

3장. 데이터 입력

1

데이터의 입력

- Base R 함수로 가능 :read.table(), readcsv()..
 - 큰 파일의 경우 많은 시간 소요, 비효율적.
- 텍스트 파일 불러오기: 패키지 readr
 - read_table()
 - read_csv()
 - read_fwf()
- Excel 파일 불러오기: 패키지 readxl
 - read_excel()
- HTML 테이블 불러오기: 패키지 rvest
 - read_html(), html_nodes(), html_table()

2

2

3.1 텍스트 파일 불러오기: 패키지 readr

- 행과 열의 2차원 형태 텍스트 파일 불러오기
- 자료의 입력 형태에 따라 다른 함수 사용
 - 빈칸 구분: `read_table()`, `read_table2()`
 - 콤마 구분: `read_csv()`
 - 고정 포맷: `read_fwf()`
- 웹 서버(<http://>, <https://>) 텍스트 파일 불러오기 가능
- 압축 파일(.gz, .zip)은 자동으로 압축 해제하고 불러옴

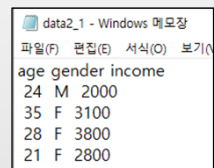
3

3

3.1.1 함수 `read_table2()`로 텍스트 파일 불러오기

- `read_table()`: 자료 사이의 간격이 일정한 경우에 사용 가능
 - `read_table2()`: 자료 사이의 간격에 제약 없음
- ① 데이터 파일 첫 줄에 변수 이름이 입력되어 있는 경우
- 파일 위치와 함께 파일 이름만 입력

```
> library(readr)
> read_table2("C:/Data/data2_1.txt")
Parsed with column specification:
cols(
  age = col_double(),
  gender = col_character(),
  income = col_double()
)
# A tibble: 4 x 3
  age gender income
<dbl> <chr> <dbl>
1    24 M      2000
2    35 F      3100
3    28 F      3800
4    21 F      2800
```



data2_1 - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V)

age gender income
24 M 2000
35 F 3100
28 F 3800
21 F 2800

4

4

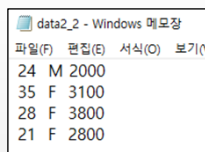
- 사용자가 변수 유형 선언
 - col_types에 유형을 나타내는 문자 입력된 변수 순서대로 나열
 - 자료 유형을 나타내는 문자: c(문자형), i(정수형), d(실수형), n(숫자형), l(논리형), f(요인), D(날짜)

```
> read_table2("C:/Data/data2_1.txt", col_types="dcd")
# A tibble: 4 x 3
  age gender income
  <dbl> <chr> <dbl>
1    24 M      2000
2    35 F      3100
3    28 F      3800
4    21 F      2800
```

5

5

② 변수 이름 없이 데이터만 입력되어 있는 경우



	X1	X2	X3
1	24	M	2000
2	35	F	3100
3	28	F	3800
4	21	F	2800

- col_names 반드시 사용
 - FALSE 지정: 변수 이름은 X1, X2, ...
 - 변수 이름이 있는 문자형 벡터 지정

```
> read_table2("C:/Data/data2_2.txt", col_names = FALSE)
# A tibble: 4 x 3
  X1 X2 X3
  <dbl> <chr> <dbl>
1    24 M      2000
2    35 F      3100
3    28 F      3800
4    21 F      2800
```

```
> read_table2("C:/Data/data2_2.txt",
  col_names = c("age", "gender", "income"))
```

6

6

③ 데이터 파일에 주석이 입력된 경우

```
data2_3 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
# x1=age, x2=gender
# x3=income
# x2=1(Male) x2=0(Female)
x1 x2 x3
24 1 2000
35 0 3100
28 0 3800
21 0 2800
```

- comment: 주석 기호 지정

```
> read_table2("C:/Data/data2_3.txt", comment="#")
# A tibble: 4 x 3
  x1 x2 x3
<dbl> <dbl> <dbl>
1 24 1 2000
2 35 0 3100
3 28 0 3800
4 21 0 2800
```

- skip: 첫 줄을 시작으로 읽지 않고 무시할 행의 개수 지정

```
> read_table2("C:/Data/data2_3.txt", skip=3)
```

7

7

④ 결측값이 NA가 아닌 다른 기호로 입력된 경우

```
data2_4 - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
age gender income
24 M 2000
. F 3100
28 . 3800
21 F NA
```

- na에 해당 기호 지정
 - 점(.)으로만 결측값 입력: na = "."

```
> read_table2("C:/Data/data2_4.txt", na=".")
# A tibble: 4 x 3
  age gender income
<dbl> <chr> <chr>
1 24 M 2000
2 NA F 3100
3 28 NA 3800
4 21 F NA
```

변수 income의 유형이
문자형이 된 이유는?

- 점(.)과 NA가 모두 결측값으로 입력: na = c(".", "NA")

```
> read_table2("C:/Data/data2_4.txt", na=c(".", "NA"))
```

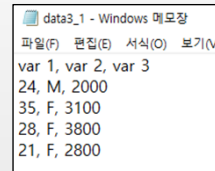
8

8

3.1.2 함수 read_csv()로 CSV 파일 불러오기

- CSV 파일: 자료들이 콤마로 구분된 텍스트 파일

```
> read_csv("C:/Data/data3_1.txt")
# A tibble: 4 x 3
  `var 1` `var 2` `var 3`
    <dbl> <chr>    <dbl>
1      24 M      2000
2      35 F      3100
3      28 F      3800
4      21 F      2800
```



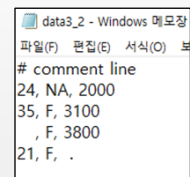
- 변수 이름 규칙: 문자로 시작하고 중간에 빈칸이 없어야 함
- 변수 이름(var 1)에 backtick 기호 (tab키 위, 인용부호와 구분)
: 규칙에 어긋난 문자열을 tibble에서 변수 이름으로 사용할 때 적용

9

9

- read_table2()에서 사용했던 입력 요소
 - col_types, col_names, na, comment, skip: 동일하게 작동

```
> read_csv("C:/Data/data3_2.txt",
           col_names = FALSE,
           comment = "#", na = c(".", "NA"))
# A tibble: 4 x 3
  X1 X2      X3
  <dbl> <lgl> <dbl>
1    24 NA    2000
2    35 FALSE 3100
3    NA FALSE 3800
4    21 FALSE  NA
```



- 빈칸: NA로 인식
- X2가 논리형?

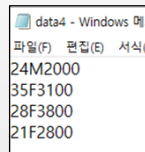
```
> read_csv("C:/Data/data3_2.txt", col_names = FALSE,
           col_types = "dcd", comment = "#", na = c(".", "NA"))
# A tibble: 4 x 3
  X1 X2      X3
  <dbl> <chr> <dbl>
1    24 NA    2000
2    35 F    3100
3    NA F    3800
4    21 F      NA
```

10

10

3.1.3 함수 read_fwf()로 고정 포맷 파일 불러오기

- 고정 포맷 파일



- age 1-2열, gender 3열, income 4-7열
- read_table2(), read_csv()는 사용할 수 없는 형태

```
> read_fwf("C:/Data/data4.txt",
  col_positions = fwf_widths(widths = c(2, 1, 4),
    col_names = c("age", "gender", "income"))
)
```

```
> read_fwf("C:/Data/data4.txt",
  col_positions = fwf_positions(start = c(1, 3, 4),
    end = c(2, 3, 7),
    col_names = c("age", "gender", "income"))
)
```

col_names 생략 시 변수 이름: X1, X2, ...

11

11

- 데이터 프레임을 외부 텍스트 파일로 저장

- write_delim(): 빈칸(디폴트)으로 구분된 자료로 저장
- write_csv(): 콤마로 구분된 자료로 저장
- write_tsv(): 탭으로 구분된 자료로 저장

예제: 데이터 프레임 women

30대 미국 여성 15명의 몸무게(파운드 단위)와 키(인치 단위) 측정 데이터

```
> write_delim(women, "C:/Data/women.txt")
> write_csv(women, "C:/Data/women.csv")
> write_tsv(women, "C:/Data/women.tsv")
```

height	weight
58	115
59	117
60	120
61	123
62	126
63	129
64	132
65	135
66	139
67	142

height,weight
58,115
59,117
60,120
61,123
62,126
63,129
64,132
65,135
66,139
67,142

1	height	weight
2	58	115
3	59	117
4	60	120
5	61	123
6	62	126
7	63	129
8	64	132
9	65	135
10	66	139
11	67	142
12	68	146
13	69	150
14	70	154
15	71	159
16	72	164

height	weight
58	115
59	117
60	120
61	123
62	126
63	129
64	132
65	135
66	139
67	142

12

12

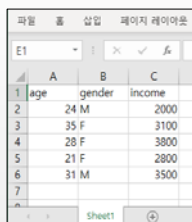
3.2 Excel 파일 불러오기

- Excel 스프레드시트
 - 데이터의 생성 및 가공 작업에 많이 사용됨
 - 단순 데이터만이 아닌 공식과 매크로 등 다양한 속성의 개체들이 공존
- 불러오는 방법
 - Excel 파일 직접 불러오기
 - 패키지 `xlsx`의 함수 `read.xlsx()`
 - 패키지 `readxl`의 함수 `read_excel()`
 - Excel 파일을 csv 파일로 변환하여 불러오기

13

13

- 패키지 `readxl`의 함수 `read_excel()`



	A	B	C
1	age	gender	income
2	24	M	2000
3	35	F	3100
4	28	F	3800
5	21	F	2800
6	31	M	3500

```
> library(readxl)
> read_excel("C:/Data/data5.xlsx")
# A tibble: 5 x 3
  age gender income
<dbl> <chr> <dbl>
1    24 M      2000
2    35 F      3100
3    28 F      3800
4    21 F      2800
5    31 M      3500
```

첫 번째 시트의 데이터
입력이 디폴트

다른 시트의 데이터 입
력은 옵션 sheet에 지정

옵션 range: 시트 전체 데이터 중 일부분만 입력

```
> read_excel("C:/Data/data5.xlsx", range = "A1:B5")
# A tibble: 4 x 2
  age gender
<dbl> <chr>
1    24 M
2    35 F
3    28 F
4    21 F
```

14

14

3.3 SAS 파일 불러오기(skip)

- SAS
 - 범용 통계 소프트웨어. 수많은 사용자 보유
 - SAS 전용 데이터 파일을 R로 불러오는 것도 중요한 작업
- SAS 전용 데이터 파일의 입력 방법
 - 패키지 haven의 함수 read_sas()

```
> library(haven)
> read_sas("D:/Data/data6.sas7bdat")
# A tibble: 5 x 3
  age gender income
<dbl> <chr> <dbl>
1    24 M      2000
2    35 F      3100
3    28 F      3800
4    21 F      3500
5    31 M      3500
```

15

15

3.4 HTML 테이블 불러오기(skip)

- 웹에 있는 엄청난 양의 데이터
 - 직접 R로 불러올 수 있다면 상당히 편리할 것임
- HTML 테이블
 - HTML 테이블
 - 패키지 rvest
 - 함수 read_html(), html_nodes(), html_table()

16

16