

1 Topic

RExcel: a bridge between R and Excel.

Combining a specialized data mining tool with a spreadsheet is a very interesting idea. Most of the people know handle a spreadsheet such as Excel (but also [LibreOffice Calc](#), [Open Office Calc](#), [Gnumeric](#), etc.). It is really popular because it is a very easy to use tool for data manipulation¹.

Many data mining tools can read XLS or XLSX file formats. But, it is even more interesting to implement a bridge between the data mining tools and Excel in a bidirectional way. So, we can lead easily the whole analysis by navigating between the tools: transforming the variables into Excel, performing the analysis into the data mining tool, and post-processing the results into Excel.

In this tutorial, we describe RExcel library for R (<http://rcom.univie.ac.at/>). It sets a new menu into Excel. Thus, we can send a dataset to R on the one hand; retrieve dataset or more generally a vector or a matrix from R on the other hand. The tool is really easy to use.

2 Dataset

	A	B	C	D	E	F	G	H	I
1	MT	RG	PRIX	BR	INV	PUB	FV	TPUB	VENTES
2	369	118	59	9	17	89	177	225	5439
3	476	138	71	18	4	63	279	206	5149
4	432	152	73	16	-50	16	245	309	4704
5	418	135	79	35	142	74	270	83	5036
6	383	104	60	21	-45	32	201	298	4110
7	554	138	81	20	42	93	324	161	6180
8	320	147	66	15	10	48	154	305	4888
9	268	129	57	29	89	51	166	263	4290
10	359	106	69	27	71	74	196	414	5397
11	461	132	82	27	-18	91	267	170	5272
12	420	136	70	10	8	91	213	429	4989
13	536	111	73	27	128	74	296	273	5927
14	311	143	67	22	-25	27	181	60	4033
15	517	142	74	27	27	75	307	345	6124
16	332	140	60	11	61	21	180	247	4708
17	336	136	60	25	-30	40	213	328	4627
18	394	146	59	13	143	52	209	407	4872
19	415	148	69	8	47	29	207	80	5151
20									

We use the “[ventes_regression_rexcel.xlsx](#)” data file, from the Tenenhaus’ book (*in French*, « *Statistique – Méthodes pour décrire, expliquer et prévoir* », Dunod, 2007). By using a regression

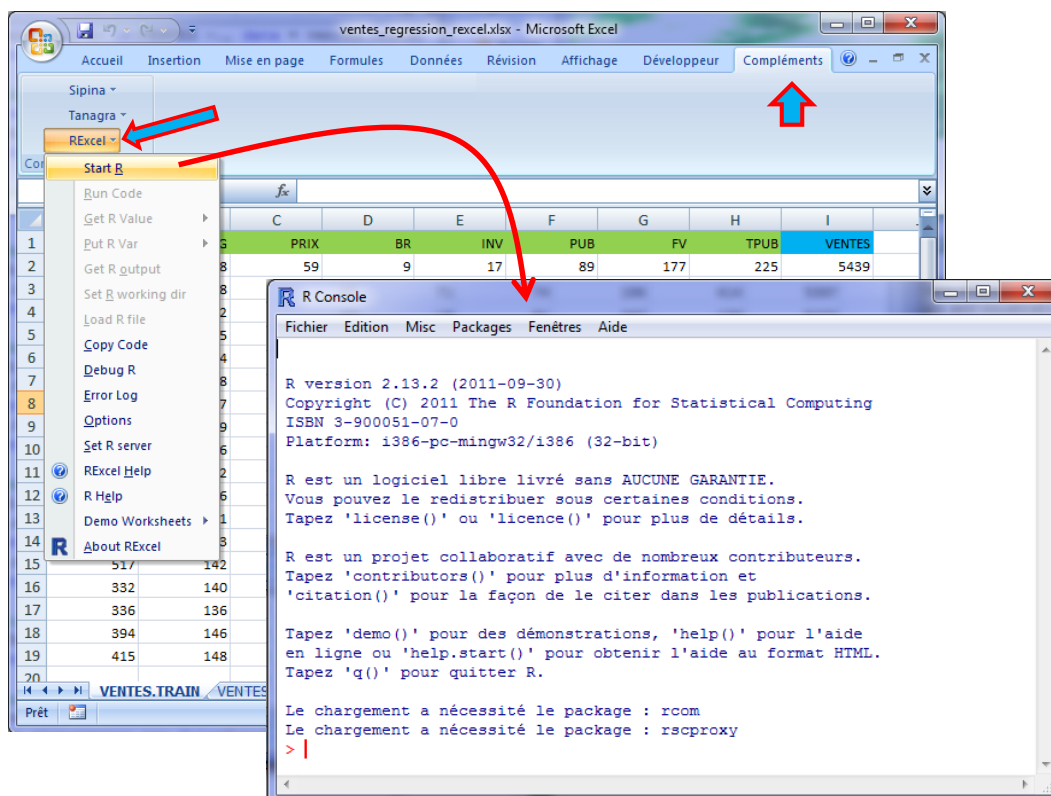
¹ <http://www.kdnuggets.com/polls/2011/tools-analytics-data-mining.html>: 3rd place for the poll « Data Mining/Analytic Tools Used » in 2011; in 2nd place in 2010.

approach, we want to explain the selling of a product (VENTES) from various independent variables (publicity, price, etc.). We have subdivided the dataset into two subsamples: 18 instances are used for the training phase (VENTES.TRAIN sheet), and 20 instances for the testing phase (VENTES.TEST).

3 Installing RExcel

Installation. The installation is difficult. It is necessary to follow carefully the instructions. I installed the tool as a standard package for R first (**RExcelInstaller**). Then, I follow the instructions described into the R console². Note that I set the instruction **installRExcel(ForegroundServer=TRUE)** to start the installation process. It seems that the option is necessary in order to make simultaneously visible RGUI and Excel.

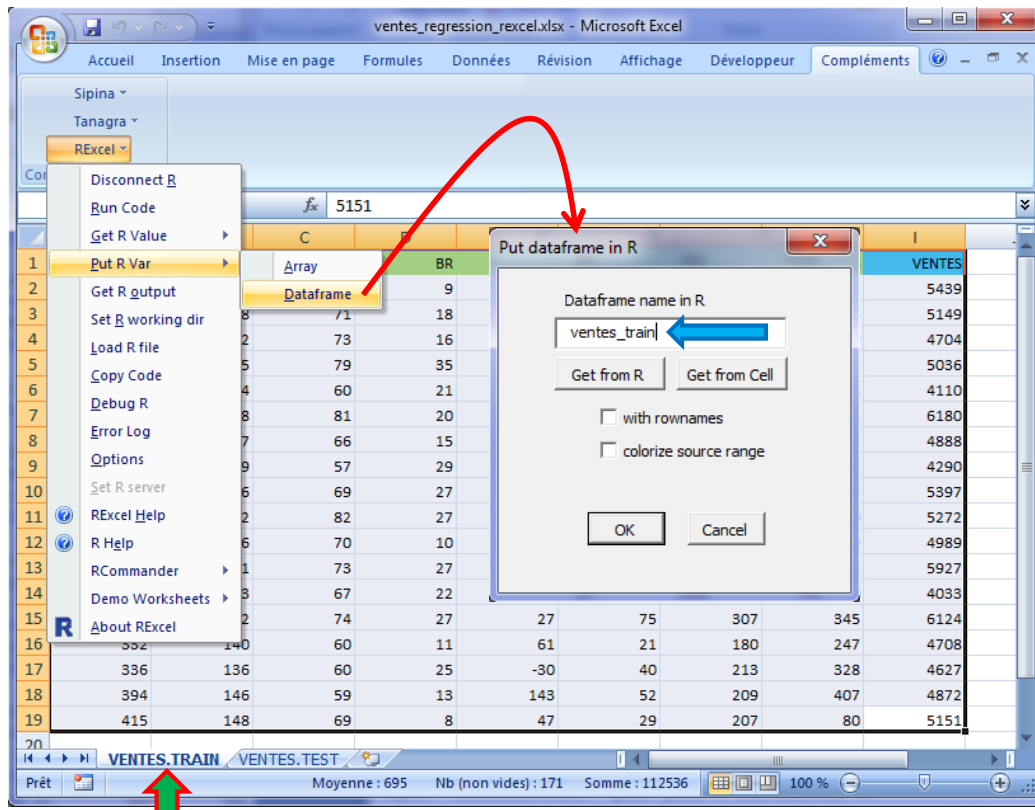
Making the connection between Excel and R. We launch Excel and we load the data file. We activate the first sheet. The RExcel menu should be visible into the "Add-Ins" tab ("Compléments" tab in French). First, we must launch R from Excel by clicking on the **START R** menu. The R Console is now available. We can enter R commands.



4 Sending the dataset from Excel to R

Transferring the training sample. We select the data range and we click on the **REXCEL / PUT R VAR / DATAFRAME** menu. We set the data.frame name that we can handle into R: **ventes_train**.

² See also http://learnserver.csd.univie.ac.at/rcomwiki/doku.php?id=wiki:how_to_install



To check the process, we select RGui and we use the `ls()` command. The data frame `ventes_train` is available into R memory. We obtain the descriptive statistics with the `summary(ventes_train)` command.

```

R Console
Fichier Edition Misc Packages Fenêtres Aide

> ls()
[1] "ventes_train"
> summary(ventes_train)
      MT          RG          PRIX          BR
Min.  :268.0   Min.  :104.0   Min.  :57.00   Min.   : 8.0
1st Qu.:341.8   1st Qu.:129.8   1st Qu.:60.00  1st Qu.:13.5
Median :404.5   Median :137.0   Median :69.00   Median :20.5
Mean   :405.6   Mean   :133.4   Mean   :68.28   Mean   :20.0
3rd Qu.:453.8   3rd Qu.:142.8   3rd Qu.:73.00   3rd Qu.:27.0
Max.   :554.0   Max.   :152.0   Max.   :82.00   Max.   :35.0
      INV          PUB          FV          TPUB
Min.  : -50.0   Min.  :16.00   Min.  :154.0   Min.   :60.0
1st Qu.: -12.5  1st Qu.:34.00  1st Qu.:184.8  1st Qu.:179.0
Median : 22.0   Median :57.50  Median :211.0  Median :268.0
Mean   : 34.5   Mean   :57.78  Mean   :226.9  Mean   :255.7
3rd Qu.: 68.5   3rd Qu.:74.75  3rd Qu.:269.2  3rd Qu.:323.2
Max.   :143.0   Max.   :93.00  Max.   :324.0  Max.   :429.0
      VENTES
Min.  :4033
1st Qu.:4705
Median :5012
Mean   :5050
3rd Qu.:5366
Max.   :6180
> |

```

Regression process. In a first time, we launch the regression analysis with all the available independent variables. In a second time, we perform a variable selection with the `stepAIC()` command of the **MASS** package. The selected variables are MT and PUB.

```

R Console
Fichier Edition Misc Packages Fenêtres Aide

> modele.full <- lm(VENTES ~., data = ventes_train)
> print(modele.full)

Call:
lm(formula = VENTES ~ ., data = ventes_train)

Coefficients:
(Intercept)      MT          RG          PRIX          BR          INV
 1862.5771    7.8966    5.2537   -3.8941    5.6258    1.6171
      PUB          FV          TPUB
   9.0311   -5.2378    0.1875

> library(MASS)
> modele.selection <- stepAIC(modele.full,direction="backward",trace=FALSE)
> print(modele.selection)

Call:
lm(formula = VENTES ~ MT + PUB, data = ventes_train)

Coefficients:
(Intercept)      MT          PUB
 2654.169    4.579    9.318

> |

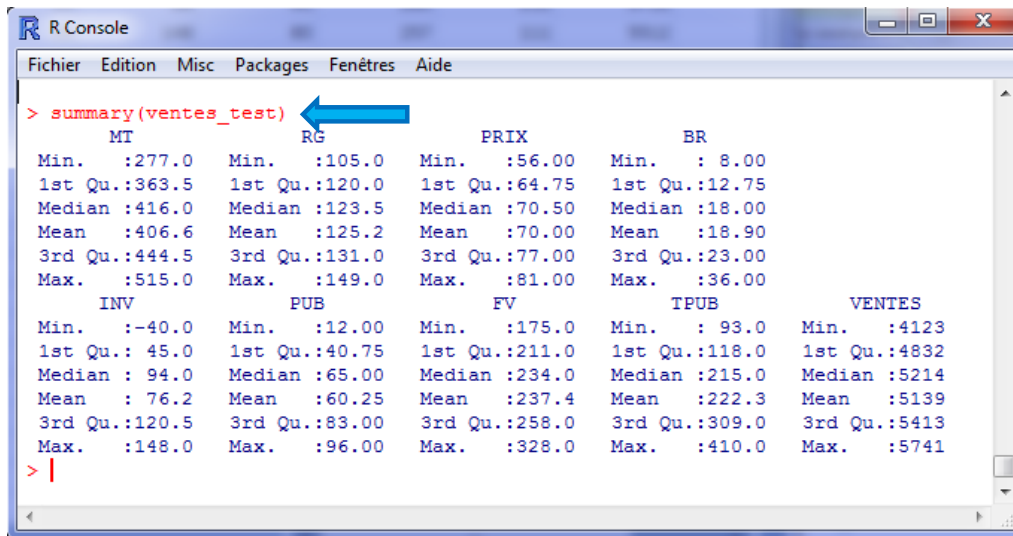
```

5 Prediction

The screenshot shows the Microsoft Excel interface with the RExcel add-in. The RExcel menu is open, and the 'Put R Var' option is selected. A dialog box titled 'Put dataframe in R' is displayed, showing the 'Dataframe name in R' field with the text 'ventes_test'. The dialog also includes options for 'with rownames' and 'colorize source range', and 'Get from R' and 'Get from Cell' buttons. The Excel spreadsheet in the background shows a table with columns labeled BR, INV, TPUB, PUB, and VENTES, and rows of data.

Transferring the test sample. We activate the VENTES.TEST sheet. We select the data range. We click on the **REXCEL / PUT R VAR / DATAFRAME** menu. We set “**ventes_test**” as data frame name.

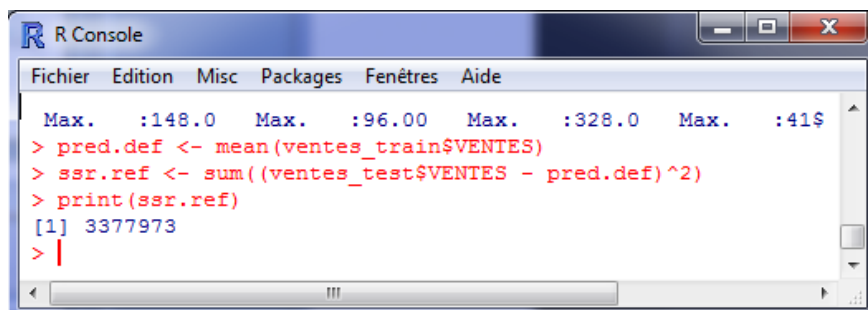
Into R, we check the dataset using the **summary(.)** command.



```

> summary(ventes_test)
  MT          RG          PRIX          BR
Min.   :277.0  Min.   :105.0  Min.   :56.00  Min.   : 8.00
1st Qu.:363.5  1st Qu.:120.0  1st Qu.:64.75  1st Qu.:12.75
Median :416.0  Median :123.5  Median :70.50  Median :18.00
Mean   :406.6  Mean   :125.2  Mean   :70.00  Mean   :18.90
3rd Qu.:444.5  3rd Qu.:131.0  3rd Qu.:77.00  3rd Qu.:23.00
Max.   :515.0  Max.   :149.0  Max.   :81.00  Max.   :36.00
  INV          PUB          FV          TPUB          VENTES
Min.   :-40.0  Min.   :12.00  Min.   :175.0  Min.   : 93.0  Min.   :4123
1st Qu.: 45.0  1st Qu.:40.75  1st Qu.:211.0  1st Qu.:118.0  1st Qu.:4832
Median : 94.0  Median :65.00  Median :234.0  Median :215.0  Median :5214
Mean   : 76.2  Mean   :60.25  Mean   :237.4  Mean   :222.3  Mean   :5139
3rd Qu.:120.5  3rd Qu.:83.00  3rd Qu.:258.0  3rd Qu.:309.0  3rd Qu.:5413
Max.   :148.0  Max.   :96.00  Max.   :328.0  Max.   :410.0  Max.   :5741
  
```

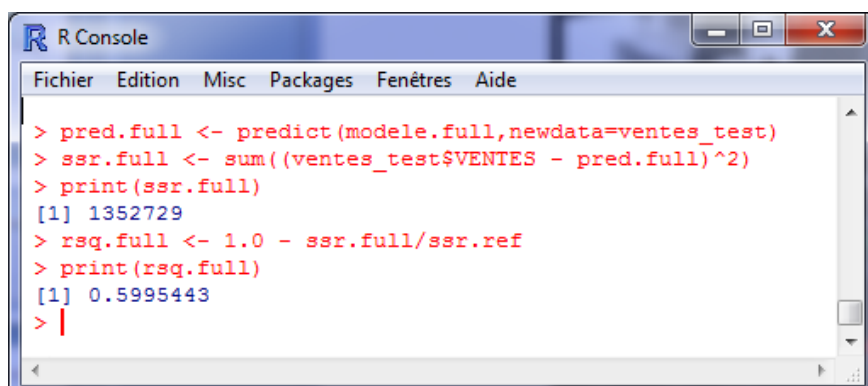
Prediction. We want to compare three types of predictions. The reference model is the default prediction (PRED.DEF). It corresponds to the mean of the target attribute computed on the learning sample. We obtain the residual sum of squares **SSR.REF = 3377973** on the test sample. The other models (MODEL.FULL with all the independent variables, and MODEL.SELECTION with a selected subset of available independent variables), which use the independent variables to predict the values of the target attribute, must be better i.e. their SSR must be lower than this reference value.



```

Max.   :148.0  Max.   :96.00  Max.   :328.0  Max.   :41$
> pred.def <- mean(ventes_train$VENTES)
> ssr.ref <- sum((ventes_test$VENTES - pred.def)^2)
> print(ssr.ref)
[1] 3377973
>
  
```

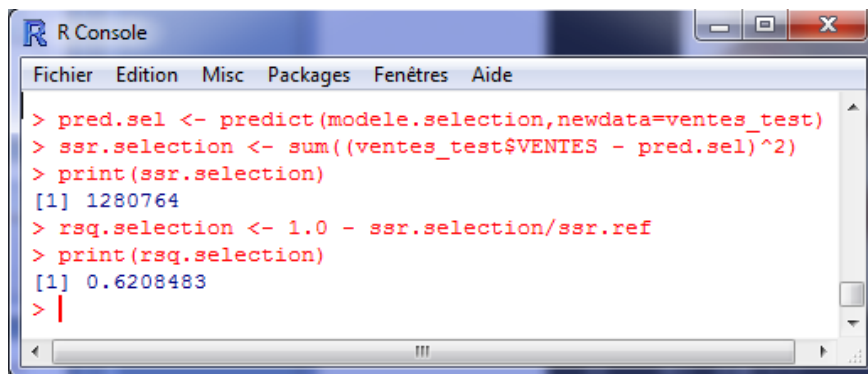
MODELE.FULL uses all the available independent variables. We obtain **SSR.FULL = 1352729**. The Pseudo-R-squared is **59.95%** i.e. we decrease the SSR by 59.95% compared to the default model.



```

> pred.full <- predict(modele.full,newdata=ventes_test)
> ssr.full <- sum((ventes_test$VENTES - pred.full)^2)
> print(ssr.full)
[1] 1352729
> rsq.full <- 1.0 - ssr.full/ssr.ref
> print(rsq.full)
[1] 0.5995443
>
  
```

The third model uses only MT and PUB. It is the most efficient with **SSR.SEL = 1280764** i.e. the pseudo-r-squared is **RSQ.SELECTION = 62.08%** compared with the default model.

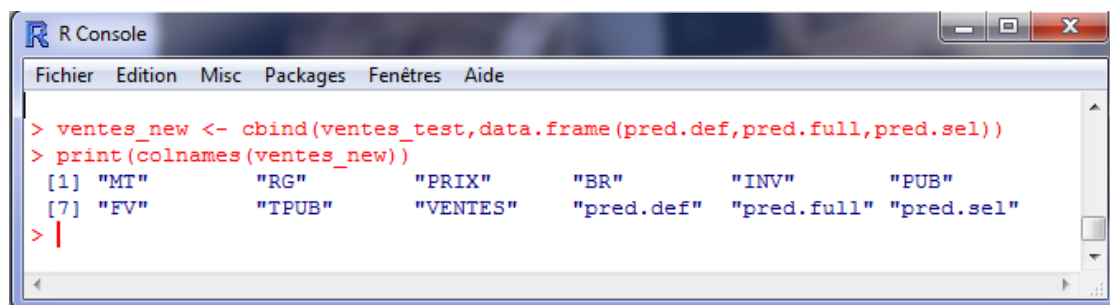


```

> pred.sel <- predict(modele.selection,newdata=ventes_test)
> ssr.selection <- sum((ventes_test$VENTES - pred.sel)^2)
> print(ssr.selection)
[1] 1280764
> rsq.selection <- 1.0 - ssr.selection/ssr.ref
> print(rsq.selection)
[1] 0.6208483
> |

```

Retrieving the predictions into Excel. We want to retrieve the test sample with the predictions into Excel. We create a new data frame into R (**ventes_new**).

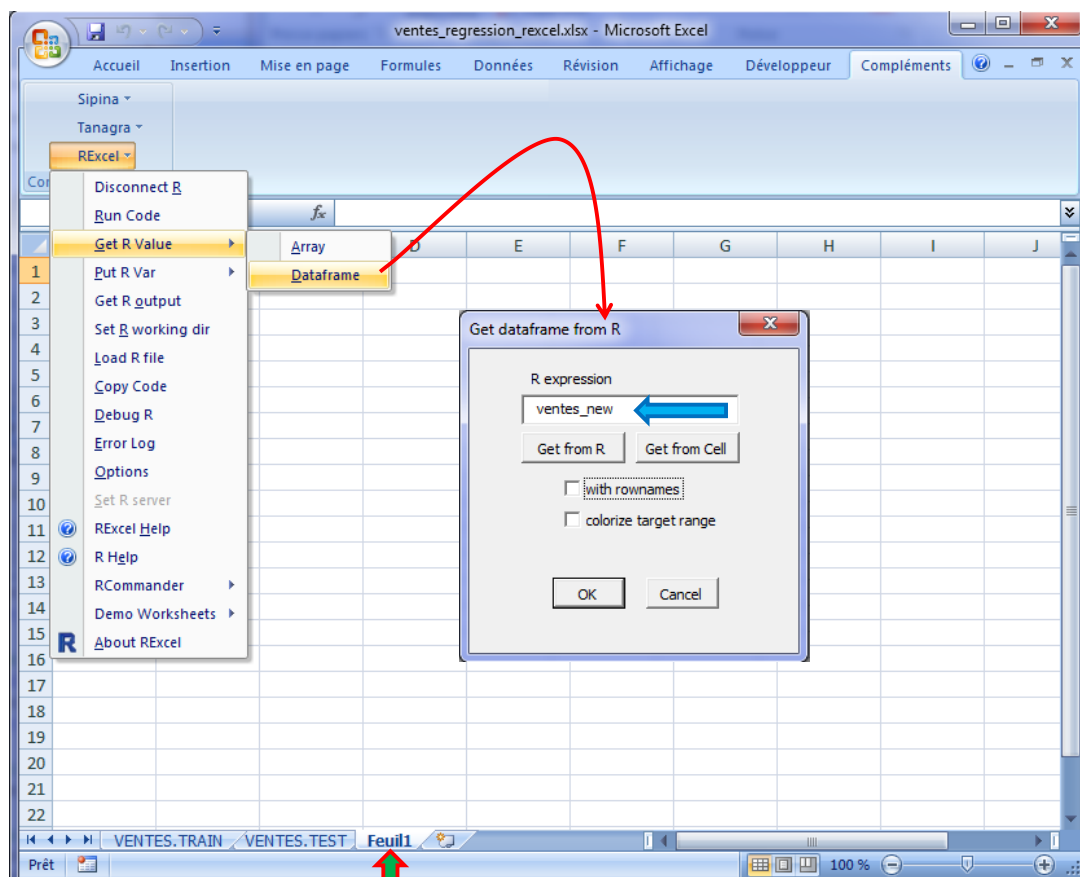


```

> ventes_new <- cbind(ventes_test,data.frame(pred.def,pred.full,pred.sel))
> print(colnames(ventes_new))
[1] "MT"      "RG"      "PRIX"    "BR"      "INV"     "PUB"
[7] "FV"      "TPUB"    "VENTES"  "pred.def" "pred.full" "pred.sel"
> |

```

Then, we activate REXCEL / GET R VALUE / DATAFRAME into Excel. We set the data frame name.



We obtain the test sample with additional columns corresponding to the predictions of models.

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L
1	MT	RG	PRIX	BR	INV	PUB	FV	TPUB	VENTES	pred.def	pred.full	pred.sel
2	328	123	77	20	59	88	211	141	4787	5049.77778	4722.93372	4976.03427
3	285	105	63	8	-28	12	176	218	4123	5049.77778	3646.52816	4070.94805
4	441	120	80	16	-22	50	267	405	4801	5049.77778	4847.31203	5139.34015
5	462	112	73	15	68	93	283	212	5712	5049.77778	5406.63009	5636.18582
6	417	120	81	35	148	83	257	111	5512	5049.77778	5330.98243	5336.95578
7	408	131	66	13	120	62	235	141	5313	5049.77778	5138.2724	5100.06035
8	362	145	67	23	117	73	220	239	4942	5049.77778	5092.37986	4991.93771
9	436	123	73	32	100	43	276	280	5366	5049.77778	5004.35614	5051.2174
10	456	128	65	22	144	52	253	93	5741	5049.77778	5401.29207	5226.65898
11	364	120	64	14	128	96	195	107	5383	5049.77778	5269.57993	5215.4183
12	433	124	68	8	122	25	258	291	5140	5049.77778	4839.73185	4869.74996
13	277	135	62	11	76	68	175	410	4842	5049.77778	4476.90264	4556.1475
14	455	126	78	22	18	95	233	118	5316	5049.77778	5626.28794	5622.77098
15	398	138	56	12	50	77	229	98	5540	5049.77778	5175.02956	5194.04818
16	412	149	78	36	30	26	258	124	4647	5049.77778	4752.76884	4782.91349
17	415	119	75	20	-40	41	211	315	4630	5049.77778	4844.77614	4936.42576
18	484	111	58	13	107	40	258	321	5502	5049.77778	5358.0644	5243.04477
19	515	120	77	23	126	21	328	398	5288	5049.77778	5139.33476	5207.93824
20	429	125	74	11	88	83	218	118	5095	5049.77778	5452.80954	5391.90141
21	355	131	65	24	113	77	208	307	5094	5049.77778	5082.22637	4997.15965
22												

6 Retrieving other objects

We can retrieve other objects from R to Excel using RExcel. But only vectors or matrix can be handled. We cannot retrieve more complex objects such as the "lm" objet, etc.

In this section, we show how to obtain the coefficients of the regression. First, into R, we copy these coefficients into a vector object named "coefs".

```

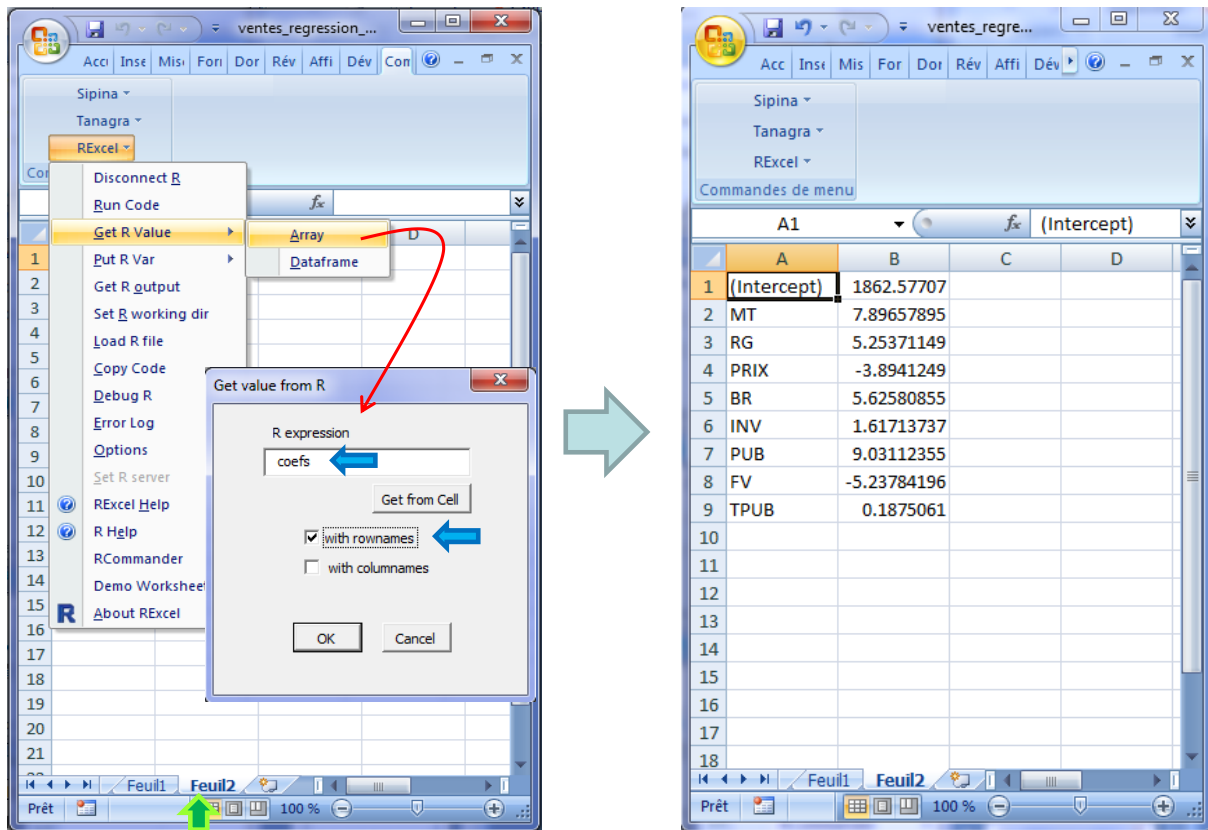
R Console
Fichier Edition Misc Packages Fenêtres Aide

> coefs <- modele.full$coefficients
> print(names(coefs))
[1] "(Intercept)" "MT"          "RG"          "PRIX"        "BR"
[6] "INV"          "PUB"         "FV"          "TPUB"
> |

```

Then, into Excel, we select an empty cell and we click on the **REXCEL / GET R VALUE / ARRAY** menu. We set the name of the vector. The « with rownames » option allows to retrieve the names of the coefficients.

We obtain the values of the model parameters with the associated variables names.



7 Reading an XLS file with the XLSLX package

Interactivity is the main asset of the RExcel library. It is really interesting during the exploration phase, when we try various models in order to detect the best one. Some calculations are easy to perform into Excel; others are easier to perform into R. Because the exchange is made easier, we can choose the best tool for each task.

By contrast, if our problem is mainly to read and write an Excel file format (XLS or XLSX), it is more appropriate to use a standard package such as XLSX³ (<http://cran.r-project.org/web/packages/xlsx/>).

For the analysis described in this tutorial, we use the following commands.

```
#set your data file directory
setwd("../")

#loading the training sample from the first sheet
library(xlsx)
ventes_train
read.xlsx(file="ventes_regression_rexcel.xlsx",sheetName="VENTES.TRAIN")
print(summary(ventes_train))

#performing the regression with all the available independent variables
modele.full <- lm(VENTES ~., data = ventes_train)
print(summary(modele.full))
```

³ Previously, I used the xlsReadWrite package (<http://cran.r-project.org/web/packages/xlsReadWrite/index.html>). But it cannot operate in a 64 bit mode.


```
#performing a variable selection with the stepAIC(.) procedure
library(MASS)
modele.selection <- stepAIC(modele.full,direction="backward")
print(summary(modele.selection))

#loading the test sample from the second sheet
ventes_test <-
read.xlsx(file="ventes_regression_rexcel.xlsx",sheetName="VENTES.TEST")

#prediction of the default model
pred.def <- mean(ventes_train$VENTES)
ssr.ref <- sum((ventes_test$VENTES - pred.def)^2)
print(ssr.ref)

#prediction of the model with all the variables
pred.full <- predict(modele.full,newdata=ventes_test)
ssr.full <- sum((ventes_test$VENTES - pred.full)^2)
print(ssr.full)
rsq.full <- 1.0 - ssr.full/ssr.ref
print(rsq.full)

#prediction of the model with the selected variables
pred.sel <- predict(modele.selection,newdata=ventes_test)
ssr.selection <- sum((ventes_test$VENTES - pred.sel)^2)
print(ssr.selection)
rsq.selection <- 1.0 - ssr.selection/ssr.ref
print(rsq.selection)

# gathering the coefficients of the model with all the variables
coefs <- modele.full$coefficients
print(coefs)

#creating a new data frame for the exportation of the test sample
#with the predictions of the models
ventes_new <- cbind(ventes_test,data.frame(pred.def,pred.full,pred.sel))
#writing the output file
write.xlsx(ventes_new,file="ventes_output.xlsx",row.names=F)
```

The main difference with the interactive mode is that we use the **read.xlsx(.)** and **write.xlsx(.)** commands for the data importation and exportation.

8 Conclusion

RExcel includes far more interesting features than data transfer. We can for instance call a R function as an Excel function with the RApply() command. We can also use the R functions into a VBA program. For more information, a video is available on the author's website (<http://rcom.univie.ac.at/RExcelDemo/>). I think that learn how to use RExcel is really a good idea.