R++, un logiciel d'analyse statistique simple et intuitif

Christophe Genolini

R++, un logiciel d'analyse statistique simple et intuitif

IHM'23 Conference

Christophe Genolini
Zebrys
Toulouse, France
cg@rplusplus.com

ABSTRACT

Data science is gradually entering all areas of society, both in academia and in the private sector. Statistical analysis software are used by data scientist but also by non-experts (medical doctors, industrial, human resources, …). Unfortunately, they are integrated into obsolete interfaces that completely ignore principles of HCI and are poorly adapted to non-expert users. The R++ project aims to develop a modern statistical analysis software program integrated into a user-friendly interface. In this paper, we present the methodology that led us to the design of R++. We also give examples that this methodology allowed us to achieve.

CCS CONCEPTS

• **Human-centered computing** → **HCI theory, concepts and models**; **Pointing**; **Visualization techniques**.

KEYWORDS

Statistical analysis, Video prototyping, R++

RÉSUMÉ

La science des données est en train de progressivement infiltrer tous les domaines de la société. De plus en plus de non-statisticiens y ont recours : chercheurs, mais aussi médecins, industriels, psychologues, commerciaux, marketeurs, communiquants… Malheureusement, les logiciels d'analyse statistique

Christophe Genolini. 2023. R++, un logiciel d'analyse statistique simple et intuitif. *IHM'23 : Actes étendus de la 34ème conférence Francophone sur l'Interaction Humain-Machine, April 03–06, 2023, Troyes, France.*

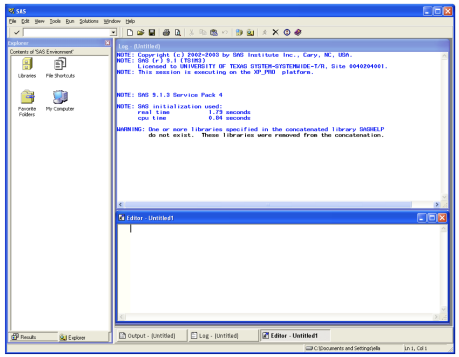

Figure 1 : Interface du logiciel SAS

sont généralement intégrés dans des interfaces obsolètes mal adaptées à des non-spécialistes et qui ignorent complètement les principes de l'IHM.

Le projet R++ est de développer un logiciel d'analyse statistique à destination des non-statisticiens. Dans cet article, nous présentons la méthodologie qui nous a conduit à la conception de R++. Nous donnons aussi des exemples de ce que cette méthodologie nous a permis de réaliser.

MOTS CLÉS

Analyse statistique, Prototypage de video, R++

1. INTRODUCTION

L'analyse statistique devient progressivement le pivot incontournable de nombre de domaines [1]. Dans le monde académique, on utilise des comparaisons de groupes, les estimations d'un modèle de régression, ou encore le machine learning. Dans le privé, les assureurs l'utilisent pour définir leurs tarifs, les banquiers s'en servent pour décider de l'octroi d'un prêt, l'industrie pharmaceutique valide ses essais cliniques et, de manière plus générale, tous les grands comptes rêvent d'un modèle qui permettrait de savoir qu'un client va partir avant même que le client prenne sa décision…

Les utilisateurs de l'analyse statistique sont donc très variés. Les data scientists sont généralement formés à l'utilisation des logiciels. Mais beaucoup d'autres corps de métiers ont besoin d'utiliser des statistiques sans pour autant être expert.

Du point de vue logiciel, les logiciels métiers (conçu par les statisticiens, SAS, R, SPSS, Stata…) comme les solutions informatiques (Oracle, C, python) ont en commun d'être intégrés dans des interfaces particulièrement difficiles à utiliser (Figure 1). Elles ne sont pas adaptées aux besoins des utilisateurs non-programmeurs et non-statisticiens.

Le projet R++ a pour objectif de développer un logiciel d'analyse statistique adapté à des utilisateurs ayant un métier principal et pour qui l'analyse statistique n'est qu'un outil secondaire. Entre autres caractérisques, ce logiciel doit être simple d'apprentissage, car il est fréquent que des utilisateurs fassent parfois de longues poses entre deux utilisations. Par exemple, les médecins-chercheurs font régulièrement quatre à six mois de clinique pendant lesquels ils ne font plus du tout de statistiques, puis un à deux mois de statistiques. Il faut donc un logiciel tellement simple que la reprise en main après six mois de pause soit évidente.

Dans cet article, nous présentons la méthodologie qui nous a amené la conception de R++. La Section 2 rappelle brièvement le concept de prototypage vidéo. La Section 3 détaille des éléments qui ont été améliorés. La Section 4 détaille un exemple d'utilisation de R++ par un non expert.

Data-Management	**33**
Gestion des valeurs aberrantes	
Variables mal codées	
Dates	
Filtres	
...	
Graphiques	**24**
Dynamique	
Interactif	
Exportable	
DPI	
Tableaux	**16**
Interactifs	
Gestion des styles	
Aide	**14**
Intégrée	
Vidéo	
Contextuelle	
Wiki	
Export	**13**
Word	
Rapport automatique	
Style de document	
Timeline de l'analyse	**9**
Autres	**41**

Figure 2 : Regroupement thématique des éléments a améliorer dans les logiciels statistiques

2. MÉTHODES

Pour concevoir l'IHM de R++, nous avons utilisé la méthode du design exploratory et du prototypage video : le principe est de réunir pendant trois heures 4 ou 5 utilisateurs (des non-statisticiens ayant besoin de statistiques) et 4 ou 5 spécialistes de l'IHM. La séance est ensuite découpée en trois phases. (1) Pendant une heure, nous lançons un brainstorming sur ce qu'il faut améliorer dans les logiciels actuels : « Quelles sont avec vos outils actuels les tâches compliquées, chronophages, pénibles, fastidieuses ou à fort risque d'erreur ? » Ce tour de table est fait sur le mode brainstorming ouvert : les participants interviennent et un modérateur note leurs remarques au tableau. À la fin de cette première étape, le groupe sélectionne un ou deux thèmes.

(2) Dans une deuxième étape, le groupe cherche des solutions aux thèmes sélectionnés. La formule introductive est : « Imaginez : vous marchez sur une plage. Vous trouvez une lampe. Vous la frottez et une fée en sort. Chance, vous avez le droit à trois vœux. Par contre, la fée est la fée des logiciels d'analyse statistique (!), les trois vœux ne peuvent servir que pour améliorer votre logiciel. Comment fonctionnerait le logiciel d'analyses statistique de vos rêves ? » Pendant 10 minutes, les participants doivent noter par écrit trois idées pour résoudre les problèmes choisis, plus une idée « farfelue ». L'objectif de l'idée farfelue est d'éviter l'autocensure : en effet, un participant peut avoir une bonne idée mais ne pas la dire par peur d'être ridicule. La consigne de l'idée farfelue permet d'éviter cela. Quand tout le monde a trouvé ses trois plus une idées, elles sont présentées au reste du groupe, puis elles sont débattues, combinées et améliorées.

(3) Enfin, dans une troisième heure, les participants créent des prototypes basses fidélités. À l'aide de papier, feutre, post-it et découpage, ils fabriquent un scenario qu'ils filment à l'aide d'un téléphone. Ces prototypes sont ensuite présentés à l'ensemble du groupe. Cela permet d'obtenir un premier retour.

Bilan, en à peine 3 heures, le groupe a identifié de vrais problèmes utilisateurs, puis a trouvé collectivement des solutions et enfin a pu avoir un premier retour sur les solutions. Au total, nous avons organisé huit sessions. Lors de la phase « choses à améliorer », de nombreuses idées ont été proposées. Elles ont ensuite été collectivement regroupées en thèmes (voir Figure 2).

Nous avons également organisé des variantes thématiques lors desquelles l'ordre du jour était annoncé en amont. Par exemple, lors de la session « Data management », seules les personnes directement concernées par le data management et ayant déjà une première expérience de ce genre de brainstorming étaient conviées. Le déroulement de la séance était alors modifié : (1) rapide tour de table sur les problèmes de data management (2) recherche de solutions (3) création et présentation des prototypes vidéo (4) présentation des vidéos conçues en séance (5) présentation de vidéos tournées lors d'autres sessions sur le thème du data management (6) à partir des vidéos du jour et des vidéos

précédentes, deuxième tour de recherche de solutions (étape 2 bis), création d'une deuxième série de vidéos intégrant l'ensemble des idées (3 bis) et présentation à l'équipe (étape 4 bis).

Au total, nous avons organisé onze sessions de variantes thématiques, soit un total de 19 sessions.

3. EXEMPLE D'AMÉLIORATION : LE NETTOYAGE DES DONNÉES

Le nettoyage des données (ou data management) consiste à préparer les données avant leur analyse. C'est une étape indispensable et malheureusement négligée par à peu près tous les logiciels d'analyse statistique, malgré son importance capitale pour les utilisateurs (voir figure 2). Elle consiste à repérer et corriger les incohérences dans les données : une personne à 3500 ans, une variable[1] binaire à cinq modalités `Femme` / `femme` / `Homme` / `hmme` / `homme` ou encore une variable `Taille` identifiée comme étant une `nominale` par le logiciel.

Cette étape est généralement considérée comme particulièrement fastidieuse : chacune des erreurs que nous venons de citer nécessite d'abord de détecter qu'il y a une erreur. Puis il faut trouver la cause de l'erreur, et enfin la corriger. Dans le cas du 3500, cela consiste à trouver la vraie valeur et remplacer la fausse par la vraie ; dans le cas de la variable mal codée, recoder `femme` en `Femme`, puis `hmme` en `Homme` et `homme` en `Homme` ; dans le cas de `Taille`, trouver la valeur qui provoque l'erreur de typage (probablement une faute de frappe, quelqu'un aura saisi une valeur non numérique comme `1.72m`, `1.72_` voir `1.72` suivit d'un espace), la corriger, puis modifier le type. Naturellement, le data management est une étape absolument essentielle car ne pas corriger les erreurs dans les données fausserait toutes les analyses subséquentes.

3.1 Détection des valeurs aberrantes

Une valeur aberrante est une valeur indiscutablement fausse : un age de 200 ans pour un être humain, une portée de 100 chatons pour une seule chatte, 100 à zéro pour un match PSG-OM. Une valeur fortement improbable n'est pas nécessairement aberrante. Par exemple, les rats mettent généralement entre 5 et 30 secondes pour traverser un labyrinthe. 1700 secondes apparait comme étant une valeur très grande, mais cela reste une valeur parfaitement possible car il arrive que les rats s'endorment 15 ou 20 minutes pendant une traversée. 1700 secondes n'est donc pas une valeur aberrante. Cette exemple illustre toute la diffuculté de la détection des valeurs aberrantes en statistique : seul l'expert du domaine (et non le statisticien) est à même de le faire.

Le meilleur moyen de repérer une valeur aberrante est de représenter graphiquement la variable concernée. L'histogramme d'une variable qui contient une valeur aberrante aura une forme particulière : toutes les valeurs seront écrasées d'un côté, le centre du graphique sera vide, et la valeur

[1]Nous utilisons ici le mot *variable* au sens statistique et non informatique : une variable est un vecteur de valeurs de même type. Une variable binaire est un vecteur de valeurs binaires (exemple : `sexe`, `avoirLeBac`), une variable numérique est un vecteur de numérique (`age`, `poids`), une varible nominale est un vecteur de mots (`groupeSanguin`, `prenom`)

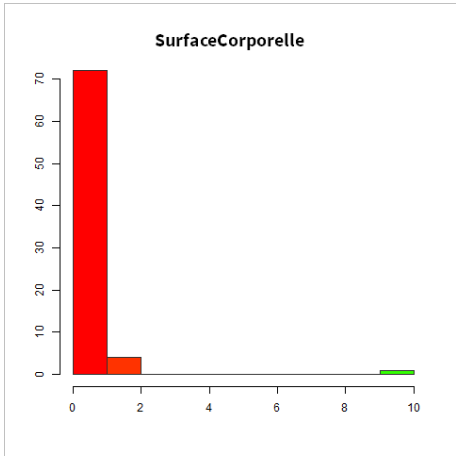

Figure 3 : Histogramme d'une variable présentant peut-être une valeur aberrante

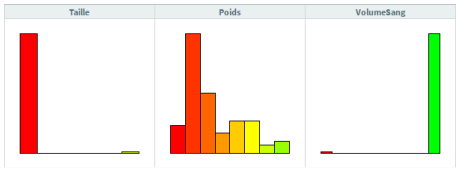

Figure 4 : Détection des valeurs aberrantes sur 3 variables

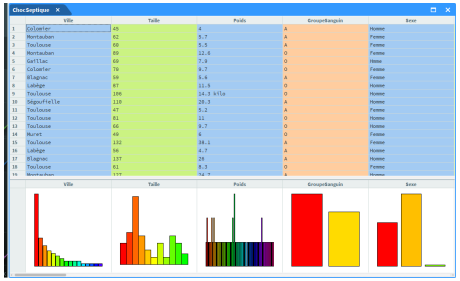

Figure 5 : Coloration des colonnes selon leur type : la première colonne est nominale (bleue) ce qui est correct pour Ville. La deuxième est numerique (verte) ce qui est correct pour Taille. La troisième est nominale (bleue) ce qui est incorrect pour Poids qui devrait être numérique (verte). La quatrième est logique (orange) ce qui est incorrect pour Groupe Sanguin qui devrait être nominale (bleue). La cinquième est nominale (bleue) ce qui est incorrect pour Sexe qui devrait être binaire (orange).

potentiellement aberrante sera de l'autre coté (variable `SurfaceCorporelle` sur la figure 3). Le problème des logiciels classiques est qu'il faut, pour chaque variable, demander à voir les graphiques un par un, pour vérifier variable par variable la présence d'une valeur aberrante.

La solution retenue lors de nos brainstormings a été de créer une fonction "graphiques en un clic" : un bouton permet à l'utilisateur d'afficher tous les graphiques de toutes les variables. Il peut ainsi d'un seul coup d'œill détecter les variables suspectes qui nécessitent un examen plus approfondi. La figure 4 donne un exemple d'utilisation. Trois variables sont représentées. Sur la première, un histogramme écrasé à gauche laisse présumer la présence d'une valeur aberrante sur la droite. La deuxième a un histogramme normal. La troisième présente un histogramme écrasé à droite, cela indique une valeur aberrante sur la gauche de l'histogramme.

3.2 Le typage incorrect

La majorité des logiciels d'analyse statistique typent les variables de manière automatique, sans demander l'avis de l'utilisateur. Les numériques sont typées en numérique, les logicals sont typées en binaire, les enum ou les string sont typées en nominale. Un typage incorrect, c'est quand le type d'une variable n'est pas le type auquel l'utilisateur s'attend. C'est généralement dû à une erreur de saisie. Par exemple, si une variable contient trois modalités `oui` / `non` / `nnon`, elle sera typée comme une nominale. L'utilisateur s'attend probablement à ce qu'elle soit typés comme une binaire car il n'a pas conscience de l'existance de la modalité nnon. Cela constitue un typage incorrect.

Les typages incorrect sont doublement problématiques : ils sont particulièrement difficiles à détecter ; Et ils produisent silencieusement des résultats faux. En effet, certaines méthodes d'analyses existent pour plusieurs types de variables, et les logiciels s'adaptent automatiquement.

Par exemple, dans une régression linéaire, on cherche à expliquer la variable numérique Y par la variable explicative X. X est généralement numérique. On cherche alors la droite de régression linéaire $y = a.x + b$ qui minimise les écarts entre Y et la droite. Mais X peut également être une variable nominale à n modalités. Le logiciel transforme alors automatiquement X en $n-1$ variables binaires X_i et cherche l'hyperplan $y = \sum_{i=1}^{n-1} a_i.x_i + b$ qui minimise les écarts entre Y et ses projections sur l'hyperplan de dimension $n-1$. Si X est une nominale, alors tout va bien. Mais si X était sensé être une numérique et se trouve être une nominale à cause d'une faute de grappe, l'utilisateur se retrouve avec un hyperplan là ou il pensait obtenir une régression toute simple. Aucun warning n'apparait, aucune mise en garde puisque c'est prévu par le logiciel. Pire, si X n'était qu'une variable d'ajustement, il peut ne pas se rendre compte qu'il y a eu erreur de typage, recodage de la variable X et que son modèle n'est absolument pas celui qu'il croit.

Une solution pour éviter ce problème est de systématiquement donner une information sur le type des variables. L'information doit être présente constamment à l'écran, faute de quoi l'utilisateur non averti ne pensera pas à aller la chercher, sans pour autant alourdir l'interface. La solution retenue a été

Figure 6 : Exemple d'étude complète

de colorier les colonnes en fonction de leur type. Les variables numériques sont en vert, les nominales en bleu, les binaires en orange. Ainsi, une variable numérique coloriée en bleu indique qu'elle est considérée comme une nominale : il y a une erreur de typage. De même, une binaire coloriée en bleu montre que la binaire a probablement plus de deux modalités. Là encore, il y a sans doute du recodage à faire. A l'inverse, une nominale en orange indique que la nominale ne comporte que deux modalités, ce qui doit a minima attirer l'attention. Un exemple est présenté figure 5 ou la coloration des types permet de repérer deux erreurs de typage et une variable "à surveiller".

4. CONCLUSION

Dans cet article, nous avons présenté la méthodologie qui nous a conduit à concevoir R++, logiciel d'anlayse statistique intégré dans une IHM conviviale.

Cette interface répond aux besoins exprimés par les utilisateurs lors de 19 sessions de prototypages. L'ensemble des résultats a été intégré dans une interface. Un chemin de fer (0) guide l'utilisateur à travers son analyse. Première étape, (1) le data management permet le nettoyage des données. (2) L'étape bivarié réalise les tests statistiques (à la fois paramétriques et non paramétriques) en choisissant automatiquement les bons tests. (3) La modélisation propose des régressions linéaires ou logisitiques, des courbes ROC, ou un modèle de COX. Enfin, (4) un module d'export permet la mise en forme des résultats et l'export facilité vers un article.

Chacune de ces étapes se fait en 2 ou 3 clics, rendant l'analyse statistique accessible à un non-statisticien.

Il reste encore de très nombreux domaines dans lequel l'analyse statistique pourrait bénéficier des apports de l'IHM. Par exemple, depuis quelques années, il est possible d'intégrer des graphiques 3D dynamiques dans du PDF. L'opération est compliquée : elle nécessite de passer par un logiciel de production de graphiques 3D dynamiques (par exemple R), puis de convertir le graphique au format Asymptote, utiliser Asymptote pour créer un .tex, et enfin utiliser LaTeX pour créer un pdf. De plus, l'ensemble de la procédure est très mal documentée, ce qui complique encore le tout. Pour ces raisons, personne n'utilise cette fonctionnalité. Une interface qui permetrait de produire simplement des graphiques 3D, de les manipuler facilement (via un device externe ?) puis de les exporter en pdf simplement pourrait être particulièrement intéressant pour les utilisateurs.

À plus long terme, l'analyse statistique nécessite l'affichage simultané d'un grand nombre d'informations. Les statisticiens ont pratiquement toujours deux écrans. Mais cela n'est pas toujours suffisant. Déporter une partie de l'interface (contrôle des données, contrôle de l'affichage des graphes 3D,...) vers un dispositif extérieur pourrait permettre d'optimiser l'affichage d'informations sur l'écran.

RÉFÉRENCES

[1] Muenchen, Robert A., The Popularity of Data Science Software, http://r4stats.com/articles/popularity/. Last accessed 19 Apr 2019

[2] Grize, Yves L. : Applications of Statistics in the Field of General Insurance : An Overview. In International Statistical Review, 83(1), 135-159 (2015)

[3] Hand, D., Statistics in Banking. Published online in Encyclopedia of Statistical Sciences, https://onlinelibrary.wiley.com/doi/10.1002/9781118445112.stat00179. Last accessed 19 Apr 2019

[4] Lewi, Paul J. : The role of statistics in the success of a pharmaceutical research laboratory : A historical case description. In Journal of Chemometrics, 19(5-7), 282-287

[5] Peterson, John J., Snee, Ronald D., McAllister, Paul R., Schofield, Timothy L. & Carella, Anthony J. : Statistics in Pharmaceutical Development and Manufacturing. In Journal of Quality Technology, 41(2), 111-134 (2009). 10.1080/00224065.2009.11917764

[6] R Core Team. : R : A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. (2019). https://www.R-project.org/.

[7] Chai, A. : Accélération des méthodes statistiques sur GPU. Master 2 internship report. https://rplusplus.com/wp-content/uploads/2018/03/Rapport_Anchen03.pdf Last accessed 19 Apr 2019.

[8] Mackinlay, Jock D., Robertson, George G., and Card, Stuart K. : The perspective wall : detail and context smoothly integrated. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91). (1991). 173-176. 10.1145/108844.108870

[9] Cockburn, A., Karlson, A., and Bederson, Benjamin B. : A review of overview+detail, zooming, and focus+context interfaces. ACM Computing Surveys. 41, 1, Article 2 (2009), 31 pages. 10.1145/1456650.1456652

[10] Bier, Eric A., Stone, Maureen C., Pier, K., Buxton, William., and DeRose, Tony D. : Toolglass and magic lenses : the see-through interface. In Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH '93). (1993). 73-80. 10.1145/166117.166126