Mann-Whitney U test)

R에서 비모수적 가설 검정하기

wilcox.test(sample, mu=) : Wilcoxon Signed-Rank test

wilcox.test(sample1, sample2, paired=TRUE) : Wilcoxon Signed-Rank test

wilcox.test(sample1, sample2) : Wilcoxon Signed-Sum test

도표 1: 비모수적 가설 검정 진행 절차

R에서 비모수적 가설 검정하기

wilcox.test(sample, mu=) : Wilcoxon Signed-Rank test

wilcox.test(sample1, sample2, paired=TRUE) : Wilcoxon Signed-Rank test

wilcox.test(sample1, sample2) : Wilcoxon Signed-Sum test

```
CBT <- subset(anorexia, Treat=='CBT')
shapiro.test(CBT$Prewt - CBT$Postwt)

##
##  Shapiro-Wilk normality test
##
## data:  CBT$Prewt - CBT$Postwt
## W = 0.89618, p-value = 0.007945
```

[해석] CBT 치료를 받은 집단의 데이터는 정규성을 띈다고 볼 수 없으므로 비모수적 방법으로 (paired) 평균 비교.

R에서 비모수적 가설 검정하기

wilcox.test(sample, mu=) : Wilcoxon Signed-Rank test

wilcox.test(sample1, sample2, paired=TRUE) : Wilcoxon Signed-Rank test

wilcox.test(sample1, sample2) : Wilcoxon Signed-Sum test

```
wilcox.test(CBT$Prewt, CBT$Postwt, paired=TRUE)

## Warning in wilcox.test.default(CBT$Prewt, CBT$Postwt, paired = TRUE):
cannot
## compute exact p-value with ties

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  CBT$Prewt and CBT$Postwt
## V = 131.5, p-value = 0.06447
## alternative hypothesis: true location shift is not equal to 0
```

[해석] 치료 이전과 이후 평균차가 0임을 기각할 수 없으므로 치료 효과가 없다고 볼 수 있다.

R에서 비모수적 가설 검정하기

```
placebo <- c(7, 5, 6, 4, 12)
new_drug <- c(3, 6, 4, 2, 1)
wilcox.test(placebo, new_drug, exact=FALSE)

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  placebo and new_drug
## W = 22, p-value = 0.05855
## alternative hypothesis: true location shift is not equal to 0
```

[해석] 데이터 수가 적으므로 비모수적 방법으로 two-sample t-test 수행. 유의수준 0.05 하에서 귀무가설을 기각할 수 없으므로 신약의 효과가 없다고 볼 수 있다.

R로 기초통계 분석하기 2 – 다수의 집단에서 평균 비교

연속형 변수로 이루어진 모집단들의 평균을 비교한다는 점에서 t-test와 같지만
집단의 개수가 3개 이상인 경우 분산분석 (Analysis of Variance, ANOVA)를 적용한다.
idea: 집단 내의 분산, 총 평균과 각 집단의 평균의 차이에 의해 생긴 집단 간 분산의 비교

일원 분산 분석 (One-way ANOVA)

k개의 집단 간, 한 개의 요인에 대한 평균 비교
k개의 모집단은 독립이며 정규분포를 따른다고 가정
귀무가설 $H_0: \mu_1 = \mu_2 = \cdots = \mu_k$

사후 검정 (Post-Hoc Analysis)

One-way Anova에서 귀무가설을 기각했을 때, 어디에서 평균에 차이가 나는지를 알기 위한 추가 검정으로 다중비교 (multiple comparison) 수행

이원 분산 분석 (Two-way ANOVA)

k개의 집단 간, 두 개의 요인에 대한 평균 비교 (자세한 내용은 뒤에서 언급)

일원 분산 분석 (One-way ANOVA)

k개의 집단 간, 한 개의 요인에 대한 평균 비교
k개의 모집단은 독립이며 등분산의 정규분포를 따른다고 가정
<자료구조>

집단 1	집단 2	…	집단 k
y_{11}	y_{21}		y_{k1}
y_{12}	y_{22}		y_{k2}
…	…		…
y_{1,n_1}	y_{2,n_2}		y_{k,n_k}

모형: $y_{ij} = \mu_i + \epsilon_i \ (i = 1,2,\cdots,k; j = 1,2,\cdots,n_i) \ where \ \epsilon_i \sim N(0,\sigma^2)$
귀무가설 $H_0: \mu_1 = \mu_2 = \cdots = \mu_k$
대립가설 H_1: μ_i들이 모두 같지는 않다

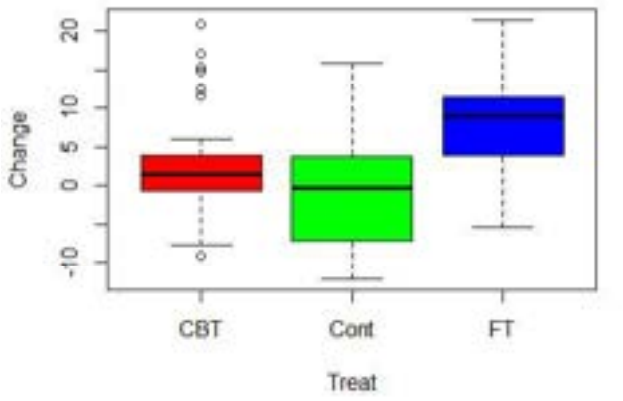
R에서 일원 분산 분석 (One-way ANOVA) 하기

aov(Response ~ Factor)

R에서 일원 분산 분석 (One-way ANOVA) 하기

aov(Response ~ Factor)

```
attach(anorexia)
Change <- Postwt - Prewt
boxplot(Change ~ Treat, col=rainbow(3))
```




```
aov.out <- aov(Change ~ Treat)
summary(aov.out)

##             Df Sum Sq Mean Sq F value Pr(>F)
## Treat        2    615  307.32   5.422 0.0065 **
## Residuals   69   3911   56.68
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[해석] Treat이 유의하므로 치료 이전과 이후의 차이에 해당하는 평균은 각 치료법의 종류에 따라 다르다.

사후 검정 (Post-Hoc Analysis)

One-way Anova에서 귀무가설을 기각했을 때, 어디에서 평균에 차이가 나는지를 알기 위한 추가 검정으로 다중비교 (multiple comparison) 수행 (두 개씩의 처리 조합별 효과 차이의 유무 검토)

- Tukey 검정, 최소유의차 검정 (LSD), Scheffe 방법 등

R에서 Tukey 검정하기

aov.out <- aov(Response ~ Factor)

TukeyHSD(aov.out)

R에서 LSD 검정하기

pairwise.t.test(Response, Factor)

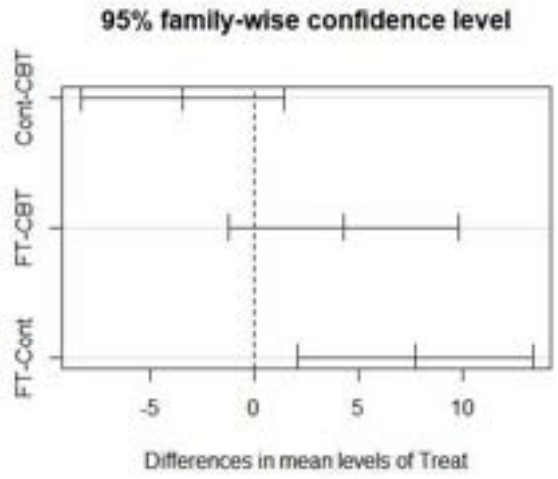
R에서 Tukey 검정하기

aov.out <- aov(Response ~ Factor)

TukeyHSD(aov.out)

```
TukeyHSD(aov.out)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Change ~ Treat)
##
## $Treat
##               diff       lwr       upr     p adj
## Cont-CBT -3.456897 -8.327276  1.413483 0.2124428
## FT-CBT    4.257809 -1.250554  9.766173 0.1607461
## FT-Cont   7.714706  2.090124 13.339288 0.0045127

plot(TukeyHSD(aov.out))
```

[해석] ANOVA의 귀무가설을 기각했으므로 어디에서 차이가 나는지 Tukey의 방법으로 분석한 결과, FT와 Cont 사이의 차이가 유의한 것으로 밝혀졌다.



R에서 LSD 검정하기

pairwise.t.test(Response, Factor)

```
pairwise.t.test(Change, Treat)

##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  Change and Treat
##
##      CBT    Cont
## Cont 0.1368 -
## FT   0.1368 0.0048
##
## P value adjustment method: holm
```

[해석] LSD를 통해서도 FT와 Cont 사이의 차이가 유의한 것으로 밝혀졌다.

이원 분산 분석 (Two-way ANOVA)

k개의 집단 간, 두 개의 요인 (각각 I, J개의 수준) 에 대한 평균 비교

k개의 모집단은 독립이며 등분산의 정규분포를 따른다고 가정

반복이 없는 이원 분산 분석하기

모형: $y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}\ (i = 1,2,\cdots,I; j = 1,2,\cdots,J)\ where\ \epsilon_{ij} \sim N(0,\sigma^2)$

요인1의 효과

- 귀무가설 $H_0: \alpha_1 = \alpha_2 = \cdots = \alpha_I = 0$ (요인1에 의한 효과가 없다)
- 대립가설 H_1: α_i들이 모두 0은 아니다

마찬가지로 요인2의 효과에 대해서도 적용

반복이 있는 이원 분산 분석하기

모형: $y_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ij}\ (i = 1,2,\cdots,I; j = 1,2,\cdots,J)\ where\ \epsilon_{ij} \sim N(0,\sigma^2)$

귀무가설: 요인1의 효과 $\alpha_i = 0$ or 요인2의 효과 $\beta_j = 0$, 교호작용의 효과 $(\alpha\beta)_{ij} = 0$ 가 없다

R에서 이원 분산 분석 (Two-way ANOVA) 하기

aov(Response ~ Factor1 + Factor2) : 반복이 없는 경우

aov(Response ~ Factor1 + Factor2 + Factor1 : Factor2) : 반복이 있는 경우

R에서 이원 분산 분석 (Two-way ANOVA) 하기

aov(Response ~ Factor1 + Factor2) : 반복이 없는 경우

```
teaching_time <- read.table("teaching_time.txt", header=TRUE, sep=" ")
teaching_time
```

```
##    ageGroup method days
## 1       <20      A    7
## 2       <20      B    9
## 3       <20      C   10
## 4     20-29      A    8
## 5     20-29      B    9
## 6     20-29      C   10
## 7     30-39      A    9
## 8     30-39      B    9
## 9     30-39      C   12
## 10    40-49      A   10
## 11    40-49      B    9
## 12    40-49      C   12
## 13      >50      A   11
## 14      >50      B   12
## 15      >50      C   14
```

```
aov.out <- aov(days ~ ageGroup + method, data=teaching_time)
summary(aov.out)

##             Df Sum Sq Mean Sq F value   Pr(>F)
## ageGroup     4 24.933   6.233   14.38 0.001002 **
## method       2 18.533   9.267   21.39 0.000617 ***
## Residuals    8  3.467   0.433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[해석] age와 method 두 요인 모두 유의하므로 teaching 기간의 평균은 age와 method 종류 각각에 따라 다르다.

R에서 이원 분산 분석 (Two-way ANOVA) 하기

aov(Response ~ Factor1 + Factor2 + Factor1 : Factor2) : 반복이 있는 경우

```
summary(ToothGrowth)

##       len        supp         dose
##  Min.   : 4.20   OJ:30   Min.   :0.500
##  1st Qu.:13.07   VC:30   1st Qu.:0.500
##  Median :19.25           Median :1.000
##  Mean   :18.81           Mean   :1.167
##  3rd Qu.:25.27           3rd Qu.:2.000
##  Max.   :33.90           Max.   :2.000

ToothGrowth$dose <- factor(ToothGrowth$dose)
summary(ToothGrowth)

##       len        supp     dose
##  Min.   : 4.20   OJ:30   0.5:20
##  1st Qu.:13.07   VC:30   1  :20
##  Median :19.25           2  :20
##  Mean   :18.81
##  3rd Qu.:25.27
##  Max.   :33.90
```

```r
aov.out <- aov(len ~ supp*dose, data=ToothGrowth)
summary(aov.out)

##             Df Sum Sq Mean Sq F value   Pr(>F)
## supp         1  205.4   205.4  15.572 0.000231 ***
## dose         2 2426.4  1213.2  92.000  < 2e-16 ***
## supp:dose    2  108.3    54.2   4.107 0.021860 *
## Residuals   54  712.1    13.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[해석] supp와 dose 두 요인 모두 유의하지만 둘 사이의 교호작용은 유의수준 0.01하에서만 유의하다.

R로 기초통계 분석하기 3 – 범주형 자료의 적합도, 독립성, 동질성 검정

범주형 변수로 이루어진 자료들의 분할표 (contingency table)를 작성하여 카이제곱 검정

집단의 개수가 1개이거나 2개인 경우, 조건에 따라 다음과 같은 검정을 진행한다.

표 2: 분할표의 예

	No heart Attack	Heart Attach
Placebo	10845	189
Aspirin	10933	104

적합성 검정 (goodness of fit)

r개의 범주형 자료에 대해 각각의 관측된 비율 값이 기댓값과 같은지 조사하는 검정

귀무가설 $H_0: p_1 = p_{10}, p_2 = p_{20}, \cdots, p_r = p_{r0}$

독립성 검정 (test for independence)

두 가지 특성의 범주가 각각 r개, c개일 때 이들 특성이 서로 독립인지 조사하는 검정

귀무가설 $H_0: p_{ij} = p_i p_j \; \forall i = 1,2,\cdots,r;\; j = 1,2,\cdots,c$

동질성 검정 (test for homogeneity)

요인 1, 2를 가진 각 subpopulation에서 정해진 표본의 크기만큼 자료를 추출할 때, subpopulation 간 비율이 동일한지 조사하는 검정

귀무가설 $H_0: p_{1j} = p_{2j} = \cdots = p_{rj} = p_j, j = 1,\cdots,c$

R에서 카이제곱 검정하기

chisq.test(observed_ratio, p) : 적합성 검정

귀무가설 $H_0: p_1 = p_{10}, p_2 = p_{20}, \cdots, p_r = p_{r0}$

```
chisq.test(c(24, 16), p=c(0.7, 0.3))  # 적합성 검정

##
##  Chi-squared test for given probabilities
##
## data:  c(24, 16)
## X-squared = 1.9048, df = 1, p-value = 0.1675
```

[해석] 귀무가설을 기각할 수 없으므로, 관측된 도수가 기댓값과 같다고 할 수 있다.

R에서 카이제곱 검정하기

chisq.test(contingency_table) : 독립성 검정

귀무가설 $H_0: p_{ij} = p_i p_j \; \forall i = 1,2,\cdots,r;\; j = 1,2,\cdots,c$

```
countTable <- matrix(c(10845, 189, 10933, 104), nrow=2, byrow=TRUE)
rownames(countTable) <- c("Placebo", "Aspirin")
colnames(countTable) <- c("No Heart Attack", "Heart Attack")
countTable

##         No Heart Attack Heart Attack
## Placebo           10845         189
## Aspirin           10933         104

chisq.test(countTable)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  countTable
## X-squared = 24.429, df = 1, p-value = 7.71e-07
```

[해석] 귀무가설을 기각했으므로, Aspirin의 유무와 Heart Attack의 유무는 서로 관련이 있다.

R에서 카이제곱 검정하기

chisq.test(contingency_table) : 동질성 검정

귀무가설 $H_0: p_{1j} = p_{2j} = \cdots = p_{rj} = p_j, j = 1, \cdots, c$

```
a <- margin.table(HairEyeColor, c(3,2))
a

##         Eye
## Sex      Brown Blue Hazel Green
##   Male      98  101    47    33
##   Female   122  114    46    31

chisq.test(a)

##
##  Pearson's Chi-squared test
##
## data:  a
## X-squared = 1.5298, df = 3, p-value = 0.6754
```

[해석] 귀무가설을 기각할 수 없으므로, 남성과 여성 간에 눈의 색깔에 차이가 없다고 할 수 있다.

Fisher's exact test

초기하분포를 이용하여 정확한 p-value를 구하는 방법이다.

샘플 수가 적거나 범주의 수가 많아서 테이블의 기대빈도가 매우 작아지게 되는 경우 (4 이하) 카이제곱 검정의 정확도가 떨어지며, 이때 Fisher's exact test를 사용한다.

- 귀무가설 H_0: 두가지 특성 간에 연관성이 없다

R에서 Fisher's exact test 수행하기

fisher.test(contingency_table) : 독립성 검정

R에서 Fisher's exact test 수행하기

fisher.test(contingency_table) : 독립성 검정 귀무가설 H_0: 두가지 특성 간에 연관성이 없다

```
TeaTasting <- matrix(c(3, 1, 1, 3), nrow=2)
colnames(TeaTasting) <- c("guessed M first", "guessed T first")
rownames(TeaTasting) <- c("true M first", "true T first")
TeaTasting

##             guessed M first guessed T first
## true M first               3               1
## true T first               1               3

fisher.test(TeaTasting)

##
##  Fisher's Exact Test for Count Data
##
## data:  TeaTasting
## p-value = 0.4857
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##    0.2117329 621.9337505
## sample estimates:
## odds ratio
##   6.408309
```

[해석] 귀무가설을 기각할 수 없으므로, 바리스타는 '차를 먼저 따르고 우유를 나중에 넣는지, 그 반대인지'를 구분하는 능력이 유의하지 않다.

R로 기초통계 분석하기 4 – 두 연속 변수 사이의 선형 관계성 분석

상관분석

두 연속 변수가 서로 선형 상관관계에 있는지 검정하는 통계 분석 기법

상관계수(Correlation Coefficient)

- 두 변수가 얼마나 선형으로 관련되어 있는지, 선형 연관성의 정도를 나타내는 값
- -1~1 사이의 값을 지니고 절댓값이 1에 가까울수록 선형 연관성이 크다는 의미
- 상관계수가 양수면 양의 상관 관계, 음수면 음의 상관 관계
- $\rho = \frac{COV(X,Y)}{\sigma(X)\sigma(Y)}$, 관측값: r

귀무가설 $H_0: \rho = 0$ 하에서 $t = \frac{\sqrt{n-2}r}{\sqrt{1-r^2}} \sim t(n-2)$

R에서 상관분석 수행하기

함수	Pearson	Spearman	Kendall	p-values	CI	Pairwise correlations
cor(data)	v	v	v			v
cor.test(data)	v	v	v	v	v	
rcorr(data)	v	v		v		v

R에서 상관분석 수행하기

```
attach(iris)
cor(Sepal.Length, Petal.Width)

## [1] 0.8179411

cor.test(Sepal.Length, Petal.Width)

##
##  Pearson's product-moment correlation
##
## data:  Sepal.Length and Petal.Width
## t = 17.296, df = 148, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7568971 0.8648361
## sample estimates:
##      cor
## 0.8179411
```

[해석] Sepal.Length와 Petal.Width 간에 양의 직선관계가 있다고 할 수 있다.

```
cor(iris[, 1:4])

##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000

pairs(iris[, 1:4])

install.packages("Hmisc",repos = "http://cran.us.r-project.org")
library(Hmisc)
```

```
rcorr(as.matrix(iris[, 1:4]))
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length         1.00       -0.12         0.87        0.82
## Sepal.Width         -0.12        1.00        -0.43       -0.37
## Petal.Length         0.87       -0.43         1.00        0.96
## Petal.Width          0.82       -0.37         0.96        1.00
##
## n= 150
##
##
## P
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length              0.1519      0.0000       0.0000
## Sepal.Width  0.1519                   0.0000       0.0000
## Petal.Length 0.0000       0.0000                  0.0000
## Petal.Width  0.0000       0.0000      0.0000
```

[해석] 나머지 변수들의 조합에 대해서도 상관계수를 확인할 수 있다.

R shiny로 구현된 Logistic Analysis 소개

Logistic Analysis

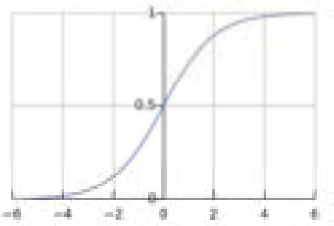
0과 1값으로 구분된 binary response variable을 설명변수 X를 통해 설명하고자 한다.

이때 주어진 X값에 대한 로그 오즈비 $\log(odds) = \log\left(\frac{P(Y=1|X)}{P(Y=0|X)}\right) = \log\left(\frac{p}{1-p}\right)$,

$where\ p = P(Y=1|X)$ 가 X에 대해 선형이라고 하면 (i.e.$\log\left(\frac{p}{1-p}\right) = aX + b$)

$$p = \frac{1}{1+e^{-(aX+b)}}$$

와 같이 시그모이드형 함수로 설명변수와 반응변수를 매핑할 수 있다.



R shiny

R 분석을 반응형 웹 애플리케이션으로 구현해주는 방법이다.

https://tinyurl.com/Logistic-and-OR-plot

https://tinyurl.com/Logistic-and-OR2

아빠가 들려주는 R통계 – 저자 김지형 참고