



#moon
all about moon

CATEGORY

oracle/R 프로그래밍

R에서 데이터 불러오기/저장하기

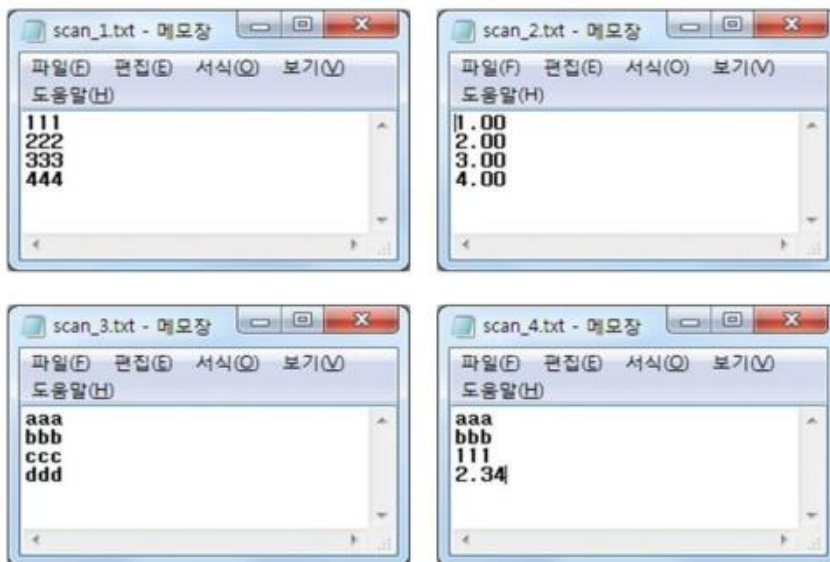
#moon 2014.11.28 10:05

[불러오기]

1. scan() 함수로 텍스트 파일 읽어서 배열에 저장하기

- c언어의 scanf()와 비슷한 역할
- 실수의 경우 소수점이 생략되어 입력되기때문에 what 옵션을 사용해야함
- 문자와 실수는 what 옵션을 사용해야함
- 특히 문자의 경우 what="" 옵션을 사용하지 않을경우 에러 발생
- scan() 함수에 입력값을 안주면 사용자에게 입력을 받겠다는 의미

접기



```
> setwd("d:\\r_temp")
```

```
> scan1 <- scan('scan_1.txt')
```

```
Read 4 items
```

```
> scan1
```

```
[1] 111 222 333 444
```

```
>
```



#moon
all about moon

CATEGORY

[1] 1 2 3 4

>

> # 실수와 문자를 호출할때 what 옵션 사용

> scan2 <- scan('scan_2.txt', what='')

Read 4 items

> scan2

[1] "1.00" "2.00" "3.00" "4.00"

>

> scan3 <- scan('scan_3.txt', what='')

Read 4 items

> scan3

[1] "aaa" "bbb" "ccc" "ddd"

>

> scan4 <- scan('scan_4.txt', what='')

Read 4 items

> scan4

[1] "aaa" "bbb" "111" "2.34"

>

> # 숫자입력받기

> input <- scan()

1: 1

2: 2

3: 3

4: # 그냥 엔터치면 입력 종료됨

Read 3 items

> input

[1] 1 2 3

>

> # 문자입력받기, what="" 옵션 사용해야함

> input2 <- scan(what='')

1: a

2: b

3:

Read 2 items

> input2

[1] "a" "b"

접기



#moon
all about moon

CATEGORY

접기

```
> input3 <- readline()
R is very fun! # 입력할 값을 직접 입력할수 있음
>
> input3
[1] "R is very fun!"
>
> input4 <- readline("Are you OK? : ")
Are you OK? : yes!!!!!!!!!!!!
>
> input4
[1] "yes!!!!!!!!!!!!"
>
```

접기

3. readLines() 함수로 파일 읽어들이어서 배열에 담기

- readLines() : 파일에서 데이터를 읽어서 배열로 만들어주는 함수

접기

```
> input5 <- readLines('scan_4.txt')
> input5
[1] "aaa" "bbb" "111" "2.34"
```

접기

4. read.table 일반 텍스트 형태의 파일을 읽어서 데이터프레임에 담기

- read.table : 데이터를 R로 불러들이고 데이터프레임에 담기
- header 파라미터 : 원본 데이터에 컬럼 이름이 있을 경우 헤더부분의 존재여부를 알려줌
- header 파라미터를 사용하지 않으면 컬럼이름(라벨명)이 자동으로 생성
- 라벨 구분이 공백인 경우는 read.table 사용하면 좋음, csv처럼 ,(콤마)인 경우는 read.csv 사용

접기



#moon
all about moon

CATEGORY

2	banana	200	2
3	peach	200	7
4	berry	50	9

```
> fruits <- read.table('fruits.txt')
```

```
> fruits
```

```
V1 V2 V3 V4
1 no name price qty
2 1 apple 500 5
3 2 banana 200 2
4 3 peach 200 7
5 4 berry 50 9
```

```
>
```

```
> # 헤더부분존재표시 header=T
```

```
> fruits <- read.table('fruits.txt', header=T)
```

```
> fruits
```

```
no name price qty
1 1 apple 500 5
2 2 banana 200 2
3 3 peach 200 7
4 4 berry 50 9
```

### Fruits Test Data			
1	apple	500	5
2	banana	200	2
### Null Test Row			
3	peach	200	7
4	berry	50	9

```
> # 주석은 자동 제외됨
```

```
> fruits2 <- read.table('fruits_2.txt')
```

```
> fruits2
```

```
V1 V2 V3 V4
1 1 apple 500 6
2 2 banana 200 2
3 3 peach 200 7
```



#moon
all about moon

CATEGORY

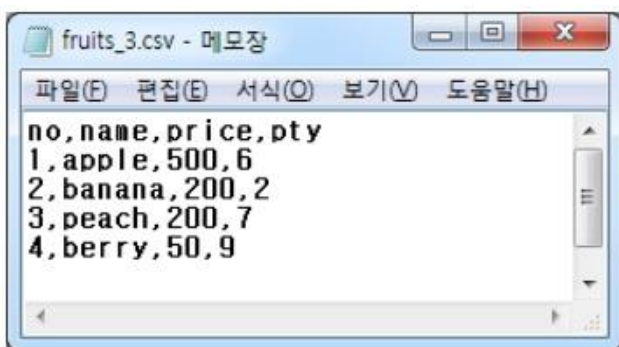
```
> fruits2 <- read.table('fruits_2.txt', skip=2)
> fruits2
  V1  V2 V3 V4
1 2 banana 200 2
2 3 peach 200 7
3 4 berry 50 9
>
> # 출력할 줄수를 지정 nrows=2
> fruits2 <- read.table('fruits_2.txt', nrows=2)
> fruits2
  V1  V2 V3 V4
1 1 apple 500 6
2 2 banana 200 2
```

접기

5. read.csv() 함수로 csv불러오기

- 라벨 구분이 ,(콤마)인 경우에 사용하기 간편함
- 라벨 구분이 tab으로 구분된 파일이라면 sep="\t" 옵션 사용 (read.delim()) 더편리함)
- 라벨명을 수동으로 지정 가능
- 원본 소스 파일에 라벨명(컬럼명)이 없다면 수동으로 지정해 주어야 함 (header=F)

접기

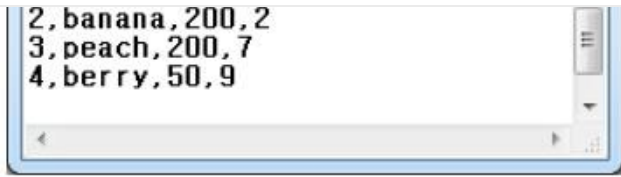


```
> fruit3 <- read.csv('fruits_3.csv')
> fruit3
  no name price qty
1 1 apple 500 6
2 2 banana 200 2
3 3 peach 200 7
4 4 berry 50 9
```



#moon
all about moon

CATEGORY



```
> fruit4 <- read.csv('fruits_4.csv')
> fruit4
  X1 apple X500 X6
1 2 banana 200 2
2 3 peach 200 7
3 4 berry 50 9
>
> # 컬럼명이 없으면 문제발생 하므로 수동으로 지정해야함 header=F
> fruit4 <- read.csv('fruits_4.csv', header=F)
> fruit4
  V1  V2 V3 V4
1 1 apple 500 6
2 2 banana 200 2
3 3 peach 200 7
4 4 berry 50 9
>
> # 라벨명을 사용자가 수동으로 지정
> label <- c('no', 'name', 'price', 'qty')
> fruit4 <- read.csv('fruits_4.csv', header=F, col.names=label)
> fruit4
  no name price qty
1 1 apple 500 6
2 2 banana 200 2
3 3 peach 200 7
4 4 berry 50 9
```

접기

6. xls (Microsoft Office Excel) 파일을 데이터프레임에 담기

- 엑셀 파일을 R로 불러와서 데이터 프레임에 저장하려면 패키지를 추가로 설치해야함
- 32bit용 엑셀은 RODBC 패키지 사용
- 64bit용 엑셀은 XLConnect 패키지 사용

접기

```
> # 필요한 패키지 설치 (실습용 컴퓨터는 32bit이므로 RODBC 패키지를 사용)
```



#moon
all about moon

CATEGORY

--- 현재 세션에서 사용할 CRAN 미러를 선택해 주세요 ---

URL 'http://cran.nexr.com/bin/windows/contrib/3.1/RODBC_1.3-10.zip'을 시도합니다

Content type 'application/zip' length 828501 bytes (809 Kb)

URL을 열었습니다

downloaded 809 Kb

패키지 'RODBC'를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다

C:\Users\stu\AppData\Local\Temp\RtmpIBJIY\downloaded_packages

> library(RODBC)

>

> setwd("d:\\r_temp")

>

> # excel이라는 변수에 연결함

> excel=odbcConnectExcel('fruits_6.xls', readOnly=F)

>

> # excel변수에 연결된것을 조회하여 data1에 담음

> data1=sqlTables(excel)

> data1

TABLE_CAT	TABLE_SCHEM	TABLE_NAME	TABLE_TYPE	REMARKS
1	d:\\r_temp\\fruits_6	<NA> Sheet1\$	SYSTEM TABLE	<NA>
2	d:\\r_temp\\fruits_6	<NA> Sheet2\$	SYSTEM TABLE	<NA>
3	d:\\r_temp\\fruits_6	<NA> Sheet3\$	SYSTEM TABLE	<NA>

>

> # excel의 sheet1\$의 데이터를 불러와서 data2에 담고 출력하기

> data2 = sqlQuery(excel, 'select * from [Sheet1\$]', as.is=T)

> data2

no	name	price	qty
1	1	apple	500 6
2	2	banana	400 4
3	3	peach	300 3
4	4	berry	200 2

접기

7. 클립보드의 내용을 사용해서 데이터프레임 생성하기

접기



#moon
all about moon

CATEGORY

```
no name price qty
1 1 apple 500 6
2 2 banana 400 4
3 3 peach 300 3
4 4 berry 200 2
```

접기

** pdf파일 참고

- . Oracle에 접속해서 데이터 가져오기
- . MySQL DBMS 내용 불러오기
- . Microsoft SQL Server 내용 불러오기

[저장하기]

8. write.table() 함수로 파일에 저장하기

- write.table() : data frame 형태의 데이터를 저장해주는 함수

접기

```
> fruits6
no name price qty
1 1 apple 500 6
2 2 banana 400 4
3 3 peach 300 3
4 4 berry 200 2
>
> write.table(fruits6, 'fruits6.txt')
> fruits6_2 <- read.table('fruits6.txt', header=T)
> fruits6_2
no name price qty
1 1 apple 500 6
2 2 banana 400 4
3 3 peach 300 3
4 4 berry 200 2
```

접기

8. excel 파일 형식으로 저장하기

- R에서 작업한 내용을 Excel 형식으로 저장가능
- R : WriteXLS / window : Perl 설치가 되어있어야함



#moon
all about moon

CATEGORY

> # WriteXLS 패키지 설치

> install.packages("WriteXLS")

Installing package into 'C:/Users/stu/Documents/R/win-library/3.1'

(as 'lib' is unspecified)

URL 'http://cran.nexr.com/bin/windows/contrib/3.1/WriteXLS_3.5.1.zip'을 시도합니다

Content type 'application/zip' length 712169 bytes (695 Kb)

URL을 열었습니다

downloaded 695 Kb

패키지 'WriteXLS'를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다

C:\Users\stu\AppData\Local\Temp\RtmpIBJIY\downloaded_packages

> library(WriteXLS)

> name <- c("Apple", "Banana", "Peach")

> price <- c(300,200,100)

> item <- data.frame(NAME=name, PRICE=price)

> item

NAME PRICE

1 Apple 300

2 Banana 200

3 Peach 100

>

> # 엑셀로 저장하기

> WriteXLS("item", "item.xls")

접기

9. XML 형식으로 저장하기

- R의 작업결과를 저장해서 웹브라우저를 통해 만드는 방법도 많이 사용
- R의 작업결과를 XML 방식으로 저장해야함
- XML 패키지와 kulife 패키지 사용 (다른패키지를 사용해도 됨)

접기

> # XML 패키지 설치

> install.packages("XML")

Installing package into 'C:/Users/stu/Documents/R/win-library/3.1'

(as 'lib' is unspecified)



#moon
all about moon

CATEGORY

downloaded 4.1 Mb

패키지 'XML'를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다

C:\Users\stu\AppData\Local\Temp\RtmpIBJIY\downloaded_packages

> library(XML)

>

> # kulife 패키지 설치

> install.packages("kulife")

Installing package into 'C:/Users/stu/Documents/R/win-library/3.1'

(as 'lib' is unspecified)

URL 'http://cran.nexr.com/bin/windows/contrib/3.1/kulife_0.1-14.zip'을 시도합니다

Content type 'application/zip' length 2151877 bytes (2.1 Mb)

URL을 열었습니다

downloaded 2.1 Mb

패키지 'kulife'를 성공적으로 압축해제하였고 MD5 sums 이 확인되었습니다

다운로드된 바이너리 패키지들은 다음의 위치에 있습니다

C:\Users\stu\AppData\Local\Temp\RtmpIBJIY\downloaded_packages

> library(kulife)

>

> # 데이터프레임을 생성

> name <- c('apple', 'banana', 'peach')

> price <- c(300,200,100)

> item <- data.frame(NAME=name, PRICE=price)

> item

NAME PRICE

1 apple 300

2 banana 200

3 peach 100

> # write.xml을 사용하여 xml형식으로 저장

> write.xml(item, file='item.xml')

접기



#moon

all about moon

CATEGORY

'oracle > R 프로그래밍' 카테고리의 다른 글

- R로 데이터를 원하는 모양으로 변형하기 ② (숫자... (0)
- R로 데이터를 원하는 모양으로 변형하기 ① (기본... (0)
- R에서 데이터 불러오기/저장하기 (0)
- R 기초 문법 ② (벡터, 행렬, list, 데이터 프레임) (0)
- R 기초 문법 ④ (R특징, 데이터형, 변수) (1)
- 패키지 관리하기 (0)

댓글 0

여러분의 소중한 댓글을 입력해주세요

이름

비밀번호

비밀글

입력

1

...

3

4

5

6

7

8

9

10

11

공지사항

최근에 올라온 글

최근에 달린 댓글

Total

○○○

as.data 포맷어떻게하죠?

128,898

Redis Sentinel

내용 중 결합인덱스 관련해 오..

Today0

MIME-Type List

좋은 정보 잘보고 갑니다

Yesterday68

git command

좋은 글 잘 보고 갑니다 ^^ 감..

링크

TAG

more

« 2019/05 »

글 보관함

조대협's 블로그

일 월 화 수 목 금 토

2016/03 (1)

MySQL DBA를 위한

1 2 3 4

2016/02 (1)

::개발자가 사는 세상::

5 6 7 8 9 10 11

2015/12 (1)

오라클 스터디

12 13 14 15 16 17 18

2015/11 (1)

일상다반사

19 20 21 22 23 24 25

오라클 취업반 올해 3기

26 27 28 29 30 31



#moon
all about moon



CATEGORY

Blog is powered by [Tistory](#) / Designed by [Tistory](#)