



3. 파일 입출력

chap03_FileIO 수업내용

1. 파일 불러오기
 - (1) 키보드 입력
 - (2) 파일 데이터 가져오기
2. 파일 저장하기
 - (1) 화면(콘솔) 출력
 - (2) 파일에 데이터 저장



3. 데이터 입출력

- 데이터 불러오기

- (1) 키보드 입력

- 1) scan() 함수를 이용

- # 숫자입력

- ```
x <- scan()
```

- ```
1: 1
```

- ```
2: 2
```

- ```
3: 3
```

- ```
4: 4
```

- ```
5: 5
```

- ```
6:
```

- Read 5 items

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



3. 데이터 입출력

```
# 문자입력
y <- scan(what="")
1: a
2: b
3: c
4: d
5: e
6:
  Read 5 items
y # [1] "a" "b" "c" "d" "e"
```



3. 데이터 입출력

(2) 파일 데이터 가져오기

1) read.table() 함수 이용

- 각 값들은 빈칸, 탭, 콜론(:), 세미콜론(;), 콤마(,) 등으로 구분
- 구분자가 공백이나 탭인 경우가 sep 의 기본값, 'header=F' 기본값

① 컬럼명이 없는 파일 불러오기

```
setwd("C:/Rwork/Part-I") # setwd("경로명")
student <- read.table(file="student.txt")
student # 컬럼명이 없으면 V1, V2, V3, V4 형태로 기본 이름 지정
#  V1  V2  V3 V4
#1 101 hong 175 65
#2 201 lee 185 85
#3 301 kim 173 60
#4 401 park 180 70
names(student) <- c("번호","이름","키","몸무게")# 컬럼명 변경
student
```



3. 데이터 입출력

② 컬럼명이 있는 파일 불러오기

```
student1 <- read.table(file="student1.txt", header=TRUE)
```

```
#student1 <-read.table(file.choose(), header=TRUE) # 파일열기대화상자
```

```
student1
```



3. 데이터 입출력

(3) 구분자가 있는 경우(세미콜론, 탭)

```
student2 <- read.table(file="student2.txt", sep=";", header=TRUE)
```

세미콜론 구분자

```
student2
```

```
#student2 <- read.table(file="student2.txt", sep="\t", header=TRUE)
```

탭키 구분자



3. 데이터 입출력

(4) 특정문자 NA 처리(- 문자열을 NA로 처리)

```
read.table(file="student3.txt", sep=" ", header=TRUE, na.strings="-")
```

- 문자를 NA 처리



3. 데이터 입출력

2) read.csv() 함수 이용

- 구분자 ","가 sep의 기본값이며, header=TRUE가 기본값

```
student4 <- read.csv(file="student4.txt", sep=";", na.strings="-")
```

```
student4
```

```
student4 <- read.csv(file.choose(), sep=";", na.strings="-") # 파일열기
```




3. 데이터 입출력

3) read.xlsx() 함수를 이용해서 엑셀데이터 읽어오기

xlsx 패키지 설치

```
install.packages("xlsx")
```

```
install.packages("rJava")
```

java의 설치경로를 지정한다.

```
Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jre1.8.0_31')
```

```
library(rJava) # 로딩
```

xlsx 패키지 로드 시 주의사항

```
library(xlsx) # rJava를 로드하기 때문에 rJava 패키지 설치 필요
```

```
studentex <- read.xlsx(file.choose(), sheetIndex=1, encoding="UTF-8")
```

#한글인코딩, file="C:/Rwork/Part-I/studentexcel.xlsx"

```
studentex
```



3. 데이터 입출력

4) 웹문서 가져오기

```
install.packages("XML")
```

```
library(XML)
```

```
# 주소확인 : http://www.infoplease.com/ipa/A0104652.html
```

```
# 미국의 각 주별 1인당 소득자료
```

```
info.url <- "http://www.infoplease.com/ipa/A0104652.html"
```

```
# readHTMLTable() 함수 역할 - <table>, <tr>, <td> 태그 이용
```

```
info.df <- readHTMLTable(info.url, header=T, which=1, stringsAsFactors=F)
```

```
# header=T : 컬럼명 있음, which=1 : 첫번째
```

```
# stringsAsFactors 문자는 범주(값의 목록)처리 안함
```

```
info.df
```

```
str(info.df) # ipContentTable:'data.frame': 58 obs. of 12 variables:
```

```
dim(info.df) #[1] 54 12
```

```
names(info.df) # table 제목 출력
```



3. 데이터 입출력

- 데이터 저장하기

- (1) 화면(콘솔) 출력

- 1) cat() 함수

- ```
x <- 10
```

- ```
y <- 20
```

- ```
z <- x * y
```

- ```
cat("x*y의 결과는 ", z, " 입니다.\n") # \n 줄바꿈
```

- ```
cat("x*y = ", z)
```

- 2) print() 함수

- ```
print(z) # 변수 또는 수식만 가능
```



3. 데이터 입출력

(2) 파일에 데이터 저장

1) sink() 함수를 이용 파일 저장

setwd("C:/Rwork/output") # 현재 작업디렉토리 확인/폴더 생성

sink("savework.txt") # 저장할 파일명

파일 선택 창에서 studentexcel.xlsx 파일 선택

studentx <- read.xlsx(file.choose(), sheetIndex=1, encoding="UTF-8")

studentx #출력되는 값이 화면에 나타나지 않고 파일에 저장됨

sink() # 해제



3. 데이터 입출력

2) write.table()함수 이용 파일 저장

getwd()

① 기본옵션으로 저장 - 행 이름과 따옴표 붙음

```
write.table(studentx, "stdt.txt") # 행 번호와 따옴표 출력
```

② 행 이름 제거하여 저장

```
write.table(studentx, "stdt2.txt", row.names=FALSE) # 행번호 제거
```

③ 따옴표 제거하여 저장

```
write.table(studentx, "stdt3.txt", row.names=FALSE, quote=FALSE)
```

```
# 행 이름과 따옴표 제거
```



3. 데이터 입출력

3) write.xlsx() 함수 이용 파일 저장

- 엑셀 파일로 데이터 저장 함수

library(xlsx) # excel data 입출력 함수 제공

studentexcel.xlsx 파일 선택

st.df <- read.xlsx(file.choose(), sheetIndex=1, encoding="UTF-8")

str(st.df) # data.frame

write.xlsx(st.df, "studentx.xlsx") # excel형식으로 저장



3. 데이터 입출력

4) write.csv() 함수 이용 파일 저장

data.frame 형식의 데이터를 csv 형식으로 저장

```
setwd("C:/Rwork/Part-I")
```

```
st.df
```

```
write.csv(st.df,"stdf.csv", row.names=F, quote=F) # 행 이름 제거
```