



Examen final de la session d'hiver 2020

Consignes

- Ce travail est à réaliser individuellement.
- Vous avez droit à un ordinateur et à toute la documentation.
- Vous pouvez accéder à Internet pour consulter de la documentation ou des bases de connaissances, mais en aucune façon pour communiquer avec d'autres personnes, de quelque manière que ce soit.
- Vous devez obligatoirement remplir la *Déclaration d'intégrité relative aux travaux et aux examens* dans monPortail.
- La remise du livrable s'effectue dans la boîte de dépôt dans monPortail.

1 Contexte

Vous êtes auxiliaire d'enseignement pour un cours de méthodes quantitatives qui aborde, entre autres sujets, l'estimation ponctuelle par la méthode du maximum de vraisemblance. La professeure a développé une application Shiny pour illustrer l'ajustement d'un modèle probabiliste à des données. Elle s'est chargée de l'interface et de la production des objets de sortie, mais la procédure de calcul des paramètres du modèle à partir des données demeure inachevée.

2 Mandat

Votre mandat consiste à rendre opérationnelle l'application Shiny livrée avec le présent énoncé. Pour ce faire, vous devez compléter le code de la fonction réactive `param` dans la partie serveur de l'application. Cette fonction doit retourner un vecteur nommé (ou étiqueté) de deux éléments contenant les paramètres estimés par la méthode du maximum de vraisemblance pour la distribution choisie dans l'interface de l'application.

3 Livrables

Le seul livrable est le fichier de script `app.R` contenant l'application Shiny opérationnelle. La [figure 1](#) illustre de ce que devrait produire l'application.

Ajustement de modèles par le maximum de vraisemblance

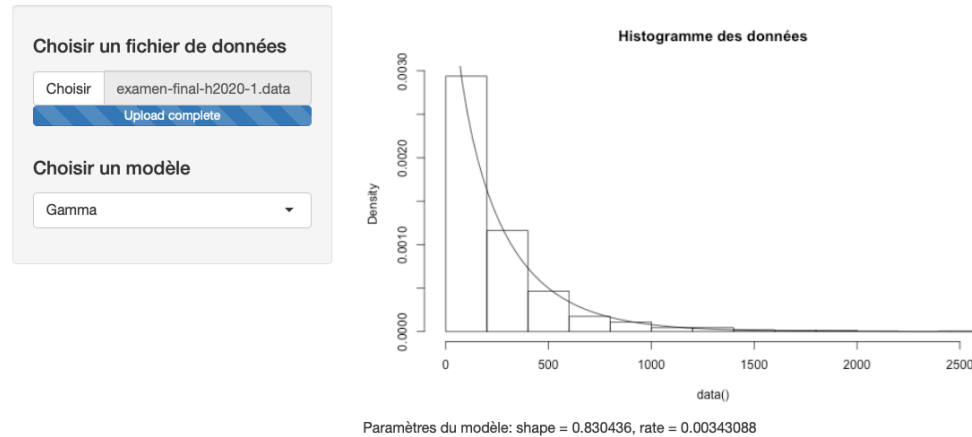


FIG. 1 – Présentation générale de l'application Shiny

4 Informations et consignes supplémentaires

- Vous devez utiliser la méthode de Newton-Raphson multivariée pour résoudre numériquement le problème de maximisation de la fonction de vraisemblance. L'[annexe A](#) présente cette généralisation de la méthode de Newton-Raphson univariée que vous connaissez.
- Les valeurs de départ fournies à la méthode de Newton-Raphson doivent être les estimateurs des moments de la distribution choisie par l'utilisateur. Les formules de ces estimateurs se trouvent à l'[annexe B](#).
- Une mise en œuvre informatique de la méthode de Newton-Raphson multivariée est fournie dans le code de l'application Shiny. Vous pouvez donc avoir recours à cette fonction.
- Lorsqu'elle est utilisée dans un contexte de maximum de vraisemblance, la fonction `nrmult` devrait recevoir les données par l'entremise de son argument « ... ».
- Les utilisateurs de l'application ont le choix entre deux modèles seulement : la distribution de Pareto et la distribution gamma. Les fonctions de densité de probabilité de ces deux distributions se trouvent à l'[annexe B](#).
- La fonction gamma $\Gamma(z)$ n'admet pas de dérivée explicite. Cependant, les deux fonctions mathématiques ci-dessous sont bien définies : la fonction

digamma

$$\Psi(z) = \frac{d}{dz} \ln \Gamma(z) = \frac{\Gamma'(z)}{\Gamma(z)}$$

et la fonction trigamma

$$\psi(z) = \frac{d}{dz} \Psi(z) = \frac{d^2}{dz^2} \ln \Gamma(z).$$

Les fonctions `gamma`, `digamma` et `trigamma` permettent d'obtenir les valeurs des fonctions mathématiques correspondantes dans R.

- Le paquetage **actuar** fournit les fonctions pour la prise en charge de la distribution de Pareto, dont `dpareto` pour calculer les valeurs de la fonction de densité.
- La procédure d'optimisation est très instable pour la distribution de Pareto.
- Des fichiers de données de test — provenant l'un d'une distribution de Pareto, l'autre d'une distribution gamma — vous sont fournis avec cet énoncé.



Si la fonction digamma est la dérivée du *logarithme* de la fonction gamma, la fonction trigamma est, quant à elle, la dérivée de la fonction digamma.



Inspirez-vous du code de l'application Shiny pour compléter la fonction `param`. Si votre fonction est adéquate, il n'y a aucune modification à apporter au code qui vous est fourni.

A Méthode de Newton–Raphson multivariée

Vous avez déjà étudié des méthodes numériques pour résoudre l'équation

$$f(x) = 0,$$

où f est une fonction non linéaire (autrement, le problème est simple). Ce problème peut se généraliser à la résolution de systèmes à n équations non

linéaires et n inconnues de la forme

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0 \\ f_2(x_1, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, \dots, x_n) &= 0. \end{aligned}$$

Soit $\mathbf{x} = (x_1, \dots, x_n)$. Le système d'équations ci-dessus peut alors s'écrire de manière plus compacte ainsi :

$$\begin{aligned} f_1(\mathbf{x}) &= 0 \\ f_2(\mathbf{x}) &= 0 \\ &\vdots \\ f_n(\mathbf{x}) &= 0, \end{aligned}$$

ou encore sous la forme

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

avec

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))^T.$$

La méthode de Newton-Raphson existe également en version multivariée pour résoudre des systèmes d'équations non linéaires. Elle est très similaire à la méthode univariée, sauf qu'elle fait appel à quelques notions simples de calcul matriciel.

Soit

$$\begin{aligned} \mathbf{J}(\mathbf{x}) &= \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \frac{\partial f_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial f_2(\mathbf{x})}{\partial x_1} & \frac{\partial f_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n(\mathbf{x})}{\partial x_1} & \frac{\partial f_n(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f_n(\mathbf{x})}{\partial x_n} \end{bmatrix} \\ &= \left[\frac{\partial f_i(\mathbf{x})}{\partial x_j} \right]_{n \times n}. \end{aligned}$$

On obtient la solution du système d'équations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ par convergence de la suite

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{J}(\mathbf{x}_{k-1})^{-1} \mathbf{f}(\mathbf{x}_{k-1}), \quad k = 1, 2, \dots,$$

où $\mathbf{x}_0 = (x_{10}, \dots, x_{n0})$ est un vecteur de valeurs de départ.

La procédure itérative s'arrête lorsque deux vecteurs successifs \mathbf{x}_k et \mathbf{x}_{k-1} sont « proches », c'est-à-dire lorsque la longueur — ou norme — du vecteur $\mathbf{x}_k - \mathbf{x}_{k-1}$ est petite. Le critère d'arrêt couramment utilisé avec la méthode de Newton-Raphson multivariée est

$$\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_\infty = \|\mathbf{J}(\mathbf{x}_{k-1})^{-1} \mathbf{f}(\mathbf{x}_{k-1})\|_\infty < \varepsilon,$$

où

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

est la norme « infini ».

Exemple 1. Soit le système d'équations

$$\begin{aligned} x_1^2 - 10x_1 + x_2^2 + 7 &= 0 \\ x_1x_2 + x_1 - 10x_2 + 7 &= 0 \end{aligned}$$

ou, dans la notation ci-dessus,

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_1^2 - 10x_1 + x_2^2 + 7 \\ x_1x_2 + x_1 - 10x_2 + 7 \end{bmatrix} = \mathbf{0}.$$

Pour résoudre ce système d'équations à l'aide de la méthode de Newton-Raphson, nous devons d'abord calculer

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} 2x_1 - 10 & 2x_2 \\ x_2 + 1 & x_1 - 10 \end{bmatrix}.$$

Par la suite, il s'agit de définir des fonctions R pour calculer le vecteur \mathbf{f} et la matrice \mathbf{J} :

```
> f <- function(p)
+ {
+   x <- p[1]
+   y <- p[2]
+   c(x^2 - 10 * x + y^2 + 7, x * y + x - 10 * y + 7)
+ }
> fp <- function(p)
+ {
+   x <- p[1]
+   y <- p[2]
+   cbind(c(2 * x - 10, y + 1), c(2 * y, x - 10))
+ }
```

Avec les valeurs de départ $x_1 = x_2 = 0$ et une précision de $\varepsilon = 10^{-5}$, nous obtenons avec la fonction `nrmult` :

```
> (r <- nrmult(f, fp, start = c(0, 0), TOL = 1E-5))
$roots
[1] 0.8447887 0.8568659

$nb.iter
[1] 3

> f(r$roots)      # vérification
[1] 7.519319e-12 3.160139e-12
```

□

Exemple 2. La méthode d'estimation du maximum de vraisemblance demande de maximiser

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^n \ln f(x_i; \boldsymbol{\theta}),$$

où f est une fonction de densité de probabilité et $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)$ est un vecteur de paramètres. Pour ce faire, il faut résoudre avec la méthode de Newton-Raphson multivariée le système d'équations

$$\mathbf{s}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial}{\partial \theta_1} \ell(\boldsymbol{\theta}) \\ \vdots \\ \frac{\partial}{\partial \theta_p} \ell(\boldsymbol{\theta}) \end{bmatrix} = \mathbf{0}.$$

□

B Informations sur les lois de probabilité

Cette annexe fournit les informations suivantes relatives aux distributions de Pareto et gamma : racine (*loi*) des fonctions R ; correspondance entre les paramètres des distributions et les noms d'arguments dans les fonctions R ; fonction de densité de probabilité ; espérance et variance ; estimateurs des moments.

Dans les expressions des estimateurs des moments, $m_1 = n^{-1} \sum_{i=1}^n x_i$ et $m_2 = n^{-1} \sum_{i=1}^n x_i^2$ sont, respectivement, le premier et le second moment empirique.

B.1 Gamma

Racine : gamma,

Paramètres : shape (α), rate ($\lambda = 1/\theta$), scale (θ)

$$f(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}, \quad x > 0$$
$$E[X] = \frac{\alpha}{\lambda}$$
$$\text{Var}[X] = \frac{\alpha}{\lambda^2}$$

$$\hat{\alpha} = \frac{m_1^2}{m_2 - m_1^2} \qquad \hat{\lambda} = \frac{m_1}{m_2 - m_1^2}$$

B.2 Pareto

Racine : pareto,

Paramètres : shape (α), scale (θ)

$$f(x) = \frac{\alpha \theta^\alpha}{(x + \theta)^{\alpha+1}}, \quad x > 0$$
$$E[X] = \frac{\theta}{\alpha - 1}$$
$$\text{Var}[X] = \frac{\alpha \theta^2}{(\alpha - 1)^2 (\alpha - 2)}$$

$$\hat{\alpha} = \frac{2(m_2 - m_1^2)}{m_2 - 2m_1^2} \qquad \hat{\theta} = \frac{m_1 m_2}{m_2 - 2m_1^2}$$