1   Theme

Data Mining with R – The "rattle" package.

R (http://www.r-project.org/) is one of the most exciting free data mining software projects of these last years. Its popularity is completely justified (see **Kdnuggets Polls – Data Mining/ Analytic Tools Used – 2011**). Among the reasons which explain this success, we distinguish two very interesting characteristics: (1) we can extend almost indefinitely the features of the tool with the packages; (2) we have a programming language which allows to perform easily sequences of complex operations.
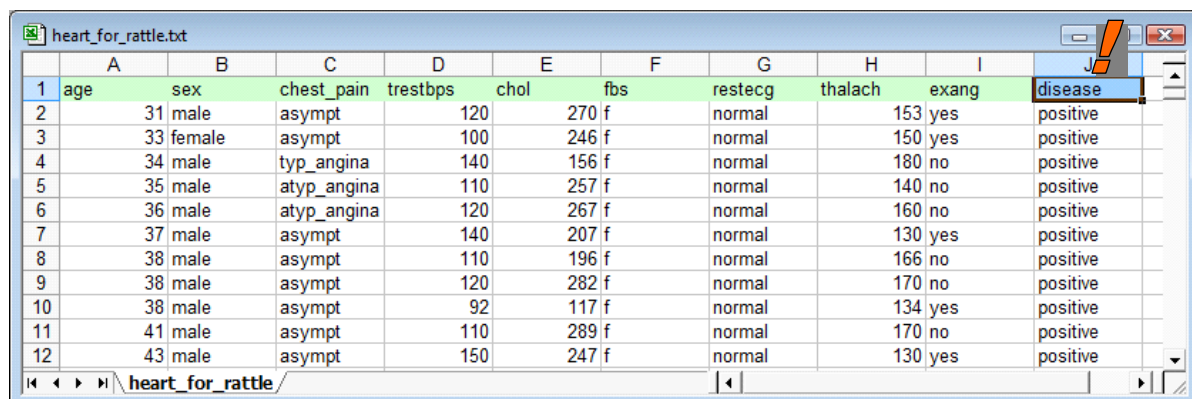
But this second property can be also a drawback. Indeed, some users do not want to learn a new programming language before being able to realize projects. For this reason, tools which allow to define the sequence of commands with diagrams (such as Tanagra, Knime, RapidMiner, etc.) still remain a valuable alternative with the data miners.

In this tutorial, we present the "Rattle" package which allows to the data miners to use R without needing to know the associated programming language. All the operations are performed with simple clicks, such as for any software driven by menus. But, in addition, all the commands are stored. We can save them in a file. Then, in a new working session, we can easily repeat all the operations. Thus, we find one of the important properties which miss to the tools driven by menus.

To describe the use of the rattle package, we perform an analysis similar to the one suggested by the rattle's author in its presentation paper (G.J. Williams, « Rattle : A Data Mining GUI for R », in *The R Journal*, volume 1 / 2, pages 45—55, December 2009, http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf). We perform the following steps: loading the data file; partitioning the instances into learning and test samples; specifying the types of the variables (target or input); computing some descriptive statistics; learning the predictive models from the learning sample; assessing the models on the test sample (confusion matrix, error rate, some curves).

2   Dataset

We use the « heart »[1] data file. We want to explain the occurrence of the DISEASE from the characteristics of patients. We show here the first instances of the dataset.

heart_for_rattle.txt

	A	B	C	D	E	F	G	H	I	J
1	age	sex	chest_pain	trestbps	chol	fbs	restecg	thalach	exang	disease
2	31	male	asympt	120	270	f	normal	153	yes	positive
3	33	female	asympt	100	246	f	normal	150	yes	positive
4	34	male	typ_angina	140	156	f	normal	180	no	positive
5	35	male	atyp_angina	110	257	f	normal	140	no	positive
6	36	male	atyp_angina	120	267	f	normal	160	no	positive
7	37	male	asympt	140	207	f	normal	130	yes	positive
8	38	male	asympt	110	196	f	normal	166	no	positive
9	38	male	asympt	120	282	f	normal	170	no	positive
10	38	male	asympt	92	117	f	normal	134	yes	positive
11	41	male	asympt	110	289	f	normal	170	no	positive
12	43	male	asympt	150	247	f	normal	130	yes	positive

heart_for_rattle

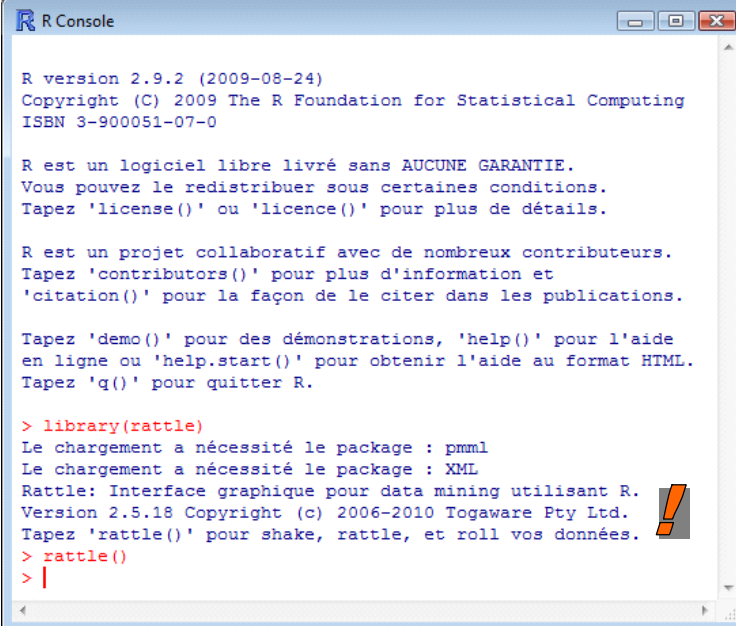[1] http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/heart_for_rattle.txt ; a description of this data file is available on the following website: http://archive.ics.uci.edu/ml/datasets/Heart+Disease

3   Data Mining with Rattle

3.1   Loading the rattle package

First, we load the rattle package [**library()**]. Then, we start the GUI with the command **rattle()**.

```
> #loading the package
> library(rattle)
> #lauching the GUI
> rattle()
```

Into the R console, we have…

From now, we perform all the operations by clicking on the appropriate menu or button. All these operations are recorded as R commands by rattle. The rattle GUI is displayed.
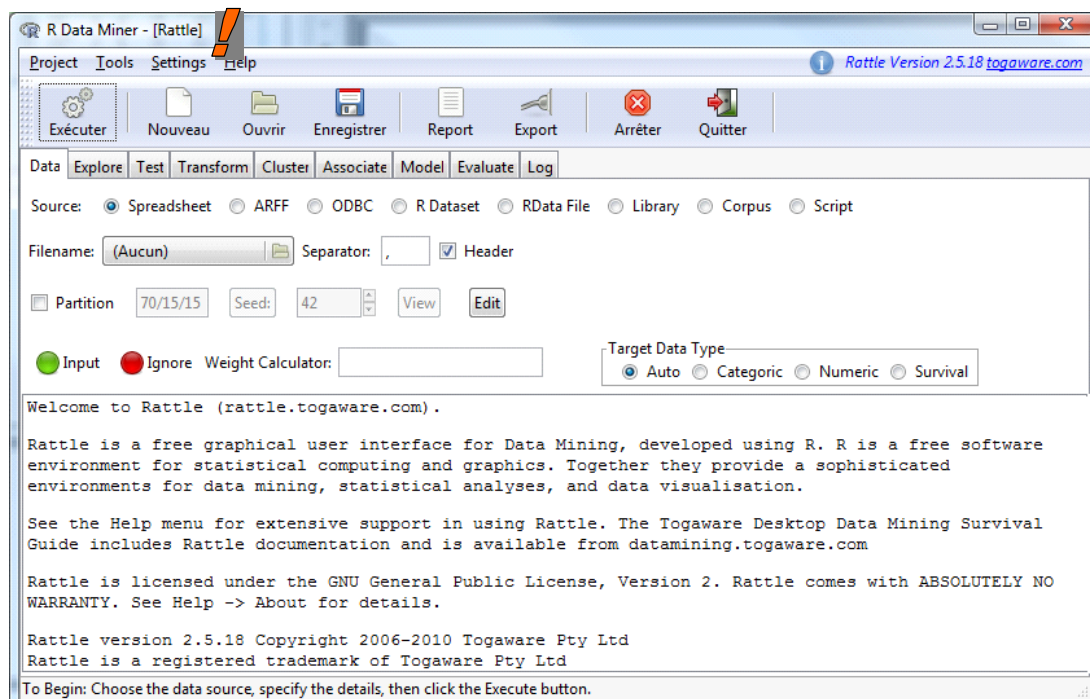
The use of rattle is always the same: we define the command by working in the appropriate tab (Data: load the dataset; Explore: some descriptive statistics; Test: some statistical tests, etc.); then, we launch the calculations by clicking on the EXECUTER button into the toolbar.

3.2   Importing the data file

Into the "Data" tab, we click on the FILENAME button. We select the" heart_for_rattle.txt" data file.
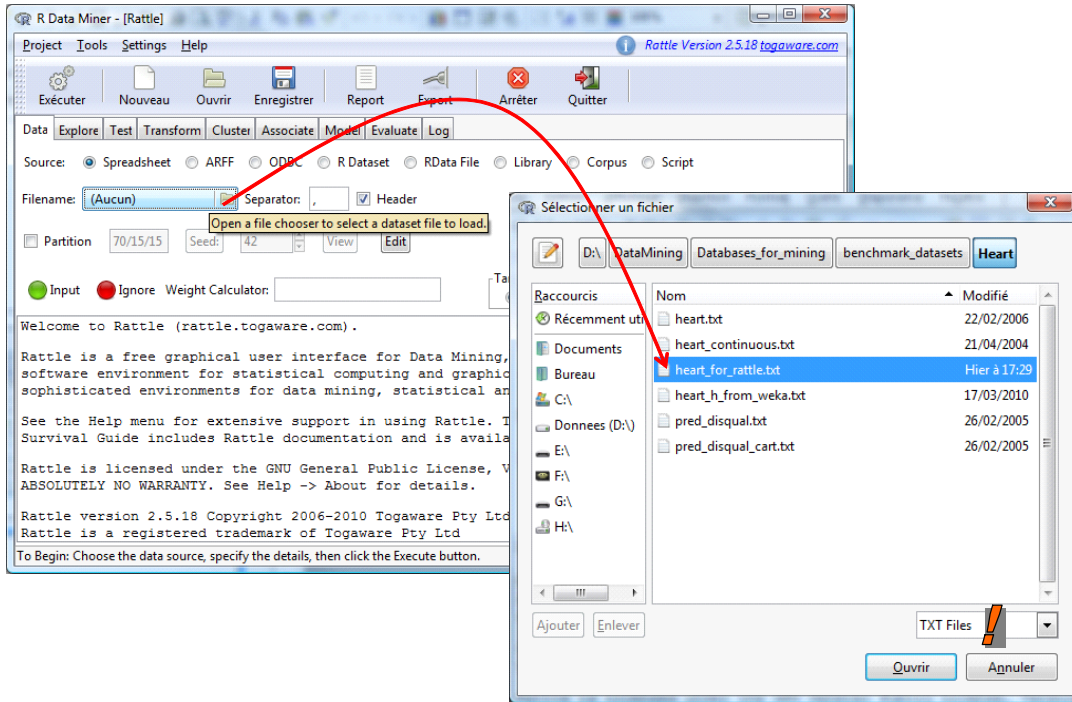
We specify the column separator: « SEPARATOR = \t ». Then we click on EXECUTER.
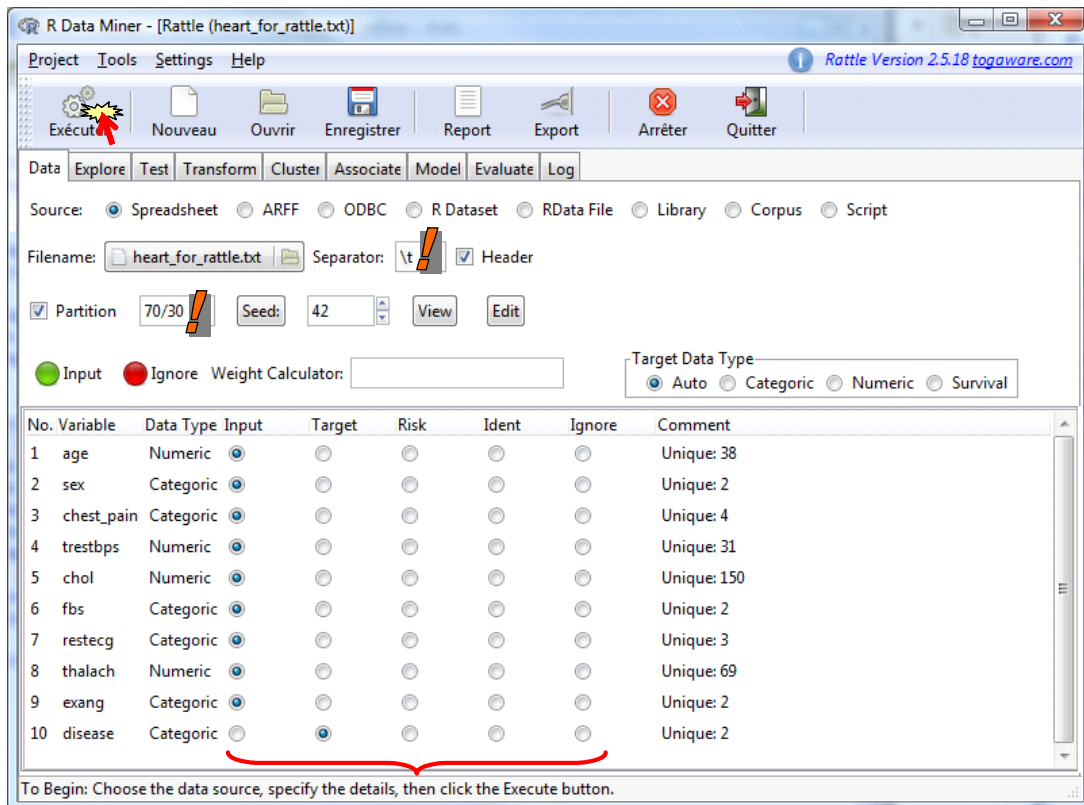
The dataset is loaded. The variable type is automatically detected from the distinct values into each column (discrete or continuous). We can define the TARGET attribute and the INPUT ones. Last, we specify the size of the training (70% of instances, drawn randomly) and test (30%) samples.

3.3   Dataset description

Into the Explore tab, we obtain some descriptive statistics indicators about the variables (SUMMARY / SUMMARY option). For the discrete variables, rattle lists the values (levels). For the continuous ones, we have the min, max, mean, quartiles. **All the indicators are computed on the learning sample**.

With the SUMMARY / DESCRIBE option, we obtain a more detailed description. Among others, for the continuous variables, the indications are useful to detect unusual values (outliers).


Into the Explore tab still, with the DISTRIBUTIONS option, we obtain some graphical representations of the distributions. We have for instance the conditional box plots of AGE and CHOL according to the values of DISEASE.
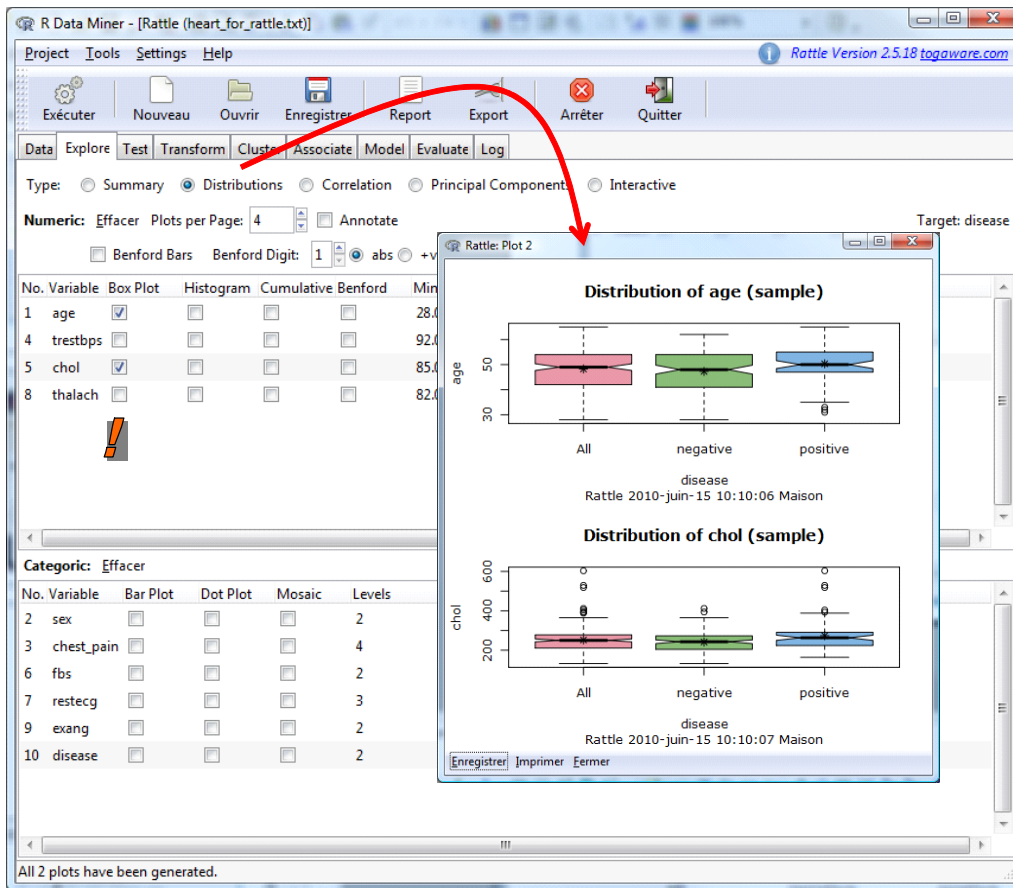
We can obtain also the conditional distribution functions.
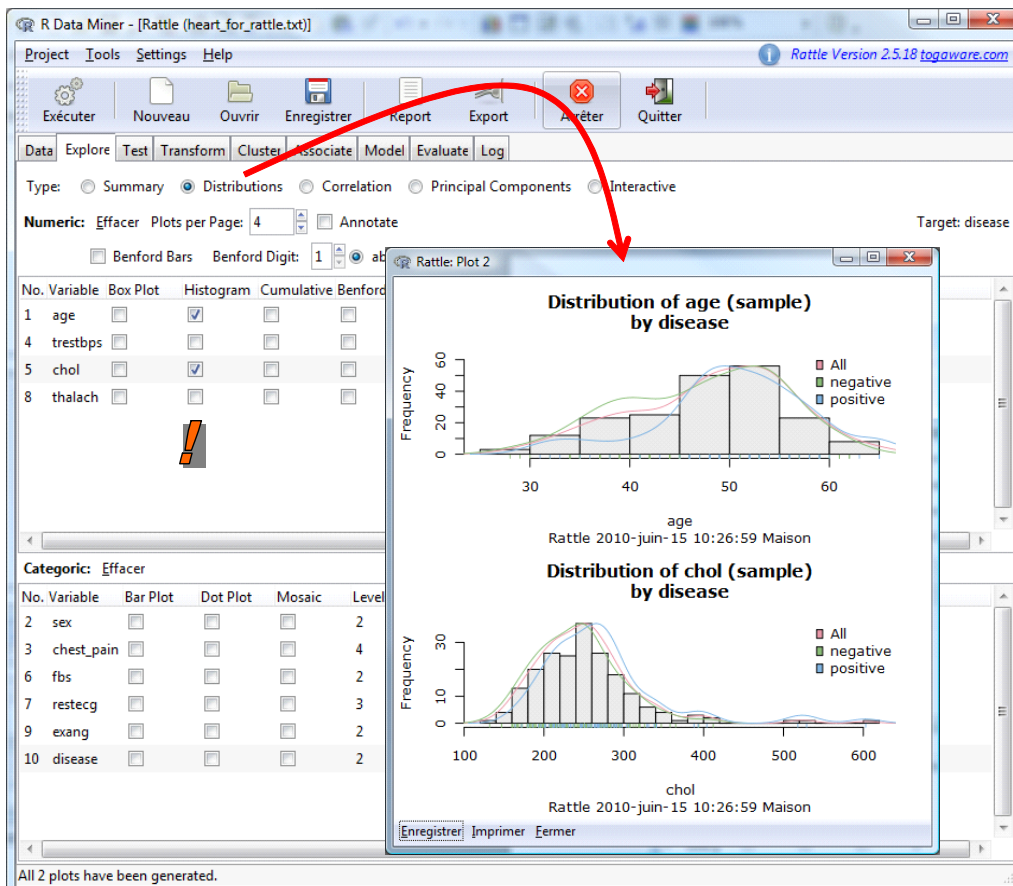
About the discrete variables, we can obtain the "Mosaic" of the variables, according still to the values of the target attribute.
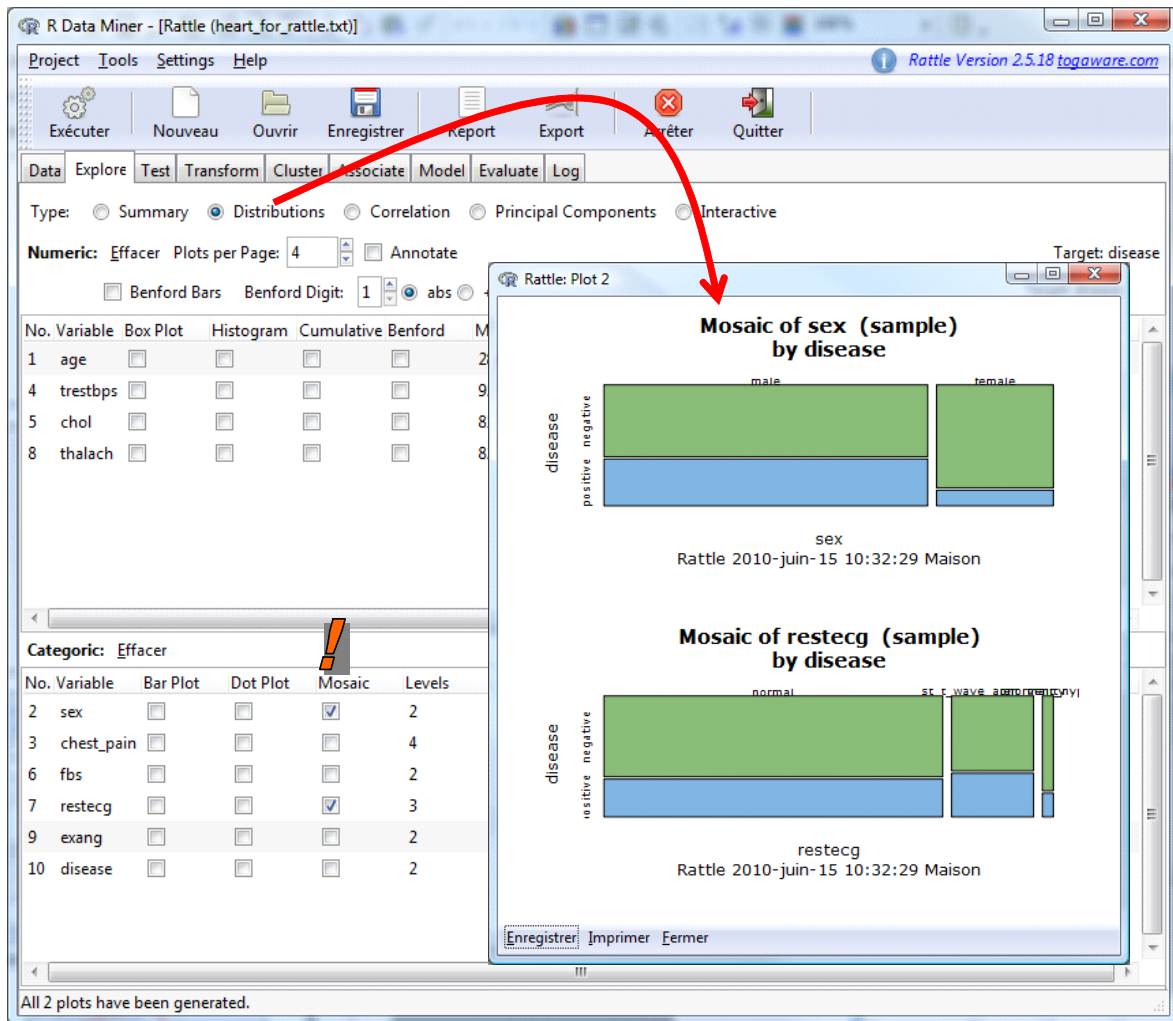

For instance, about SEX, the men (MALE) are more numerous than women (FEMALE) into the sample; and the proportion of disease is higher for the men.

We can also obtain the correlations about the continuous input attributes. The correlations are described in a hierarchical structure. It is useful for instance for the detection of the redundant variables.
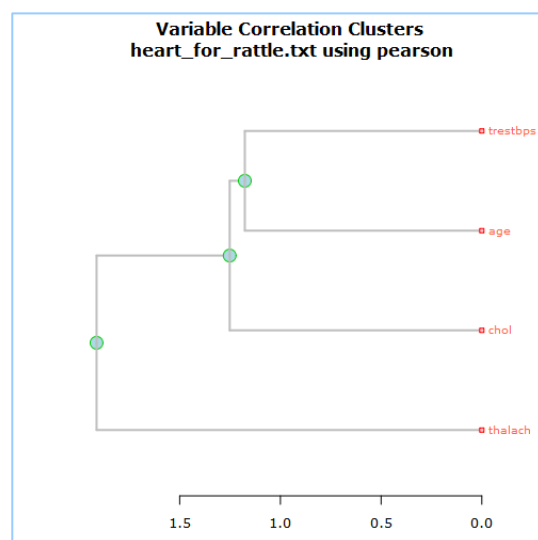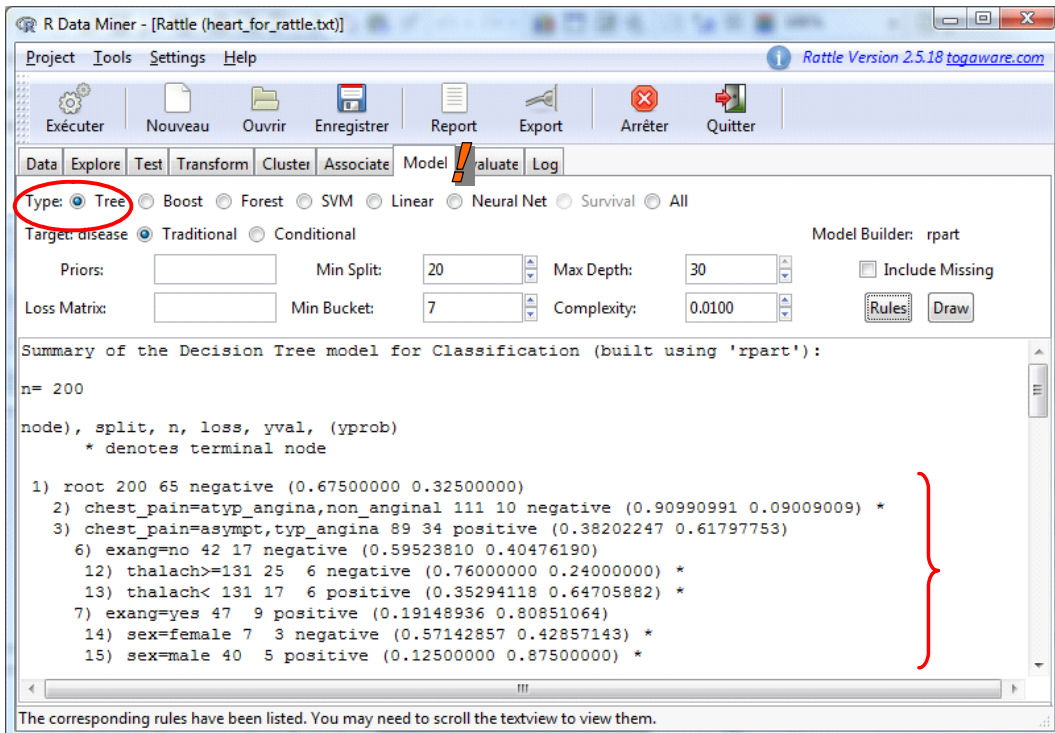
3.4   Data transformation

The "Transform" tab is dedicated to the variable transformation. Some usual operators are available (e.g. logarithm, rank, etc.).
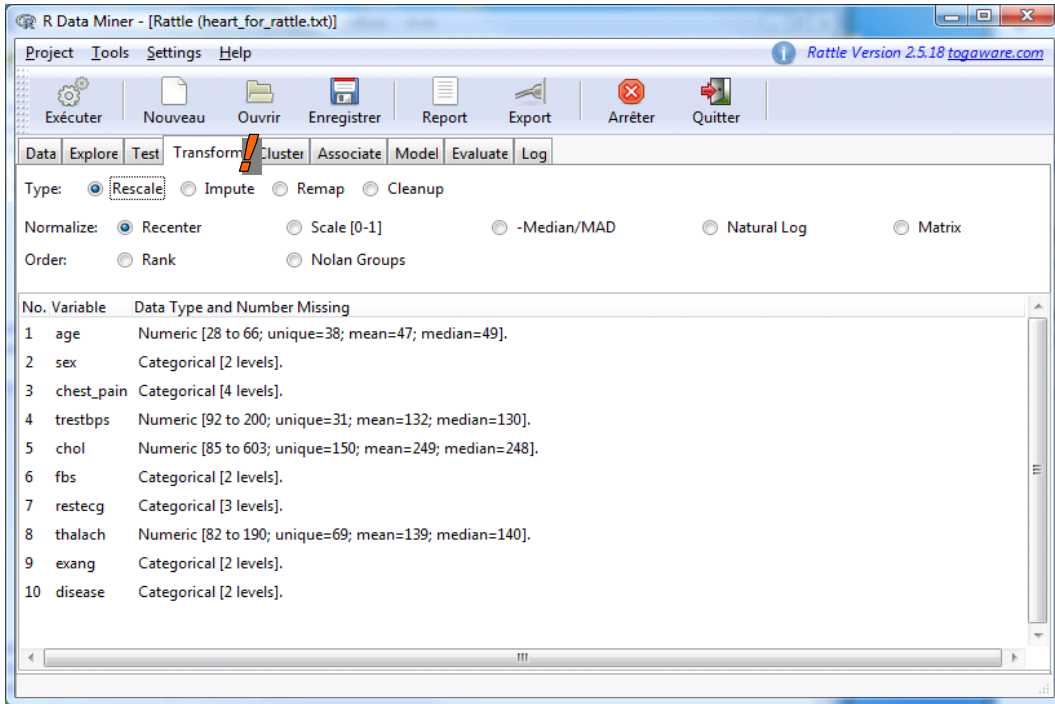
3.5   Supervised learning

This step is at the heart of our analysis. We select the "Model" tab. We want to evaluate three methods: decision tree induction, random forest, logistic regression.

About the **decision tree**, rattle uses the rpart command from the rpart package. We note the default parameters used. We click on the EXECUTER button. We obtain the rules associated to the tree by clicking on the RULES button.

```
Data | Explore | Test | Transform | Cluster | Associate | Model | Evaluate | Log

Type: ● Tree  ○ Boost  ○ Forest  ○ SVM  ○ Linear  ○ Neural Net  ○ Survival  ○ All

Target: disease ● Traditional  ○ Conditional                          Model Builder:  rpart

        Priors:  [          ]    Min Split:  [20  ]    Max Depth:  [30  ]      ☐ Include Missing

Loss Matrix:  [          ]    Min Bucket:  [7   ]    Complexity:  [0.0100]    [Rules] [Draw]
=================================================================
Tree as rules:

 Rule number: 15 [yval=positive cover=40 (20%) prob=0.88]
    chest_pain=asympt,typ_angina
    exang=yes
    sex=male

 Rule number: 13 [yval=positive cover=17 (8%) prob=0.65]
    chest_pain=asympt,typ_angina
    exang=no
    thalach< 131

 Rule number: 14 [yval=negative cover=7 (4%) prob=0.43]
    chest_pain=asympt,typ_angina
    exang=yes
    sex=female

 Rule number: 12 [yval=negative cover=25 (12%) prob=0.24]
    chest_pain=asympt,typ_angina
    exang=no
    thalach>=131

 Rule number: 2 [yval=negative cover=111 (56%) prob=0.09]
    chest_pain=atyp_angina,non_anginal

[1] 9 7 6 3 8 4 1 5 2

Generated by Rattle 2010-06-15 10:59:16 Maison
=================================================================
```

We can obtain also a graphical representation of the tree with the DRAW option.
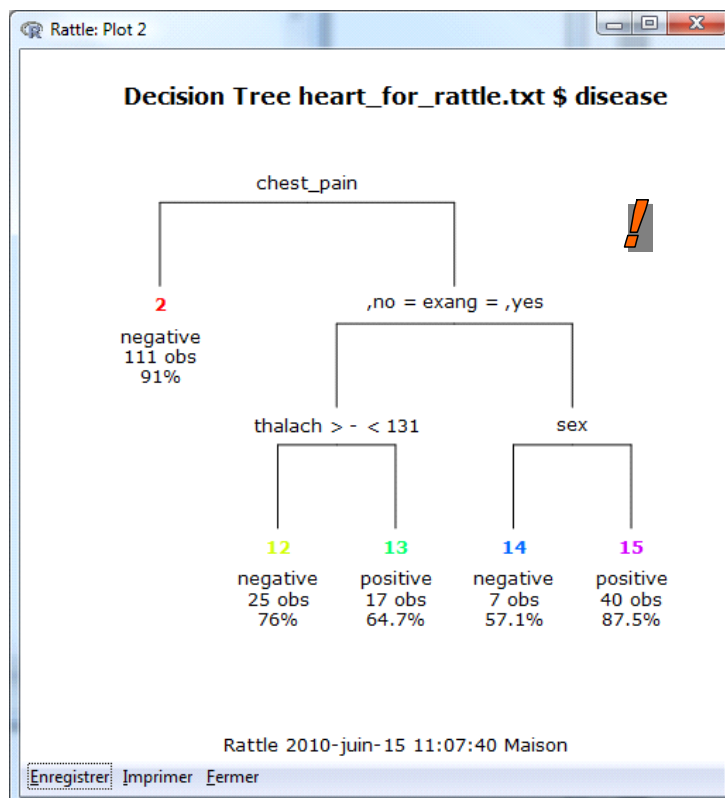
About the random forest approach, rattle uses the randomForest command from the randomForest package. We obtain the following results with the default settings.

The OOB (out-of-bag) error estimation is 16.5%. We will compare this value to the one obtained on the test set below.

About the logistic regression, we use the glm() command. It automatically transforms the discrete predictors using dummy variables. We obtain the following results.

3.6 Measuring the generalization performance

Last step of our analysis, we want to evaluate the performances of the classifiers on the test sample (30% of the whole dataset).

We activate the "Evaluate" tab. First, we want to obtain the confusion matrix and the associated error rate. We select the "Error Matrix" option. For the "Data" item, we must select the "Testing" option. Only the models learned into the "Model" tab are available here.

We click on the EXECUTER menu. We observe that the logistic regression is the better here with a test error rate equal to 18.18%.

We note also that the OOB error rate (16.5%) seems underestimate the error rate for the random forest (20.45% on the test set). But, because the test set size is small, and the test error rate being also an estimation of the "true" error rate, we consider with many cautions this result.

Actually, the error rate is not a good criterion here. We note that the differences between the methods are based only on one misclassified instance. In our context, it is perhaps more interesting to use the ROC curve which highlights the ability of the methods to assign higher score to the positive instances compared with the negative ones (see http://data-mining-tutorials.blogspot.com/2008/11/roc-curve-for-classifier-comparison.html or http://data-mining-tutorials.blogspot.com/2008/10/computing-roc-curve.html).

We select the ROC option under rattle.
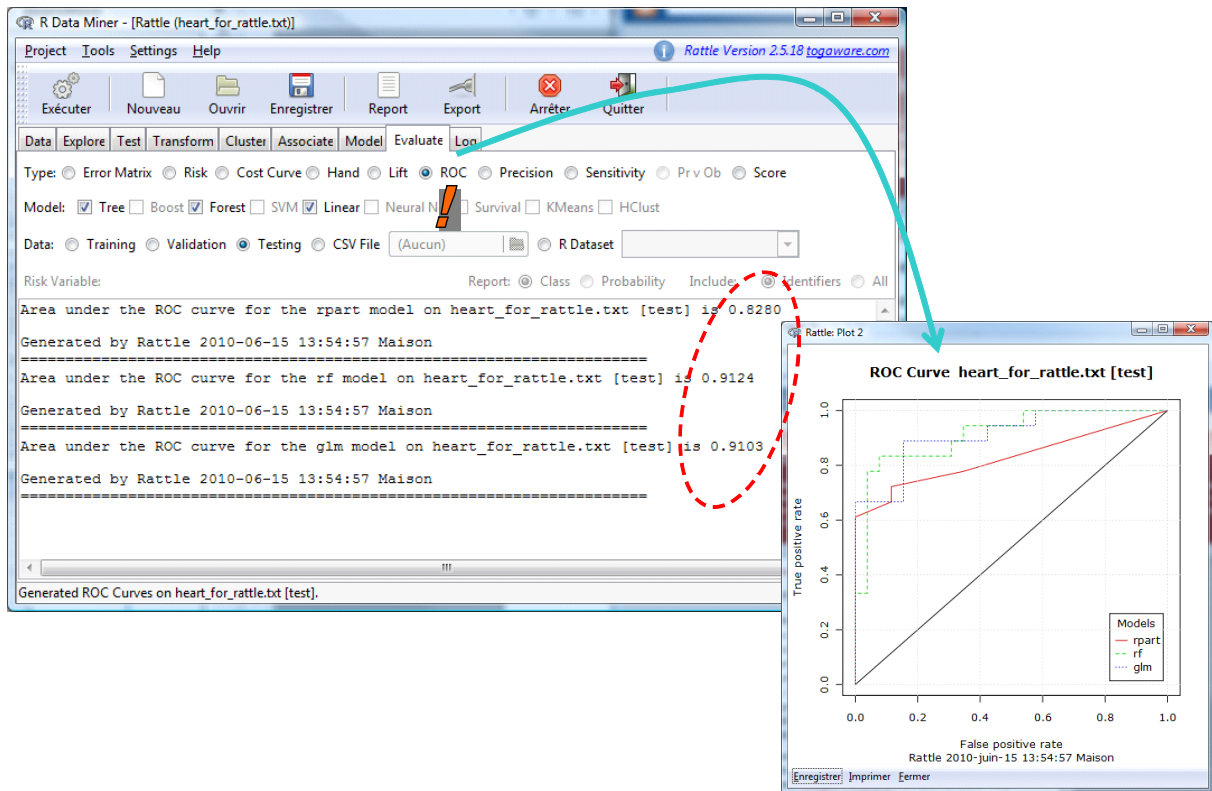
According the AUC criterion, the decision tree is definitely the worst compared with the two other classifiers, which are similar in terms of performance. It is not surprising. We know that the decision tree is not well adapted to the scoring process.

### 3.7   R commands associated to the treatments

One of the main criticisms which we make for the software driven by menu is that once the process is finalized, when we close the software, we have no recollection of the sequence of operations we performed. In the next working session, it is complicated to reproduce them as before. It is necessary to have an excellent memory, or to have taken care of noting all that we made.

Rattle allows to overtake this drawback by translating all the operations (corresponding to a click on the EXECUTER menu) performed by the user in a sequence of R commands. We can visualize them in the "Log" tab. We can store these commands (and the comments) into a file. In the next working session, it is very easy to perform the same data processing by loading these commands.

4   Rattle under Linux (Ubuntu)

The installation of the Rattle package under Linux is not easy. We must follow carefully the description available on the website. In case of problem, a troubleshooting procedure is proposed. This is the one that I used (see http://datamining.togaware.com/survivor/Install_GNU_Linux.html).

When the installation is finalized, Rattle works properly under Linux (Ubuntu) as we see below.
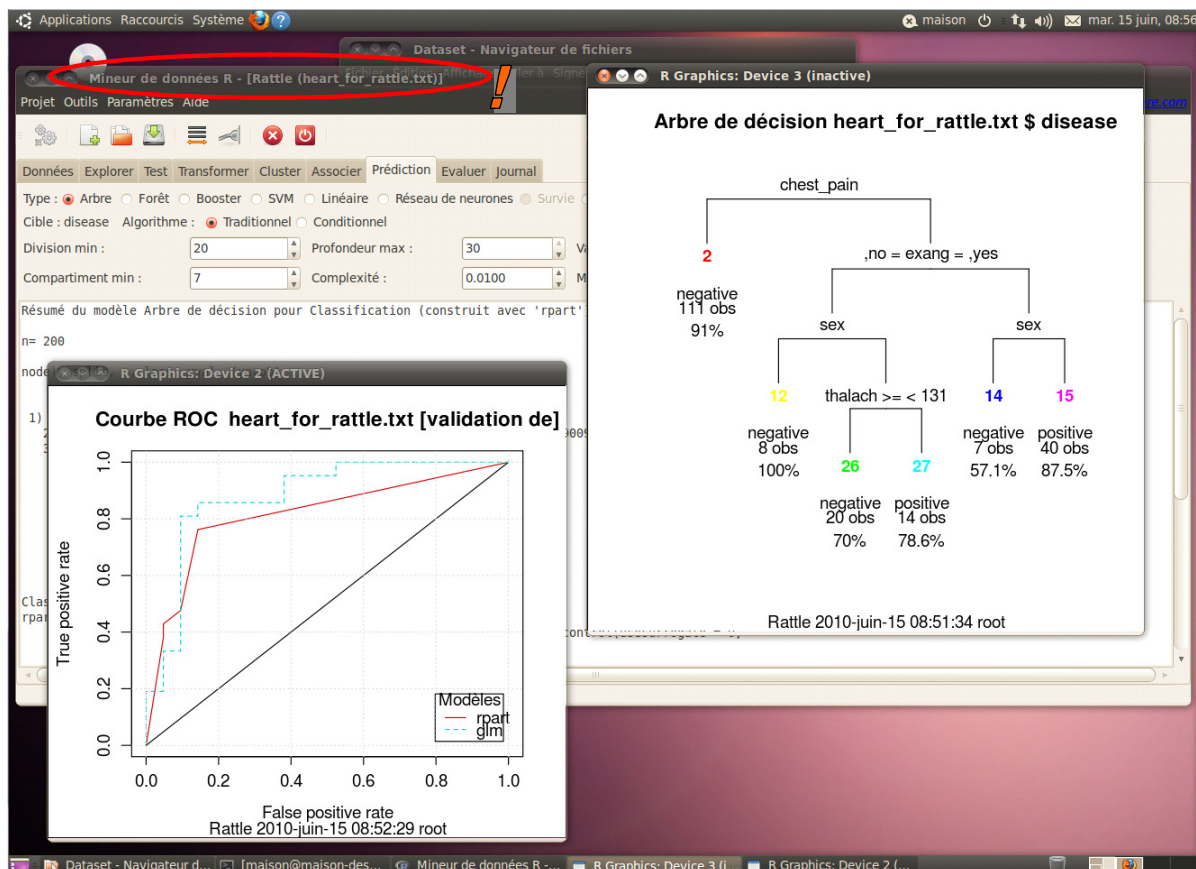
5   Conclusion

In this tutorial, we showed that it was possible to use R without knowledge about its programming language with the help of the rattle package. This package is rather specialized about the data mining methods. For the statisticians, there are other packages such as "R Commander".