

XLConnect : read, write and manipulate Microsoft Excel files from within R

XLConnect is a package that allows for reading, writing and manipulating Microsoft Excel files from within R.

It does not require any installation of Microsoft Excel or any other special drivers to be able to read & write Excel files. The only requirement is a recent version of a Java Runtime Environment (JRE).

In order to get started have a look at the [package vignette](#), the numerous demos available via `demo(package = "XLConnect")` or browse through the comprehensive [reference manual](#).

Installation

Software Requirements

- R, version 2.10.0 or higher
- Java Runtime Environment (JRE), version 6.0 or higher

if you don't have JRE on your computer, you have to install it :
on linux /ubuntu use this command : `apt-get install openjdk-6-jdk`

Package Installation

XLConnect can be easily installed by using the `install.packages()` command in your R session:

*Copy to Clipboard*Code R :

```
install.packages("XLConnect")
```

Writing Excel files

Basic functions

*Copy to Clipboard*Code R :

```

#loading the package
require(XLConnect)

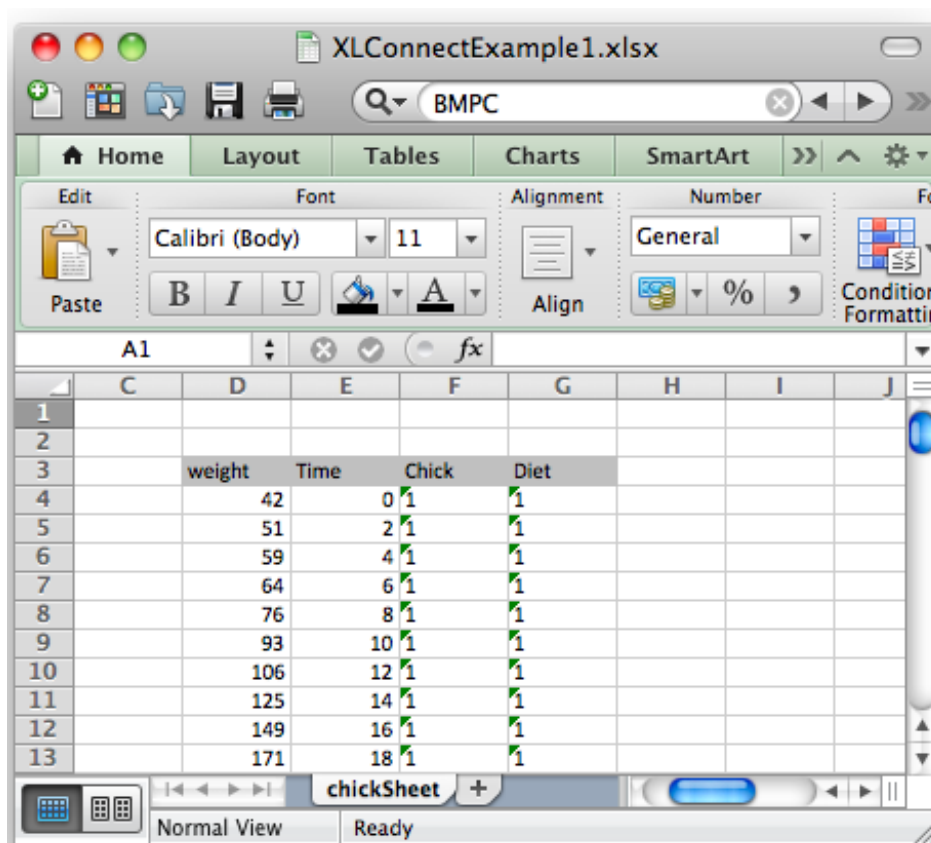
#creating an Excel workbook. Both .xls and .xlsx file formats can be used.
wb <- loadWorkbook("XLConnectExample1.xlsx", create = TRUE)

#creating sheets within an Excel workbook
createSheet(wb, name = "chickSheet")

#writing into sheets within an Excel workbook :
#writing ChickWeight data frame into chickSheet
writeWorksheet(wb, ChickWeight, sheet = "chickSheet", startRow = 3, startCol =
4)

#saving a workbook to an Excel file :
#saves a workbook to the corresponding Excel file and writes the file to disk.
saveWorkbook(wb)

```



Writing sheet with one call

The four lines of code presented in the previous example can be replaced with a single call of the **writeWorksheetToFile()** function:

*Copy to Clipboard*Code R :

```
require(XLConnect)
writeWorksheetToFile("XLConnectExample2.xlsx", data = ChickWeight,
                    sheet = "chickSheet", startRow = 3, startCol =
4)
```

writeWorksheetToFile() loads the workbook, creates the sheet and finally saves the workbook. When you only need to write one sheet into an Excel file, this is probably the better choice. If you need to write more sheets, however, using the functions presented in the previous example will be more efficient. This is because calling `writeWorksheetToFile()` multiple times will open, write and close the Excel file with each call. Using the functions in the first example will, in contrast, allow you to open the workbook, do multiple operations on it and only then close it.

Reading from an Excel sheet

For this purpose, we will use the file created in the above example. We set the `endRow` argument to 10, to limit the result. We set the rest of arguments specifying boundaries as 0, so that they are automatically determined. Please note, that alternatively to setting the `sheet` argument to the name of the sheet we want to read from, "chickSheet", we could also specify it with the sheet index, as `sheet = 1`.

Basic functions

Copy to Clipboard Code R :

```
#loading the package
require(XLConnect)

#Loading an Excel workbook. Both .xls and .xlsx file formats can be
used.
wb = loadWorkbook("XLConnectExample1.xlsx", create = TRUE)

#reading worksheets of an Excel workbook
data = readWorksheet(wb, sheet = "chickSheet", startRow = 0, endRow =
10,
startCol = 0, endCol = 0)

#print data
data
```

```
weight Time Chick
Diet
1      42     0     1
1
2      51     2     1
1
3      59     4     1
1
4      64     6     1
1
5      76     8     1
1
6      93    10     1
1
7     106    12     1
1
```

Reading from an Excel sheet with one call

*Copy to Clipboard*Code R :

```
require(XLConnect)
data = readWorksheetFromFile("XLConnectExample1.xlsx", sheet = "chickSheet",
startRow = 0, endRow = 10, startCol = 0, endCol = 0)
```

Adds an image to a worksheet using a named region

*Copy to Clipboard*Code R :

```
# Load workbook (create if not existing)
wb = loadWorkbook("add_image_to_excel.xls", create=TRUE)

# Create a sheet named 'boxplot'
createSheet(wb, name = "boxplot")

# Create a named region called 'boxplot' referring to the sheet
# called 'boxplot'
createName(wb, name = "boxplot", formula = "boxplot!$B$2")

# Create R plot to a png device
png(filename = "boxplot.png", width = 800, height = 600)
boxplot(count ~ spray, data = InsectSprays, col = "lightgray")
dev.off()

# Write image to the named region created above
addImage(wb, filename = "boxplot.png", name="boxplot", originalSize =
TRUE)

# Save workbook (this actually writes the file to disk)
saveWorkbook(wb)
```

As a result, we obtain the following graph, written into Excel file

