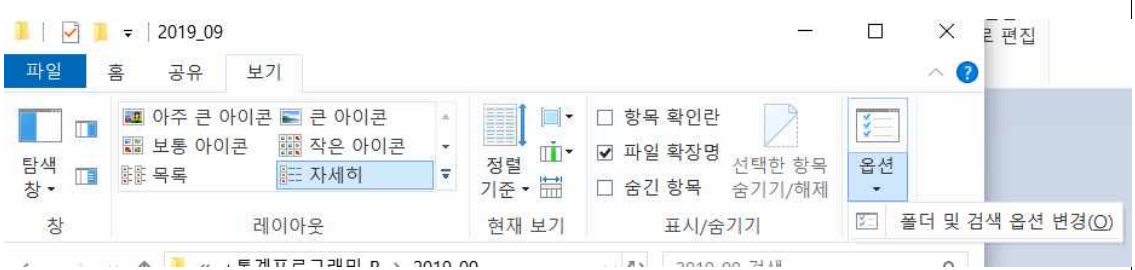
제7주 : 외부파일로부터 자료 불러오기

R에서는 R 환경 밖에서 저장된 파일로부터 자료를 읽을 수 있습니다. 그리고 자료를 파일에 써서 저장을 하여 외부에서 접근할 수 있게 할 수 있습니다.

7.1. 개요: 바이너리 파일과 텍스트 파일

- 컴퓨터의 파일은 크게 바이너리 파일(binary file)과 텍스트 파일(text file)로 구분할 수 있습니다.
- 바이너리 파일은 0과 1이라는 2진수 데이터로만 이루어진 파일입니다.
- 바이너리 파일의 예:
  ▷ 파일 이름의 확장자가 .exe, .dll로 끝나는 프로그램 파일
  ▷ .zip, .rar 등 압축파일
  ▷ .mp3, .jpg, .gif 등 멀티미디어 파일
- 텍스트 파일은 글자(text)들이 쓰여 있는 파일입니다. 사람이 눈으로 내용을 읽고 이해할 수 있습니다.
- 텍스트 파일의 예:
  ▷ 파일 이름의 확장자가 .txt로 끝나는 파일

❖ 윈도우즈(Windows) 탐색기 창에서 파일들의 확장자 보기 설정:
- 보기 탭 선택 --> 옵션 --> 폴더 및 검색 옵션 변경(O)
- “폴더 옵션” 창에서 “보기” 탭 선택 --> “알려진 파일 형식의 파일 확장명 숨기기” 체크표시를 하거나 없앨 수 있다.



7.2 윈도우즈에서 텍스트 파일 만들기

❖ 윈도우즈(Windows) 탐색기 창에서 텍스트 파일 만들기:
- 마우스 우클릭 --> 새로 만들기(N) --> 텍스트 문서

❖ 연습 1
- 파일 이름을 height_weight.txt로 부여하시오.
- 파일을 마우스로 더블클릭하면 notepad.exe를 실행하여 텍스트 파일의 내용을 보거나 편집할 수 있다.
- 아래와 같은 내용으로 파일을 작성하고 저장하시오.

```
height   weight   age
167      78       29
172      72       24
182      88       25
169      54       21
```

7.3 아스키 코드, 유니코드

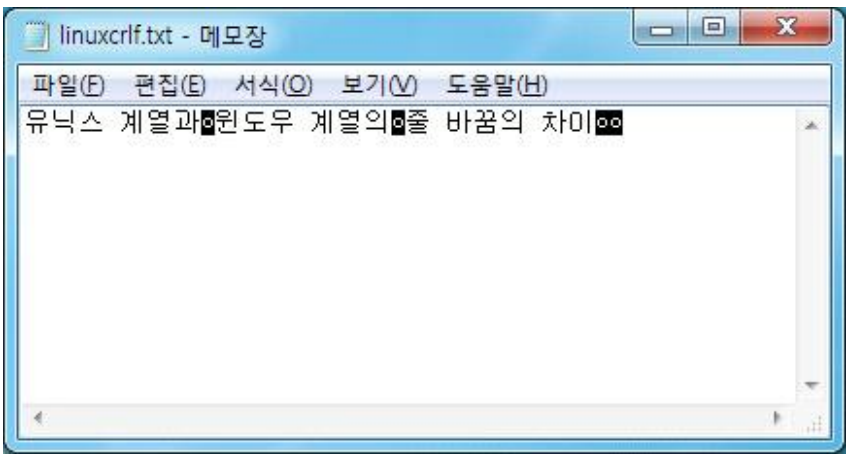
- 파일의 확장자가 .txt라고 해서 텍스트 파일로 구분하는 것은 아닙니다.
- 텍스트 파일의 문자를 컴퓨터에 약속된 방식으로 저장하기 때문에 사람이 읽고 이해할 수 있습니다.
- 아스키 코드(ASCII Code): 0부터 127까지 총 128개의 부호를 사용하여 문자를 표현하고 저장한다.

▷ 예: 10진수 65를 사용해 문자 A를 나타내고 저장한다.
65 (10진수) == 1000001 (저장된 2진수) == A (약속된 문자)
▷ ASCII Table: http://www.asciitable.com/

- 유니코드(Unicode): 전 세계의 모든 문자를 컴퓨터에 일관되게 표현하고 다룰 수 있는 표준으로 제정. UCS, UTF-8 등 여러 인코딩 방식이 있다.

7.4 텍스트 파일의 행과 열의 구분

- 일반 텍스트 파일에 테이블 구조(행과 열)를 저장하는 것은 열 구분자와 행 구분자를 사용한다. 이를 통해 R이나 다른 Application이 이를 읽을 때 테이블 구조로 인식하도록 한다.
- 행 구분자: 글을 입력하면서 커서가 줄의 오른쪽 끝에 이르면 줄바꿈을 해야 한다. 유닉스 계열(리눅스 등)과 윈도우 모두 줄바꿈을 할 때 누르는 키는 '엔터(Enter)'키이고 화면에 나타나는 모습은 같다. 하지만 내부적으로는 다른 행 구분자를 입력(저장)한다.
  ▷ 유닉스 계열의 줄바꿈은 New Line (₩n)을 사용
  ▷ 반면, 윈도우즈 계열에서는 기존 타자기에서처럼 Carriage Return (₩r)과 New Line (₩n)을 같이 사용하여 ₩r₩n으로 줄바꿈합니다.



이런 이유에서 메모장 등에서는 유닉스 계열에서 만든 텍스트 파일을 전부 한 줄에 표시하거나 하는데 R에서 읽어올 경우 이에 구애받지 않고 잘 읽어옵니다.

- 열 구분자: 구분자(Separator)를 통해 열을 구별합니다.
  - ⊙ 공백(white space): " "
  - ⊙ 탭(tab): "₩t"
  - ⊙ 세미콜론(semicolon): ";"
  - ⊙ 쉼표(comma): ","
  - ⊙ 기타

❖ 연습 2

- height_weight.txt를 엑셀을 이용해 불러오기를 수행해 보고 어떤 일을 하는지 관찰하자. (메뉴: 파일 -> 열기)

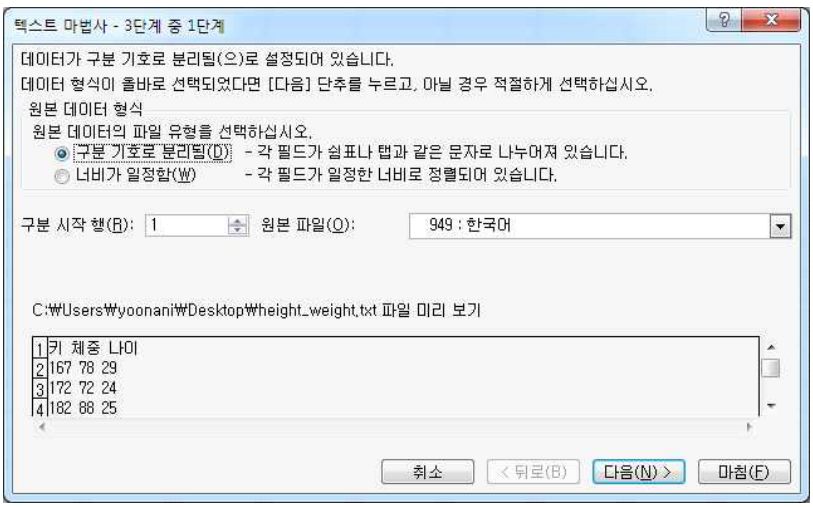


그림 4. 텍스트 마법사 1단계

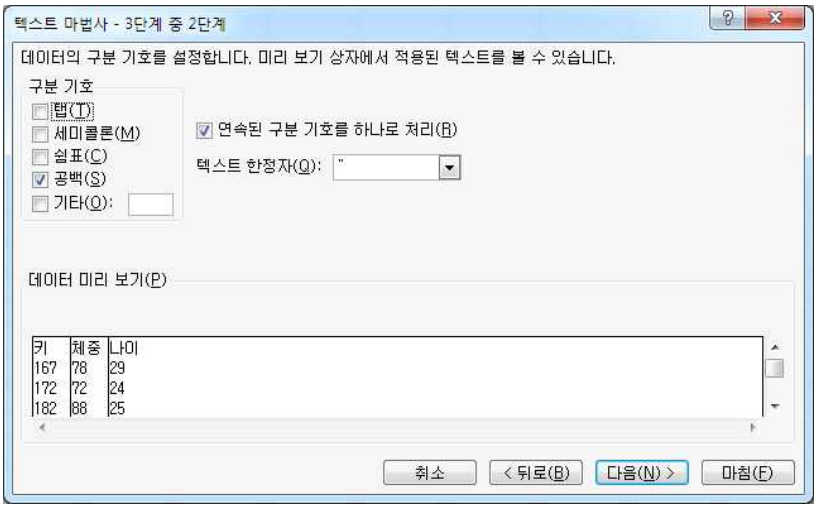


그림 5. 텍스트 마법사 2단계

그림 5. 텍스트 마법사 3단계

7.5 현재 작업폴더 파악하기: getwd(), setwd(), dir()

- 현재 R의 작업폴더 (working directory)를 파악하기 위해 getwd()를 수행한다.

```
> getwd()
[1] "C:/Users/namgi/Documents"
```

- 만일 데이터 파일 위치가 “D:/Lecture/RProgramming/” 이라면 setwd()를 이용해 현재 작업폴더를 변경한다.

```
> setwd(“D:/Lecture/RProgramming/”)
> getwd()
[1] “D:/Lecture/RProgramming/”
```

- dir(): 현재 작업 폴더에 있는 파일들의 목록 출력

```
> dir()
[1] “height_weight.txt“
```

7.6 R에서 파일 읽어오기 : read.table()

- read.table() 함수를 이용해 텍스트 파일로부터 테이블 형태의 자료를 읽을 수 있다.

- 간단한 사용법

```
dat <- read.table(file = 파일명문자열,
                  header = FALSE,
                  sep = "")
```

  - file: 불러올 파일명을 문자열로 입력한다.
  - header: 첫 줄을 열의 이름으로 읽을 것인지(TRUE) 아니면 데이터로 읽을 것인지(FALSE) 여부.
  - sep: 열 구분자로서, 기본값은 공백문자이다. 연속된 구분자는 하나의 구분자로 처리한다. (예를 들어, 공백문자가 연이어 나타날 경우 하나의 공백문자처럼 처리함)

- 그 이외의 전달인자들 (?read.table)

```
read.table(file, header = FALSE, sep = "", quote = "'"",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```

```
> students <- read.table("height_weight.txt",
                         header = T)
```
경고메시지:
```
In read.table("height_weight.txt", header = T) :
incomplete final line found by readTableHeader on
'height_weight.txt'
```
➔이 오류는 파일이 제대로 끝나지 않았음을 의미합니다. 마지막 데이터 입력 후 줄바꿈을 해줘야 합니다.

```
> students <- read.table("height_weight.txt",
                         header = T)
```
➔현재의 작업경로에서 "height_weight.txt"를 읽어오는데 첫 줄은 각 열의 이름으로 인식합니다.
```
> students
  height weight age
1 167     78   29
2 172     72   24
3 182     88   25
4 169     54   21

> str(students)
```
➔자료에 대한 정보를 확인

7.7 R에서 파일 쓰기 : write.table()

- write.table()은 특정 데이터프레임을 텍스트 파일로 저장합니다.

```
write.table(x, file = 파일명문자열, sep = " ",
            row.names = TRUE)
```
- x: 저장할 객체. 행렬이나 데이터프레임
- file: 파일명(문자)
- sep: 열 구분자

- row.names: x의 행이름을 추가로 함께 저장할 것인지 지정하는 논리값. 또는 행이름에 해당하는 문자벡터.

```
> height_weight$total <- height_weight$height + height_weight$weight
```
➔새로운 변수를 데이터에 추가
```
> write.table(height_weight,
              file = "hw_test.txt",
              row.names = FALSE)
```
➔데이터 height_weight를 파일로 저장. 행이름은 저장하지 않는다.

❖ 연습 3.

- 전달인자 row.names = TRUE로 변경하여 다시 저장하고 결과를 확인하시오.

7.8 csv 파일 읽고 쓰기

- 열 구분자를 콤마(,)로 하는 경우 파일 형식을 CSV(Comma Separated Value)라고 합니다.
- 장점: 공백문자를 열 구분자로 할 경우 반복되는 공백문자가 나타날 경우의 난해함을 해소하고, 열 구분이 사람 눈에 잘 드러나도록 콤마(,)를 구분자를 사용합니다.
- 파일확장자는 .csv를 사용합니다.
- R 함수 read.csv() 및 write.csv()는 read.table() 및 write.table()과 기본적으로 동일하나 전달인자 sep= 의 값이 기본적으로 콤마(,)입니다.

```
> students2 <- read.csv("height_weight.csv",
                        header = T)
```
➔현재의 작업경로에서 "height_weight.csv"를 읽어 오는데 첫 줄은 각 열의 이름으로 인식합니다.

```
> students2
 height weight age
1 167   78   29
2 172   72   24
3 182   88   25
4 169   54   21
> students2$ratio <- students2$height / students2$weight
> write.csv(students2, file = "hw_ratio.csv", row.names =
FALSE)
```

7.9 scan() 함수 이용하기

scan()함수를 이용하면 숫자 또는 문자로 된 벡터를 파일이나 키보드로부터 읽어 들일 수 있다.

이용법

```
scan(file = "", what = double(), sep = "")
```

- file: 값을 읽어들일 파일 이름. 만일 "" 로 설정되면 입력값은 키보드에서 읽는다.
- what: 읽어들일 값의 기본자료형을 typeof(what)으로 간주함

z1.txt, z2.txt, z3.txt, z4.txt라는 파일이 있다고 하자.
z1.txt 라는 파일에는 다음과 같은 내용이 들어있다:

```
123
4 5
6
```

z2.txt 라는 파일에는 다음과 같은 내용이 들어있다:

```
123
4.2 5
6
```

z3.txt 에는 다음의 내용이 들어있다:

```
abc
de f
g
```

z4.txt 파일에는 다음의 내용이 있다:

```
abc
123 6
y
```

scan()을 이용해 이 파일들로 무엇을 할 수 있는지 살펴보자.

```
> scan("z1.txt")
Read 4 items
[1] 123 4 5 6
> scan("z2.txt")
Read 4 items
[1] 123.0 4.2 5.0 6.0
> scan("z3.txt")
Error in scan(file, what, nmax, sep, dec, quote, skip,
              nlines, na.strings, :
  scan() expected 'a real', got 'abc'
> scan("z3.txt", what = "")
Read 4 items
[1] "abc" "de" "f" "g"
> scan("z4.txt", what = "")
Read 4 items
[1] "abc" "123" "6" "y"
```

줄이 바뀔 때마다 문자열로 받고 싶다면, 다음처럼 열구분자를 sep="₩n"로 설정하면 된다.

```
> scan("z1.txt", sep = "\n")
Read 3 items
[1] 123  45   6
> scan("z3.txt", what = "")
Read 4 items
[1] "abc" "de" "f" "g‟
> scan("z3.txt", what = "", sep = "\n")
Read 3 items
[1] "abc" "de f" "g"
```

scan()에 파일명 대신 빈 문자열을 넣으면 키보드로부터 입력을 받을 수 있다.

```
> v <- scan("")
1: 12 5 13
4: 3 5 6
7:
Read 6 items
> v
[1] 12 5 13 3 5 6
← 입력을 끝내기 위해서는 마지막을 빈 줄로 남겨둔다.
```

7.10 SPSS 데이터와 Excel 데이터 읽기

foreign 패키지의 read.spss()로 SPSS의 데이터 파일을 읽을 수 있습니다. xlsx 패키지의 read.xlsx()로 엑셀 데이터파일을 읽을 수 있습니다.