REVUE DE STATISTIQUE APPLIQUÉE

B. LAUMON

J. L. MARTIN

D. BALLAND

Un algorithme de calcul de la probabilité qu'une variable de Fisher-Snedecor, à premier degré de liberté impair, excède une valeur donnée

Revue de statistique appliquée, tome 33, nº 4 (1985), p. 29-40

<http://www.numdam.org/item?id=RSA_1985__33_4_29_0>

UN ALGORITHME DE CALCUL DE LA PROBABILITÉ QU'UNE VARIABLE DE FISHER-SNEDECOR, A PREMIER DEGRÉ DE LIBERTÉ IMPAIR, EXCÈDE UNE VALEUR DONNÉE

B. LAUMON [1], J.L. MARTIN [1], D. BALLAND [2]
(1) INSERM U265, 151, cours Albert Thomas, 69003 Lyon
(2) CHEMDATA, 17, quai Joseph Gillet, 69004 Lyon

RESUME

Les auteurs proposent un algorithme qui permet de calculer la probabilité qu'une variable de Fisher-Snedecor excède une valeur donnée, lorsque le premier degré de liberté est impair. Il peut être utilisé pour la détermination de valeurs très faibles, sa seule limite étant la puissance de résolution de l'ordinateur utilisé.

SUMMARY

An algorithm is proposed to compute the probability associated with the upper tail of the F or variance-ratio distribution with odd first degree of freedom. The power of computer is the only limit and very low expected values may be computed.

INTRODUCTION

Il existe divers algorithmes permettant de calculer la probabilité qu'une variable de Fisher-Snedecor excède une valeur donnée. Ces algorithmes diffèrent essentiellement selon la parité du premier degré de liberté ν_1. Lorsque ν_1 est pair, ils permettent de calculer cette probabilité même lorsqu'elle s'avère être très faible. Par contre, lorsque ν_1 est impair, les algorithmes généralement proposés ne le permettent pas.

Après avoir rappelé les raisons de cette différence, nous présentons un algorithme qui, pour ν_1 impair, offre les mêmes possibilités que ceux existant pour ν_1 pair.

I. LE PROBLÈME

Nous désirons pouvoir calculer, dans tous les cas, la probabilité $Q(F, \nu_1, \nu_2)$ qu'une variable de Fisher-Snedecor, à ν_1 et ν_2 degrés de liberté, excède une valeur donnée F.

L'expression à calculer est la suivante :

Mots-clés : **Variable de Fisher-Snedecor, Fonction de répartition, Algorithme.**

$$Q(F, \nu_1, \nu_2) = \frac{1}{\beta\left(\frac{\nu_1}{2}, \frac{\nu_2}{2}\right)} \int_F^{+\infty} \frac{\left(\frac{\nu_1}{\nu_2}\right)^{\nu_1/2} t^{\nu_1/2 - 1}}{\left(1 + \frac{\nu_1}{\nu_2} t\right)^{(\nu_1 + \nu_2)/2}} dt$$

avec $F \geqslant 0$ et

$$\beta\left(\frac{\nu_1}{2}, \frac{\nu_2}{2}\right) = \frac{\Gamma\left(\frac{\nu_1}{2}\right) \Gamma\left(\frac{\nu_2}{2}\right)}{\Gamma\left(\frac{\nu_1 + \nu_2}{2}\right)}$$

Lorsque ν_1 est pair, cette expression peut se calculer en sommant $\nu_1/2$ termes, tous positifs :

$$Q(F, \nu_1, \nu_2) = x^{\nu_2/2} \left[1 + \sum_{p=1}^{p=\nu_1/2-1} (1 - x)^p \prod_{i=1}^{i=p} \frac{\nu_2 + 2i - 2}{2i}\right]$$

avec

$$x = \frac{\nu_2}{\nu_2 + \nu_1 F}$$

Lorsque ν_1 est impair, il est habituel de considérer la parité de ν_2. Si ν_2 est pair, les propriétés de symétrie de Q permettent de se ramener au cas précédent :

$$Q(F, \nu_1, \nu_2) = 1 - Q(1/F, \nu_2, \nu_1)$$

Et si les deux degrés de liberté sont impairs, un changement de variable trigonométrique permet de proposer une solution :

$$Q(F, \nu_1, \nu_2) = 1 - A'_{\nu_2} + B'_{\nu_1}$$

avec, si l'on pose $\vartheta = \operatorname{arctg}\left(\frac{\nu_1}{\nu_2} F\right)^{1/2}$:

$$A'_{\nu_2} \left| \begin{array}{ll} = \dfrac{2}{\pi}\left[\vartheta + \sin\vartheta\cos\vartheta\left(1 + \displaystyle\sum_{p=1}^{p=(\nu_2-3)/2} (\cos\vartheta)^{2p} \prod_{i=1}^{i=p} \frac{2i}{2i+1}\right)\right] & \text{si } \nu_2 > 1 \\ = \dfrac{2\vartheta}{\pi} \quad \text{si} \quad \nu_2 = 1 & \end{array} \right.$$

$$\text{et } B'_{\nu_1} \left| \begin{array}{ll} = \dfrac{2}{\sqrt{\pi}} \dfrac{\left(\frac{\nu_2 - 1}{2}\right)!}{\left(\frac{\nu_2 - 2}{2}\right)!} \sin\vartheta\,(\cos\vartheta)^{\nu_2} & \\ \qquad \left[1 + \displaystyle\sum_{p=1}^{p=(\nu_1-3)/2} (\sin\vartheta)^{2p} \prod_{i=1}^{i=p} \frac{\nu_2 + 2i - 1}{2i+1}\right] & \text{si } \nu_1 > 1 \\ = 0 \quad \text{si} \quad \nu_1 = 1 & \end{array} \right.$$

Ces développements sont ceux proposés par M. ABRAMOWITZ et A. STEGUN [1]. Dans les trois cas ainsi définis selon les parités de ν_1 et ν_2, ils constituent des solutions exactes qu'utilisent certaines bibliothèques scientifiques de programmes, telles IMSL [2] ou NAG [7]. De telles applications posent le problème des limites de validité numérique des algorithmes correspondants.

Deux caractéristiques de l'ordinateur utilisé peuvent alors intervenir : sa puissance de résolution, c'est-à-dire la plus petite valeur qu'il distingue de zéro (classiquement de l'ordre de 10^{-39}), et sa précision de calcul, c'est-à-dire le nombre de chiffres significatifs pris en compte pour caractériser un nombre quelconque (classiquement 6 en simple précision). L'examen des développements de $Q(F, \nu_1, \nu_2)$ proposés montre que les limites de validité diffèrent selon la parité de ν_1. En effet, lorsque ν_1 est pair, on somme un nombre fini de termes tous positifs. Et sous réserve de prendre certaines précautions (notamment celle de commencer par sommer entre eux les termes les plus faibles), on peut calculer précisément des probabilités très faibles, voisines de la puissance de résolution de la machine utilisée. Par contre, lorsque ν_1 est impair, quelle que soit la parité de ν_2, on est amené à retrancher de 1 une quantité comprise entre 0 et 1 ($Q(1/F, \nu_2, \nu_1)$ si ν_2 est pair, $A'_{\nu_2} - B'_{\nu_1}$ sinon). Aussi dès que cette quantité sera voisine de 1 à la précision des calculs près, $Q(F, \nu_1, \nu_2)$ sera considérée comme nulle [5, 6]. Ainsi, sur une machine classique, on peut calculer précisément Q même pour des valeurs très faibles de l'ordre de 10^{-39} lorsque ν_1 est pair, mais seulement pour des valeurs supérieures à 10^{-7} lorsque ν_1 est impair. L'algorithme que nous proposons ici remédie à cette difficulté.

II. L'ALGORITHME

Le problème est d'exprimer, pour ν_1 impair, $Q(F, \nu_1, \nu_2)$ sous la forme d'une somme de termes tous positifs.

Soit

$$Q'_\delta = \frac{1}{\beta\left(\frac{\delta}{2}, \frac{\nu_2}{2}\right)} \int_F^{+\infty} \frac{\left(\frac{\nu_1}{\nu_2}\right)^{\delta/2} t^{\delta/2-1}}{\left(1 + \frac{\nu_1}{\nu_2} t\right)^{(\delta+\nu_2)/2}} dt$$

où δ est un entier quelconque.

Q apparaît comme un cas particulier de Q' (pour $\delta = \nu_1$).

Une intégration par partie permet d'établir une relation de récurrence sur Q', en fonction de δ :

$$Q'_\delta = \frac{\Gamma\left(\frac{\delta + \nu_2}{2} - 1\right)\left(\frac{\nu_1}{\nu_2}\right)^{\delta/2-1} F^{\delta/2-1}}{\Gamma\left(\frac{\delta}{2}\right)\Gamma\left(\frac{\nu_2}{2}\right)\left(1 + \frac{\nu_1}{\nu_2} F\right)^{(\delta+\nu_2)/2-1}} + Q'_{\delta-2}$$

et plus généralement :

$$Q'_\delta = \sum_{i=1}^{i=n} \frac{\Gamma\left(\frac{\delta + \nu_2}{2} - i\right)\left(\frac{\nu_1}{\nu_2}\right)^{\delta/2-i} F^{\delta/2-i}}{\Gamma\left(\frac{\delta}{2} + 1 - i\right)\Gamma\left(\frac{\nu_2}{2}\right)\left(1 + \frac{\nu_1}{\nu_2} F\right)^{(\delta+\nu_2)/2-i}} + Q'_{\delta-2n}$$

où n est un entier quelconque non nul.

ν_1 étant supposé impair et supérieur à 1 (le cas particulier $\nu_1 = 1$ sera évoqué ultérieurement), on choisit : $\delta = \nu_1$ et $n = \frac{\nu_1 - 1}{2}$ et on pose : $p = \dfrac{\nu_1 + 1}{2} - i$.

On obtient : $Q(F, \nu_1, \nu_2) = Q'_{\nu_1}$

$$\text{soit } Q(F, \nu_1, \nu_2) = \sum_{p=1}^{p=(\nu_1-1)/2} \frac{\Gamma\left(\frac{\nu_2 - 1}{2} + p\right)}{\Gamma\left(p + \frac{1}{2}\right)\Gamma\left(\frac{\nu_2}{2}\right)} x^{\nu_2/2}(1 - x)^{p-1/2} + Q'_1$$

$$\text{avec} \quad x = \frac{1}{1 + \frac{\nu_1}{\nu_2} F} \quad \text{et} \quad \nu_1 > 1$$

Le problème est ainsi ramené au calcul de Q'_1. Le changement de variable défini par

$$u = \arcsin\left(\frac{1}{1 + \frac{\nu_1}{\nu_2} t}\right)^{1/2}$$

permet d'obtenir

$$Q'_1 = \frac{2}{\beta\left(\frac{1}{2}, \frac{\nu_2}{2}\right)} \int_0^{\arcsin\sqrt{x}} (\sin u)^{\nu_2 - 1}\, du$$

avec

$$x = \frac{1}{1 + \frac{\nu_1}{\nu_2} F}$$

On remarque :

$$Q'_1 = \frac{2}{\beta\left(\frac{1}{2}, \frac{1}{2}\right)} \arcsin\sqrt{x} \qquad \text{si} \quad \nu_2 = 1$$

$$\text{soit} \quad Q'_1 = \frac{2}{\pi} \arcsin\sqrt{x} \qquad \text{si} \quad \nu_2 = 1$$

Dans le cas général $\nu_2 > 1$, les formules classiques de récurrence sur de telles intégrales, que citent notamment M. ABRAMOWITZ et A. STEGUN [1], permettent d'en proposer un développement. On sait en effet :

$$\int_0^t (\sin u)^n\, du = \frac{(\sin t)^{n+1}\cos t}{n + 1} + \frac{n + 2}{n + 1} \int_0^t (\sin u)^{n+2}\, du$$

avec n entier non nul, t réel
ou plus généralement :

$$\int_0^t (\sin u)^n\, du = \frac{(\sin t)^{n+1}\cos t}{n + 1}\left[1 + \sum_{p=1}^{p=N} (\sin t)^{2p} \prod_{i=1}^{i=p} \frac{n + 2i}{n + 2i + 1}\right] + \prod_{i=1}^{i=N+1} \frac{n + 2i}{n + 2i - 1} \int_0^t (\sin u)^{n+2N+2}\, du$$

En prenant $t = \arcsin\sqrt{x}$ et $n = \nu_2 - 1$, on obtient :

$$Q'_1 = \frac{\Gamma\left(\frac{\nu_2+1}{2}\right)}{\sqrt{\pi}\,\Gamma\left(\frac{\nu_2}{2}+1\right)} x^{\nu_2/2}(1-x)^{1/2}\left[1 + \sum_{p=1}^{p=N} x^p \prod_{i=1}^{i=p} \frac{\nu_2+2i-1}{\nu_2+2i}\right] + R_{N+1}$$

avec

$$R_{N+1} = \frac{2\Gamma\left(\frac{\nu_2+1}{2}\right)}{\sqrt{\pi}\,\Gamma\left(\frac{\nu_2}{2}\right)} \prod_{i=1}^{i=N+1} \frac{\nu_2+2i-1}{\nu_2+2i-2} \int_0^{\arcsin\sqrt{x}} (\sin u)^{\nu_2+2N+1}\,du$$

où N est un entier quelconque non nul.

Or R_{N+1} est une expression positive dont on peut exprimer un majorant qui, si F n'est pas nul, décroît et tend vers zéro lorsque N augmente, et ce en majorant u par sa borne supérieure $\arcsin\sqrt{x}$:

$$\int_0^{\arcsin\sqrt{x}} (\sin u)^{\nu_2+2N+1}\,du \leqslant x^{(\nu_2+2N+1)/2} \arcsin\sqrt{x}$$

d'où $$R_{N+1} \leqslant \frac{2\Gamma\left(\frac{\nu_2+1}{2}\right)}{\sqrt{\pi}\,\Gamma\left(\frac{\nu_2}{2}\right)} x^{(\nu_2+2N+1)/2} \arcsin\sqrt{x} \prod_{i=1}^{i=N+1} \frac{\nu_2+2i-1}{\nu_2+2i-2}$$

Au cours du développement précédent, deux restrictions ont été faites : l'une supposant F non nul, l'autre ν_1 supérieur à 1. Lorsque F est nul, $Q(0, \nu_1, \nu_2)$ vaut 1 par définition et tout ce qui précède est bien sûr superflu. Lorsque ν_1 vaut 1, $Q(F, 1, \nu_2)$ s'identifie à Q'_1 et la première partie du développement est inutile. On obtient finalement :

$$Q(F, \nu_1, \nu_2) \begin{cases} = A_{\nu_1} + B_N + R_{N+1} & \text{si } F > 0 \\ = 1 \quad \text{si} \quad F = 0 \end{cases}$$

$$\text{avec } A_{\nu_1} \begin{cases} = x^{\nu_2/2}(1-x)^{1/2} \displaystyle\sum_{p=1}^{p=(\nu_1-1)/2} \frac{\Gamma\left(\frac{\nu_2-1}{2}+p\right)}{\Gamma\left(p+\frac{1}{2}\right)\Gamma\left(\frac{\nu_2}{2}\right)} (1-x)^{p-1} & \text{si } \nu_1 > 1 \\ = 0 \quad \text{si} \quad \nu_1 = 1 \end{cases}$$

$$\text{et } B_N \begin{cases} = \dfrac{\Gamma\left(\frac{\nu_2+1}{2}\right)}{\sqrt{\pi}\,\Gamma\left(\frac{\nu_2}{2}+1\right)} x^{\nu_2/2}(1-x)^{1/2} \cdot \left[1 + \displaystyle\sum_{p=1}^{p=N} x^p \prod_{i=1}^{i=p} \frac{\nu_2+2i-1}{\nu_2+2i}\right] & \text{si } \nu_2 > 1 \\ = \dfrac{2}{\pi} \arcsin\sqrt{x} \quad \text{si} \quad \nu_2 = 1 \end{cases}$$

$$
\text{et } R_{N+1}\left|\begin{array}{l} \leqslant \dfrac{2\Gamma\left(\dfrac{\nu_2+1}{2}\right)}{\sqrt{\pi}\,\Gamma\left(\dfrac{\nu_2}{2}\right)}\, x^{(\nu_2+2N+1)/2} \arcsin\sqrt{x} \prod_{i=1}^{i=N+1} \dfrac{\nu_2+2i-1}{\nu_2+2i-2} \quad \text{si } \nu_2 > 1 \\[2ex] = 0 \quad \text{si } \nu_2 = 1 \end{array}\right.
$$

avec
$$
x = \frac{1}{1+\dfrac{\nu_1}{\nu_2}F} \quad \text{et} \quad \nu_1 \text{ impair}
$$

Ainsi, à un terme R_{N+1} près (dont on peut calculer un majorant qui tend vers zéro lorsque N augmente), on peut décomposer $Q(F, \nu_1, \nu_2)$, pour ν_1 impair, en deux termes A_{ν_1} et B_N : A_{ν_1} est la somme de $(\nu_1 - 1)/2$ termes, B_N la somme de $(N + 1)$ termes. Ces termes sont tous positifs, et compte tenu de la valeur de x comprise entre 0 et 1, ces termes sont de valeurs décroissantes, tant pour A_{ν_1} que pour B_N. Aussi, à condition de choisir une valeur de N suffisamment grande pour que R_{N+1} soit inférieur à une valeur ε quelconque choisie à l'avance, et à condition de calculer A_{ν_1} et B_N en commençant par sommer leurs termes les plus faibles, il est possible, pour ν_1 impair, de calculer $Q(F, \nu_1, \nu_2)$ de façon précise, quel que soit son ordre de grandeur. Et, comme cela est le cas pour ν_1 pair, la seule limite est la puissance de résolution de la machine sur laquelle est implanté l'algorithme.

Enfin, selon la parité de ν_2, les fonctions Γ non encore explicitées présentent des développements différents qui conduisent à tenir compte de cette parité. On obtient finalement :

$$
Q(F, \nu_1, \nu_2)\left|\begin{array}{lll} = A_{\nu_1} + B_N & & \text{à } \varepsilon \text{ près si } F > 0 \\ = 1 & \text{si} & F = 0 \end{array}\right.
$$

avec ν_1 impair, $x = \dfrac{1}{1+\dfrac{\nu_1}{\nu_2}F}$ et

$$
A_{\nu_1}\left|\begin{array}{l} = x^{\nu_2/2}(1-x)^{1/2} \displaystyle\sum_{p=(\nu_1-1)/2}^{p=1} (1-x)^{p-1} \left|\begin{array}{l} \times \displaystyle\prod_{i=\nu_2/2-1}^{i=1} \dfrac{2p+2i-1}{2i} \quad \text{si } \nu_2 \text{ pair} > 2 \\ \times\, 1 \quad \text{si } \nu_2 = 2 \\ \times \dfrac{2}{\pi} \displaystyle\prod_{i=1}^{i=p} \dfrac{2i}{2i-1} \left|\begin{array}{l} \times \displaystyle\prod_{i=1}^{i=(\nu_2-3)/2} \dfrac{2p+2i}{2i+1} \\ \text{si } \nu_2 > 3 \text{ impair} \\ \times \dfrac{1}{2p} \text{ si } \nu_2 = 1 \\ \times\, 1 \text{ si } \nu_2 = 3 \end{array}\right. \end{array}\right. \\[2ex] = 0 \quad \text{si } \nu_1 = 1 \end{array}\right.
$$

$$
B_N \left\{
\begin{array}{l}
= x^{\nu_2/2}(1-x)^{1/2}
\left\{
\begin{array}{ll}
\times \displaystyle\sum_{p=N}^{p=0} x^p \prod_{i=1}^{i=p+\nu_2/2} \frac{2i-1}{2i} & \text{si } \nu_2 \text{ pair} \\[2ex]
\times \dfrac{2}{\pi} \displaystyle\sum_{p=N}^{p=0} x^p \prod_{i=1}^{i=p+(\nu_2-1)/2} \frac{2i}{2i+1} & \text{si } \nu_2 \text{ impair} > 1
\end{array}
\right. \\[4ex]
= \dfrac{2}{\pi} \arcsin \sqrt{x} \qquad \text{si } \nu_2 = 1
\end{array}
\right.
$$

$$
\text{et} \left\{
\begin{array}{l}
\text{N tel que } \varepsilon \geq \displaystyle\prod_{i=1}^{i=N+1} \frac{\nu_2+2i-1}{\nu_2+2i-2} \, x^{(\nu_2+1)/2+N} \arcsin\sqrt{x}
\left\{
\begin{array}{ll}
\times \displaystyle\prod_{i=1}^{i=\nu_2/2-1} \frac{2i+1}{2i} & \text{si } \nu_2 \text{ pair} > 2 \\[2ex]
\times 1 \quad \text{si } \nu_2 = 2 & \\[1ex]
\times \dfrac{2}{\pi} \displaystyle\prod_{i=1}^{i=(\nu_2-1)/2} \frac{2i}{2i-1} & \text{si } \nu_2 > 1 \text{ impair}
\end{array}
\right. \\[4ex]
\varepsilon = 0 \qquad \text{si } \nu_2 = 1
\end{array}
\right.
$$

III. COMMENTAIRES

L'algorithme proposé est valable, pour ν_1 impair, quelles que soient les valeurs respectives de ν_1, ν_2, F. En pratique, il peut présenter l'inconvénient de nécessiter le calcul d'un très grand nombre de termes pour certaines valeurs de ν_1, ν_2, F. Le tableau 1 illustre les variations de N sur un exemple, pour lequel on choisit $\varepsilon = 2.94\ 10^{-39}$ (puissance de résolution de la machine utilisée). Pour ν_1 et ν_2 fixés, N est très élevé pour des valeurs faibles de F, mais décroît très rapidement. De même, N s'avère très élevé pour des valeurs élevées de ν_2 ou faibles de ν_1.

Une considération pratique permet cependant de réduire sensiblement le nombre de termes à calculer. En effet, compte tenu du nombre limité de chiffres significatifs du résultat final, il est totalement inutile de calculer des termes qui n'en modifient pas l'expression. Aussi, si l'on souhaite que le résultat final soit connu avec n chiffres significatifs, il suffit de calculer A_{ν_1} avant B_N afin de pouvoir choisir N tel que le majorant de R_{N+1}, défini ci-dessus, soit inférieur à ε' :

$$\varepsilon' = 10^{-(n+1)} A_{\nu_1}$$

Les valeurs de N ainsi obtenues, sur le même exemple que précédemment, figurent dans le tableau 1. En pratique, et afin de prévoir tous les cas possibles (notamment ceux où A_{ν_1} est nul), on retient pour déterminer N le majorant de ε (fixé a priori par l'utilisateur) et de ε' (fonction de A_{ν_1} et du nombre de chiffres significatifs souhaités dans l'expression du résultat final).

On peut remarquer que chercher à minimiser N a un double intérêt car la valeur de N est obtenue par un algorithme d'ordre N, et surtout l'algorithme général est d'ordre N^2.

TABLEAU I

Variations de N en fonction de F, ν_1, ν_2

$\nu_1 = 5$ et $\nu_2 = 19$			$\nu_1 = 5$ et $F = 10$			$\nu_2 = 240$ et $F = 10$		
F	N *	N **	ν_2	N *	N **	ν_1	N *	N **
10^{-3}	363 647	78 273	3	30	3	1	2 176	2 159
10^{-2}	35 950	6 923	5	35	4	3	664	112
10^{-1}	3 582	602	10	45	5	5	367	53
1	384	53	50	106	14	11	123	4
10	61	7	100	176	24	15	68	0
10^2	17	2	500	721	109	21	25	0
10^3	6	1	1 000	1 404	219	25	8	0
10^4	1	1	5 000	6 927	1 159	31	0	0

* pour calculer Q à 2.94 10^{-39} près $(R_{N+1} \leq \varepsilon)$

** pour connaître Q avec 4 chiffres significatifs $(R_{N+1} \leq \varepsilon')$

Notons enfin que N pourrait être encore minimisé à condition de rechercher ε' de façon itérative :

$$\varepsilon' = 10^{(n+1)} (A_{\nu_1} + B_N)$$

$$\text{avec N tel que } R_{N+1} \leqq \varepsilon'$$

Cette dernière solution n'est cependant pas retenue car le calcul de B_N s'effectue en commençant par sommer les termes de plus grand indice, contrainte qui nécessiterait de calculer $(N+1-j)$ fois B_j et donc qui augmenterait les temps de calcul alors que la recherche d'un N plus faible avait pour seul but de les réduire. De plus, cette ultime optimisation n'aurait d'intérêt que dans les cas où l'utilisation de l'algorithme lui-même ne se justifie pas.

En effet, nous pouvons rappeler que l'algorithme proposé, bien que valable quels que soient ν_1 (impair), ν_2 et F, a pour but de calculer précisément $Q(F, \nu_1, \nu_2)$ lorsque les algorithmes classiques sont inopérants, c'est-à-dire lorsque les valeurs attendues de $Q(F, \nu_1, \nu_2)$ sont faibles et plus précisément, si l'on considère le contexte de l'analyse de la variance, lorsque ν_1 est faible, ν_2 élevé et F élevé. Le tableau II en donne un exemple et montre que, dans ce contexte, N reste modéré.

Enfin, il est certain que l'emploi d'un tel algorithme ne s'impose pas lors des tests statistiques habituels où seule la « signification à 5 % » est importante. Par contre, il peut être utile dans certaines méthodes d'analyse multivariées proches de la segmentation où $Q(F, \nu_1, \nu_2)$, résultat d'une comparaison multiple de moyennes par analyse de variance, peut être utilisé comme critère de choix des variables [3, 4].

TABLEAU 2

	$\nu_1 = 3$, $\nu_2 = 25$		$\nu_1 = 15$, $\nu_2 = 150$		$\nu_1 = 9$, $\nu_2 = 2\ 000$	
F	(N)*	Q	(N)*	Q	(N)*	Q
1.5	(68)	.238	(25)	.111	(1 312)	.142
5	(22)	7.47×10^{-3}	(0)	6.65×10^{-8}	(211)	1.10×10^{-6}
10	(12)	1.63×10^{-4}	(0)	3.60×10^{-16}	(53)	3.75×10^{-15}
15	(9)	8.64×10^{-6}	(0)	6.17×10^{-23}	(14)	7.86×10^{-24}
20	(7)	7.96×10^{-7}	(0)	1.38×10^{-28}	(0)	1.72×10^{-32}

* pour connaître Q avec 3 chiffres significatifs.

BIBLIOGRAPHIE

[1] M. ABRAMOWITZ, I.A. STEGUN. — *Handbook of mathematical functions,* Dover Publications Inc., New-York, 1972.

[2] IMSL Inc. — Mathematical and statistical library package, edition 8, Houston, 1980.

[3] B. LAUMON. — Une méthode de reconnaissance de formes pour l'estimation d'une variable continue : application à la docimologie. *Thèse Docteur-Ingénieur,* Lyon I, 1979, n° 373.

[4] B. LAUMON, D. TOUNISSOUX, D. BALLAND, M. TERRENOIRE. — Estimation automatique d'une variable continue à l'aide de variables hétérogènes. *Journées d'Informatique médicale,* Toulouse, mai 1980.

[5] M. LENTNER. — On the exact calculation of cumulative F probabilities. *Comm. Statist. Simula. Computa.,* 1976, B5 (4), 149-154.

[6] A.J. MILLER. — Cancellation errors in calculating F probabilities. *Comm. Statist. Simula. Computa.,* 1978, B7 (31), 297-301.

[7] NUMERICAL ALGORITHMS GROUP Ltd. — *NAG fortran library manual,* Mark 8, Oxford, 1980.

ANNEXE : PROGRAMME (FORTRAN 77) : CALCUL DE $Q(F, \nu_1, \nu_2)$;

$F > 0$, ν_1 et ν_2 quelconques (pairs ou impairs).

```
*=====================================================================
*
      FUNCTION  Q  (F ,NU1,NU2,NBCHIF)
*
*        Q          =   probabilite calculee
*
*        F          =   valeur seuil de la variable de Fischer Snedecor
*
*        NU1 (V1) =   premier degre de liberte, en valeur entiere (reelle)
*
*        NU2 (V2) =   deuxieme degre de liberte, en valeur entiere (reelle)
*
*        NBCHIF     =   nombre de chiffres signicatifs exacts imposes pour Q
*
*                          ν2
*        X          =   ----------
*                        ν2 + ν1 F
*
*
*        XP         =   x^(ν2/2) (1-x)^(1/2)
*
*
*     en general :
*
*        XP*A       =   A_ν1
*
*        XP*BN      =   B_N
*
*     -----------------------------------------------------------------
      REAL        Q  ,F
      INTEGER           NU1,NU2,NBCHIF
*
      INTEGER     P
      REAL        V1,V2,X,XP,A,BN,PROD,RI,PI
*
      DATA        PI  / 3.1415 92653 58979 /
*---------------------------------------------------------------------
*
      Q=0.
      A=0.
      BN=0.
      V1=REAL(NU1)
      V2=REAL(NU2)
      X=V2/(V2+(V1*F))
      IF(MOD(NU1,2).NE.0) THEN
*                                                 -- NU1 impair --
         XP=X**(V2/2)*SQRT(1.-X)
         IF( MOD(NU2,2).EQ.0 ) THEN
            IF(NU1.GT.1) THEN
               DO 2 P= (NU1-1)/2,1,-1
                  PROD=1.
                  DO 1 I=1,(NU2/2)-1
                     RI=2.*REAL(I)
                     PROD=PROD*(2.*REAL(P)-1.+RI)/RI
1                 CONTINUE
                  A=A+(PROD*(1.-X)**(P-1))
2              CONTINUE
            ENDIF
```

```
            N=NTERME (  NU2 , A  , NBCHIF , X , XP )
            DO 4 P=N,0,-1
               PROD=1.
               DO 3 I=1,P+ (NU2/2)
                  RI=2.*REAL(I)
                  PROD=PROD*((RI-1.)/RI)
3              CONTINUE
               PROD=PROD*X**P
               BN=BN+PROD
4           CONTINUE
            Q=XP*(A+BN)
         ELSE
            XP=XP*2./PI
            IF(NU2.NE.1) THEN
               IF(NU1.GT.1) THEN
                  DO 7  P=(NU1-1)/2,1,-1
                     PROD=1.
                     IF(NU2.GT.3) THEN
                        DO 5  I=1,(NU2-3)/2
                           RI=2.*REAL(I)
                           PROD=PROD*((2*REAL(P)+RI)/(RI+1.))
5                       CONTINUE
                     ENDIF
                     DO 6  I=1,P
                        RI=2.*REAL(I)
                        PROD=PROD*RI/(RI-1.)
6                    CONTINUE
                     A=A+(PROD*(1.-X)**(P-1))
7                 CONTINUE
               ENDIF
               N=NTERME (  NU2 , A  , NBCHIF , X , XP )
               DO 9 P=N,0,-1
                  PROD=1.
                  DO 8  I=1,P+(NU2-1)/2
                     RI=2.*REAL(I)
                     PROD=PROD*(RI/(RI+1.))
8                 CONTINUE
                  BN=BN+(PROD*X**P)
9              CONTINUE
                  Q=XP*(A+BN)
            ELSE
               IF(NU1.GT.1) THEN
                  DO 11 P=(NU1-1)/2,1,-1
                     PROD=1.
                     DO 10 I=1,P
                        RI=2.*REAL(I)
                        PROD=PROD*(RI/(RI-1.))
10                   CONTINUE
                     A=A+(PROD*(1.-X)**(P-1)/(2.*REAL(P)))
11                CONTINUE
               ENDIF
               BN= (2./PI) *ASIN(SQRT(X))
               Q=XP*A + BN
            ENDIF
         ENDIF
      ELSE
```

```
*                                                -- NU1 pair --
         IF(NU1.EQ.2) THEN
            Q=X**(V2/2.)
         ELSE
            DO 13 P=(NU1/2)-1,1,-1
               PROD=1.
               DO 12 I=1,P
                  RI=2.*REAL(I)
                  PROD=PROD*(V2+RI-2.)/RI
12             CONTINUE
               Q=Q+PROD*(1.-X)**P
13          CONTINUE
            Q=X**(V2/2.)*(1.+Q)
         ENDIF
      ENDIF
      RETURN
      END
*
*================================================================
*
      FUNCTION  NTERME (  NU2 , A  , NBCHIF , X , XP )
*
*        NTERME   =   nombre (N) de termes necessaire au calcul de Q
*
*        PCE      =   puissance de resolution de la machine utilisee
*     ------------------------------------------------------------
      REAL                       A            , X , XP
      INTEGER              NU2 ,       NBCHIF
*
      REAL      PI  ,PCE
      DATA      PI  / 3.1415 92653 58979 /
     -,         PCE / 2.94 E-39          /
*-----------------------------------------------------------------
      NTERME=0
      V2=REAL(NU2)
      EPS=XP * (A * 10.**-(NBCHIF+1))
      EPS=MAX ( EPS , PCE)
      IF(MOD(NU2,2).EQ.0) THEN
         RESTE=1.
         DO 1 I=1,(NU2/2)-1
            RI=2*REAL(I)
            RESTE=RESTE*(RI+1.)/RI
1        CONTINUE
      ELSE
         RESTE=2./PI
         DO 2 I=1,(NU2-1)/2
            RI=2*REAL(I)
            RESTE=RESTE*(RI/(RI-1.))
2        CONTINUE
      ENDIF
      RESTE=RESTE* ((V2+1.)/V2) * ASIN(SQRT(X)) * X**((V2+1.)/2.)
3     IF(RESTE.GT.EPS) THEN
         NTERME=NTERME+1
         RESTE = RESTE * ((V2+2.*REAL(NTERME)+1.)
     -                /  (V2+2.*REAL(NTERME)))  * X
         GOTO 3
      ENDIF
*
*     precaution possible si nombre de termes N important
*
      RETURN
      END
```