Excel using RODBC & XLConnect

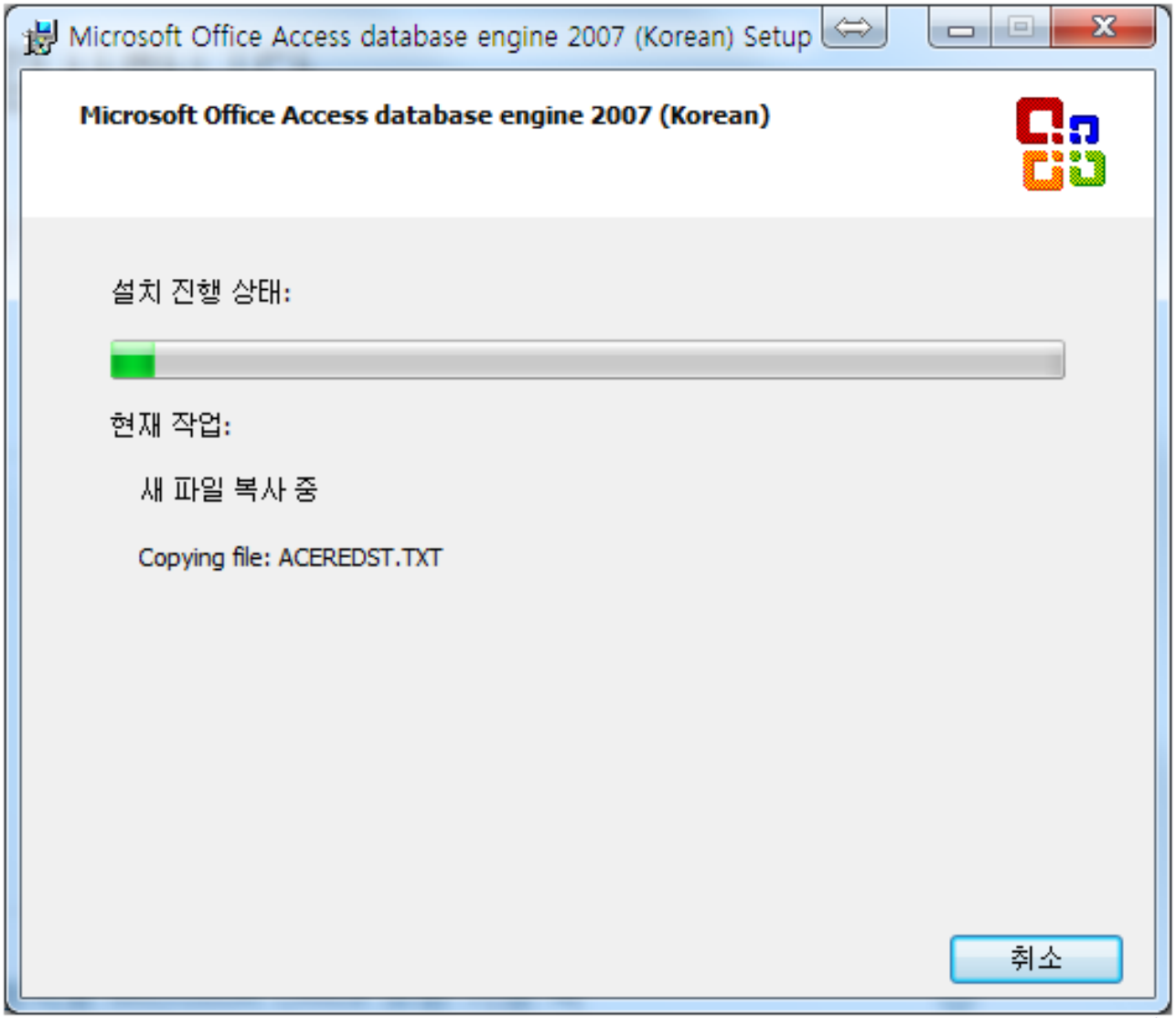
Jinseog Kim

Objectives

- ODBC, XLconnect를 이용하여 Excel 자료를 분석할 수 있다.
- sqldf를 이용하여 R data.frame을 핸들링할 수 있다.

ODBC excel driver의 설치

https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=13255



RODBC를 이용한 엑셀파일 핸들링

- 엑셀파일의 버전에 따라 *odbcConnectExcel*과 *odbcConnectExcel2007*를 제공

```
library(RODBC)
```

```
## Warning: package 'RODBC' was built under R version 3.2.5
```

```
odbcCloseAll()
```

```
odbcConnectExcel(xls.file, readOnly = TRUE, ...)
odbcConnectExcel2007(xls.file, readOnly = TRUE, ...)
```

예를 들어 *T:/data/test.xlsx*파일을 접속하는 코드는 아래와 같다.

```
con <- odbcConnectExcel2007("H:/data/pop_density.xlsx", readOnly=F)
```

ODBC: Excel 파일 연결정보 확인

```
odbcGetInfo(con)
```

```
##                         DBMS_Name                         DBMS_Ver
##                           "EXCEL"                      "12.00.0000"
##                  Driver_ODBC_Ver                  Data_Source_Name
##                           "03.51"                               ""
##                       Driver_Name                        Driver_Ver
##                      "ACEODBC.DLL" "Microsoft Access database engine"
##                          ODBC_Ver                      Server_Name
##                       "03.80.0000"                          "EXCEL"
```

ODBC: Excel 파일정보 확인

- sqlTables() : 접속한 엑셀파일의 정보를 보여주는 함수
  - **TABLE_NAME**항목은 엑셀파일에 있는 워크쉬트의 이름
  - 워크시트 이름이 실제 *연봉*이지만 *연봉$*로 출력됨
  - 쿼리문을 사용할 때는 *[연봉$]*를 이용

```
sqlTables(con)
```

```
##                       TABLE_CAT TABLE_SCHEM    TABLE_NAME   TABLE_TYPE REMARKS
## 1 H:\\data\\pop_density.xlsx        <NA> pop_density$ SYSTEM TABLE     <NA>
## 2 H:\\data\\pop_density.xlsx        <NA>         연봉$ SYSTEM TABLE     <NA>
```

ODBC: SQL을 이용한 데이터 검색

- sqlQuery()
  - 엑셀파일의 워크시트를 읽어오기 위한 RODBC 함수
  - 수행결과는 R 데이터프레임 객체 저장

```
a <- sqlQuery(con, "select * from [pop_density$]", as.is=T)
head(a)
```

```
##   region year   pop
## 1   전국 2011   497
## 2   서울 2011 16592
## 3   부산 2011  4498
## 4   대구 2011  2798
## 5   인천 2011  2659
## 6   광주 2011  2999
```

ODBC: SQL을 이용한 데이터 검색

```
b <- sqlQuery(con, "select * from [pop_density$] where year=2014", as.is=T)
head(b)
```

```
##   region year   pop
## 1   전국 2014   503
## 2   서울 2014 16343
## 3   부산 2014  4432
## 4   대구 2014  2784
## 5   인천 2014  2728
## 6   광주 2014  3025
```

ODBC: SQL UPDATE을 이용한 데이터 갱신

- 엑셀파일 접속시 옵션 *readOnly=F*를 사용
- 아래처럼 SQL의 *UPDATE*문을 사용

```
sqlQuery(con, "update [API$] set type='H' where id='016111901302229'")
```

연습문제

*Excel 소스: http://datamining.dongguk.ac.kr/data/test.xlsx

```
con = odbcConnectExcel2007("H:/data/test.xlsx", readOnly=F)
odbcGetInfo(con)
sqlTables(con)

a  <- sqlQuery(con, "select * from [bbb$]", as.is=T)
us <- sqlQuery(con, "select * from [us$]", as.is=T)
head(a)
head(us)

us1 <- sqlQuery(con, "SELECT * FROM [us$] WHERE Murder > 10 AND UrbanPop <= 50", as.
us1
```

XLConnect를 이용한 엑셀파일 핸들링

- XLConnect는 R에서 Microsoft Excel 데이터를 핸들링하기 위한 패키지로 다양한 OS에서 사용

```
library(XLConnect)
df <- readWorksheetFromFile("<file name and extension>",
                            sheet=1,
                            startRow = 4,
                            endCol = 2)
wb <- loadWorkbook("<name and extension of your file>")
df <- readWorksheet(wb, sheet=1)
```

- sheet : sheet name or index.
- startRow/startCol: row or column the data set should be imported,
- endRow/endCol
- region: range (eg A5:B5)

XLConnect를 이용한 엑셀파일 불러오기

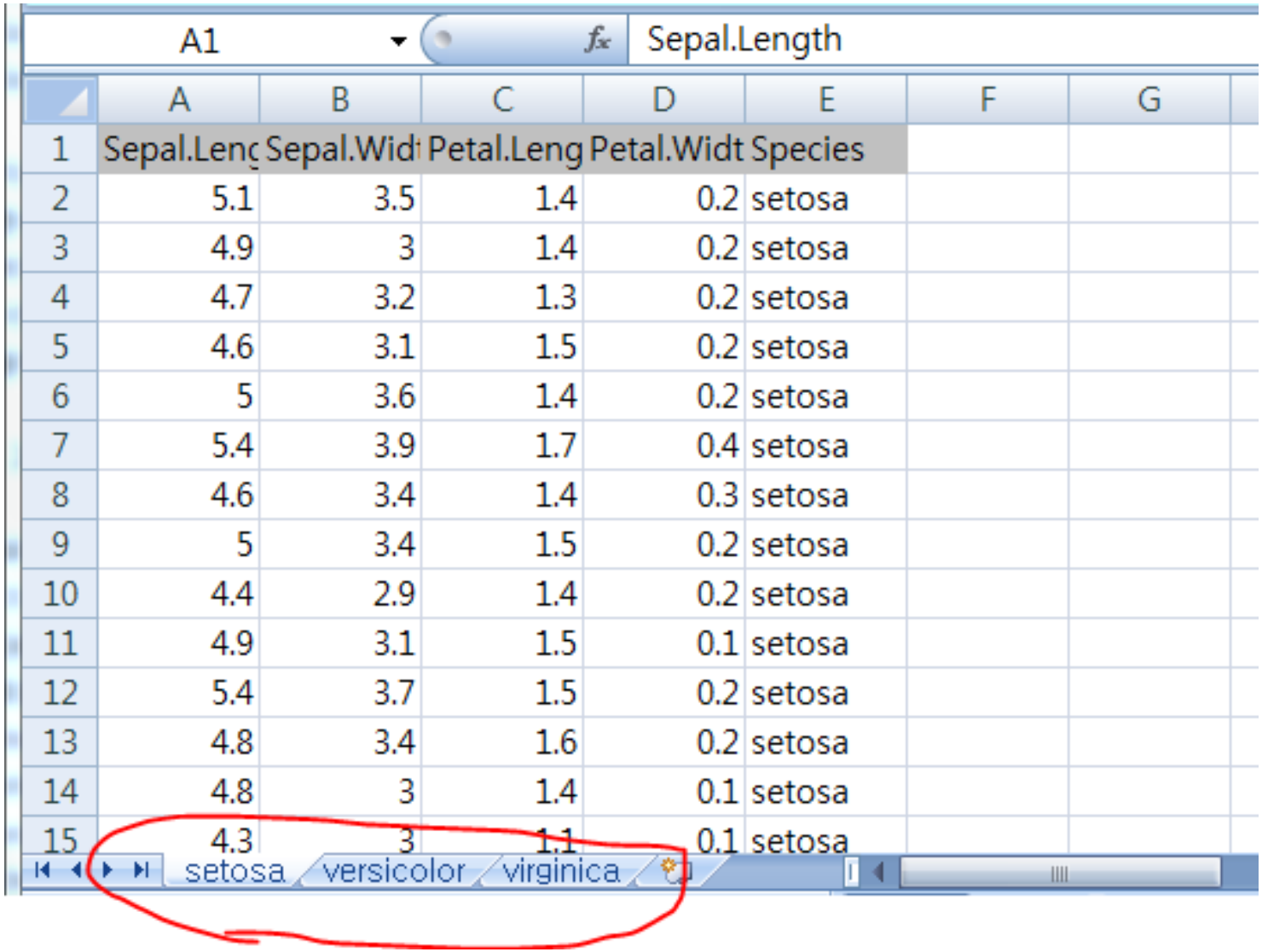
```
library(XLConnect)
File <- system.file("demoFiles/mtcars.xlsx", package = "XLConnect")
# Load workbook
wb <- loadWorkbook(File)
# Read table 'MtcarsTable' from sheet 'mtcars_table'
dt <- readTable(wb, sheet = "mtcars_table",
                table = "MtcarsTable")
head(dt)
```

XLConnect를 이용한 엑셀파일에 쓰기

iris data.frame을 XLConnect를 이용하여 품종별로 엑셀의 서로다른 워크시트에 저장하는 프로그램

```
# Load workbook (create if not existing)
wb <- loadWorkbook("iris.xlsx", create = TRUE)
Species <- as.character(unique(iris$Species))
for(sp in Species){
  # Create worksheet
  createSheet(wb, name = sp)
  # Write data to worksheet
  writeWorksheet(wb, iris[iris$Species==sp,],
                 sheet = sp, header=TRUE)
}
# Writes the file to disk
saveWorkbook(wb)
```

원본 파일



A1 | Sepal.Length

	A	B	C	D	E	F	G
1	Sepal.Lenç	Sepal.Wid	Petal.Leng	Petal.Widt	Species		
2	5.1	3.5	1.4	0.2	setosa		
3	4.9	3	1.4	0.2	setosa		
4	4.7	3.2	1.3	0.2	setosa		
5	4.6	3.1	1.5	0.2	setosa		
6	5	3.6	1.4	0.2	setosa		
7	5.4	3.9	1.7	0.4	setosa		
8	4.6	3.4	1.4	0.3	setosa		
9	5	3.4	1.5	0.2	setosa		
10	4.4	2.9	1.4	0.2	setosa		
11	4.9	3.1	1.5	0.1	setosa		
12	5.4	3.7	1.5	0.2	setosa		
13	4.8	3.4	1.6	0.2	setosa		
14	4.8	3	1.4	0.1	setosa		
15	4.3	3	1.1	0.1	setosa		

setosa | versicolor | virginica

그림 1: 원본 엑셀 파일

sqldf를 이용한 데이터프레임 핸들링

- sqldf : R의 데이터프레임 객체를 SQL을 이용하여 조작하도록 지원하는 패키지

```
library(sqldf)## Load the package
# Use the iris data set
sqldf('select count(*) `N`,
      AVG("Sepal.Width") `Sepal.Length`
      from iris group by Species')
```

sqldf

```
system.time({
a8r <- aggregate(iris[1:2], iris[5], mean)
})

system.time({
a8s <- sqldf('select Species,
      avg("Sepal.Length") `Sepal.Length`,
      avg("Sepal.Width") `Sepal.Width`
      from iris group by Species')
})
```