Few steps to connect R with Excel: XLConnect!

By Anna Longari

Excel is the basic tool for any type of analysis.

Often using R we have the need to communicate with Excel, for example

- read an excel file with multiple sheets
- write an excel file with multiple sheets
- read/write formulas
- export the results of an analysis in an excel report

XLConnect permits to create a formatted spreadsheet usable as a dynamic report of the R analysis.

The package installation is very easy:

```
1
2    install.packages("XLConnect")
```

After the installation the package is to be loaded:

```
1
2    require("XLConnect")
```

XLConnect offers many features to realize a good connection between R and Excel.

An Excel data grid becomes an R data.frame!

An Excel named region is reported in R and used to place R objects in the right position!

Reading and writing named ranges enables to process complex inputs and outputs in an efficient way.

We can introduce XLConnect by few sequential examples.


Create a new empty xlsx file trough R

We want to create a new empty .xlsx file with one empty sheet named 'Input'; the syntax is

```
1
2    fileXls <- paste(outDir,"newFile.xlsx",sep='/')
3    unlink(fileXls, recursive = FALSE, force = FALSE)
4    exc <- loadWorkbook(fileXls, create = TRUE)
5    createSheet(exc,'Input')
6    saveWorkbook(exc)
```

Command *loadWorkbook* creates an R workbook object.

Command *saveWorkbook* save the R object in a xlsx file.

The result is represented in next figure:

Create an Excel empty sheet trough R using XLConnect
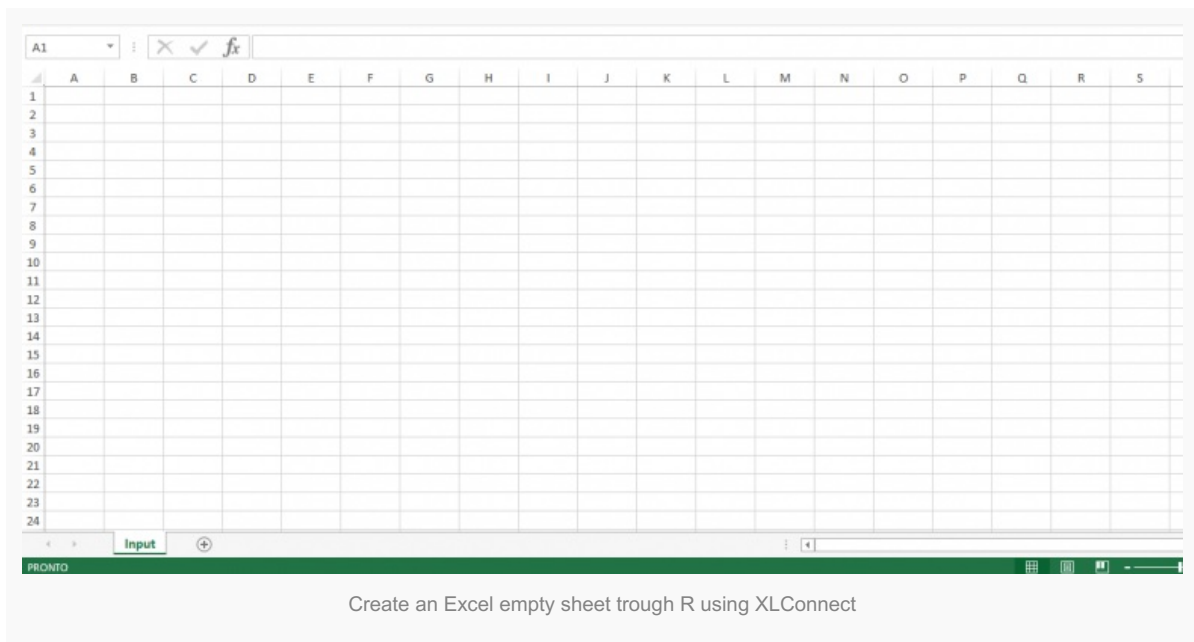
Populate an empty xlsx sheet with R

We want to add something to the empty sheet *input*

```
1
2   input <- data.frame('inputType'=c('Day','Month'),'inputValue'=c(2,5))
3   writeWorksheet(exc, input, sheet = "input", startRow = 1, startCol = 2)
4   saveWorkbook(exc)
```

The *input* data.frame, with 2 rows and 2 column, is created and the command *writeWorksheet* writes the content of this data.frame in the *input* sheet starting from the cell (1,2).

Write inside an Excel sheet with R using XLConnect

Add other sheets to the workbook

At this point a second sheet could be created

```
1
2   require(reshape)
3   createSheet(exc,'Airquality')
4   airquality$isCurrent<-NA
5   createName(exc,
6   name='Airquality',formula='Airquality!$A$1')
7   writeNamedRegion(exc, airquality, name = 'Airquality',
    header = TRUE)
    saveWorkbook(exc)
```

Command *createName* create a named region 'Airquality' starting from the cell $A$1 of sheet Airquality. Command *writeNamedRegion* writes airquality data.frame with headers in the named region 'Airquality'.

Add another variable to an Excel sheet with R using XLConnect

Add an Excel formula to an xlsx sheet trough R

The empty column *isCurrent* could be populated with a formula that links the Input sheet with the Airquality sheet.

Dioxide	Ozone	Solar.R	Wind	Temp	Month	Day	isCurrent
0,75	41	190	7,4	67	5	1	0
0,7	36	118	8	72	5	2	1
0	12	149	12,6	74	5	3	0
0,65	18	313	11,5	62	5	4	0
0			14,3	56	5	5	0
0,95	28		14,9	66	5	6	0
0,6	23	299	8,6	65	5	7	0
0,35	19	99	13,8	59	5	8	0
0,15	8	19	20,1	61	5	9	0
1		194	8,6	69	5	10	0
0,3	7		6,9	74	5	11	0
0,9	16	256	9,7	69	5	12	0
0,6	11	290	9,2	66	5	13	0
0,85	14	274	10,9	68	5	14	0
1	18	65	13,2	58	5	15	0
0,1	14	334	11,5	64	5	16	0
0	34	307	12	66	5	17	0
0	6	78	18,4	57	5	18	0
0,5	30	322	11,5	68	5	19	0
0,15	11	44	9,7	62	5	20	0
0,05	1	8	9,7	59	5	21	0
0,55	11	320	16,6	73	5	22	0
0	4	25	9,7	61	5	23	0
0,5	32	92	12	61	5	24	0
0,1		66	16,6	57	5	25	0
0,5		266	14,9	58	5	26	0
0,6			8	57	5	27	0
0,65	23	13	12	67	5	28	0
0,45	45	252	14,9	81	5	29	0
0,3	115	223	5,7	79	5	30	0
0,85	37	279	7,4	76	5	31	0

```
1
2    colIndex <- which(names(airquality) == 'isCurrent')
3    letterDay <- idx2col(which(names(airquality) == 'Day'))
4    letterMonth <- idx2col(which(names(airquality) == 'Month'))
5    formulaXls <- paste('IF(AND(',
6               letterMonth,
7               2:(nrow(airquality)+1),
8               '=Input!C3,',
9               letterDay,
10              2:(nrow(airquality)+1),
11              '=Input!C2)',
12              ',1,0)',sep='')
13   setCellFormula(exc, sheet='Airquality',2:(nrow(airquality)+1),colIndex,formulaXls)
14   saveWorkbook(exc)
```

The function idx2col returns the correspondig excel letter for the index column. With the syntax

```
1
2    letterDay <- idx2col(which(names(airquality) == 'Day'))
```

the variable letterDay contains the excel letter for the column 'Day'

```
1
2    letterDay= F
```

Run an Excel formula with R trough XLConnect

Read and modify with R an existing xlsx file

Suppose that we have to modify the xlsx file just created.

```
1
2  exc2 <- loadWorkbook(fileXls, create = FALSE)
3  dtAir <- readWorksheet(exc2,'Airquality')
4  createSheet(exc2, name = "OzonePlot")
5  createName(exc2, name='OzonePlot',formula='OzonePlot!$A$1')
6  saveWorkbook(exc2)
```

In this case

- exc2 is a new XLC object created from the existing excel file (the old file Xls)

- the new sheet *OzonePlot* is added to exc2 object

- the new named region *OzonePlot* is creating starting from *OzonePlot!$A$1* cell

Adding an R plot (as image) to Excel

After creating a new sheet it is possible to put in this sheet a picture of a graph created in R

```
1
2    library(ggplot2)
3    fileGraph <- paste(outDir,'graph.png',sep='/')
4    png(filename = fileGraph, width = 800, height = 600)
5    ozone.plot <- ggplot(dtAir, aes(x=Day, y=Ozone)) +
6    geom_point() +
7    geom_smooth()+
8    facet_wrap(~Month, nrow=1)
9    print(ozone.plot)
10   invisible(dev.off())
11   addImage(exc2,fileGraph, 'OzonePlot',TRUE)
12   saveWorkbook(exc2)
```

The command *addImage* insert the png image of the graph just saved in the named region *OzonePlot*.

The result is represented in next figure:

Add an R plot to Excel using XLConnect
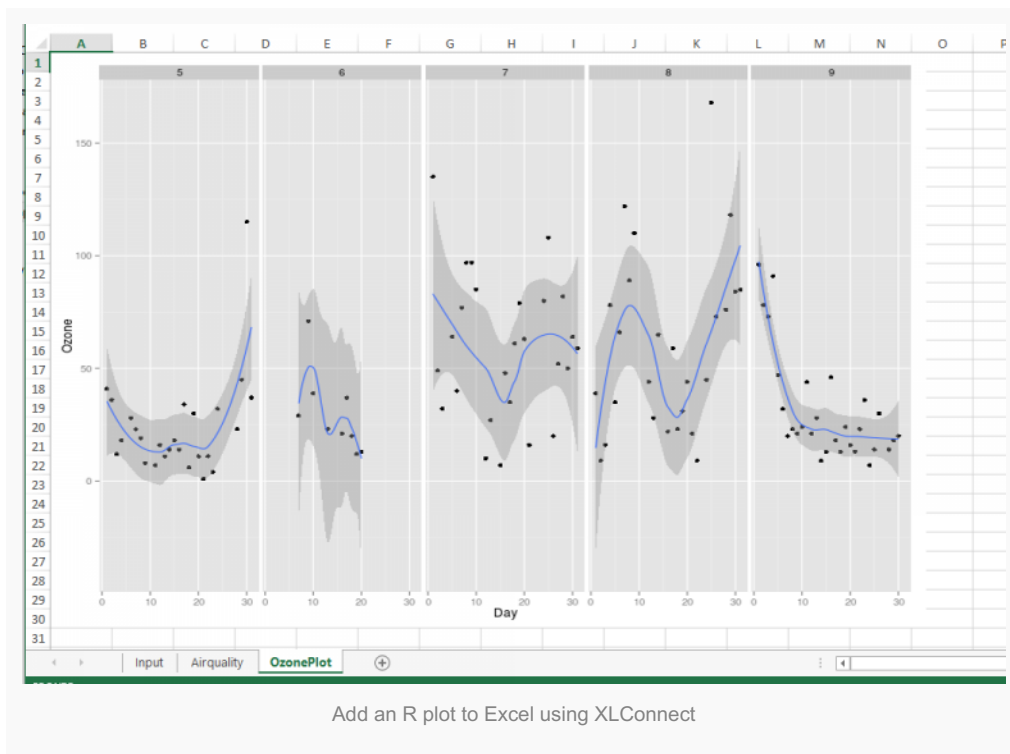
Extra functions of XLConnect

In this brief article we don't focus the attention on the excel layout but on the sheets content. For example we can automatically create an excel calculator with an input sheet and an output sheet managing all information in R.

XLConnect has also some features as defining cellstyles (data formats, borders, back fill colors..) , hide/unhide sheets, specyfing column width and row height, merged cells, auto-filters, style actions (using templates) and behavior with error cells.