

# 5. 입력과 출력

Input & Output

# 강의 목표

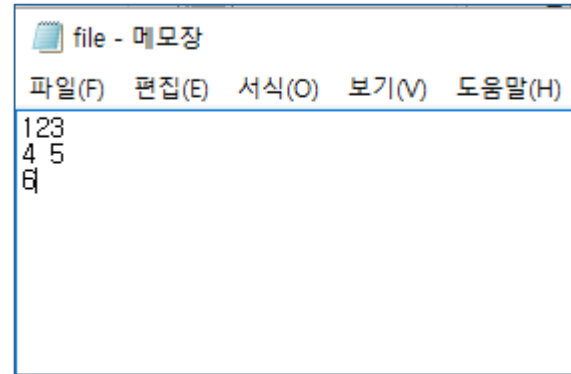
- ▶ 기본 입출력 과정과 방법을 이해한다
- ▶ R 또는 패키지에서 제공되는 데이터세트를 사용하는 방법을 이해한다.
- ▶ File을 읽고 쓰는 과정과 방법을 이해한다

# Input & Output

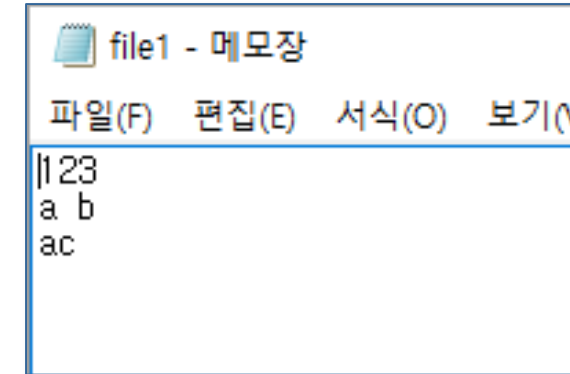
# Input

## ➤ Scan()

- 기초적인 입력 명령어
- File도 읽어올 수 있음
  - Scan("file.txt")
  - 기본적으로 numeric data만 입력 받음
  - Character data를 입력 받을 경우 what=""을 추가
  - what 안에 무엇이 들어가도 상관 없음
  - sep = "" 에서 문장의 끝을 구분할 기준 값을 설정할 수 있음
- Scan("")을 입력하면 키보드에서 값을 직접 입력 받을 수 있음
  - Character 데이터를 입력하고 싶다면 역시 what=""을 줘야 함
  - 입력을 끝내고 싶다면 마지막 줄에 빈 줄을 줌

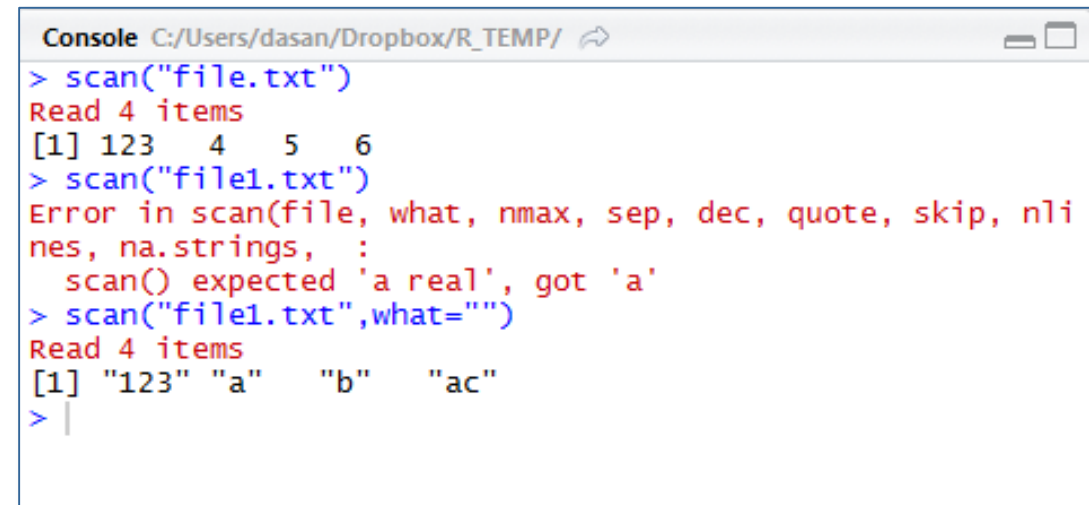


```
file - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
123
4 5
a
```



```
file1 - 메모장
파일(F) 편집(E) 서식(O) 보기(V)
123
a b
ac
```

```
> scan("")
1: 1 2 3
4: 3 4
6: 5
7:
Read 6 items
[1] 1 2 3 3 4 5
```



```
Console C:/Users/dasan/Dropbox/R_TEMP/
> scan("file.txt")
Read 4 items
[1] 123 4 5 6
> scan("file1.txt")
Error in scan(file, what, nmax, sep, dec, quote, skip, nlines, na.strings, :
  scan() expected 'a real', got 'a'
> scan("file1.txt",what="")
Read 4 items
[1] "123" "a" "b" "ac"
> |
```

# Input

## ➤ readline()

- 한 줄 단위로 입력을 받는 명령어
- readline()을 입력하면 한 줄짜리 character 값 입력 가능
  - 괄호()안에 “”를 쓰고 사이에 문장을 넣으면 문장 먼저 출력 후 입력을 받음
  - 단, 입력 값은 항상 character 값임

```
> a<-readline("수를 입력하세요: ")
수를 입력하세요: 12
> class(a)
[1] "character"
```

# Output

## ➤ print()

- 기초적인 출력명령어
- 변수 이름만 실행해도 출력이 되지만, 나중에 배울 for문이나 함수 안에서는 출력명령어를 써주지 않을 경우 출력이 되지 않음
- 한번에 하나의 객체만 출력 가능함

## ➤ cat()

- 기초적인 출력명령어
- 여러 객체 값을 출력할 수 있음
- \t, \n 등의 연산자 적용가능
- 출력할 때 한 칸씩 띄어쓰기가 적용됨
  - sep="" 으로 개체간 구분자 설정 가능

```
> print(1,2)
[1] 1
> cat(1,2)
1 2
```

```
> a<-1:10
> b<-2:12
> print(a,b)
[1] 1 2 3 4 5 6 7 8 9 10
> cat(a,b)
1 2 3 4 5 6 7 8 9 10 2 3 4 5 6 7 8 9 10 11 12
```

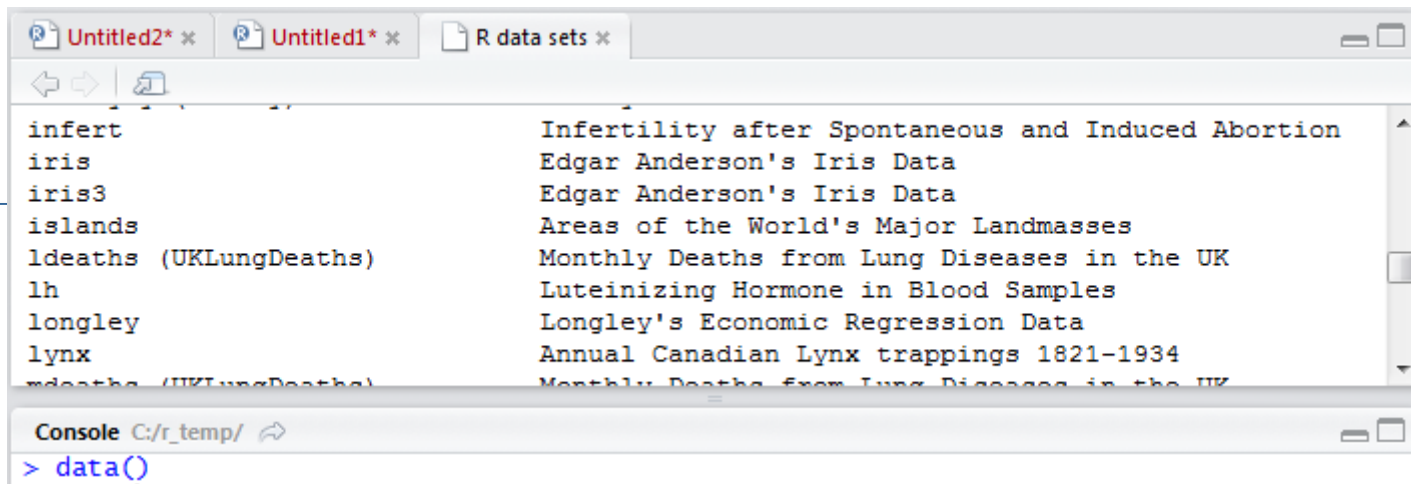
# Load Dataset

# Load Dataset

- ▶ R은 통계를 위한 언어로 R의 기능을 테스트하기 위한 R 자체에서 제공하는 내부 dataset이 존재
  - data() 명령어를 사용하면 현재 사용 가능한 데이터세트 목록을 보여줌
  - 다음은 내부 dataset인 iris의 예제임
  - data(iris)를 사용하면 RStudio에 data frame으로 읽어옴
  - 아무 명령어 없이 iris만 입력해도 읽어올 수 있음

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4          0.2  setosa
2          4.9         3.0          1.4          0.2  setosa
3          4.7         3.2          1.3          0.2  setosa
4          4.6         3.1          1.5          0.2  setosa
5          5.0         3.6          1.4          0.2  setosa
6          5.4         3.9          1.7          0.4  setosa

> str(iris)
'data.frame':  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```





# Load Dataset

- ▶ 그 외에도 많은 통계 패키지 또는 머신러닝 패키지들이 데모 목적의 데이터세트를 함께 제공함
  - 이들 데이터는 R 자체 내장 데이터세트가 아닌 Package를 설치하여 사용해야 하는 데이터세트임.
  - 이들 데이터세트를 사용하여 패키지가 제공하는 함수들을 사용해 볼 수 있음.  
ex) 기계학습 벤치마킹 데이터 저장한 mlbench, 또는 다양한 통계 연습 데이터 및 함수를 저장한 MASS
  - `install.packages` 명령어로 먼저 패키지를 설치하고 `library` 명령어로 패키지를 로드해야 함  
`>install.packages("MASS")`  
`>library("MASS")`
  - `data()`로 사용 가능한 데이터세트 확인 후 `?data` 로 그 데이터세트 사용법 확인  
`>?Cars93`

# File I/O

# File Input – scan()

- R은 보통 데이터를 파일 형태 또는 database 로부터 읽어들이어 처리함.
- scan() : 단순한 벡터 형태의 파일 데이터를 읽어들이는 함수
  - scan(“파일이름”) 으로 불러올 수 있으며 읽은 결과값은 vector 형태임
  - scan(“파일이름”)이면 numeric값만 입력 가능
  - scan(“파일이름”,what=“”)이면 문자 값까지 읽어올 수 있음

인수	설명
file	파일을 불러올 경로를 입력한다.
what	입력될 데이터의 유형을 지정한다.
sep	데이터 구분 기호를 입력한다. 기본값은 공백문자(띄어쓰기 또는 TAB)이다.
skip	데이터를 불러오는 과정에서 제외할 최대 행의 수를 지정한다. 예를 들어 2를 입력하면 세 번째 행부터 입력이 시작된다.
nlines	불러들일 최대 행의 수를 지정한다. 예를 들어 2를 입력하면 두 번째 행까지만 입력을 받게 된다.
na.strings	R에서 결측값으로 인식할 데이터의 형태를 입력한다.

# File Input – scan()

## ▶ scan() 함수 사용 예

```
> b<-scan("android_151211.csv",sep="\n",what="")
Read 25 items
> head(b)
[1] ",Top in Android Apps,Top paid in Android Apps,Top Grossing Android Apps,Top in Games,Top Paid in Games,Top Grossing Games"

[2] "Top 1, Facebook Messenger , Nova Launcher Prime , Clash of Clans , Kill Shot Bravo , Minecraft: Pocket Edition , Clash of Clans "
[3] "Top 2, Facebook , Minecraft: Pocket Edition , Game of War - Fire Age , Blossom Blast Saga , Minecraft: Story Mode , Game of War - Fire Age "
[4] "Top 3, Pandora® Radio , Minecraft: Story Mode , Candy Crush Saga , Temple Run 2 , Bloons TD 5 , Candy Crush Saga "
[5] "Top 4, Snapchat , Bloons TD 5 , Candy Crush Soda Saga , Triple Double Slots Slots , Geometry Dash , Candy Crush Soda Saga "
[6] "Top 5, Instagram , Geometry Dash , Clash of Kings , Battle Tales , Lifeline , Clash of Kings "

> class(b)
[1] "character"
```

# File Input – read.csv()

## ➤ CSV 파일이란?

- A comma-separated values (CSV) file stores tabular data (numbers and text) in plain text.
- Each line of the file is a data record.
- Each record consists of one or more fields, separated by commas.
- The use of the comma as a field separator is the source of the name for this file format.
- Comma Separated Value의 약자로 ‘,’를 기준으로 열을 구분하는 파일로 보통 엑셀을 활용하여 파일을 읽거나 생성함
- 일반 메모장에서도 파일을 열 수 있으며, database 등과도 포맷 변환이 가능함
- ‘,’는 데이터를 구분하기 위한 구분자 (delimiter) 또는 separator 로서 CSV 포맷은 ‘,’로 구분되나 텍스트 파일에서 데이터를 구분하기 위한 구분자로 , space(" "), tab(\t), ‘:’, ‘;’, ‘.’ 등 다양한 구분자를 사용할 수 있음. ex) tsv 파일

Data Table

Year	Make	Model	Description	Price
1997	Ford	E350	ac, abs, moon	3000.00
1999	Chevy	Venture "Extended Edition"		4900.00
1999	Chevy	Venture "Extended Edition, Very Large"		5000.00
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.00

CSV format

```
Year,Make,Model,Description,Price
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""",,,4900.00
1999,Chevy,"Venture ""Extended Edition, Very Large""",,5000.00
1996,Jeep,Grand Cherokee,"MUST SELL!
air, moon roof, loaded",4799.00
```

# File Input – read.csv()

- csv 파일을 Data frame으로 읽는 함수
- 다양한 옵션을 제공해 파일을 읽어오는 과정에서 데이터 전처리를 도와주고 있음.
- file = : 읽어올 file 경로를 포함한 이름값 넘겨주기
- header = : 파일의 첫 행을 열 이름으로 가져 올 것인지 여부
- sep = : 행을 구분짓는 기준을 정해줌. csv파일은 ,(comma)를 기준으로 열을 구분짓기 때문에 default값은 ,임. 하지만 나중에 .tsv파일을 읽으실 때 열 구분 기준값이 \t(tab)이기 때문에 sep = "\t" 로 줘야 함.
- stringsAsFactors = : 이 값을 False로 주지 않으면 모든 character 값들이 factor 값으로 들어오게 됨.
- Na.strings = : NA값으로 처리할 string들을 처리해줄 수 있음
- fileEncoding = : 불러읽어들일 파일의 인코딩을 지정해 줄 수 있음.

# File Input – read.csv()

- ▶ csv 파일을 Data frame으로 읽는 함수로 데이터를 읽을 때 가장 빈번하게 사용하는 함수임.
- ▶ 다양한 옵션을 제공해 파일을 읽어오는 과정에서 데이터 전처리를 도와주고 있음.

```
> a<-read.csv("imdb1.csv",header=T,stringsAsFactors = F)
> head(a,5)
  X      1.\n      The Shawshank Redemption\n      Rank...Title IMDb.Rating
1 1 \n      1.\n      The Shawshank Redemption\n      (1994)\n      9.2
2 2      \n      2.\n      Daeboo\n      (1972)\n      9.2
3 3 \n      3.\n      The Godfather: Part II\n      (1974)\n      9.0
4 4      \n      4.\n      The Dark Knight\n      (2008)\n      8.9
5 5      \n      5.\n      12 Angry Men\n      (1957)\n      8.9
> class(a)
[1] "data.frame"
```

# File Input – read.csv()

## ➤ read.csv() 인수

인수	설명
file	경로를 포함한 파일명
header	파일의 첫 행을 헤더로 처리할 것인지 여부로 default 가 TRUE 임
sep	열 구분자를 지정해 주는 인수로 read.csv 함수는 default 값이 ‘,’임
na.strings	데이터에 결측치가 포함되어 있을 경우 R의 NA에 대응시킬 값을 지정하는 인수 일상 데이터의 경우 값이 없는 경우 NA로 채워져 있기 보다는 빈칸 또는 -999, 0 등 NA를 대신하는 값들로 이루어져 있는데 이들을 NA로 매칭할 수 있음.
stringsAsFactors	문자열을 팩터로 저장할지 또는 문자열로 저장할 지 여부를 지정하는 데 사용함. 디폴트가 TRUE임
fileEncoding	불러읽어들일 파일의 인코딩을 지정해 줄 수 있음. ex) fileEncoding="UTF-8"
row.names	불러들일 file의 선택한 위치의 열을 행이름으로 지정함 ex) row.names=1 이면 첫번째 열을, row.names=2이면 두번째 열을 행이름으로 함
col.names	불러들일 file의 선택한 위치의 행을 열이름으로 지정함 따라서 colnames=1과 header=T는 같은 의미임



# File Input – read.table()

- ▶ 테이블 형식으로 저장된 파일을 Data frame으로 불러옴
- ▶ Table형태로 된 data들을 읽을 때 사용
- ▶ 단, 모든 행과 열이 같은 개수를 가지고 있어야 함

```
> c<- read.table("imdb1.csv",sep=",")
> head(c)
  v1                                     v2          v3
1 NA                                     Rank & Title IMDb Rating
2 1 \n      1. \n      The Shawshank Redemption\n      (1994)\n      9.2
3 2 \n      \n      2. \n      Daeboo\n      (1972)\n      9.2
4 3 \n      3. \n      The Godfather: Part II\n      (1974)\n      9
5 4 \n      \n      4. \n      The Dark Knight\n      (2008)\n      8.9
6 5 \n      \n      5. \n      12 Angry Men\n      (1957)\n      8.9
> class(c)
[1] "data.frame"
```

# File Input – read.table()

- read.csv() vs. read.table()
  - read.csv()는 read.table()에 기반을 둔 함수로서, .csv파일은 각 열마다 구분을 ","(comma)로 구분하는데, 매번 read.table()로 할 때마다 sep=","을 주기 귀찮고, .csv파일이 많기 때문에 별도로 준 것임.
  - 동일한 이유로 read.delim()이라는 함수가 존재하는데, 이것은 "\t"(tab)키로 열을 구분하는 .tsv파일을 읽기 위함.
- '함수명()'에서 뒤에 가로만 빼고 '함수명'만 적으시면 해당 파일을 .R형태로 제공하는 경우에 함수 내용을 볼 수 있음. ex) >read.csv
- 따라서 read.table이 상위 함수이고, 많이 사용하는 .csv나 .tsv 파일을 읽기 위해 read.csv와 read.delim함수를 따로 만들어 놓은 것임.

```
> read.csv
function (file, header = TRUE, sep = ",", quote = "\"", dec = ".",
  fill = TRUE, comment.char = "", ...)
read.table(file = file, header = header, sep = sep, quote = quote,
  dec = dec, fill = fill, comment.char = comment.char, ...)
```

# File Input – read.xlsx()

## ➤ read.xlsx()

- 엑셀 파일(.xlsx)을 읽기 위한 명령어
- “xlsx”이라는 package를 설치해야 실행가능
- read.xlsx(“파일명”, sheetIndex=“, sheetName=“)으로 사용함
  - 일반적으로, sheetIndex나 sheetName 둘 중 하나를 사용하여 load함
  - 둘 다 지정을 안 해 줄 경우, 에러가 발생하고 파일을 읽지 못함

```
> library(xlsx)
> d<-read.xlsx("stkbidu.xlsx",sheetIndex = 1, stringsAsFactors=F)
> head(d)
   open  high  low close volume adjusted      day company
1 138.32 138.78 135.31 137.53 2994700   137.53 2015-10-01  BIDU
2 136.60 149.09 136.31 148.51 6124400   148.51 2015-10-02  BIDU
3 149.55 150.38 145.50 149.62 3717400   149.62 2015-10-05  BIDU
4 149.28 154.47 148.13 149.80 3164200   149.80 2015-10-06  BIDU
5 146.00 149.00 143.81 144.77 6780400   144.77 2015-10-07  BIDU
6 144.00 144.17 139.90 141.26 5520000   141.26 2015-10-08  BIDU
> class(d)
[1] "data.frame"
```

# File Input – readHTMLTable()

- 해당 함수는 XML이라는 패키지에 있음.
- readHTMLTable( " 홈페이지 주소 ")로 사용하시면 해당 페이지의 표에 들어 있는 정보를 저장 가능하며, 일종의 웹 크롤링임.
- 웹 페이지에 있는 모든 표를 Data frame으로 가져오고, 각각의 표를 List에 담아 제공함.
- 따라서 첫번째에 있는 표를 출력하고 싶으시다면 [[1]]을 붙여주시면 됨.
- 다음은 "http://www.worldometers.info/world-population/" 에서 표를 가져오는 예임.

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

```
> library(XML)
```

```
> world_pop <- readHTMLTable("http://www.worldometers.info/world-population/")
```

## 홈페이지의 표 예제

World Population Forecast

Year	Population	Yearly % Change	Yearly Change	Median Age	Fertility Rate	Density (P/Km <sup>2</sup> )	Urban Pop %	Urban Population
2020	7,758,156,792	1.09 %	81,736,939	31	2.47	60	55.9 %	4,338,014,924
2025	8,141,661,007	0.97 %	76,700,843	32	2.43	63	57.8 %	4,705,773,576
2030	8,500,766,052	0.87 %	71,821,009	33	2.38	65	59.5 %	5,058,158,460
2035	8,838,907,877	0.78 %	67,628,365	34	2.35	68	61 %	5,394,234,712
2040	9,157,233,976	0.71 %	63,665,220	35	2.31	70	62.4 %	5,715,413,029
2045	9,453,891,780	0.64 %	59,331,561	35	2.28	73	63.8 %	6,030,924,065
2050	9,725,147,994	0.57 %	54,251,243	36	2.25	75	65.2 %	6,338,611,492

# File Input

- ▶ R은 현재 자체적으로 다음의 네가지 형태의 파일을 읽어들이 수 있음.
  - files ending '.R' or '.r' are source()d in, with the R working directory changed temporarily to the directory containing the respective file.  
ex) `source("myfile.R")` # load and execute a script of R commands
  - files ending '.RData' or '.rda' are load()ed.
  - files ending '.tab', '.txt' or '.TXT' are read using `read.table(..., header = TRUE, as.is=FALSE)`, and hence result in a data frame.
  - files ending '.csv' or '.CSV' are read using `read.table(..., header = TRUE, sep = ";", as.is=FALSE)`, and also result in a data frame.
- ▶ 이 외의 SAS, SPSS, Excel 등의 다른 데이터 포맷은 관련 패키지를 설치한 후 읽어들이면 됨.
  - 다음의 URL 참조 : R Data Import Tutorial
  - <https://www.datacamp.com/community/tutorials/r-data-import-tutorial>

# File output – save()

- ▶ .Rdata 형식의 file을 만들어주는 함수
  - 여러 변수들은 하나의 file에 넣어서 저장 가능함.
  - load()로 읽고, 여러 변수를 한번에 받을 수 있음.
  - 다음의 예제는 java라는 변수와 later라는 변수를 student.Rdata라는 파일에 담아서 저장함  
    > save(java, later, file="students.RData")
  - load로 읽어들이м  
    > load("student.RData")

# File output – write.table()

- table 형태의 2차원 데이터를 파일을 생성해서 저장함
- txt, csv, tsv 등의 format 가능
- `write.table(variable, "filename", sep=" ", row.names=)`  
`> write.table(later, "test.txt", sep=" ", row.names=F)`

## ➤ 주요 인수

- Variable : 데이터를 가지고 있는 변수
- Filename : 생성하고자 하는 파일이름, 확장자까지 포함해서 적어야 함
- sep=" " : csv나 tsv로 만들고자 할 경우, 열을 구분하는 기준을 설정할 수 있음
- row.names = : rowname을 파일에 저장할건지를 정하는 인수, T면 저장 F면 저장 안함
  - 일반적으로 row.names=F로 하는 경우가 많음.
  - 나중에 이 파일을 읽는다면 rowname이 1열이 되어 들어오기 때문
- quote: 문자열의 따옴표를 없애주는 옵션으로 quote=T는 따옴표를 넣겠다는 뜻이고 quote=F는 따옴표를 넣지 않겠다는 뜻임.

# File output – write.csv()

## ▶ 데이터 프레임을 csv로 저장함

```
> write.csv(java, "java.csv", row.names=F)
```

## ▶ 메모리 객체 삭제

- 메모리에 있는 모든 객체의 삭제

```
> rm(list=ls())
```

- 메모리에 있는 원하는 객체만 삭제 : rm(객체명)

```
> rm(java)
```



# 참고문헌

- ▶ R을 활용한 데이터 분석 - 김성근
- ▶ R 리뷰 - 서진수

END

Thank you for your attention



경청해주셔서 감사합니다