

# R

## data import and export

A. Blejec  
andrej.blejec@nib.si

October 27, 2013

# Read

## R can read ...

- ▶ plain text tables and files
- ▶ Excel files
- ▶ SPSS, SAS, Stata formatted data
- ▶ databases (like MySQL)
- ▶ XML, HTML files

# Write

## R can write ...

- ▶ plain text tables and files
- ▶ Excel files - use tab delimited plain text
- ▶ SPSS, SAS, Stata formatted data
- ▶ databases (like MySQL)
- ▶ XML, HTML files

## Reading texts

---

<code>read.table()</code>	Read data formatted as a table
<code>read.delim() ...</code>	<code>read.table()</code> variants
<code>readLines()</code>	Read lines from a file
<code>readline()</code>	Reads a line from the Console
<code>read.clipboard()</code>	Read text from clipboard

---

# Reading text data

Main function to read text data is `read.table()`.

Data table:

```
gender age weight height
F 18 68.0 1.720
M 17 68.2 1.753
M 17 68.4 1.730
F 18 65.6 1.743
M 17 65.2 1.765
```

File: `bmi.txt`

## Column names: header

```
> data <- read.table("../data/bmi.txt", header=TRUE)  
> head(data)
```

	gender	age	weight	height
1	F	18	68.0	1.720
2	M	17	68.2	1.753
3	M	17	68.4	1.730
4	F	18	65.6	1.743
5	M	17	65.2	1.765
6	F	18	64.6	1.690

## Other arguments

```
> if(interactive())  
+ ?read.table
```

```
read.table(file, header = FALSE, sep = "", quote = "\"'",  
           dec = ".", row.names, col.names,  
           as.is = !stringsAsFactors,  
           na.strings = "NA", colClasses = NA, nrows = -1,  
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,  
           strip.white = FALSE, blank.lines.skip = TRUE,  
           comment.char = "#",  
           allowEscapes = FALSE, flush = FALSE,  
           stringsAsFactors = default.stringsAsFactors(),  
           fileEncoding = "", encoding = "unknown")
```

## read.table() variants

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

```
read.csv2 (file, header = TRUE, sep = ";", quote="\\"",  
          dec=",", fill = TRUE, comment.char="", ...)
```

```
read.delim (file, header = TRUE, sep = "\t", quote="\\"",  
           dec=".", fill = TRUE, comment.char="", ...)
```

```
read.delim2(file, header = TRUE, sep = "\t", quote="\\"",  
            dec=",", fill = TRUE, comment.char="", ...)
```



## What next?

If you want to use variables by names, attach your data!

```
> dim(data)           # dimension: n rows and columns
[1] 45  4

> names(data)         # variable names
[1] "gender" "age"      "weight" "height"

> try(mean(weight))  # variables are not available yet
> attach(data)       # make variables usable
> mean(weight)       # variables are available
[1] 61.00889

> table(gender, age)

      age
gender 17 18
  F    9 16
  M    9 11
```

## Reading files line by line

```
> txt <- readLines("../data/bmiCom.txt")
> str(txt)

chr [1:49] "# Data for BMI calculation\t\t\t\t" ...

# Data for BMI calculation
# weight in kg
# height in m
gender age weight height
F 18 1.720 # weight not known
M 17 68.2 1.753
M 17 68.4 1.730
F 18 65.6 1.743
M 17 65.2 1.765
F 18 64.6 1.690
```

read.table() can skip comment lines

```
> data <- read.table("../data/bmiCom.txt", header=TRUE, sep=";")  
> head(data)
```

	gender	age	weight	height	X
1	F	18	NA	1.720	NA
2	M	17	68.2	1.753	NA
3	M	17	68.4	1.730	NA
4	F	18	65.6	1.743	NA
5	M	17	65.2	1.765	NA
6	F	18	64.6	1.690	NA

## Read lines from console/terminal

```
> name <- "unknown"  
> if(interactive()) {name <- readline("Who are you?")  
+ cat("Hello", name, "!")  
+ }
```

## Reading text from clipboard

First four lines from file File: `bmi.txt` were copied to clipboard:

```
> read.clipboard<-  
+ function (header = TRUE, sep = "\t", ...) {  
+   read.table (file = "clipboard", header = header, s  
+ }  
> if(interactive()) {  
+ data <- read.clipboard(sep="\t")  
+ data  
+ }
```

	gender	age	weight	height	X
1	F	18	68.0	1.720	NA
2	M	17	68.2	1.753	NA
3	M	17	68.4	1.730	NA

Empty cells are marked as NA

# Reading Excel files

- ▶ function `readClipboard()` and `read.table()` variant
- ▶ package **xlsReadWrite**
- ▶ package **RODBC** (R Open DataBase Connectivity)
- ▶ package **xlsx** provides reading and writing of `.xlsx` files

## read.clipboard

Many times the fastest way to get data on the fly

Grab first four lines from the file

<http://ablejec.nib.si/pub/I2R/dat/bmi.xls> bmi.xls

```
> if(interactive()) {  
+ data <- read.clipboard()  
+ data  
+ }
```

	gender	age	weight	height
1	F	18	68.0	1.720
2	M	17	68.2	1.753
3	M	17	68.4	1.730

Empty cells are marked as NA

# xlsReadWrite

- ▶ Package **xlsReadWrite** provides function `read.xls()` to read .xls files.
- ▶ Package **xlsx** provides function `read.xlsx()` to read .xlsx files.



# Package `xlsx`

## Reference by number

```
> library(xlsx)
> X <- read.xlsx("../data/bmi.xls", 1)
> head(X)
```

	gender	age	weight	height
1	F	18	68.0	1.720
2	M	17	68.2	1.753
3	M	17	68.4	1.730
4	F	18	65.6	1.743
5	M	17	65.2	1.765
6	F	18	64.6	1.690

```
> #DataName <- latexTranslate(paste(lfn, "/ Sheet", Sheet,
```

## More on package `xlsx`

### Reference by name

```
> lfn <- "../data/bmi.xls"  
> wb <- loadWorkbook(lfn)  
> sheets <- getSheets(wb)  
> names(sheets) [1]
```

```
[1] "bmi"
```

```
> X <- read.xlsx(lfn, sheetName="bmi")  
> head(X)
```

	gender	age	weight	height
1	F	18	68.0	1.720
2	M	17	68.2	1.753
3	M	17	68.4	1.730
4	F	18	65.6	1.743
5	M	17	65.2	1.765
6	F	18	64.6	1.690

```
> #DataName <- latexTranslate(paste(lfn, "/ Sheet", Sheet,
```

# Relacijske baze podatkov

## package RODBC

### ODBC Open Database Connectivity za R

- ▶ `odbcConnect(dsn, uid = "", pwd = "", ...)`
- ▶ `odbcConnectAccess(access.file, uid = "", pwd = "", ...)`
- ▶ `odbcConnectAccess2007(access.file, uid = "", pwd = "", ...)`
- ▶ `odbcConnectDbase(dbf.file, ...)`
- ▶ `odbcConnectExcel(xls.file, readOnly = TRUE, ...)`
- ▶ `odbcConnectExcel2007(xls.file, readOnly = TRUE, ...)`

## sql\* funkcije za dostop do baz podatkov

[1]	"sqlClear"	"sqlColumns"	"sqlCopy"
[4]	"sqlCopyTable"	"sqlDrop"	"sqlFetch"
[7]	"sqlFetchMore"	"sqlGetResults"	"sqlPrimaryKeys"
[10]	"sqlQuery"	"sqlSave"	"sqlTables"
[13]	"sqlTypeInfo"	"sqlUpdate"	

# Izvedeni paketi

Za MySQL in Oracle sta razvita paketa:

- ▶ **RMySQL**
- ▶ **ROracle**

# RODBC

Note: works only for 32-bit Windows

```
> library(RODBC)
> read.xls <- function(file, ..., sheet = "Sheet1", cch
+ comment = FALSE, rownames = TRUE, colnames = TRUE) {
+ z <- odbcConnectExcel(file)
+ myframe <- sqlFetch(z, sheet, rownames = rownames,
+ colnames = !colnames, ...)
+ close(z)
+ if(rownames){
+ rownames(myframe) <- myframe[, 1]
+ myframe <- myframe[, -1]
+ }
+ commentLines <- grep(paste("^", cch, sep = ""), rowna
+ if (!is.null(commentLines) & length(commentLines) >
+ 0 & !comment)
+ myframe <- myframe[-commentLines, ]
+ if (comment)
+ myframe <- rownames(myframe[commentLines, ])
+ invisible(myframe)
```

## read.xls

```
> X <- read.xls("../data/bmi.xls", sheet="bmi", rownames=F)
> #head(X)
```

## Reading SPSS files

You can use `read.spss()` from package **foreign**

```
> library(foreign) # you have to install it first!  
> X<-read.spss("../data/bmi.sav")
```

Result X is a list with VARLABELS and VALUELABELS as attributes:

```
> attr(X, "variable.labels")  
  
      gender                age  
"Gender" "Age at measurement"  
      weight                height  
"Weight (kg)"            "Height (m)"
```

You can convert the list to a data frame:

```
> Y <- as.data.frame(X)
```



## Reading SPSS files

You can also use `spss.get()` from package **Hmisc**

```
> library(Hmisc) # you have to install it first!  
> X<-spss.get("../data/bmi.sav")
```

Which produces a labeled data frame

```
> head(X)
```

	gender	age	weight	height
1	Male	17	64.2	1.770
2	Male	17	74.8	1.705
3	Male	17	55.8	1.770
4	Male	17	68.4	1.730
5	Male	17	68.2	1.753
6	Male	17	88.0	1.910

# Reading SPSS files

Labels are preserved

```
> label(X$weight)
```

```
      weight  
"Weight (kg)"
```

## Choosing files interactively

```
> if(interactive())  
+ file.choose()
```

For more than one file use `choose.files()`

## Write tab delimited tables

```
> data <- head(X)
> write.table(data, "../data/results.txt", sep="\t")
> write.table(data, "", sep="\t")
```

```
"gender" "age" "weight" "height"
"1" "Male" 17 64.2 1.77
"2" "Male" 17 74.8 1.705
"3" "Male" 17 55.8 1.77
"4" "Male" 17 68.4 1.73
"5" "Male" 17 68.2 1.753
"6" "Male" 17 88 1.91
```

## Write tab delimited tables

```
> write.table(data, "../data/results2.txt", sep="\t", col.names = NA)
> write.table(data, "", sep="\t", col.names = NA)
```

```
" " "gender" "age" "weight" "height"
"1" "Male" 17 64.2 1.77
"2" "Male" 17 74.8 1.705
"3" "Male" 17 55.8 1.77
"4" "Male" 17 68.4 1.73
"5" "Male" 17 68.2 1.753
"6" "Male" 17 88 1.91
```

# Writing Excel tables

## Package `xlsx`

```
> lfn <- "./results2.xls"  
> if(file.exists(lfn))  
+   file.remove(lfn)
```

```
[1] TRUE
```

```
> dir(pattern="*.xls")
```

```
character(0)
```

```
> write.xlsx(data, "results2.xls")  
> dir(pattern="*.xls")
```

```
[1] "results2.xls"
```

## More ...

- ▶ R Help/Manuals (in PDF) / R Data Import/Export
- ▶ <http://ablejec.nib.si/R/xlsReadWrite.pdf>
- ▶ <http://ablejec.nib.si/R/I2R/DOC/I2R.pdf>