



문공 A0015

R 프로그래밍

김 태 완

kimtwan21@dongduk.ac.kr

데이터 입력과 출력

- R에서의 입력과 출력
 - 출력 : 화면 / 파일

```
# 데이터 입력
age <- c(28, 17, 35, 46, 23, 67, 30, 50)
age

# 정보 추출
young <- min(age)
old <- max(age)

# 처리 결과 출력
cat('가장 젊은 사람의 나이는 ', young, '이고,',
    '가장 나이든 사람의 나이는', old, '입니다. \n')
```

데이터 입력과 출력

- R에서의 입력과 출력
 - 출력 : 화면 / 파일

```
install.packages('svDialogs')      # 패키지 설치
library(svDialogs)

user.input <- dlgInput('Input income')$res
user.input

income <- as.numeric(user.input)    # 문자열을 숫자로
income

tax <- income * 0.05                # 세금 계산
cat('세금: ', tax)
```

데이터 입력과 출력

- R에서의 입력과 출력
 - 출력 : 화면 / 파일
 - print() 함수와 cat() 함수
 - 화면에서 프로그램의 처리 결과를 확인하는 가장 간단한 방법은 다음과 같이 결과가 담긴 변수의 내용을 출력하는 것

함수	사용상의 특징
print()	<ul style="list-style-type: none">• 하나의 값을 출력할 때• 데이터프레임과 같은 2차원 자료구조를 출력할 때• 출력 후 자동 줄바꿈
cat()	<ul style="list-style-type: none">• 여러 개의 값을 연결해서 출력할 때 (벡터는 출력되나 2차원 자료구조는 출력되지 않음)• 출력 후 줄바꿈을 하려면 '\n' 필요

예시

- 체질량 지수 계산하기

```
library(svDialogs)
```

```
height <- dlgInput('Input height(cm)')$res  
weight <- dlgInput('Input weight(kg)')$res
```

?

?

?

```
bmi <- weight/(height^2)
```

```
cat('입력한 키는 ', height*100, 'cm, 몸무게는 ', weight, 'kg 입니다. \n', sep = "")  
cat('BMI는 ', bmi, '입니다.', sep = "")
```

데이터 읽기와 쓰기

- 데이터 쓰기
 - 출력 : 화면 / **파일**
 - 데이터 쓰기 관련 함수

함수	설명
getwd()	현재 작업 영역 확인(파일을 저장할 영역 확인)
setwd()	새로운 작업 영역 지정
write.table()	파일을 TXT 형태로 저장
write.csv()	파일을 CSV 형태로 저장

화면에서 출력

파일에서 출력

csv 파일 이용

엑셀 파일 이용

txt 파일 이용

데이터 읽기와 쓰기

- 데이터 쓰기
 - 출력 : 화면 / **파일**
 - R에서 어떤 파일을 읽으려면 그 파일이 위치한 폴더의 경로와 함께 파일 이름을 지정해야 함

```
getwd() # 현재 작업 폴더 알아내기  
setwd('C:/Users/enoug/Desktop/tmp') # 작업 폴더 변경하기  
getwd( )
```

데이터 읽기와 쓰기

- .csv 파일 **읽기**와 쓰기
- read.csv()

```
setwd('C:/Rworks') # 작업 폴더 지정  
air <- read.csv('airquality.csv', header=T) # .csv 파일 읽기  
head(air)  
class(air) # air의 자료구조 확인
```


데이터 읽기와 쓰기

- .csv 파일 읽기와 쓰기
- write.csv()

```
setwd('C:/Rworks')           # 작업 폴더 지정

# setosa 품종 데이터만 추출

my.iris <- subset(iris, Species=='setosa')

# .csv 파일에 저장하기

write.csv(my.iris, 'my_iris.csv', row.names=F)
```

row.names=F : my.iris를 저장할 때 행 번호/행 이름은 제외하라는 의미

데이터 읽기와 쓰기

- 엑셀파일 (.xls / .xlsx) 파일 **읽기**와 쓰기
- read.xlsx()

```
install.packages('xlsx')           # 패키지 설치하기  
library(xlsx)                       # 패키지 불러오기  
air <- read.xlsx('C:/Rworks/airquality.xlsx', header=T, sheetIndex=1)  
air <- read.xlsx('airquality.xlsx', header=T, sheetIndex=1, encoding='UTF-8')  
head(air)
```

- *header=T* : 파일의 첫 번째 행은 데이터의 값이 아닌 열 이름이라는 의미
- *sheetIndex=1* : 엑셀 파일의 첫 번째 시트를 읽으라는 의미

데이터 읽기와 쓰기

- 엑셀파일 (.xls / .xlsx) 파일 읽기와 쓰기
- write.xlsx()

```
library(xlsx)                                # 패키지 불러오기  
my.iris <- subset(iris, Species=='setosa')  # setosa 품종 데이터만 추출  
write.xlsx(my.iris, 'my_iris.xlsx', row.names=F)  # 파일에 저장하기
```

row.names=F : my.iris를 저장할 때 행 번호/행 이름은 제외하라는 의미

데이터 읽기와 쓰기

- .txt 파일 **읽기**와 쓰기
- read.table()

```
d_tab <- read.table("read_tab.txt", sep="\t")
```

- read.delim() : tab을 자동 인식

```
d4 <- read.delim("read_tab.txt", header=T)
```

데이터 읽기와 쓰기

- .txt 파일 읽기와 **쓰기**
- write.table()

```
1 num <- c(1, 2, 3)
2 char <- c("a", "b", "c")
3 log <- c(TRUE, FALSE, TRUE)
4 tx_data <- data.frame(num, char, log)
5 setwd("C:/Users/enoug/Desktop/tmp ")
    ↓
    폴더 이름과 폴더 이름을 구분하는 구분자
    '/'나 '\\'([w] + [shift]) 사용

6 getwd( )

7 write.table(tx_data, file = 'tx_data.txt')
    ↑
    저장할 파일명
    ↓
    tx_data에 저장할 데이터가 들어있음
```

데이터 읽기와 쓰기

- .txt 파일 읽기와 **쓰기**
 - write.table()함수의 매개변수

매개변수	설명	초기값
file	파일명 설정	
append	TRUE -> 기존 파일 내용 뒤에 내용 추가 FALSE -> 기존 파일 내용 제거 후 덮어쓰기	TRUE
quote	TRUE -> 데이터 값에 쌍따옴표(" ") 표시 FALSE -> 데이터 값에 쌍따옴표(" ") 삭제	TRUE
eol	행과 행을 어떻게 구분할지 입력	\n
na	결측치(NA)를 어떻게 표시할지 입력	NA
row.names	행의 이름을 텍스트 파일에 표시할지 입력	TRUE
col.names	열의 이름을 텍스트 파일에 표시할지 입력	TRUE

데이터 읽기와 쓰기

- .txt 파일 읽기와 쓰기
- write.table()

```
1 write.table(tx_data, file = 'tx_data.txt', append = TRUE)
# 기존 tx_data.txt 파일에 tx_data 내용추가
2 write.table(num, file = 'tx_data.txt', append = FALSE)
# 기존 tx_data.txt 파일 내용을 삭제하고 num 벡터로 내용 변경
```

```
"num" "char" "log"
"1" 1 "a" TRUE
"2" 2 "b" FALSE
"3" 3 "c" TRUE
```

tx_data.txt

```
"num" "char" "log"
"1" 1 "a" TRUE
"2" 2 "b" FALSE
"3" 3 "c" TRUE
"num" "char" "log"
"1" 1 "a" TRUE
"2" 2 "b" FALSE
"3" 3 "c" TRUE
```

append = TRUE

```
"x"
"1" 1
"2" 2
"3" 3
```

append = FALSE

화면에서 출력

파일에서 출력

csv 파일 이용

엑셀 파일 이용

txt 파일 이용

데이터 읽기와 쓰기

- .txt 파일 읽기와 쓰기
- write.table()

```
3 write.table(tx_data, file = 'tx_data.txt', quote = FALSE)
# tx_data.txt 파일에 쌍따옴표(" ") 삭제
4 write.table(tx_data, file = 'tx_data.txt', eol = '/')
# tx_data.txt 파일의 행을 '/'으로 구분
```

```
"num" "char" "log"
"1" 1 "a" TRUE
"2" 2 "b" FALSE
"3" 3 "c" TRUE
```

tx_data.txt

```
num char log
1 1 a TRUE
2 2 b FALSE
3 3 c TRUE
```

quote = FALSE

```
"num" "char" "log"/"1" 1 "a" TRUE/"2" 2 "b" FALSE/"3" 3 "c" TRUE/
eol = '/'
```


데이터 읽기와 쓰기

- .txt 파일 읽기와 쓰기
- write.table()

```
5 write.table(tx_data, file = 'tx_data.txt', row.names = FALSE)
# tx_data.txt 파일의 행 이름 삭제
6 write.table(tx_data, file = 'tx_data.txt', col.names = FALSE)
# tx_data.txt 파일의 열 이름 삭제
```

```
"num" "char" "log"
"1" 1 "a" TRUE
"2" 2 "b" FALSE
"3" 3 "c" TRUE
```

tx_data.txt

```
"num" "char" "log"
1 "a" TRUE
2 "b" FALSE
3 "c" TRUE
```

row.names = FALSE

```
"1" 1 "a" TRUE
"2" 2 "b" FALSE
"3" 3 "c" TRUE
```

col.names = FALSE

데이터 읽기와 쓰기

- 실행 결과를 파일로 출력

```
setwd('C:/Rworks')           # 작업 폴더 지정
print('Begin work')           # 화면으로 출력
a <- 10; b <- 20
sink('result.txt', append=T)  # 파일로 출력 시작
cat('a+b=', a+b, '\n')
sink( )                       # 파일로 출력 정지
cat('hello world \n')         # 화면으로 출력
sink('result.txt', append=T)  # 파일로 출력 시작
cat('a*b=', a*b, '\n')
sink( )                       # 파일로 출력 정지
print('End work')             # 화면으로 출력
```

데이터 읽기와 쓰기

- 실행 결과를 파일로 출력

```
setwd('C:/Rworks')  
print('Begin work')  
a <- 10; b <- 20  
sink('result.txt', append=T)  
cat('a+b=', a+b, '\n')  
sink( )  
cat('hello world \n')  
sink('result.txt', append=T)  
cat('a*b=', a*b, '\n')  
sink( )  
print('End work')
```

파일로 출력 시작

파일로 출력 정지

*append=T: 'result.txt'의 내용 맨 마지막에 덧붙여서 출력하라는 의미로,
append=F이면 기존에 있던 내용을 지우고 새로 출력하는 것을 의미.*

예시

- 예시 1 : TXT파일 쓰기

```
1 setwd("C:/R_data")
2 air_6 <- 
3 air_6
4 
```

- 실행 결과

```
> setwd("C:/R_data")
> air_6 <- 
> air_6
  Ozone Solar.R Wind Temp Month Day
32   NA    286   8.6   78     6    1
33   NA    287   9.7   74     6    2
34   NA    242  16.1   67     6    3
...
> 
```

6월의 데이터만 추출하여 air_6에 저장

air_6의 데이터를 'air_6.txt'로 쓰기

예시

- 예시 2 : CSV파일 쓰기

```
1  swiss_F <-   
2  swiss_F  
3  
```

- 실행 결과

```
> swiss_F <-  # swiss의 Fertility가 90 이상인 데이터를 추출  
> swiss_F  
      Fertility Agriculture Examination Education Catholic ...  
Courtelary    80.2    17.0         15         12      9.96  
Delemont      83.1    45.1          6          9     84.84  
...  
>  # swiss_F의 데이터를 'swiss_F.csv'로 쓰기
```

예시

- 예시 3 : 데이터 읽기

```
1 swd_F <-   
2 swd_F
```

- 실행 결과

```
> swd_F <-  # swiss_F.csv의 데이터를  
> swd_F                 데이터프레임 형식으로 읽기
```

	X	Fertility	Agriculture	Examination	Education	Catholic ...
1	Courtelary	80.2	17.0	15	12	9.96
2	Delemont	83.1	45.1	6	9	84.84
...						

예시

- 예시 4 : swd_F 데이터 분석하기

```
# coding here #
```

- 실행 결과

```
>  # Education 행의 합
[1] 500

>  # Examination 행의 평균
[1] 17.11364

>  # Agriculture 행 정렬 값 출력
[1] 1.2 7.7 15.2 16.7 17.0 17.6 18.7 19.4 26.8 27.7 34.0 35.3 36.5 37.6 38.4
[16] 43.5 45.1 45.2 46.6 49.5 50.9 53.3 54.1 55.1 58.1 59.8 60.7 60.8 62.0 63.1
...
```

예시

- 예시 5 : swd_F 데이터 분석하기

```
# coding here #
```

- 실행 결과

```
> swd_F1 <-  # Infant.Mortality가 20보다 큰 데이터를 추출  
>  # swd_F1의 행의 개수 출력  
[1] 21  
>  # swd_F1의 데이터를 표 형식으로 추출
```

	X	Fertility	Agriculture	Examination	Education	Catholic	Infant.Mortality
1	Courtelay	80.2	17.0	15	12	9.96	22.2
2	Delemont	83.1	45.1	6	9	84.84	22.2
3	Moutier	85.8	36.5	12	7	33.77	20.3
4	Neuveville	76.9	43.5	17	15	5.16	20.6

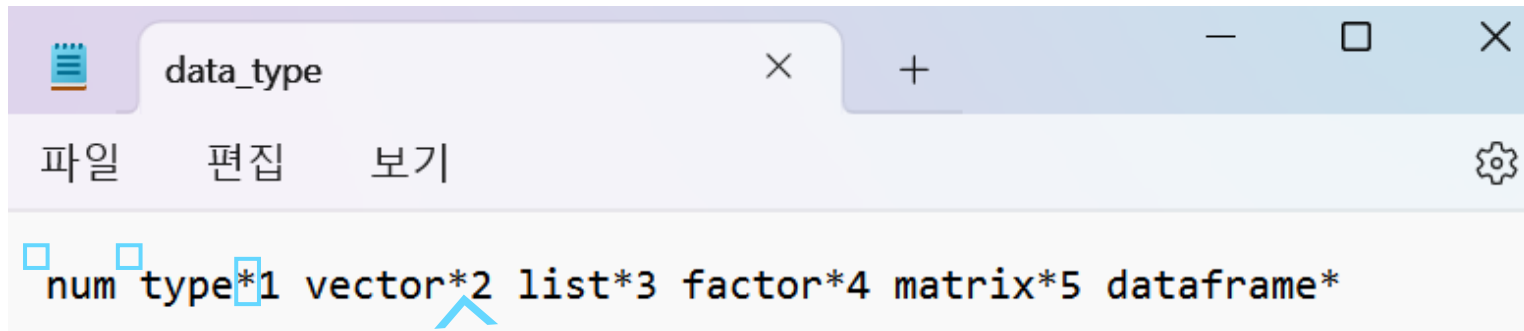
예시

- 예시 6 : txt 파일 쓰기

```
1 num <- c(1,2,3,4,5)
2 type <- c("vector", "list", "factor", "matrix", "dataframe")
3 dt <- data.frame(num, type)
4 # cording here #
```

- 실행 결과

>



예시

- 예시 7 : 파일로 출력하기

```
1 print('안녕하세요')
2 a <- 1; b <- 3
3 sink('dtsc.txt', append=T)
4 cat('저는', '₩n')
5 sink( )
6 cat('데이터 ₩n')
7 sink('dtsc.txt', append=F)
8 cat(a+b, b-a, '언스전공생₩n')
9 sink( )
10 print('입니다.')
```

- 저장된 dtsc.txt 파일로 옳은 것은?

① 저는
데이터
4 2언스전공생
입니다.

② 저는
4 2언스전공생
입니다.

③ 저는
4 2언스전공생

④ 4 2언스전공생

감사합니다

kimtwan21@dongduk.ac.kr

김 태 완