



[4]DataSet



DataSet



- 파일 IO
- 내장 데이터
- R 작업공간
- 데이터 속성 접근
- Type 판별, 변경

텍스트 파일 입력



- 텍스트 파일 읽기

**read.table(file, header = F, sep = “” , skip =, nrow= ..,
stringsAsFactor = FALSE, na.strings = “ ”)**

- file : “파일명”, “URL”
- header : 컬럼명 포함여부
- sep: 구분자
- skip : 건너뛴 라인수
- nrow : 읽는 라인수
- stringsAsFactor : 문자형 데이터를 factor 타입으로 할지 여부
- na.strings : # 결측값 표시, “”, “.”, “NA”

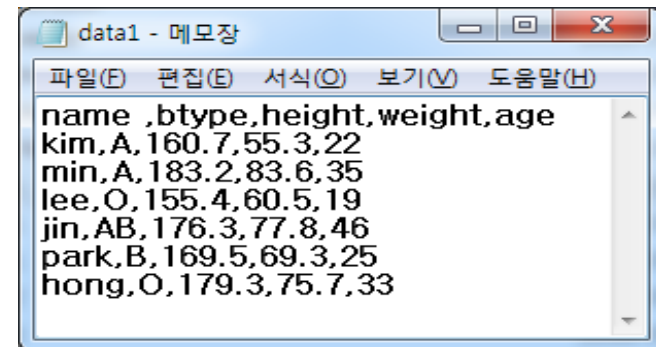
텍스트 파일 입력



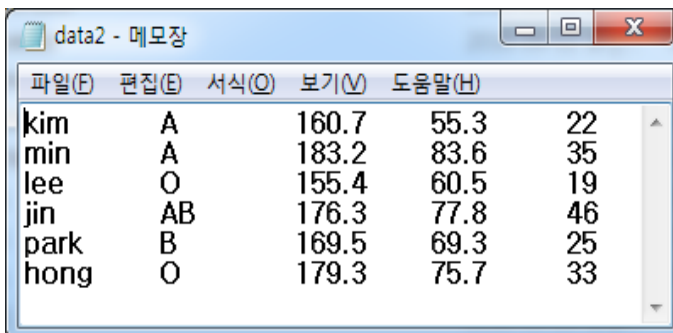
- (1) `read.table("c:/data/data1.txt", head=T, sep="")`
: c:/data/data1.txt파일을 read, 헤더포함, 구분자는 공백
- (2) `read.table("data1.txt", head=T, sep=",")`
: 현재 작업폴더의 data1.txt파일 read, 구분자는 콤마
- (3) `read.table("data2.txt", sep="")`
: data2.txt파일 read, 헤더없이 공백으로 구분
- (4) `read.table("data3.txt", "", head=T, sep="", skip = 1)`
: data3.txt파일 read, 헤더포함 공백으로 구분, 한줄 제외



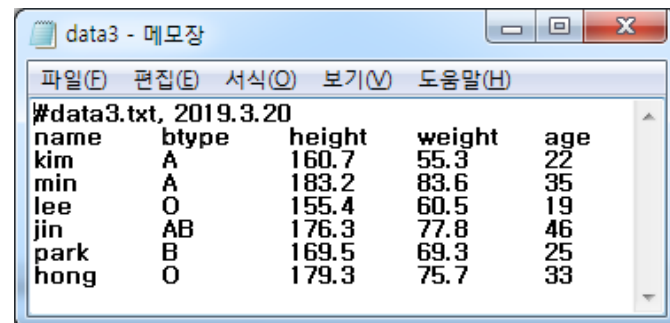
name	btype	height	weight	age
kim	A	160.7	55.3	22
min	A	183.2	83.6	35
lee	O	155.4	60.5	19
jin	AB	176.3	77.8	46
park	B	169.5	69.3	25
hong	O	179.3	75.7	33



name	btype	height	weight	age
kim,A	160.7	55.3	22	
min,A	183.2	83.6	35	
lee,O	155.4	60.5	19	
jin,AB	176.3	77.8	46	
park,B	169.5	69.3	25	
hong,O	179.3	75.7	33	



kim	A	160.7	55.3	22
min	A	183.2	83.6	35
lee	O	155.4	60.5	19
jin	AB	176.3	77.8	46
park	B	169.5	69.3	25
hong	O	179.3	75.7	33



#data3.txt, 2019.3.20				
name	btype	height	weight	age
kim	A	160.7	55.3	22
min	A	183.2	83.6	35
lee	O	155.4	60.5	19
jin	AB	176.3	77.8	46
park	B	169.5	69.3	25
hong	O	179.3	75.7	33

텍스트 파일 출력



- 텍스트 파일 저장

write.table(x, file = "", append = F, quote = T, sep = " ",...)

– x : 데이터프레임, file: 파일명, append : 추가, quote : “” 추가 여부, sep : 구분자

예>

– write.table(d1, “c:/data/mydata1.txt”,quote=F, sep=“:”)

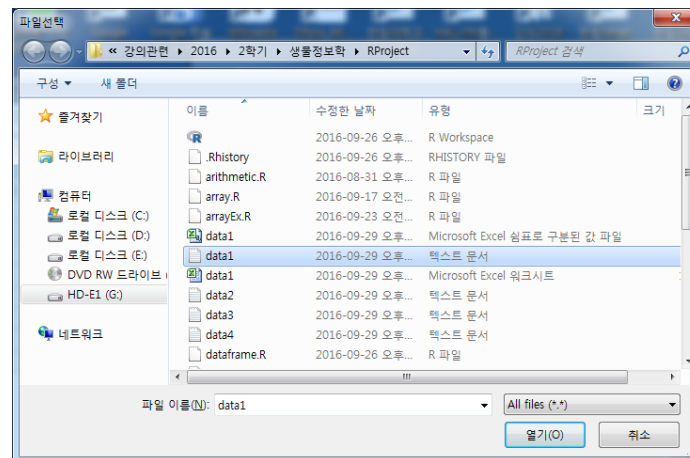
:d1 데이터프레임을 c:/data/mydata1.txt파일로 write, “”없이, “:”으로 분리

– write.table(d1, “mydata2.txt”,quote=F, append = T)

:현재 작업폴더의 mydata2.txt파일에 d1 write, “”없이 추가

- 파일 열기 창 이용

read.table(file.choose())



CSV 파일읽기



- **read.csv(file, header = FALSE, stringsAsFactor = FALSE, ..)**

```
>d = read.csv('./data/data.csv', header=T)
> str(d)
'data.frame':      6 obs. of  5 variables:
 $ name  : Factor w/ 6 levels "hong","jin","kim",...: 3 5 4 2 6 1
 $ btype : Factor w/ 4 levels "A","AB","B","O": 1 1 4 2 3 4
 $ height: num  161 183 155 176 170 ...
 $ weight: num  55.3 83.6 60.5 77.8 69.3 75.7
 $ age   : int  22 35 19 46 25 33
```

```
> d = read.csv('./data/data.csv', header=T, stringsAsFactor = FALSE)
> str(d)
'data.frame':      6 obs. of  5 variables:
 $ name  : chr  "kim" "min" "lee" "jin" ...
 $ btype : chr  "A" "A" "O" "AB" ...
 $ height: num  161 183 155 176 170 ...
 $ weight: num  55.3 83.6 60.5 77.8 69.3 75.7
 $ age   : int  22 35 19 46 25 33
```

name	btype	height	weight	age
kim	A	160.7	55.3	22
min	A	183.2	83.6	35
lee	O	155.4	60.5	19
jin	AB	176.3	77.8	46
park	B	169.5	69.3	25
hong	O	179.3	75.7	33

기타파일 읽기



#URL file open

```
wine = read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data", sep=',') # ,콤마로 분리된 파일 열기
```

```
n= c('class', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash',  
      'Magnesium', 'Total phenols', 'Flavanoids', 'Nonflavanoid phenols',  
      'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline')  
names(wine) = n #열이름 설정
```

#Json file

```
install.packages('jsonlite') #json file read package 설치  
library(jsonlite) #jsonlite 라이브러리 사용  
jsonData = fromJSON('./data/data.json') #json 파일 읽기
```

#Excel file

```
install.packages('readxl') #readxl 라이브러리 설치 및 사용  
library(readxl)  
xlsData = read_excel('./data/data.xlsx', sheet=1) #파일명, 시트번호
```

내장데이터 접근

- R에서 제공되는 다양한 내장 데이터 접근

- 내장데이터 목록 : data()
- 내장 데이터 가져오기 : data(데이터명)

예>

(1) data(sleep)

: 두 약물에 대한 10명의 학생들에 대한
초과수면시간 측정 데이터

	extra	group	ID
1	0.7	1	1
2	-1.6	1	2
3	-0.2	1	3
4	-1.2	1	4
5	-0.1	1	5
6	3.4	1	6
7	3.7	1	7
8	0.8	1	8
9	0.0	1	9
10	2.0	1	10

(2) data(iris)

: 붓꽃의 3가지 종(setosa, versicolor, virginica)에 대해 꽃받침(sepal)과 꽃잎(petal)의 길이를 정리한 데이터

	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
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa



내장데이터 접근

```
> data() #R 내장데이터 확인

> data(sleep) #sleep 내장데이터 가져오기
> str(sleep) #sleep 구조 확인
'data.frame':      20 obs. of  3 variables:
 $ extra: num  0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0 2 ...
 $ group: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ ID   : Factor w/ 10 levels "1","2","3","4",...: 1 2 3 4 5 6 7 8 9 10 ...
> head(sleep) #상위 6개 확인
  extra group ID
1  0.7     1  1
2 -1.6     1  2
3 -0.2     1  3
4 -1.2     1  4
5 -0.1     1  5
6  3.4     1  6
```

Data sets in package `j@datasets`:

AirPassengers	Monthly Airline Passenger Numbers 1949–1960
BJsales	Sales Data with Leading Indicator
BJsales.lead (BJsales)	Sales Data with Leading Indicator
BOD	Biochemical Oxygen Demand
CO2	Carbon Dioxide Uptake in Grass Plants
ChickWeight	Weight versus age of chicks on different diets
DNase	Elisa assay of DNase
EuStockMarkets	Daily Closing Prices of Major European Stock Indices,
...	



R 객체, 작업공간

- `objects()` : R 객체 확인
- `rm(객체명)` : 특정 객체 삭제
- `save(객체1, ..., file="파일명.RData")`
: R 객체를 파일로 저장(.RData)
- `setwd("경로명")` : 작업 디렉토리 설정
- `save.image()` : 작업공간 저장
- `ls()` : 작업공간 확인
- `rm(list =ls ())`
: 메모리 상에 있는 모든 객체를 삭제
- `load(".Rdata")`
: 파일로부터 객체 읽기

```
> #object save/load
> x=1:10
> y=c("A","B","C")
> z=matrix(1:9, nrow=3)
> objects()
[1] "x" "y" "z"
> save(x,y,z, file="xyz.RData")
> rm(x)
> objects()
[1] "y" "z"
> load("xyz.RData")
> objects()
[1] "x" "y" "z"
>
> #workspace save/load
> ls()
[1] "x" "y" "z"
> setwd("c:/Rworkdata")
> save.image()
> rm(list=ls())
> ls()
character(0)
> load(".RData")
> ls()
[1] "x" "y" "z"
```

데이터 속성 접근



- `attach()` : 데이터프레임이나 리스트 명 없이 접근 가능
- `detach()` : 이름 없이 접근하는 상태 해제
- `with(데이터명, { 처리.. })` : 데이터 명 없이 처리 구문 작성 가능

```
> #attach/detach
> sleep$extra
[1] 0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0.0 2.0 1.9 0.8 1.1 0.1 -0.1 4.4 5.5 1.6
[19] 4.6 3.4
> attach(sleep)
The following objects are masked from sleep (pos = 3):
    extra, group, ID
> extra
[1] 0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0.0 2.0 1.9 0.8 1.1 0.1 -0.1 4.4 5.5 1.6
[19] 4.6 3.4
> detach(sleep)
> with(sleep, {
+   extra
+   x1= sleep[extra<0,]
+ })
> x1
  extra group ID
2  -1.6     1  2
3  -0.2     1  3
4  -1.2     1  4
5  -0.1     1  5
15 -0.1     2  5
```

데이터 속성 접근



- 복합식 : { 식1; 식2; ... } 여러 개의 식을 묶는 방법, 줄 바꿈이나 “;”으로 식 구분

```
> data(sleep) #sleep 내장데이터 가져오기
> str(sleep)
'data.frame':      20 obs. of  3 variables:
 $ extra: num  0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0 2 ...
 $ group: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ ID   : Factor w/ 10 levels "1","2","3","4",...: 1 2 3 4 5 6 7 8 9 10 ...

> with(sleep, mean(extra)) == mean(sleep$extra)
[1] 1.54

> with(sleep, {
+   print(extra); print(group);
+   x1= sleep[extra<0,]
+   x1
+ })
[1] 0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0.0 2.0 1.9 0.8
[13] 1.1 0.1 -0.1 4.4 5.5 1.6 4.6 3.4
[1] 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
Levels: 1 2
  extra group ID
2  -1.6     1  2
3  -0.2     1  3
4  -1.2     1  4
5  -0.1     1  5
15 -0.1     2  5
```

타입 판별 / 변환



- 타입판별

- `mode()` : 데이터의 저장타입 , `class()` : 데이터의 객체타입
- ‘is.*’ 형태의 데이터 유형 검증 함수
 - `is.numeric(x)` 수치형 여부 , `is.na(x)` NA 여부
 - `is.double(x)` 실수형 여부 , `is.null(x)` NULL 여부
 - `is.integer(x)` 정수형 여부, `is.nan(x)` NaN 여부
 - `is.logical(x)` 논리형 여부, `is.infinite(x)` 무한 수치 여부
 - `is.complex(x)` 복소수형 여부, `is.finite(x)` 유한 수치 여부
 - `is.character(x)` 문자형 여부

- 타입변환

- ‘as.*’ 형태의 데이터 유형 변환 함수
 - `as.numeric(x)` : 수치형으로 변환 , `as.logical(x)` : 논리형으로 변환
 - `as.double(x)` : 실수형으로 변환, `as.complex(x)` : 복소수형으로 변환
 - `as.integer(x)` : 정수형으로 변환, `as.character(x)` : 문자형으로 변환



타입 판별 / 변환

```
> str(d1)
'data.frame': 6 obs. of 5 variables:
 $ name : Factor w/ 6 levels "hong","jin","kim",...: 3 5 4 2 6 1
 $ btype : Factor w/ 4 levels "A","AB","B","O": 1 1 4 2 3 4
 $ height: num 161 183 155 176 170 ...
 $ weight: num 55.3 83.6 60.5 77.8 69.3 75.7
 $ age : int 22 35 19 46 25 33
```

```
> #dataType/type conversion
```

```
> mode(d1)
```

```
[1] "list"
```

```
> mode(d1$name)
```

```
[1] "numeric"
```

```
> mode(d1$age)
```

```
[1] "numeric"
```

```
> class(d1)
```

```
[1] "data.frame"
```

```
> class(d1$name)
```

```
[1] "factor"
```

name	btype	height	weight	age
kim	A	160.7	55.3	22
min	A	183.2	83.6	35
lee	O	155.4	60.5	19
jin	AB	176.3	77.8	46
park	B	169.5	69.3	25
hong	O	179.3	75.7	33

dataframe : d1

```
> class(d1$age)
```

```
[1] "integer"
```

```
> is.list(d1)
```

```
[1] TRUE
```

```
> is.data.frame(d1)
```

```
[1] TRUE
```

```
> is.factor(d1$btype)
```

```
[1] TRUE
```

```
> is.numeric(d1$age)
```

```
[1] TRUE
```

```
> is.factor(d1$name)
```

```
[1] TRUE
```

```
>
```

```
> d1$name = as.character(d1$name)
```

```
> is.factor(d1$name)
```

```
[1] FALSE
```

연습문제



1) iris dataset을 이용하여 다음을 처리

- (1) Iris 의 객체타입, 크기, 구조 확인
- (2) 상위 10개 확인 후 파일(txt)로 저장
- (3) Species로 분리
- (4) 분리된 것을 iris2로 결합

2) <https://www.kaggle.com/datasets>

위에서 제공되는 데이터 셋을 받아서 데이터 설명과 속성을 정리하고
객체타입, 크기, 구조 , 상위 10개 확인하는 R script 작성