1 Objectif

Description succincte de la bibliothèque RExcel établissant une passerelle entre Excel et R.

Le couplage entre un logiciel spécialisé de data mining et un tableur est un argument certain pour la praticabilité du premier. Quasiment tout le monde sait manipuler un tableur, ne serait-ce que pour ouvrir un fichier de données et en visualiser le contenu. De même, les opérations de vérification, les calculs statistiques simples, les transformations de données, sont très facilement réalisables dans un tableur. D'ailleurs, un signe qui ne trompe pas, outre les enquêtes du site KDNUGGETS qui montre la popularité d'Excel auprès des data miners[1], tous les logiciels dignes de ce nom savent importer directement les fichiers au format XLS.

Mettre en place une passerelle simplifiée entre Excel et un logiciel de data mining ou de statistique est donc un atout fort[2]. En prenant un exemple qui nous touche de près, depuis que la macro-complémentaire tanagra .xla[3] a été mise en place, je ne reçois presque plus d'e-mail posant des questions concernant l'importation des données. Je dispose de plus de temps pour répondre à d'autres questions. Tout le monde s'en porte mieux.

Très récemment, un étudiant me demandait s'il existait l'équivalent pour R. L'enjeu n'est pas tant l'importation des données au format XLS (Excel 2003 et antérieures) ou même XLSX (Excel 2007 et 2010), des packages s'en chargent très bien (Section 7), mais surtout de disposer des possibilités d'échanges bidirectionnels entre Excel et R, que ce soit pour les data frame (ensemble de données) ou, plus généralement, pour tout vecteur et matrice de données. En cherchant un peu, très rapidement, la réponse a été oui. RExcel répond exactement à ce cahier des charges. En fouillant un peu plus, je me suis même rendu compte que la solution proposée est de très grande qualité et va nettement au-delà du simple échange de vecteurs de valeurs.

Nous présentons donc la bibliothèque RExcel (http://rcom.univie.ac.at/) dans ce tutoriel. Nous nous contenterons de décrire le transfert des données. Nous ferrons un très rapide tour d'horizon des autres fonctionnalités dans la conclusion.

2 Données

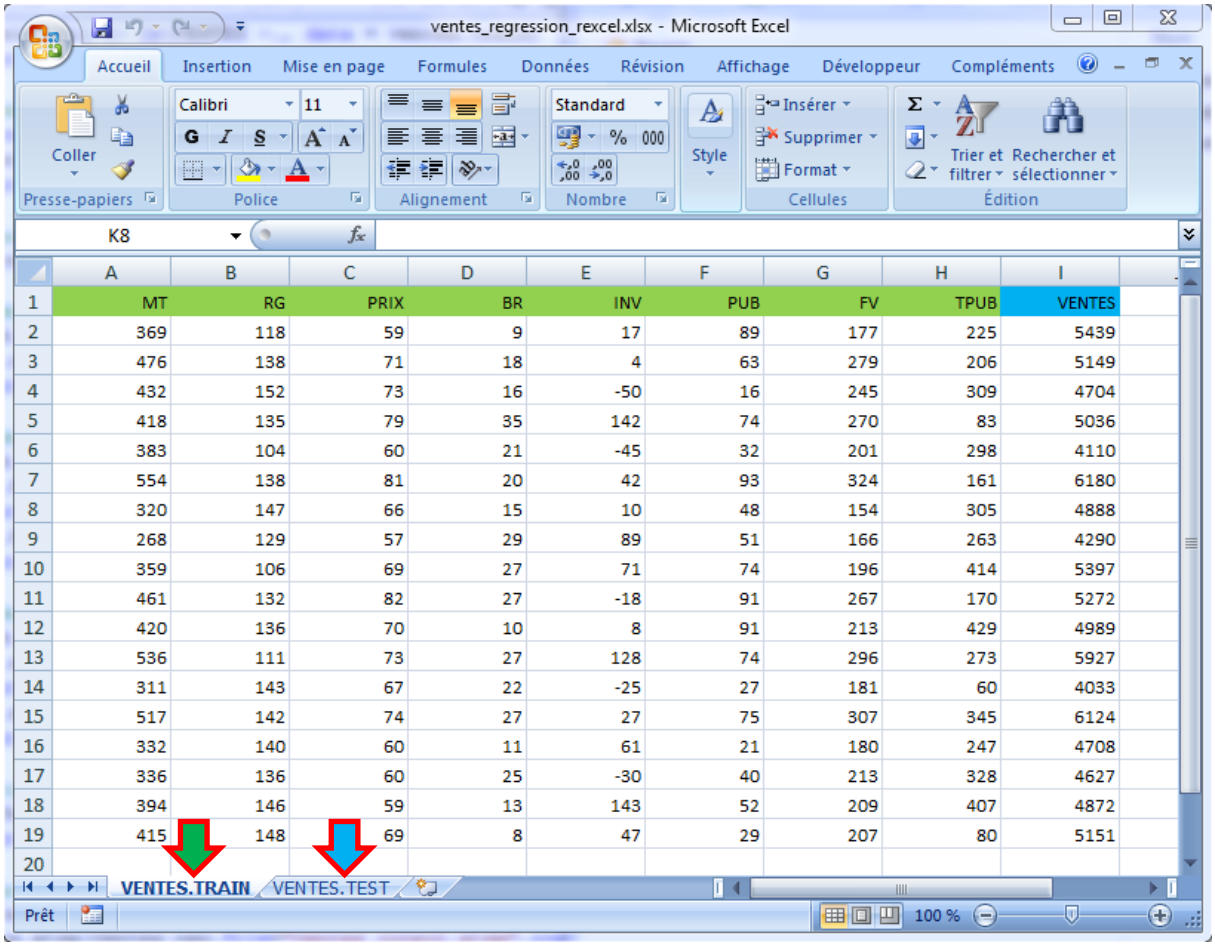
Pour illustrer notre propos, nous utilisons le fichier « ventes_regression_rexcel.xlsx », issu de l'ouvrage de Michel Tenenhaus (« Statistique – Méthodes pour décrire, expliquer et prévoir », Dunod, 2007 ; tableau 5.1, page 101). L'objectif est d'expliquer les ventes semestrielles d'un produit (VENTE) en fonction de plusieurs variables (publicité, prix, etc.).

[1] http://www.kdnuggets.com/polls/2011/tools-analytics-data-mining.html : en 3e position pour le sondage « Data Mining/Analytic Tools Used » de mai 2011 ; en 2nde position en 2010.

[2] Nous pouvons facilement étendre notre propos aux tableurs des distributions libres LibreOffice ou OpenOffice.

[3] Voir http://tutoriels-data-mining.blogspot.com/2011/12/ladd-in-tanagra-pour-excel-2010-64-bits.html pour Excel en 64 bits, et http://tutoriels-data-mining.blogspot.com/2010/08/ladd-in-tanagra-pour-excel-2007-et-2010.html pour les versions 32 bits. De même, un add-on existe pour OpenOffice et LibreOffice, voir http://tutoriels-data-mining.blogspot.com/2011/07/tanagra-addon-pour-openoffice-33.html. Ce dernier est fonctionnel sous Linux : http://tutoriels-data-mining.blogspot.com/2009/04/connexion-open-office-calc-sous-linux.html

Nous avons scindé aléatoirement la base en deux parties : 18 observations (feuille VENTES.TRAIN) sont réservées pour la construction du modèle par la régression linéaire multiple ; 20 observations (feuille VENTES.TEST) pour son évaluation c.-à-d. sur lesquelles nous calculerons la SSR – somme des carrés des erreurs de prédiction.

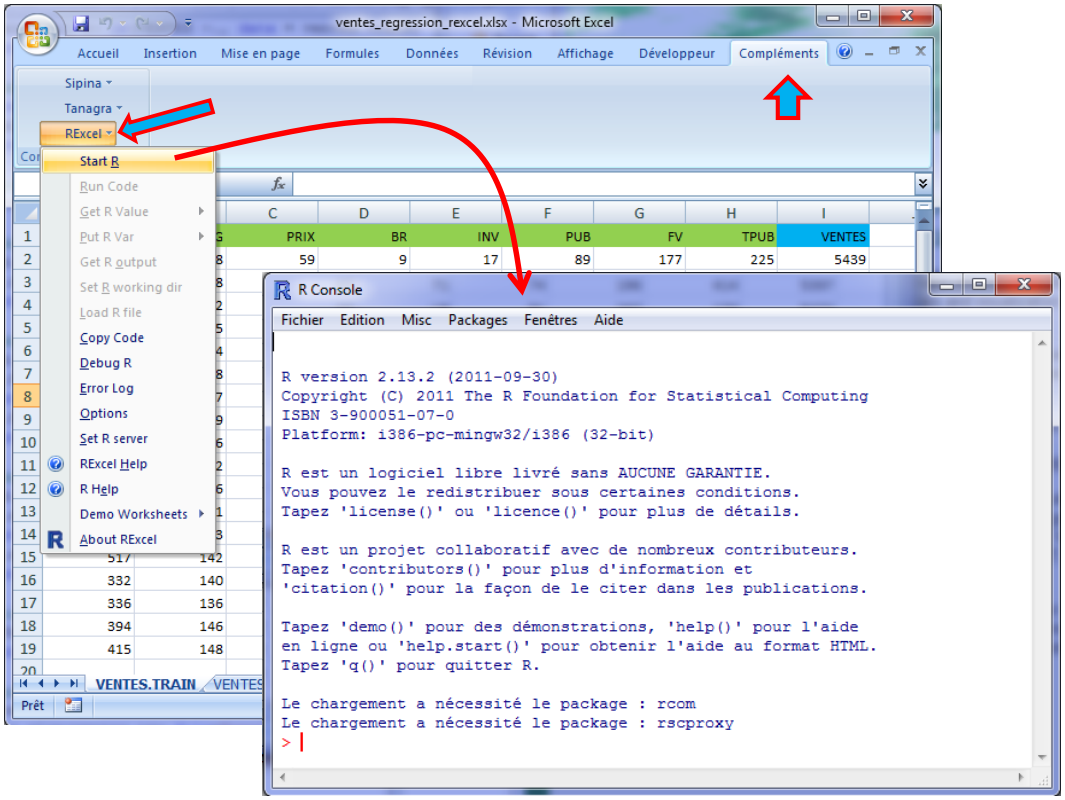


	A	B	C	D	E	F	G	H	I
1	MT	RG	PRIX	BR	INV	PUB	FV	TPUB	VENTES
2	369	118	59	9	17	89	177	225	5439
3	476	138	71	18	4	63	279	206	5149
4	432	152	73	16	-50	16	245	309	4704
5	418	135	79	35	142	74	270	83	5036
6	383	104	60	21	-45	32	201	298	4110
7	554	138	81	20	42	93	324	161	6180
8	320	147	66	15	10	48	154	305	4888
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									

3 Installation et connexion

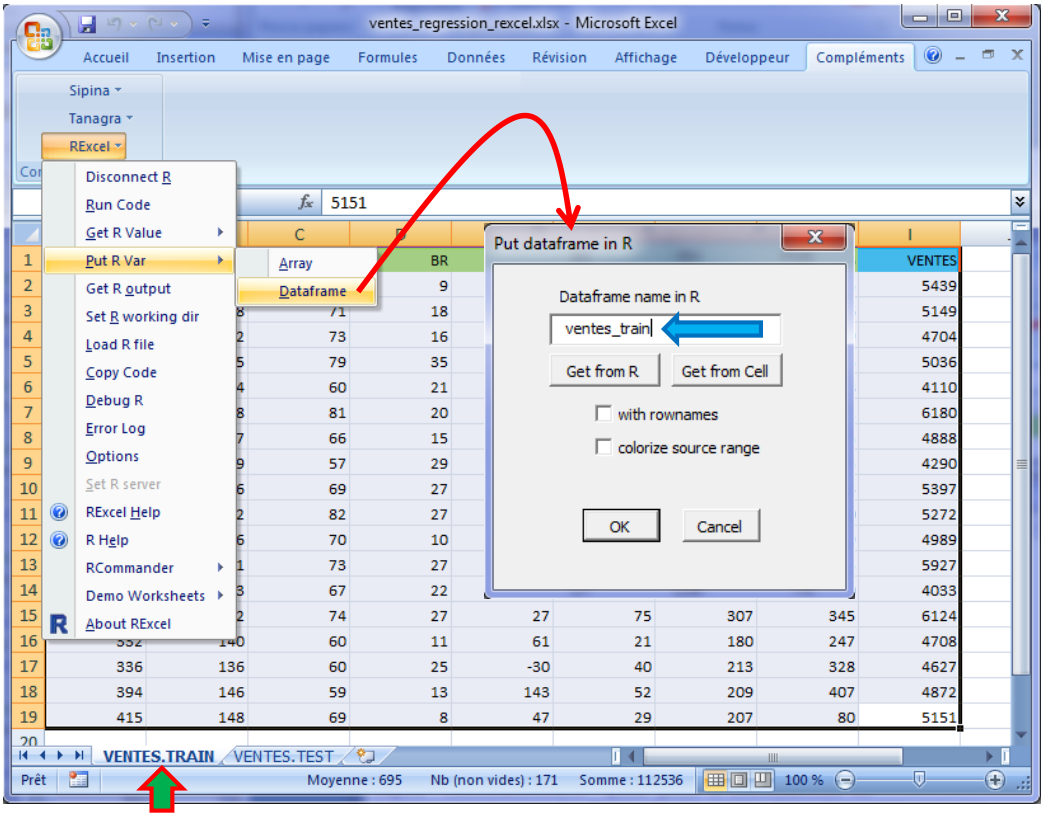
Installation. L'installation n'est pas très aisée. Il faut suivre scrupuleusement les innombrables instructions qui émaillent le processus. En ce qui me concerne, j'ai installé l'outil comme un package classique de R dans un premier temps (**RExcelInstaller**). Ensuite, j'ai suivi à la lettre les instructions affichées dans la console R (voir aussi les indications sur le wiki des auteurs de la bibliothèque : http://learnserver.csd.univie.ac.at/rcomwiki/doku.php?id=wiki:how_to_install). Notons que j'ai utilisé l'instruction **installRExcel(ForegroundServer=TRUE)** pour lancer la procédure d'installation. L'option semble nécessaire pour que RGUI soit simultanément visible avec Excel, avec la possibilité pour nous d'y saisir directement les instructions en langage R.

Connexion. Nous démarrons Excel et chargeons le fichier de données. Nous nous plaçons dans la première feuille VENTES.TRAIN puis nous sélectionnons l'onglet COMPLEMENTS. Le menu REXCEL devrait être présent si l'installation s'est déroulée correctement. La première étape consiste à initier la connexion entre Excel et R. Pour ce faire, nous actionnons le menu START R. Le logiciel R est automatiquement démarré. Il est accessible dans la barre de tâches de Windows.

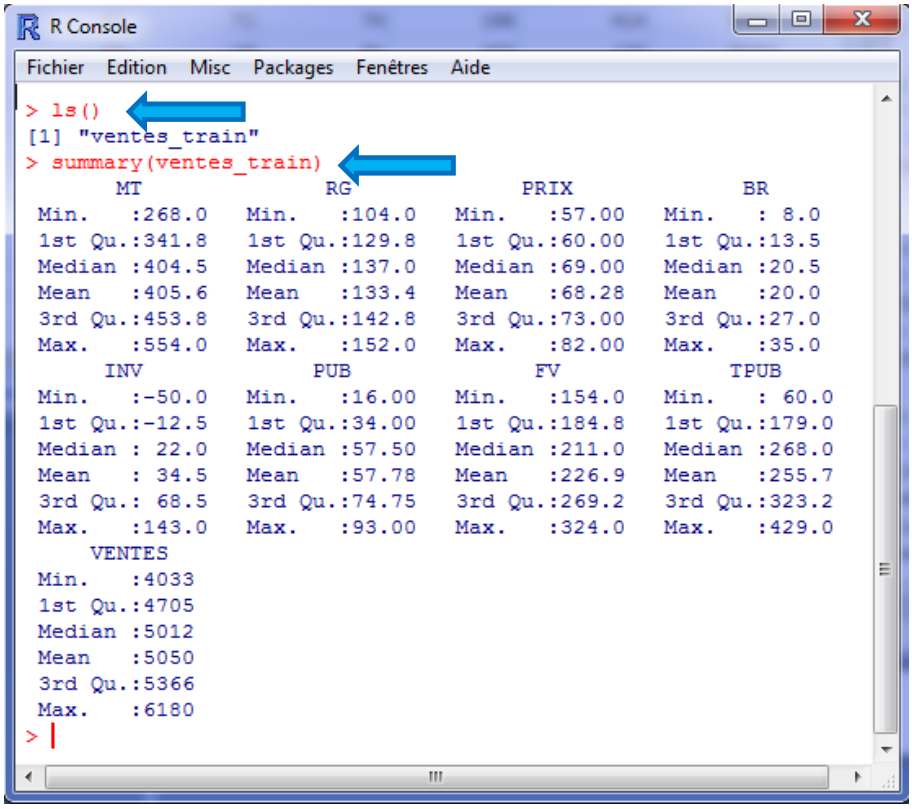


4 Envoi des données d'Excel vers R et modélisation

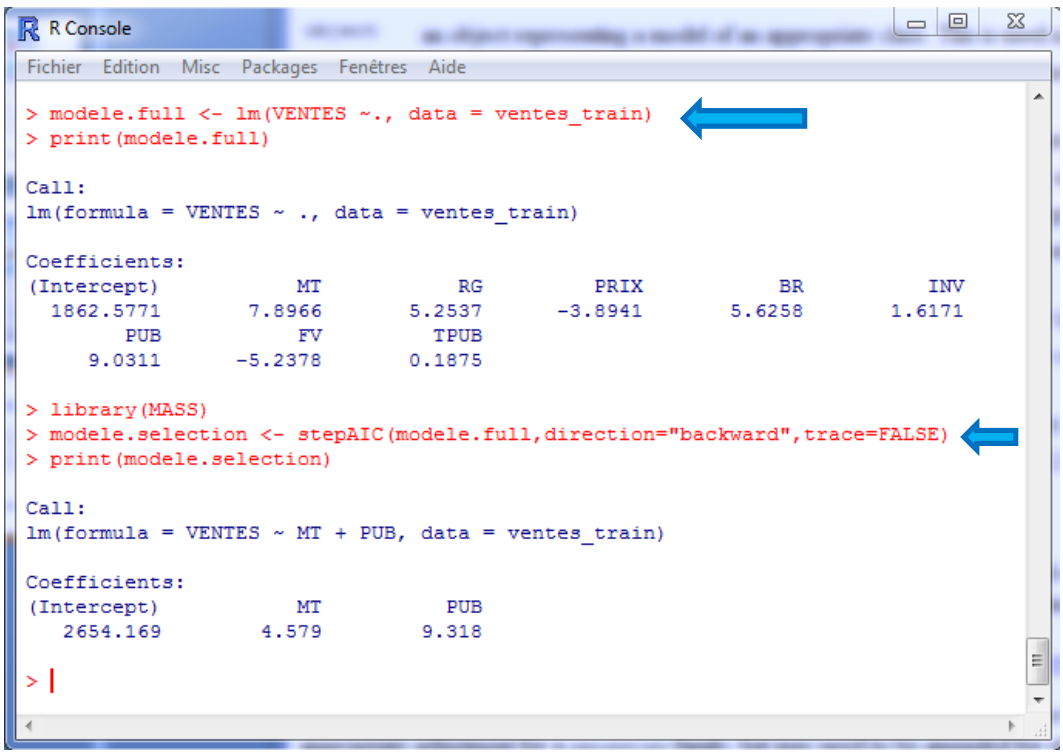
Transfert de l'échantillon d'apprentissage. Nous sélectionnons la plage de valeurs (feuille VENTES.TRAIN) puis nous cliquons sur le menu REXCEL / PUT R VAR / DATAFRAME. Une boîte de dialogue apparaît, nous spécifions le nom du data frame envoyé vers R : **ventes_train**.



Pour vérifier le transfert, nous activons R et nous demandons l'affichage des objets disponibles en mémoire avec **ls()**. VENTES_TRAIN est bien présent. Nous calculons les statistiques descriptives avec **summary(ventes_train)**.

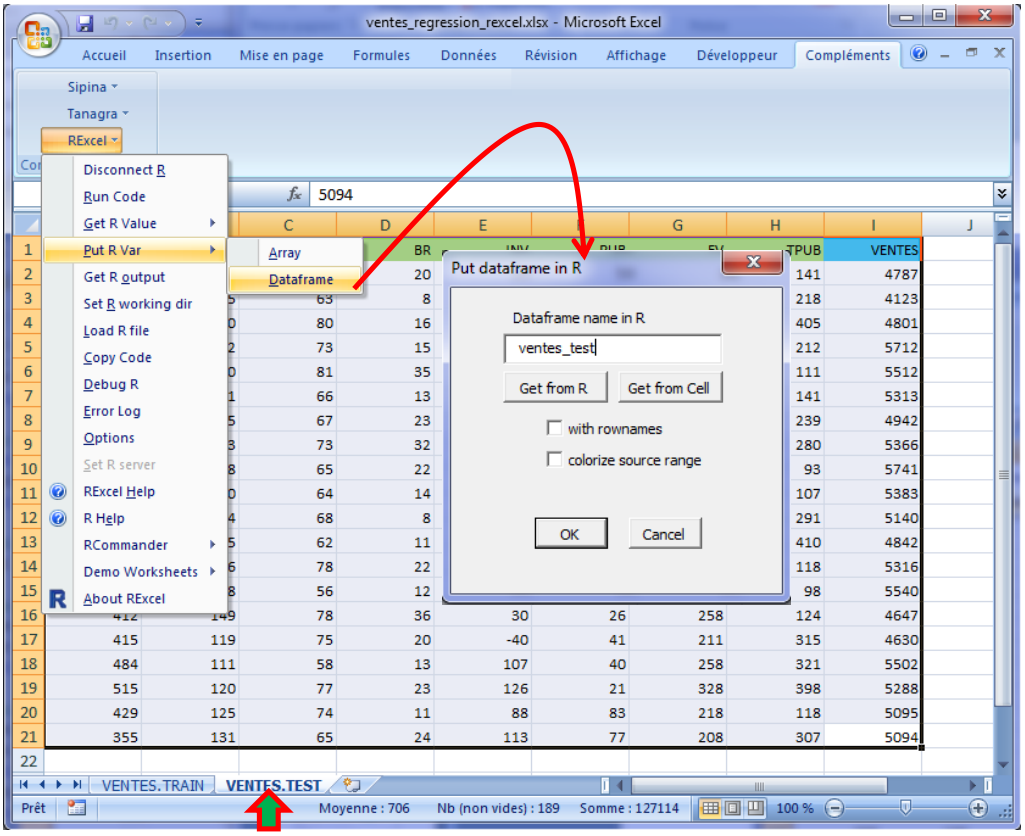


Modélisation. Il ne nous reste plus qu'à lancer la régression sur l'échantillon d'apprentissage. Le premier modèle exploite toutes les variables prédictives disponibles, le second opère une sélection via la procédure **stepAIC(.)** du package **MASS** (seules les variables MT et PUB sont retenues).

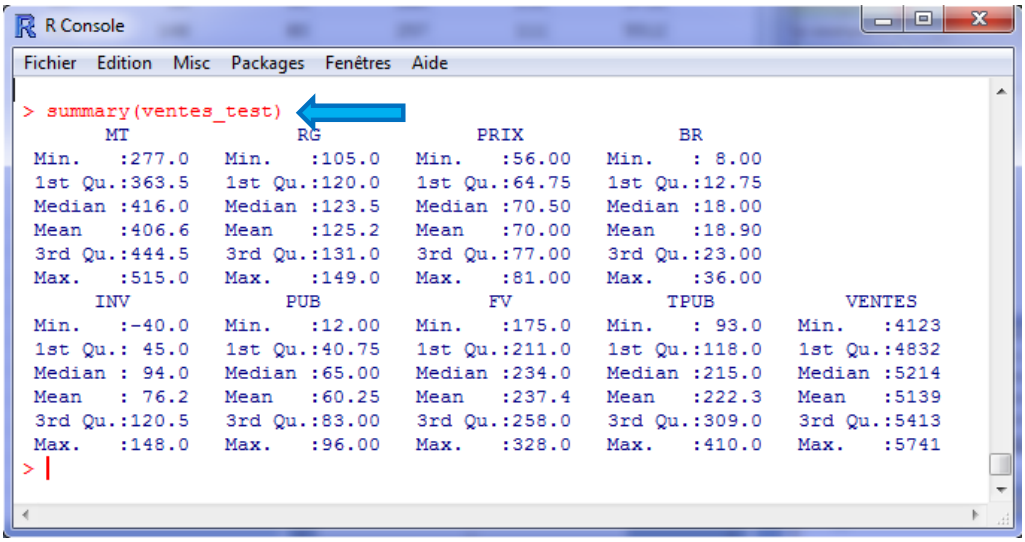


5 Prédictions et récupération des données de R vers Excel

Transfert de l'échantillon test. Tout d'abord, nous activons le second onglet VENTES.TEST. Nous sélectionnons les données et nous actionnons le menu REXCEL / PUT R VAR / DATAFRAME. Nous spécifions le nom du data frame : **ventes_test**.

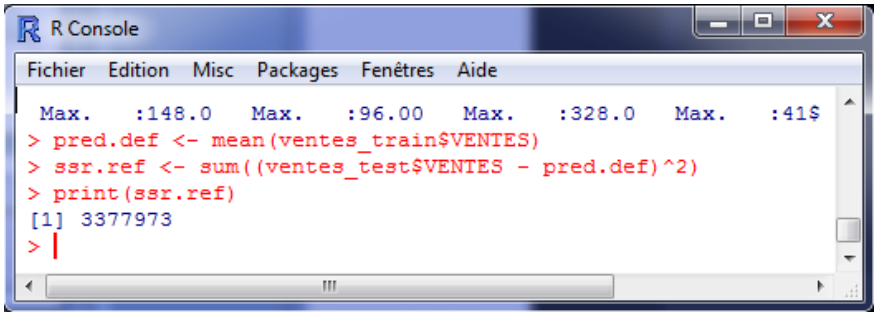


Dans R, nous vérifions le bon fonctionnement de la transmission en calculant les statistiques descriptives à l'aide de la commande **summary(.)**.

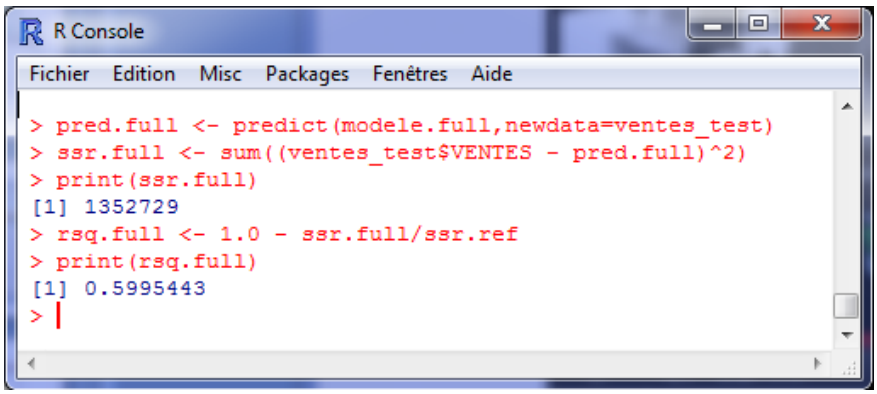


Prédictions. Nous opposerons 3 types de prédictions dans ce qui suit. La prédiction triviale (PRED.DEF) correspond à celle du modèle qui n'exploite pas les variables prédictives. On parle de « modèle par défaut » dans la communauté du data mining. Il s'agit tout simplement d'une

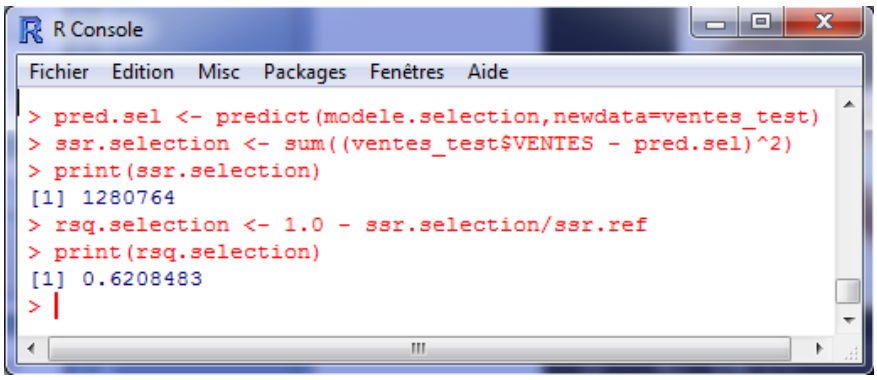
régression formée par la seule constante (c.-à-d. $y = b + \varepsilon$). Et on sait que l'estimateur des moindres carrés ordinaires de la constante est dans ce cas $\hat{b} = \bar{y}$, en l'occurrence la moyenne de la variable endogène VENTE, ***obtenue sur l'échantillon d'apprentissage***. Nous calculons alors la somme des carrés des erreurs de prédiction (**SSR.REF = 3377973**). Ce sera notre valeur de référence. On s'attend à ce que les autres modèles, ceux qui s'appuient sur tout ou partie des variables exogènes disponibles, fassent mieux.



Le second modèle justement (MODELE.FULL) utilise toutes les variables. Nous construisons la prédiction (PRED.FULL) à l'aide de la commande PREDICT. L'erreur de prédiction est **SSR.FULL = 1352729**. Elle réduit l'erreur de **59.95%** par rapport à la référence (RSQ.FULL, on parle de **pseudo-R^2,** attention il peut être négatif si le modèle est pire que la simple prédiction par la moyenne). Cette régression s'avère nettement meilleure que le modèle par défaut. Les variables explicatives, tout du moins certaines, sont pertinentes pour prédire les valeurs de VENTE.

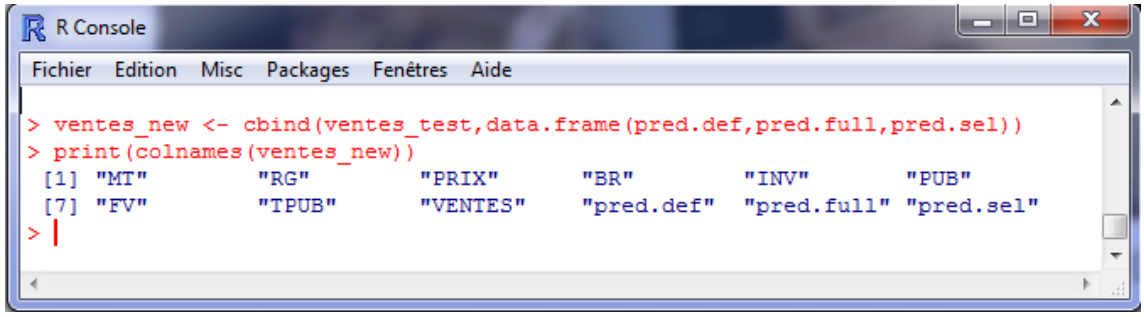


Le troisième modèle n'exploite que MT et PUB au terme d'un processus de sélection de variables cherchant à optimiser le critère Akaike (AIC). Il s'avère être le plus efficace avec une réduction de l'erreur (**SSR.SEL = 1280764**) de **RSQ.SELECTION = 62.08%** par rapport à la référence.

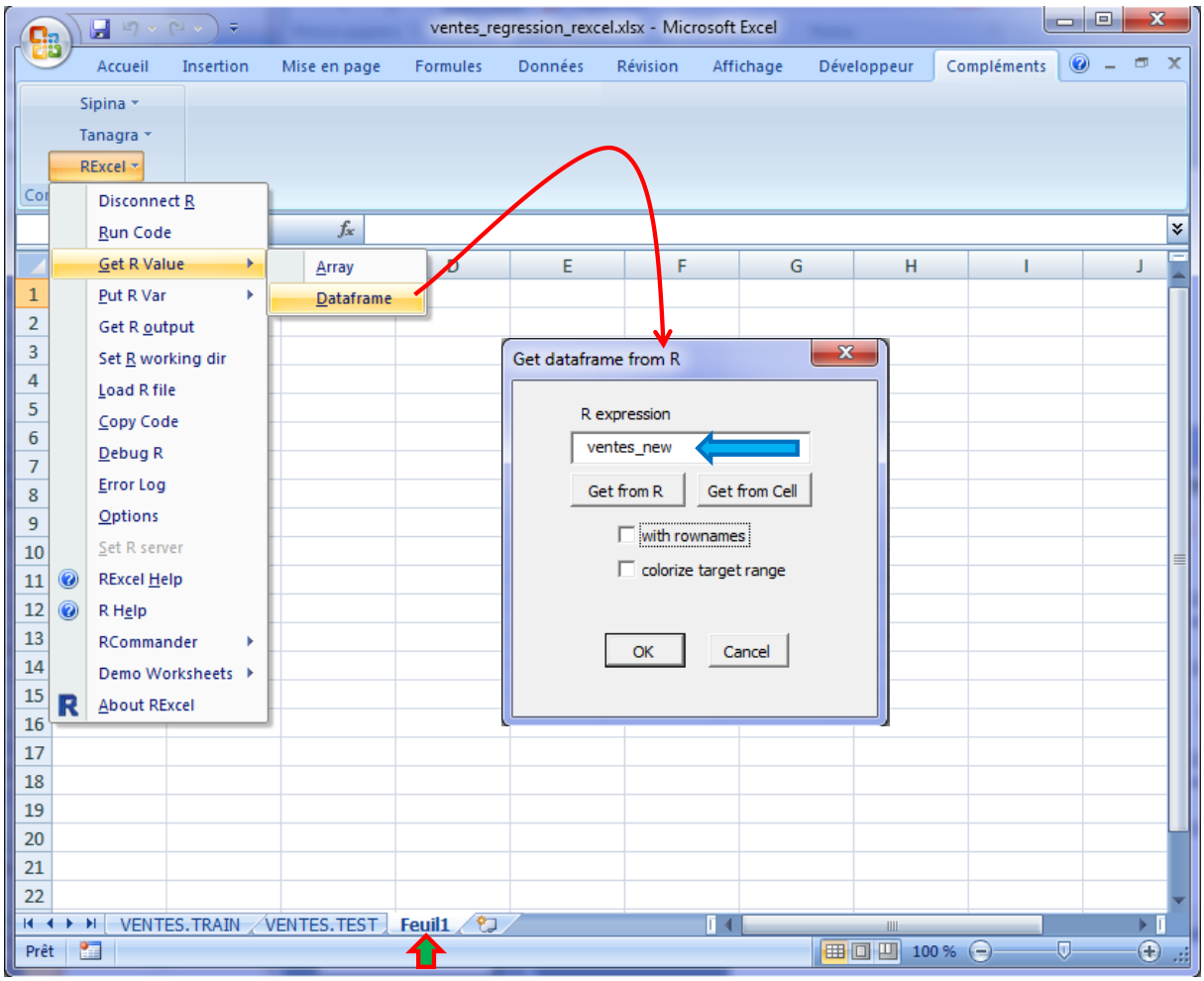


Ainsi, il est tout à fait possible de produire un modèle avec peu de variables qui soit plus efficace en généralisation que le modèle intégrant toutes les variables candidates. Les variables redondantes et non pertinentes ont un effet perturbateur. On retrouve souvent ce phénomène dans la pratique, la recherche des modèles parcimonieux n'est pas qu'une vue de l'esprit.

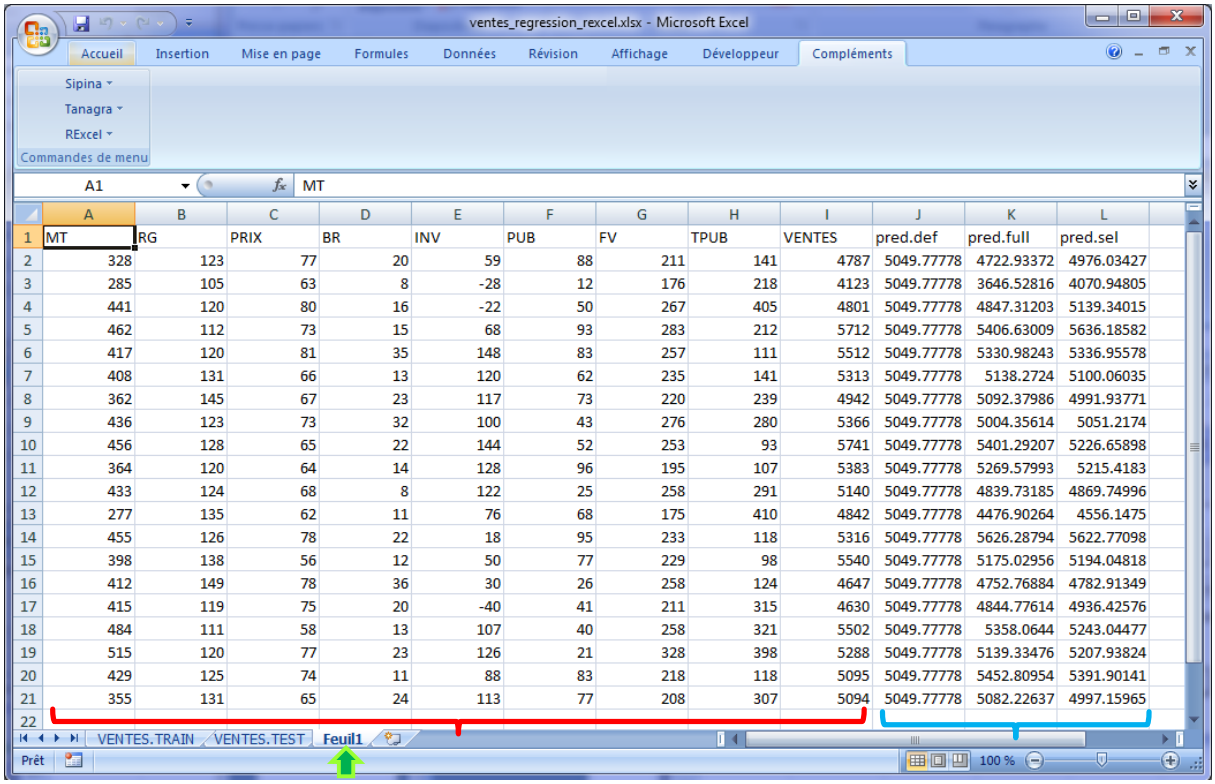
Récupération des prédictions dans Excel. Nous souhaitons adjoindre les prédictions aux colonnes de l'échantillon test et recopier le tout dans Excel. Nous créons le data frame **ventes_new** dans R,



Puis, dans Excel, après avoir ajouté une nouvelle feuille dans notre classeur, nous actionnons le menu REXCEL / GET R VALUE / DATAFRAME. Nous spécifions le nom du data frame (**ventes_new**) dans la boîte de paramétrage :



Les données composées des variables originelles de l'échantillon test et des prédictions des modèles sont copiées dans la nouvelle feuille de calcul.

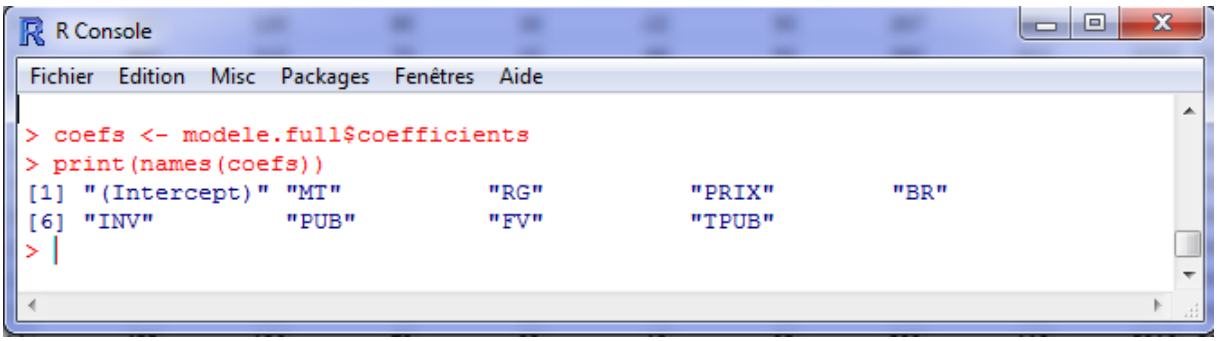


	A	B	C	D	E	F	G	H	I	J	K	L
1	MT	RG	PRIX	BR	INV	PUB	FV	TPUB	VENTES	pred.def	pred.full	pred.sel
2	328	123	77	20	59	88	211	141	4787	5049.77778	4722.93372	4976.03427
3	285	105	63	8	-28	12	176	218	4123	5049.77778	3646.52816	4070.94805
4	441	120	80	16	-22	50	267	405	4801	5049.77778	4847.31203	5139.34015
5	462	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												

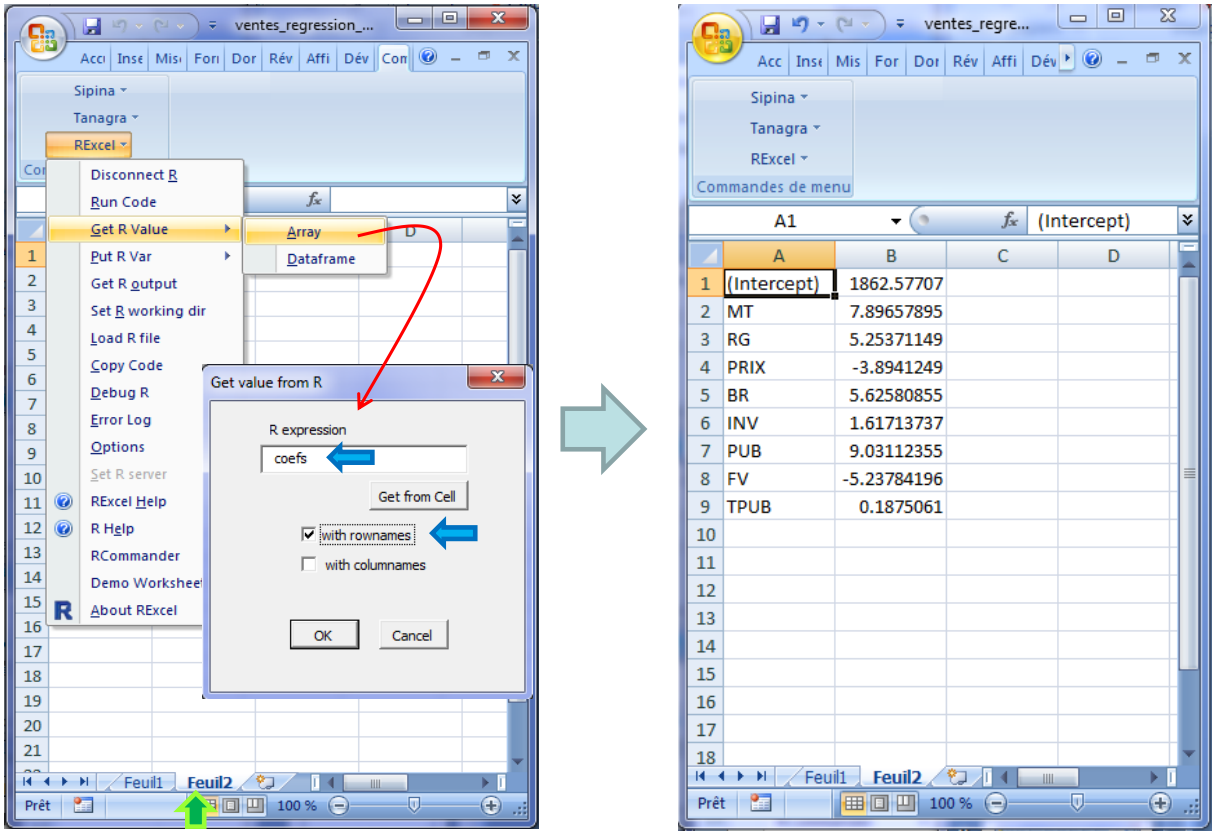
6 Récupération d'autres objets

Les échanges ne sont pas possibles pour les objets. Par exemple, nous ne pouvons pas récupérer l'objet régression pour l'afficher dans les cellules d'Excel. En revanche, nous pouvons récupérer les coefficients ou autres indicateurs tels que les t de Student ou les p-value des variables explicatives. Pour ce faire, nous devons inclure les valeurs dans des vecteurs ou matrices. Il est dès lors possible de les copier dans des cellules d'Excel.

Dans cette section, nous souhaitons récupérer les coefficients du modèle complet, incluant toutes les variables prédictives potentielles de la base, dans une nouvelle feuille de calcul (Feuil2). Nous copions tout d'abord les coefficients dans un vecteur dédié **coefs** :



Dans Excel, après avoir sélectionné une cellule vide, nous actionnons le menu REXCEL / GET R VALUE / ARRAY, nous spécifions le nom du vecteur en veillant à ce que les noms (« with rownames » dans la terminologie RExcel) soient inclus.



Nous obtenons les coefficients du modèle.

7 Accès aux fichiers EXCEL : le package XLSLX

Le principal intérêt de REXCEL est l'interactivité. Elle est précieuse dans la phase exploratoire, lorsque nous tâtonnons pour trouver la bonne manière d'analyser les données, pour produire des résultats pertinents par rapport à l'objectif de l'étude. Certaines opérations sont très faciles à mener sous R, d'autres sous Excel. Les échanges étant facilitées, nous pouvons choisir le bon outil au bon moment, en fonction des objets à manipuler et des résultats à fournir.

En revanche, si notre seul problème est d'accéder facilement aux fichiers au format EXCEL (XLS et XLSX), il existe d'autres solutions, autrement plus simples à mettre en œuvre. Dans cette section, nous utilisons le package XLSX (http://cran.r-project.org/web/packages/xlsx/). Les commandes **read.xlsx(.)** et **write.xlsx(.)** permettent respectivement de lire et écrire les fichiers au format Excel. De fait, l'étude ci-dessus peut se résumer aux instructions suivantes :

```
rm(list=ls())

#changer le répertoire courant
setwd("... votre répertoire des données ...")

#charger les données -> échantillon d'apprentissage
library(xlsx)
ventes_train <-
read.xlsx(file="ventes_regression_rexcel.xlsx",sheetName="VENTES.TRAIN")
print(summary(ventes_train))
```

```
#régression complète sur l'échantillon d'apprentissage
modele.full <- lm(VENTES ~., data = ventes_train)
print(summary(modele.full))

#modèle avec sélection de variables
library(MASS)
modele.selection <- stepAIC(modele.full,direction="backward")
print(summary(modele.selection))

#charger les données -> échantillon test
ventes_test                                                    <-
read.xlsx(file="ventes_regression_rexcel.xlsx",sheetName="VENTES.TEST")

#prédiction du modèle par défaut
pred.def <- mean(ventes_train$VENTES)
ssr.ref <- sum((ventes_test$VENTES - pred.def)^2)
print(ssr.ref)

#prédiction du modèle complet
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)

#prédiction du modèle avec sélection de 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)

#récupération des coefficients du modèle complet
coefs <- modele.full$coefficients
print(coefs)

#exportation des données test avec les prédictions
ventes_new <- cbind(ventes_test,data.frame(pred.def,pred.full,pred.sel))
#il est préférable de créer un nouveau fichier pour éviter les erreurs
#si nous avons à relancer plusieurs fois le code source
write.xlsx(ventes_new,file="ventes_output.xlsx",row.names=F)
```

Ce programme peut être élaboré à partir de l'historique des commandes de l'analyse précédente. Il est facile de les récupérer dans R (menu FICHIER / SAUVER L'HISTORIQUE DES COMMANDES). Seules les instructions relatives à la manipulation des fichiers [**read.xlsx()** et **write.xlsx()**] sont à rajouter.

8 Conclusion

REXCEL ne se limite pas au transfert des données entre R et Excel. Il inclut d'autres fonctionnalités très intéressantes. Nous pouvons par exemple appeler des fonctions de R dans une feuille de calcul avec la commande **RApply(.)**[4]**.** Nous pouvons également le faire en programmation VBA (Visual Basic pour Applications) avec **RInterface(.)**… Pour en savoir plus, une vidéo de démonstration est accessible sur le site web des auteurs de la bibliothèque http://rcom.univie.ac.at/RExcelDemo/.

Il faut certes un peu de temps pour bien intégrer les possibilités de REXCEL, mais je pense que c'est un investissement qui en vaut la peine.

[4] http://learnserver.csd.univie.ac.at/rcomwiki/doku.php?id=wiki:excel_worksheet_functions_using_r