STATISTICIENS : ATTENTION, LOGICIELS !

Dominique LADIRAY (*), Benoît QUENNEVILLE (**)

(*) Statistique Canada, Division des Données Fiscales
(**) Statistique Canada Division des Méthodes d'Enquêtes Entreprises

Tous les commentaires sont bienvenus

Introduction

Quel statisticien oserait aujourd'hui se passer d'un logiciel pour faire ses calculs, du simple tableau croisé à la régression non linéaire sous contraintes la plus complexe ? Aucun sans doute et chacun a son logiciel, son outil, préféré. Les critères de choix d'un logiciel statistique sont variés : le prix, la facilité d'apprentissage, les possibilités, l'ergonomie, la documentation, les «jolis graphiques », la maintenance, la portatibilité, les bibliothèques de « programmes utilisateurs » disponibles et réutilisables etc. Parmi toutes les qualités souhaitables, il y en a une qui paradoxalement semble peu présente à l'esprit de nombreuses personnes : la fiabilité des algorithmes et la précision des résultats.

Depuis quelques années, sous l'impulsion de statisticiens comme McCullough, Vinod, Knüsel etc., la question de la fiabilité des logiciels a refait surface et des protocoles de tests ont été élaborés pour l'évaluer.

Nous présentons dans cet article les résultats de quelques tests simples pour des logiciels réputés comme SAS, Gauss, Excel, TSP, Mathematica etc. Ces résultats montrent que les logiciels sont de qualité variable et que certains « monstres » présentent des lacunes bien surprenantes et bien inquiétantes.

La première partie de cet article concerne les tests et la méthodologie mis au point pour juger de la fiabilité des logiciels statistiques. Dans la seconde partie, les logiciels sont comparés en fonction de leurs résultats pour des opérations simples comme le calcul de moyennes, de variances, de coefficients de corrélation, d'analyse de la variance à un facteur et de régression linéaire par moindres carrés. Les résultats de tests sur la qualité des générateurs de nombres aléatoires sont aussi présentés. Dans une troisième partie, nous verrons comment les performances d'un logiciel varient dans le temps, au gré des versions successives, mais aussi dans le logiciel même, en fonction des instructions utilisées.

1. Tester la fiabilité d'un logiciel statistique

Dire que la question de la fiabilité des algorithmes statistiques ne préoccupe pas les statisticiens serait à l'évidence faux[1]. Chaque année, des dizaines d'articles sur ce thème sont publiés dans des revues spécialisées et des congrès tel COMPSTAT rassemblent des centaines de personnes. De même, de nombreux analystes de données ont vécu cette étrange expérience qui consiste à faire la même analyse sur des machines différentes, sur des logiciels différents ou sur des versions différentes du même

[1] Bien avant les ordinateurs, les statisticiens se sont appliqués à mettre au point des algorithmes simples. On peut par exemple citer les moyennes mobiles de Spencer (1904, [23]) ou les travaux précurseurs des soeurs Maballée (1925, [10]) sur les algorithmes permettant de calculer des estimateurs linéaires sans biais.

logiciel, et à obtenir des résultats différents. Un nombre respectable de ces mêmes personnes sait que le problème vient essentiellement de la représentation des nombres en machines qui entraîne des arrondis, des troncatures etc.

Mais, d'un autre côté, les statisticiens font une confiance aveugle à leur logiciel, croyant que «*les entreprises de logiciels statistiques font subir des tests intensifs à leurs produits pour s'assurer que les algorithmes mis en œuvre font des calculs précis* » (SAS Institute, [21]). Comme nous le verrons, cette profession de foi est quelque peu exagérée.

Les articles présentant et évaluant les logiciels sont assez nombreux mais font rarement allusion à la précision de ces logiciels. Ainsi, après avoir examiné de 1990 à 1997 les numéros de cinq journaux publiant régulièrement de tels articles, McCullough et Vinod ([16]) constatèrent que 3 articles seulement sur 120 s'en préoccupaient.

Pourtant les jeux d'essais existent depuis longtemps. L'un des plus fameux est sans doute celui de Longley en 1967 ([9]), jeu de données réelles présenté au tableau 1, et qui, à cause de la forte colinéarité existant entre les variables, continue à mettre à mal bon nombre de programmes de régression linéaire.

<u>Tableau 1 :</u> les données de Longley. Modèle $y_t = a_t + \sum_{i=1}^{i=6} b^i x_t^i + e_t$ **.**

Y Emploi total	x1 Déflateur du PIB	x2 PIB	x3 Chômage	x4 Effectif des forces armées	x5 Population de 14 ans et plus	x6 Année
60323	83.0	234289	2356	1590	107608	1947
61122	88.5	259426	2325	1456	108632	1948
60171	88.2	258054	3682	1616	109773	1949
61187	89.5	284599	3351	1650	110929	1950
63221	96.2	328975	2099	3099	112075	1951
63639	98.1	346999	1932	3594	113270	1952
64989	99.0	365385	1870	3547	115094	1953
63761	100.0	363112	3578	3350	116219	1954
66019	101.2	397469	2904	3048	117388	1955
67857	104.6	419180	2822	2857	118734	1956
68169	108.4	442769	2936	2798	120445	1957
66513	110.8	444546	4681	2637	121950	1958
68655	112.6	482704	3813	2552	123366	1959
69564	114.2	502601	3931	2514	125368	1960
69331	115.7	518173	4806	2572	127852	1961
70551	116.9	554894	4007	2827	130081	1962

Wilkinson ([29]) est aussi l'auteur d'une batterie de tests très simples qu'il conseille de faire subir à tout logiciel. Seuls ceux qui passent cette épreuve avec succès méritent de faire l'objet de tests plus poussés. Les six variables du tableau 2 se déduisent toutes l'une de l'autre par une simple transformation linéaire et la matrice des coefficients de corrélation linéaire doit donc être composée de 1. Tout programme en simple précision aura le plus grand mal à passer avec succès cette épreuve somme toute relativement banale.

<u>Tableau 2 :</u> le « vilain fichier» de Wilkinson.

X	Big	Little	Huge	Tiny	Round
1	99999991	0.99999991	1E12	1E-12	0.5
2	99999992	0.99999992	2E12	2E-12	1.5
3	99999993	0.99999993	3E12	3E-12	2.5
4	99999994	0.99999994	4E12	4E-12	3.5
5	99999995	0.99999995	5E12	5E-12	4.5
6	99999996	0.99999996	6E12	6E-12	5.5
7	99999997	0.99999997	7E12	7E-12	6.5
8	99999998	0.99999998	8E12	8E-12	7.5
9	99999999	0.99999999	9E12	9E-12	8.5

SPAD 5.0, par exemple, a des difficultés à lire la variable *Little*. En ramenant le nombre de chiffres 9 dans chaque valeur à six au lieu des sept initiaux, SPAD accepte le fichier mais calcule une matrice des corrélations bien surprenante (tableau 3) pour un logiciel d'analyse factorielle [2] !

Tableau 3 : La matrice des corrélations calculée par SPAD sur le jeu d'essai de Wilkinson, modifié pour la variable *Little*.

	X	Big	Little modifiée	Huge	Tiny	Round
X	1.00					
Big	0.69	1.00				
Little modifiée	1.15	0.79	1.00			
Huge	1.00	0.69	1.15	1.00		
Tiny	1.00	0.69	1.15	1.00	1.00	
Round	1.00	0.69	1.15	1.00	1.00	1.00

Si des jeux d'essai existent, il n'est pas toujours simple de les trouver et de les mettre en œuvre. En 1998, McCullough ([13], [14]) a proposé une méthodologie pour tester la fiabilité des logiciels statistiques, en insistant sur 3 aspects essentiels :

- Les estimations, en utilisant les jeux d'essai du *National Institute of Standards and Technology* (NIST, [20]) pour mesurer la précision des procédures de statistique descriptive univariée, d'analyse de la variance à un facteur, de régression linéaire et non linéaire ;
- La génération de nombres aléatoires, en utilisant la batterie de tests DIEHARD mise au point par Marsaglia ([11]) ;
- Les distributions statistiques, qui permettent le calcul des valeurs critiques des tests et des p-values, en utilisant les programmes ELV (Knüsel, [7]) ou DCDFLIB (Brown, [2]). Knüsel est l'auteur de nombreuses études (voir par exemple [5], [6]) sur les performances des logiciels en la matière. Ses conclusions, que nous ne détaillerons pas dans cette présentation, sont en général assez négatives.

1.1. Les jeux d'essais du NIST

Le NIST a mis à disposition du public un ensemble de jeux d'essai de référence, regroupés sous l'acronyme StRD (*Statistical Reference Datasets*, [20]), et divisés en quatre blocs : l'analyse descriptive univariée, l'analyse de la variance à un facteur, la régression linéaire par moindres carrés et la régression non linéaire. Chaque problème a été classé selon son niveau de difficulté - facile, moyen et difficile – puis résolu avec un grand degré de précision.

- La suite de tests pour l'analyse descriptive univariée est composée de 9 jeux de données contenant de 3 à 5000 observations : six faciles, deux de difficulté moyenne et un difficile.
- La suite de tests pour l'analyse de la variance à un facteur est composée de 11 jeux de données dont 2 sont issus de données réelles et 9 sont des problèmes mis au point par Simon et Lesage ([22]). Quatre problèmes sont réputés faciles, trois de difficulté moyenne et trois difficiles.
- La suite de tests pour la régression linéaire est composée de 11 jeux de données contenant de 3 à 82 observations : deux faciles, deux de difficulté moyenne et sept difficiles. Outre les données de Longley, cette suite contient aussi des données simulées proposées par Wampler ([27], [28]).
- La suite de tests pour la régression non linéaire contient 27 jeux de données de 6 à 250 observations et impliquant l'estimation de 2 à 9 paramètres : huit faciles, onze de difficulté moyenne et huit difficiles. Le NIST fournit aussi deux ensembles de valeurs pour initialiser les algorithmes itératifs : les valeurs « Start I » sont assez loin des solutions et les valeurs « Start II » au contraire très proches.

[2] Essayez aussi de compléter la matrice en dupliquant simplement les variables *Little* (modifiée) et *Big*. La diagonale de la matrice reste égale à 1 mais le coefficient de corrélation de *Little* (respectivement *Big*) avec elle-même est « égal » à 1.34 (respectivement 0.64) !

Pour les problèmes linéaires, les calculs ont ainsi été faits, en utilisant le FORTRAN Bailey (compilateur et routines disponibles sur NETLIB), et avec une précision de 500 décimales. Les résultats ont ensuite été arrondis pour assurer une précision de 15 chiffres significatifs. Pour les problèmes de régression non linéaire, les calculs ont été faits en quadruple précision et en utilisant deux programmes du domaine public, différents algorithmes et différentes plateformes. Les résultats ont ensuite été arrondis à 11 chiffres significatifs.
La base de données StDR contient donc les problèmes et les solutions certifiées par le NIST. Le tableau 4 donne par exemple les solutions certifiées pour les données de Longley.

<u>Tableau 4</u> : Valeurs certifiées du problème de Longley (voir données au tableau 1).

Estimation des paramètres :

Paramètre	Valeur	Ecart-type
B0	-3482258.63459582	890420.383607373
B1	15.0618722713733	84.9149257747669
B2	-0.358191792925910E-01	0.334910077722432E-01
B3	-2.02022980381683	0.488399681651699
B4	-1.03322686717359	0.214274163161675
B5	-0.511041056535807E-01	0.226073200069370
B6	1829.15146461355	455.478499142212
Ecart-type des résidus	304.854073561965	
R2	0.995479004577296	

Tableau d'analyse de la variance:

	Degrés de liberté	Somme des carrés	Moyenne des carrés	Statistique F
Régression	6	184172401.944494	30695400.3240823	330.285339234588
Résidus	9	836424.055505915	92936.0061673238	

Comme d'un problème à l'autre les valeurs des paramètres peuvent être très différentes, McCullough ([13]) propose d'utiliser comme indicateur de fiabilité le logarithme en base 10 de l'erreur relative (LRE) ou absolue (LAR) :

$$LRE = l = -\log_{10}\left[|q-c|/|c|\right] \text{ si c est non nul et } LAR = -\log_{10}\left[|q|\right] \text{ sinon.}$$

Dans ces expressions, q est la valeur estimée et c la valeur certifiée. Par exemple, le R2 du problème de Longley calculé par la procédure REG de SAS 8.2 est 0.995479004577370. Le LRE associé sera donc :

$l = -\log_{10}\left[|0.995479004577370 - 0.995479004577296|/|0.995479004577296|\right] = 13.1285$, ce qui correspond bien à 13 chiffres corrects.

Pour que le LRE soit un bon indicateur du nombre de chiffres significatifs exacts, il faut que les valeurs q et c soient « proches ». En effet, si par exemple q=20 et c=1, alors LRE=1.28 bien que la valeur calculée soit très différente.

1.2. Une remarque importante

Les sociétés développant ou commercialisant les logiciels statistiques ont souvent relevé, et critiqué, le caractère « pathologique » des jeux d'essais utilisés pour apprécier la fiabilité des logiciels. Même si cette remarque est de bonne guerre, il est facile de justifier la nature de ces tests :

- En général, on ne teste pas la fiabilité d'une voiture, la résistance d'un matériau ou de tout autre produit, dans des conditions agréables ! Les normes de sécurité sont souvent sévères et reposent sur un « principe de précaution » légitime.

- Les jeux d'essai sont de difficulté graduée et certains d'entre eux correspondent à des données réelles, à des cas rencontrés dans la pratique. Les cas de multi colinéarité, illustrés par le jeu d'essai de Longley, sont courants dans l'estimation de modèles économétriques. Il existe d'ailleurs de nombreux diagnostics permettant de les repérer.
- A l'évidence, ces jeux d'essai ne seraient d'aucun intérêt si tous les logiciels les résolvaient aisément ou si aucun d'entre eux ne le faisait. Les résultats, que nous allons commenter dans la partie suivante, montrent clairement que certains logiciels pourraient améliorer leurs algorithmes.
- Enfin, même improbables, ces situations « extrêmes » peuvent fort bien se produire, soit lors d'une étude par simulation, de type Monte Carlo, Bootstrap etc., ou dans le cadre de la résolution itérative d'un problème non linéaire.

1.3. Tester un générateur de nombres aléatoires

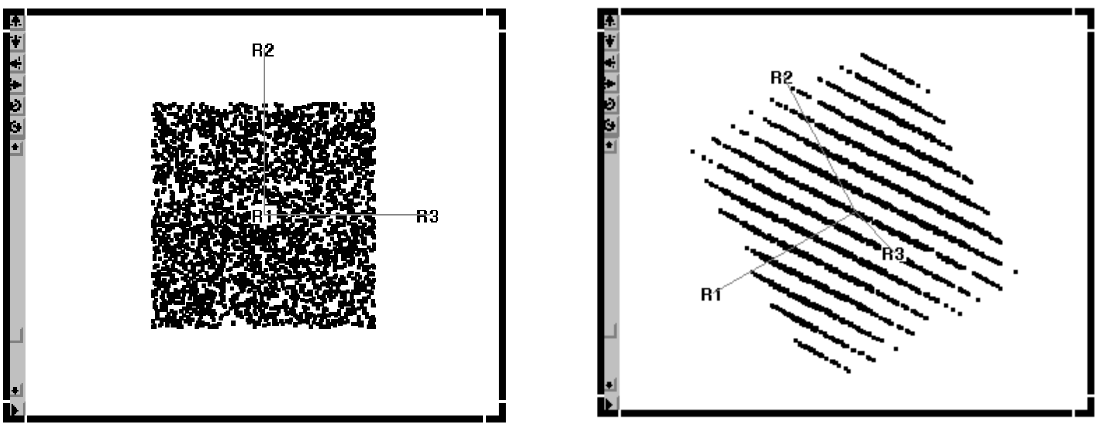
Les générateurs de nombres aléatoires ont pris depuis plusieurs années une place très importante dans la recherche en statistique et en économétrie. Les méthodes de ré-échantillonage comme le Bootstrap, les études par simulation, les méthodes de validation croisée sont de plus en plus utilisées et reposent par nature sur l'utilisation d'un bon générateur de nombres aléatoires. La quasi-totalité des générateurs de nombres uniformes actuellement disponibles dans les logiciels statistiques sont des générateurs à congruence linéaire (LCG) pour lesquels les valeurs produites reposent sur une équation du type :

$$X_n \equiv aX_{n-1} + c \pmod m$$

Un générateur LCG est donc défini par 2 constantes, a et c, un module m et une valeur initiale X_0. Le choix de c et m détermine la période p du générateur, c'est-à-dire le nombre d'appels maximal que vous pouvez faire avant que le générateur ne commence à se répéter. Cette période p doit évidemment être la plus grande possible tant les nouvelles méthodes statistiques sont gourmandes en nombres aléatoires. Ainsi l'étude de McKinnon (citée par McCullough [13]) sur les tests de racine unité (McKinnon, [18]) a nécessité plus de 100 milliards de nombres aléatoires. Knuth ([8]) estime que le nombre maximal d'appels au générateur ne devrait pas excéder *p/1000*, ce qui dans ce cas, aurait nécessité un générateur de période supérieure à 2^{46}.

Par nature, les LCG produisent des séquences de nombres qui s'avèrent corrélées dans des espaces de dimension k plus ou moins élevée selon le générateur (Marsaglia, [12]). Ce phénomène est illustré par le graphique 1 où sont représentées, en dimension 2 et 3, des séquences de nombres générés par le vieux (et déplorable) générateur RANDU qui équipait tous les gros systèmes IBM 360.

Figure 1 : Séquences de nombres générés par RANDU, en dimension 2 (graphique de gauche) et 3 (graphique de droite). En principe, le cube devrait être « rempli ».



Tester un générateur de nombres aléatoires n'est pas chose facile. Heureusement, il suffit en général de tester les générateurs produisant des nombres distribués uniformément entre 0 et 1 dans la mesure où la plupart des autres lois s'en déduisent. L'une des difficultés tient au fait que les créateurs et distributeurs de logiciels sont curieusement très discrets sur le type de générateur utilisé. Ainsi, SAS ([21]) donne le module de son générateur RANUNI (2^{31}-1) et cite un article de Fisher et Moore (1982), sans donner plus de détail. De même, la documentation en ligne de S-plus fait référence à un générateur « Super-Duper modifié » sans en dire plus.

En 1996, Marsaglia ([11]) a mis au point le programme DIEHARD qui vérifie le caractère aléatoire d'une séquence de nombres, à partir d'une batterie de 18 tests statistiques. Ces tests, et leur interprétation, sont décrits en détail dans la documentation du programme ; certains d'entre eux sont aussi décrits dans Knuth ([8]).

2. Comparaison des performances des logiciels

2.1. Sources et méthodologie

Cette partie présente les résultats obtenus par de grands logiciels aux tests du NIST et DIEHARD. Les sources sont nombreuses : articles publiés dans des revues, résultats trouvés sur les sites même des logiciels, résultats de tests menés par les élèves de l'ENSAE dans le cadre du cours «Logiciels Statistiques », propres calculs des auteurs et de leurs amis etc.

Cette diversité des sources est en soi une faiblesse. En effet, la machine sur laquelle vous effectuez les tests, et en particulier son processeur, a une influence sur les résultats numériques mêmes. Néanmoins, la quasi totalité de ces tests a été réalisée sur des PC à processeur Pentium et les vérifications que nous avons faites conduisent à des résultats cohérents, à la première décimale près.

Certains tests sont encore en cours, notamment pour prendre en compte les versions les plus récentes des logiciels.

Les logiciels testés et les sources sont les suivants :

- Excel 2000 et XP : les tests ont été faits par McCullough et Wilson ([17]) et vérifiés par nous mêmes.
- Stata 6 : les résultats sont disponibles sur le site de Stata ([24]).
- SAS 6.12 et 8.2 : les tests ont été faits par nous-mêmes. SAS donne aussi quelques résultats, cohérents avec nos calculs, sur son site ([21]).
- JMP 5.0 : les tests sont disponibles sur le site de JMP ([3]). L'article de Altman ([1]) donnant des résultats pour une version antérieure est fortement contesté par JMP (avec raison selon nos propres vérifications).
- TSP 4.4 : les tests sont disponibles sur le site de TSP ([25]).
- Mathematica 4 : les tests ont été faits par McCullough ([15]), Nerlove ([19]) et vérifiés par nous-mêmes.
- Gauss 3.2.37 : les tests ont été faits par Vinod ([26]) et nous sommes en train de les faire tourner sur une version plus récente.
- S-plus 4.0 : les tests ont été faits par McCullough ([14]) et nous sommes en train de les faire tourner sur une version plus récente.
- SPSS 7.5 : les tests ont été faits par McCullough ([14]) et nous sommes en train de les faire tourner sur une version plus récente.

Le nombre de résultats générés par les problèmes du NIST, à traiter et à commenter est très important. Par exemple, chaque problème de régression produit un tableau d'analyse de la variance et les estimations et écart-types des paramètres. Nous nous sommes donc restreints à quelques statistiques de base pour lesquelles le nombre de chiffres significatifs exacts a été calculé.

Plus précisément :

- Pour les problèmes de statistique univariée, la moyenne, l'écart-type et le coefficient d'autocorrélation d'ordre 1 ont été conservés ;
- Pour les problèmes d'analyse de la variance, qui sont des cas particuliers de régression linéaire, nous nous sommes concentrés sur la statistique de Fisher ;
- Pour les problèmes de régression linéaire, nous nous sommes intéressés à l'estimation des paramètres et à la précision de ces estimations. Dans ce cas, nous avons reporté dans les tableaux le LRE obtenu pour l'estimation la moins précise.
- Le tableau sur les problèmes non linéaires présente le LRE obtenu pour l'estimation la moins précise des paramètres du modèle.

Tableau 5: Résultats des différents logiciels sur les problèmes univariés.

Moyenne

Données	Pidigits	Lottery	Lew	Mavro	Michelso	NumAcc1	NumAcc2	NumAcc3	NumAcc4
	Facile	Facile	Facile	Facile	Facile	Facile	Moyen	Moyen	Difficile
Excel XP	15.0	15.0	15.0	15.0	15.0	15.0	14.0	15.0	14.0
Gauss 3.2.37	15.0	15.0	15.0	15.0	15.0	15.0	14.0	15.0	14.0
JMP 5.0	15.0	15.0	15.0	15.0	15.0	15.0	14.0	15.0	15.0
Math 4 (A)	15.0	15.0	15.0	15.0	15.0	15.0	14.0	15.0	14.0
Math 4 (B)	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
SAS 8.2	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
S-Plus 4.0	15.0	15.0	15.0	15.0	15.0	15.0	14.0	15.0	14.0
SPSS 7.5	14.7	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
Stata 6	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
TSP 4.4	15.0	15.0	15.0	15.0	15.0	15.0	14.0	15.0	15.0

Ecart-type

Données	Pidigits	Lottery	Lew	Mavro	Michelso	NumAcc1	NumAcc2	NumAcc3	NumAcc4
	Facile	Facile	Facile	Facile	Facile	Facile	Moyen	Moyen	Difficile
Excel XP	15.0	15.0	15.0	9.4	8.3	15.0	11.6	1.1	0.0
Gauss 3.2.37	15.0	15.0	15.0	13.1	13.8	15.0	15.0	9.5	8.3
JMP 5.0	15.0	15.0	15.0	13.1	15.0	15.0	14.6	9.5	8.3
Math 4 (A)	15.0	15.0	15.0	13.1	13.8	15.0	14.0	9.5	8.3
Math 4 (B)	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
SAS 8.2	15.0	15.0	15.0	13.1	13.8	15.0	14.2	9.5	8.3
S-Plus 4.0	15.0	15.0	15.0	13.1	13.8	15.0	15.0	9.5	8.3
SPSS 7.5	15.0	15.0	13.2	12.1	12.4	15.0	15.0	9.5	8.3
Stata 6	15.0	15.0	15.0	13.1	13.8	15.0	15.0	9.5	8.3
TSP 4.4	15.0	15.0	15.0	13.1	13.8	15.0	14.6	9.5	8.3

Coefficient d'autocorrélation

Données	Pidigits	Lottery	Lew	Mavro	Michelso	NumAcc1	NumAcc2	NumAcc3	NumAcc4
	Facile	Facile	Facile	Facile	Facile	Facile	Moyen	Moyen	Difficile
Excel XP	4.0	2.1	2.6	1.8	3.6	0.0	3.3	3.3	3.3
Gauss 3.2.37	15.0	15.0	14.8	13.7	13.4	15.0	15.0	11.2	9.0
JMP 5.0	13.0	15.0	15.0	13.8	15.0	15.0	13.7	11.2	9.0
Math 4 (A)	15.0	14.9	14.8	13.7	13.4	15.0	13.7	11.2	9.0
Math 4 (B)	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
SAS 8.2	15.0	14.9	14.8	13.8	13.4	ns	15.0	11.9	10.7
S-Plus 4.0	6.8	7.4	7.0	7.1	7.3	15.0	7.1	7.1	7.3
SPSS 7.5	0	3.4	3.0	4.9	3.4	ns	15.0	15.0	15.0
Stata 6	14.9	15.0	14.8	13.7	13.4	15.0	15.0	11.9	10.7
TSP 4.4	13.0	15.0	15.0	13.8	13.4	15.0	13.7	11.2	9.0

2.2. Performances des logiciels sur les problèmes du NIST

Tout d'abord, une nouvelle rassurante : un logiciel réussit le score parfait sur les 58 problèmes du NIST. Il s'agit de Mathematica (nous avons testé la version 4.0), à condition d'utiliser deux options du logiciels qui permettent d'améliorer assez nettement la précision des calculs :

- La commande « $MinPrecision = *n* » où *n* est le nombre de chiffres significatifs souhaité,
- La commande « Rationalize » qui permet de distinguer entre des nombres réels et des nombres rationnels.

Dans les tableaux, Mathematica apparaîtra donc sous deux formes : (A) avec les options par défaut et (B) avec les options ci-dessus.

2.2.1. Les statistiques univariées

Les résultats sont présentés dans le tableau 5. Tous les logiciels testés ont passé avec succès l'épreuve du calcul de la matrice des corrélations du « vilain fichier » de Wilkinson (tableau 2).

- Tous les logiciels testés calculent correctement les moyennes.
- Excel XP a quelques problèmes sur le calcul des écart-types, les autres logiciels réussissant des performances très voisines.
- Pour le calcul des coefficients d'autocorrélation, les choses se gâtent un peu. Les mauvais résultats affichés par Excel XP pourraient venir d'une définition différente du coefficient d'autocorrélation. Ce n'est pas le cas pour SPlus et SPSS qui obtiennent des résultats vraiment « très moyens ». La réponse de Splus a ces problèmes est assez amusante : « *The article tested the S-PLUS autocorrelation procedure "acf", which uses single precision and produces about 7 digits of accuracy. The correlation procedure "cor" in S-PLUS 4.0 and later uses double-precision and extremely accurate algorithms*. ». En d'autres termes, si vous voulez calculer une autocorrélation, n'utilisez pas la commande qui calcule ces autocorrélations !

2.2.2. Analyse de la variance

Les résultats, présentés dans le tableau 6, montrent une plus grande diversité dans la qualité des logiciels. Excel et SPlus ne réussissent à traiter que les problèmes faciles, les autres logiciels ayant des difficultés avec les 3 problèmes les plus difficiles de Simon et Lesage. SAS et Stata ont des performances légèrement supérieures à celles des autres logiciels.

Néanmoins, si on compare ces performances avec celles, parfaites, de Mathematica, on est amené à conclure que les algorithmes mis en œuvre par tous ces logiciels sont de qualité bien insuffisante.

2.2.3. Régression linéaire

Dans ce domaine (voir tableau 7), la surprise vient du logiciel Gauss qui, pour un logiciel prisé par les économètres, réussit des performances assez médiocres, inférieures à celles de Excel !

Les logiciels ont tous, à l'exception de SPlus, de grosses difficultés à traiter le jeu de données Filippelli qui présente un cas de très forte colinéarité. Il est étonnant de constater une telle variété dans les résultats, variété qui traduit celle des algorithmes utilisés. De façon générale, si on compare encore les résultats obtenus à ceux de Mathematica, force est de constater que les développeurs ont encore bien des progrès à faire.

2.2.4. Régression non linéaire

Les problèmes non linéaires du NIST sont les plus difficiles à résoudre et les plus sujets à controverse. Il existe en effet souvent plusieurs procédures disponibles dans les logiciels pour les traiter et chaque procédure possède en général une multitude d'options. Il est ainsi possible de discuter les résultats obtenus en invoquant, dans tel cas particulier, l'utilisation de telle option ou de telle valeur du paramètre. Les résultats des tests sont présentés dans le tableau 8. La Figure 2 en donne une présentation plus synthétique.
Ce graphique traduit les performances médiocres de Gauss et, au contraire, le bon score de JMP, un logiciel d'analyse exploratoire a priori peu spécialisé dans ces problèmes.

Tableau 6 : **Résultats des différents logiciels sur les problèmes d'analyse de la variance (statistique de Fisher).**

Données	Logiciel	Excel XP	GAUSS 3.2.37	JMP 5.0	Mathematica 4.0 (B)	SAS 8.2	S-Plus 4.0	SPSS 7.5	Stata 6	TSP 4.4
SiRstv	Facile	8.5	12.4	12.4	15.0	12.7	13.3	9.6	13.1	13.1
SmLs01	Facile	14.3	14.5	14.0	15.0	15.0	14.5	15.0	14.4	14.6
SmLs02	Facile	12.5	14.1	13.4	15.0	13.9	14.3	15.0	13.3	14.7
SmLs03	Facile	12.6	12.7	12.4	15.0	12.7	12.9	12.7	14.7	12.3
AtmWtAg	Moyen	1.8	8.5	8.4	15.0	8.8	9.7	miss	10.2	10.2
SmLs04	Moyen	1.7	8.5	8.2	15.0	10.4	10.4	0.0	10.4	10.4
SmLs05	Moyen	1.1	8.3	8.0	15.0	10.2	10.2	0.0	10.2	10.2
SmLs06	Moyen	0	6.5	6.2	15.0	10.2	10.2	0.0	10.2	10.2
SmLs07	Difficile	0	2.7	2.4	15.0	4.4	4.6	0.0	4.4	4.6
SmLs08	Difficile	0	2.2	1.9	15.0	4.2	2.7	0.0	4.4	1.9
SmLs09	Difficile	0	0	0.3	15.0	4.2	0.0	0.0	4.2	0.8

miss : résultat mis à valeur manquante

Tableau 7 : **Résultats des différents logiciels sur les problèmes de régression linéaire.**

Données	Logiciel	Excel XP		Gauss 3.2.37		JMP 5.0		Mathematica 4.0 (A)		Mathematica 4.0 (B)		SAS 8.2		S-Plus 4.0		SPSS 7.5		Stata 6		TSP 4.4	
		$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$
Norris	Facile	12.1	13.8	12.2	10.5	12.2	11.7	12.7	13.9	15.0	15.0	15.0	11.8	12.5	14.1	12.3	10.2	12.8	13.5	12.2	14.2
Pontius	Facile	11.2	14.3	11.6	7.9	11.2	8.4	5.3	5.3	15.0	15.0	15.0	8.6	12.7	13.2	12.5	8.9	11.5	13.0	11.9	12.7
NoInt1	Moyen	14.7	15.0	14.7	13.4	14.7	13.5	15.0	15.0	15.0	15.0	15.0	14.0	14.7	14.4	14.7	12.5	14.7	15.0	14.7	14.8
NoInt2	Moyen	15.0	15.0	15.0	14.3	15.0	14.6	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	14.3	15.0	15.0	15.0	14.9
Filippelli	Difficile	0	0	0	0	ns	ns	ns	ns	15.0	15.0	2.2	0.0	7.1	7.0	ns	ns	ns	ns	0.0	0.0
Longley	Difficile	7.4	8.6	8.5	10.0	ns	ns	11.6	12.1	15.0	15.0	13.1	10.3	13.0	14.2	12.1	13.3	12.1	12.9	11.7	12.4
Wampler1	Difficile	6.6	7.2	6.1	0	8.0	15.0	9.2	15.0	15.0	15.0	15.0	15.0	9.8	15.0	6.6	6.6	6.9	15.0	9.2	15.0
Wampler2	Difficile	9.7	11.8	9.4	4.4	10.6	15.0	10.1	15.0	15.0	15.0	15.0	15.0	13.5	15.0	9.7	9.7	9.7	15.0	12.5	15.0
Wampler3	Difficile	6.6	11.2	6.1	6.3	8.0	10.8	6.5	10.1	15.0	15.0	15.0	11.2	9.2	13.5	7.4	10.6	6.5	10.8	9.0	13.5
Wampler4	Difficile	6.6	11.2	6.1	10.1	8.0	10.9	4.5	10.1	15.0	15.0	15.0	11.2	7.5	13.6	7.4	10.8	6.5	10.8	8.9	13.8
Wampler5	Difficile	6.6	11.2	6.1	10.5	8.0	10.9	2.5	10.1	15.0	15.0	13.7	11.2	5.5	13.5	5.8	10.8	6.4	10.8	7.3	13.8

ns : pas de solution

<u>Tableau 8</u> : Résultats des différents logiciels sur les problèmes de régression non linéaire.

Dataset	Logiciel	Excel XP	Gauss 3.2.37	JMP 5.0	Mathematica 4 (A)	Mathematica 4 (B)	SAS 6.12	S-Plus 4.0	SPSS 7.5	Stata 6	TSP 4.4
Misra1a	Facile	4.8	7.4	11.0	10.1	11.0	9.2	9.3	6.1	9.1	9.5
Chwirut2	Facile	4.6	5.0	10.9	10.3	11.0	7.6	7.6	7.5	7.9	8.6
Chwirut1	Facile	4.9	5.6	9.9	8.8	11.0	8.6	7.3	7.1	7.6	10.6
Lanczos3	Facile	0.0	3.2	10.5	9.8	11.0	6.7	6.6	6.9	6.2	8.3
Gauss1	Facile	0.0	8.8	10.6	10.6	11.0	8.7	8.7	7.4	8.6	8.8
Gauss2	Facile	0.0	9.0	10.3	10.0	11.0	8.4	8.4	7.4	8.2	10.3
DanWood	Facile	5.5	7.9	11.0	10.0	11.0	10.1	8.0	9.5	8.6	10.2
Misra1b	Facile	4.4	8.5	11.0	9.5	11.0	10.1	9.3	6.7	9.3	10.9
Kirby2	Moyen	1.1	0.0	9.9	8.5	11.0	7.5	7.4	7.7	8.1	10.6
Hahn1	Moyen	0.0	0.0	10.8	7.2	11.0	7.8	7.6	5.4	7.1	8.6
Nelson	Moyen	1.3	0.0	10.1	8.1	11.0	7.1	7.6	6.5	7.1	8.9
MGH17	Moyen	0.0	0.0	10.8	9.6	11.0	8.8 *	7.9	7.6	9.4 *	10.3 *
Lanczos1	Moyen	0.0	0.0	10.6	10.6	11.0	10.7	10.6	9.6	10.6	10.6
Lanczos2	Moyen	0.0	3.2	10.4	10.4	11.0	10.3	10.3	8.7	7.4	10.4
Gauss3	Moyen	0.0	8.2	10.5	9.2	11.0	9.2	9.2	7.6	8.2	9.2
Misra1c	Moyen	4.6	7.2	10.8	9.1	11.0	10.5	8.1	5.9	9.2	10.8
Misra1d	Moyen	5.3	3.0	11.0	9.2	11.0	8.7	9.4	6.1	9.3	11.0
Roszman1	Moyen	3.7	7.1	10.9	8.1	11.0	8.6	7.0	6.6	7.9	7.5
ENSO	Moyen	3.4	5.8	10.7	6.6	11.0	7.1	5.6	0.0	4.7	7.4
MGH09	Difficile	0.0	5.3	7.5 *	7.7	11.0	6.5 *	6.7	7.6	7.0 *	8.1 *
Thurber	Difficile	1.8	6.0	10.4	7.9	11.0	6.4	6.9	8.2	6.5	7.8
BoxBod	Difficile	0.0	8.2	9.8 *	11.0	11.0	7.1 *	7.8	6.9	7.3	8.4
Rat42	Difficile	5.3	0.0	11.0	9.7	11.0	8.3	7.6	6.8	7.6	8.8
MGH10	Difficile	0.0	ns	10.9 *	8.9	11.0	0.0	10.3	7.1	7.5 *	10.9 *
Eckerle4	Difficile	0.0	9.1	10.4 *	9.6	11.0	8.3 *	9.2	9.9	8.3 *	9.9 *
Rat43	Difficile	0.0	6.7	11.0 *	8.7	11.0	0.0	8.2	8.8	6.0 *	7.4
Bennett5	Difficile	0.0	ns	11.0	11.0	11.0	0.0	10.3	9.9	6.3	11.0

* : solution obtenue à partir de Start II.

Figure 2 : Représentation des résultats des différents logiciels aux tests de régression non linéaire (boxplots des 27 LRE pour chaque logiciel).

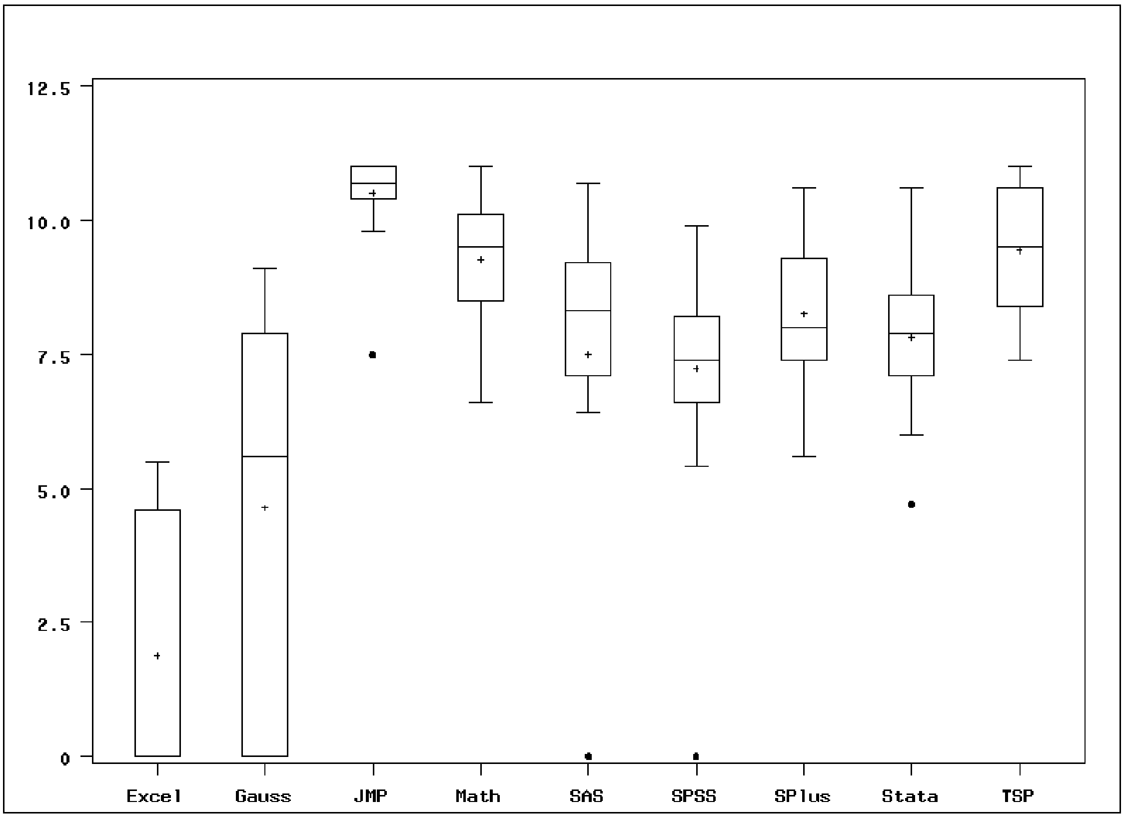


Tableau 9 : Résultats des différents générateurs aux tests de Marsaglia (DIEHARD).

p : pass, F : fail

Test	Excel	GAUSS	JMP	Mathematica	SAS	S-Plus	SPSS
Birthday Spacings Test	p	F	p	F	p	p	p
Overlapping 5-Permutation Test	p	F	p	p	p	p	p
Binary Rank For 31x31 Matrices	p	F	p	p	p	p	p
Binary Rank For 32x32 Matrices	p	F	p	p	p	p	p
Binary Rank For 6x8 Matrices	p	F	p	p	p	p	p
Bitstream Test (p values)	p	F	p	p	p	p	p
OPSO Test	F	p	p	p	p	p	p
OQSO Test	F	p	p	p	p	F	p
DNA Test	F	p	p	p	p	F	p
Count the Ones Test (stream of bytes)	F	p	p	p	F	F	F
Count the Ones Test (specific byte)	F	F	p	p	p	F	p
Parking Lot Test	p	F	p	p	p	p	p
Minimum Distance Test	p	F	p	p	p	p	p
3-D Spheres Test	F	F	p	p	p	p	p
Squeeze Test	F	p	p	p	p	p	p
Overlapping Sums Test	p	p	p	p	p	p	p
Runs Test	p	p	p	p	p	p	p
Craps Test	p	p	p	p	p	p	p

2.3. Les générateurs de nombres aléatoires

Les résultats des différents générateurs de nombres aléatoires, uniformément répartis entre 0 et 1, sont présentés dans le tableau 9. Ces tests sont plutôt destinés à des générateurs ayant une période supérieure ou égale à 2^{32}, mais un bon générateur de période 2^{31}-1, générateurs habituellement disponibles dans les logiciels, devrait les passer avec succès.

Seul le générateur de JMP réussit le score parfait. Mathematica, SAS, SPSS ont des générateurs corrects qui pourraient néanmoins être améliorés. Les générateurs de Gauss et Excel sont simplement obsolètes.

JMP ([4]) utilise depuis sa version 4.0.5 un algorithme de Mersenne-Twister, de période 2^{19937}-1, manifestement supérieur aux autres RNG. Ce qui est surprenant, c'est que ces générateurs sont en général du domaine public et on comprend mal que les logiciels ne soient pas, en la matière, à la pointe du progrès.

3. Variations autour d'un même logiciel

Même si vous choisissez un « bon » logiciel, vous n'êtes pas à l'abri de certaines surprises. En effet, il existe souvent plusieurs commandes permettant de calculer une moyenne, une variance, une droite de régression etc. Et en général les algorithmes, et donc la précision des estimations, varient d'une commande à l'autre. Par ailleurs, les algorithmes peuvent évoluer d'une version à l'autre d'un logiciel ; le plus souvent, ils sont améliorés ….. mais pas toujours !

3.1. Différentes commandes, différentes qualités

Il est ainsi surprenant de constater que des algorithmes simples, calcul de variance ou de droite de régression par exemples, peuvent varier assez fortement selon la commande que vous utilisez. C'est le cas, dans la version SAS 8.2, pour les procédures MEANS et UNIVARIATE. Le tableau 10 illustre cette différence de précision.

<u>Tableau 10</u> : Précision du calcul de l'écart-type dans les procédures MEANS et UNIVARIATE de SAS 8.2

		PROC MEANS	PROC UNIVARIATE
PiDigits	facile	15.0	15.0
Lottery	facile	15.0	15.0
Lew	facile	15.0	15.0
Mavro	facile	13.1	12.8
Michelso	moyen	13.8	12.4
NumAcc1	moyen	15.0	15.0
NumAcc2	moyen	14.2	15.0
NumAcc3	moyen	9.5	9.4
NumAcc4	difficile	8.3	8.3

Le tableau 11 illustre le même phénomène, pour la régression linéaire dans SAS 8.2, en comparant les procédures REG, ORTHOREG et IML. La procédure ORTHOREG est sensée être plus adaptée dans les cas où les données sont « difficiles », par exemple dans le cas de forte colinéarité. De fait, la précision est globalement meilleure sur les problèmes du NIST.

Dans le module IML, vous pouvez directement programmer les résultats d'une régression linéaire. Usuellement le statisticien ou économètre va directement traduire les formules mathématiques, en écrivant, par exemple pour les estimations des paramètres du modèle :

```
beta= INV(T(x)*x)*T(x)*y;
```

Comme le montrent les résultats du tableau 11, les résultats obtenus seront de précision moindre. C'est une règle générale : l'algorithmique est un métier et il est pratiquement impossible d'écrire sans formation un bon algorithme. En particulier, l'utilisation directe d'une formule mathématique pour calculer une expression est très rarement une manière efficace de procéder.
De même Nerlove ([19]) constate que le module statistique développé par NAG pour compléter Excel (les fameux « add ins »), ne donne pas des résultats bien meilleurs qu'Excel lui même. Vinod ([26]) fait la même remarque pour Gauss.

<u>Tableau 11</u> : Précision de la régression linéaire dans les procédures REG, ORTHOREG et IML de SAS 8.2

Données		REG		ORTHOREG		IML	
		$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$	$l_{\hat{b}}$	$l_{\hat{s}}$
Norris	facile	12.3	11.8	11.9	13.8	12.5	14.1
Pontius	facile	11.5	8.6	12.1	12.3	10.6	13.4
NoInt1	moyen	14.7	14.0	14.7	15.0	14.7	15.0
NoInt2	moyen	15.0	15.0	15.0	14.6	15.0	15.0
Filip	difficile	0.0	0.0	0.0	0.0	ns	ns
Longley	difficile	8.6	10.3	13.6	14.6	7.0	10.2
Wampler1	difficile	6.6	15.0	10.2	15.0	4.8	5.2
Wampler2	difficile	9.6	15.0	13.2	15.0	8.6	9.9
Wampler3	difficile	6.6	11.2	9.8	13.6	4.8	10.7
Wampler4	difficile	6.6	11.2	8.1	13.6	4.8	10.7
Wampler5	difficile	6.6	11.2	6.1	13.6	4.8	10.7

3.2. Versions différentes, qualités différentes

Les articles sur la précision des logiciels commencent à avoir des répercussions assez positives sur la qualité des logiciels. Ainsi JMP, Stata, TSP présentent sur leur site web les résultats de leur logiciel aux différents tests. La référence à ces tests et la publication complète des résultats est souvent un gage de qualité du logiciel et, a contrario, l'absence de référence à ces tests devrait inciter à une certaine méfiance.

En général, les algorithmes des logiciels tendent à s'améliorer d'une version à l'autre, parfois très nettement. C'est le cas de la procédure ANOVA de SAS qui, entre les versions 6.12 et 8.2, a été complètement revue, comme le montrent les résultats du tableau 12.

<u>Tableau 12</u> : Précision de la procédure ANOVA dans SAS 6.12 et SAS 8.2

		SAS 6.12	SAS 8.2
SiRstv	Facile	8.3	12.7
SmLs01	Facile	13.3	15.0
SmLs02	Facile	11.4	13.9
SmLs03	Facile	11.8	12.7
AtmWtAg	Moyen	0.9	8.8
SmLs04	Moyen	0.8	10.4
SmLs05	Moyen	0.0	10.2
SmLs06	Moyen	0.0	10.2
SmLs07	Difficile	0.0	4.4
SmLs08	Difficile	0.0	4.2
SmLs09	Difficile	0.0	4.2

Ce n'est malheureusement pas toujours le cas. Ainsi, McCullough et Wilson ([17]) regrettent que Microsoft n'ai apporté aucune amélioration aux algorithmes statistiques de Excel : les résultats obtenus par ce logiciel sont les mêmes dans les versions 97, 2000 et XP. Ils sont par ailleurs particulièrement sévères sur les corrections apportées au générateur de nombres aléatoires distribués selon une loi normale.

4. Conclusion

Les différents jeux d'essai du NIST, les données de Wilkinson, les tests DIEHARD mis au point par Marsaglia pour les générateurs de nombres aléatoires et ceux de Knüsel pour les distributions statistiques ont permis de définir un cadre méthodologique standard d'évaluation et de comparaison de la précision des logiciels statistiques.

Les diverses expériences faites montrent que les logiciels statistiques sont de qualité assez variable : le critère « précision des calculs » doit être sérieusement pris en compte lors du choix d'un logiciel. Mathematica est le seul logiciel réussissant un score parfait sur les jeux d'essai du NIST. Excel et Gauss sont, a contrario, les logiciels testés présentant les résultats les plus décevants. Ce n'est en soi pas très inquiétant pour Excel qui n'est pas un logiciel statistique proprement dit. Au contraire, Gauss est très populaire auprès des économètres et on ne peut que s'inquiéter de ses performances assez décevantes en régression.

Ces jeux d'essai tendent à devenir des standards auxquels les meilleurs logiciels n'hésitent plus à faire référence sur leurs sites internet. Indubitablement, ils entraînent aussi une amélioration de la qualité des algorithmes et des générateurs de nombres aléatoires.

Bibliographie

[1] Altman, M., "A Review of JMP 4.03 with Special Attention to its Numerical Accuracy", *American Statistician*, vol. 56, pp. 72-75, 2003.

[2] Brown, B. W., "DCDFLIB v1.1" (Double precision Cumulative Distribution Function LIBrary), disponible sur ftp://odin.mdacc.tmc.edu/pub/source, 1998.

[3] Creighton, L., Ding, J., "Assessing the Numerical Accuracy of JMP", disponible sur http://www.jmp.com/product/NIST.pdf

[4] Creighton, L., "Assessing Random Numbers in JMP", disponible sur http://www.jmp.com/product/RNG.pdf

[5] Knüsel, L., "On the accuracy of statistical distributions in Microsoft Excel 97", *Computational Statistics and Data Analysis*, 26, pp. 375–377, 1998.

[6] Knüsel, L., "On the accuracy of statistical distributions in Gauss", *Computational Statistics and Data Analysis*, 20, pp. 699–702, 1995.

[7] Knüsel, L., "Computergestützte Berechnung Statistischer Verteilungen", Oldenburg, München-Wien, 1989. Une version anglaise du programme est disponible sur www.stat.uni-muenchen.de/~knuesel/elv.

[8] Knuth, D.E., The Art of Computer Programming, Addison-Wesley, 1997.

[9] Longley, J. W., "An Appraisal of Computer Programs for the Electronic Computer from the Point of View of the User", *Journal of the American Statistical Association*, 62, pp. 819-841, 1967.

[10] Maballée, Colette et Berthe, "Algorithms and Best Linear Unbiased EstimatorS », *Journal of the Statistical Society of Dublin*, vol. 49, pp. 469-475, 1925.

[11] Marsaglia, G., "DIEHARD: A Battery of Tests of Randomness", http://stat.fsu.edu/pub/~diehard, 1996.

[12] Marsaglia, G., "Random Numbers Fall Mainly in the Planes", in *Proceedings of the National Academy of Sciences of the USA*, 60, pp. 25-28, 1968.

[13] McCullough, B.D., “Assessing the reliability of statistical software: Part I”, *The American Statistician*, vol. 52, n° 4, pp. 358–366, November 1998.

[14] McCullough, B. D., “Assessing the reliability of statistical software: Part II”, *The American Statistician*, vol. 53, n° 2, pp. 149-15.09, May 1999.

[15] McCullough, B. D., “The Accuracy of Mathematica 4 as a statistical package”, *Computational Statistics*, vol. 15.0, pp. 279-299, 2000.

[16] McCullough, B. D., Vinod, H. D., “The Numerical Reliability of Econometric Software”, *Journal of Economic Litterature*, vol. XXXVII, pp. 633-665, 1999.

[17] McCullough, B.D., Wilson, B., “On the accuracy of statistical procedures in Microsoft Excel 2000 and XP”, *Computational Statistics & Data Analysis*, vol. 40, 3, pp. 325-332, 2002.

[18] McKinnon, J. G., “Numerical Distribution Functions for Unit Root and Cointegration tests”, *Journal of Applied Econometrics*, 11, pp. 601-618, 1996.

[19] Nerlove, M., “On the numerical accuracy of Mathematica 4.1 for doing ordinary least squares regression”, Manuscript, AREC, University of Maryland, 2001, disponible sur http://www.arec.umd.edu/mnerlove/mnerlove.htm.

[20] NIST, “StRD: Statistical Reference Datasets for Assessing the Numerical Accuracy of Statistical Softwares”, disponible sur http://www.nist.gov/itl/div898/strd, 1998.

[21] SAS Institute, “Assessing the Numerical Accuracy of SAS Software”, disponible sur http://www.sas.com/rnd/app/papers/statisticalaccuracy.pdf

[22] Simon, S. D., Lesage, J. P., "Assessing the Accuracy of ANOVA Calculations in Statistical Software", *Computational Statistics & Data Analysis*, 8, pp. 325-332, 1989.

[23] Spencer, J., “On the graduation of rates of sickness and mortality”, *Journal of the Institute of Actuaries*, vol. 38, 1904.

[24] Stata, Statistical Software, résultats des tests disponibles sur http://www.stata.com/support/cert/

[25] TSP International, Benchmarks, http://www.tspintl.com/products/tsp/benchmarks/index.htm

[26] Vinod, H. D., “Review of Gauss for Windows, Including its Numerical Accuracy”, *Journal of Applied Econometrics*, vol. 15.0, pp. 211-220, 2000.

[27] Wampler R. H., “A Report on the Accuracy of Some Widely-Used Least Squares Computer Programs”, *Journal of the American Statistical Association*, 65, pp. 549-565, 1970.

[28] Wampler R. H., “Test Procedures and Test Problems for Least Squares Algorithms”, Journal of Econometrics, 12, pp. 3-22, 1980.

[29] Wilkinson, L., *Statistics Quiz*, Evanston, IL: SYSTAT Inc., 1985, (disponible sur http://www.tspintl.com/benchmarks)