

Simulation et Monte Carlo

Modélisation d'une file d'attente

Balthazar COURVOISIER, Baptiste PASQUIER & Luca TEODORESCU

ENSAE Paris

Mai 2021

- 1 Modélisation de la file d'attente
- 2 Probabilité de dépassement du nombre de clients
- 3 Quasi Monte Carlo
- 4 ABC
- 5 Loi de Weibull

Paramètres de la file d'attente

- On note $(A_n)_{n \geq 0}$ la suite des dates d'arrivées des clients.
 $\forall n \geq 0, A_n \leq A_{n+1}$.
- $\forall n \geq 0, \tau_n = A_{n+1} - A_n$, on suppose

$$(\tau_n)_{n \geq 0} \stackrel{iid}{\sim} \text{Exp}(\lambda_1)$$

- On note $(d_n)_{n \geq 0}$ les durées de service de chaque client, on suppose

$$(d_n)_{n \geq 0} \stackrel{iid}{\sim} \text{Exp}(\lambda_2)$$

- On note enfin $(S_n)_{n \geq 0}$ les dates de début de service de chaque client (c'est à dire sa date de prise en charge).

Relation de récurrence

La suite $(S_n)_{n \geq 0}$ vérifie la relation de récurrence suivante :

- $S_0 = A_0$: le premier client est servi dès son arrivée ;
- $\forall n \geq 1, S_n = \max(S_{n-1} + d_{n-1}, A_n)$: un client est servi dès son arrivée s'il arrive après la fin de service du client précédent, sinon il doit attendre.

Description de la loi

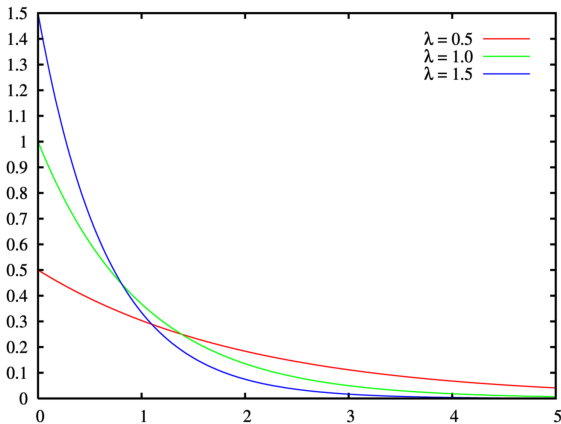


FIGURE – Densité de lois exponentielles

Générateur de la loi exponentielle

- Soit $X \sim \text{Exp}(\lambda)$, $\lambda > 0$
On a $F_X(t) = 1 - e^{-\lambda t}$, donc

$$u = 1 - e^{-\lambda t} \Leftrightarrow t = -\frac{\ln(1-u)}{\lambda} = F_X^{-1}(u)$$

- Méthode d'inversion : soit $U \sim \mathcal{U}[0, 1]$, alors $F_X^{-1}(U) \sim \text{Exp}(\lambda)$.

Exemple

Client	Date arrivée	Date début de service	Temps de service	Date fin de service
1	9.23		4.28	
2	13.48		24.65	
3	16.00		1.73	
4	16.93		0.68	
5	17.48		0.18	
...				

TABLE – Simulation d'une file d'attente avec $\lambda_1 = 0.2$ et $\lambda_2 = 0.1$

Exemple

Client	Date arrivée	Date début de service	Temps de service	Date fin de service
1	9.23	9.23	4.28	
2	13.48		24.65	
3	16.00		1.73	
4	16.93		0.68	
5	17.48		0.18	
...				

TABLE – Simulation d'une file d'attente avec $\lambda_1 = 0.2$ et $\lambda_2 = 0.1$

Exemple

Client	Date arrivée	Date début de service	Temps de service	Date fin de service
1	9.23	9.23	4.28	13.51
2	13.48		24.65	
3	16.00		1.73	
4	16.93		0.68	
5	17.48		0.18	
...				

TABLE – Simulation d'une file d'attente avec $\lambda_1 = 0.2$ et $\lambda_2 = 0.1$

Exemple

Client	Date arrivée	Date début de service	Temps de service	Date fin de service
1	9.23	9.23	4.28	13.51
2	13.48	13.51	24.65	
3	16.00		1.73	
4	16.93		0.68	
5	17.48		0.18	
...				

TABLE – Simulation d'une file d'attente avec $\lambda_1 = 0.2$ et $\lambda_2 = 0.1$

Exemple

Client	Date arrivée	Date début de service	Temps de service	Date fin de service
1	9.23	9.23	4.28	13.51
2	13.48	13.51	24.65	38.16
3	16.00		1.73	
4	16.93		0.68	
5	17.48		0.18	
...				

TABLE – Simulation d'une file d'attente avec $\lambda_1 = 0.2$ et $\lambda_2 = 0.1$

Exemple

Client	Date arrivée	Date début de service	Temps de service	Date fin de service
1	9.23	9.23	4.28	13.51
2	13.48	13.51	24.65	38.16
3	16.00	38.16	1.73	39.89
4	16.93	39.89	0.68	40.57
5	17.48	40.58	0.18	40.75
...				

TABLE – Simulation d'une file d'attente avec $\lambda_1 = 0.2$ et $\lambda_2 = 0.1$

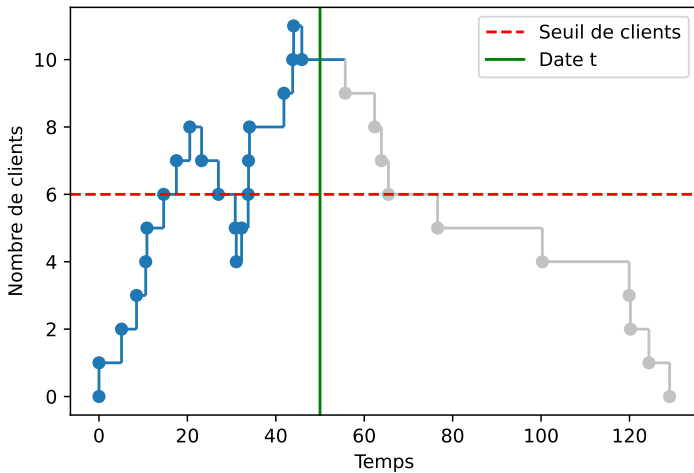


FIGURE – Simulation d'une file d'attente avec $\lambda_1 = 0.2$, $\lambda_2 = 0.1$, $n = 6$ et $t = 50$

Estimation de la probabilité

- On souhaite calculer la probabilité que le nombre de clients dans la salle d'attente dépasse un certain seuil critique n (entre le temps 0 et le temps t).
- Notons X le nombre maximum de clients dans la salle d'attente entre le temps 0 et le temps t .
- On cherche à estimer $P(X > n) = \mathbb{E}[\mathbb{1}(X > n)]$. On remplace l'espérance par sa contrepartie empirique avec N simulations :

$$\mathbb{E}[\mathbb{1}(X > n)] \approx \sum_{i=1}^N \mathbb{1}(X_i > n)$$

Influence de λ_1 et λ_2

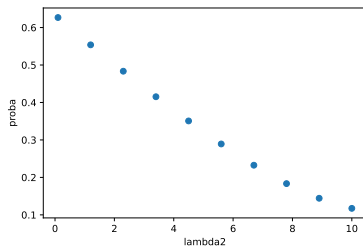
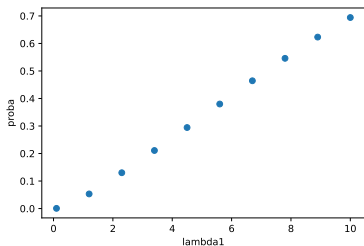


FIGURE – Représentation de la probabilité de dépassement du nombre de clients dans la file d'attente en fonction de λ_1 et λ_2

Influence de n et t

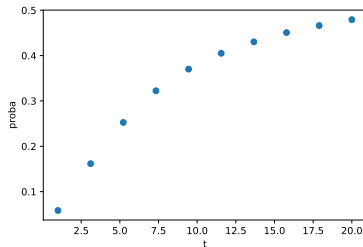
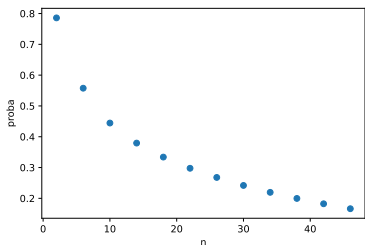


FIGURE – Représentation de la probabilité de dépassement du nombre de clients dans la file d'attente en fonction de n et t

Stratification

Dans le cas d'une seule génération de va $Exp(\lambda_1)$ et $Exp(\lambda_2)$ pour chaque simulation :

- Monte Carlo :

$$\hat{p} = \frac{1}{N} \sum_{i=1}^N \varphi(U_{1,i}, U_{2,i}), \quad U_{1,i}, U_{2,i} \sim \mathcal{U}[0, 1]$$

- Quasi-Monte Carlo avec stratification : découpage de l'espace $[0, 1]^2$ en N^2 zones et tirage de N zones A_1, \dots, A_N :

$$\hat{p} = \frac{1}{N} \sum_{i=1}^N \varphi(U_{1,i}, U_{2,i}), \quad (U_{1,i}, U_{2,i}) \sim \mathcal{U}[A_i]$$

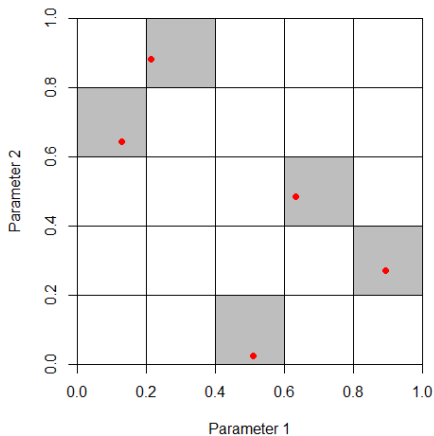


FIGURE – Latin hypercube sampling en dimension 2 avec $N=5$

Difficulté

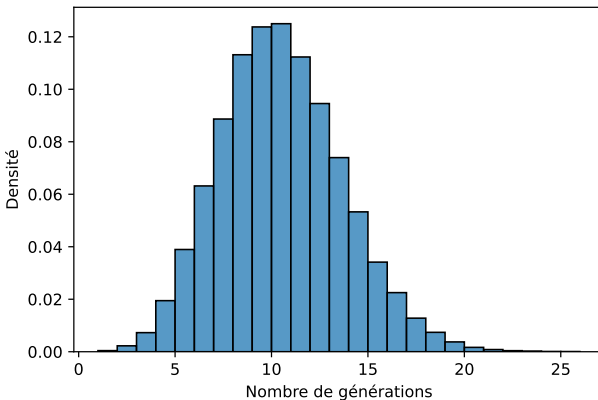


FIGURE – Nombre de générations de $Exp(\lambda_1)$ nécessaires pour atteindre le seuil de temps $t = 50$ avec $\lambda_1 = 0.2$

Méthode 1 - Stratification locale sur $Exp(\lambda_2)$

Pour chaque simulation :

- Génération d'un nombre suffisant de $Exp(\lambda_1)$ noté $n_{clients}$. On sait alors qu'il faut générer $n_{clients}$ va $Exp(\lambda_2)$.
- Stratification en dimension 1 avec $n_{clients}$ points pour définir $U_1, \dots, U_{n_{clients}}$.
- Calcul de $F_X^{-1}(U_1), \dots, F_X^{-1}(U_{n_{clients}})$.

Méthode 2 - Stratification globale sur $Exp(\lambda_2)$

On note N le nombre de simulations effectuées.

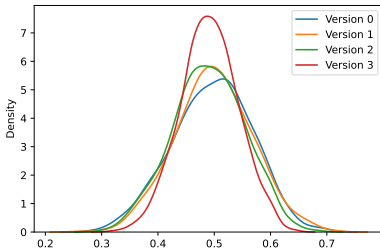
- Pour chacune des N simulations, on génère un nombre suffisant de $Exp(\lambda_1)$. On note $n_{max_clients}$ le nombre maximum de générations sur les N simulations.
- On sait alors qu'il faut générer au maximum $n_{max_clients}$ va $Exp(\lambda_2)$ dans chaque simulation.
- Stratification en dimension $n_{max_clients}$ avec N points.
- Chaque simulation utilise ensuite un de ces N points pour définir $U_1, \dots, U_{n_{max_clients}}$.

Méthode 3 - Stratification globale sur $Exp(\lambda_1)$ et $Exp(\lambda_2)$

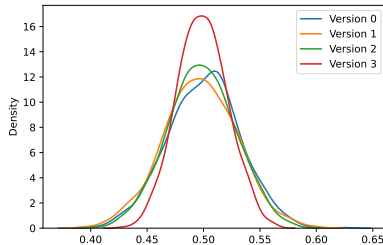
- On calcule un majorant M du nombre de générations de $Exp(\lambda_1)$ nécessaires pour atteindre le temps t dans une majorité des cas.
[▶ Démonstration](#)
- On sait alors qu'il faut générer au maximum $n_{max_clients}$ va $Exp(\lambda_1)$ et $Exp(\lambda_2)$ dans chaque simulation.
- Stratification en dimension $2M$ avec N points.
- Chaque simulation utilise ensuite un de ces N points pour définir $U_{1,1}, \dots, U_{1,M}$ et $U_{2,1}, \dots, U_{2,M}$.

Comparaison des méthodes

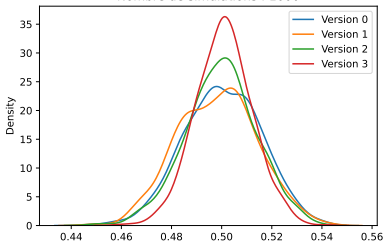
Nombre de simulations : 50



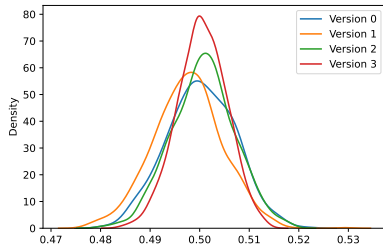
Nombre de simulations : 250



Nombre de simulations : 1000



Nombre de simulations : 5000



Comparaison des méthodes

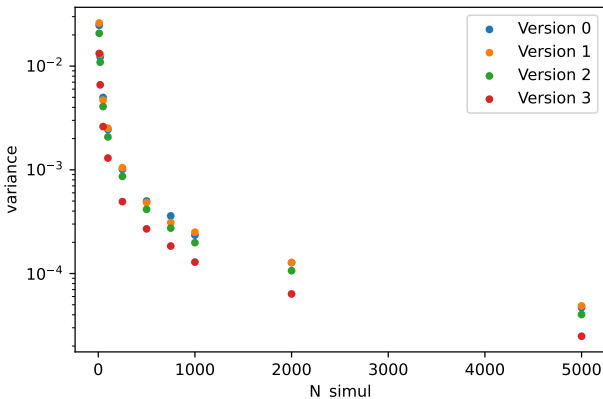


FIGURE – Variance en fonction du nombre de simulations pour chacun des modèles

Comparaison des méthodes

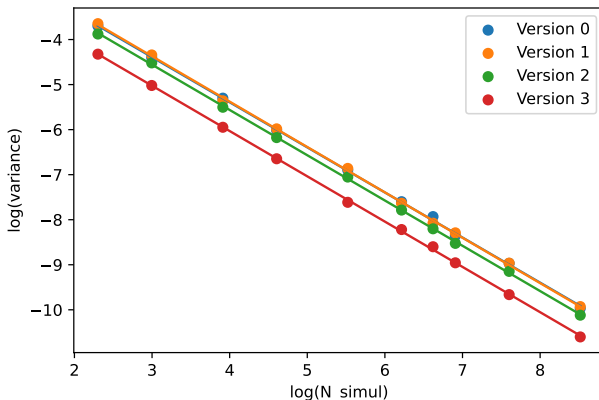


FIGURE – Régression en log-log

Approximate Bayesian Computation : Principe

L'Approximate Bayesian Computation et surtout son algorithme de *rejection*, répondent au problème suivant :

- Nous avons une observation de données D dans le cadre d'une simulation ;
- On se demande de quelle paramétrisation du problème ce jeu de données fait vraisemblablement partie.

Approximate Bayesian Computation : Illustration

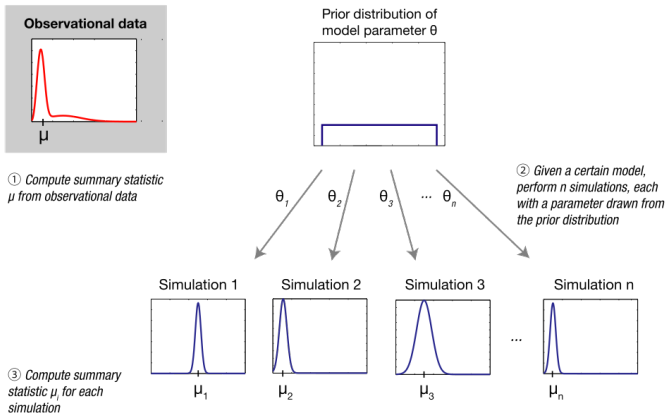


FIGURE – Le fonctionnement conceptuel de l'ABC par *Sunnåker M, Busetto AG, Numminen E, Corander J, Foll M, et al. dans (2013) Approximate Bayesian Computation*

Approximate Bayesian Computation : Illustration

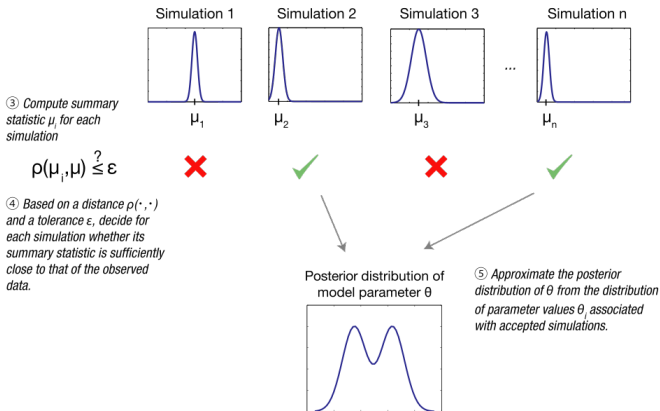


FIGURE – Le fonctionnement conceptuel de l'ABC par *Sunnåker M, Busetto AG, Numminen E, Corander J, Foll M, et al. dans (2013) Approximate Bayesian Computation*

Approximate Bayesian Computation : Notre situation

Dans le cas de la file d'attente on peut raisonnablement s'intéresser au lien entre (λ_1, λ_2) et la liste D des dates de sortie des clients.

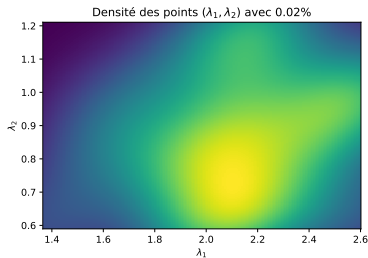
Client	Date arrivée	Date début de service	Temps de service	Date fin de service
1	9.23	9.23	4.28	13.51
2	13.48	13.51	24.65	38.16
3	16.00	38.16	1.73	39.89
4	16.93	39.89	0.68	40.57
5	17.48	40.58	0.18	40.75
...				

TABLE – Simulation d'une file d'attente avec $\lambda_1 = 0.2$ et $\lambda_2 = 0.1$

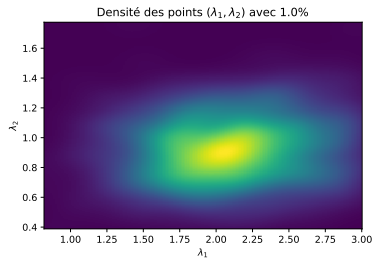
Approximate Bayesian Computation : Notre situation

- On génère 100000 cas fictifs, c'est à dire 100000 couples (λ_1, λ_2) et on simule la file d'attente pour chaque (λ_1, λ_2) et on s'intéresse à la liste "Date fin de service" pour chaque simulation.
- Une question fondamentale dans la création de notre algorithme ABC est le choix du critère de sélection pour faire la sous-sélection et obtenir une distribution a posteriori. Nous avons décidé de considérer 2 pseudo-distances pour comparer deux listes de dates de fin de service.
 - 1 La première est la distance euclidienne (norme 2). On rallonge la liste trop courte en répétant le dernier terme.
 - 2 La deuxième est la distance entre la médiane des écarts des date de service des clients 2 à 2.

Approximate Bayesian Computation : Densités



(a) Très restrictif



(b) Moins restrictif

FIGURE – Représentation des densité pour la première distance énoncée

Approximate Bayesian Computation : Densités

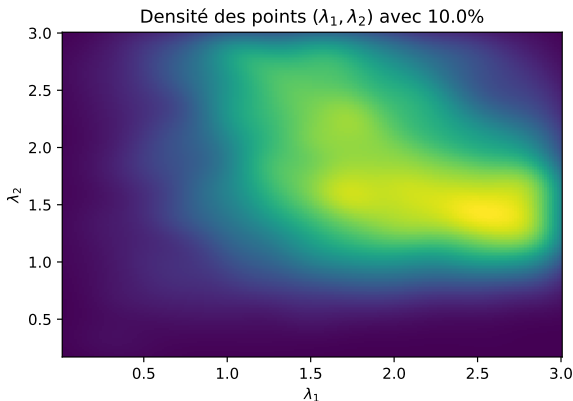


FIGURE – Représentation d'une densité pour la deuxième distance énoncée

Description de la loi de Weibull

- La loi exponentielle ne comporte qu'un seul paramètre, ce qui limite les possibilités. La loi de Weibull possède deux paramètres : k et λ , ce qui fait l'intérêt de cette loi.
- La densité de la loi est donnée par :

$$f(x; k, \lambda) = \frac{k}{\lambda} \left(\frac{x}{\lambda} \right)^{k-1} e^{-(x/\lambda)^k}$$

,

- Pour $k = 1$ on retrouve une loi exponentielle.

Description de la loi de Weibull

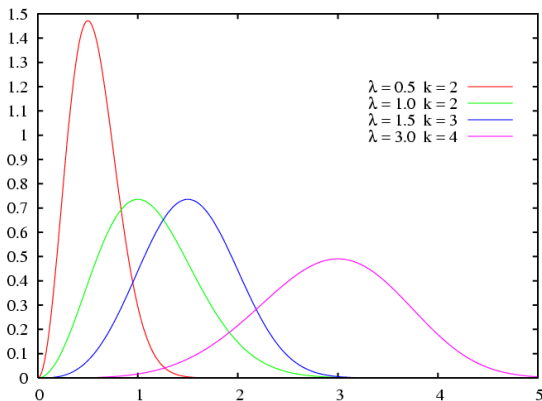


FIGURE – Densité de lois de Weibull

Génération d'une loi de Weibull

- Soit $X \sim \text{Weibull}(\lambda, k)$, $\lambda > 0, k > 0$

On a $F_X(t) = 1 - e^{-(t/\lambda)^k}$, donc

$$u = 1 - e^{-(t/\lambda)^k} \Leftrightarrow t = \lambda \times [-\ln(1 - u)]^{1/k} = F_X^{-1}(u)$$

- Méthode d'inversion : soit $U \sim \mathcal{U}[0, 1]$, alors $F_X^{-1}(U) \sim \text{Weibull}(\lambda, k)$.

Exemple

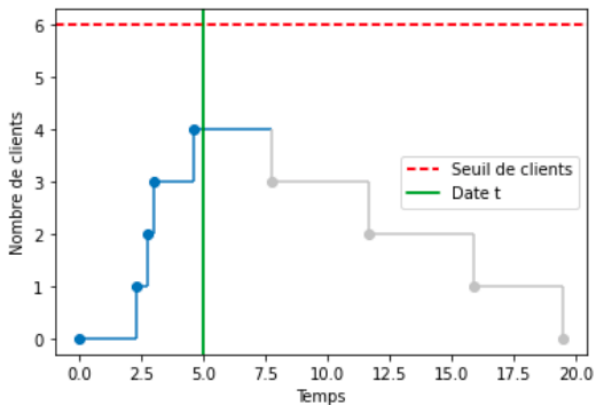


FIGURE – Simulation pour $\lambda_1 = 1$, $\lambda_2 = 2$, $k_1 = 0.5$ et $k_2 = 1$

Influence de λ_1 et λ_2

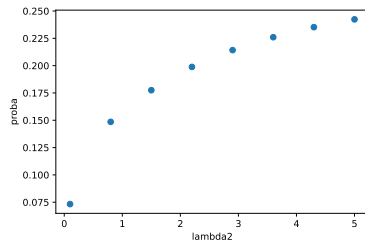
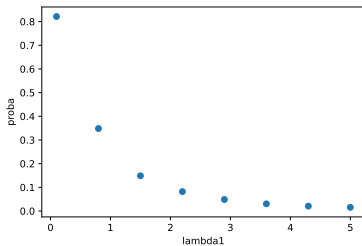


FIGURE – Représentation de la probabilité de dépassement du nombre de clients dans la file d'attente en fonction de λ_1 et λ_2

Influence de k_1 et k_2

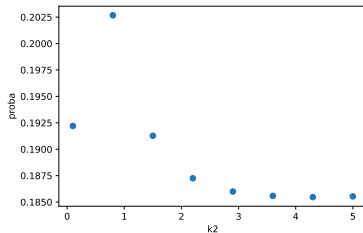
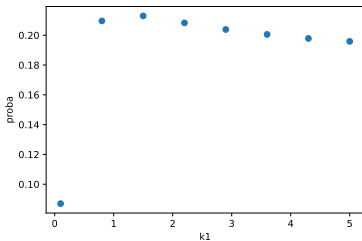


FIGURE – Représentation de la probabilité de dépassement du nombre de clients dans la file d'attente en fonction de k_1 et k_2

Comparaison des deux lois

- L'utilisation de lois de Weibull nous permet d'avoir, pour deux variables de même espérance, deux variances différentes.
- Il est légitime pour la modélisation d'une file d'attente de considérer que le temps de service et la durée entre deux clients n'ont pas, à espérance égale, la même variance.
- Un magasin d'une grande chaîne servant des cafés n'aura par exemple qu'une faible variance pour le temps de service.

Comparaison des deux lois

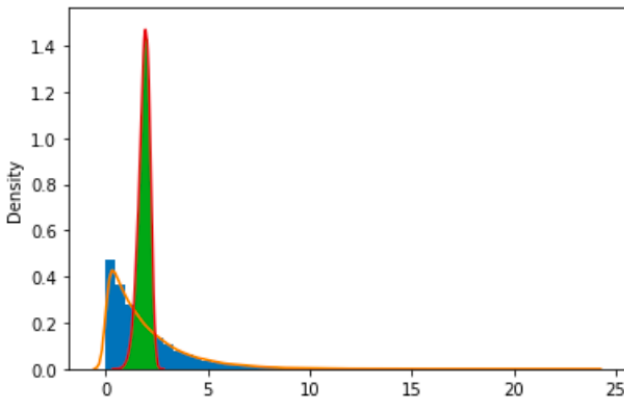


FIGURE – Deux lois de même espérance

Probabilités obtenus

- On simule des files d'attentes de deux manières différentes.
- Dans les deux cas, on utilise une même loi exponentielle pour simuler la durée entre deux arrivées de clients.
- On utilise une loi de Weibull de paramètre (1,8) et une loi exponentielle d'espérance égale pour simuler les temps de service des clients.
- On obtient dans les deux cas une probabilité proche de 0.45.

Boxen plot des simulations

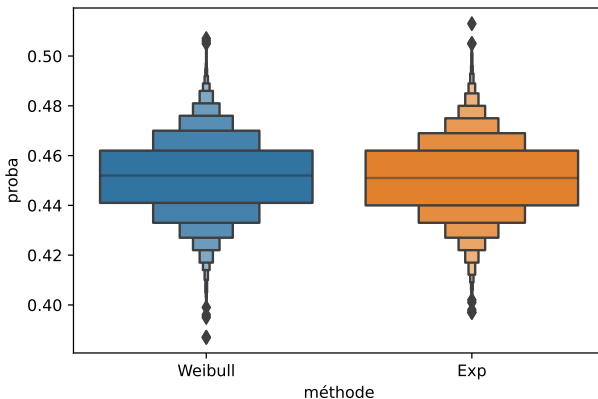


FIGURE – Probabilités obtenues selon la méthode

Majorant du nombre de générations de $Exp(\lambda_1)$ nécessaires

I

On recherche le plus petit $M \in \mathbb{N}^*$ qui vérifie :

$$P\left(\sum_{i=1}^M X_i \geq t\right) \geq s \quad \text{avec} \quad X_1, \dots, X_M \sim Exp(\lambda_1) \quad \text{et} \quad s \text{ proche de } 1$$

Cela nous assure que lorsque nous générerons M arrivées de clients, nous atteindrons dans une grande majorité des cas le temps t . Cela revient à déterminer M tel que :

$$P(\Gamma(M, \lambda_1) > t) \geq s$$

$$1 - P(\Gamma(M, \lambda_1) < t) \geq s$$

$$F_{\Gamma(M, \lambda_1)}(t) \leq 1 - s$$

Majorant du nombre de générations de $Exp(\lambda_1)$ nécessaires

II

Comme M est entier, la loi Gamma correspond à une loi d'Erlang et sa fonction de répartition est donnée par la formule suivante :

$$F_{\Gamma(M,\lambda)}(t) = 1 - \sum_{i=0}^{M-1} e^{-\lambda t} \frac{(\lambda t)^i}{i!}$$

On cherche donc M tel que :

$$\sum_{i=0}^{M-1} e^{-\lambda_1 t} \frac{(\lambda_1 t)^i}{i!} \geq s$$