

# 1 Objectif

## Data Mining sous R – Le package « rattle ».

Le père de Tanagra est un fan de R. Cela peut paraître étrange et/ou contradictoire. Mais en réalité, je suis surtout un grand fan de Data Mining. Et le logiciel en est un maillon essentiel. Je passe ainsi beaucoup de temps à les disséquer, à évaluer leur comportement face aux données, et analyser leur code source lorsque cela est possible, bref, à les étudier sous toutes les coutures. Ce travail me passionne tout simplement. Je l'ai toujours fait. Avec Internet, je peux partager le fruit de mes réflexions avec d'autres utilisateurs.

Pour en revenir à R, force est de reconnaître qu'il cumule les qualités. Le système des packages permet de l'enrichir considérablement. Ils couvrent très largement les méthodes de traitement de données, quelles que soient leurs origines (statistique, analyse de données, data mining, etc.). Les librairies les plus populaires font référence. Certes, certaines sont de qualité parfois inégale. Il nous appartient dans ce cas de vérifier attentivement les calculs avant de communiquer sur les résultats. Mais, d'une part, le principe de l'accès au code source doit nous rassurer, on sait exactement ce qui a été programmé ; d'autre part, la description des librairies dans des revues telles que le « [Journal of Statistical Software](#) » est assurément un gage de sérieux.

L'autre atout de R est le langage de programmation qui lui est associé. Il s'agit d'un véritable langage avec tous les éléments qui permettent de mettre en œuvre des traitements sophistiqués : branchements conditionnels, actions répétitives, programmation modulaire, etc. Mais cet avantage est aussi un inconvénient. Il impose aux utilisateurs l'apprentissage d'un langage de programmation, chose qui peut leur paraître insurmontable, surtout lorsqu'ils sont réfractaires à l'informatique.

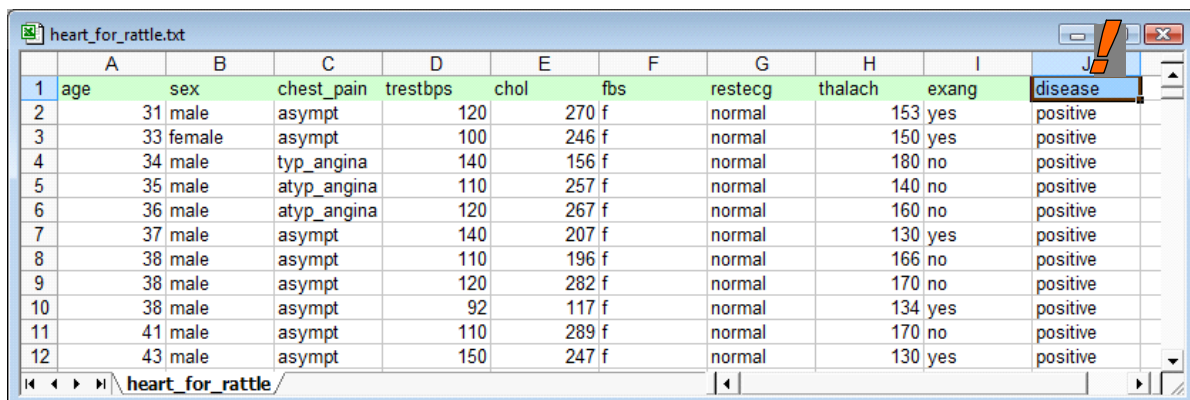
Dans ce tutoriel, nous présentons le package **rattle** (<http://rattle.togaware.com/>; <http://cran.r-project.org/web/packages/rattle/index.html>) spécialisé dans le Data Mining. Il cherche à dépasser l'inconvénient ci-dessus. En effet, il n'intègre pas de nouvelles méthodes d'apprentissage, il vise plutôt à rajouter une interface utilisateur graphique (GUI en anglais, « graphical user interface ») à R. Ainsi, un praticien, ignorant tout du langage de programmation R, pourra néanmoins piloter ses analyses en cliquant simplement sur des menus ou des boutons, un peu à l'image du mode « Explorer » du logiciel [Weka](#). Rien de bien révolutionnaire donc, mais ô combien important pour les utilisateurs novices qui veulent aller à l'essentiel : traiter leurs données à l'aide de R sans avoir à investir dans l'apprentissage fastidieux de la programmation.

**Rattle** intègre une autre fonctionnalité très intéressante. A l'instar de l'enregistreur de macros des outils Office qui traduit les actions de l'utilisateur en code VBA, les manipulations peuvent être traduites en code programme R. Nous pouvons donc, à tout moment, stocker dans un fichier le code source associé à l'analyse et, à la prochaine session de travail, reproduire exactement les mêmes opérations en faisant exécuter les commandes sauvegardées. Nous pouvons, si nous le souhaitons, compléter le code source pour rendre les traitements plus performants. Rattle répond ainsi à l'un des principaux reproches que l'on fait aux logiciels pilotés par menu : l'impossibilité de reproduire à l'identique une série de manipulations réalisées par l'utilisateur durant une session de travail.

Pour décrire le fonctionnement de rattle, nous reprenons la trame du document de présentation publié par son auteur dans le journal de R (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](http://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf)). Nous réaliserons la succession d'opérations suivantes : charger le fichier, le scinder en échantillons d'apprentissage et de test, définir le rôle des variables (cible vs. prédictives), réaliser quelques statistiques descriptives et graphiques pour appréhender les données, construire les modèles prédictifs sur l'échantillon d'apprentissage, les jauger sur l'échantillon test à travers les outils usuels d'évaluation (matrice de confusion, quelques courbes).

## 2 Données

Nous utilisons les données « heart »<sup>1</sup> pour illustrer notre propos. Le fichier est au format texte avec séparateur tabulation. Nous cherchons à prédire la présence ou l'absence d'une maladie cardiaque (DISEASE). Nous disposons de 286 observations. Nous affichons les premières lignes du fichier.



	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

## 3 Data Mining avec Rattle

### 3.1 Démarrer le package rattle

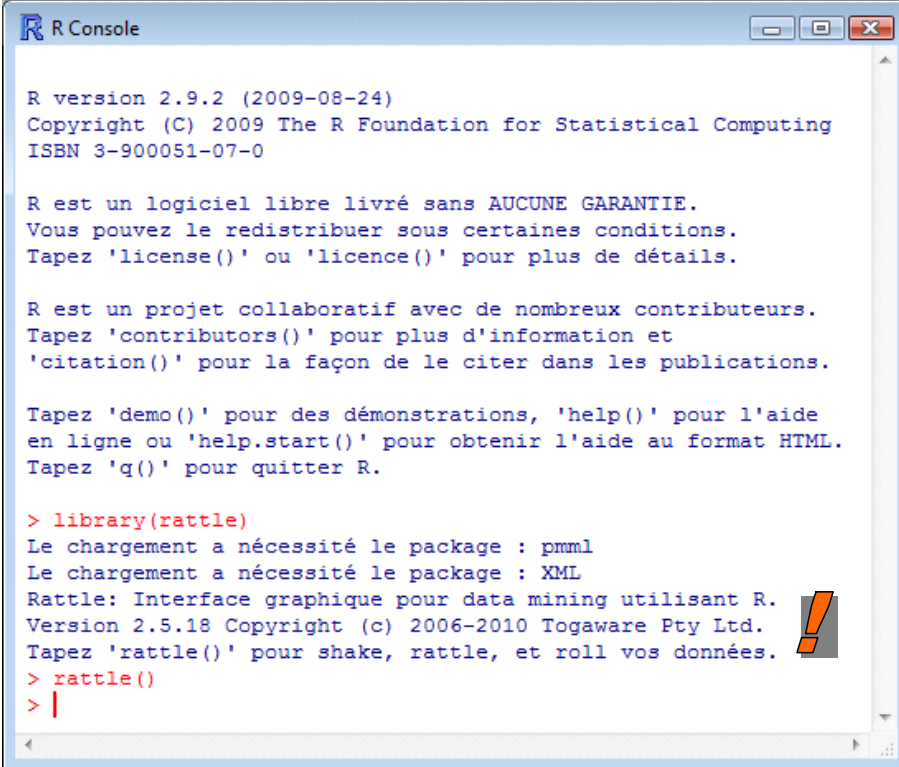
La première étape consiste à installer la librairie. Si le principe de l'installation d'un package n'est guère compliqué (<http://tutoriels-data-mining.blogspot.com/2009/05/installation-des-packages-sous-r.html>), il faut être attentif pour rattle car il intègre de multiples dépendances.

Par la suite, nous introduisons les commandes suivantes pour lancer l'interface graphique.

```
> #charger le package
> library(rattle)
> #lancer l'application
> rattle()
```

Dans la console R, nous avons...

<sup>1</sup> [http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/heart\\_for\\_rattle.txt](http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/heart_for_rattle.txt) ; une description du jeu de données est accessible sur le serveur UCI : <http://archive.ics.uci.edu/ml/datasets/Heart+Disease>



```

R Console

R version 2.9.2 (2009-08-24)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

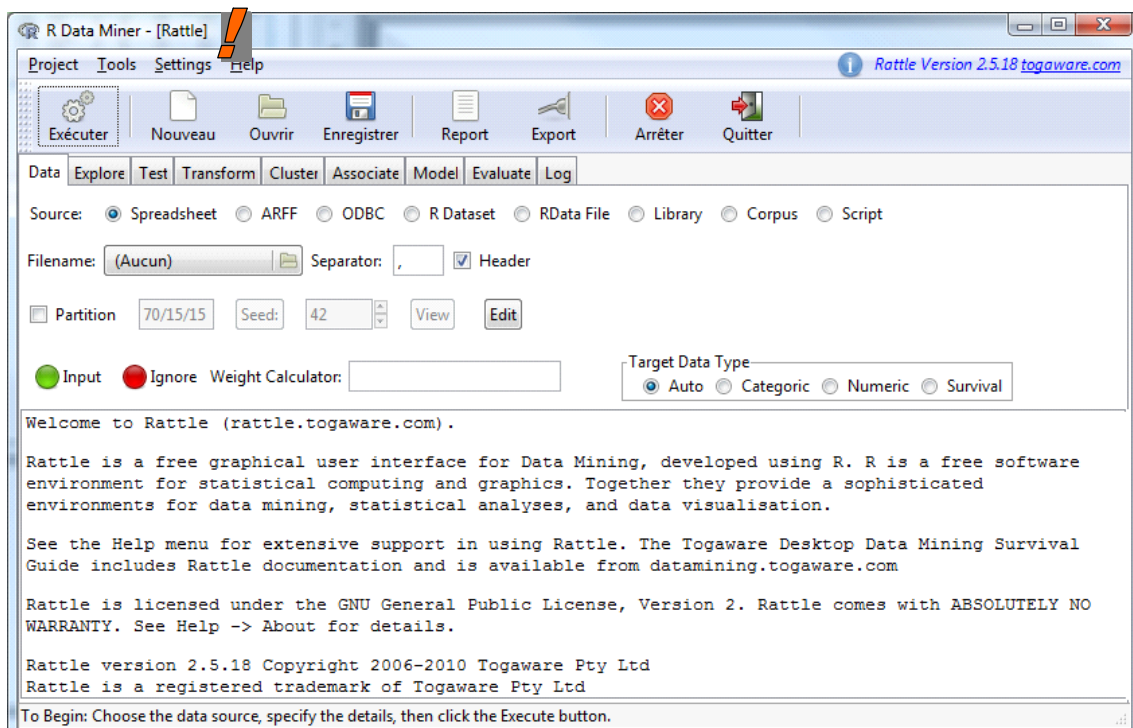
R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

> library(rattle)
Le chargement a nécessité le package : pmml
Le chargement a nécessité le package : XML
Rattle: Interface graphique pour data mining utilisant R.
Version 2.5.18 Copyright (c) 2006-2010 Togaware Pty Ltd.
Tapez 'rattle()' pour shake, rattle, et roll vos données.
> rattle()
> |

```

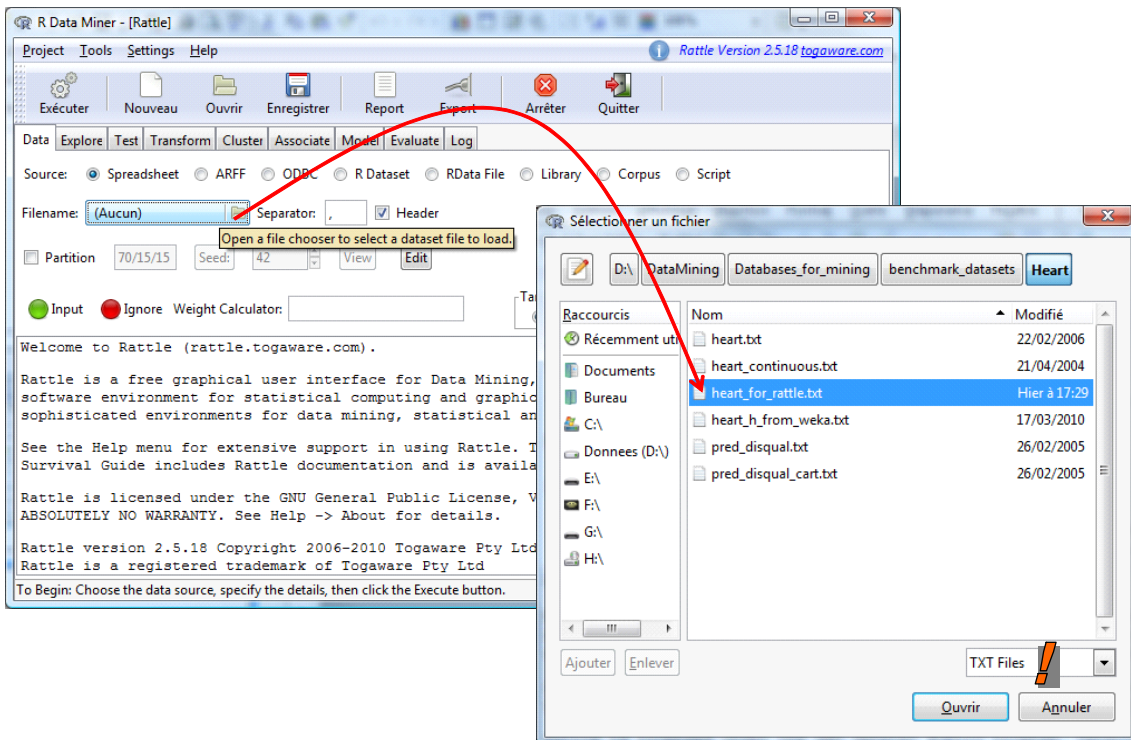
Ce sera la dernière fois où nous aurons à introduire manuellement des commandes. Tout le reste de l'analyse sera pilotée par menu. L'interface de rattle apparaît dans une nouvelle fenêtre.



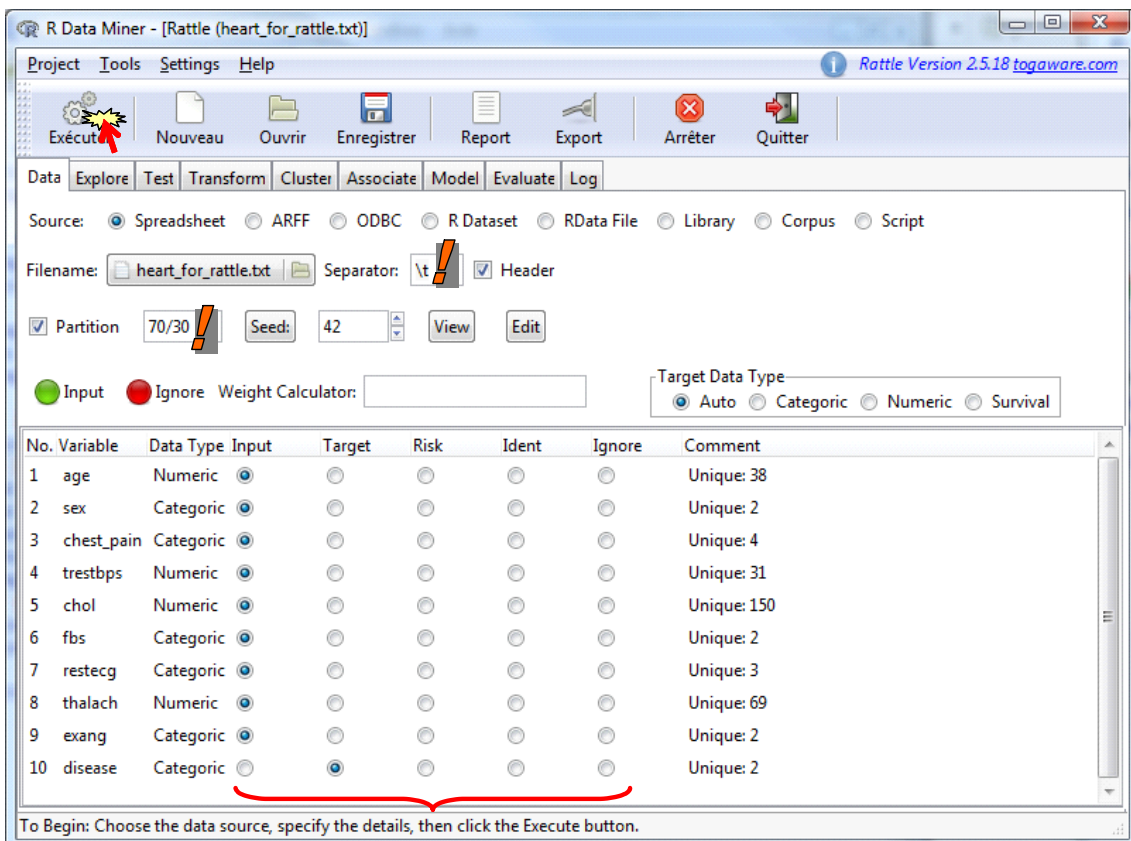
Le mode de fonctionnement du logiciel est immuable : nous définissons les opérations dans la partie basse de l'interface, en travaillant dans l'onglet adéquat selon le type d'analyse que l'on veut mener (ex. Explore : statistiques descriptives ; Test : tests statistiques ; etc.) ; puis nous lançons le calcul en cliquant sur le bouton EXECUTER dans la barre d'outils.

### 3.2 Importation des données

Dans l'onglet « Data », nous cliquons sur le bouton FILENAME pour spécifier le fichier de données. Nous sélectionnons « heart\_for\_rattle.txt ».



Nous spécifions le séparateur : « SEPARATOR = \t ». Puis nous cliquons sur le bouton EXECUTER.



Les données sont chargées, les variables sont typées (numérique ou catégorielle) automatiquement. Rattle dénombre le nombre de valeurs distinctes pour chaque variable (« Comment »). Cela devrait nous aider à identifier leur type réel. Nous spécifions leur statut dans l'analyse. Nous distinguons principalement les variables prédictives (INPUT) et la variable cible (TARGET).

Enfin, nous partitionnons les données en échantillons d'apprentissage (70%) et de test (30%) avec l'option PARTITION. Notons qu'il est possible de subdiviser les données en 3 parties : apprentissage, validation et test. La seconde sert souvent d'échantillon de réglage, il est utilisé pour régler (par tâtonnement) les paramètres des méthodes de data mining.

### 3.3 Description des données

R Data Miner - [Rattle (heart\_for\_rattle.txt)]

Project Tools Settings Help Rattle Version 2.5.18 togaware.com

Exécuter Nouveau Ouvrir Enregistrer Report Export Arrêter Quitter

Data Explore Transform Cluster Associate Model Evaluate Log

Type:  Summary  Distributions  Correlation  Principal Components  Interactive

Summary  Describe  Basics  Kurtosis  Skewness  Show Missing

Below is a summary of the dataset.  
The data is limited to the training dataset.  
Data frame: crs\$dataset[crs\$sample, ] 200 observations and 10 variables Maximum # NAs:0

Variable	Levels	Storage
age		integer
sex	2	integer
chest_pain	4	integer
trestbps		integer
chol		integer
fbs	2	integer
restecg	3	integer
thalach		integer
exang	2	integer
disease	2	integer

```

-----+-----+-----+
|Variable |Levels |
+-----+-----+
|sex      |female,male |
+-----+-----+
|chest_pain|asympt,atyp_angina,non_anginal,typ_angina |
+-----+-----+
|fbs      |f,t |
+-----+-----+
|restecg  |left_vent_hyper,normal,st_t_wave_abnormality|
+-----+-----+
|exang    |no,yes |
+-----+-----+
|disease  |negative,positive |
+-----+-----+

```

For the simple distribution tables below the 1st and 3rd Qu.  
refer to the first and third quartiles, indicating that 25% of the observations have values of  
or greater than (respectively) the value listed.

age	sex	chest_pain	trestbps	chol
Min. :28.00	female: 53	asympt :81	Min. : 98.0	Min. :132.0
1st Qu.:42.00	male :147	atyp_angina:77	1st Qu.:120.0	1st Qu.:211.0
Median :49.00		non_anginal:34	Median :130.0	Median :250.0
Mean :48.27		typ_angina : 8	Mean :133.7	Mean :252.0
3rd Qu.:54.00			3rd Qu.:140.0	3rd Qu.:277.5
Max. :65.00			Max. :190.0	Max. :603.0

fbs	restecg	thalach	exang	disease
f:184	left_vent_hyper : 5	Min. : 82.0	no :142	negative:135
t: 16	normal :157	1st Qu.:122.0	yes: 58	positive: 65
	st_t_wave_abnormality: 38	Median :140.0		
		Mean :139.6		
		3rd Qu.:155.2		
		Max. :190.0		

Generated by Rattle 2010-06-15 09:50:20 Maison

Find:  Rechercher Suivant

Data summary generated.

L'onglet « Explore » est dédié à la description des données. L'option SUMMARY / SUMMARY fournit les statistiques descriptives. Nous obtenons l'énumération des valeurs des variables catégorielles, ainsi que leur distribution de fréquences. Pour les variables quantitatives, nous avons les quartiles et la moyenne. Tous les indicateurs sont calculés sur l'échantillon d'apprentissage.

Avec l'option SUMMARY / DESCRIBE, la description est plus détaillée. Nous obtenons le nombre de valeurs distinctes, les déciles ainsi que les 5 plus grandes et les 5 plus petites valeurs pour les variables quantitatives. Ces informations sont précieuses pour détecter rapidement la présence de valeurs aberrantes dans notre échantillon.

R Data Miner - [Rattle (heart\_for\_rattle.txt)]

Project Tools Settings Help

Rattle Version 2.5.18 togaware.com

Exécuter Nouveau Ouvrir Enregistrer Report Export Arrêter Quitter

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type:  Summary  Distributions  Correlation  Principal Components  Interactive

Summary  Describe  Basics  Kurtosis  Skewness  Show Missing

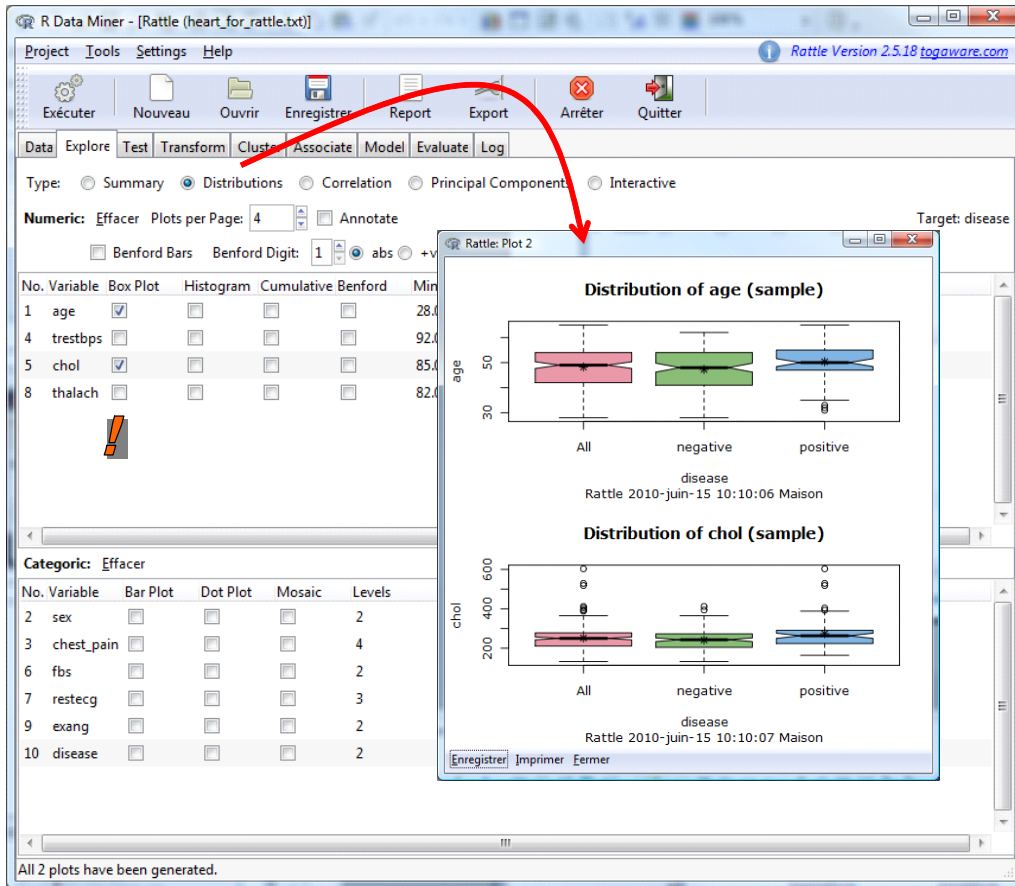
Below is a description of the dataset.  
The data is limited to the training dataset.

```
crs$dataset[crs$sample, ]
10 Variables      200 Observations
-----
age
  n missing unique   Mean   .05   .10   .25   .50   .75   .90   .95
  200     0     37 48.27 34.00 37.00 42.00 49.00 54.00 58.00 59.05
lowest : 28 29 30 31 32, highest: 60 61 62 63 65
-----
sex
  n missing unique
  200     0     2
female (53, 26%), male (147, 74%)
-----
chest_pain
  n missing unique
  200     0     4
asympt (81, 40%), atyp_angina (77, 38%), non_anginal (34, 17%), typ_angina (8, 4%)
-----
trestbps
  n missing unique   Mean   .05   .10   .25   .50   .75   .90   .95
  200     0     25 133.7 110.0 110.0 120.0 130.0 140.0 160.0 160.5
lowest : 98 100 105 106 108, highest: 150 160 170 180 190
-----
```

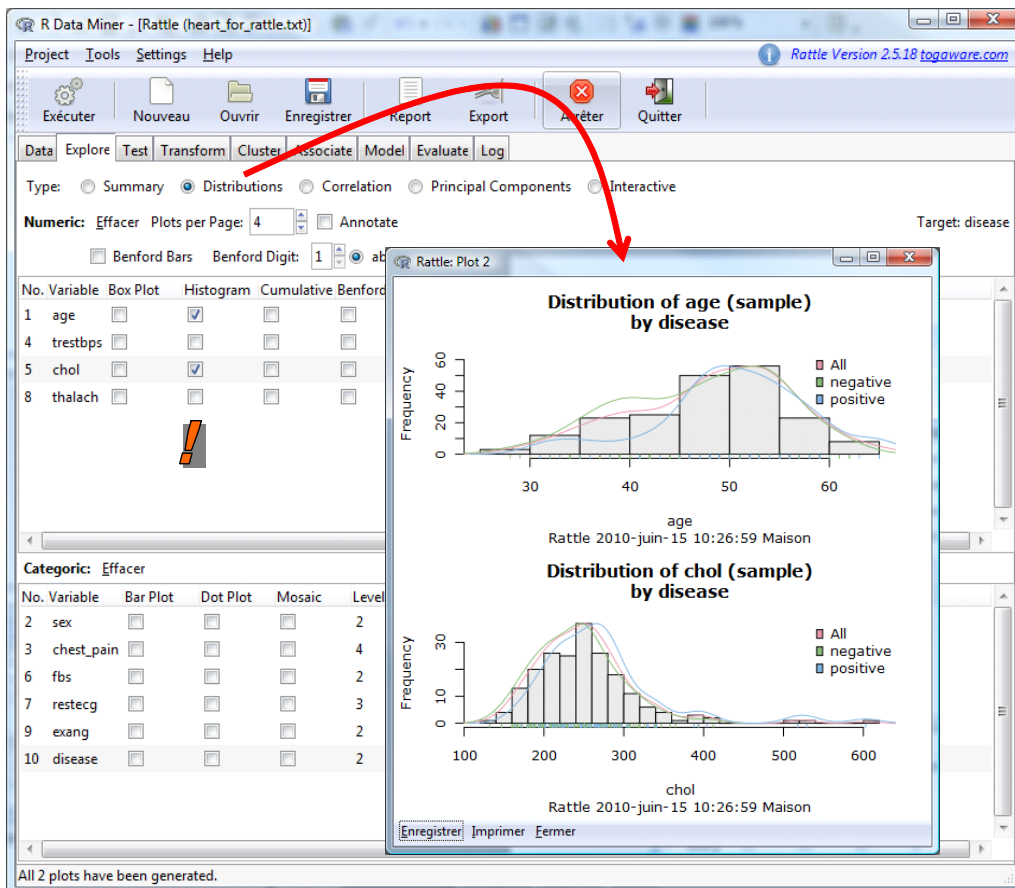
Find:  Rechercher Suivant

Data summary generated.

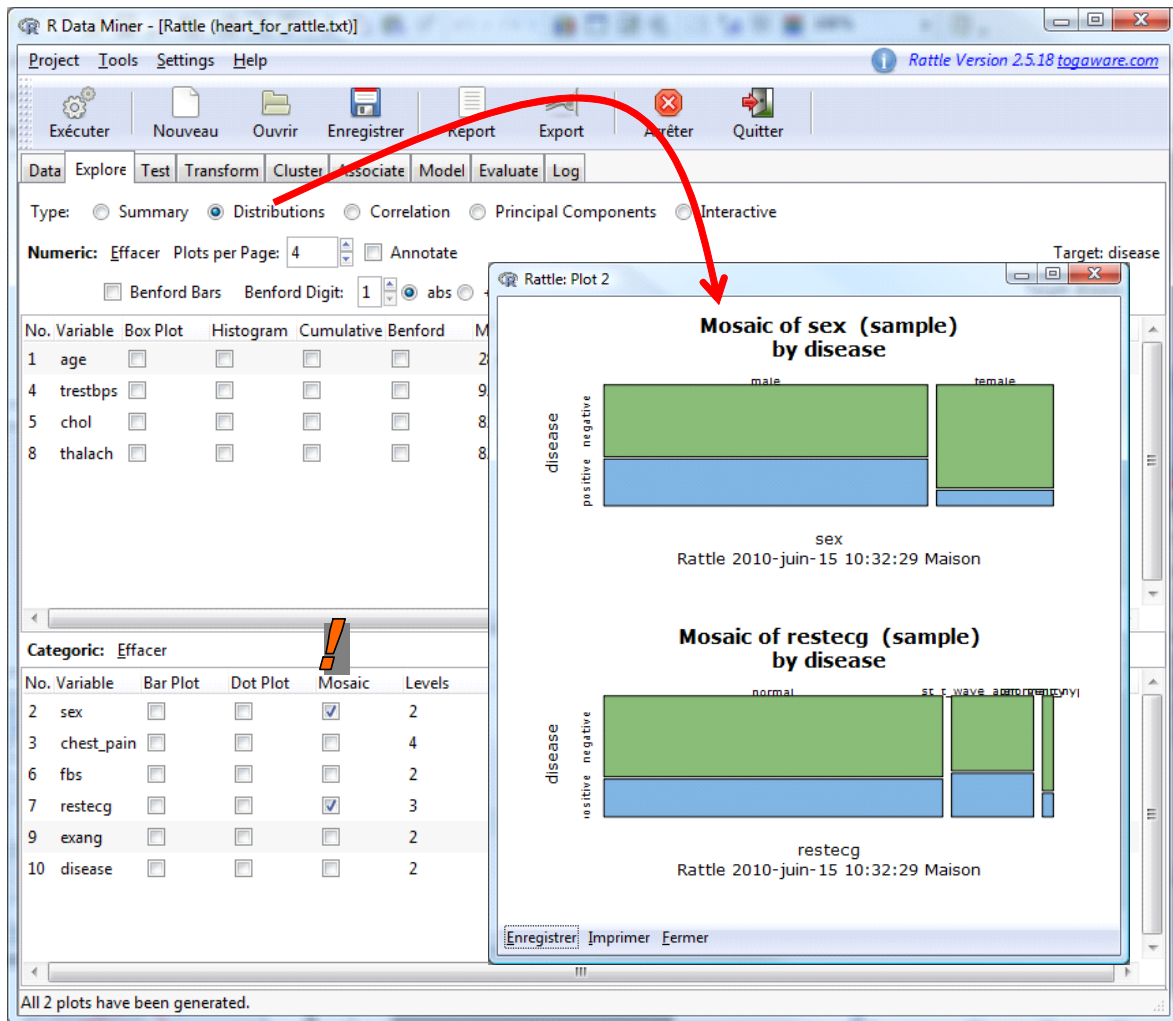
Toujours dans l'onglet « Explore », lorsque nous passons à l'outil DISTRIBUTIONS, nous avons accès aux outils graphiques. Nous souhaitons par exemple obtenir les boîtes à moustaches des variables AGE et CHOL globalement et conditionnellement aux valeurs de la variable à prédire DISEASE.



Nous pouvons aussi comparer les fonctions de densité.

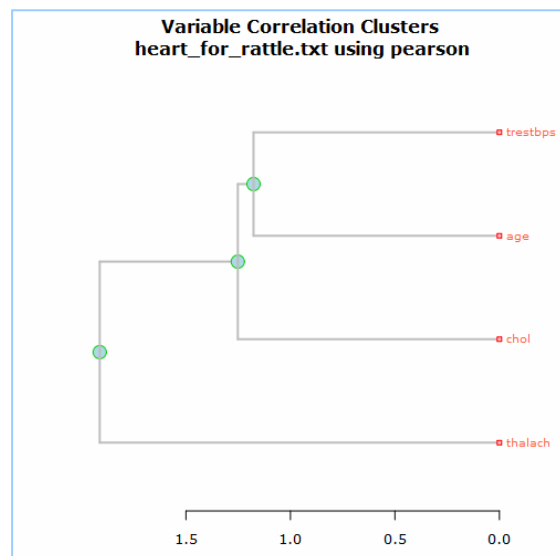


Concernant les variables prédictives catégorielles, nous disposons de plusieurs types de graphiques, dont la mosaïque.



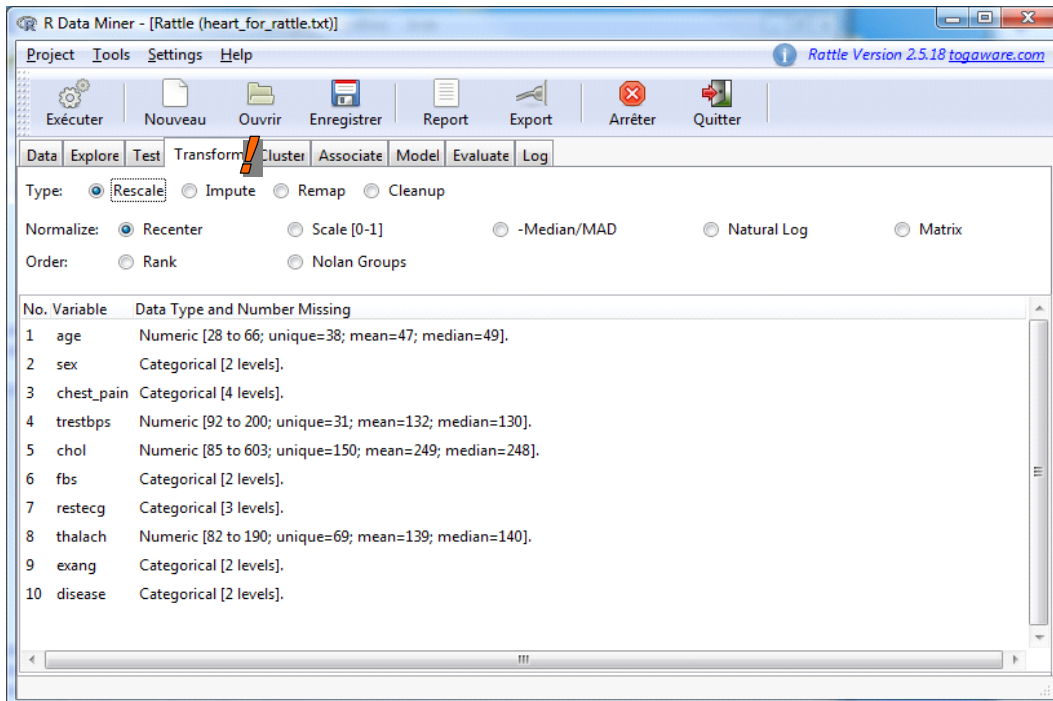
En un coup d’œil, nous obtenons une série d’informations très instructives. Prenons l’exemple de la variable SEX, nous observons que les hommes (MALE) sont plus nombreux que les femmes (FEMALE), il y a proportionnellement plus de malades parmi eux (DISEASE = POSITIVE).

D’autres outils sont disponibles. CORRELATION calcule les corrélations entre les variables prédictives numériques. Il peut décrire une hiérarchie de proximités à l’aide d’un dendrogramme. Lorsqu’elles sont toutes numériques, cela peut être utile pour créer des groupes de variables et ainsi réduire leur nombre lors de la prédiction. Concernant notre fichier, nous observons...



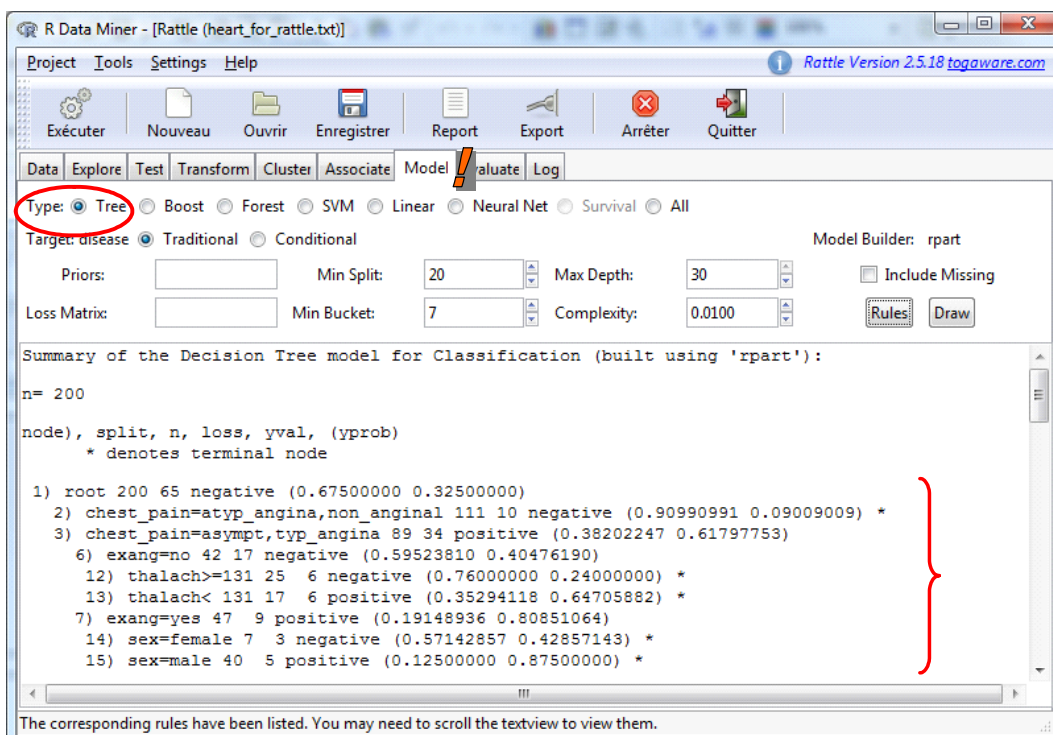
### 3.4 Transformation des données

L'onglet « Transform » est dédié à la transformation de variables. Nous n'en effectuerons pas dans ce didacticiel, néanmoins il est intéressant de noter que la majorité des opérateurs usuels sont disponibles (ex. normalisation, passage au logarithme, passage aux rangs, etc.).



### 3.5 Apprentissage supervisé

Nous abordons le cœur du sujet lorsque nous activons l'onglet « Model ». Nous utiliserons 3 méthodes : un arbre de décision, une forêt aléatoire et une régression logistique.



Concernant l'arbre de décision, rattle s'appuie sur la fonction `rpart` du package éponyme. Nous conservons les paramètres par défaut, nous actionnons directement le bouton EXECUTER. L'arbre comporte 5 feuilles. Nous pouvons lister les règles en cliquant le bouton RULES.

The screenshot shows the Rattle software interface. At the top, there are tabs for Data, Explore, Test, Transform, Cluster, Associate, Model, Evaluate, and Log. Below the tabs, there are radio buttons for Type (Tree, Boost, Forest, SVM, Linear, Neural Net, Survival, All) and Target (disease, Traditional, Conditional). The Model Builder is set to rpart. Parameters include Priors, Min Split (20), Max Depth (30), Loss Matrix, Min Bucket (7), and Complexity (0.0100). There are buttons for Rules and Draw. Below the interface, the 'Tree as rules:' section lists five rules with their respective parameters and probabilities. A red arrow points from the 'Rules' button to the list of rules.

```

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

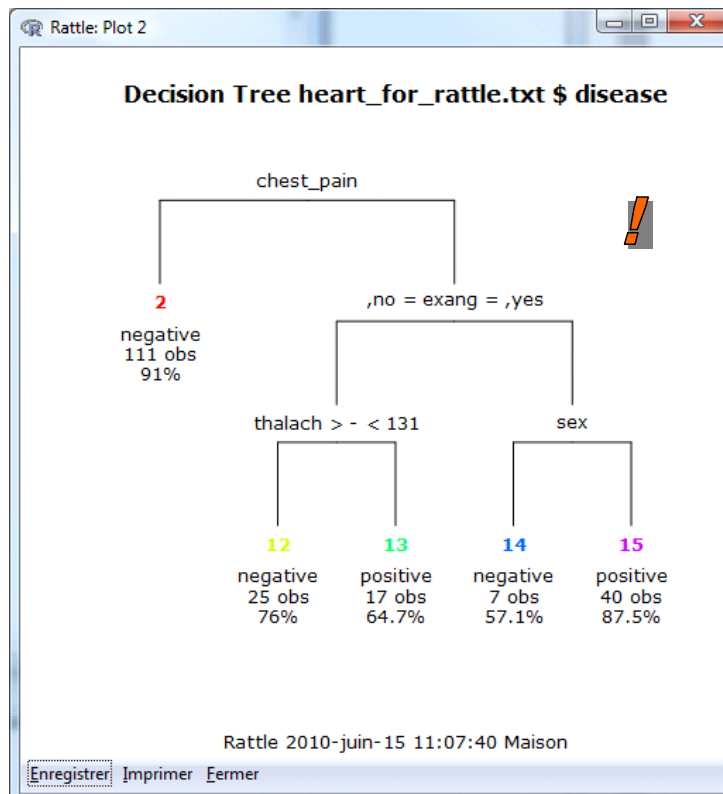
Rule number: 12 [yval=negative cover=25 (12%) prob=0.24]
chest_pain=asympt,typ_angina
exang=no
thalach>=131

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
    
```

Nous pouvons aussi obtenir une représentation graphique de l'arbre en actionnant le bouton DRAW.



Les **forêts aléatoires** sont accessibles via la fonction **randomForest** du package du même nom. Nous obtenons le résultat suivant avec les paramètres par défaut.

Summary of the Random Forest model:

```
Call:
randomForest(formula = disease ~ ., data = crs$dataset[crs$sample, ], ntree = 500, mtry = 
              Type of random forest: classification
              Number of trees: 500
No. of variables tried at each split: 3

              OOB estimate of error rate: 16.5%
Confusion matrix:
              negative positive class.error
negative      123      12 0.08888889
positive       21      44 0.32307692

Variable Importance
              negative positive MeanDecreaseAccuracy MeanDecreaseGini
chest_pain    1.77    3.80           2.17           19.89
thalach        0.70    2.46           1.40           14.69
exang         1.68    2.89           1.85           13.66
chol          0.66    0.54           0.56           12.60
age          -0.07    0.39           0.11           9.78
trestbps     -0.46    0.67          -0.02           7.44
sex           0.33    2.82           1.44           4.36
restecg      0.36   -0.73          -0.12           1.93
fbs          0.39   -0.57          -0.05           1.43
```

Display the Model

To view model 5, for example, execute the command  
`printRandomForests(crs$rf, 5)`  
in the R console. Generating all models will take quite some time.

Time taken: 0.42 secs

Generated by Rattle 2010-06-15 13:19:14 Maison

The Random Forest model has been built. Time taken: 0.42 secs

On notera entre autres la section « Variable Importance » qui indique la pertinence des prédicteurs. Le taux d'erreur « Out-of-Bag » est de 16.5%. Nous nous en rappellerons lorsque nous appliquerons le modèle sur l'échantillon test.

Dernier modèle que nous souhaitons mettre en œuvre : la régression logistique avec la fonction **glm()**. A priori, elle n'est pas utilisable directement dans notre situation puisque certaines variables prédictives sont catégorielles. Leur recodage 0/1 est nécessaire. La fonction **glm()** s'en occupe directement. Pour chaque variable, il prend comme catégorie de référence la première modalité du type « factor » c.-à-d. la première par ordre alphabétique.

A la sortie, nous avons les résultats suivants.

R Data Miner - [Rattle (heart\_for\_rattle.txt)]

Project Tools Settings Help Rattle Version 2.5.18 togaware.com

Exécuter Nouveau Ouvrir Enregistrer Report Export Arrêter Quitter

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type:  Tree  Boost  Forest  SVM  Linear  Neural Net  Survival  All

Numeric  Generalized  Poisson  Logistic  Probit  Multinomial Model Builder: glm (logit)

Plot

Summary of the Logistic Regression model (built using glm):

Call:  
`glm(formula = disease ~ ., family = binomial(link = "logit"),  
 data = crs$dataset[crs$sample, ])`

Deviance Residuals:  
 Min 1Q Median 3Q Max  
 -2.5490 -0.4750 -0.2661 0.4889 2.8434

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.294096	4.036321	0.073	0.941916
age	-0.007997	0.033876	-0.236	0.813390
sexmale	1.189818	0.587423	2.025	0.042818 *
chest_painatyp_angina	-2.186895	0.575810	-3.798	0.000146 ***
chest_painnon_anginal	-1.586384	0.627181	-2.529	0.011426 *
chest_paintyp_angina	0.517539	0.923208	0.561	0.575079
trestbps	-0.004770	0.012892	-0.370	0.711408
chol	0.005226	0.003447	1.516	0.129502
fbst	1.251407	0.833696	1.501	0.133346
restecgnormal	1.098985	2.277026	0.483	0.629351
restecgst_t_wave_abnormality	0.426301	2.314418	0.184	0.853862
thalach	-0.023186	0.011624	-1.995	0.046071 *
exangyes	1.874358	0.494388	3.791	0.000150 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 252.23 on 199 degrees of freedom  
 Residual deviance: 144.08 on 187 degrees of freedom  
 AIC: 170.08

Number of Fisher Scoring iterations: 6

Log likelihood: -72.039 (13 df)  
 Null/Residual deviance difference: 108.153 (12 df)  
 Chi-square p-value: 0.00000000  
 Pseudo R-Square (optimistic): 0.70482172

### 3.6 Évaluation

Dernière étape de notre processus exploratoire, nous souhaitons évaluer les performances de nos 3 modèles prédictifs sur l'échantillon test (30% des observations disponibles, section 3.2).

Nous passons à l'onglet « Evaluate ». Dans un premier temps, nous demandons la matrice de confusion ERROR MATRIX. Attention, il faut veiller à sélectionner l'option DATA = TESTING. Enfin, les trois modèles que nous avons élaborés dans la section précédente sont automatiquement sélectionnés. Nous pouvons évaluer qu'une partie d'entre eux, nous ne pouvons pas en revanche tester un modèle qui n'aurait pas été construit dans l'onglet « model ».

Nous cliquons sur le bouton EXECUTER de la barre d'outils. Nous obtenons les matrices de confusion en valeur et en pourcentage.

R Data Miner - [Rattle (heart\_for\_rattle.txt)]

Project Tools Settings Help Rattle Version 2.5.18 togaware.com

Exécuter Nouveau Ouvrir Enregistrer Report Export Arrêter Quitter

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type:  Error Matrix  Risk  Cost Curve  Hand  Lift  ROC  Precision  Sensitivity  Prv Ob  Score

Model:  Tree  Boost  Forest  SVM  Linear  Neural Net  Survival  KMeans  HClust

Data:  Training  Validation  Testing  CSV File (Aucun)  R Dataset

Risk Variable: Report:  Class  Probability Include:  Identifiers  All

Error matrix for the Decision Tree model on heart\_for\_rattle.txt [test] (counts):

	Actual	
Predicted	negative	positive
negative	23	6
positive	3	12

Error matrix for the Decision Tree model on heart\_for\_rattle.txt [test] (%):

	Actual	
Predicted	negative	positive
negative	52	14
positive	7	27

Overall error: 0.2045455

Generated by Rattle 2010-06-15 13:48:39 Maison

Error matrix for the Random Forest model on heart\_for\_rattle.txt [test] (counts):

	Actual	
Predicted	negative	positive
negative	25	8
positive	1	10

Error matrix for the Random Forest model on heart\_for\_rattle.txt [test] (%):

	Actual	
Predicted	negative	positive
negative	57	18
positive	2	23

Overall error: 0.2045455

Generated by Rattle 2010-06-15 13:48:39 Maison

Error matrix for the Linear model on heart\_for\_rattle.txt [test] (counts):

	Actual	
Predicted	negative	positive
negative	24	6
positive	2	12

Error matrix for the Linear model on heart\_for\_rattle.txt [test] (%):

	Actual	
Predicted	negative	positive
negative	55	14
positive	5	27

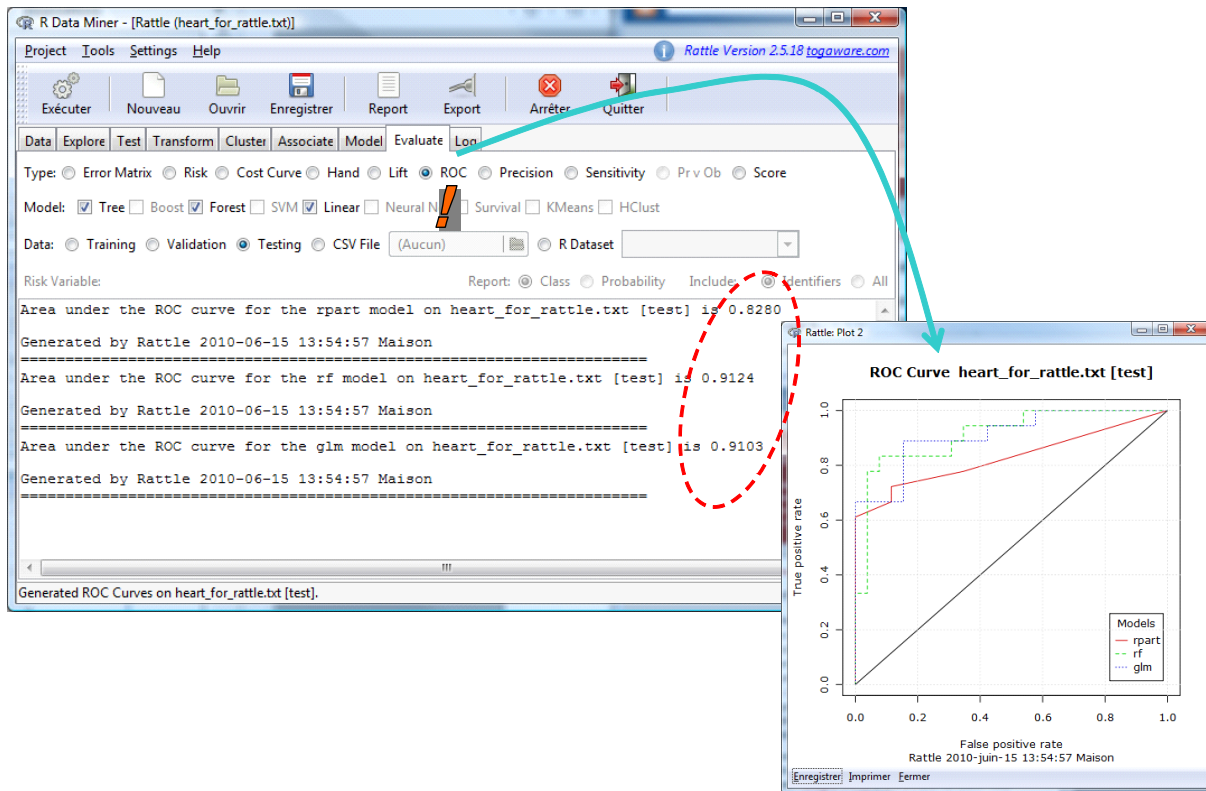
Overall error: 0.1818182

Generated by Rattle 2010-06-15 13:48:39 Maison

Generated Error Matrix.

Les taux d'erreur en test sont : arbre de décision = 20.45% ; forêt aléatoire = 20.45% et régression logistique = 18.18%. Au-delà de ces valeurs trompeuses, nous constatons surtout que le nombre de mal classés est seulement inférieur à unité pour la régression logistique (8 mal classés vs. 9 pour l'arbre et la forêt aléatoire). En réalité, les modèles se valent en classement.

Pour évaluer leur capacité à scorer c.-à-d. à attribuer un score plus élevé aux positifs par rapport au négatifs, nous utilisons la courbe ROC (<http://tutoriels-data-mining.blogspot.com/2009/10/evaluation-des-classifieurs-quelques.html>).



Si l'on se réfère aux courbes et à l'indicateur AUC (Area under curve), nous constatons que la régression logistique et la forêt aléatoire sont proches, l'arbre de décision est un peu en retrait. Ce n'est pas une surprise, un arbre de décision n'est pas un outil adapté au scoring.

### 3.7 Programme associé aux traitements

The screenshot shows the 'Log' window in R Data Miner. The code in the window is as follows:

```
# little effort the log can be used to score a new dataset. The logical variable
# 'building' is used to toggle between generating transformations, as when building
# a model, and simply using the transformations, as when scoring a dataset.

building <- TRUE
scoring <- !building

# The colorspace package is used to generate the colours used in plots, if available.

library(colorspace)

#=====
# Rattle timestamp: 2010-06-15 14:34:33 i386-pc-mingw32

# Load the data.

crs$dataset <- read.csv("file:///D:/DataMining/Databases_for_mining/benchmark_datasets/Heart/heart_for_rattle.txt", sep="\t", na.strings=c("", "
#=====
# Rattle timestamp: 2010-06-15 14:34:34 i386-pc-mingw32

# Note the ...
```

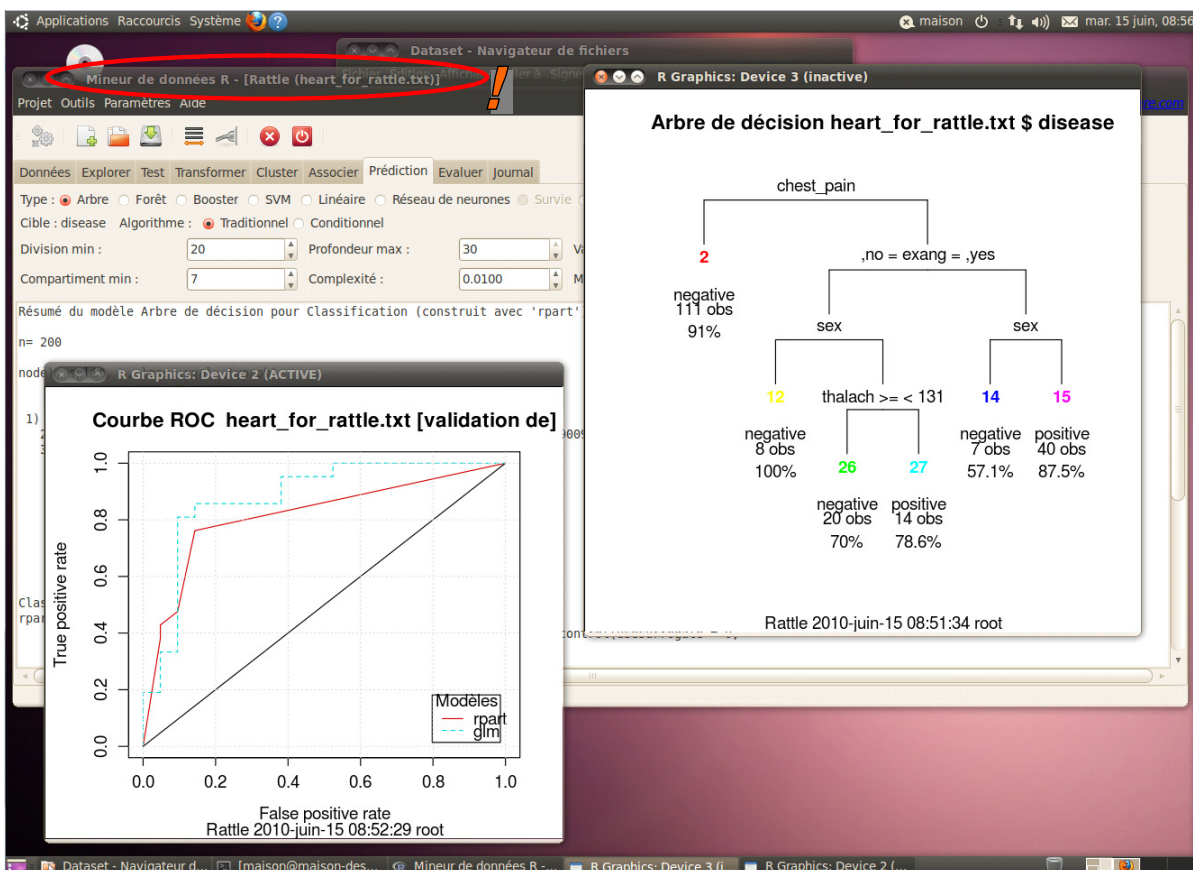
L'un des principaux reproches que l'on fait aux logiciels pilotés par menu dans le data mining est qu'une fois l'analyse finalisée, lorsque nous refermons le logiciel, nous n'avons plus de traces des traitements effectués. Au prochain démarrage du logiciel, il est compliqué de reproduire à l'identique la séquence des opérations, il faut avoir une excellente mémoire, ou avoir pris soin de noter un bout de papier tout ce qu'on a fait.

Rattle échappe à cet inconvénient en transcrivant en code R toutes les manipulations réalisées et validées par un clic sur le bouton EXECUTER. Le programme, accessible dans l'onglet « Log » peut être sauvegardé. A notre prochaine session de travail, pour reproduire exactement l'analyse, il suffit de faire exécuter le code avec la commande `source()` de R.

## 4 Rattle sous Linux (Ubuntu)

L'installation du package rattle pour R sous Linux (Ubuntu) n'est pas facile. Il faut suivre à la lettre les instructions ([http://datamining.togaware.com/survivor/Install\\_GNU\\_Linux.html](http://datamining.togaware.com/survivor/Install_GNU_Linux.html)). Et encore, nous ne sommes pas sûr que cela fonctionne correctement. Une procédure de secours est indiquée, c'est celle que j'ai utilisée pour parvenir à mes fins.

Passé cet écueil, rattle fonctionne parfaitement sous Linux.



## 5 Conclusion

Dans ce didacticiel, nous avons montré qu'il était possible d'utiliser R en le pilotant par menu à l'aide du package rattle. Cette librairie est plutôt spécialisée dans les méthodes dites de Data Mining. Le

package « R Commander » ([Rcmdr](#)) fait la même chose, mais il est plutôt tourné vers les méthodes de statistique et d'analyse de données (<http://cran.r-project.org/web/packages/Rcmdr/index.html> ; <http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/>).