Projet - Statistiques

Méthodes capture-marquage-recapture

Par LIAO Puwei et HATON Romain

21 mai 2021

Table des matières

1 Introduction **2**

2 Simulation d'après l'inverse de fonction répartition **2**

3 Un nombre d'individu connu **6**
3.1 Estimation de la capacité 6
3.2 Étude de la loi a posteriori 6

4 Simulation avec l'estimateur de Petersen **8**

5 Approche bayésienne **9**
5.1 Algorithme MCMC 10
5.2 Échantillonnage de Gibbs 11
5.3 Algorithme de Metropolis-Hastings 11
5.4 Choix du saut k 13

6 Conclusion **23**

1 Introduction

La méthode de capture-marquage-re-capture est utilisée dans le domaine de l'environnement et de la survie de la population étudiée. Le premier point de son principe est de capturer une partie des individus, de les marquer puis de les relâcher dans leur environnement d'origine. Son second point est de les re-capturer après un certain temps. Son but est, selon la proportion d'individus marqués dans la re-capture par rapport au nombre total de captures, d'estimer le nombre de la population totale dans cette environnement. Il s'agit de l'une des méthodes les plus courantes dans l'étude de la densité de population et elle est adaptée aux populations animales très mobiles et aux grands espaces.
Nous considérons le cas de 2 expériences successives de pêche, avec marquage et remise, réalisées afin d'estimer le nombre inconnu de poissons N dans un lac. On appelle C_1 et C_2 le nombre total de poissons capturés et marqués lors des pêches 1 et 2 respectivement. On appelle C_{20} le nombre de poissons non marqués capturés lors de la deuxième pêche et C_{21} le nombre de poissons marqués capturés lors de la deuxième pêche tel que $C_2 = C_{20} + C_{21}$. Les données disponibles proviennent d'une expérience réelle miniature de capture-marquage-re-capture réalisée par des étudiants à l'aide d'un saladier rempli de riz qui correspond au lac remplit d'eau et de haricots blancs qui représentent les poissons. Les données observées par les étudiants sont les suivantes : $C_1 = 125$, $C_{20} = 134$ et $C_{21} = 21$. On considère le modèle probabiliste $\mathcal{M}$ suivant :

$C_1 \sim \mathcal{B}(N, \pi)$
$C_{20}|C_1 \sim \mathcal{B}(N - C_1, \pi)$
$C_{21}|C_1 \sim \mathcal{B}(C_1, \pi)$

2 Simulation d'après l'inverse de fonction répartition

Dans un premier temps, nous écrivons le log-vraisemblance de $\log([C_1 = c_1, C_{20} = c_{20}, C_{21} = c_{21}|\pi, N])$, nous obtenons :

$$\log([C_1 = c_1, C_{20} = c_{20}, C_{21} = c_{21}|\pi, N]) =$$
$$\log(\binom{N}{c_1}\pi^{c_1}(1-\pi)^{N-c_1}\binom{N-c_1}{c_{20}}\pi^{c_{20}}(1-\pi)^{N-c_1-c_{20}}\binom{c_1}{c_{21}}\pi^{c_{21}}(1-\pi)^{c_1-c_{21}}) =$$
$$\log(\binom{N}{c_1}\binom{N-c_1}{c_{20}}\binom{c_1}{c_{21}}) + (c_1+c_2)\log(\pi) + (2N-c_1-c_2)\log(1-\pi)$$

Maintenant, nous écrivons l'inverse d'une fonction répartition d'après la définition, pour une loi binomiale de paramètres N et π, nous avons :

$$X \sim \mathcal{B}(N, \pi)$$
$$\forall k \in \mathbb{N},\ F_X(k) = \mathbb{P}(X = k) = \binom{N}{k}\pi^k(1-\pi)^{(N-k)}$$

$\forall u \in [0; 1]$, considérons X avec $\mathbb{P}(k) = \mathbb{P}(X = k)$ la construction $F_X^{-1}(u)$ est clairement donnée par :

$$x = k,\ \sum_{i=0}^{k-1}\mathbb{P}(i) < u \leq \sum_{i=0}^{k}\mathbb{P}(i),\ k \geq 1$$

pour une loi binomiale, nous avons :

$$X \sim \mathcal{B}(N, \pi)$$

$$F_X^{-1}(k) = \binom{N}{k} \pi^k (1-\pi)^{(N-k)}, \quad \text{avec} \quad 0 \leq k \leq n$$

En implémentant une fonction sur `R`, nous sommes capable de comparer les fréquences obtenues avec les fréquences théoriques, et nous pouvons remarquer, sur la figure suivante, que la courbe suit bien la forme de l'histogramme, nous pouvons dire qu'ils sont presque identique. Ici la courbe représente la densité d'une loi binomiale avec $N = 125$ et $\pi = 0.15$

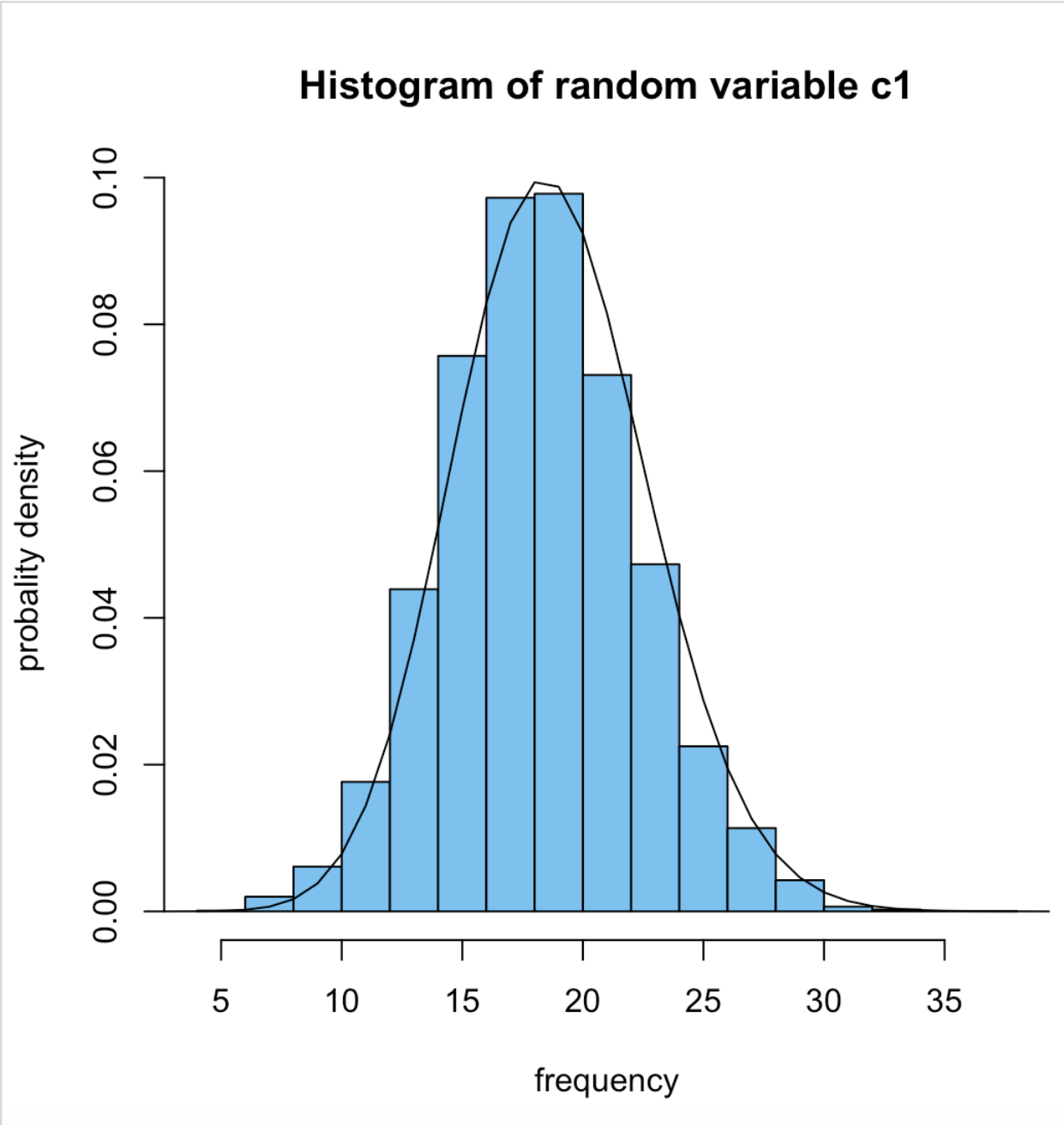

FIGURE 1 –

Maintenant, nous faisons la même expérience avec les 3 variables C_1, C_{20}, C_{21} suivant le modèle $\mathcal{M}$ en mettant les fréquences théoriques sous forme d'une courbe et nous observons, sur les 3 figures suivantes,

les mêmes résultats que précédemment, c'est-à-dire, que les fréquences observées sont très proche des fréquences théoriques.

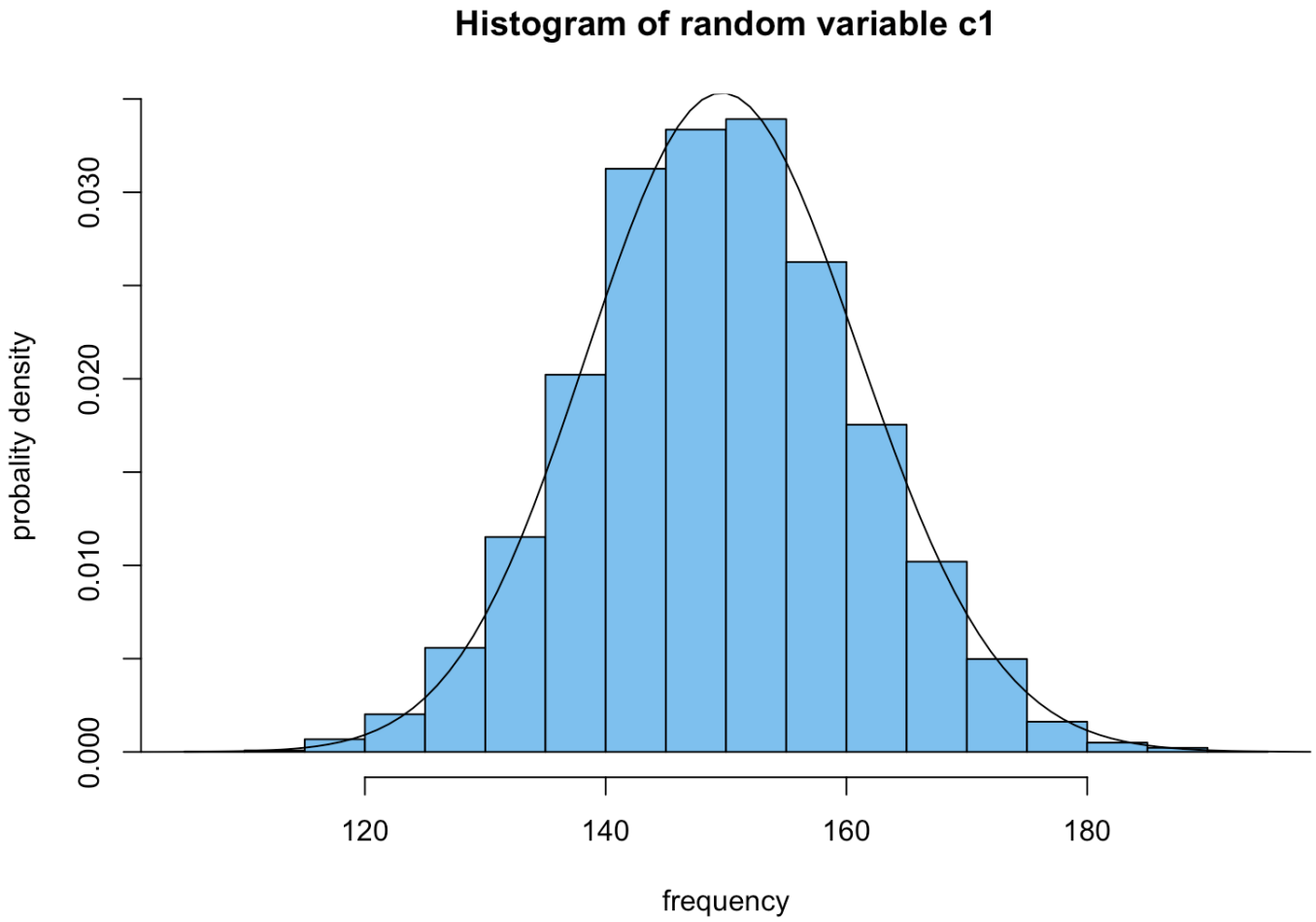

FIGURE 2 –

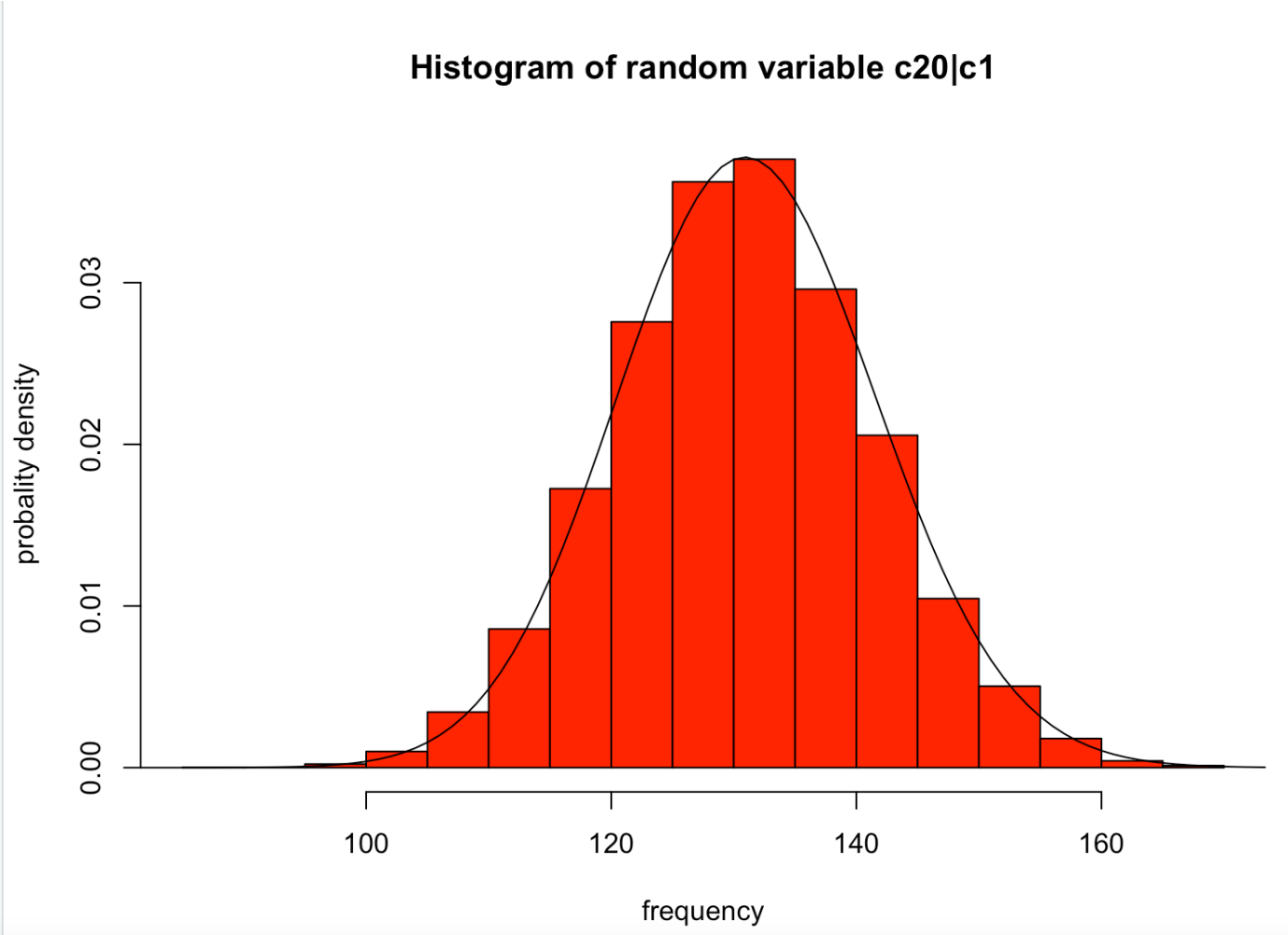
Histogram of random variable c20|c1
probality density
0.00
0.01
0.02
0.03
100
120
140
160
frequency

FIGURE 3 –

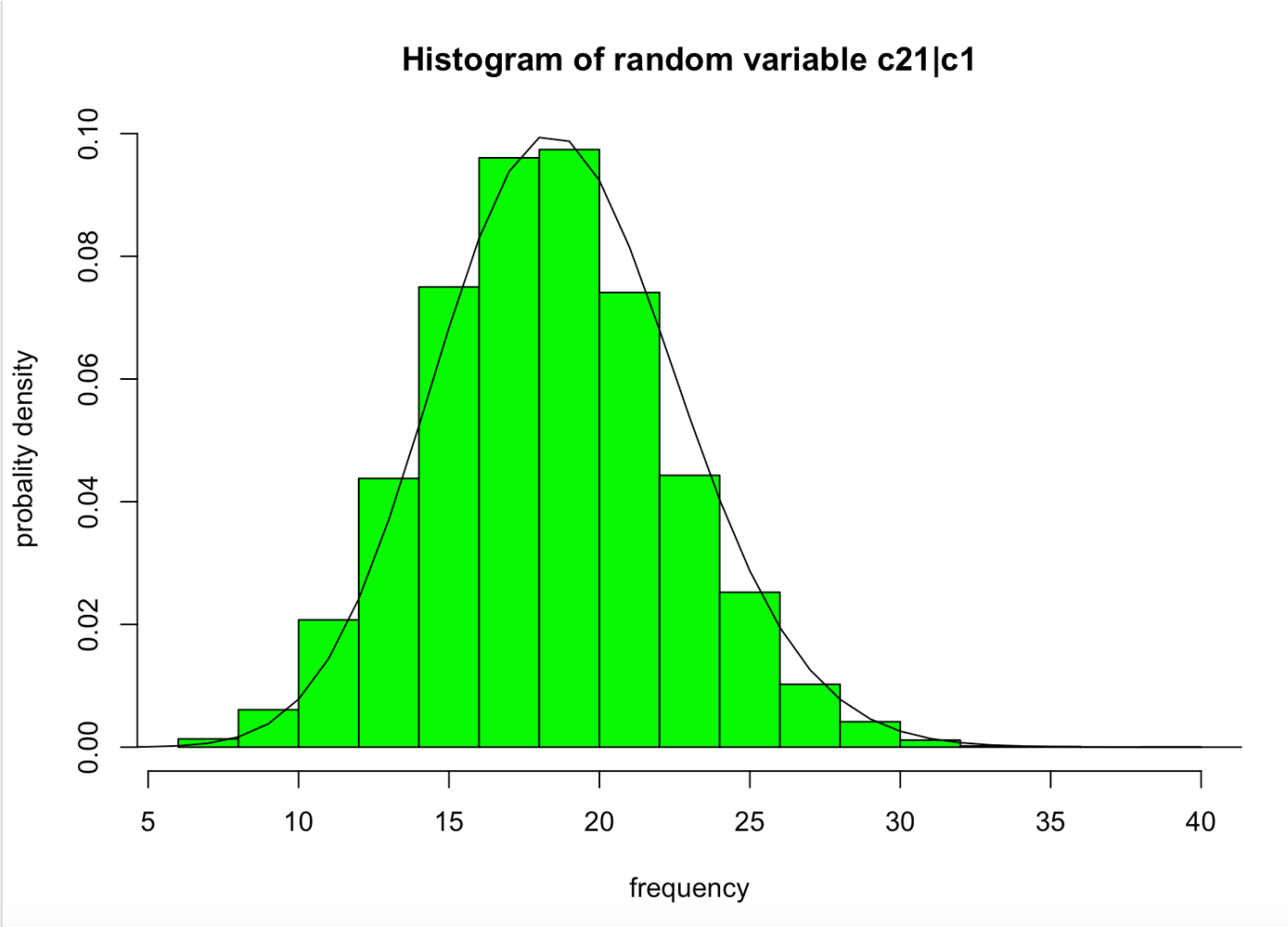

FIGURE 4 –

Nous remarquons que les histogrammes observés correspondent bien aux courbes théoriques.

3 Un nombre d'individu connu

3.1 Estimation de la capacité

Dans cette partie, nous supposons que nous connaissons le nombre d'individus $N = 950$. Pour faire l'estimation de la capacité, nous implémentons une fonction de la vraisemblance sur `R`, puis à l'aide de la fonction `nlm`, nous obtenons une estimation de la capacité π qui maximise la fonction de vraisemblance. Nous obtenons une estimation dans notre cas de $\hat{\pi}_{MLE} = 0.147368$.

3.2 Étude de la loi a posteriori

Dans cette partie, nous assignons une loi a priori beta(α, β) sur le paramètre d'efficacité π. Donc la loi a priori est une loi beta d'où la densité de probabilité s'écrit :

$$\frac{\pi^{\alpha-1}(1-\pi)^{\beta-1}}{\int_0^1 \pi^{\alpha-1}(1-\pi)^{\beta-1}d\pi}$$

Nous écrivons la vraisemblance, nous avons :

$$\binom{N}{C_1}\binom{N-C_1}{C_{20}}\binom{C_1}{C_{21}}\pi^{C_1+C_2}(1-\pi)^{2N-C_1-C_2}$$

La loi posteriori est proportionelle au produit de vraisemblance et de la loi priori, nous avons donc que la loi a posteriori est proportionnelle à :

$$\binom{N}{C_1}\binom{N-C_1}{C_{20}}\binom{C_1}{C_{21}}\pi^{C_1+C_2}(1-\pi)^{2N-C_1-C_2}\frac{\pi^{\alpha-1}(1-\pi)^{\beta-1}}{\int_0^1 \pi^{\alpha-1}(1-\pi)^{\beta-1}d\pi}$$

Ce qui est lui-même proportionnelle à :

$$\pi^{C_1+C_2+\alpha-1}(1-\pi)^{2N-C_1-C_2+\beta-1}$$

Donc la loi posteriori est proportionnelle à :

$$\pi^{(C_1+C_2+\alpha)-1}(1-\pi)^{(2N-C_1-C_2+\beta)-1}$$

On a

$$\mathcal{L}_{posteriori} = \frac{\pi^{(C_1+C_2+\alpha)-1}(1-\pi)^{(2N-C_1-C_2+\beta)-1}}{\int_0^1 \pi^{C_1+C_2+\alpha-1}(1-\pi)^{2N-C_1-C_2+\beta-1}}$$

D'après cela, nous avons bien montré que cette loi posteriori $\mathcal{L}_{posteriori}$ suit une loi gamma de paramètres $C_1 + C_2 + \alpha$ et $2N - C_1 - C_2 + \beta$
Nous calculons facilement son espérance et nous trouvons :

$$\mathbb{E}(X) = \frac{C_1 + C_2 + \alpha}{(C_1 + C_2 + \alpha) + (2N - C_1 - C_2 + \beta)}$$

$$= \frac{C_1 + C_2 + \alpha}{2N + \beta + \alpha}$$

Quand la valeur α augmente, l'espérance de cette loi augmente également et que la majeure partie de la distribution de probabilité se déplacera vers la droite. Alors qu'une augmentation de β faut diminuer l'espérance et déplacera la distribution vers la gauche.
Avec les résultats que nous avons obtenu précédemment, nous décidons de représenter sur un même graphique le MLE que nous fixons à 0.15, la loi a prior, la loi a posteriori et la vraisemblance :

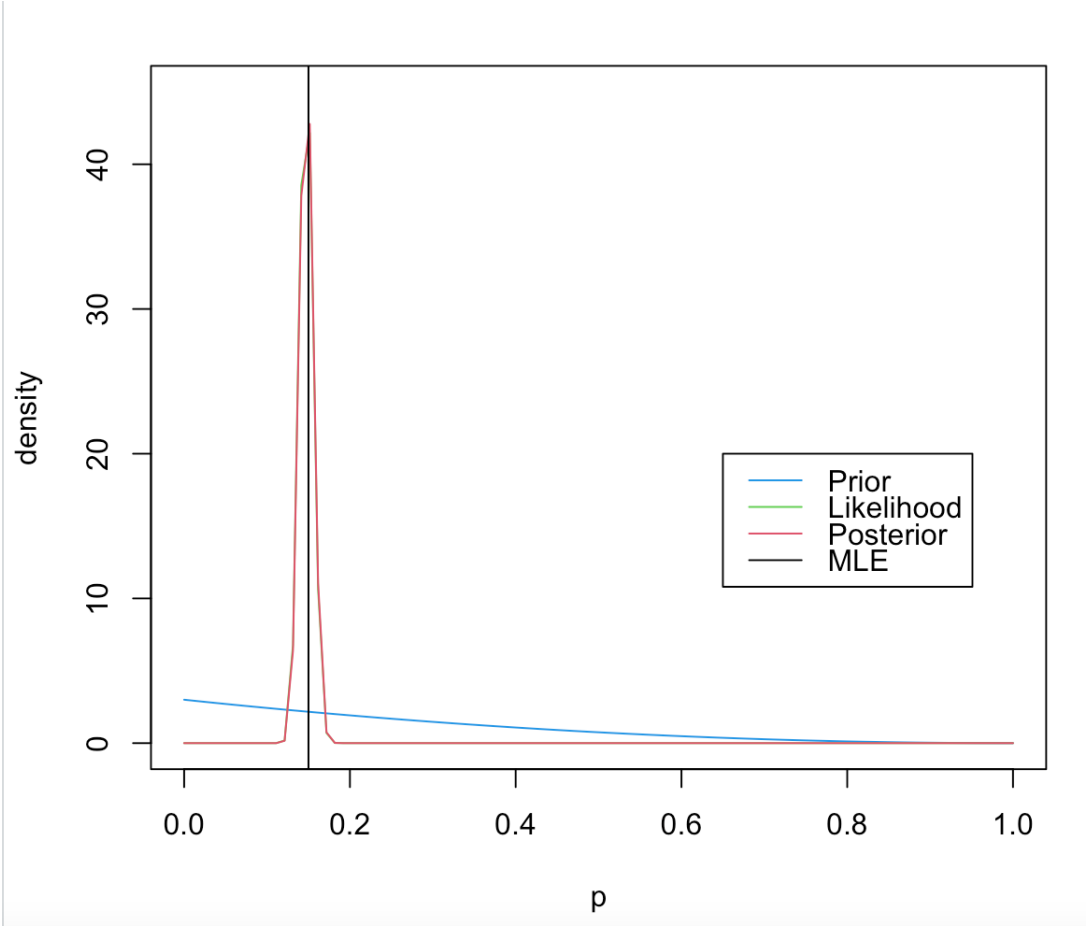

FIGURE 5 –

Nous constatons que la vraisemblance et la loi a posteriori coïncident, donc nous pouvons dire que la loi a posteriori est identique à la fonction de vraisemblance. Le pic est obtenu sur la valeur de l'estimation du maximum de vraisemblance.

4 Simulation avec l'estimateur de Petersen

Pour identifier le nombre d'individus N dans une population d'intérêt à partir de deux expériences de pêche de type capture-marquage-re-capture, nous définissons l'estimateur fréquentiste Petersen par :

$$\hat{N} = \frac{C_1 C_2}{C_{21}}$$

Dans notre cas, nous trouvons que le nombre d'individus estimé est de 922.619048. Ce qui revient à 923 poissons dans le lac.

Nous supposons que les vraies valeurs des paramètres soient $N_{true} = 923$ et $\pi_{true} = 0.15$. À partir de ces vraies valeurs, nous simulons 100 jeux de données à l'aide de la fonction implémentée, et d'après cela nous pouvons en déduire 100 estimations du paramètre N :

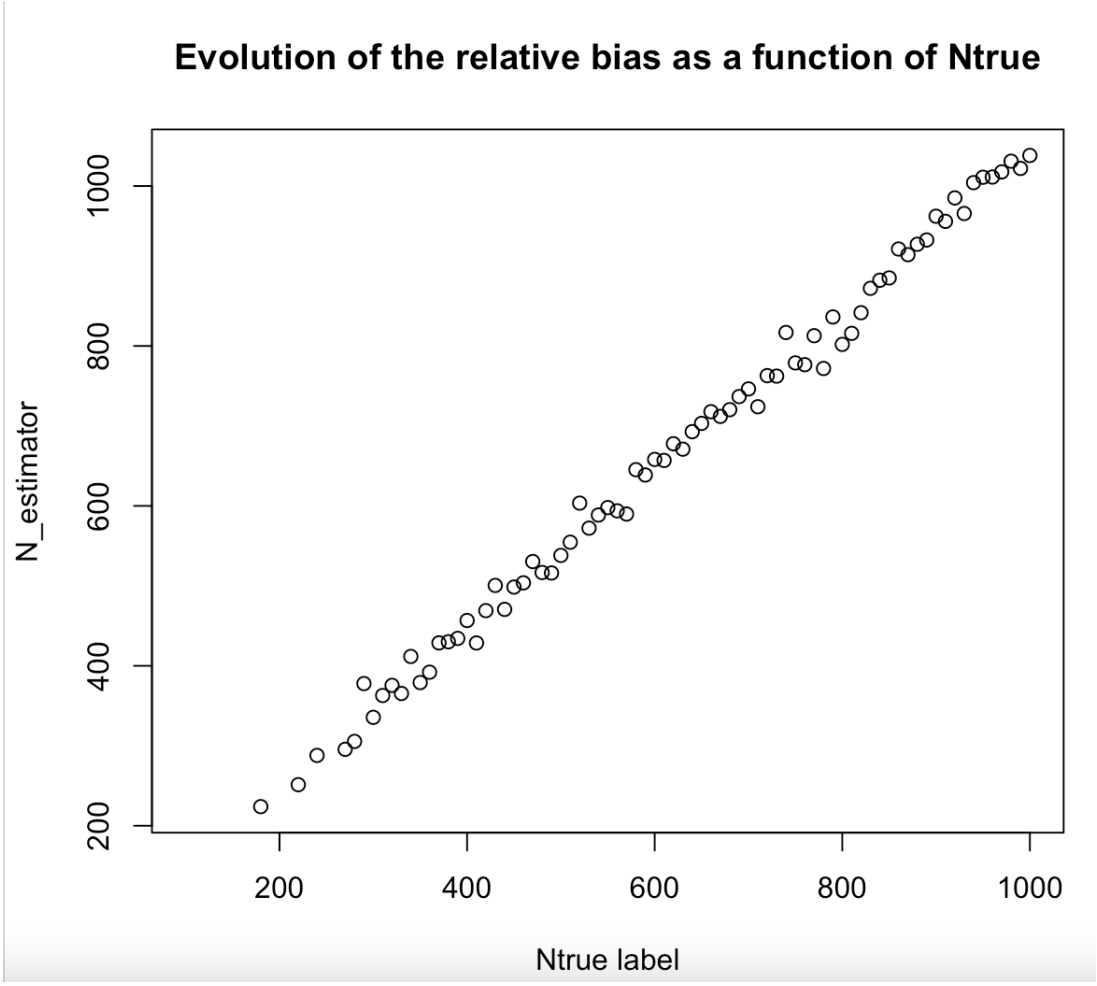

FIGURE 6 –

Nous avons obtenu le graphique ci-dessus de N_{true} variant de 100 à 1000 par pas de 10. Dans cette figure de N, estimateur de Monte-Carlo et de N_{true}, nous avons obtenu une ligne droite, c'est-à-dire que nous avons toujours une petite erreur et standard qui ne change pas beaucoup. Nous pouvons donc dire que cet estimateur est qualifié d'après ce graphe.

5 Approche bayésienne

Dans cette partie, pour aller plus loin dans notre analyse, considérons que la loi priori est une loi beta de paramètres $\alpha = 1$ et $\beta = 3$. Nous cherchons l'expression de la loi conditionnelle complète de N. La loi a priori est uniforme U, nous connaissons déjà que le vraisemblance est :

$$\binom{N}{C_1}\binom{N-C_1}{C_{20}}\binom{C_1}{C_{21}}\pi^{C_1+C_2}(1-\pi)^{2N-C_1-C_2}$$

La loi a posteriori est toujours proportionnelle au produit de vraisemblance et à la loi a priori :

$$\binom{N}{C_1}\binom{N-C_1}{C_{20}}\binom{C_1}{C_{21}}\pi^{C_1+C_2}(1-\pi)^{2N-C_1-C_2}U$$

Ce qui est proportionelle à

$$\binom{N}{C_1}\binom{N-C_1}{C_{20}}\binom{C_1}{C_{21}}\pi^{C_1+C_2}(1-\pi)^{2N-C_1-C_2}$$

Ce qui est proportionelle à

$$\frac{N!}{N-C_1-C_2!}\pi^{C_1+C_2}(1-\pi)^{2N-C_1-C_2}$$

Ce qui est proportionelle à

$$\frac{N!}{(N-C_1-C_2)!}\beta(C_1+C_2+1, 2N-C_1-C_2+1)$$

On a alors que

$$\mathcal{L}_{posteriori} = K\frac{N!}{(N-C_1-C_2)!}\beta(C_1+C_2+1, 2N-C_1-C_2+1)$$

5.1 Algorithme MCMC

Dans cette partie, nous nous consacrerons aux algorithmes MCMC qui échantillonne dans la loi jointe à posteriori. Nous implémentons deux algorithmes respectivement, un par échantillonnage de Gibbs, et l'autre par l'algorithme de Metropolis-Hastings. Nous constatons d'abord un 3D-plot pour le loi joint du couple (π, N) :

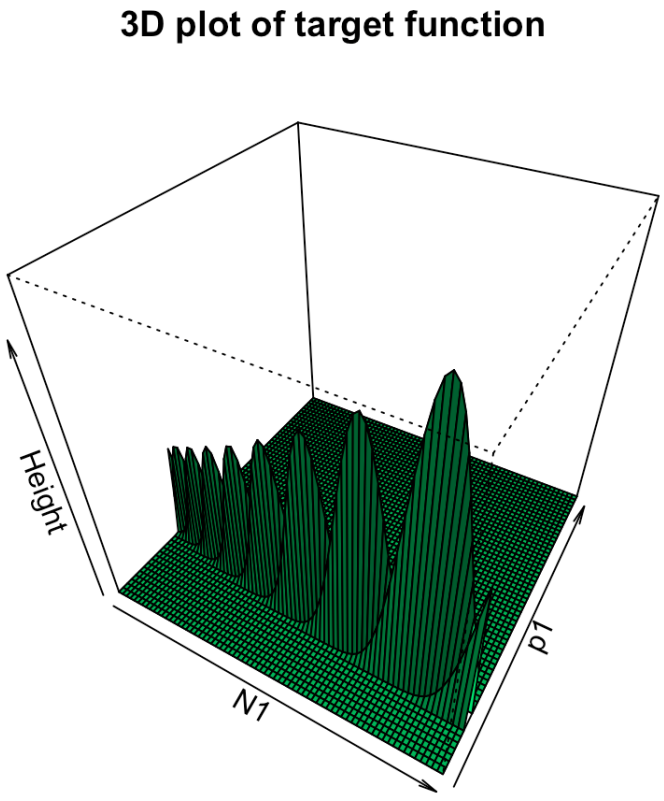

FIGURE 7 –

5.2 Échantillonnage de Gibbs

Nous commençons par construire une fonction qui utilise l'algorithme de Gibbs pour mettre à jour l'estimation de l'efficacité π L'échantillonnage de Gibbs permet d'estimer π en faisant varier le nombre d'échantillage :

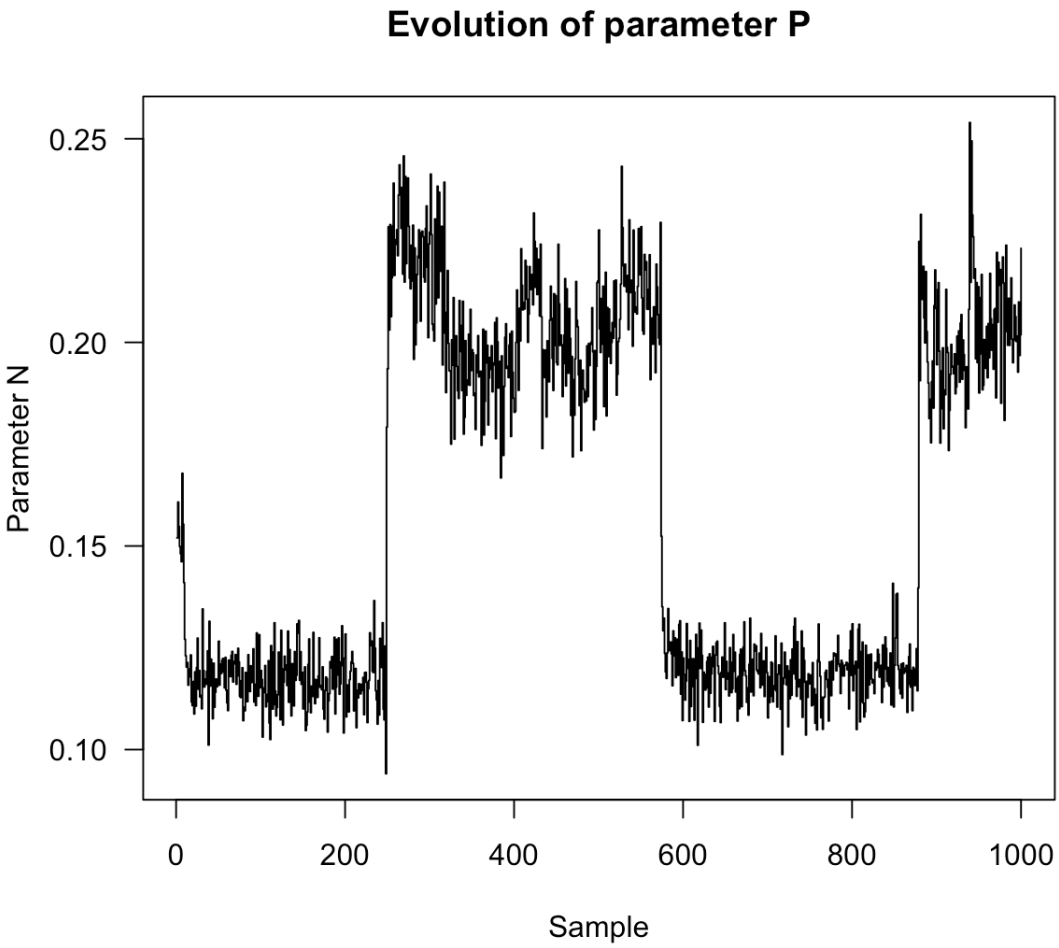

FIGURE 8 –

Nous constate que l'estimation de l'efficacité π est autour de 0.12 ou 0.21.

5.3 Algorithme de Metropolis-Hastings

Ensuite, nous faisons une fonction qui utilise l'algorithme de Metropolis-Hastings pour estimer l'estimation de nombre total d'individu N. Nous traçons les deux graphiques suivants :

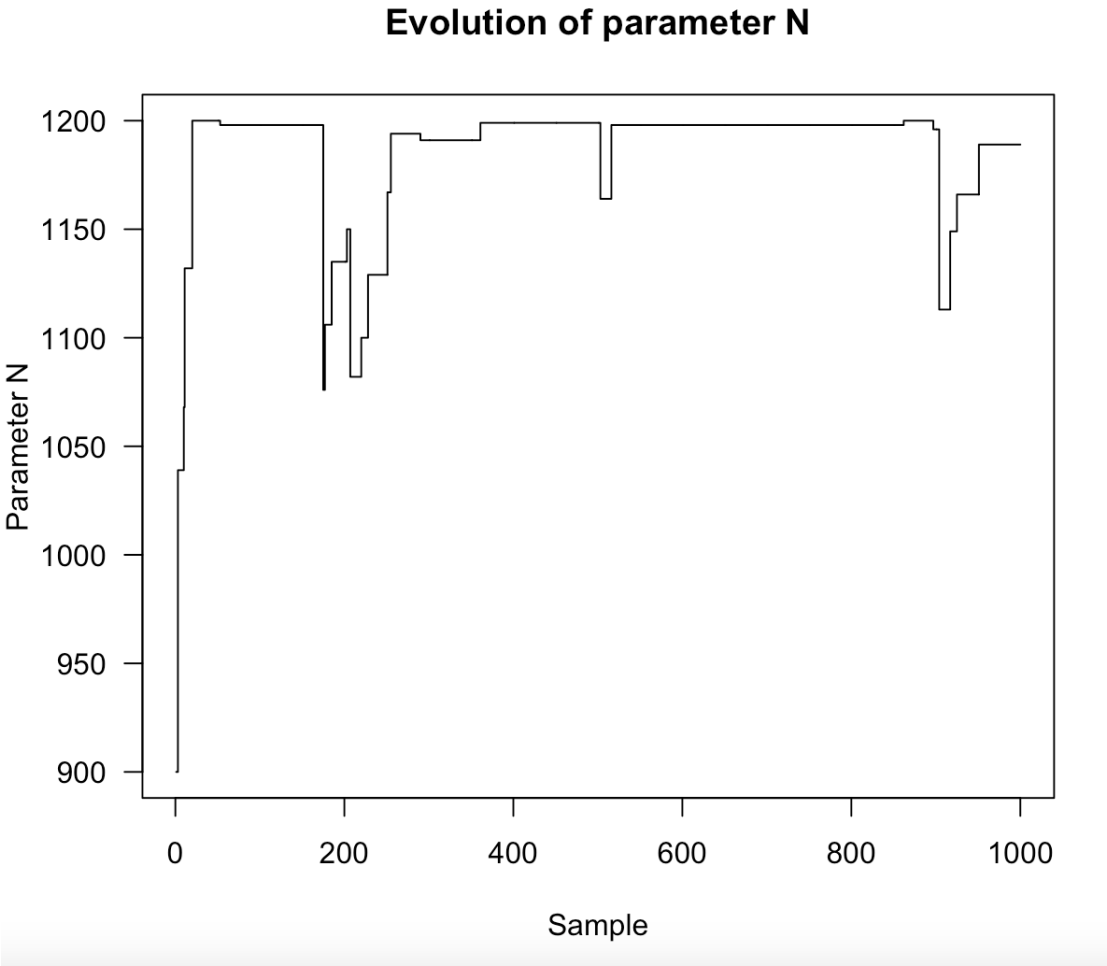
Evolution of parameter N
Parameter N
1200
1150
1100
1050
1000
950
900
0
200
400
600
800
1000
Sample

FIGURE 9 –

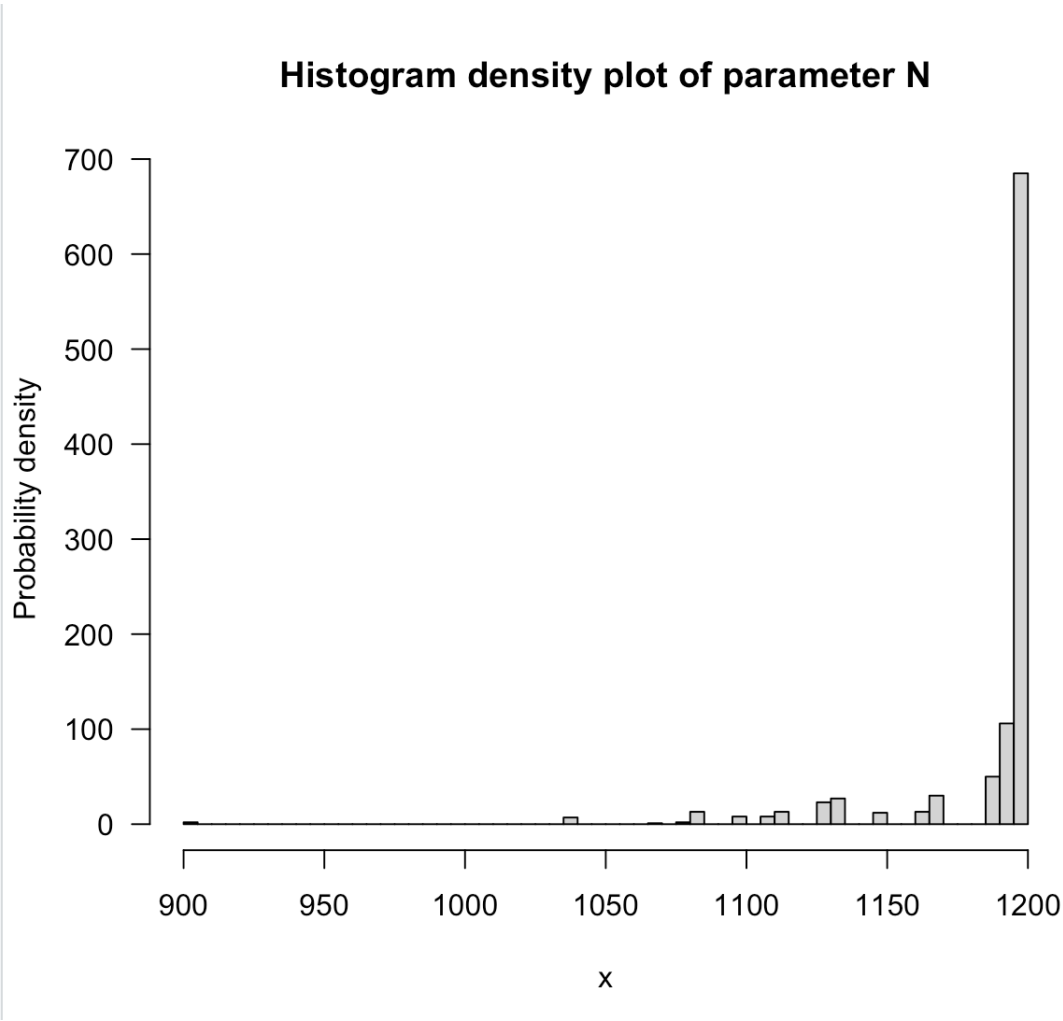

FIGURE 10 –

D'après ces 2 figures, le premier fait varier le nombre d'échantillons et l'autre calcule la probabilité de densité du nombre total d'individu. Nous remarquons, sur les deux graphes, que le nombre total de poissons est au alentour de 1200 poissons. Nous pouvons supposer que le nombre total de poisson est d'environ 1200 dans ce lac.

5.4 Choix du saut k

Pour commencer, nous fixons le nombre de poissons $N = 900$ et $\pi = 0.1$ pour initialiser les valeurs. Dans cette partie, nous souhaitons choisir le saut k d'une façon logique. En utilisant la fonction qui utilise l'algorithme MCMC précédemment, nous pouvons tracer l'évolution du taux d'acceptation associée de la mise à jour du nombre N en fonction de différentes valeurs du paramètre du saut k. Maintenant, pour chaque valeur de k, nous pouvons faire tourner cette fonction pendant 10 000 itérations avec une seule chaîne de Markov, c'est l'étape de calibration. Avec la figure suivante, nous pouvons choisir le meilleur saut k :

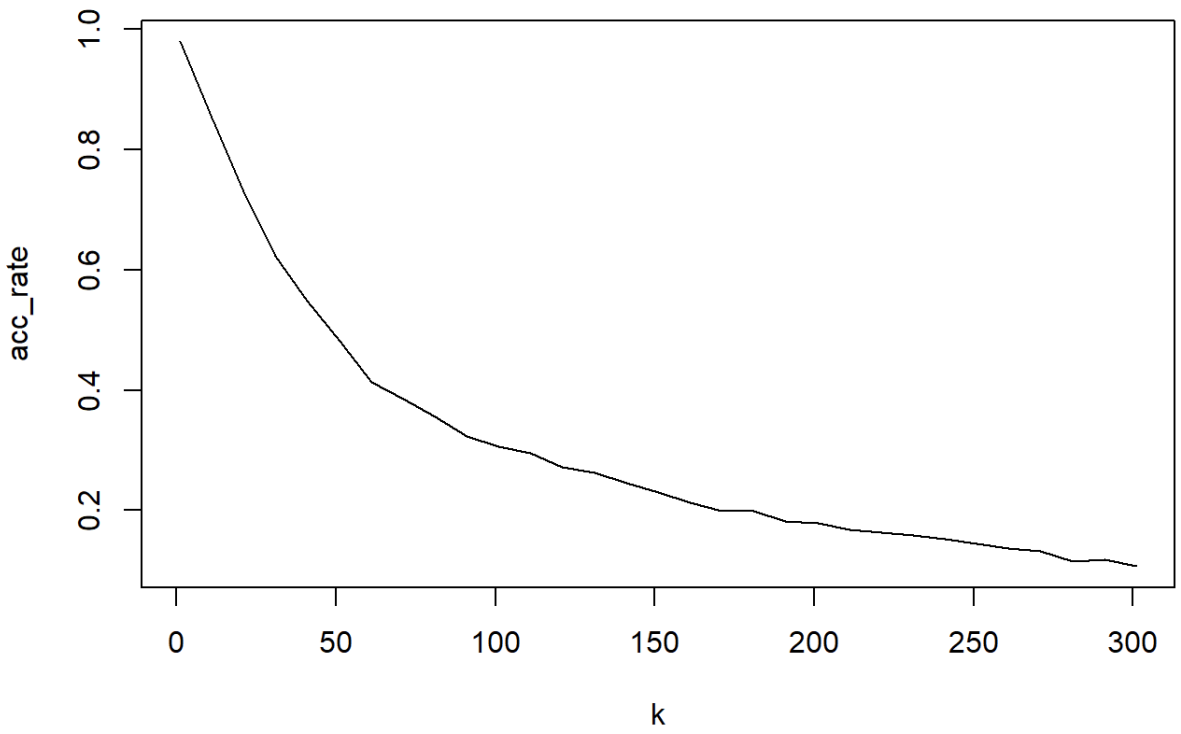

FIGURE 11 –

Pour un taux d'acceptation d'environ 40%, nous pouvons considérer le saut optimal $k = 71$. Nous garderons cette valeur de k dans la suite.

Une fois le meilleur saut k obtenu, nous traçons l'évolution du nombre d'individu N de trois chaînes de Markov. Visuellement, nous pouvons dire, qu'en fonction du paramètre N et π, les chaînes se concentrent autour de valeur différentes. Nous pouvons calculer la statistique de Gelman-Rubin, qui nous permettra de trouver la convergence de l'algorithme MCMC. Cet statistique analyse la différence entre les chaînes de Markov.

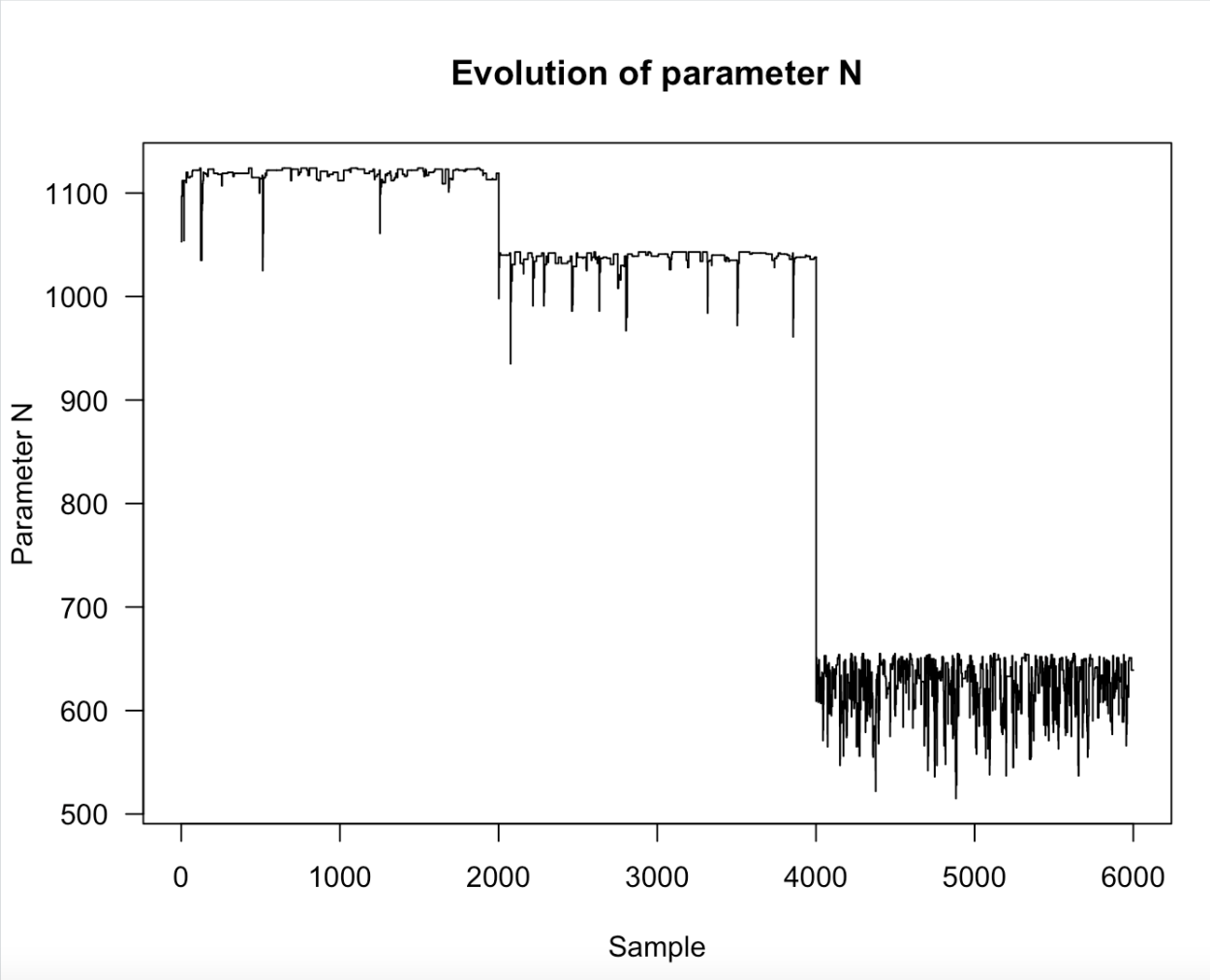
Evolution of parameter N
Parameter N
1100
1000
900
800
700
600
500
0
1000
2000
3000
4000
5000
6000
Sample

FIGURE 12 –

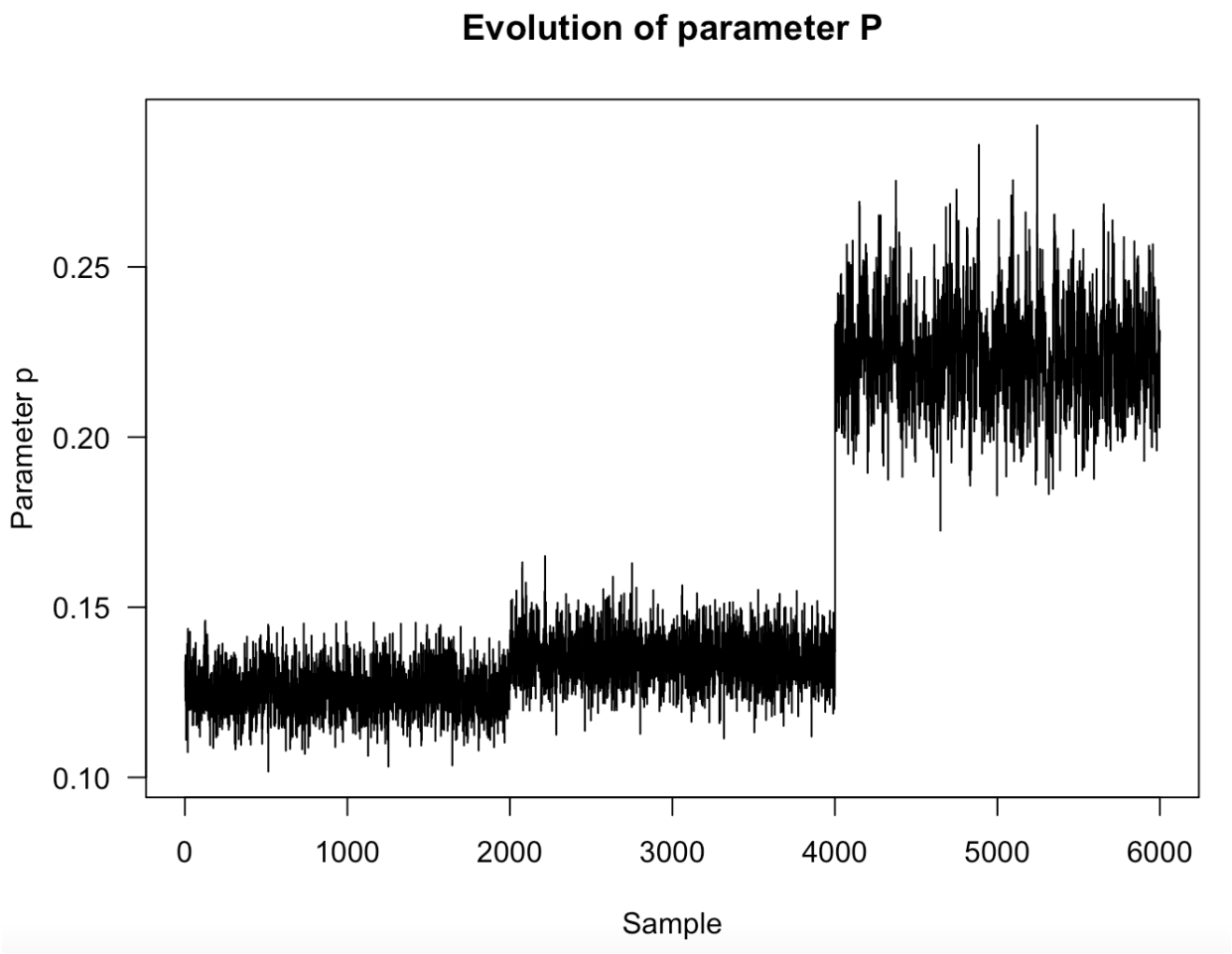

FIGURE 13 –

Nous utilisons la statistique de Gelman-Rubin pour identifier la convergence. Nous traçons les auto-corrélations intra-chaînes, nous avons les graphiques suivants :

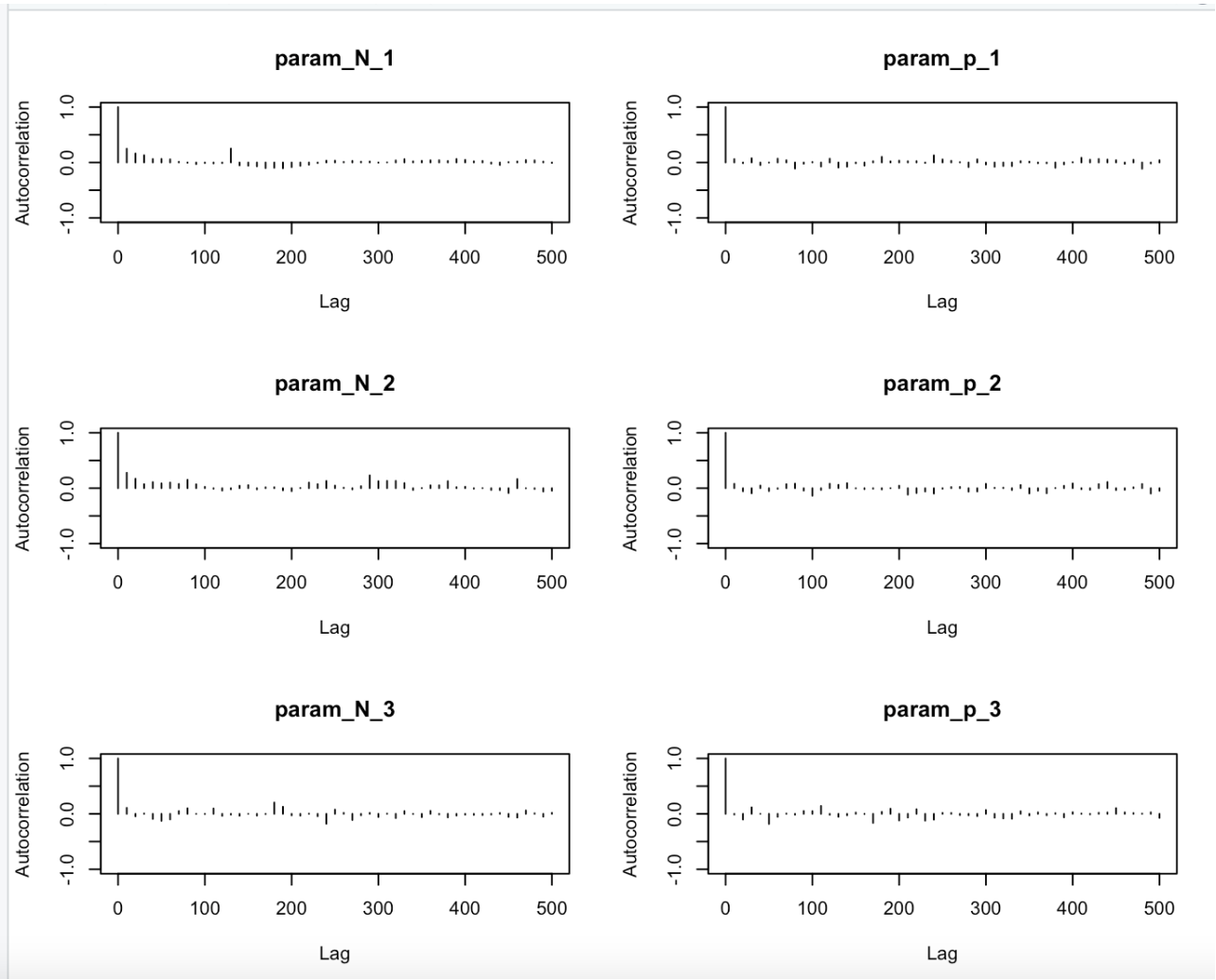

FIGURE 14 –

D'après ces graphes, nous pouvons dire que les résultats de l'estimation du nombre total d'individu N et de lefficacité π ne sont pas corrélés au mesure du nombre d'échantillonnage.

En supprimant les $X = 2000$ premières itérations correspondant au temps de chauffe estimé, nous pouvons retrouver l'évolution des paramètres N et π. Nous obtenons les figures suivantes :

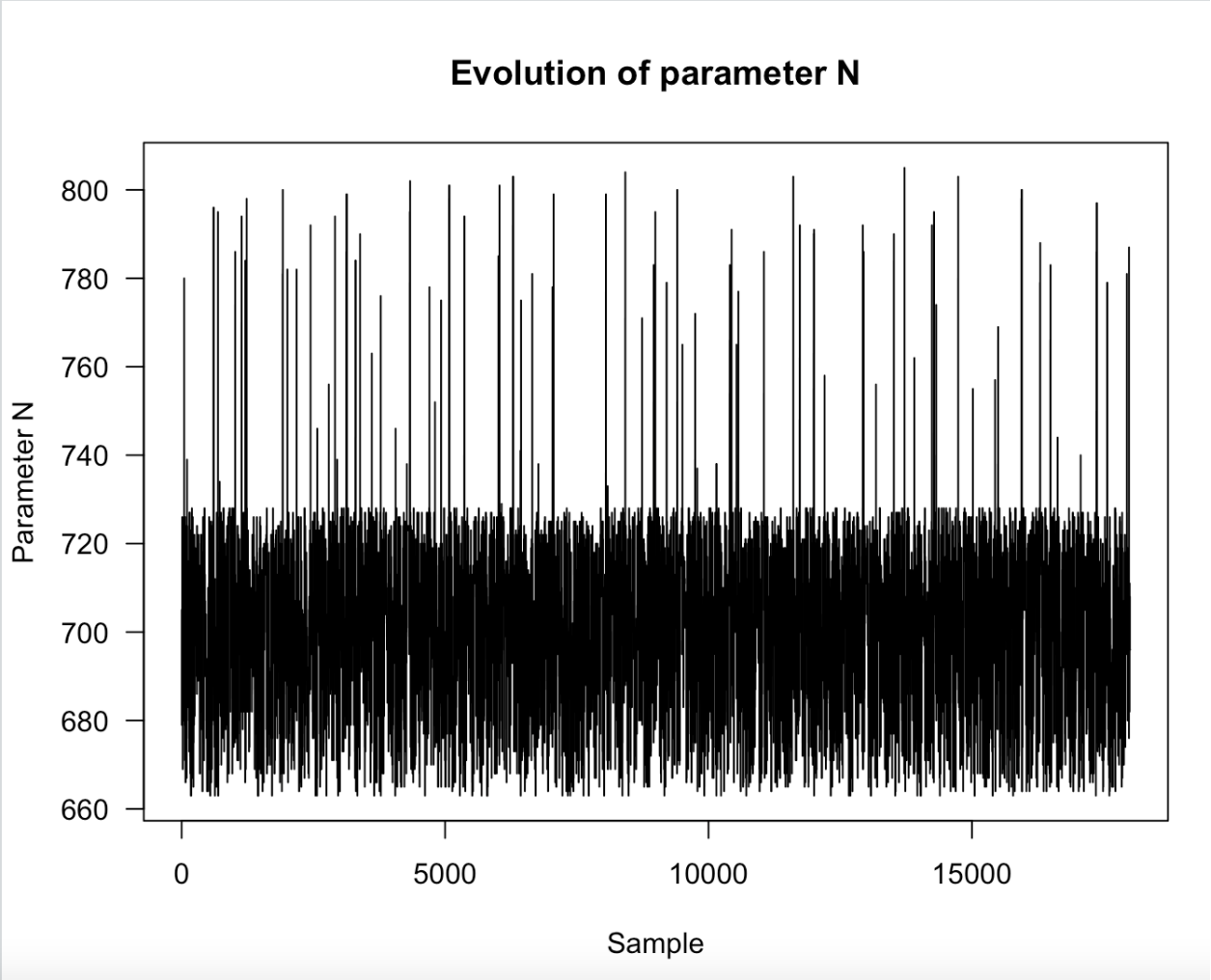
Evolution of parameter N
Parameter N
800
780
760
740
720
700
680
660
0
5000
10000
15000
Sample

FIGURE 15 –

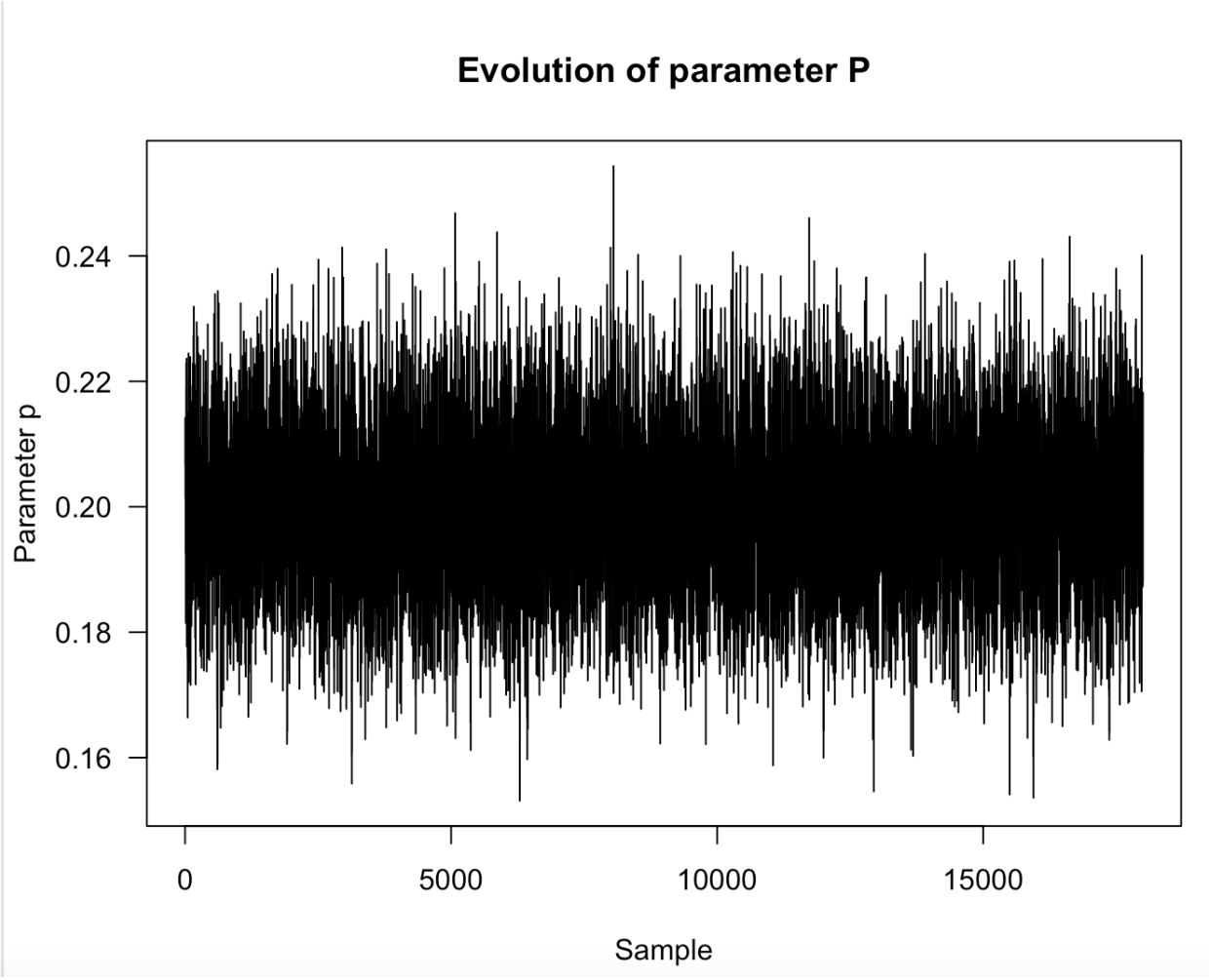
Evolution of parameter P
Parameter p
0.24
0.22
0.20
0.18
0.16
0
5000
10000
15000
Sample

FIGURE 16 –

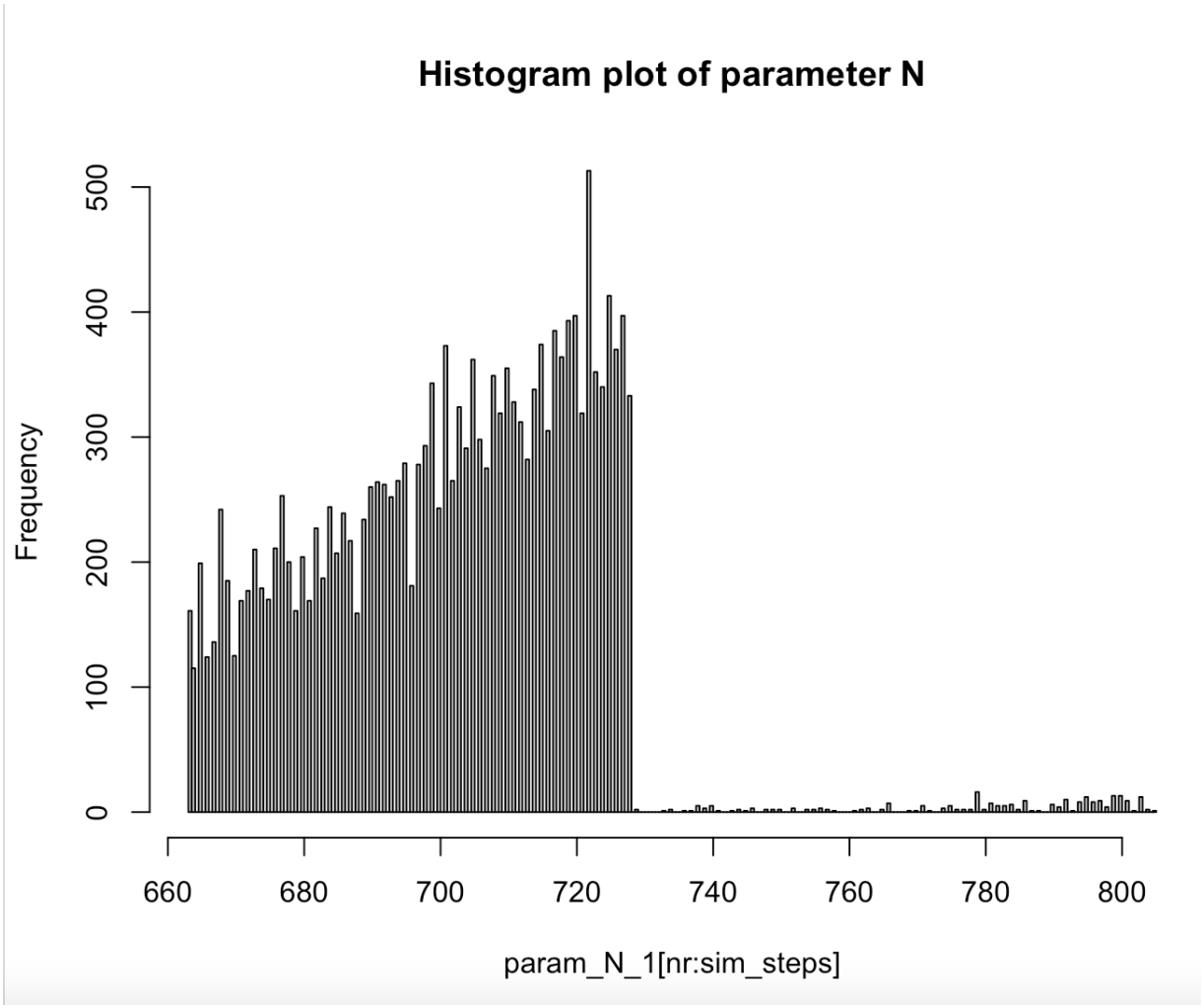
Histogram plot of parameter N
Frequency
0
100
200
300
400
500
660
680
700
720
740
760
780
800
param_N_1[nr:sim_steps]

FIGURE 17 –

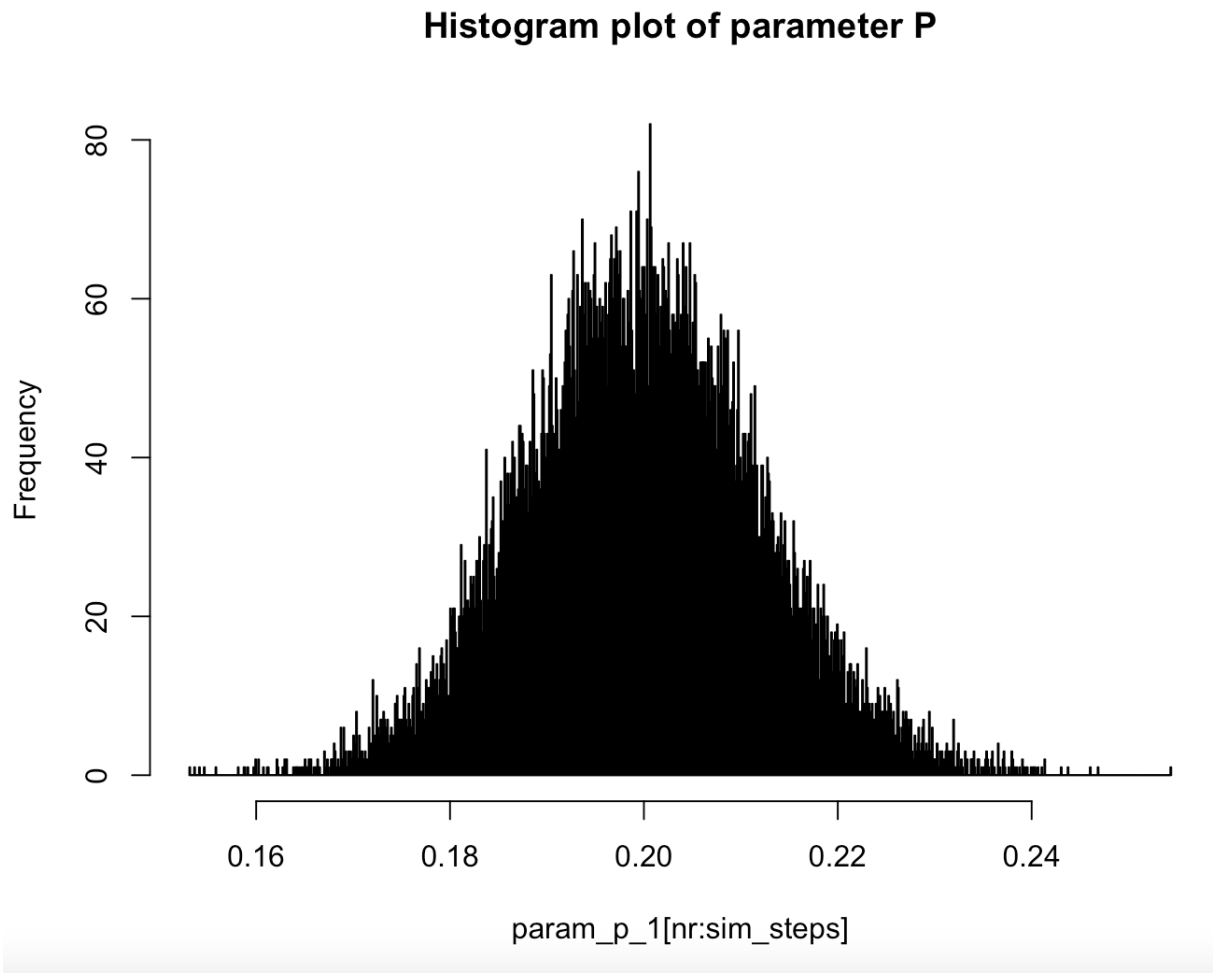

FIGURE 18 –

Nous trouvons que les estimations de N et π sont environ de 700 et 0.2 respectivement. Nous pouvons donc, pour le moment, fixer le nombre total de poissons N à 700 et l'efficacité π à 0.2.

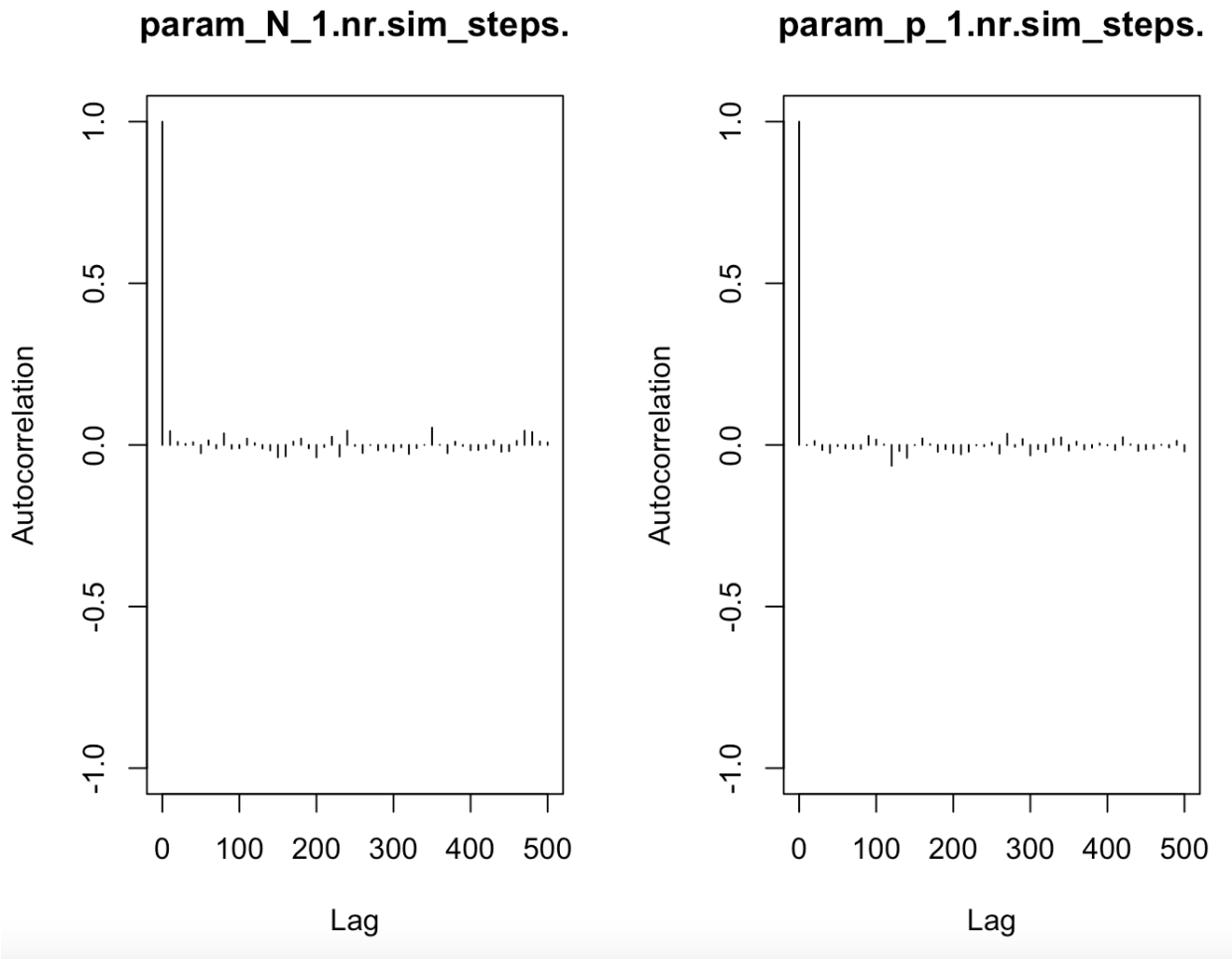

FIGURE 19 –

Nous pouvons également tracer les auto-corrélations pour ces deux estimations. Nous voyons ici que ce n'est pas corrélées au mesure de l'échantillonnage.

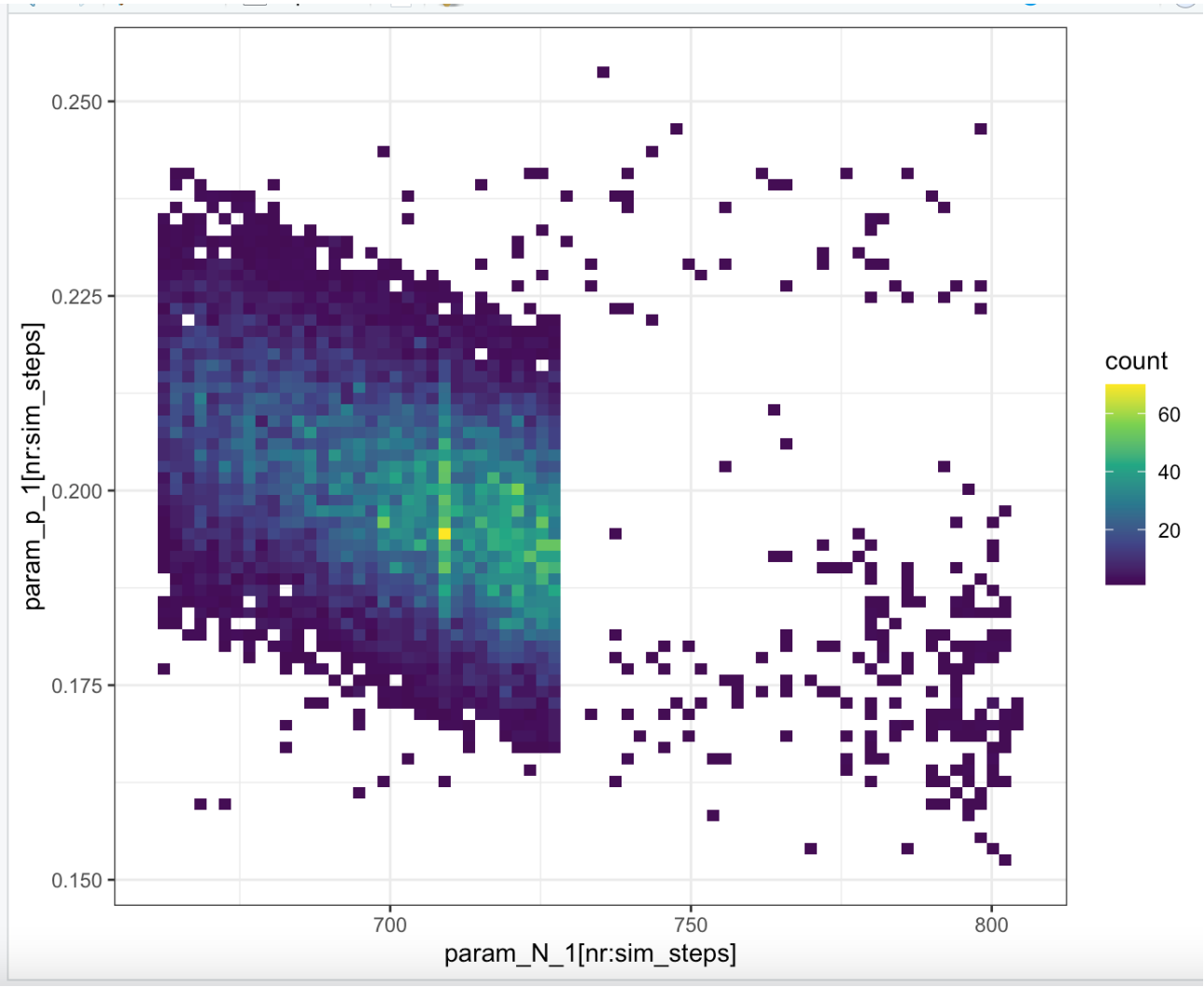

FIGURE 20 –

D'après cette graphe, nous pouvons dire qu'il y a de grandes probabilités pour que le nombre total de poissons soit d'environ 700 et que l'efficacité de la pêche soit d'environ 0.2.

6 Conclusion

Le but de ce projet est d'estimer le nombre total de poissons dans le lac et l'efficacité de la pêche. Pour commencer, nous avons supposé que nous connaissions le nombre de poissons. Cela nous a permit d'estimer le paramètre π qui correspond à l'efficacité de la pêche. Nous l'avons estimer par maximum de vraisemblance. Grâce à cet estimation, nous avons pu faire des simulations qui nous ont montré que la vraisemblance était la même que le loi a posteriori avec un pic qui correspond au maximum de vraisemblance.
Dans la partie suivante, nous avons supposer qu'aucun des paramètres n'étaient connus et nous avons procédé par une approche fréquentiste. Cela nous utile pour calculer l'estimation de Peterson.
Puis nous avons simulé plusieurs jeux de données avec, comme supposition que, $N_{true} = 923$ et $\pi_{true} = 0.15$.
Ensuite, nous avons procédé par une approche bayésienne. Cela nous a permit d'utiliser les algorithmes de Gibbs et de Metropolis-Hastings. Le premier nous a été utile pour estimer le paramètre π et le second le nombre de poissons N.

Enfin, nous avons tester ces valeurs sur trois différentes chaînes de Markov et nous avons observé que les résultats n'étaient pas les mêmes selon la chaîne de Markov. Nous avons aussi calculé les auto-corrélations pour aller plus loin dans notre comparaison.
Les approches utilisées ont été concluantes au vu des bons estimations que nous avons trouvé.