

Travaux d'Études et de Recherche
Rapport Intermédiaire

Julien Henry
Nicolas Lacourte-Barbadaux
Alexandre Masson
Léo Rousseau

14 Janvier 2013

Table des matières

1	Résumé du projet	3
2	Introduction du domaine	4
3	Analyse de l'existant	5
3.1	le logiciel R	5
3.2	Le logiciel Explorer3D	7
3.3	Packages R utilisables	9
3.3.1	RGL	9
3.3.2	RGTK	11
4	Besoins non fonctionnels	13
4.1	Montée en charge	13
4.2	Documentation des technologies	14
5	Besoins fonctionnels	15
6	Description des prototypes	17
7	Planning, affectation des taches	18
8	Bibliographie	19

1 Résumé du projet

R est un logiciel de statistiques open source. De notre côté, nous développons depuis plusieurs années un logiciel de visualisation de données en java doté d'une interface graphique 3D interactive.

Il s'agit ici d'étudier le portage de cette application vers R, en se focalisant notamment sur les capacités d'interaction des librairies 3D proposées par ce système.

2 Introduction du domaine

La classification La classification est une forme d'apprentissage automatique. Elle consiste, étant donné un ensemble d'éléments, à les répartir dans différentes catégories. Il existe deux courants majoritaires de classification.

La classification supervisée Dans la classification supervisée on cherche à déterminer la étant-donné un ensemble d'apprentissage dont on connaît déjà la classification. Par exemple si on classe des animaux on doit pouvoir déterminer que le cobra est un reptile parce qu'il a une description proche des reptiles déjà présents dans l'ensemble initial.

La classification non-supervisée La classification non-supervisée, ou encore clustering, a un but différent. On cherche à classer un ensemble de données en plusieurs classes sans les connaître à l'avance. On calcule une répartition pour l'ensemble. C'est ce que fait par exemple l'algorithme des K-Moyennes[1].

3 Analyse de l'existant

3.1 le logiciel R

Le logiciel R est à la fois un environnement interactif de calcul, en lignes de commandes, et un langage de programmation. Il existe sur différentes plateformes (Linux, MacOS et Windows). Il peut être utilisé pour des traitements de données ou des analyses statistiques. R est un logiciel libre, ouvert à la participation de développeurs extérieurs. En effet, l'environnement s'enrichit par l'intégration de packages provenant de la communauté ajoutant des modules et des fonctionnalités complémentaires au logiciel initial.



Le logiciel R

La communauté R est très active et de nombreux modules sont régulièrement proposés. R permet facilement de charger des données, de sortir des informations statistiques, de visualiser diverses courbes extraites des données, etc.

Par défaut, R se présente sous la forme d'une ligne de commande. Il est cependant possible de développer des interfaces utilisateur pour contrôler l'application ainsi que pour visualiser des résultats de traitement.

```
R version 2.11.1
Copyright (C) 2010 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.
```

```
>
```

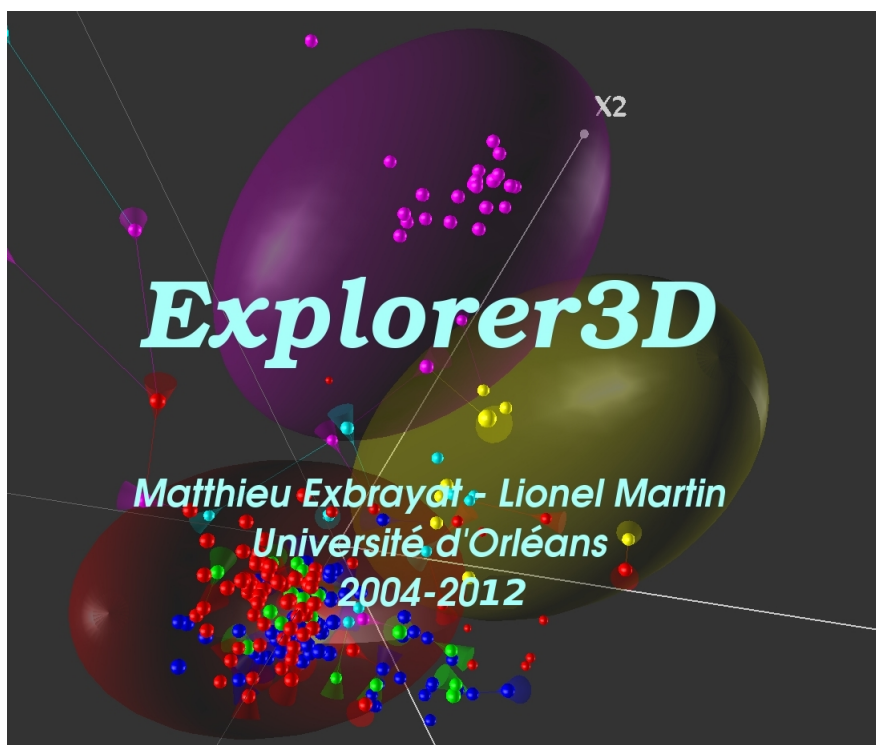
Aperçu de la console R

Le logiciel R nous permet de traiter des données tabulaires, sous forme de vecteurs, ou de data-frame (matrice contenant les instances de objets en ligne, avec un attribut par colonne). Il propose ensuite des opérations sur les vecteurs, tout en discriminant les objets par leurs attributs non numériques. Ce qui permet d'étiqueter les objets avant de faire des calculs et pouvoir ensuite différencier les résultats des calculs en fonction des étiquettes des objets.

On utilisera sans doute cette spécificité pour colorer les objets dans la fenêtre 3D, en fonction de leurs propriétés.

3.2 Le logiciel Explorer3D

De notre côté, nous développons depuis plusieurs années un logiciel de visualisation de données doté d'une interface graphique 3D interactive. Ce logiciel est écrit en Java. [http ://www.univ-orleans.fr/lifo/software/Explorer3D/](http://www.univ-orleans.fr/lifo/software/Explorer3D/)



Explorer3D

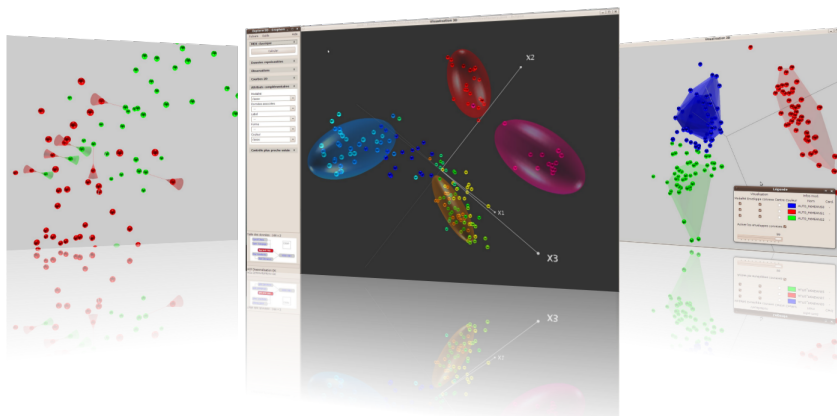
Nous avons acquis une bonne maîtrise de la structuration d'un tel outil afin de le rendre évolutif.

Nous avons également mis en place un certain nombre de fonctionnalités interactives. Nous étudions maintenant les possibilités d'interactions entre ce logiciel et la plateforme R, afin d'intégrer rapidement divers outils existant sous R, mais également de diffuser notre logiciel vers cette communauté.

Au moins deux approches sont possibles :

- intégrer des invocations de R depuis java.
- migrer notre logiciel vers R.

Concernant le premier point, l'invocation de R depuis java est relativement simple. Elle ne sera pas abordée dans ce TER. Concernant le second point, R propose la création de fenêtres de contrôle et de fenêtres graphiques 2D et 3D (voir par exemple <http://rgl.neoscientists.org/>). Toutefois, nous ne connaissons pas les possibilités d'interaction réelles de l'interface 3D.



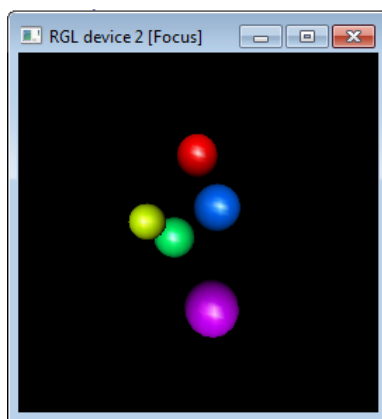
Les possibilités d'Explorer3D

3.3 Packages R utilisables

3.3.1 RGL

RGL est un package pour le langage de programmation R. Il étend les possibilités de R avec l'ajout d'outils de visualisation 3D en temps réel.

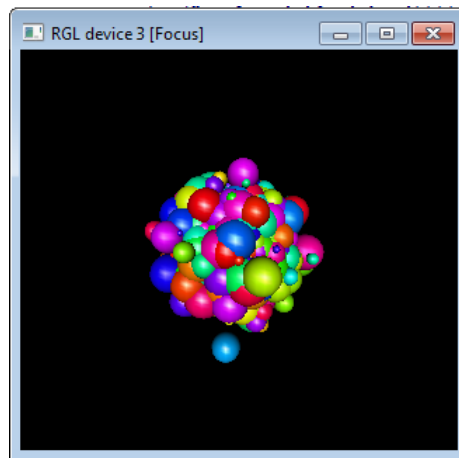
Les univers 3D ont besoin d'être projetées sur des images 2D affichées à l'écran, c'est pourquoi il est nécessaire d'avoir un outils qui effectue cette étape afin d'afficher des objets tri-dimensionnel à l'écran. Cet outils doit simuler la lumière, la structure des objets et leurs textures afin de donner l'illusion de 3D sur les images 2D projetées.



Des sphères dans RGL

Le coeur de RGL est codé en c++ et utilise OpenGL. RGL est ainsi une interface entre R et OpenGL. OpenGL est un standard des applications graphiques utilisé dans un grand nombre d'applications et langages. Il résout notamment le problème d'affichage en 2D d'univers 3D cité plus tôt.

Plusieurs fonctionnalités sont accessibles simultanément via RGL, telles que l'intégration drag/drop dans la fenêtre de visualisation et la réception de nouvelles instructions depuis la console de commande de R. Ainsi les objets peuvent être examinés en trois-dimension grâce au zoom et la scène peut pivoter sur elle-même.



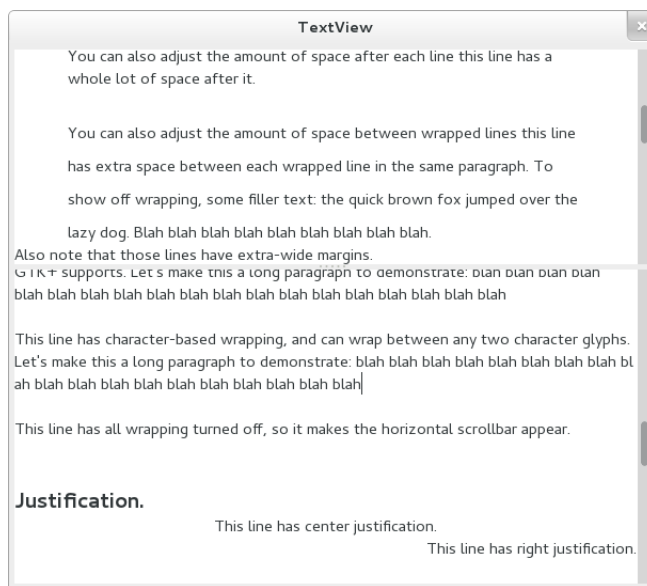
Plus de sphères dans RGL

Le but de RGL est donc de surcoucher OpenGL pour le langage R. Il permet ainsi via des instruction en R de générer des scènes 3D et d'interagir avec celles-ci via de nombreux outils.

3.3.2 RGTK

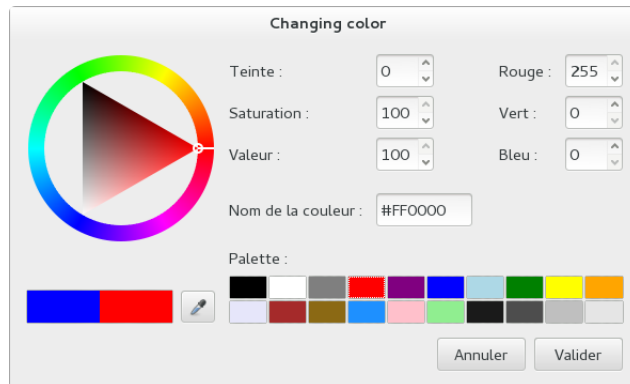
Le package RGtk2 s'ajoute en tant que librairie externe pour inclure dans R la possibilité de créer des interfaces graphiques.

Comme dans Explorer3d, on a besoin de gérer les données grâce à l'interface graphique. Cette extension de R est essentielle dans le développement de l'application.



fenêtre divisé en plusieurs parties

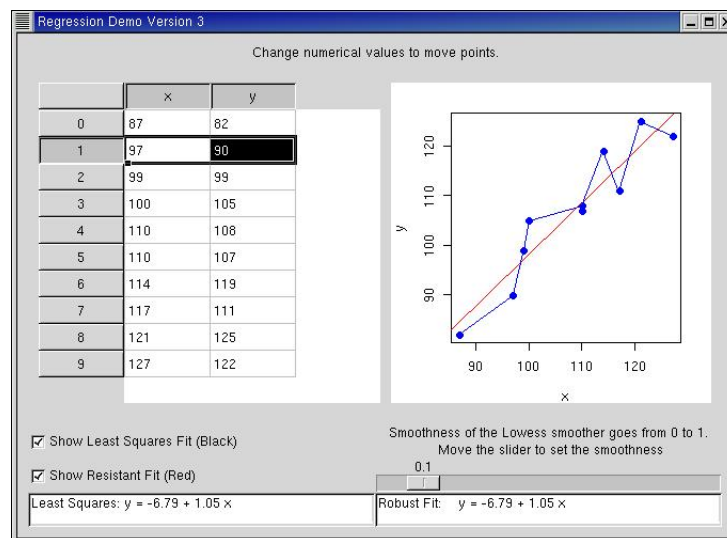
RGtk est un package dérivé de GTK2. Il est écrit en c++. GTK est un standard pour la création d'interface utilisateurs en c++ et bien d'autres langages. Grâce à RGtk2 on résout le problème de création d'interface graphique. Comme dans la capture d'écran précédente on peut créer une fenêtre avec plusieurs compartiment à l'intérieur de celle ci. ON peut aussi avoir une fenêtre pour choisir la couleur d'un objet objet affiché dans une vue RGL.



Sélectionneur de couleur RGtk

Le but de RGtk2 est de surcoucher GTK. Avec ce package on pourra interagir avec le logiciel.

Nous trouvons ce package intéressant, car il nous offre la possibilité de créer des interfaces graphiques pour l'utilisateur, ce qui lui permettra de faire les demandes de calculs, sans passer par la console.



Un autre exemple d'interface graphique avec RGtk

4 Besoins non fonctionnels

- Montée en charge
- Documentation des technologies à utiliser

4.1 Montée en charge

Comme expliqué plus haut, nous allons différer de la méthode utilisée dans Explorer3D pour toute la partie affichage. Pour savoir si les solutions choisies sont acceptables, il sera nécessaire de faire des tests de montée en charge.

Pour cela nous allons tester la réactivité du système aux différents événements, que ce soit des demandes de calculs, ou des interactions avec la scène 3D (rotation, zoom, déplacement, etc...).

Pour réaliser ces tests, nous procéderons à la main aux différentes manipulations, et nous nous baserons à notre jugement en terme de vitesse de réponse, car il n'est pas facile de chronométrer la réactivité de la scène 3D, nous allons bien entendu refaire ces tests avec un nombre de plus en plus grand d'objets à afficher, pour pouvoir déterminer une limite en terme de quantité d'objets représentable.

L'objectif de ce TER consiste à : étudier le mécanisme de développement d'interfaces utilisateur sur le plan 1/ des fenêtres de contrôle, et 2/ des fenêtres de visualisation 3D et 2D ; • en produire une synthèse ; • réaliser une maquette dont le contour précis sera discuté avec l'enseignant (charger un jeu de données, visualiser en 3D après préparation des données, sélectionner des objets de la scène 3D grâce à la souris, afficher des images).

4.2 Documentation des technologies

Dans le cadre de cette étude nous allons être amenés à utiliser certaines technologies qui ne seront pas forcément faciles à mettre en œuvre. Cette étude devant se placer dans la continuité d'un projet plus large, nous allons aussi devoir produire et introduire dans notre rendu, une documentation sur les technologies utilisées.

Mais plus qu'une documentation technique, nous souhaitons produire plusieurs petits tutoriaux, qui permettront à nos successeurs de gagner du temps, et de prendre en main plus facilement le résultat de notre travail, ainsi que les technologies utilisées pour mener à bien ce projet.

5 Besoins fonctionnels

Les principaux besoins fonctionnels sont les fonctionnalités d'Explorer3D qu'il est nécessaire de porter sous R, du moins autant qu'il sera possible d'en porter.

Nous allons lister les différentes fonctionnalités souhaitées et nous les commenterons.

Nous souhaitons pouvoir fournir à l'utilisateur une interface graphique lui permettant d'agir sur les données, offrant diverses possibilités, telles que le chargement de données, ou la demande de calcul. Il doit aussi avoir la possibilité d'ouvrir une nouvelle fenêtre 3D interactive.

Nous allons donc déterminer si il est possible de créer ces différentes fenêtres, aussi nous chercherons à savoir comment pourrions nous agir depuis cette interface utilisateur sur la ou les fenêtres 3D.

Nous allons maintenant préciser ce que nous attendons des fenêtres 3D.

Nous l'avons évoqué dans les besoins non fonctionnels, nous voulons avoir une fenêtre qui nous permettent d'afficher un très grand nombre d'objets, nous ne donnerons pas d'ordre de grandeur pour le moment, car nous en saurons plus après les premiers tests. - gestion de fenêtres de dialogue permettant le chargement de données tabulaire, le calcul de projections, l'ouverture d'une fenêtre 3D interactive.

Nous souhaitons aussi avoir la possibilité de modifier la scène 3D, plus précisément nous souhaitons connaître les possibilités de zoom et de déplacement de la scène dans la fenêtre 3D. Nous nous intéressons aussi à la rotation de la scène, nous aimerions pouvoir faire tourner la scène et ainsi avoir un autre point de vue de la répartition des objets.

Comme dans Explorer3D, nous voulons offrir la possibilité de sélectionner certains objets, par un clic sur l'objet dans la fenêtre 3D. il sera donc nécessaire de pouvoir connaître l'état d'un objet, s'il est sélectionné ou non. Ici Nous cherchons, plus qu'un côté algorithmique, à savoir comment pourrions nous, si possible, implémenter ça avec RGL.

Nous souhaiterions aussi avoir la possibilité de changer les propriétés des objets, par exemple la couleur, afin de pouvoir mettre en évidence les objets sélectionnés, et ceux de manière la plus fluide et rapide possible.

Nous souhaitons aussi offrir la possibilité de repositionner les objets dans l'espace, et nous souhaitons pouvoir afficher dynamiquement ce repositionnement, nous chercherons ici à afficher une animation, et pas seulement rafraîchir la scène avec la nouvelle position de l'objet déplacé.

Nous aimerons aussi pouvoir rajouter de façon dynamique des objets dans la scène tels que des ellipses ou ces enveloppes connexes, encore une fois, sans avoir à rouvrir une nouvelle fenêtre, mais juste en mettant à jour la fenêtre courante.

Nous souhaiterions avoir la possibilité de pouvoir faire cohabiter plusieurs fenêtres 3D, par exemple pour comparer plusieurs représentations des objets.

Explorer3D propose aussi un système de loupe, qui en plus d'un zoom optique sur une partie de la scène où se trouve le pointeur de la souris, sélectionne tous les objets présents sous la loupe, ce qui permet par exemple de mettre en évidence, la présence d'un groupe d'objets dans une vue (une fenêtre) mais ce même "groupe" serait dispersé dans une autre vue.

- AJOUT : possibilité de se mettre en écoute sur un port pour réception de commandes de type jeu de données + affichage.

6 Description des prototypes

7 Planning, affectation des taches

8 Bibliographie