Reading and Writing Data

Michael Malick

Updated: 15 May 2013

R Data Manual

http://cran.r-project.org/doc/manuals/R-data.pdf

Reading and Writing Data

The three most common data file formats for reading into R and writing from R to a text file are:

- Comma separated value files (CSV)
- Tab separated files
- Space separated files

Example: CSV File

CSV files are plain text files that MS Excel can open and edit

```
"Time","Count","Species"
1,10,120
1,13,111
2,17,130
2,12,122
3,11,109
3,10,170
4,15,132
4,14,129
```

Working Directory Assumption

- **Reading Data**: by default, R assumes the data to read in are located in the working directory

- **Writing Data**: by default, R will write all data files to the working directory

- I always keep my data in the working directory or in a subdirectory of the working directory

Writing Data

write.table()

The `write.table()` function is the primary means to write data from R to a text file

```
write.table(iris, file = "iris.txt",
      row.names = FALSE, sep = " ")
```

write.csv()

The `write.csv()` function is a special case of the `write.table()` function

```
?write.csv # same as write.table

write.table(iris, file = "iris.csv",
    row.names = FALSE, sep = ",")

write.csv(iris, file = "iris.csv",
    row.names = FALSE)
```

Reading Data

Reading Data Into R

R can read data stored in numerous formats

- Text files
- Webpages
- Databases (e.g., SQL; RODBC package)
- Binary files (e.g., netCDF; ncdf4 package)

R and Excel

Out of the box, R cannot read .xls or .xlsx files

- To read data stored in an Excel spreadsheet you need to save the file as a .csv file:

1. In Excel select "save as ..." under the file menu
2. Select the format "Comma Separated Values (.csv)"

read.table()

The `read.table()` function is the primary function used to read data into R

```
dat1 <- read.table(file = "iris.txt",
    sep = " ", header = TRUE)
```

```
dat2 <- read.table(file = "iris.csv",
    sep = ",", header = TRUE)
```

read.csv()

The `read.csv()` function is a special case of the `read.table()` function

```
?read.csv # same as read.table

dat2 <- read.table(file = "iris.csv",
    sep = ",", header = TRUE)

dat3 <- read.csv(file = "iris.csv")
```

Non-Standard Encodings

Occasionally on a Mac a .csv file will have a non-standard encoding, the correct encoding is latin1 for a Mac

```
dat6 <- read.csv(file = "iris.csv",
    fileEncoding = "latin1")
```

Reading From Webpages

http://www.o3d.org/npgo/data/NPGO.txt

```
npgo <- read.table(
    "http://www.o3d.org/npgo/data/NPGO.txt",
    header = FALSE)
```

Many More Options

http://cran.r-project.org/doc/manuals/R-data.pdf

```
?read.table

?write.table
```

You Try...

1. Write the `mtcars` dataset to a .csv file in your working directory
   - Hint: use `write.csv()`
2. Read in the .csv file you just wrote to the working directory
   - Hint: use `read.csv()`

Append: `write.table()`

You can append data to an existing data file

```
dat <- data.frame(X = 1, Y = 2, Z = 3)

write.table(dat, file = "dat.txt",
    row.names = FALSE, sep = " ")


new.dat <- 4:6
write.table(t(new.dat), file = "dat.txt",
    row.names = FALSE, append = TRUE,
    col.names = FALSE)
```

Writing to the Clipboard

You can write data to the system clipboard

```
rand <- rnorm(10)

# Mac
write.table(rand, file = pipe("pbcopy"),
    row.names = FALSE)

# Windows
write.table(rand, file = "clipboard",
    row.names = FALSE)
```

sink()

The `sink()` function is used to divert output that is usually printed to the R console to a text file

```
sink(file = "sink.txt")
    x <- rnorm(10)
    y <- rnorm(10)
    summary(lm(x~y))
sink()
```

- NOTE: you need to close the connection to sink by calling `sink()`

Set Class: `read.table()`

You can set the class of each variable when you read in the data

```
class(dat1$Sepal.Length)
```

```
dat4 <- read.table(file = "iris.txt",
    sep = " ", header = TRUE, colClasses =
    c(Sepal.Length = "character"))

class(dat4$Sepal.Length)
```

Factors: `read.table()`

By default R will convert all strings to factors; you can override this by using the `stringsAsFactors` argument

```
class(dat4$Species)

dat5 <- read.table(file = "iris.txt",
    sep = " ", header = TRUE,
    stringsAsFactors = FALSE)

class(dat5$Species)
```

Reading From the Clipboard

You can read data from the system clipboard

```
# Mac
cb <- read.table(pipe("pbpaste"))
```

```
# Windows
cb <- read.table("clipboard")
```