Algorithme de Metropolis-Hastings et méthodes de Monte Carlo

Cet article vise à présenter avec mes mots l'algorithme de Metropolis, et sa variante l'algorithme de Metropolis-Hastings.

Pour présenter rapidement ces algorithmes, ce sont des algorithmes issus de la famille des méthodes de Monte Carlo. Pour faire court, les méthodes de Monte Carlo sont des méthodes permettant d'estimer une certaine quantité en se basant sur des techniques probabilistes (d'où le nom de « Monte Carlo » en référence aux jeux de hasard pratiqués dans ces casinos). Il en existe pléthore, comme l'échantillonnage de Gibbs, le Random Walk MH Sampler, …

Dans notre cas, nous allons utiliser ce type d'algorithme d'abord pour échantillonner à partir d'une fonction de densité, puis pour estimer les paramètres optimaux d'une loi de probabilité pour que celle-ci colle le mieux possible à nos données, où plus mathématiquement, qui maximise la vraisemblance. Si cela n'est pas encore très clair, cela prendra du sens au fil de la lecture. Cependant, avant de parler de l'algorithme, commençons par 2 petits rappels pour mieux comprendre le travail réalisé.

Je précise que le code ayant permis l'ensemble des calculs et des graphiques réalisés au sein de cet article est présent sur mon GitHub.

Baptiste Gautier

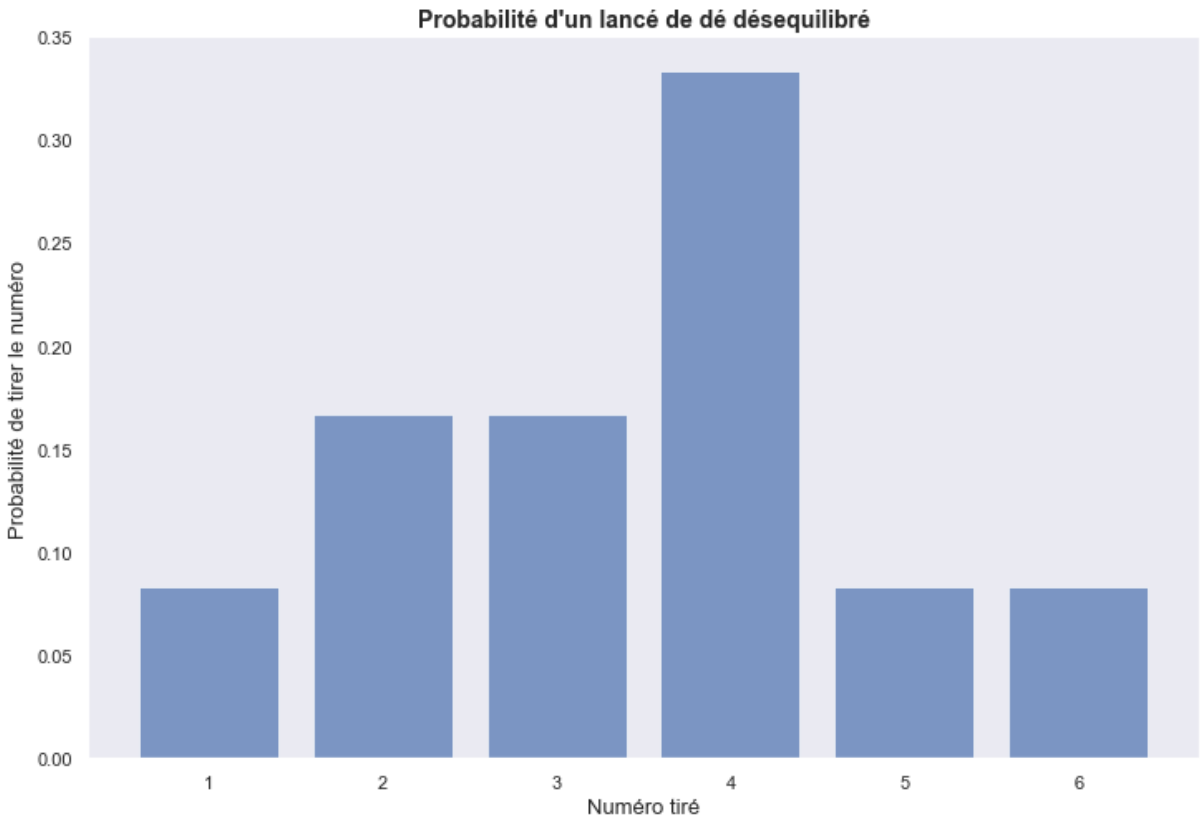
Table des matières

Rappel

a) Probabilité discrète et probabilité continue

i) Les probabilités discrètes

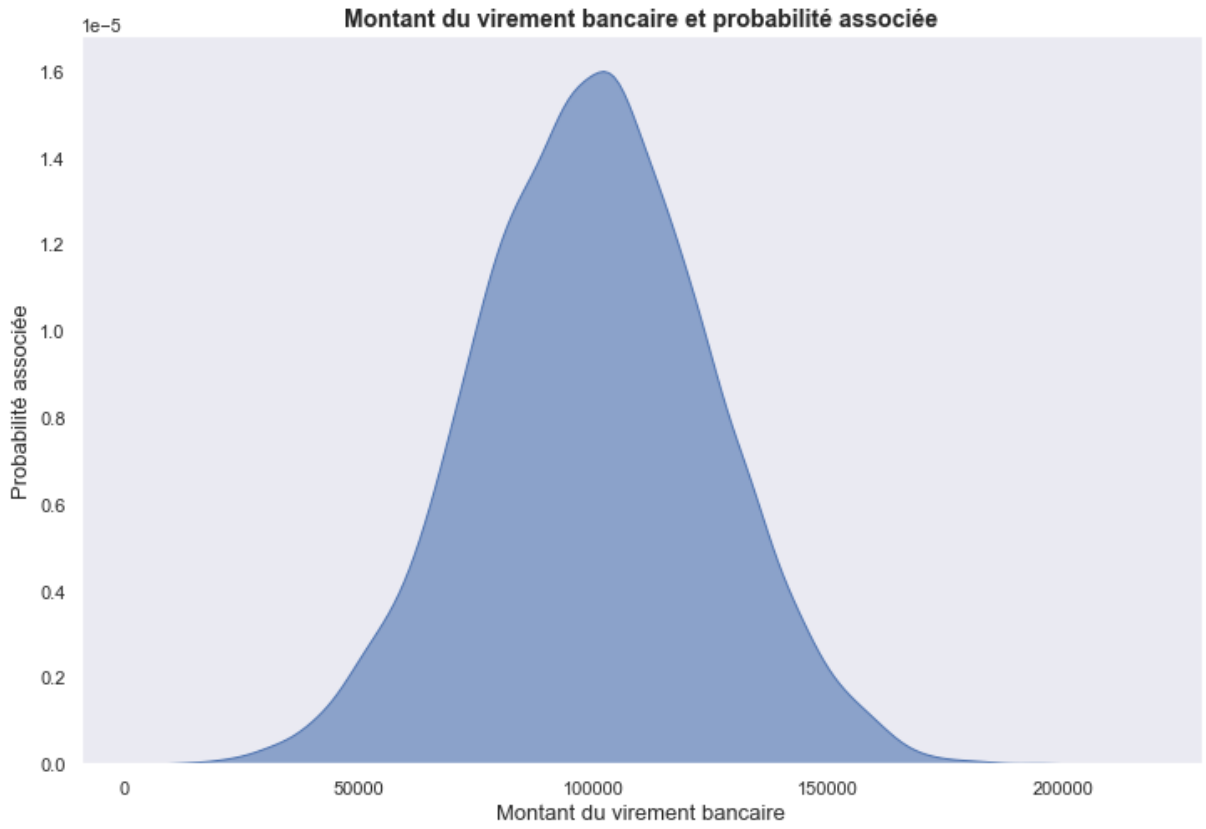
En probabilité, on distingue souvent deux cas : le cas discret et le cas continu. Le cas discret correspond à un problème où les résultats observés sont dénombrables : Si je lance un dé à 6 faces, je n'aurais que 6 résultats possibles. On peut représenter graphiquement les probabilités discrètes par des histogrammes. Par exemple, voici la loi de probabilité régissant un lancer de dé déséquilibré :



Dans cet exemple simple, on voit que le dé n'est pas équilibré. Si je veux obtenir par exemple la probabilité d'obtenir 4 ou moins, je dois faire $P(X = 1) + P(X = 2) + P(X = 3) + P(X = 4)$ soit $\frac{1}{12} + \frac{1}{6} + \frac{1}{6} + \frac{2}{6} = 0.75$. Il y'a donc 75% de chances d'obtenir 4 ou moins. Vous voyez que l'on somme des éléments discrets, un à un. A noter que la somme de toutes les probabilités sont toujours égales à 1.

ii) Les probabilités continues

Le cas continu correspond lui à un cas où les résultats ne sont plus dénombrables. Par exemple, imaginez que vous recensez l'ensemble des virements bancaires effectués dans la journée. Vous modélisez les virements bancaires par la loi normale associée à la densité suivante :



Cet exemple bien que fictif et simpliste illustre ce qu'est une probabilité continue. Comme dit précédemment, on ne dispose plus de points discrets mais d'une fonction de densité. Autrement dit, nous n'avons plus comme résultats possibles : 1, 2, 3, … mais des nombres variants entre 25 000 et 175 000. Le nombre de valeurs possibles ne les rends plus dénombrables.

Dans notre exemple, on voit que la majorité des virements bancaires ont une valeur d'environ 100 000 €. Toutefois, contrairement aux probabilités discrètes, on ne peut plus estimer la probabilité de tirer un virement exactement égal à une certaine valeur (il y'a trop de valeurs, la probabilité de tomber sur un point précis est pratiquement nulle dans le cas continu). En revanche, il est possible d'obtenir une valeur supérieure ou égale à un certain seuil donné. Par exemple, si je veux la probabilité pour que le virement observé soit inférieur ou égal à 100 000 €, je dois faire :

$$P(X \leq 100\,000) = \int_{-\infty}^{100\,000} f(x)\,dx$$

De façon plus globale, si je veux la probabilité de tirer un virement entre 2 valeurs, a et b, je dois faire :

$$P(a < X < b) = \int_a^b f(x)dx$$

Dans notre cas, les virements suivent une loi normale centrée en 100 000 et d'écart type 25 000. Ainsi si je veux la probabilité que le virement observé soit inférieur ou égal à 100 000 €, j'aurais :

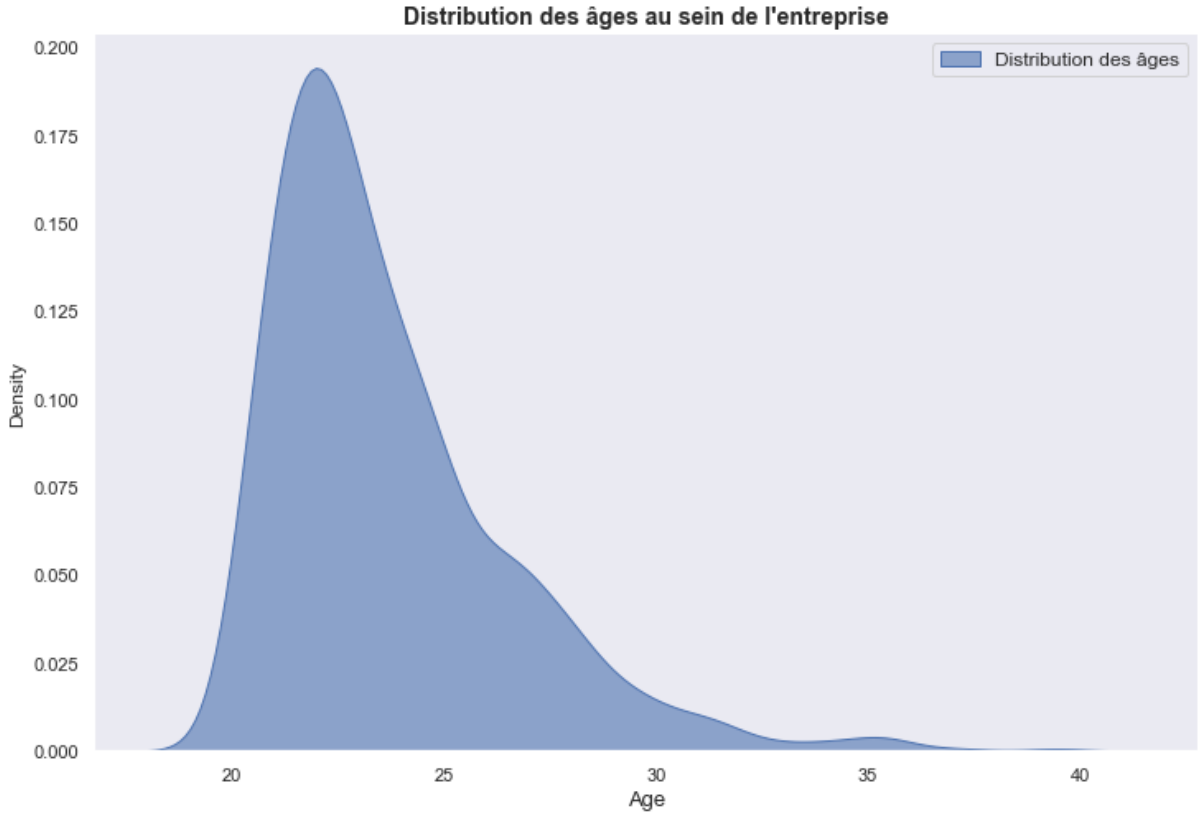
$$P(X \leq 100\,000) = \int_{-\infty}^{100\,000} f(x)dx = \int_{-\infty}^{100\,000} \frac{1}{25\,000\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x - 100\,000}{25\,000}\right)^2 = \frac{1}{2}\right.$$

En réalité, au-delà des notations et des calculs qui peuvent sembler complexes, il faut retenir que **l'intégrale est aux probabilités continues ce que la somme est aux probabilités discrètes**. Elles remplissent le même rôle mais dans des cas différents. Ainsi, de même que dans le cas discret, la somme des probabilités doit être égale à 1, dans le cas continu, l'intégrale de la fonction de densité entre les infinis doit être égale à 1. Enfin, il convient de préciser qu'intégrer une fonction de densité n'est pas toujours possible. C'est d'ailleurs exactement dans ce type de situation que les MCMC peuvent nous être utile.

Si vous souhaitez plus de détails sur la distinction entre ces notions de probabilités discrètes et continues, vous pouvez cliquer ici.

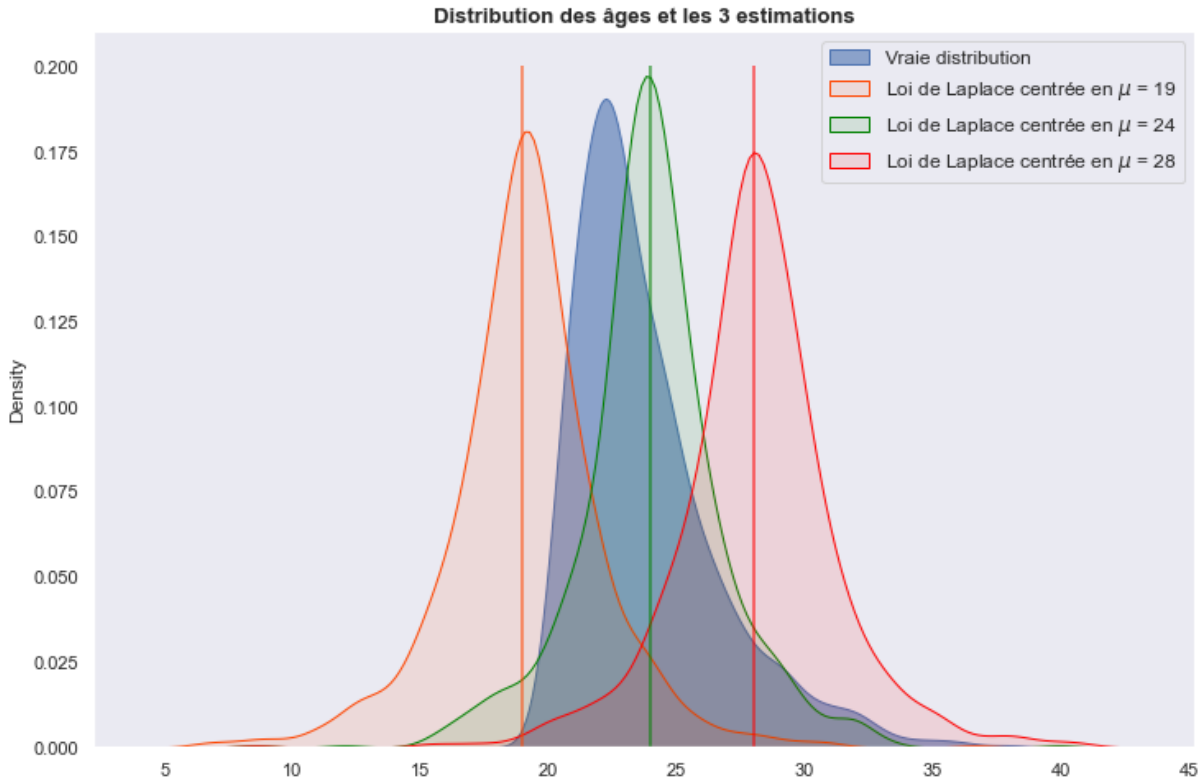
b) La vraisemblance

Imaginez que vous disposez d'un jeu de données, par exemple les différents âges dans votre entreprise. Imaginez maintenant que les âges sont distribués selon la répartition suivante :



La majorité de vos collègues ont donc entre 20 et 25 et une petite part à jusqu'à 40 ans. Au vu de la distribution des âges, vous souhaiteriez représenter la distribution des âges avec une loi de probabilité pour ainsi pouvoir réaliser des prédictions et des estimations plus facilement. Imaginez que vous souhaitiez représenter les âges avec une loi de Laplace (ou double exponentielle).

En faisant appel à vos souvenirs en mathématiques, vous vous souvenez que la loi de Laplace est centrée sur un paramètre, la moyenne (notée μ) qu'il convient de choisir correctement pour que la loi de Laplace colle le mieux possible aux données. Vous décidez de retenir 3 moyennes possibles : 19, 24 et 28. Voyons ce que cela donne :



Il semblerait que la loi qui colle le plus possible à la distribution réelle soit la loi centrée sur 24. Mais, existe-t-il une façon mathématique d'exprimer cela ? En effet, ici le cas est unidimensionnel et simple mais dans des cas plus complexes il se peut que nous ne puissions pas visualiser aussi simplement les choses. C'est alors qu'intervient le concept de **vraisemblance**, fondamental en statistiques inférentielles.

La vraisemblance, c'est la probabilité d'obtenir les données observées pour un modèle particulier choisi. Choisir un modèle qui maximise la vraisemblance, comme on a pu le faire ici, c'est choisir le modèle qui « colle le plus possible » aux données parmi notre collection de modèles. Toutefois, cela ne signifie pas pour autant que le modèle explique « bien » les données, mais simplement qu'il les explique « mieux » que les autres.

Pour rentrer dans les détails mathématiques, si l'on prend une loi P_θ où θ correspond aux paramètres de la loi (dans notre exemple, nous n'avions qu'un paramètre : la moyenne μ, car b était fixée) et un échantillon $(x_1, x_2, \dots, x_n)$. La vraisemblance revient à se demander : « Quelle est, à partir de ma loi théorique, la probabilité d'obtenir la première donnée observée et la deuxième donnée observée, et la troisième … Cela jusqu'à la nième donnée observée ». De façon plus formelle on écrira :

$$L(x_1, \dots, x_n; \theta) = \prod_{i=1}^{n} P_\theta(x_i)$$

Pour des soucis de simplicité calculatoire et de stabilité lors de l'implémentation, on peut passer le tout au logarithme (car le logarithme d'un produit est la somme des logarithmes des éléments qui le compose), ce qui nous donne la log-vraisemblance (ou *log likelihood)* :

$$\log L(x_1, \ldots, x_n;\, \theta) = \sum_{i=1}^{n} \log P_\theta(x_i)$$

Calculons maintenant les log-vraisemblances pour notre exemple :

μ	Log vraisemblance
19	-3783
24	-2483
28	-3711

On remarque qu'on obtient la même chose que par notre approche graphique, à savoir que le modèle « collant » le mieux aux données est le modèle avec une moyenne de 24. Pour aller plus loin, on aurait pu calculer l'estimateur de maximum de vraisemblance, c'est-à-dire le paramètre μ pour lequel on maximise la vraisemblance. Formulé autrement, on aurait pu calculer le paramètre μ de sorte que la loi de Laplace colle le plus possible aux données.

Note : La loi de Laplace aussi nommée double exponentielle ressemble à la loi normale, de par sa fonction de densité assez similaire :

$$\text{Loi normale} : f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right.$$

$$\text{Loi de Laplace} : f(x; \mu, \lambda) = \frac{1}{2\lambda} \exp\left(-\left(\frac{|x-\mu|}{\lambda}\right)\right)$$

La boîte magique

a) Description du problème

Imaginez que vous êtes avec un ami et que celui-ci vous propose un jeu. Il vous propose de tirer un nombre dans une boîte magique. Si le nombre est supérieur à 1, vous gagnez 1000 €, sinon vous lui donnez 100 €. Vous voulez relever le défi mais vous sentez l'entourloupe. Toutefois votre ami en gage de bonne foi, vous donne la fonction de densité décrivant l'apparition des valeurs au sein de sa boite, elle ressemble à cela :

$$f(x) = \frac{\exp(-x^2).2 + \sin(5x) + \cos(10x)}{\int_{-\infty}^{\infty} \exp(-x'^2).2 + \sin(5x') + \cos(10x')}$$

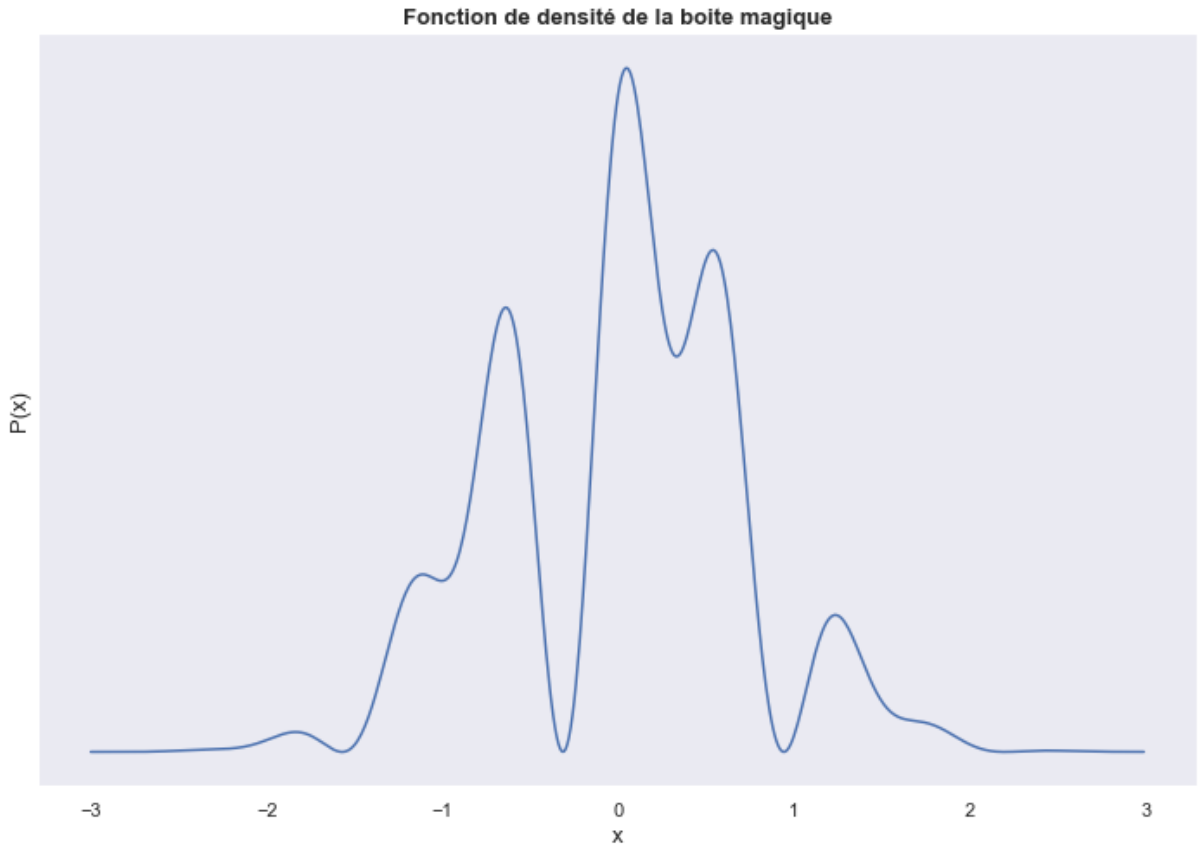
Grâce à vos connaissances en mathématiques, vous savez que pour calculer la probabilité d'obtenir un nombre supérieur à 1 avec une fonction comme celle-ci vous devez faire : $\int_1^{\infty} f(x)dx$.

Or vos mathématiques sont un peu rouillées et vous ne savez pas comment intégrer une telle fonction (c'est même peut être impossible). Dans ce cas, pas de panique, **l'algorithme de Metropolis est là pour vous !**

Avant de rentrer dans les détails sur l'explication de l'algorithme, vous remarquerez l'égalité suivante (le symbole signifie proportionnel) :

$$f(x) \propto \exp(-x^2).2 + \sin(5x) + \cos(10x)$$

Cela signifie que quel que soit le dénominateur, la "forme" de la fonction ne changera pas. En effet si je prends une courbe en cloche, que je multiplie les valeurs qui la composent par 10 ou que je divise ces valeurs par 100, cela ne changera pas la forme de la courbe. Ce sera toujours une cloche. En termes de probabilités, les valeurs changeront mais les probabilités resteront identiques ! C'est intéressant car, il est facile de calculer l'image de cette fonction pour un x donné. Regardons maintenant à quoi ressemble notre fonction de densité :



Hegel disait : « *Ce n'est qu'au crépuscule que la chouette de Minerve prend son envol* ». Bien sûr, il ne parlait pas des MCMC mais l'idée est la même ici. En effet : Si l'on disposait d'un échantillon suffisamment grand (donc représentatif) et dont la répartition suivrait la même forme que la fonction ci-dessus, alors nous n'aurions pas besoin de faire une intégrale. Il nous faudrait simplement regarder la part des éléments supérieurs à 1. Ainsi, pour estimer la probabilité que l'on a de gagner le jeu sans faire de calculs complexes, il nous faut générer un échantillon représentatif de la distribution que suit la boîte magique. En effet, nous pourrons sur celui-ci observer quelle est la part des nombres sortis de la boîte magique qui sont supérieurs à 1. C'est précisément là qu'intervient l'algorithme de Metropolis : Il va nous permettre de générer un échantillon issu de la fonction de densité. Voyons son fonctionnement.

b) L'algorithme de Metropolis-Hastings

i) Présentation et propriétés

Imaginons que nous souhaitions échantillonner à partir d'une distribution de probabilité (que l'on nommera π). Dans notre cas, $\pi(.)$ correspond à la proportionnelle de la fonction de densité $f(.)$ décrivant l'apparition des nombres de la boîte magique. L'algorithme part d'un point donné, et va à partir de ce point générer un nouveau point aléatoirement. Si ce point est plus « vraisemblable » que le précédent, alors on se déplacera vers ce point. Dans le cas contraire, on garde le même point. On répète ce processus un certain nombre de fois. Cela génère une chaîne de Markov (ou chaque point ne dépend que du précédent) pour laquelle la distribution stationnaire sera π sous certaines conditions.

Pour le dire autrement, tous les points que nous aurons acceptés suivrons la même « forme » que la distribution de probabilité à partir de laquelle nous souhaitons échantillonner (π). Les 3 conditions pour que la chaîne converge sont les suivantes :

- La chaine doit être **ergodique**. Cela signifie que tout état de la chaîne est atteignable à partir de tout autre état.
- La chaine doit être **récurrente positive**. Cela signifie que la chaîne doit pouvoir revisiter des points ou elle est déjà passée en un nombre fini d'états.
- La chaine doit être **apériodique**. Cela signifie que la chaîne ne peut pas être piégée dans une boucle.

ii) Le noyau de transition

Pour comprendre l'algorithme, il faut d'abord introduire la notion de noyau de transition, aussi appelé loi instrumentale ou loi de proposition. Cette loi va permettre de générer un nouveau candidat de façon aléatoire et d'en évaluer la vraisemblance. Si le nouveau point est plus « vraisemblable » que l'ancien, on garde le nouveau point. Sinon, on garde l'ancien point. De façon un peu plus formelle, étant donné un point x et un nouveau point y, la loi de proposition sera une distribution de y étant donné x que l'on note $Q(y|x)$.

iii) Le fonctionnement de l'algorithme

1. On initialise le premier point : x_0
2. Pour $i = 1$ à $i = N$:
    - On génère un candidat $y \sim Q(y|x)$, (x étant le point actuel)
    - On génère un point $u \sim U(0,1)$ (un point issu d'une loi uniforme
    - On calcule le ratio de Hastings :

$$H = \frac{\pi(y)Q(x|y)}{\pi(x)Q(y|x)}$$

    - Si $u < H$ alors $x_{i+1} = y$ (on « accepte » le nouveau candidat)
    - Sinon $x_{i+1} = x$ (on « rejette » le nouveau candidat et on reste sur le point actuel)

L'origine de la formule du ratio de Hastings peut sembler un peu mystérieuse, mais puise sa source dans les probabilités Bayésienne et la célèbre formule de Bayes :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Vous pouvez consulter cet article qui décrit l'origine du ratio de Hastings plus en détail et le sens bayésien qu'à ce calcul.

iv) L'algorithme de Metropolis

Si le noyau de transition est symétrique, c'est à dire qu'il respecte la condition $Q(x|y) = Q(y|x)$, alors le calcul du ratio de Hastings se simplifie en $\frac{\pi(y)}{\pi(x)}$. On parle dans ce cas de l'algorithme de Metropolis car il correspond à l'implémentation initiale de l'algorithme proposé par Metropolis et al. en 1953. En 1970, Hastings viendra ajouter sa patte en permettant au noyau de transition d'être asymétrique avec la formule du ratio de Hastings.

v) Notre implémentation de l'algorithme de Metropolis

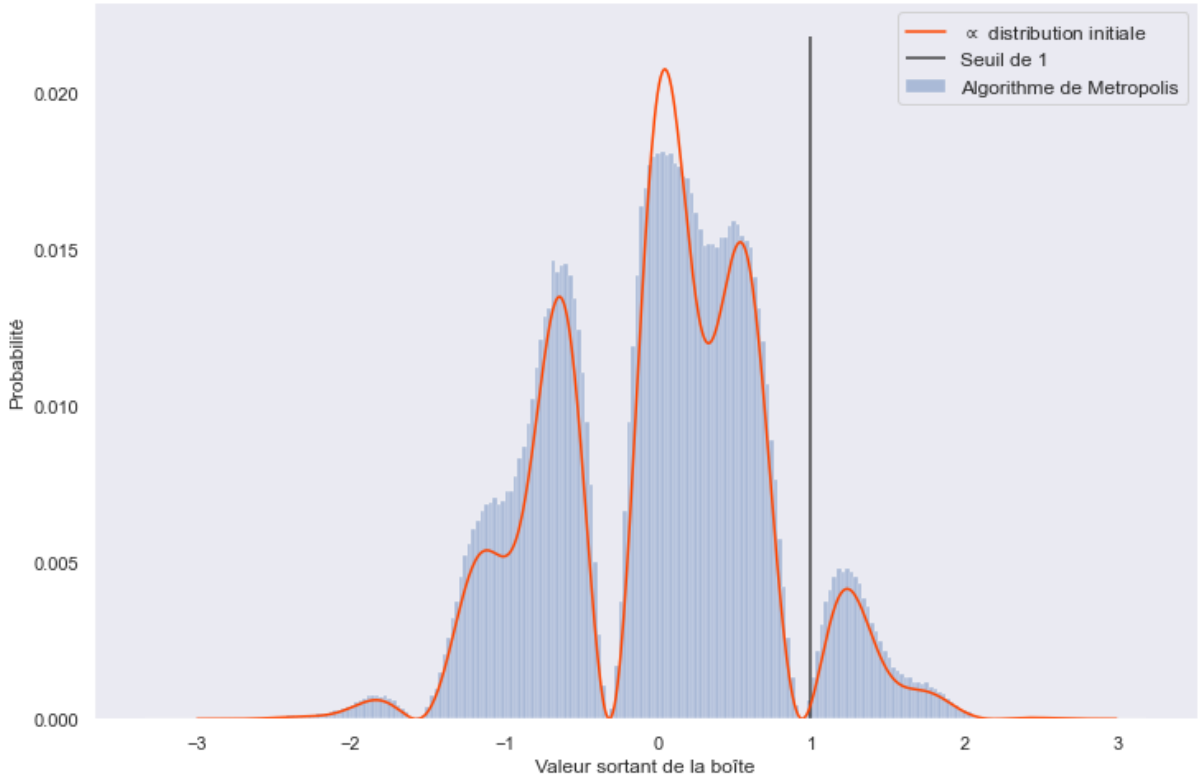
Pour échantillonner selon la fonction de densité, nous générons les candidats de la façon suivante :

$$y = x + \epsilon \text{ où } \epsilon \sim N(0,1)$$

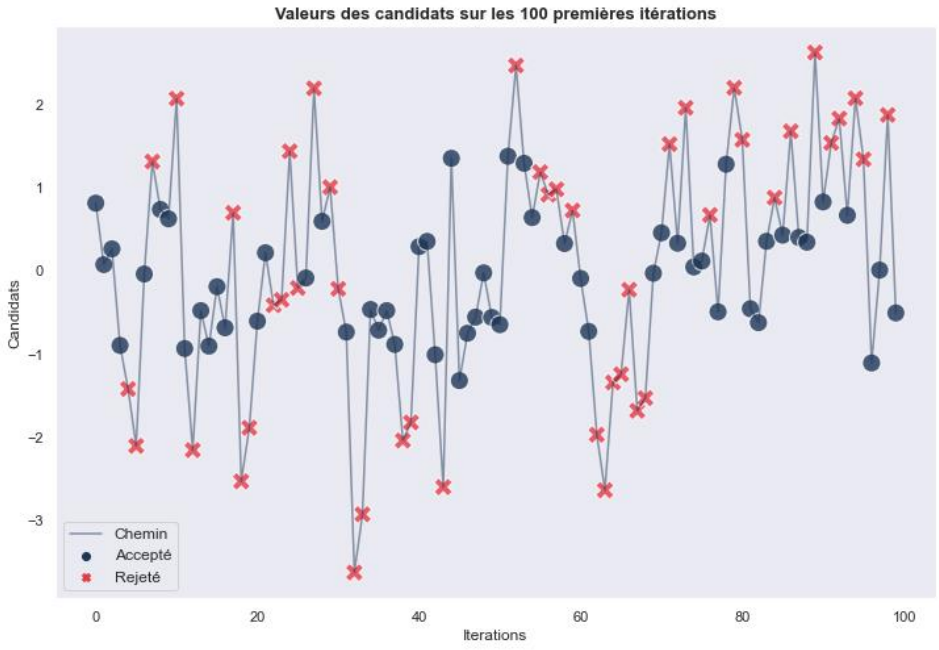
On ajoute un nombre aléatoire issu d'une distribution gaussienne centrée réduite au point actuel pour générer notre candidat. Le noyau de transition étant ainsi symétrique, nous sommes dans le cas de l'algorithme de Metropolis. Il n'y a donc pas à calculer le ratio des vraisemblances $Q(x|y) et\ Q(y|x)$.

vi) Résultats et analyse de la chaîne de Markov

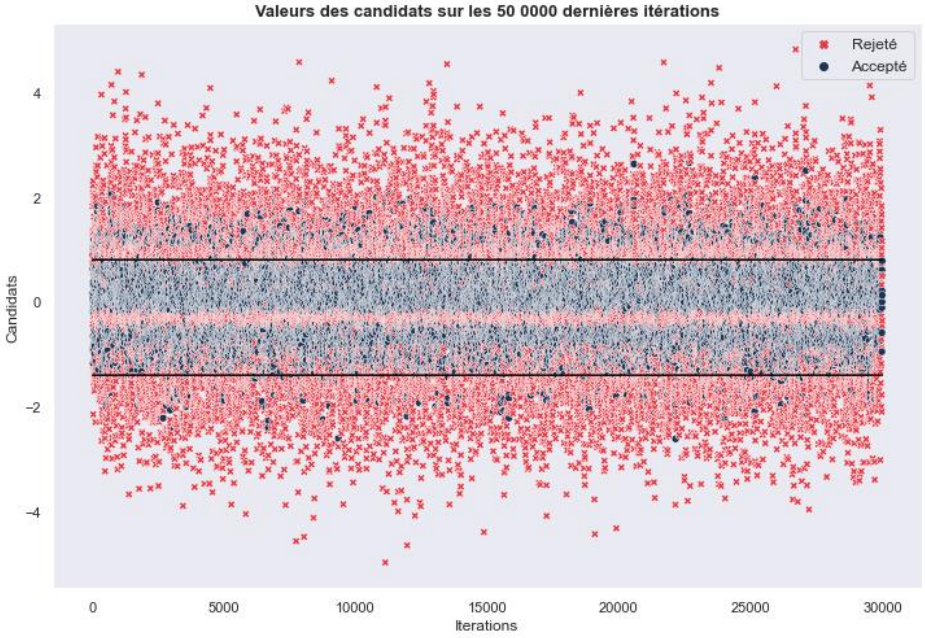
Voici un graphique montrant les résultats obtenus avec notre algorithme (pour 1 millions d'itérations) :



L'algorithme semble avoir bien fonctionné, l'échantillon généré est bien représentatif de notre fonction de densité et cela sans avoir eu à faire d'intégrale ! A partir de notre échantillon, nous pouvons déterminer la probabilité pour que le nombre tiré de la boîte magique soit supérieur à 1. Mais avant cela, observons l'évolution de la chaîne de Markov :



Une certaine convergence semble déjà avoir été atteinte après 100 candidats, car on voir que les candidats acceptés semblent tous centrés entre 1 et -1. Observons maintenant ce que l'on obtient sur les 20 000 dernières itérations :



Clairement, la convergence supposée à été atteinte. On voit que la grande majorité des valeurs stagnent entre les 2 bandes noires. C'est une bonne chose que l'algorithme ait convergé, il est stable et les résultats obtenus seraient donc similaires mêmes si on augmentait le nombre d'itérations.

Le moment est donc venu de percer les mystères de la boîte, quelle sont donc nos chances d'obtenir une valeur supérieure ou égale à 1…

8% !

Et oui, les chances d'obtenir une valeur supérieure ou égale à 1 dans la boîte magique sont de 8%. Pour être un peu plus précis, sur les 492 391 éléments de notre chaîne de Markov, 42 267 sont supérieurs à 1. A noter également que sur nos 1 millions d'itérations, seules 492 391 ont été acceptés soit 507 609 rejets.

Voilà vous disposez maintenant des cartes pour pouvoir prendre votre décision sans vous faire avoir !

A vous de jouer !

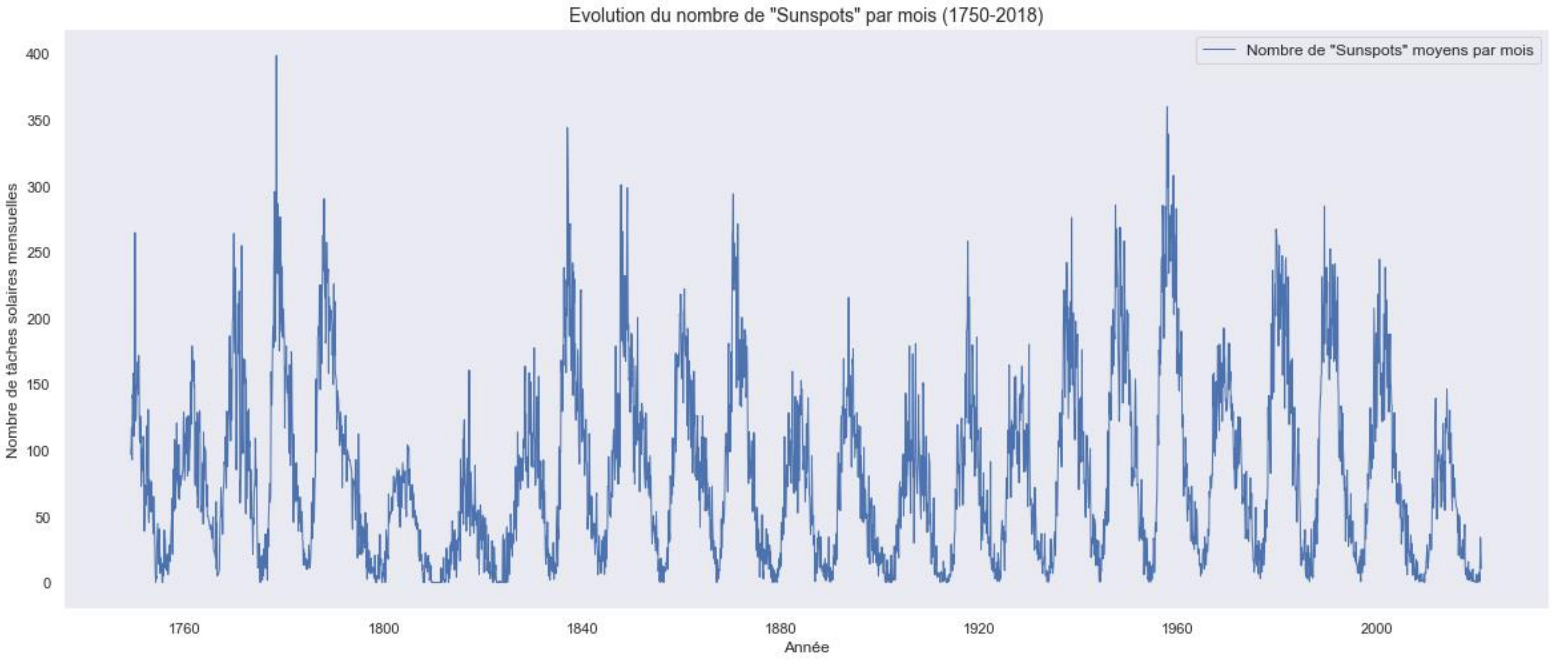
Les *Sunspots*

a) Description du problème et visualisation des données

A présent voyons une autre application des méthodes de Monte Carlo avec un cas plus concret. Il convient de préciser que la partie suivante a été inspirée par le travail de Joseph Moukarzel sur les MCMC (notamment sur le choix de la loi Gamma, le calcul des log vraisemblances, etc). Il y fait d'ailleurs une explication de la philosophie derrière les probabilités bayésiennes et l'inférence qui en découle. Je vous recommande d'aller consulter son travail si vous êtes intéressés par plus de détails.

Ici, on s'intéresse à un phénomène qui touche notre astre solaire : Les tâches solaires ou « sunspots » en anglais (les deux termes seront interchangés dans la suite de l'article). Selon le site de l'observation du soleil et des rayonnements cosmiques, une tâche solaire est « *une région sur la surface du Soleil (photosphère) qui est marquée par une température inférieure à son environnement. Une tache solaire est associée avec un champ magnétique particulièrement intense* ». Etant donné que nous sommes de grands curieux, nous vient alors la question suivante : Ces phénomènes apparaissent-ils de façon aléatoire et si non est-il possible d'en prédire les occurrences ?

Par chance, nous disposons d'un jeu de données *kaggle* relatant l'ensemble du nombre d'occurrences moyennes mensuelles de ces tâches solaires. Cela entre 1749 et 2018. Si vous voulez récupérer les données, le dataset est accessible en cliquant ici. Voyons à présent l'évolution des tâches solaires de 1749 à 2018 :

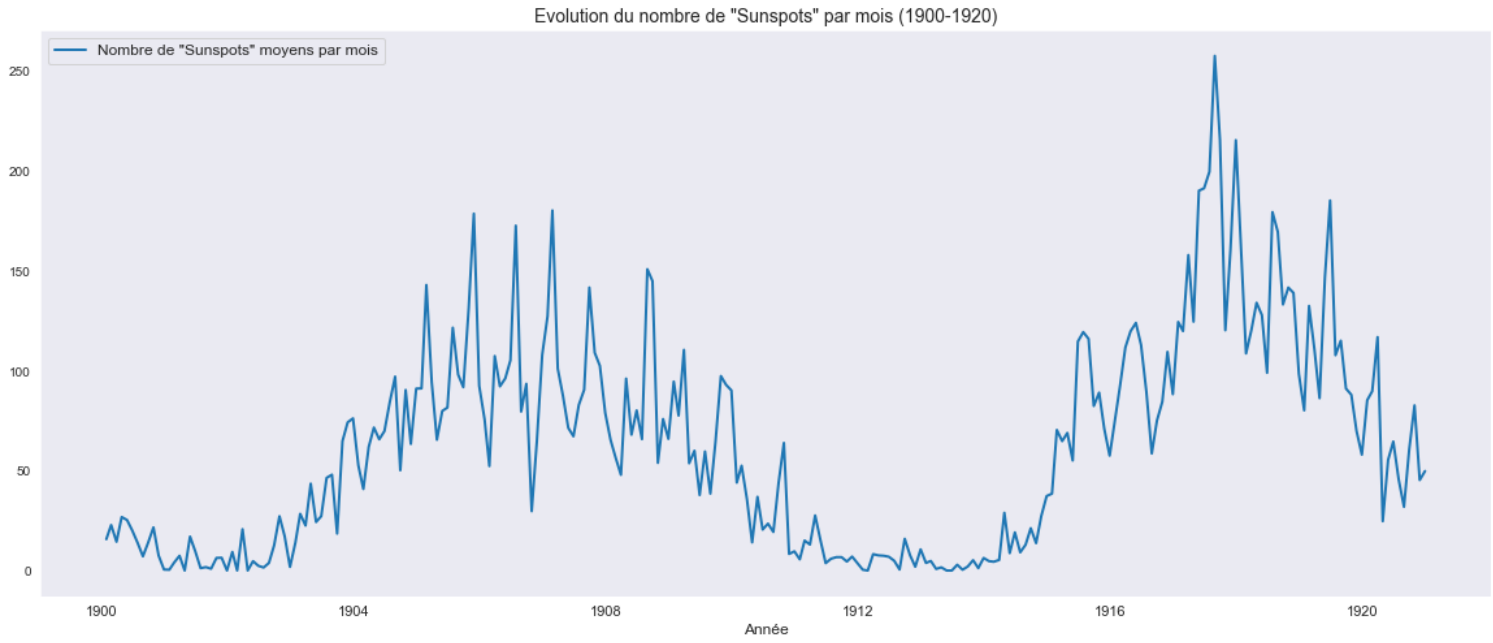


L'évolution est très intéressante ! Déjà, on constate que la fréquence d'apparition des sunspots soit cyclique. De plus, il semblerait que nos données soient stationnaires (le pattern qui régit la variation avec le temps semble être toujours le même). Au sens statistique, si l'on prend les valeurs $E_1, E_2, \dots, E_t$ issues de la série alors $f(E_1, E_2, \dots, E_k)$ et $f(E_{1+k}, E_{2+k}, \dots, E_{t+k})$ suivront la même loi de probabilité (k étant un entier donné).

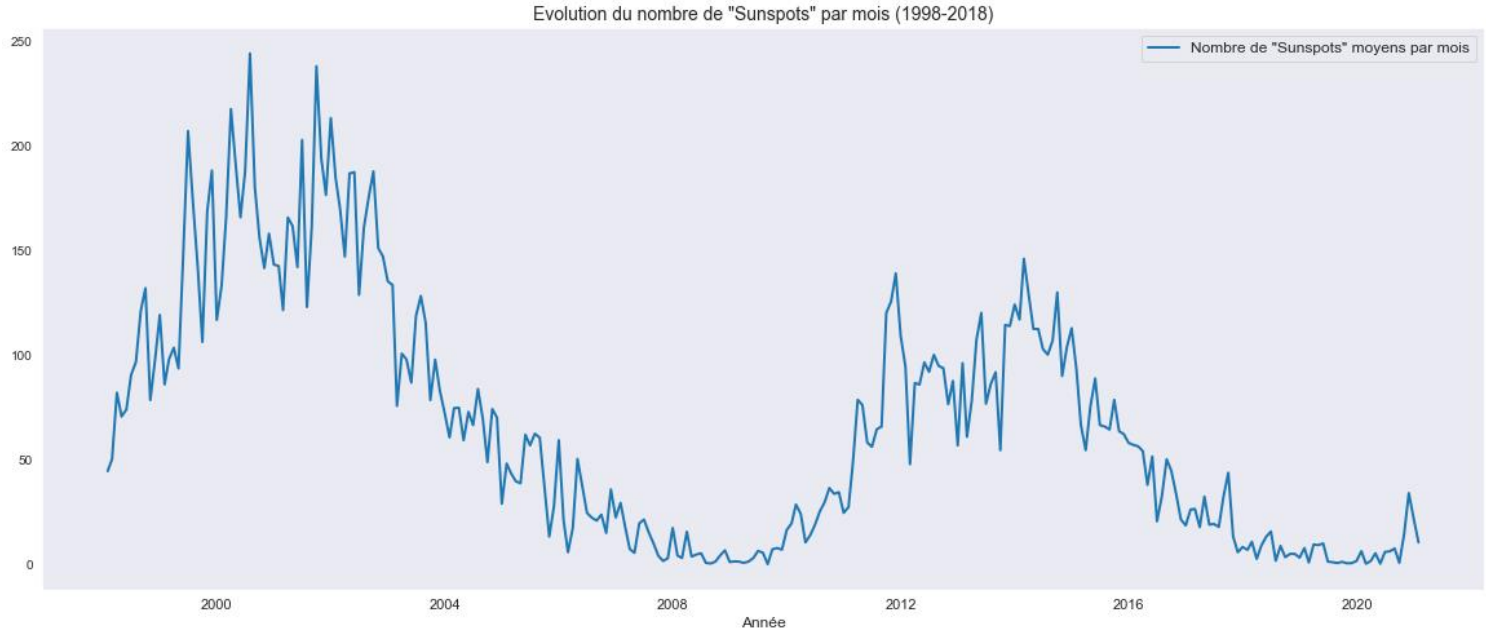
Ce que tout cela signifie de façon plus concrète, c'est que l'on peut modéliser la fréquence d'apparition des tâches solaires à partir d'une certaine loi de probabilité. Reste à déterminer quel est l'intervalle de temps entre chaque cycle et quelle est la loi de probabilité régissant nos données.

Pour déterminer cela, réalisons deux "zooms" sur des intervalles de temps plus courts :

Entre 1900 et 1920 :

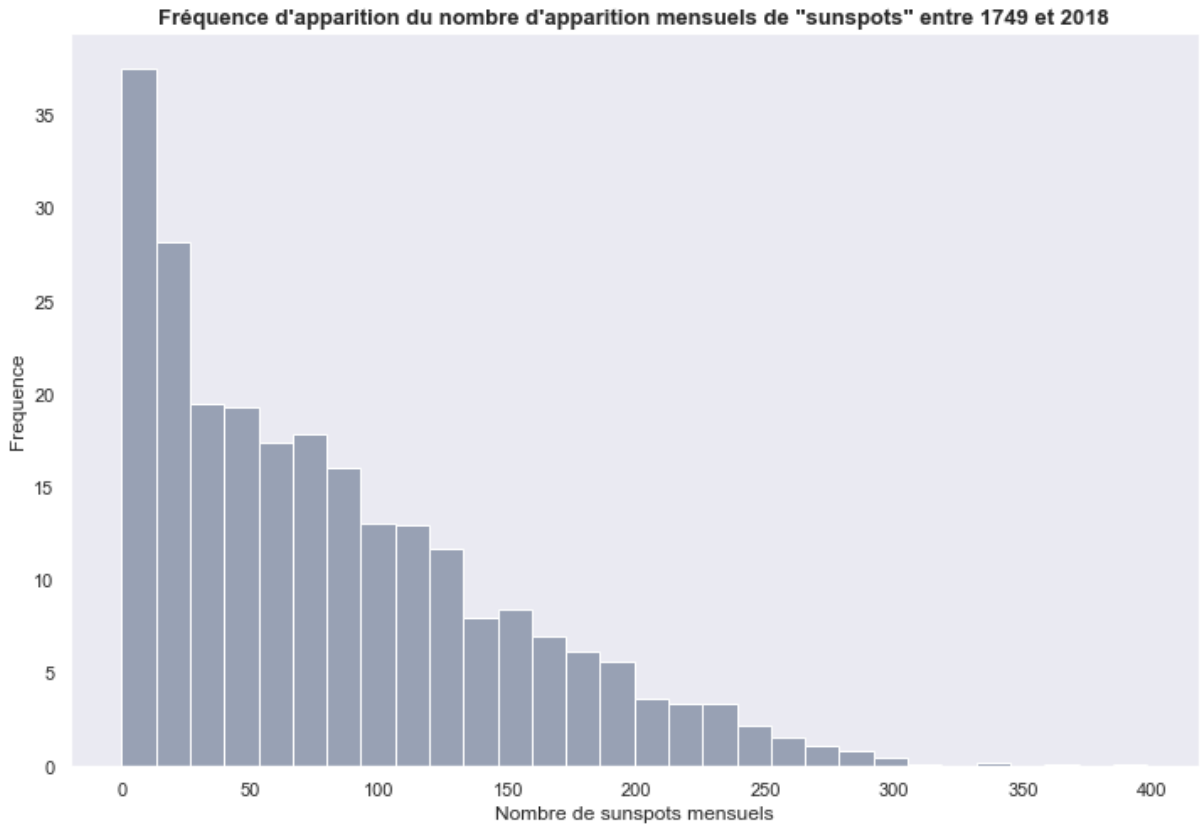


Entre 1998 et 2018 :



Il apparait que le cycle semble subvenir environ tous les 10 ou 12 ans.

Pour ce qui est de déterminer la loi de probabilité nous permettant d'effectuer notre modélisation, nous pouvons afficher la fréquence d'apparition du nombre de tâches solaires sur la période :



Il semblerait au vu de cette répartition décroissante progressivement que l'on puisse retenir une loi Gamma pour notre modélisation.

b) Modélisation

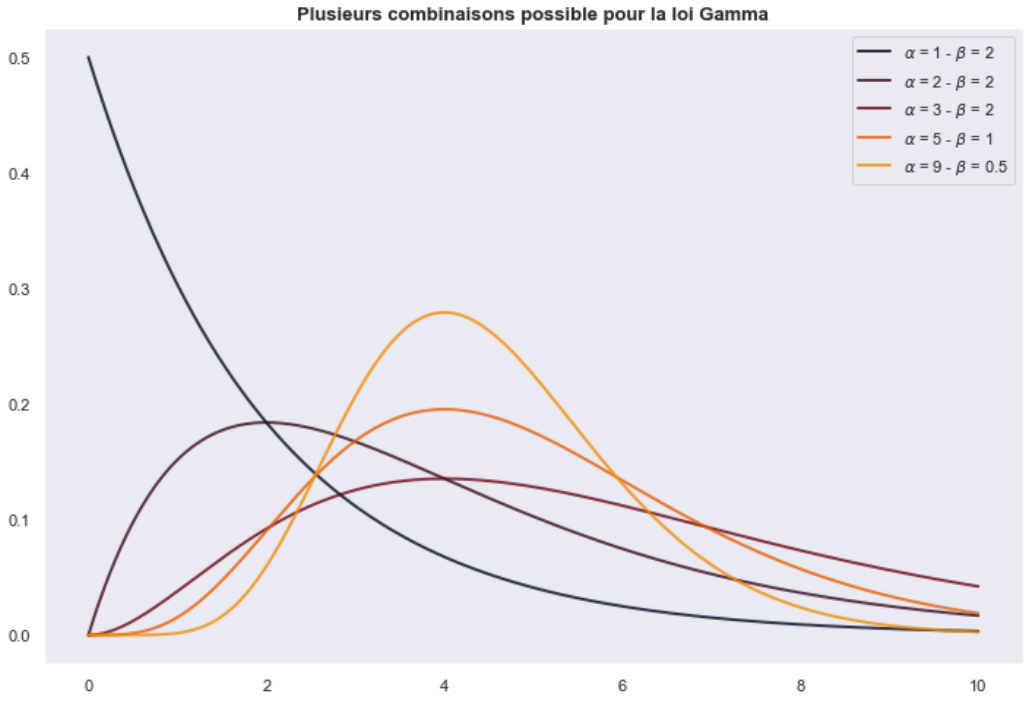
Selon la documentation de scipy (la librairie de stats de Python), la distribution de la loi gamma s'exprime de la façon suivante :

$$f(x;\alpha,\beta) = \frac{\beta^{\alpha} x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$$

$\Gamma(\alpha)$ faisant référence à la fonction Gamma d'Euler.

La distribution est donc régie par deux paramètres α et β.

Observons maintenant comment se comporte la densité d'une loi gamma lorsque l'on fait varier ces paramètres :



Comme vous le voyez, il existe plusieurs paramétrages possibles pour la loi Gamma, lui donnant une « forme » différente. Notre problématique est alors de trouver la combinaison de paramètre optimale (*i.e* qui maximisera la vraisemblance) pour notre problème : On parle parfois du principe de **Boucles d'Or** (Goldilocks Principle) pour ce genre de problèmes.

C'est une référence au conte éponyme ou Boucles d'Or en arrivant chez les Ours, cherche la soupe ni trop chaude ni trop froide ou le lit ni trop grand ni trop petit.

c) Algorithme de Metropolis-Hastings

i) Description de l'algorithme

L'algorithme utilisé est le même que le précédent à deux distinctions près.

D'abord, ici on ne cherchera pas à générer des points directement mais on cherchera à estimer les paramètres optimaux de la loi gamma pour qu'elle colle le mieux possible à nos données : α et β. L'idée est la même que précédemment mais on se déplace à présent aléatoirement dans l'espace des paramètres.

De plus, ici, nous allons avoir recours à l'algorithme de Metropolis Hastings et non plus simplement l'algorithme de Metropolis. On vient générer deux paramètres issus de deux lois normales :

$$Q(y|x_i) = \alpha_{i+1} \sim N(\alpha_i, 0.05) \; et \; \beta_{i+1} \sim N(\beta_i, 5)$$

1. On initialise le premier vecteur de paramètres : $x_0 = [\alpha_0, \beta_0]$
2. Pour $i = 1$ à $i = N$:
    - On génère un candidat $y = [\alpha_y, \beta_y] \sim Q(y|x_i)$, avec x_i le vecteur des paramètres actuels
    - On génère un nombre $u \sim U[0,1]$ où U fait référence à la loi uniforme
    - On calcule le ratio de Hastings :

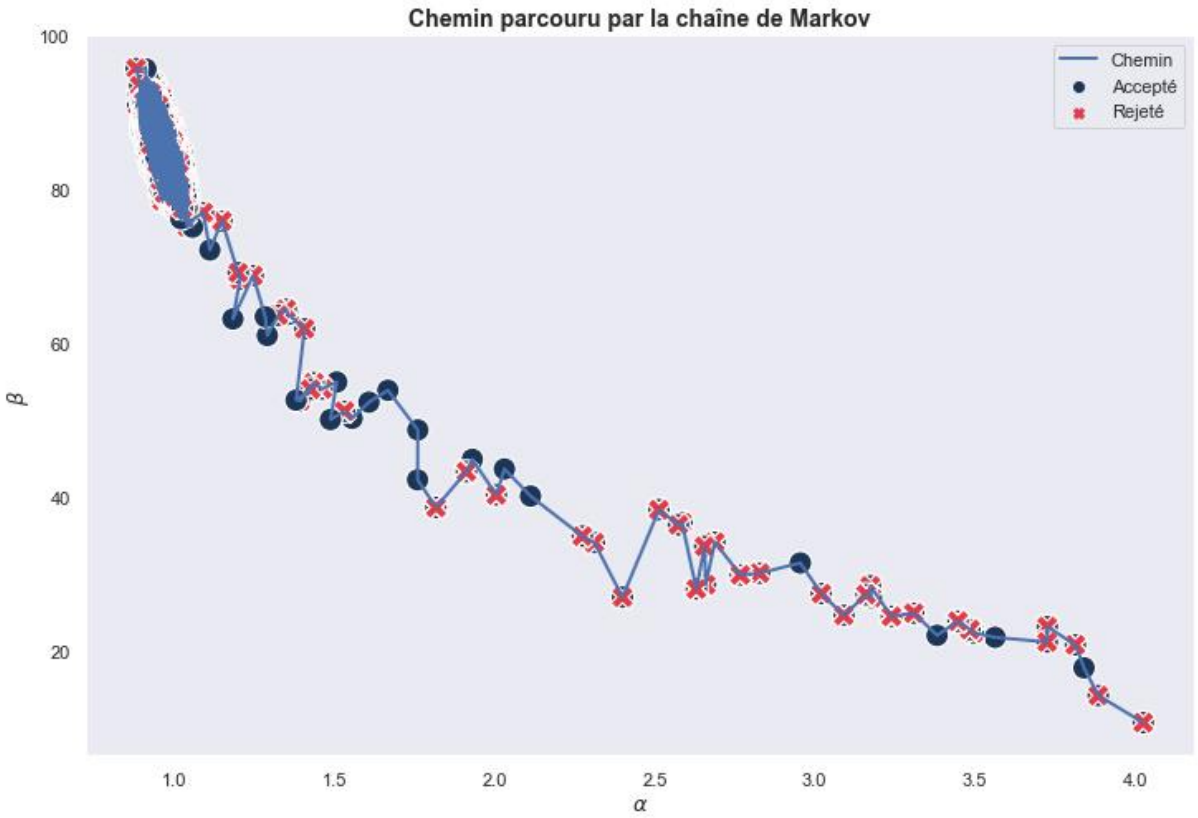
$$H = \frac{\pi(y)Q(x|y)}{\pi(x)Q(y|x)}$$

    - Si $u < H$ alors $x_{i+1} = y$ (on se déplace vers le nouveau vecteur de paramètres)
    - Sinon $x_{i+1} = x$ (on reste sur le vecteur de paramètres actuel)

Implémentons ceci et observons les résultats obtenus !

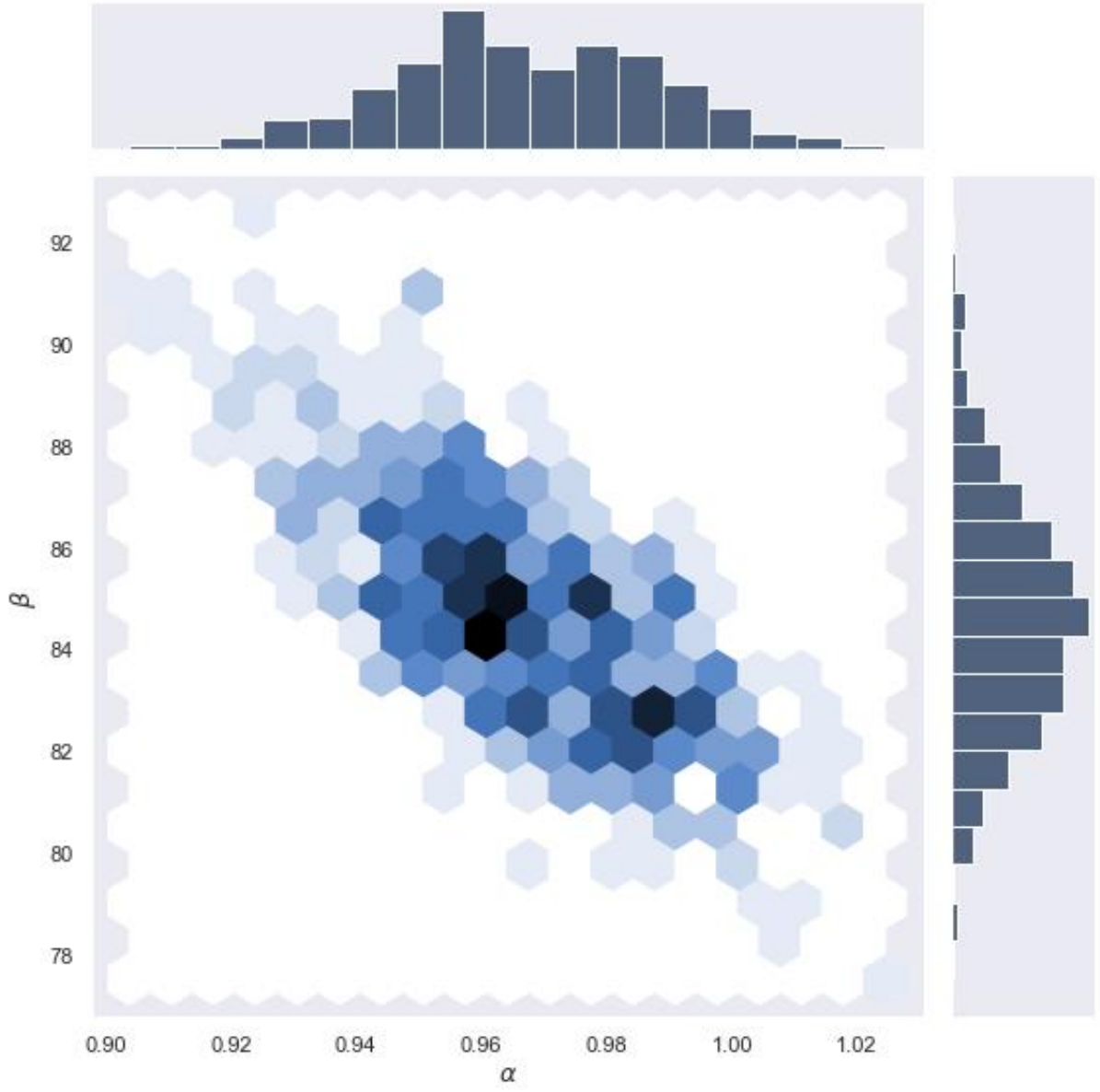
ii) Analyse de la chaîne de Markov

Voici le chemin parcouru par la chaine de Markov au cours des 100 000 itérations (le point de départ était $\alpha = 4$ et $\beta = 10$) :



Le graphique est un peu contre intuitif car dans l'ordre des itérations, on part d'en bas à droite pour arriver en haut à gauche. On voit toutefois qu'on semble avoir atteint un point de convergence pour nos 2 paramètres. En effet, autour d'$\alpha = 1$ et de $\beta = 80$, les points semblent stagner.

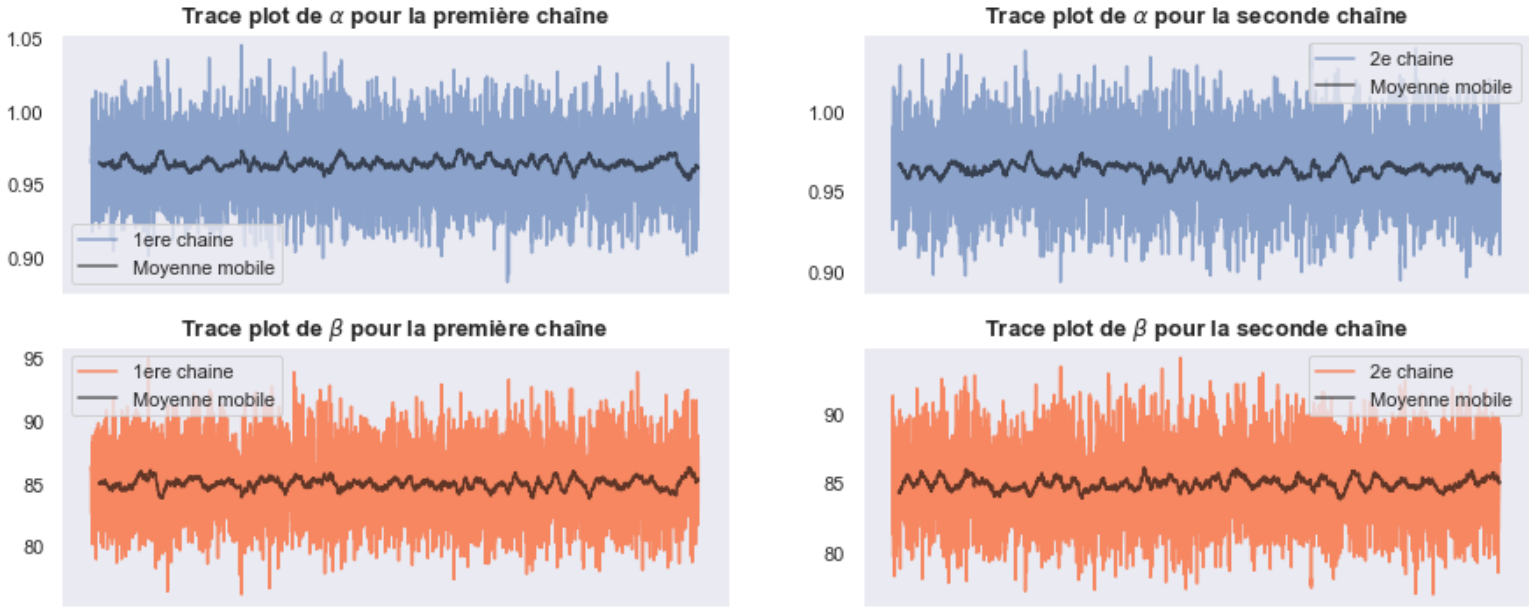
Voyons plus en détail la répartition conjointe de α et β sur les 500 dernières itérations acceptées :



On voit que nos 2 paramètres ont convergés et oscillent sur des faibles échelles de valeur. La solution retenue est donc de prendre la valeur moyenne des 500 dernières itérations acceptées pour chacun des paramètres. Nous reviendrons plus en détail là-dessus ci-après.

iii) Point sur la convergence de la chaîne de Markov

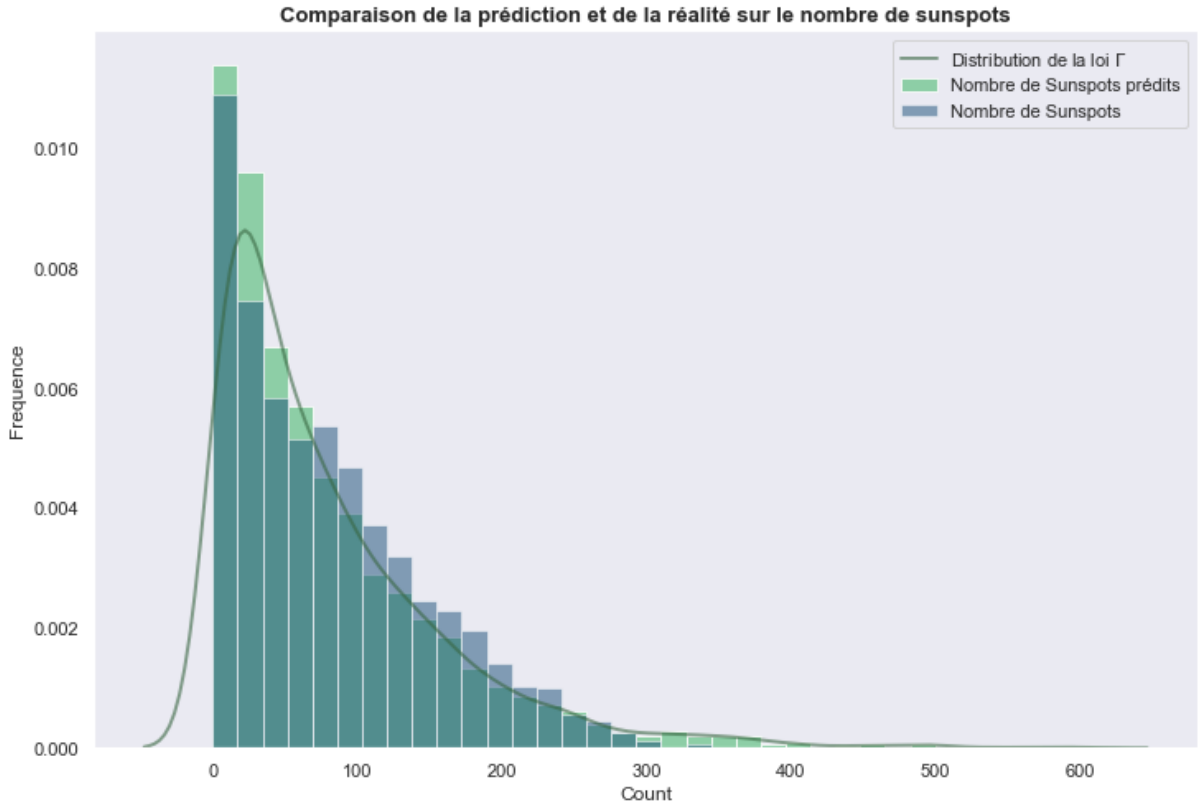
Avant de passer à la phase de prédiction, il peut être intéressant d'analyser la convergence de la chaîne de Markov. Pour ce faire, une des possibilités est de générer deux chaînes distinctes via l'algorithme et de comparer les résultats obtenus pour nos différents paramètres. L'objectif est de voir si les résultats obtenus sont les mêmes, ce qui signifie une certaine « robustesse » de la part de notre algorithme. Si les résultats diffèrent complétements, cela voudrait dire que l'algorithme n'est pas stable et les résultats obtenus ne seraient pas réellement exploitables. Voyons ce que cela donne :



On voit que la chaîne est stable, c'est à dire que la valeur des paramètres est sensiblement la même sur les 2 itérations. C'est bon signe. Nous pouvons à présent passer à l'inférence.

iv) Prédictions et résultats obtenus

Pour les prédictions, on calcule des paramètres "finaux" en calculant la moyenne de chacun des paramètres sur les 500 dernières itérations acceptées. Cela nous donne : $\alpha = 0{,}967$ et $\beta = 83{,}84$. Implémentons tout ça et voyons le résultat :



Il semblerait que notre algorithme ait bien fonctionné. En effet la distribution que nous obtenons à partir de notre loi Gamma épouse bien la distribution réelle du nombre de tâches solaires.

Pour les amateurs de statistiques, remarquons que la loi $\Gamma(1, \beta)$ n'est autre que la loi exponentielle $E(\beta)$. Or le α que nous avons obtenu est proche de 1. C'est pourquoi la densité de la loi gamma finale est similaire à la densité d'une loi exponentielle.

Conclusion

Cela conclu cet article. Nous avons donc vu ce qu'était l'algorithme de Metropolis-Hastings et deux de ses applications possibles : Un moyen d'échantillonner à partir d'une fonction de densité complexe ainsi qu'un moyen pour estimer les paramètres d'une loi de probabilité étant donné un échantillon donné.

Pour ceux qui veulent aller plus loin sur le sujet, il existe nombre d'autres algorithmes de Monte Carlo comme l'échantillonnage de Gibbs par exemple. Bien que l'on retrouve beaucoup d'applications de cet algorithme dans la physique statistique, on en retrouve également en intelligence artificielle par exemple avec notamment les machines de Boltzmann. De plus, comme on a pu le voir, c'est une méthode assez élégante pour échantillonner à partir d'une fonction de densité (et pouvoir calculer des probabilités à partir de cette fonction). Cela tout en évitant la méthode assez "brutale" des intégrales, souvent complexes et nécessitants des pirouettes calculatoires ne s'avérant pas toujours fructueuses.

Références

- The Metropolis Hastings Algorithm, *Matthew Stephens*, 2018
- Convergence Diagnostics, *Rob Hicks*
- A Practical Guide to MCMC Part 1: MCMC Basics, *Justin Ellis*
- MCMC, *Joseph Moukarzel*
- Cours 2 : Metropolis-Hastings, *Olivier Cappé*
- Algorithme de Metropolis, *Frédéric Legrand*