동국대학교
dongguk university

데이터 프레임의 세계로!

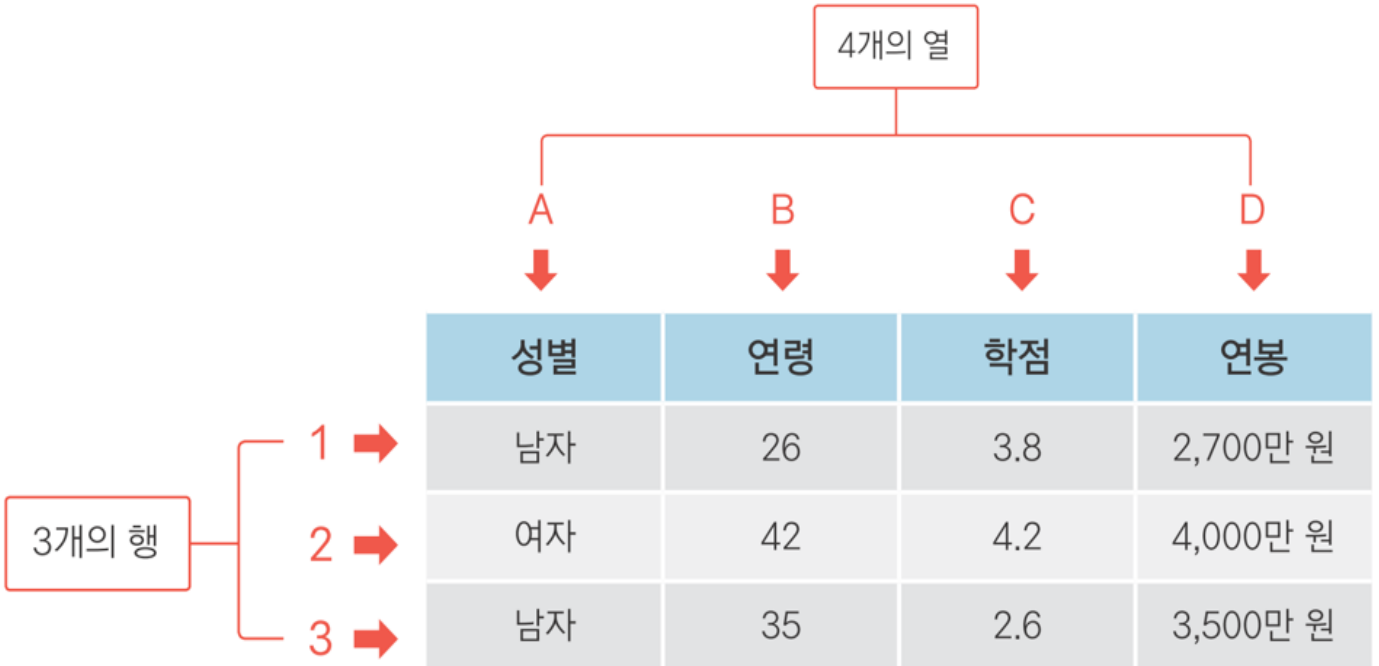
엄진영

데이터는 어떻게 생겼나? - 데이터 프레임 이해하기

- 데이터 프레임
  - 가장 많이 사용하는 데이터 형태
  - 행과 열로 구성된 사각형 모양의 표처럼 생김

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

데이터 프레임



	성별	연령	학점	연봉
1	남자	26	3.8	2,700만 원
2	여자	42	4.2	4,000만 원
3	남자	35	2.6	3,500만 원

- '열'은 속성
- '행'은 한 사람의 정보

데이터가 크다 = 행이 많다 또는 열이 많다

- 행이 많다 ➔ 컴퓨터가 느려짐 ➔ 고사양 장비 구축
- 열이 많다 ➔ 분석 방법의 한계 ➔ 고급 분석 방법

데이터의 행이 늘어난다면?

번호	성별	연령
1	남자	26
2	여자	42
⋮	⋮	⋮
1,000,000	남자	27

데이터의 열이 늘어난다면?

번호	성별	연령	학점	연봉	…	출신지	전공
1	남자	26	3.8	2,700만	…	서울	경영
2	여자	42	4.2	4,000만	…	부산	심리
3	남자	27	2.6	3,200만	…	대전	사회

데이터 프레임 만들기 - 시험 성적 데이터를 만들어 보자!

- 데이터 입력해 데이터 프레임 만들기
  - 변수 만들기
  - 데이터 프레임 만들기

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

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, math로 데이터 프레임 생성해서 df_midterm에 할당
df_midterm <- data.frame(english, math)
df_midterm
##   english math
## 1      90   50
## 2      80   60
## 3      60  100
## 4      70   20
```

데이터 프레임 만들기 - 시험 성적 데이터를 만들어 보자!


```
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$english)  # df_midterm의 english로 평균 산출
## [1] 75
mean(df_midterm$math)     # df_midterm의 math로 평균 산술
## [1] 57.5
```

데이터 프레임 만들기 - 시험 성적 데이터를 만들어 보자!

- 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
```

실습

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

제품	가격	판매량
사과	1800	24
딸기	1500	38
수박	3000	13

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

외부 데이터 이용하기 - 축적된 시험 성적 데이터를 불러오자!

- 깃허브에서 실습에 사용할 excel_exam.xlsx 파일 다운
  - bit.ly/doit_ra 에서 Data 폴더에서 다운
- 프로젝트 폴더에 엑셀파일 삽입
- Readxl 패키지 설치하고 로드하기

```
# readxl 패키지 설치
install.packages("readxl")
# readxl 패키지 로드
library(readxl)
df_exam <- read_excel("excel_exam.xlsx")
# 엑셀 파일을 불러와서 df_exam에 할당
df_exam                              # 출력
## # A tibble: 20 x 5
##       id class  math english science
##  1     1     1    50      98      50
##  2     2     1    60      97      60
##  3     3     1    45      86      78
##  4     4     1    30      98      58
```

외부 데이터 이용하기 - 축적된 시험 성적 데이터를 불러오자!

- 직접 경로 지정

```
df_exam <- read_excel("d:/easy_r/excel_exam.xlsx")
```

- 분석하기

```
mean(df_exam$english)
## [1] 84.9
mean(df_exam$science)
## [1] 59.45
```

엑셀 파일 첫 번째 행이 변수명이 아니라면?

- Read_excel()은 기본적으로 엑셀 파일의 첫 번째 행을 변수명으로 인식해 불러온다.
  - 변수명 없이 첫 번째 행부터 바로 데이터가 시작되는 경우 첫번째 행의 데이터가 변수명으로 지정되면서 유실되는 문제가 발생한다.
  - col_names = F : 첫번째 행을 변수명이 아닌 데이터로 인식

```
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
```

- 문자가 들어 있는 파일을 불러올 때는 stringsAsFactors = F

```
df_csv_exam <- read.csv("csv_exam.csv", stringsAsFactors = F)
```

데이터 프레임을 CSV 파일로 저장하기

1. 데이터 프레임 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
```

2. CSV 파일로 저장하기

```
write.csv(df_midterm, file = "df_midterm.csv")
```

RData 파일 활용하기

- R 전용 데이터 파일
- 다른 파일 들에 비해 R에서 읽고 쓰는 속도가 빠르고 용량이 작다
  - R에서 분석 작업을 할 때: RData 파일
  - R을 사용하지 않는 사람과 파일을 주고 받을 때: CSV 파일

1. 데이터 프레임을 RData 파일로 저장하기

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

2. Rdata 파일 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
```

RData 파일 활용하기

- 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_exam에 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")
```

정리하기


```
# 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
```