M2 - Probabilités et statistiques des nouvelles données

Projet n°1 Simulation et Copules

Auteur :
Confiac Hendrick
Wague Yakhoub

Novembre 2021

Préface

Ce projet a pour but d'approximer la valeur de l'intégrale de la fonction suivante :
$f(x,y) = \frac{(x*y)^2}{x+y}$, à support dans [2, 4] x [0, 1].

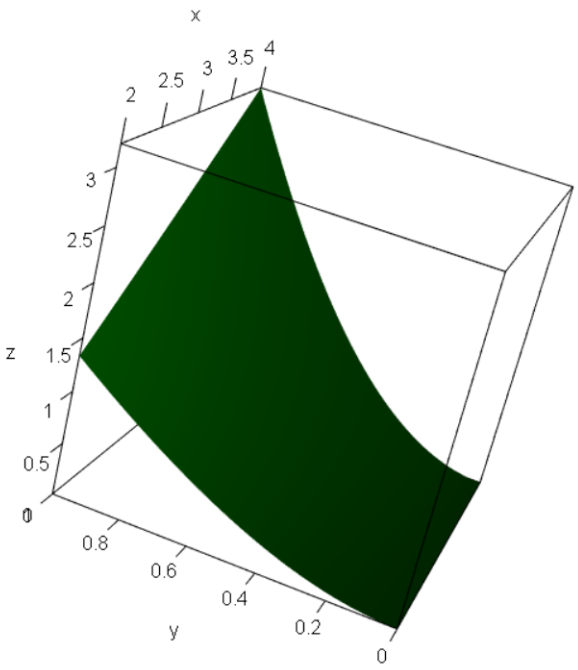


FIGURE 1 – densité de f

Pour ce faire, nous allons utiliser différentes méthodes, puis sélectionner celle qui donnera la meilleure approximation de :

$$I = \int_2^4 \int_0^1 f(x,y)\,\mathrm{d}x\mathrm{d}y.$$

Dans chacune de ces méthodes, nous allons répéter 100 fois le procesus d'estimation. Nous aurons alors pour chacune d'elles, un vecteur de taille 100, dans lequel sera stocké 1 estimation pour la simulation des variables U,V et W. Plus tard, nous allons comparer les i^{eme} composantes entre ces vecteurs afin de déterminer la meilleure méthode.

Table des matières

Chapitre 1

Partie I

Notons que par le biais de la library pracma ainsi que la fonction quad2d on obtient une valeur aproximative de I, $I_{theorique} \simeq 1.606988$ que l'on cherchera à approcher.

1.1 Méthode 1

Dans cette partie on utilisera la méthode de fréquence empirique :
Considérons $U \sim \mathcal{U}[2,4]$,$V \sim \mathcal{U}[0,1]$ et $W \sim \mathcal{U}[0, max(f)]$. Avec $max(f) = \max_{(x,y)\in[2,4]X[0,1]}$ (f(x,y))=3.2

Notons que :

$$\mathbb{E}(\mathbb{1}_{f(U,V)>W}) = \frac{P(f(U,V) > W)}{\mathcal{A}} = \frac{I}{\mathcal{A}}$$

où $\mathcal{A}$ correspond au volume du cube dans la préface ($i.e$ $(4-2)*(1-0)*max(f) = 6.4$)
Par la Méthode de Monte-Carlo, calculer I revient à déterminer la valeur de l'estimateur :

$$\hat{I} = \mathcal{A} * \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{f(U_i,V_i)>W_i}$$

Algorithme principal

1- On effectue ($n = 1000$) simulations des variables aléatoires utilisées dans la méthode
2- On calcule l'estimateur associé à la méthode
3- On réitère l'opération 100 fois et on stock les 100 estimations dans un vecteur.

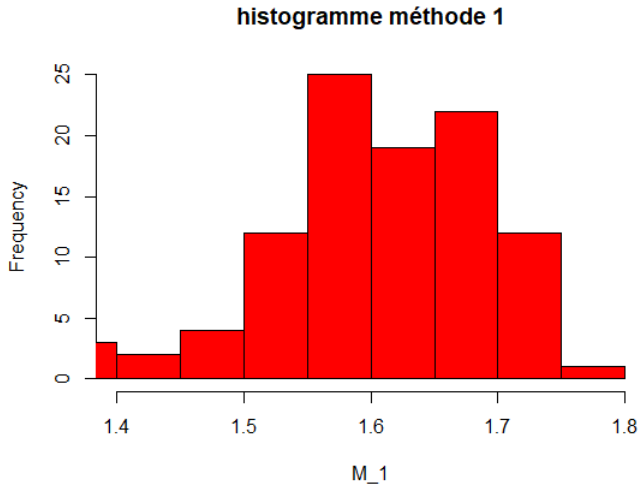
vecteur des estimations de $\hat{I}$ *:*

$M_1 = (1.6768, 1.7024, 1.6064, 1.7408, 1.5168, 1.6832, ..., 1.4912, 1.4528, 1.5872, 1.7024, 1.6384, 1.6832)$

vecteur de variance dont chaque composante correspond à la variance associée à la i^e*me estimations :*

$VAR_{M1} = (7.966658, 7.708147, 7.584160, ..., 7.748820, 8.024347, 7.809215)$

On note donc une estimation de I égale à : $mean(M_1) = 1.608064$ ainsi qu'une variance moyenne de la méthode 1 égale à : $mean(VAR_{M1}) = 7.721581$



Pour les deux méthodes qui vont suivre, nous alllons considérer l'estimateur suivant :

$$\hat{I}_j = \frac{1}{n}\sum_{i=1}^{n} \frac{f(U_i, V_i)}{p_j(U_i, V_i)}, \qquad j = 1, 2$$

où p_j représante la densité construite spécifiquement pour la méthode concernée et (U, V), un couple de variables aléatoires de densité p_j. Notons que :

$$\hat{I}_j \underset{n\longrightarrow\infty}{\longrightarrow} \mathbb{E}[\frac{f(U_1, V_1)}{p_j(U_1, V_1)}] = \int_2^4 \int_0^1 \frac{f(u, v)}{p_j(u, v)} * p_j(u, v)dudv = I$$

Donc trouver I revient à estimer $\hat{I}_j$.

1.2 Méthode 2

Construisons $\mathbf{p}_1$

Soient $U \sim \mathcal{U}[2, 4]$ et $V \sim \mathcal{U}[0, 1]$, telles que $U \perp\!\!\!\perp$ V.
p_1 sera construit comme étant le produit des densités de ces deux variables aléatoires indépendantes.

$$p_1(u, v) = \mathbb{1}_{[2,4]}(u) * \mathbb{1}_{[0,1]}(v)$$

Simulation de (U,V)

Par construction il suffit de simuler n uniformes sur $[2, 4]$ et n uniformes sur $[0, 1]$.

- $U < -runif(n, 2, 4)$
- $V < -runif(n, 0, 1)$

Simulation de la méthode 2

De manière analogue à la méthode 1, on utilise l'algorithme principal et on obtient :

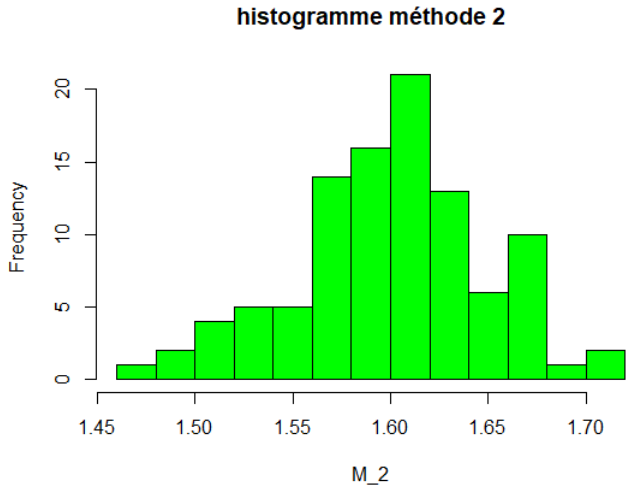
vecteur des estimations de $\hat{I}_1$:

$$M_2 = (1.598746, 1.628629, 1.597118, 1.525716, , ..., 1.510320, 1.558152, 1.617782, 1.660441)$$

vecteur de variance dont chaque composante correspond à la variance associée à la i^eme estimations :

$$VAR_{M2} = (2.154800, 2.035193, 2.270559, ..., 2.162761, 2.079924, 2.051180)$$

On note donc une estimation de I égale à : $mean(M_2) = 1.600149$ ainsi qu'une variance moyenne de la méthode 2 égale à : $mean(VAR_{M2}) = 2.101421$



1.3 Méthode 3 :

Dans cette méthode, on note $\hat{I}_2$ notre estimateur :

$$\hat{I}_2 = \frac{1}{n}\sum_{i=1}^{n}\frac{f(U_i, V_i)}{p_2(U_i, V_i)}$$

Construisons $\mathbf{p_2}$

De manière analogue à la méthode précédente, p_2 sera le produit de deux densités p_x, p_y resp(densité de X,densité de Y), avec $X \perp\!\!\!\perp Y$; le but étant d'avoir une densité p_2 qui ressemble à notre fonction f.

On pose :

$$p_2(x, y) = p_x * p_y$$

avec comme choix :

- $p_x(x) = \frac{1}{6}x * \mathbb{1}_{[2,4]}(x)$
- $p_y(y) = 4y^3 * \mathbb{1}_{[0,1]}(y)$

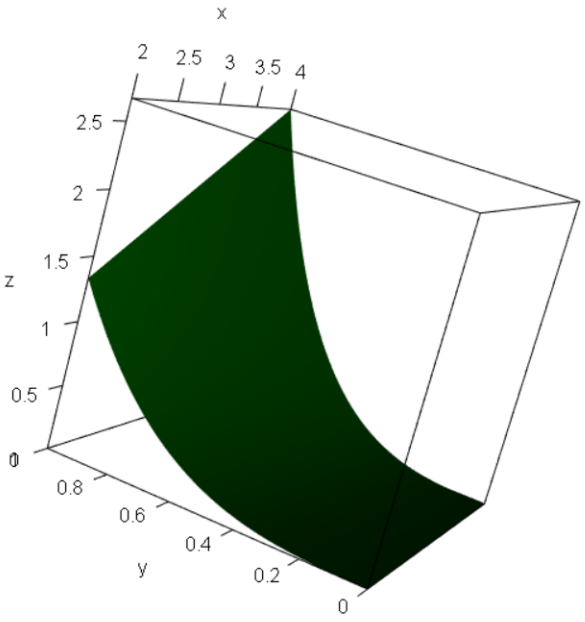


FIGURE 1.1 – densité p_2

Pour simuler le couple (X, Y) de densité p_2, nous allons utiliser la méthode d'inversion de la fonction de répartition . Puisque $X \perp\!\!\!\perp Y$, il s'agira simplement de simuler X, puis de simuler Y.

Simulation de X :

Soit F_{p_x} la fonction de répartion de X. On a :

$$F_{p_x}(x) = \frac{1}{12}(x^2 - 4)$$

Et son inverse :

$$F_{p_x}^{-1}(x) = \sqrt{12x + 4}$$

Algorithme :

On fait n simulations uniformes sur $[0, 1]$, puis on simule Y.

- $v = runif(n, 0, 1)$
- $X = F_{p_x}^{-1}(v)$

Simulation de Y :

Soit F_{p_y} la fonction de répartion de Y. On a :

$$F_{p_y}(y) = y^4$$

Et son inverse :

$$F_{p_y}^{-1}(y) = y^{\frac{1}{4}}$$

Algorithme :

On fait n simulations uniformes sur $[0, 1]$, puis on simule X.

- $v < -runif(n, 0, 1)$
- $Y = F_{p_y}^{-1}(v)$

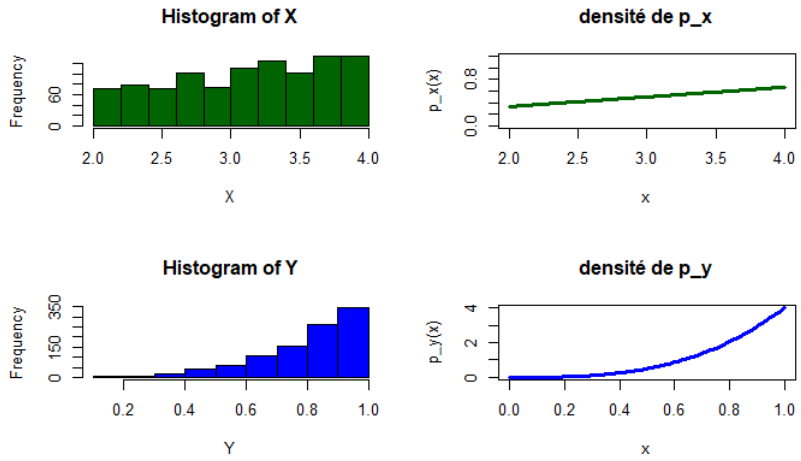


FIGURE 1.2 – Simulation de X et de Y

Simulation de la méthode 3 :

On applique l'algorithme principal et on obtient :
vecteur des estimations de $\hat{I}_2$ *:*

$$M_3 = (1.6072511.6341221.566163, , ..., 1.6170931.5916831.623208)$$

vecteur de variance dont chaque composantes correspond à la variance associée à la i^e*me estimations :*

$$VAR_{M3} = (0.51405930.51718260.3116170, ..., 0.30520040.48310900.3285152)$$

On note donc une estimation de I *égale à :* $mean(M_3) = 1.602257$ *ainsi qu'une variance moyenne de la méthode 2 égale à :* $mean(VAR_{M2}) = 0.382314$

1.4 Comparaisons numériques et graphiques des méthodes :

Comparaisons numériques

Nous allons voir quelle méthode a été la plus précise le plus grand nombre de fois. Cela revient à calculer

$$\sum_{j}^{100} \mathbb{1}_{|M_{i,j}-I_{theorique}|>|M_{k,j}-I_{theorique}|}$$

avec $i \neq k$ *;* $M_{i,j}$*etant la* j^{eme} *compansante du vecteur* M_i*.*

Sur R nous allons utiliser la fonction summary afin de déterminer combien de fois les évènements $\{|M_{i,j} - I_{theorique}| > |M_{k,j} - I_{theorique},|\ i \neq k \in \{1,2,3\}, pour j \in [|1,100|]\}$ *et son complémentaire, se sont produits.*

```
> compar_M_1_M_2 = abs(M_1 - I) > abs(M_2 - I)
> summary(compar_M_1_M_2)
   Mode   FALSE    TRUE
logical      34      66
```

FIGURE 1.3 – comparaison M_1, M_2

```
> compar_M_2_M_3 = abs(M_2 - I) > abs(M_3 - I)
> summary(compar_M_2_M_3)
   Mode   FALSE    TRUE
logical      23      77
```

FIGURE 1.4 – comparaison M_2, M_3

On note que 66% *des estimations par la méthode 2 sont plus precises que celles de la méthode 1. Et* 77% *des estimations par la méthode 3 sont plus precises que celles de la méthode 2.*
De plus la variance de la méthode 3 (0.382314) *est plus petite que celle de la méthode* 2 (2.101421) *qui est elle même plus petite que celle de la méthode* 1 (7.721581)

Comparaisons graphiques

Afin d'illustrer nos précédentes comparaisons, nous allons comparer les histogrammes et la convergence de ces 3 méthodes.

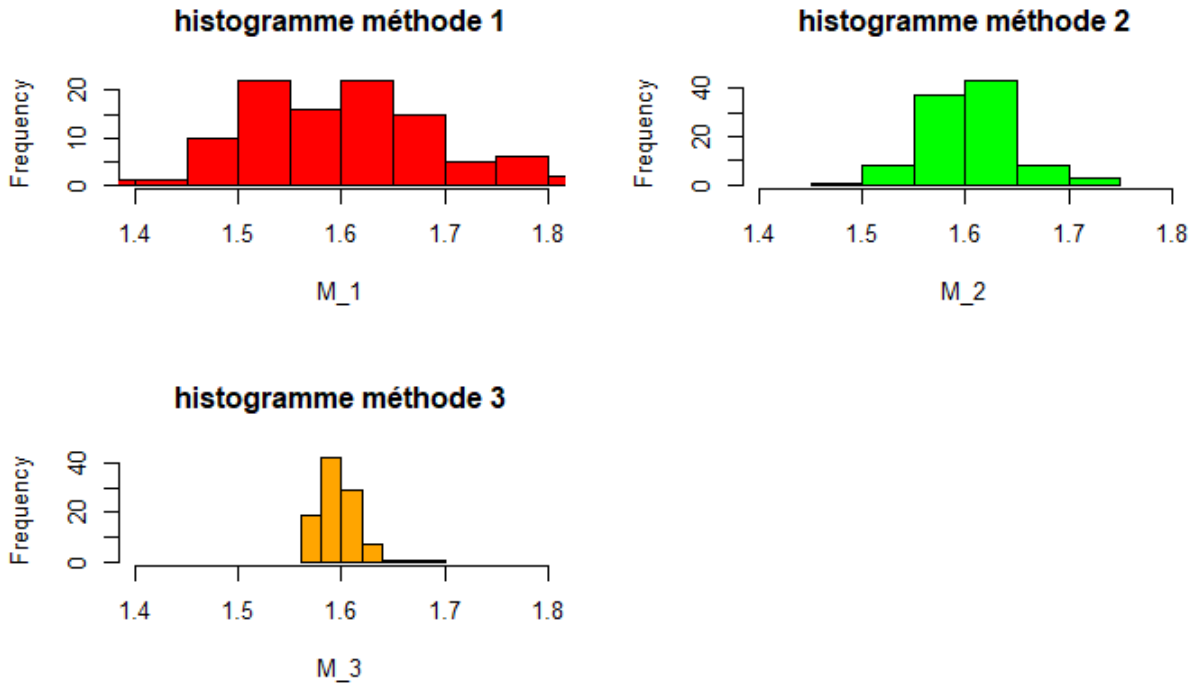


FIGURE 1.5 – Histogrammes des 3 méthodes sous la même échelle

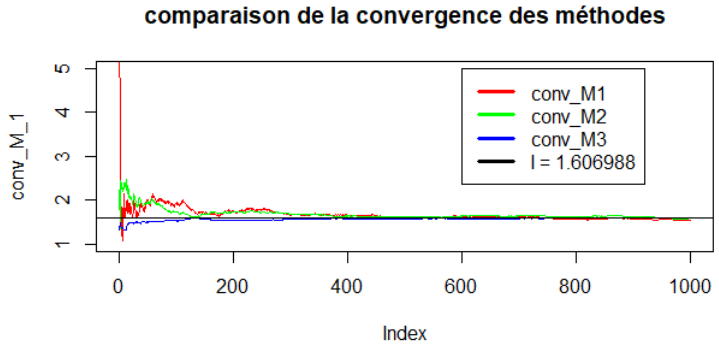


FIGURE 1.6 – convergence des méthodes

— *Les estimations par la méthode* 3 *sont plus concentrées autour de la valeur de I que celles par la méthode* 2 *et convergent plus rapidement.*

— *Les estimations par la méthode* 2 *sont plus concentrées autour de la valeur de I que celles par la méthode* 1 *et convergent plus rapidement.*

Par ailleurs, on sait que pour tout $(X_1, ..., X_n)$*iid d'espérance* m *et de variance* σ^2 *, la moyenne empirique*

$$\hat{X_n} = \frac{1}{n}\sum_{i=0}^{n} X_i \underset{n\longrightarrow\infty}{\longrightarrow} \mathcal{N}(m, \frac{\sigma}{\sqrt{n}})$$

Nous pouvons donc remarquer une différence au niveau des écart-types des 3 méthodes par le graphe ci-dessous.

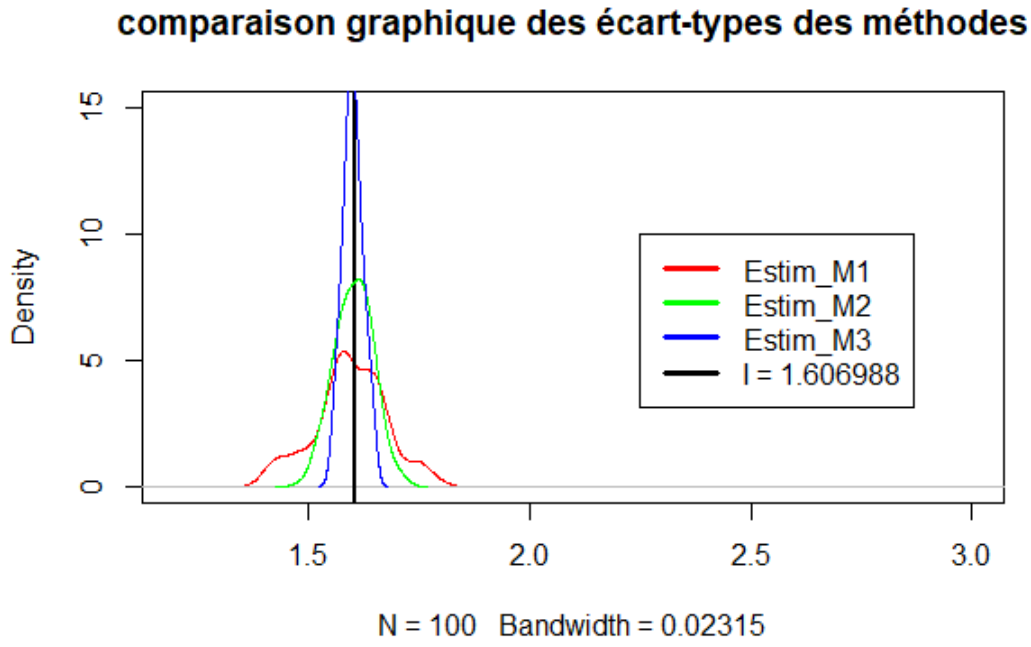


FIGURE 1.7 – graphe des méthodes avec la fonction "density"

On retiendra ainsi, que la méthode 3 est meilleure que la méthode 2 qui elle même est meilleure que la méthode 1 avec à chaque fois, une variance réduite.

Chapitre 2

Partie II

2.1 Méthode 4

Dans cette partie, nous verrons une autre méthode de simulation en utilisant la copule de Clayton.

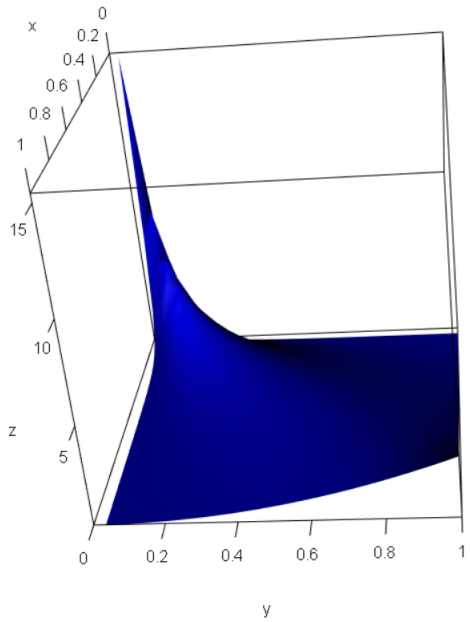


FIGURE 2.1 – Copule de Clayton $\theta = 2$

Reprenons le même estimateur que précédement :

$$\hat{I}_3 = \frac{1}{n} \sum_{i=1}^{n} \frac{f(U_i, V_i)}{p_3(U_i, V_i)},$$

avec :

- $p_3 = f(x) * g(y) * c(F(x), G(y))$ *où* $c(,)$ *est la densité de la copule de Clayton.*

$$c(x,y) = (1+\theta) * (x*y)^{-(1+\theta)} * (-1 + (x)^{-\theta} + (y)^{-\theta})^{(-\frac{1+2*\theta}{\theta})}, x, y \in [0,1]$$

- f, g, *densités de fonction de répartition* $resp(F,G)$ *de lois uniformes qui seront adaptées au support de notre fonction.*

On choisira :

- $f(x) = \frac{1}{2}\mathbb{1}_{[2,4]}(x)$ *avec* $F(x) = \frac{x-2}{2}\mathbb{1}_{[2,4]}(x)$
- $f(y) = \mathbb{1}_{[0,1]}(y)$ *avec* $F(y) = y\mathbb{1}_{[0,1]}(y)$

Notons que notre fonction f a un pic pour des valeurs de x et y grandent. Ainsi dans notre cas, la simulation de (U_2, V_2) *de densité* p_3 *devra nous donner un plot avec beaucoup plus de points en* $x = 4$ *et en* $y = 2$. *Pour se faire nous allons procéder par étape.*

1^{ere} *étape*

On simule (U, V) *dont la loi jointe est la densité de la copule de Clayton. Nous allons utiliser la "méthode d'inversion" de la fonction de répartition conditionnelle, fonction donnée par :* $F_u(v) = ((u^{-\theta} + v^{-\theta} - 1)^{-\frac{\theta+1}{\theta}})u^{-(\theta+1)}$. *Son inverse est* $F_u^{-1}(z) = ((z^{\frac{-\theta}{\theta+1}} - 1) * u^{-\theta} + 1)^{\frac{-1}{\theta}}$. *On simule* U *et* Z *en tant qu'uniforme sur* $[0,1]$. *On pose*

$$V = F_u^{-1}(Z)$$

et le couple (U, V) *a la structure de dépendance de Clayton.*

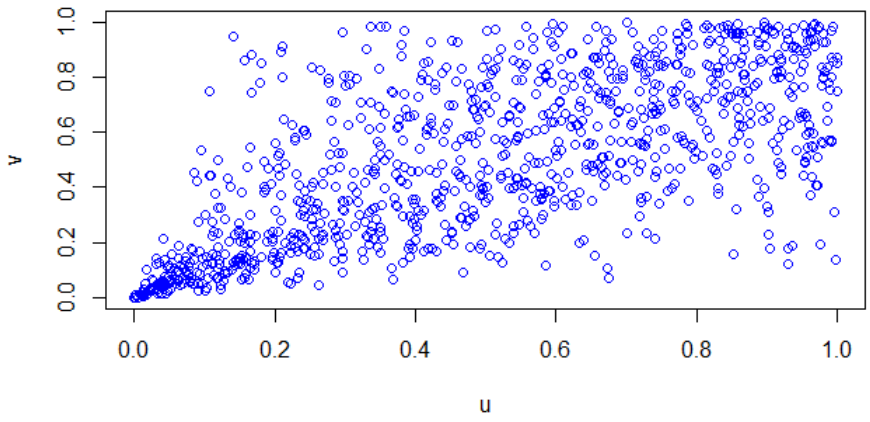


FIGURE 2.2 – Simulation de la densité de la copule de Clayton,$\theta = 2$

2^{eme} *étape :*

Notre but sera de se ramener à un nombre de points plus important en $(1,1)$*. On utilise la structure de dépendance de Clayton et on pose :*

- $U_1 = 1 - U$
- $V_1 = 1 - V$

On simule alors $(1-U, 1-V)$*, structure de dépendance de Clayton (avec changement de variable). On note la densité associée :*

$$c_{chang.var}(x,y) = (1+\theta) * ((1-x)*(1-y))^{-(1+\theta)} * (-1+(1-x)^{-\theta}+(1-y)^{-\theta})^{(-\frac{1+2*\theta}{\theta})}$$

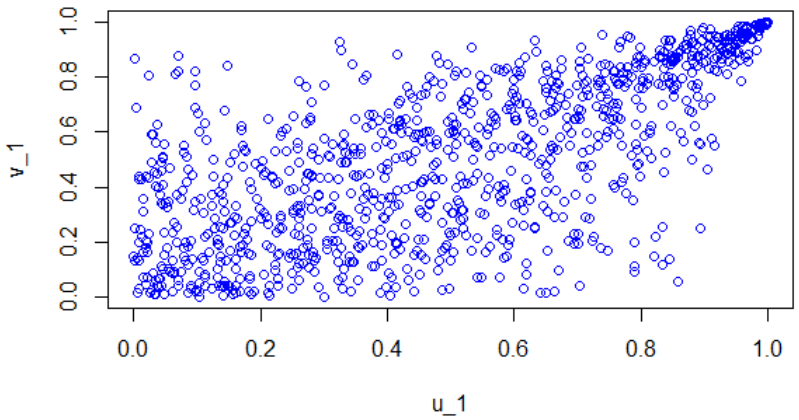


FIGURE 2.3 – Simulation de la densité $c_{chang.var}, sur[0,1]$ X $[0,1]$ $\theta = 2$

3^{eme} *étape :*

On souhaite ramener nos résultats au support de notre fonction f. Sur R, nous posons :

- $U_2 = qunif(U_1, 2, 4)$
- $V_2 = qunif(V_1, 0, 1)$

Et on obtient le plot suivant :

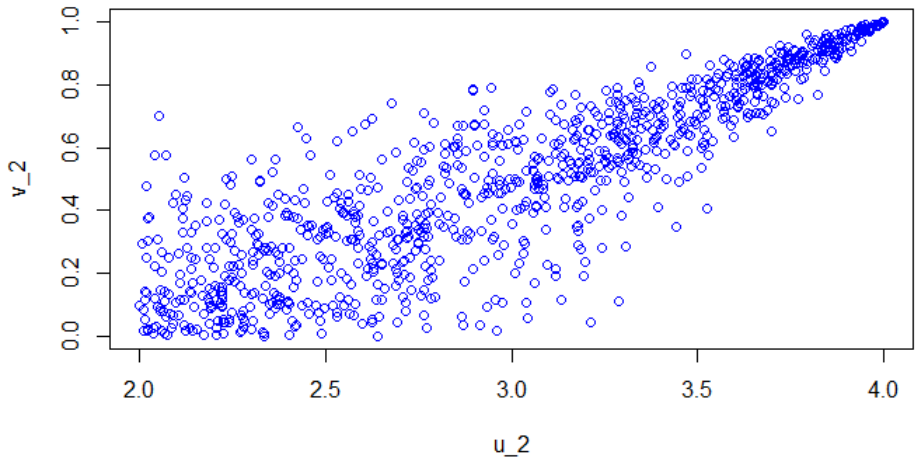


FIGURE 2.4 – Simulation de la densité $c_{chang.var}, sur[2,4]$ X $[0,1]$ $\theta = 2$

Tous les éléments sont réunies afin de simuler la méthode 4. Celle-ci se fera pour 5 valeurs de $\theta = (1,2,3,4,5)$ *à partir desquelles nous choisirons la meilleure estimation de I. Ci-joint, le graphe en question obtenu grâce à la fonction cumsum() ;*

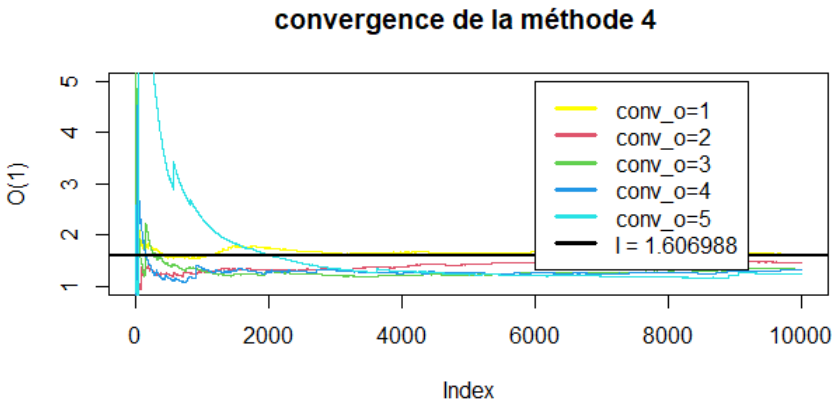


FIGURE 2.5

Nous voyons bien que la convergence de l'estimaion pour $\theta = 1$*est la meilleure.*

Cette méthode se révèle moins efficace en utilisant les marges de la méthode 3 :

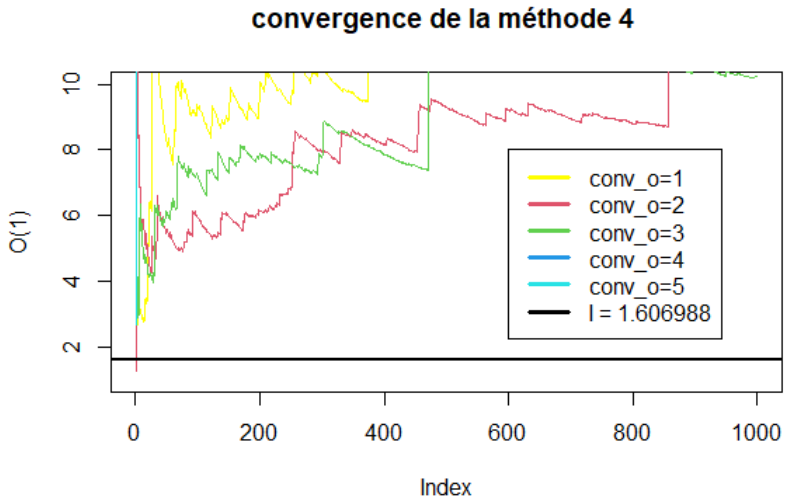


FIGURE 2.6

CONCLUSION

La méthode de Monte-Carlo est un outil efficace pour approximer la valeur d'intégrales "nontriviales". Elle se révèle plus efficace, lorsque l'on arrive à réduire la varaiance de l'estimateur choisi, comme il a été le cas lors de la partie I. L'ensemble des méthodes pour estimer notre intégrale I converge, donnant une approximation plus précise pour certaines et entraînant par le biais de réduction de variance un coup de simulation moins important.

Bibliographie

[1] T.JEANTHEAU, cours Simulation et Copules, Université Gustave Eiffel(2021)