# XLSTAT R

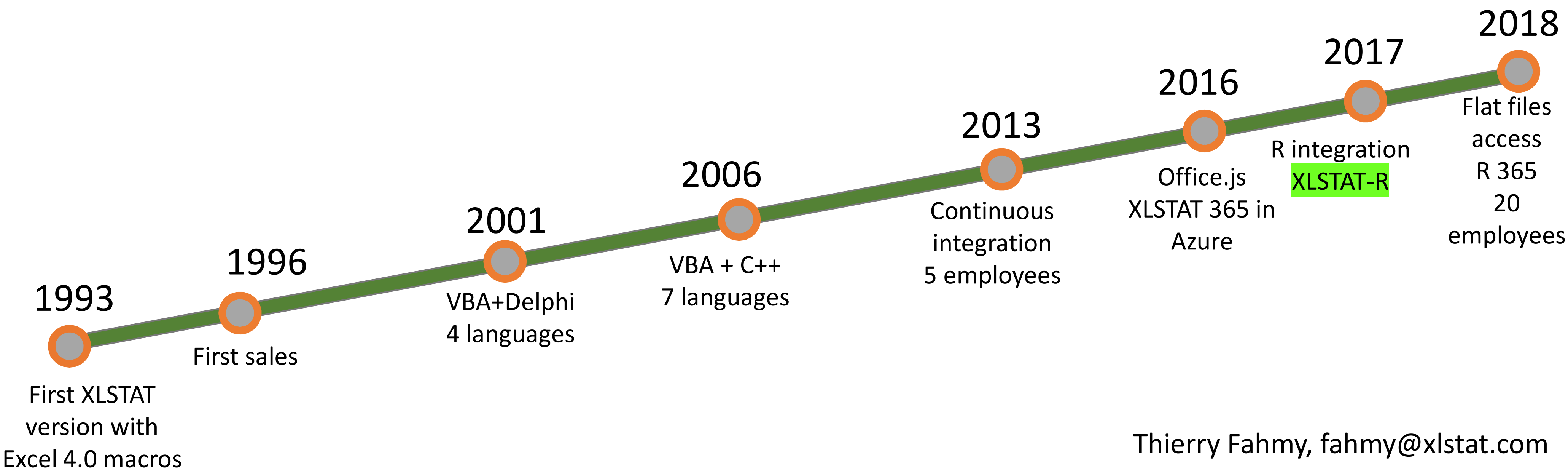
Making R « any-user-friendly »

Thierry Fahmy, fahmy@xlstat.com

Leaders in analytical solutions for MS Excel

# Where we’re coming from

XLSTAT
by Addinsoft



1993
First XLSTAT version with Excel 4.0 macros

1996
First sales

2001
VBA+Delphi
4 languages

2006
VBA + C++
7 languages

2013
Continuous integration
5 employees

2016
Office.js
XLSTAT 365 in Azure

2017
R integration
XLSTAT-R

2018
Flat files access
R 365
20 employees

Thierry Fahmy, fahmy@xlstat.com

The leaders in analytical solutions for MS Excel

# Dealing with R weaknesses and strengths

XLSTAT by Addinsoft

- Some packages are buggy or with slow code

  -> XLSTAT functions solve the issues identified in the R functions through its own functions developed in C++

- Some packages are limited in scope

  -> XLSTAT allows a unique and easy interface to access functions

- Some packages require strong expertise

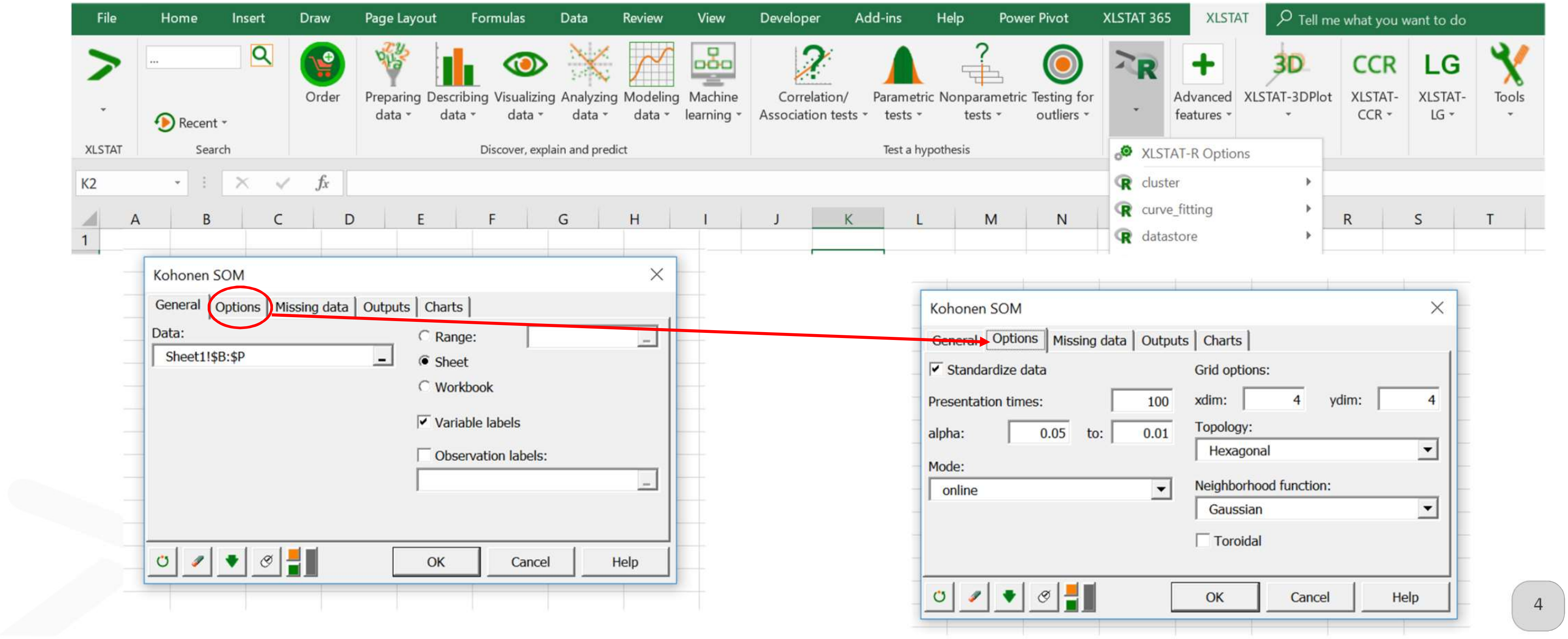
  -> XLSTAT-R interfaces simplify the options to what's mostly used

- Some packages are the state of the art

  -> XLSTAT-R gives access to these methods with a point and click approach

# 1. First approach: XLSTAT-R

- Make R functions available within the XLSTAT interface



Kohonen SOM – General

Data: Sheet1!$B:$P

○ Range:

● Sheet

○ Workbook

☑ Variable labels

□ Observation labels:

Kohonen SOM – Options

☑ Standardize data

Presentation times: 100

alpha: 0.05 to: 0.01

Mode: online

Grid options:

xdim: 4 ydim: 4

Topology: Hexagonal

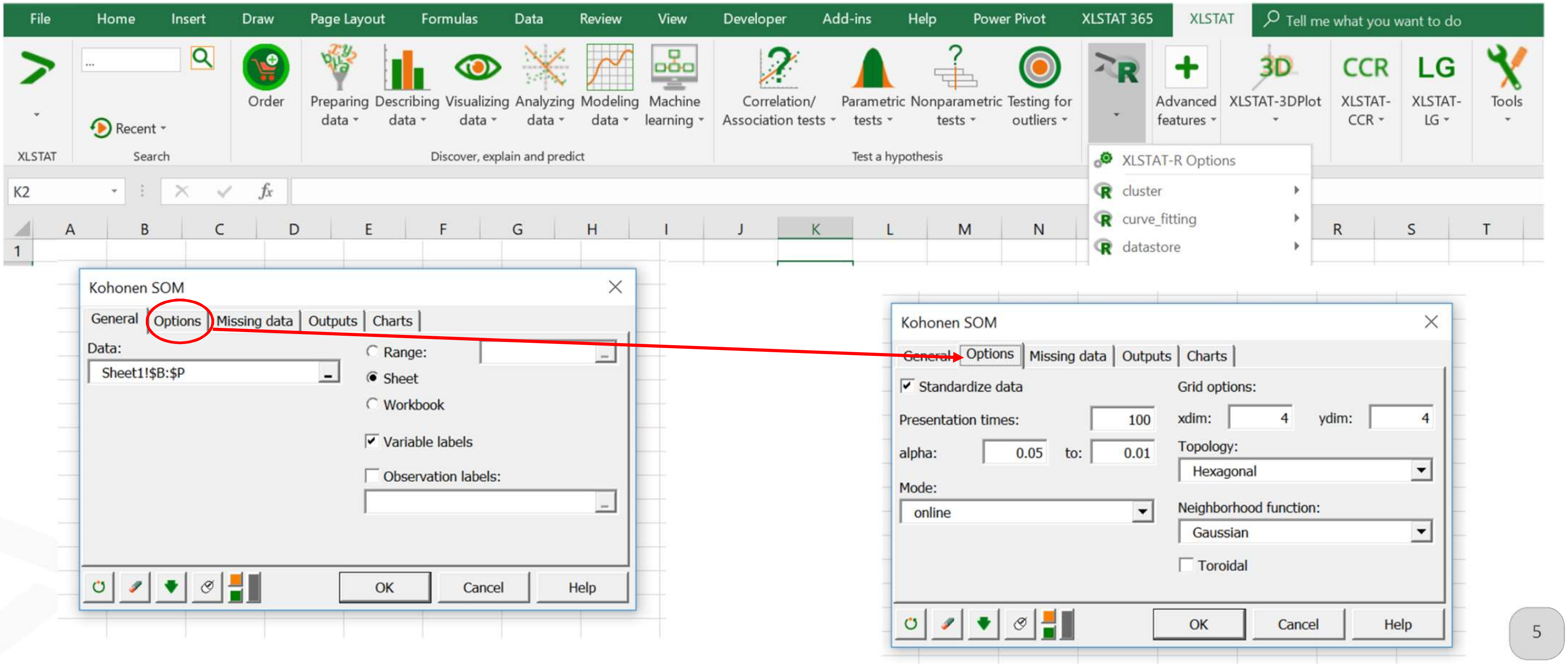
Neighborhood function: Gaussian

□ Toroidal

# 1. First approach: XLSTAT-R

XLSTAT by Addinsoft

- Make R functions available within the XLSTAT interface



Kohonen SOM

General | Options | Missing data | Outputs | Charts

Data: Sheet1!$B:$P

○ Range:
● Sheet
○ Workbook

☑ Variable labels

□ Observation labels:

Kohonen SOM

General | Options | Missing data | Outputs | Charts

☑ Standardize data

Presentation times: 100

alpha: 0.05 to: 0.01

Mode: online

Grid options:

xdim: 4 ydim: 4

Topology: Hexagonal

Neighborhood function: Gaussian

□ Toroidal

# 1.1 First approach: XLSTAT-R

- How does it work?

1. Build your XML code to
    1. Describe the function
    2. Generate the interface
    3. Inject the R code

2. Update the function list in XLSTAT, and run it from the XLSTAT-R menu

3. Get your results in Excel

# 1.2 XML based

XLSTAT
by Addinsoft

- One XML per R script (or at least per function)
  - XML has been found to be well adapted as it is visual, trans-platform, it is compatible with many editors
  - (But we are preparing a translation to json for our cloud version)
  - Copy and paste makes the creation pretty fast
  - Header:

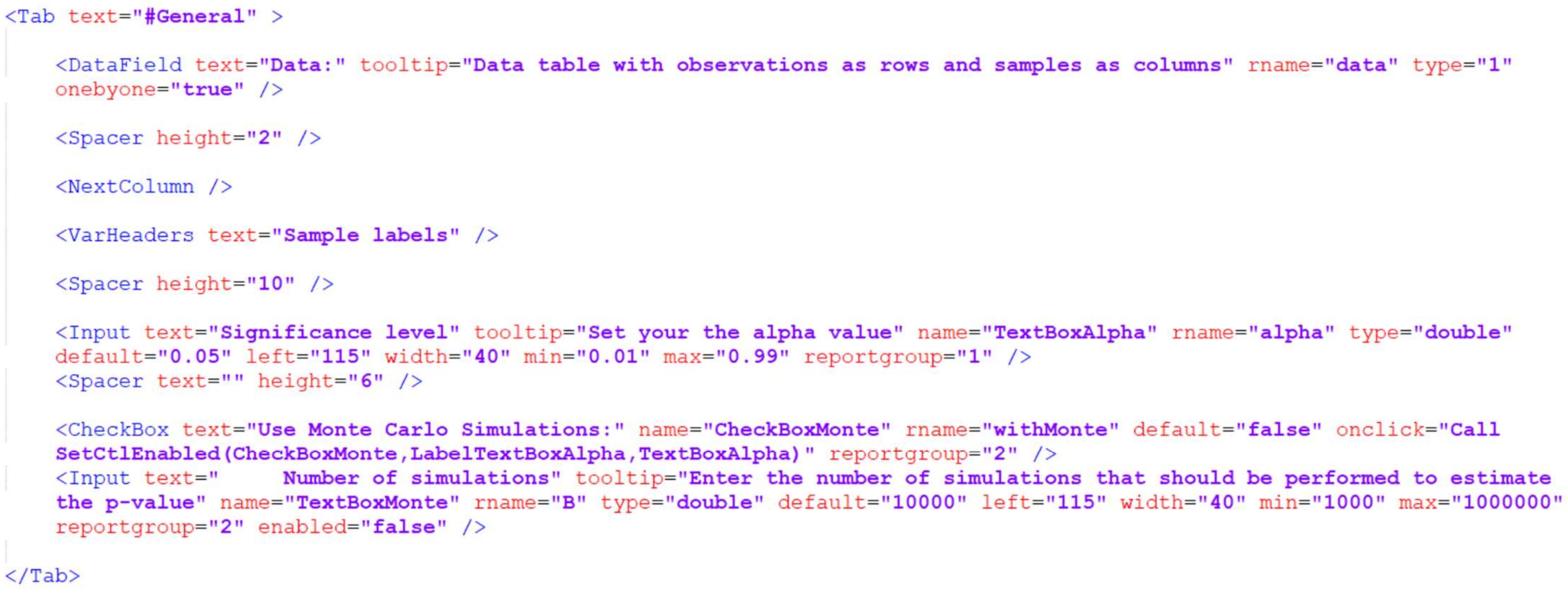
```
<Method text="Kohonen SOM" datastructure="data" function="som" group="kohonen" packages="kohonen"
family="Classification methods,Neural networks,Self organizing maps" question="How can I classify data into
homogeneous groups?" synonyms="Kohonen self organizing maps" keywords="kohonen,self,organizing,map,clustering" >
```

  - Identify the authors

```
<AuthorRFunction>Ron Wehrens and Johannes Kruisselbrin</AuthorRFunction>
```

# 1.3 Interface description

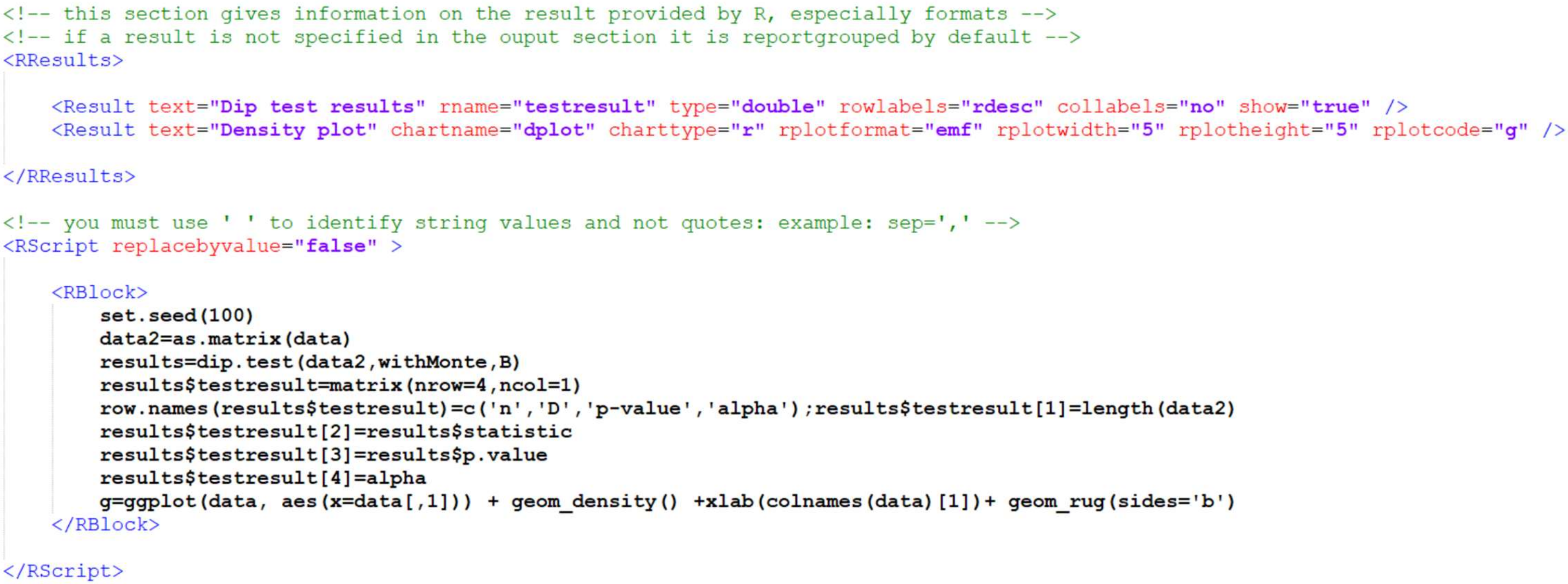
- Script the VBA interface in XML language



```
<Tab text="#General" >

    <DataField text="Data:" tooltip="Data table with observations as rows and samples as columns" rname="data" type="1"
    onebyone="true" />

    <Spacer height="2" />

    <NextColumn />

    <VarHeaders text="Sample labels" />

    <Spacer height="10" />

    <Input text="Significance level" tooltip="Set your the alpha value" name="TextBoxAlpha" rname="alpha" type="double"
    default="0.05" left="115" width="40" min="0.01" max="0.99" reportgroup="1" />
    <Spacer text="" height="6" />

    <CheckBox text="Use Monte Carlo Simulations:" name="CheckBoxMonte" rname="withMonte" default="false" onclick="Call
    SetCtlEnabled(CheckBoxMonte,LabelTextBoxAlpha,TextBoxAlpha)" reportgroup="2" />
    <Input text="     Number of simulations" tooltip="Enter the number of simulations that should be performed to estimate
    the p-value" name="TextBoxMonte" rname="B" type="double" default="10000" left="115" width="40" min="1000" max="1000000"
    reportgroup="2" enabled="false" />

</Tab>
```

# 1.4 Interface description

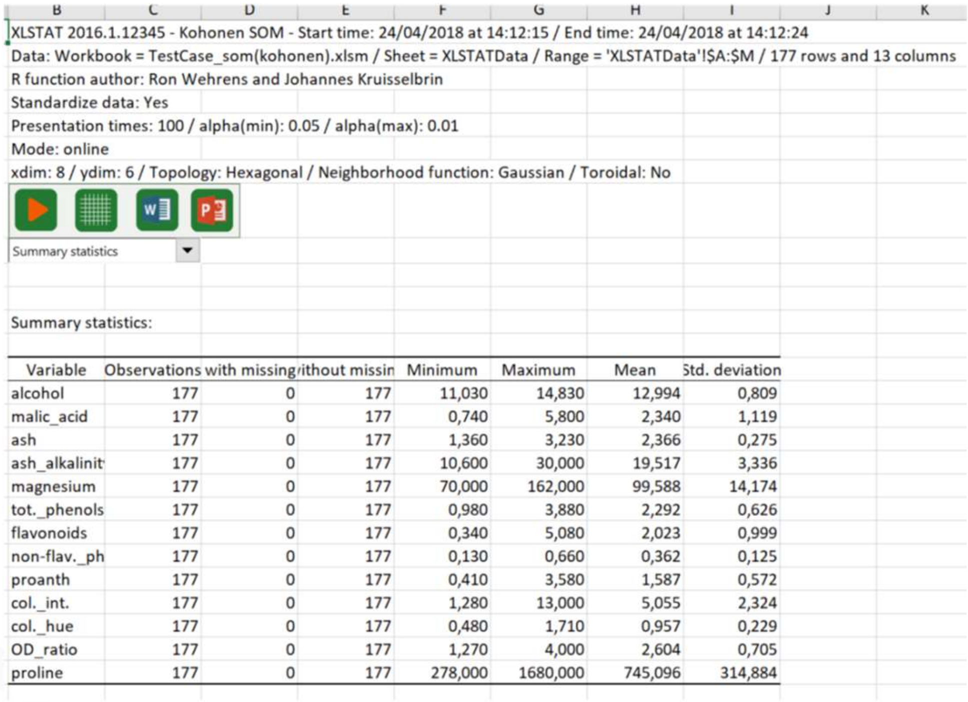
- Enter the R code and specify which results should be diplayed



```
<!-- this section gives information on the result provided by R, especially formats -->
<!-- if a result is not specified in the ouput section it is reportgrouped by default -->
<RResults>

    <Result text="Dip test results" rname="testresult" type="double" rowlabels="rdesc" collabels="no" show="true" />
    <Result text="Density plot" chartname="dplot" charttype="r" rplotformat="emf" rplotwidth="5" rplotheight="5" rplotcode="g" />

</RResults>

<!-- you must use ' ' to identify string values and not quotes: example: sep=',' -->
<RScript replacebyvalue="false" >

    <RBlock>
        set.seed(100)
        data2=as.matrix(data)
        results=dip.test(data2,withMonte,B)
        results$testresult=matrix(nrow=4,ncol=1)
        row.names(results$testresult)=c('n','D','p-value','alpha');results$testresult[1]=length(data2)
        results$testresult[2]=results$statistic
        results$testresult[3]=results$p.value
        results$testresult[4]=alpha
        g=ggplot(data, aes(x=data[,1])) + geom_density() +xlab(colnames(data)[1])+ geom_rug(sides='b')
    </RBlock>

</RScript>
```

# 1.5 Results in Excel

- The report format is the same as for XLSTAT



XLSTAT 2016.1.12345 - Kohonen SOM - Start time: 24/04/2018 at 14:12:15 / End time: 24/04/2018 at 14:12:24

Data: Workbook = TestCase_som(kohonen).xlsm / Sheet = XLSTATData / Range = 'XLSTATData'!$A:$M / 177 rows and 13 columns

R function author: Ron Wehrens and Johannes Kruisselbrin

Standardize data: Yes

Presentation times: 100 / alpha(min): 0.05 / alpha(max): 0.01

Mode: online

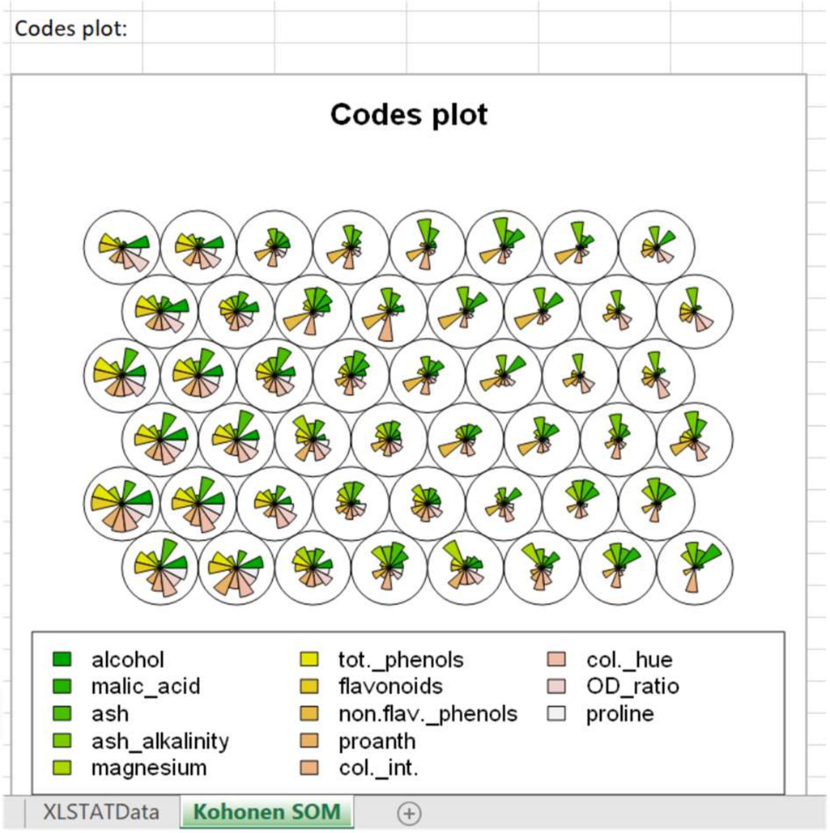
xdim: 8 / ydim: 6 / Topology: Hexagonal / Neighborhood function: Gaussian / Toroidal: No

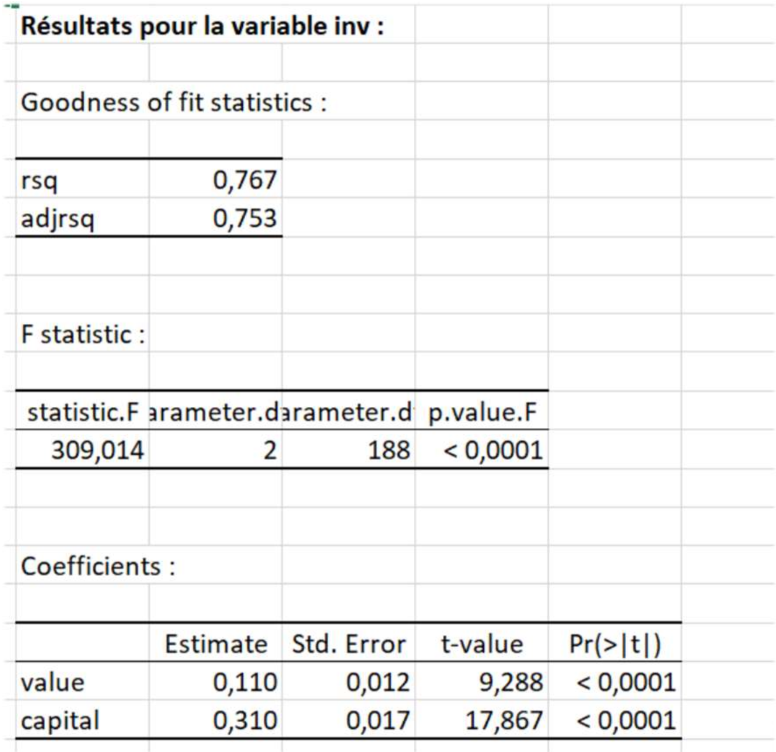
Summary statistics:

| Variable | Observations | with missing | ithout missin | Minimum | Maximum | Mean | Std. deviation |
|---|---|---|---|---|---|---|---|
| alcohol | 177 | 0 | 177 | 11,030 | 14,830 | 12,994 | 0,809 |
| malic_acid | 177 | 0 | 177 | 0,740 | 5,800 | 2,340 | 1,119 |
| ash | 177 | 0 | 177 | 1,360 | 3,230 | 2,366 | 0,275 |
| ash_alkalinit | 177 | 0 | 177 | 10,600 | 30,000 | 19,517 | 3,336 |
| magnesium | 177 | 0 | 177 | 70,000 | 162,000 | 99,588 | 14,174 |
| tot._phenols | 177 | 0 | 177 | 0,980 | 3,880 | 2,292 | 0,626 |
| flavonoids | 177 | 0 | 177 | 0,340 | 5,080 | 2,023 | 0,999 |
| non-flav._ph | 177 | 0 | 177 | 0,130 | 0,660 | 0,362 | 0,125 |
| proanth | 177 | 0 | 177 | 0,410 | 3,580 | 1,587 | 0,572 |
| col._int. | 177 | 0 | 177 | 1,280 | 13,000 | 5,055 | 2,324 |
| col._hue | 177 | 0 | 177 | 0,480 | 1,710 | 0,957 | 0,229 |
| OD_ratio | 177 | 0 | 177 | 1,270 | 4,000 | 2,604 | 0,705 |
| proline | 177 | 0 | 177 | 278,000 | 1680,000 | 745,096 | 314,884 |

# 1.6 Results in Excel

- XLSTAT-R imports in Excel what is computed by R


Codes plot:


**Résultats pour la variable inv :**

Goodness of fit statistics :

| | |
|---|---|
| rsq | 0,767 |
| adjrsq | 0,753 |

F statistic :

| statistic.F | arameter.d | arameter.d | p.value.F |
|---|---|---|---|
| 309,014 | 2 | 188 | < 0,0001 |

Coefficients :

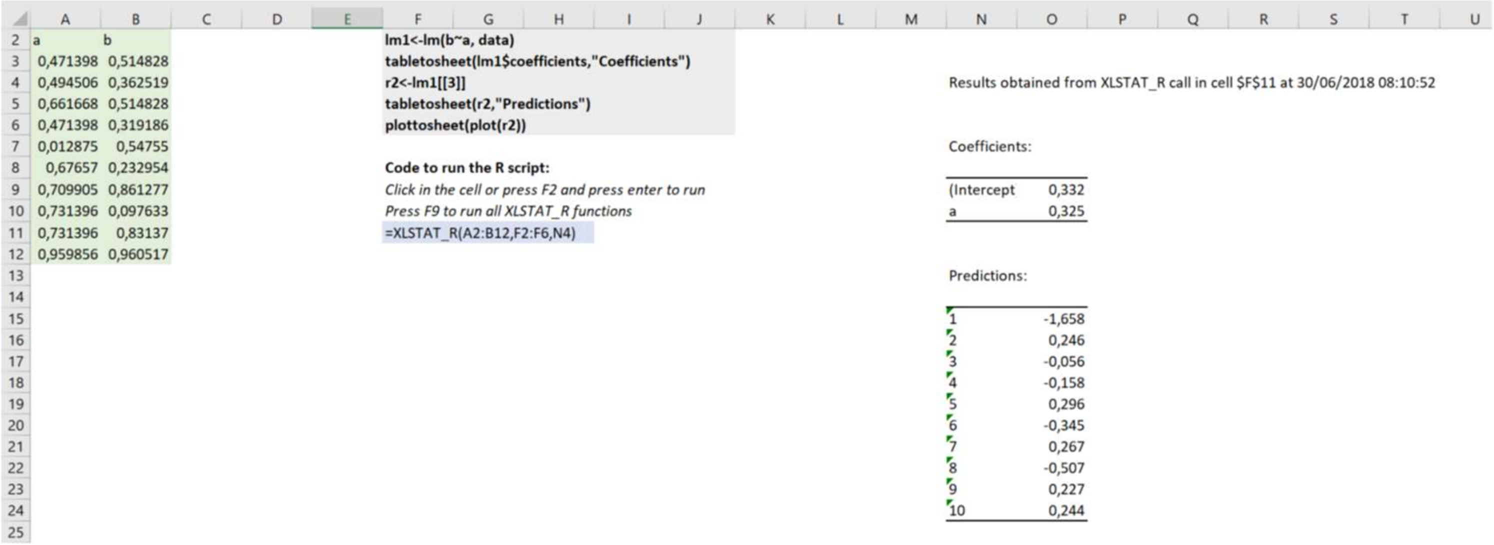
| | Estimate | Std. Error | t-value | Pr(>\|t\|) |
|---|---|---|---|---|
| value | 0,110 | 0,012 | 9,288 | < 0,0001 |
| capital | 0,310 | 0,017 | 17,867 | < 0,0001 |

# 2. Second approach: XLSTAT-R-Notebook

- Make R functions available in a sheet, with just one formula

=XLSTAT_R(Data,Rcode,OutPutCell)

- Example:



| a | b |
|---|---|
| 0,471398 | 0,514828 |
| 0,494506 | 0,362519 |
| 0,661668 | 0,514828 |
| 0,471398 | 0,319186 |
| 0,012875 | 0,54755 |
| 0,67657 | 0,232954 |
| 0,709905 | 0,861277 |
| 0,731396 | 0,097633 |
| 0,731396 | 0,83137 |
| 0,959856 | 0,960517 |

```
lm1<-lm(b~a, data)
tabletosheet(lm1$coefficients,"Coefficients")
r2<-lm1[[3]]
tabletosheet(r2,"Predictions")
plottosheet(plot(r2))
```

**Code to run the R script:**

*Click in the cell or press F2 and press enter to run*

*Press F9 to run all XLSTAT_R functions*

=XLSTAT_R(A2:B12,F2:F6,N4)

Results obtained from XLSTAT_R call in cell $F$11 at 30/06/2018 08:10:52

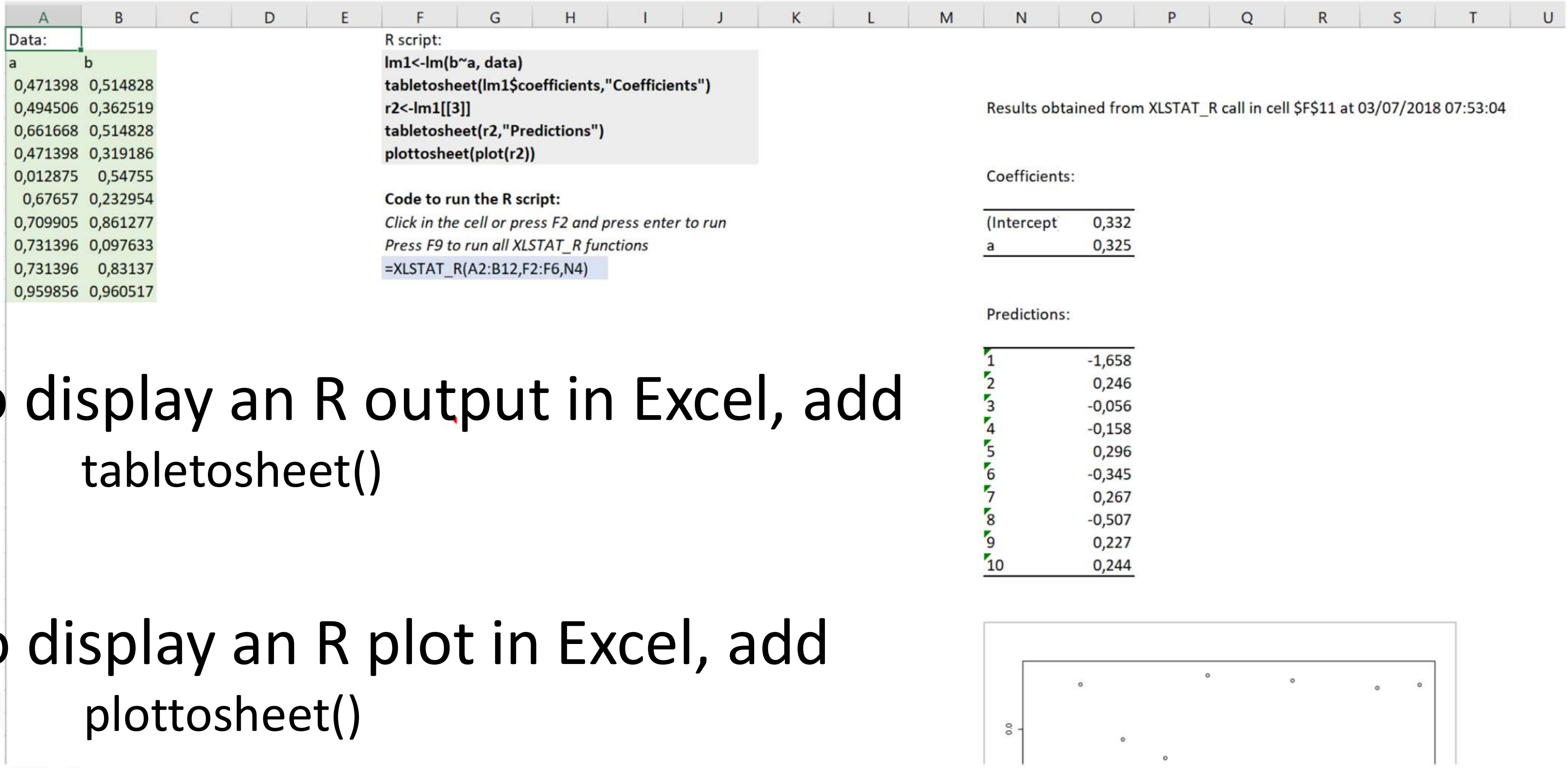
Coefficients:

| | |
|---|---|
| (Intercept | 0,332 |
| a | 0,325 |

Predictions:

| | |
|---|---|
| 1 | -1,658 |
| 2 | 0,246 |
| 3 | -0,056 |
| 4 | -0,158 |
| 5 | 0,296 |
| 6 | -0,345 |
| 7 | 0,267 |
| 8 | -0,507 |
| 9 | 0,227 |
| 10 | 0,244 |

# 2. Second approach: XLSTAT-R-Notebook



Data:

| a | b |
|---|---|
| 0,471398 | 0,514828 |
| 0,494506 | 0,362519 |
| 0,661668 | 0,514828 |
| 0,471398 | 0,319186 |
| 0,012875 | 0,54755 |
| 0,67657 | 0,232954 |
| 0,709905 | 0,861277 |
| 0,731396 | 0,097633 |
| 0,731396 | 0,83137 |
| 0,959856 | 0,960517 |

R script:

```
lm1<-lm(b~a, data)
tabletosheet(lm1$coefficients,"Coefficients")
r2<-lm1[[3]]
tabletosheet(r2,"Predictions")
plottosheet(plot(r2))
```

**Code to run the R script:**

*Click in the cell or press F2 and press enter to run*

*Press F9 to run all XLSTAT_R functions*

```
=XLSTAT_R(A2:B12,F2:F6,N4)
```

Results obtained from XLSTAT_R call in cell $F$11 at 03/07/2018 07:53:04

Coefficients:

| | |
|---|---|
| (Intercept | 0,332 |
| a | 0,325 |

Predictions:

| | |
|---|---|
| 1 | -1,658 |
| 2 | 0,246 |
| 3 | -0,056 |
| 4 | -0,158 |
| 5 | 0,296 |
| 6 | -0,345 |
| 7 | 0,267 |
| 8 | -0,507 |
| 9 | 0,227 |
| 10 | 0,244 |

To display an R output in Excel, add

tabletosheet()

To display an R plot in Excel, add

plottosheet()

# Contact us

- Please email us if:
    - You want us to make your R function available in XLSTAT-R
    - You want to send us an XLSTAT-R-Notebook that includes a scenario you would like to share
    - Contact: support@xlstat.com