

MA-510-34-1

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

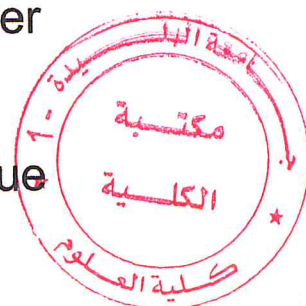
Université SAAD DAHLAB de Blida
Faculté des sciences
Département de Mathématiques



MEMOIRE DE FIN D'ETUDE

en vue de l'obtention du diplôme de Master
en Mathématiques

Option : Modélisation stochastique et statistique



THEME :

**Méthode de MONTE
CARLO, théorie et
application sur les risques de
crédit d'un portefeuille**

Réalisé par :

- Bouabdallah Sihem
- Djerboua Fatima Zohra

Encadré par :

Mr. Rassoul A.

Les membres du jurys :

- Mr. Tami O. (président)
- Mr. Frihi R. (examinateur)
- Mr. Boukhari M. (examinateur)

Promotion 2018

MA-510-34-1

Remerciement

*Tout d'abord, nous tenons à remercier dieu, le généreux,
le tout puissant qui nous a donnés la force et le courage, la volonté
et les moyens nécessaires pour réaliser ce modeste travail.*

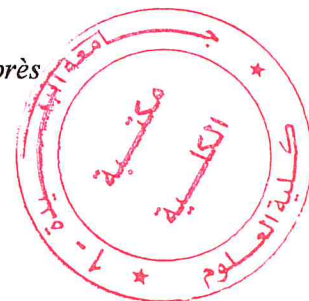
*Notre première reconnaissance va à notre encadreur le professeur RASSOUL Abdelaziz
d'avoir accepté de diriger ce travail, ses apports et ses conseils toujours
judicieux et ses encouragements . nous lui dis, Merci.*

*Ensuite pour messieurs TAMI O. et FRIHI R. les membres du jury qui ont accepté d'évaluer
et de juger notre travail.*

*Notre gratitude va également pour nos enseignants de Master qui nous ont donné l'envie de
poursuivre dans cette voie,*

*Nos remerciements s'adressent aussi aux enseignants et tout membres du département
des mathématiques qui nous ont aidée tout au long de notre cursus universitaire
en particulier monsieur Houssine et madame Noria qui
sont toujours à notre service.*

*Enfin, notre sincères remerciements vont à tous ceux qui ont contribué de près
ou de loin à la réalisation de ce mémoire, pour leurs soutien
et leurs encouragements, ... merci à tous.*



BOUABDALLAH S. et DJERBOUA F

Dédicace

Je dédie ce modeste travail à :

• A mes très chères parents qui m'ont entouré d'amour, m'encouragé et qui m'ont guidé durant les moments les plus pénibles de ce long chemin, ma mère qui a été à mes côtés et m'a soutenu durant toute ma vie et mon père qui a sacrifié toute sa vie afin de me voir devenir ce que je suis, merci mes parents et que dieu vous protège contre tout mal.

Mes frères : MOUHAMED et SOUHAIB

Mes sœurs : NESRINE et RITADJ

Ma belle-mère et le père de mon mari (papa)

Mon cher mari AMINE qui m'a toujours aidé et m'a encouragé

Ma belle-sœur SARA et mes frères ABDELLAH et YUCEF et sa femme et leur petit garçon

ABDELMALEK

Ma collègue de binôme SIHEM et sa famille.

Mes grandes mères et mes grands pères

Mes tantes

Mes oncles et leurs femmes et a les petites IBTIHEL et OTHMAN

A Toutes la famille « DJERBOUA » et « LAALOUTI » sans exception et toutes mes amies

surtout DJAHIDA et SOUMIA

Fatima Zahra D.

Dédicace

Je dédie ce mémoire à...

Mes très chers parents ,

Grace à dieu et à ceux que je suis devant vous aujourd'hui, leur soutien sans faille, sans qui rien n'aurait été possible, tout au long de mon cursus vous représentez pour moi le symbole de la bonté par excellence, la source de tendresse, et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de vos sacrifices que vous avez consentis pour ma formation.

Mes très chers sœurs et frères ,

Une très bon vie et meilleures voeux, Je vous souhaite un très bon avenir, plein de joie, de bonheur, réussite et de sérénité,

tous mes enseignants,

A ma chère amie depuis l'enfance Souad B., mes précieuse amies Fatima D. qui a partagé avec moi ce modeste travail, Sakina B., Khedidja L., Amina B., Khedidja C., Nabila, Zineb et tous ce qui m'aiment de prés ou de loin,

Merci à tous.

Sihem B.

Table des matières

0.1		
0.2	Résumé	6
0.3	Abstract	6
0.4	Introduction générale	7
1	Généralités sur les lois de probabilités	9
1.1	INTRODUCTION	9
1.2	Variables aléatoires	9
1.2.1	variable discrète	9
1.2.2	Exemples	10
1.2.3	Variables continues	14
1.2.4	Exemples	14
1.3	Rappel sur les théorie des probabilités	17
1.3.1	Intervalles de confiance	17
1.4	Théorèmes limites	18
1.4.1	Introduction	18
1.4.2	Loi (faible) des grands nombres	19
1.4.3	Loi forte des grands nombres	19
1.4.4	Théorème de limite centrale	19
1.4.5	Application du Théorème Central Limite	21
2	Méthodes de Simulation	23
2.1	Introduction	23
2.2	Méthode d'inversion de la fonction de répartition	23
2.2.1	Algorithme de la transformation inverse	24
2.3	La méthode de Box-Muller	26
2.3.1	Algorithme de la méthode Box-Muller	28
2.4	Simulation par la méthode de rejet	28
2.4.1	Algorithme	29
2.4.2	Simulation d'une variable normale	30
2.4.3	Calcul d'intégrale par la méthode de rejet	30

3	Méthode de Monté Carlo (MC)	33
3.1	Introduction	33
3.2	Histoire de MC	34
3.3	Principe de la méthode de Monté Carlo	34
3.3.1	Calcul d'intégrale par la méthode de Monte Carlo	35
3.3.2	Vers l'évaluation de l'erreur	37
3.3.3	Intégration par simulation de Monte-Carlo	39
3.3.4	Calcul d'aire (Approximation de π)	41
3.4	Techniques de réduction de la variance	42
3.4.1	Introduction	42
3.4.2	Echantillonnage préférentiel (importance sampling)	43
3.4.3	Variation de contrôle	47
3.4.4	Échantillonnage stratifié	50
3.4.5	Variation antithétiques	53
3.4.6	Entropie Croisé (CE)	56
3.5	Application de la méthode de réduction de la variance	58
4	Méthodes de Monte Carlo pour le risque de crédit d'un portefeuille	59
4.1	Introduction	59
4.2	Modélisation des pertes du crédit d'un portefeuille	60
4.3	Mesures de risques	62
4.3.1	Dépendance de modélisation	63
4.4	Estimation des mesures de risque via Monte Carlo	64
4.4.1	Echantillonnage préférentiel (Importance Sampling (IS))	66
4.4.2	Échantillonnage préférentiel adaptatif	69
4.5	Modèles spécifiques	71
4.5.1	Le modèle de mélange de Bernoulli	71
4.5.2	Échantillonnage préférentiel en une étape	71
4.5.3	Échantillonnage préférentiel en deux étapes	73
4.5.4	Modèles de facteurs	74
4.5.5	Echantillonnage préférentiel	76
5	Présentation de R	78
5.0.6	Généralités	78
5.0.7	Installation des packages	79
5.0.8	Charger un package déjà installé	80
5.0.9	Exemple d'utilisation du package : rpart	80
5.0.10	Syntaxe générale	81
5.0.11	Densité μ	81
5.0.12	Commandes	81

5.0.13	Commandes	84
5.0.14	Quantiles	85
5.0.15	Commandes	86
5.0.16	Calculs	86
5.1	Simulation	87
5.1.1	La commande sample	87
5.1.2	Options	88
5.1.3	Les lois de probabilité	89
5.1.4	Aide intégrée au logiciel R	91
5.1.5	Quelques commandes complémentaires	91
5.1.6	Créer et manipuler des données	93
5.1.7	Les objets : création et types	93
5.1.8	Vecteurs	94
5.1.9	Matrices	95
5.1.10	Les listes	95
5.1.11	Création	95
5.1.12	Créer un tableau de données sous R	96
5.1.13	Calcul arithmétique et fonctions simples	96
5.1.14	Graphiques	97
5.1.15	Partitionner une fenêtre graphique	97
5.1.16	Les fonctions	98
5.1.17	Structure générale pour créer des fonctions	98
5.1.18	Éléments de programmation	98
5.1.19	Les instructions de sélection	98
5.1.20	L'expression if else :	99
5.1.21	Les instructions de répétition	99
5.1.22	L'expression for :	99
5.1.23	R est merveilleux !	99
5.2	Conclusion	99

Liste des tableaux

1.1	Lois Binomial pour l'expérience d'un dé	10
1.2	Approximation d'une lois Binomial par une lois de Poisson quand N assez grand	13
1.3	Table récapulatif des lois de probabilités	26
1.4	Table récapulatif des caractéristiques des lois de probabilités	26
1.5	Syntaxes des lois de probabilités sous R	34
3.1	Valeurs simulées de l'intégrale avec les bornes de confiances pour un seuil $\alpha = 0.05$	60
3.2	Approximation de la valeur de l'intégrale I en fonction de N	63
3.3	Approximation de π	64
3.4	Approximation de la valeur de l'intégrale I	66
3.5	Approximation de la valeur de l'intégrale I en fonction de N	75
4.1	Approximation de VaR et ES par MCC pour un modèle exponentiel avec $N = 1000$	88
4.2	Approximation de VaR et ES par MCC pour un modèle exponentiel avec $N = 10000$	88
4.3	Approximation de VaR et ES par IS pour un modèle expo- nentiel avec $N = 1000$	91
4.4	Approximation de VaR et ES par IS pour un modèle expo- nentiel avec $N = 10000$	91
4.5	Approximation de VaR et ES par IS pour un modèle normal avec $N = 1000$	92
4.6	Approximation de VaR et ES par IS pour un modèle normal avec $N = 10000$	92

Table des figures

3.1	Convergence de l'intégrale simulé par la méthode de Monte-Carlo.	38
3.2	Convergence de la valeur de l'intégrale	39
5.1	Diagramme en batons d'une v.a.binomial	83
5.2	Graphe d'une densité de la loi normale $N(5,2.25)$	84
5.3	Fonction de répartition d'une lois normale	86

العرض من هذه الرسالة هو تقديم طريقة مونت كارلو من أجل تحقيق هذا الهدف، فمنا أولاً بالتعامل مع قوانين الاحتمالات المعقدة المطبقة في عملنا و عرض عدم على البرامج الإحصائية R، ثم قدمنا بعض أساليب المحاكاة مع أمثلة بسيطة من التطبيقات، بعد ذلك حددنا طريقة تركيب مونت كارلو مع تطبيق على أمثلة بسيطة لهذا العرض، في النهاية تعاملنا مع مشكلة القيمة المعرضة للخطر (Var) والعجز المتوقع (ES) من قبل CMC، IS ومفترات CE على أساس طريقة مونت كارلو.

0.1

0.2 Résumé

L'objectif de ce mémoire est de présenter la méthode de Monté Carlo . Afin d'atteindre cet objectif nous avons abordé dans un premier temps les lois de probabilités usuelles et applicables dans notre travail et une présentation générale sur le logiciel statistique R ,ensuite nous avons présenté quelques méthodes de simulation avec des simples exemples d' applications, après nous avons défini la méthode de monté carlo avec une application sur simples exemples.

A cet effet à la fin nous avons traité le problème de la valeur-à-risque (VaR) et le déficit attendu (ES) par les estimateurs CMC , IS et CE en se basant par la méthode MONTE CARLO.

0.3 Abstract

The purpose of this thesis is to present the Monté Carlo method. In order to achieve this objective, we first approached the usual and applicable probability laws in our work and a general presentation on R statistical software, then we presented some simulation methods with simple examples of applications, after we defined the monte carlo method with an application on simple examples.

For this purpose, in the end we dealt with the problem of the value-at-risk (VaR) and the expected deficit (ES) by the CMC , IS and CE estimators based on the MONTE CARLO method.

0.4 Introduction générale

La modélisation et la simulation occupent un vaste champ pour les recherches relatives à l'enseignement scientifique et technologique, exploré depuis antiquité. La notion de modèle en mathématique se présente sous un double aspect : d'une part, la mathématique permet de modéliser, c'est-à-dire de représenter, toutes sortes de situations des objets et de structures du monde réel, l'étude mathématique ou les simulations informatiques de ces représentations nous informant ; lorsque les représentations sont bonnes sur le monde réel, d'autre part, une notion précise et formelle de modèle est définie et étudiée en mathématique par la théorie des modèles, ce qui établit un lien précis entre objets syntaxiques (les formules) et structures mathématique puis produit sous la forme de théorèmes toutes sortes d'informations sur la nature de ce lien.

La simulation est la phase intermédiaire, cruciale et la plus délicate en mathématique et/ou en statistique pour modéliser la situation ou la nature d'un objet donné (fonctionnement d'un système ; phénomène physique , biologique ou chimique ; fabrication industrielle ; état d'un moteur ; capacité de la résistance d'un matériau au cours d'une opération précise, etc), dont son exécution se base du différents logiciels (Mathematica, Matlab, Scilab, Statistica, SPSS, R...) La simulation s'appuie à appliquer des méthodologies essentiellement adoptée sous forme des équations différentielles, calcul numérique, modèles théorique, etc. qui permettent à la réalisation un modèle (produit) des objets déjà existant.

En outre, leurs importances pour l'étude analytique des sciences et de la connaissance scientifique , la modélisation et la simulation peuvent dans l'enseignement scientifique , fonctionner comme un pont entre théorie scientifique et monde réel et être soumises à l'expérience.

Au fil du temps, le progrès scientifique occupe des aspects de la vie, ce qui rend les modèles et les méthodes d'évaluation des projets non seulement dans l'économie mais aussi dans le social et dans la production (agriculture, industrie,...) ont beaucoup évolué en même temps que les marchés financiers sont devenus de plus en plus complexes et de plus en plus risqués. Dans ce contexte, la problématique de l'évaluation que se soit des projets, des objets (système et des phénomènes etc ...) nécessite une méthode absolue.

Et dans cette perspective, le présent travail vise à étudier la méthode de MONTE CARLO qui est l'une des méthodes les plus importantes pour la modélisation et la simulation dans les domaines des mathématiques et des statistiques. Dans ce cadre nous avons essayé de mettre en évidence la méthode de MONTE CARLO et ses applications, afin par exemple d'évaluer un projet financier pour savoir si ceci se traduit par un gain net ou par une

perte.

Notre travail est organisé en quatre grands chapitres :

Chapitre 1 : représente des lois de probabilités usuelles, une présentation générale sur le logiciel ou le langage de programmation statistique R et quelque notions probabilistes générales qu'on va utiliser.

Chapitre 2 : est consacré pour la présentation des différentes méthodes de simulation des variables aléatoires, en particulier la méthode d'inversion de la fonction de répartition, la méthode de Box et Muller et la méthode de rejet.

Chapitre 3 : est pour présenter la méthode de Monte Carlo et ses applications dans différents domaines.

Chapitre 4 : application de la méthode de Monte Carlo pour estimer le risque de crédit d'un portefeuille.

En fin, notre travail est clôturé par une conclusion.

Chapitre 1

Généralités sur les lois de probabilités

1.1 INTRODUCTION

Les statisticiens et les probabilistes ont besoin dans leurs domaines à les modèles statistiques probabilistes et des langages statistiques, à cet effet nous allons présenter les lois de probabilité les plus utiles et les plus utilisées dans les diverses applications de la Statistique. Nous distinguons les lois des variables discrètes et celles des variables continues, avec les caractéristiques principales, ainsi que leurs propriétés essentielles. Aussi nous allons bien comprendre l'utilisation de logiciel statistique R.

1.2 Variables aléatoires

Une variable aléatoire est un nombre dépendant du résultat d'une expérience aléatoire. L'enjeu est la localisation de ce nombre : déterminer quelles sont ses chances de tomber sur telle ou telle partie de \mathbb{R} . Cette localisation conduit à associer à toute variable aléatoire une loi de probabilité sur \mathbb{R} .

Une variable aléatoire X est une fonction de l'ensemble fondamental Ω à valeurs dans \mathbb{R} ,

$$X : \Omega \rightarrow \mathbb{R}$$

1.2.1 variable discrète

par définition, les variables aléatoires discrètes prennent des valeurs entières discontinues sur un intervalle donné. Ce sont généralement le résultat

de dénombrement.

Caractéristiques

- l'espérance d'une loi discrète est comme suit :

$$\mathbb{E}(X) = \sum_{i=1}^n x_i P(X = x_i) \quad (1.1)$$

- la variance d'une loi discrète est comme suit :

$$V(X) = \mathbb{E}(X^2) - [\mathbb{E}(X)]^2 = \mathbb{E}((X - \mathbb{E}(X))^2) \quad (1.2)$$

- La fonction caractéristique d'une variable aléatoire réelle X est la fonction à valeurs complexes définie sur \mathbb{R} par :

$$\phi_X(t) = \mathbb{E}[e^{itX}] = \mathbb{E}[\cos(tX)] + i\mathbb{E}[\sin(tX)].$$

- La fonction génératrice

$$G_X(t) = \mathbb{E}[e^{tX}] = \mathbb{E}[\cos(tX)] + \mathbb{E}[\sin(tX)].$$

1.2.2 Exemples

Loi uniforme

Une distribution de probabilité suit une loi uniforme lorsque toutes les valeurs prises par la variable aléatoire sont équiprobables. Si n est le nombre de valeurs différentes prises par la variable aléatoire, $\forall i, P(X = x_i) = \frac{1}{n}$.

Dans le cas particulier d'une loi discrète uniforme où les valeurs de la variable aléatoire X correspondent au rang

$$x_i = i, \forall i \in [1, n], \mathbb{E}(X) = \frac{n+1}{2}, V(X) = \frac{n^2-1}{12}$$

La distribution des chiffres obtenus au lancer de dé (si ce dernier est non pipé) suit une loi uniforme dont la loi de probabilité est la suivante :

X	1	2	3	4	5	6
$P(X = x_i)$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

avec pour espérance : $\mathbb{E}(X) = \frac{1}{6} \sum_{i=1}^6 i = 3.5$ d'après (1.1) et pour variance

: $V(X) = \frac{1}{6} \sum_{i=1}^6 i^2 - [\mathbb{E}(X)]^2 = 2.92$ d'après (1.2), où les valeurs x_i correspondent au rang i de la variable X dans la série.

Loi de Bernoulli

Cette loi est celle de toute variable aléatoire X modélisant une expérience dont l'issue ne possède que deux alternatives de types "succès ou échec", "vrai ou faux", "marche ou arrêt", "pile ou face", etc. Un succès est représenté par l'événement $\{X = 1\}$ tandis que $\{X = 0\}$ correspond à un échec, $X(\Omega) = \{0, 1\}$. Puisque l'on a :

$$P[X = 1] = p, P[X = 0] = 1 - p.$$

*son espérance est $\mathbb{E}(X) = p$ d'après (1.1)
*sa variance est $V(X) = p(1 - p) = p \times q$ avec $p + q = 1$
d'après (1.2)
est appelée loi Bernoulli notée $X \rightsquigarrow \mathcal{B}(1; p)$

Loi Binomiale

Soit l'application $S_n : \Omega^n \rightarrow \mathbb{R}^n$

$S_n = X_1 + X_2 + \dots + X_i + \dots + X_n$ où X_i est une variable de Bernoulli. La variable binomiale S_n Représente le nombre de succès obtenus lors de la répétition de n épreuves identique et indépendantes, chaque épreuve ne pouvant donner que deux résultats possibles. Ainsi la loi de probabilité suivie par la somme de n variables de Bernoulli où la probabilité associée au succès est p , est la loi binomiale de paramètres n et p .

$$S_n : \Omega^n \rightarrow \mathbb{R}^n$$

$$S_n = \sum_{i=1}^n X_i \rightsquigarrow \mathcal{B}(n, p)$$

La probabilité que $S_n = k$; c'est à dire l'obtention de k succès au cours de n épreuves indépendantes est :

$$P(S_n = k) = C_n^k p^k q^{n-k}, k = 0, \dots, n.$$

Il est facile de démontrer que l'on a bien une loi de probabilité car :

$$\sum_{k=0}^n P(S_n = k) = \sum_{k=0}^n C_n^k p^k q^{n-k} = (p + q)^n = 1 \text{ car } p + q = 1$$

son espérance est $\mathbb{E}(S_n) = \sum_{k=0}^n \mathbb{E}(X_i) = \sum_{k=0}^n p = np$ avec $\mathbb{E}(X_i) = p$ X_i variable de bernoulli donc

$$\mathbb{E}(S_n) = np, V(S_n) = \sum_{k=0}^n V(X_i) = \sum_{k=0}^n pq = npq.$$

Loi de Poisson

La loi de poisson découverte au début du XIXe siècle par le magistrat français Siméon-Denis Poisson s'applique souvent aux phénomènes accidentels où la probabilité p est très faible ($p < 0.05$). Elle peut également dans certaines conditions être définie comme limite d'une loi binomiale.

Lorsque n devient grand, le calcul des probabilités d'une loi binomiale $P(S_n = k) = C_n^k p^k q^{n-k}$ devient très fastidieux. On va donc, sous certaines conditions, trouver une approximation de p_k plus maniable.

Comportement asymptotique : si $n \rightarrow \infty$ et $p \rightarrow 0$ alors

$$X : \mathcal{B}(n, p) \rightarrow \mathcal{P}(\lambda), \text{ avec } np \rightarrow \lambda.$$

Remarque 1.1 Cette approximation est correcte si $n \geq 50$ et $np \leq 5$.

Une variable aléatoire X à valeurs dans \mathbb{R} Suit une loi de poisson de paramètre λ ($\lambda > 0$) si les réels p_k sont données par :

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad \text{on note : } X \rightsquigarrow \mathcal{P}(\lambda)$$

l'espérance d'une variable aléatoire de poisson est $\mathbb{E}(X) = \lambda$, la variance d'une variable de poisson est $V(X) = \lambda$.

Remarque : Si X et Y sont deux variables indépendantes suivant des lois de poisson respectivement : $X \rightsquigarrow \mathcal{P}(\lambda)$ et $Y \rightsquigarrow \mathcal{P}(\mu)$. Alors :

$$X + Y \rightsquigarrow \mathcal{P}(\lambda + \mu)$$

Soit une loi binomiale de paramètres $\mathcal{B}(100, 0.01)$, les valeurs des probabilités pour k de 0 à 5 ainsi que leur approximation à 10^{-3} avec une loi de poisson de paramètre ($\lambda = np = 1$) sont données dans le tableau ci-dessous :

k	0	1	2	3	4	5
$P(X = k)$	0.366	0.370	0.185	0.061	0.015	0.000
approximation	0.368	0.368	0.184	0.061	0.015	0.003

Loi géométrique

La loi géométrique est une loi de probabilité apparaissant dans de nombreuses applications.

La loi géométrique de paramètre p ($0 < p < 1$) correspond au modèle suivant :

On considère une épreuve de Bernoulli dont la probabilité de succès est p et celle d'échec $q = 1 - p$.

On renouvelle cette épreuve de manière indépendante jusqu'au premier succès. On appelle X la variable aléatoire donnant le rang du premier succès.

Les valeurs de X sont les entiers naturels non nuls 1, 2, 3, ... La probabilité que $X = k$ est :

$$\mathbf{P}(X = k) = q^{k-1}p, k = 1, 2, 3, ..$$

On dit que X suit une loi géométrique de paramètre p .

– L'espérance d'une variable aléatoire X suivant une loi géométrique de paramètre p est :

$$\mathbb{E}[X] = \frac{1}{p}.$$

– et sa variance est :

$$V[X] = \frac{1 - P}{P^2} = \frac{q}{p^2}. \quad (\text{où } q = 1 - p \text{ est la probabilité d'échec})$$

Loi Hypergéométrique

Soit une expérience aléatoire dont le processus est constitué d'un tirage sans remise de n objets parmi N , c-à-d n épreuves de Bernoulli non indépendantes, avec :

$$p = P(\text{succès lors de la 1ère vérification})$$

$$q = P(\text{échec lors de la 1ère vérification}) = 1 - p$$

Soit X , une variable aléatoire discrète qui comptabilise le nombre de succès ou d'objets gagnants lorsqu'on tire simultanément n objets dans une urne contenant pN objets gagnants et qN objets perdants (avec $q = 1 - p$, soit un nombre total d'objets valant $pN + qN = N$).

X suit une loi hypergéométrique de paramètres n , N et p , notée $X \rightsquigarrow \mathcal{H}(n; N; p)$ avec :

$$p(x) = P(X = x) = \frac{C_x^{Np} C_{n-x}^{Nq}}{C_n^N} \text{ avec } x = 0, 1, 2, 3, \min n, Np$$

- son espérance est : $\mathbb{E}(X) = np$,
- sa variance est : $Var(X) = npq \left(\frac{N-n}{N-1} \right)$.

1.2.3 Variables continues

Une variable aléatoire continue X est une fonction qui associe à chaque résultat d'une expérience aléatoire un nombre réel. On note : $X : \Omega \rightarrow \mathbb{R}$

Caractéristiques

- l'espérance d'une loi continue est comme suit :

$$\mathbb{E}(X) = \int_{-\infty}^{\infty} x f(x) dx \quad (1.3)$$

- la variance d'une loi continue est comme suit :

$$V(X) = \mathbb{E}(X^2) - [\mathbb{E}(X)]^2 = [\mathbb{E}(X - \mathbb{E}(X))^2] \quad (1.4)$$

- Fonction caractéristique : Si X est une variable aléatoire qui admet pour fonction de densité f_x , sa fonction caractéristique est définie par :

$$\phi_X(t) = \mathbb{E}(e^{itX}) = \int_{-\infty}^{\infty} e^{itx} f_X(x) dx$$

- Fonction génératrice des moments : Si X est une variable aléatoire qui admet pour fonction de densité f_x , sa fonction génératrice des moments est définie par :

$$M_X(t) = \mathbb{E}(e^{tX}) = \int_{-\infty}^{\infty} e^{tx} f_X(x) dx$$

1.2.4 Exemples

Loi Uniforme On dit que X suit la loi uniforme sur $[a; b]$, ce que l'on note $X \rightsquigarrow \mathcal{U}([a, b])$ si elle est absolument continue, et admet pour densité :

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{si } x \in [a, b] \\ 0 & \text{sinon} \end{cases}$$

et pour fonction de répartition :

$$F_X(x) = \begin{cases} 0 & \text{si } x < a \\ \frac{x-a}{b-a} & \text{si } x \in [a, b] \\ 1 & \text{si } x \geq b \end{cases}$$

- L'espérance d'une variable aléatoire uniforme est : $\frac{a+b}{2}$ d'après (1.3)
- La variance d'une variable aléatoire uniforme est : $\frac{(b-a)^2}{12}$ d'après (1.4)
- La fonction génératrice des moments d'une variable aléatoire continue suivant une loi uniforme est :

$$M_X(t) = \mathbb{E}(e^{xt}) = \frac{e^{tb} - e^{ta}}{t(b-a)}$$

- Le moment d'ordre k d'une variable aléatoire continue suivant une loi uniforme :

$$m_k = \frac{1}{k+1} \sum_{i=0}^k a^i b^{k-i}.$$

Loi normale La loi normale (non centrée et non réduite) est la loi de probabilité absolument continue qui dépend de deux paramètres : son espérance $\mathbb{E}(X)$, un nombre réel noté μ , et son écart type, un nombre réel positif noté σ alors que σ^2 est sa variance $V(X)$.

De plus, de nombreuses autres lois statistiques peuvent être approchées par la loi normale, tout spécialement dans le cas des grands échantillons. une variable aléatoire X qui suit la loi normale est notée $X \rightsquigarrow \mathcal{N}(\mu, \sigma)$ ou $X \rightsquigarrow \mathcal{N}(\mu, \sigma^2)$.

- La densité de probabilité est donnée par la fonction $f : \mathbb{R} \rightarrow \mathbb{R}^+$ définie par :

$$f(t) = \frac{1}{\sigma^2 \sqrt{2\pi}} e^{-\frac{1}{2} \frac{(t-\mu)^2}{\sigma^2}}, \text{ pour tout } t \in \mathbb{R};$$

- La fonction de répartition $F : \mathbb{R} \rightarrow \mathbb{R}^+$ est définie par :

$$F(x) = \frac{1}{\sigma^2 \sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2} \frac{(t-\mu)^2}{\sigma^2}} dt, \text{ pour tout } x \in \mathbb{R};$$

- La fonction caractéristique $\phi(t) : \mathbb{R} \rightarrow \mathbb{C}$ définie par :

$$\phi(t) = e^{\mu it - \frac{1}{2} \sigma^2 t^2}$$

- La fonction génératrice des moments $M : \mathbb{R} \rightarrow \mathbb{R}$ est définie par :

$$M(t) = e^{\mu t + \frac{1}{2}\sigma^2 t^2} \text{ pour tout } t \in \mathbb{R}, \text{ où } \mu \in \mathbb{R} \text{ et } \sigma \in \mathbb{R}_+^*$$

Pour le cas où $\sigma = 0$, les fonctions de densité et de répartition ne sont pas définies. Ce cas correspond à un comportement dégénéré de la loi normale, parfois appelée loi normale impropre. C'est alors la mesure de Dirac au point μ .

Loi Exponentielle

- La loi exponentielle décrit la durée de vie d'un phénomène sans vieillissement (particule radioactive, temps d'attente, ...).
- La densité de probabilité d'une variable aléatoire continue suivant une loi exponentielle $\mathcal{E}(\lambda)$ est :

$$f(x) = \lambda e^{-\lambda x}, \quad x \in \mathbb{R}_+$$

où λ est un nombre réel strictement positif, d'espérance $\mathbb{E}(X) = \frac{1}{\lambda}$, et de $V(X) = \frac{1}{\lambda^2}$.

Loi Gamma Une variable aléatoire X suit une loi Gamma de paramètres k et θ (strictement positifs), ce que l'on note aussi $X \sim \Gamma(k, \theta)$ (où la lettre grecque Γ est la majuscule du γ si sa fonction de densité de probabilité peut se mettre sous la forme :

$$f(x; k, \theta) = \frac{x^{k-1} \exp^{-\frac{x}{\theta}}}{\Gamma(k) \theta^k}, \quad x > 0$$

où Γ , au dénominateur, désigne la fonction Gamma d'Euler.

Alternativement, la distribution Gamma peut être paramétrée à l'aide d'un paramètre de forme $\alpha = k$ et d'un paramètre d'intensité $\beta = \frac{1}{\theta}$:

$$f(x; \alpha, \beta) = x^{\alpha-1} \frac{\beta^\alpha \exp^{-\beta x}}{\Gamma(\alpha)}, \quad x > 0.$$

Loi Khi-deux ou χ^2 de Pearson Soient X_1, \dots, X_k , k variables aléatoires indépendantes suivant des lois normales de moyennes 0 et d'écart-type 1, alors par définition, la variable X , telle que :

$$X = \sum_{i=1}^k X_i^2$$

suit une loi du χ^2 à k degrés de liberté. On notera $\chi^2(k)$ ou χ_k^2 la loi de X .

La densité de probabilité de X notée f_x sera :

$$f_X(x; k) = \frac{1}{2^{\frac{k}{2}} \Gamma\left(\frac{k}{2}\right)} x^{\frac{k}{2}-1} \exp\left(-\frac{x}{2}\right) \text{ pour tout } x > 0,$$

où Γ est la fonction gamma.

Loi de Student Soit Z une variable aléatoire suivant la loi normale centrée et réduite et soit U une variable indépendante de Z et distribuée suivant la loi du χ^2 à k degrés de liberté. Par définition, la variable $T = \frac{Z}{\sqrt{\frac{U}{k}}}$ suit une

loi de Student à k degrés de liberté.

La densité de T , notée f_T , est donnée par :

$$f_T(t) = \frac{1}{\sqrt{k\pi}} \frac{\Gamma\left(\frac{k+1}{2}\right)}{\Gamma\left(\frac{k}{2}\right)} \left(1 + \frac{t^2}{k}\right)^{-\frac{k+1}{2}} \text{ pour } k > 0$$

où Γ est la fonction Gamma d'Euler.

La densité f_T associée à la variable T est symétrique, centrée sur 0, en forme de cloche.

Loi de Fisher-Snedecor On dit qu'une variable aléatoire X suit la loi de Fisher-Snedecor de paramètres m et n si sa densité vaut :

$$f(x) = \frac{\Gamma\left(\frac{m+n}{2}\right)}{\Gamma\left(\frac{m}{2}\right) \Gamma\left(\frac{n}{2}\right)} n^{\frac{n}{2}} m^{\frac{m}{2}} \frac{x^{\frac{m}{2}-1}}{(mx+n)^{\frac{m+n}{2}}}$$

Suivant les valeurs de m et n , X admet alors une espérance et une variance qui sont :

$$\mathbb{E}(X) = \frac{n}{n-2} \text{ sin } \geq 3$$

$$V(X) = \left(\frac{n}{n-2}\right)^2 \frac{2(m+n-2)}{m(n-4)} \text{ sin } \geq 5$$

1.3 Rappel sur les théorie des probabilités

1.3.1 Intervalles de confiance

Question : L'estimateur \hat{I}_N donne une valeur approchée de I , d'autant meilleure que N est grand. Comment quantifier précisément l'erreur ?

Réponse : On construit un intervalle de confiance au niveau 0.95, i.e. un intervalle $[a_N, b_N]$ tel que

$$P(I \in [a_N, b_N]) \geq 0.95$$

Construction basée sur le théorème central limite :

$$P\left(|\hat{I}_N - I| < c \frac{\text{Var}(f(X))^{1/2}}{\sqrt{N}}\right) \xrightarrow{N \rightarrow \infty} \frac{2}{\sqrt{2\pi}} \int_0^c e^{-x^2/2} dx$$

Le membre de droite vaut 0.95 si $c = 1.96$

$$P\left(I \in \left[\hat{I}_N - 1.96 \frac{\text{Var}(f(X))^{1/2}}{\sqrt{N}}, \hat{I}_N + 1.96 \frac{\text{Var}(f(X))^{1/2}}{\sqrt{N}}\right]\right) \simeq 0.95$$

où

$$\sigma_N = \left(\frac{1}{N} \sum_{k=1}^N f(X^{(k)})^2 - \hat{I}_N^2\right)^{1/2}$$

Conclusion : Il n'y a aucun intervalle borné de \mathbb{R} dont on puisse dire avec certitude qu'il contient I , mais il y a des intervalles, dits intervalles de confiance, dont on peut dire qu'ils contiennent I avec une probabilité très proche de 1.

1.4 Théorèmes limites

1.4.1 Introduction

Deux théorèmes mathématiques ont une place particulière en théorie des probabilités et en statistique : la loi des grands nombres et le théorème central limite. Ils interviennent dans l'étude de phénomènes aléatoires comportant un grand nombre de v.a.r. indépendantes de même loi. Par exemple, pour le premier cité, il apparaît lorsque l'on étudie la proportion de "pile" dans un jeu de pile ou face, ou encore la moyenne de lancers de dé successifs. Quand au second, il nous donne de façon informelle une estimation précise de l'erreur que l'on commet en approchant l'espérance mathématique par la moyenne arithmétique.

1.4.2 Loi (faible) des grands nombres

Théorème 1.1 Soient X_1, \dots, X_n des v.a.r. indépendantes, de même loi, et admettant une variance. On note $\mu = \mathbb{E}(X_1)$ Alors, pour tout $\varepsilon > 0$

$$P\left(a \frac{X_1 + \dots + X_n}{n} - \mu > \varepsilon\right) \xrightarrow{n \rightarrow \infty} 0.$$

Dans ce cas, on dit que la moyenne arithmétique $\frac{X_1 + \dots + X_n}{n}$ converge en probabilité vers l'espérance mathématique lorsque n tend vers $+\infty$.

1.4.3 Loi forte des grands nombres

Soit $(X_i, i \geq 1)$, une suite de réalisations de la variable aléatoire X . On suppose que $\mathbb{E}(|X|) < +\infty$. Alors, pour presque tout ω (i.e. $\exists N \in \Omega$ avec $p(N) = 0$ et $\omega \notin N$)

$$\mathbb{E}(X) = \lim_{n \rightarrow +\infty} \frac{1}{n} (X_1(\omega) + \dots + X_n(\omega)).$$

1.4.4 Théorème de limite centrale

Soit $(X_i, i \geq 1)$ une suite de variables aléatoires indépendantes et de même loi que X , telles que $\mathbb{E}|X^2| < +\infty$

On note σ^2 la variance de X : Alors la suite

$$\frac{\sqrt{n}}{\sigma} \varepsilon_n = \frac{\sqrt{n}}{\sigma} \left(\frac{1}{n} \sum_{i=1}^n (X_i - \mathbb{E}(X)) \right) = \frac{1}{\sigma \sqrt{n}} \sum_{i=1}^n (X_i - \mathbb{E}(X))$$

Converge en loi vers une gaussienne centrée réduite. C'est-à-dire :

$$\forall a < b, \lim_{n \rightarrow +\infty} P\left(\frac{\sigma}{\sqrt{n}} a \leq \varepsilon_n \leq \frac{\sigma}{\sqrt{n}} b\right) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{x^2}{2}} dx$$

Cette méthode ne dépend pas de la régularité de f qui doit être juste mesurable.

Souvent on cherche à calculer une intégrale plus générale :

$$I \approx \int_{\mathbb{R}^d} g(x) f(x) dx = \int g(x_1, \dots, x_d) f(x_1, \dots, x_d) dx_1, \dots, dx_d$$

En considérant f positive et $\int_{\mathbb{R}^d} f(x) dx = 1$, on obtiendra alors :

$$I = \mathbb{E}(g(X))$$

où X est une variable aléatoire à valeur dans \mathbb{R}^d de loi f . Ainsi on peut approcher l'intégrale I par l'expression suivante :

$$I \approx I_N = \frac{1}{N} \sum_{i=1}^N g(X_i)$$

C'est la loi forte des grands nombres qui permet de justifier la convergence de la méthode et le théorème de la limite centrale qui précise la vitesse de convergence. Pour avoir une idée de l'intérêt de la méthode, il faut pouvoir évaluer l'erreur commise définie par :

$$|\varepsilon_N| = \left| \mathbb{E}(g(X)) - \frac{1}{N} \sum_{i=1}^N g(X_i) \right|$$

Le théorème de la limite centrale donne un asymptotique de l'erreur ε_N , mais de nature aléatoire. Dans ce cas, la loi de l'erreur finit par ressembler à une loi gaussienne centrée. Dans les applications, on remplace ε_N par une gaussienne centrée de variance $\frac{\sigma^2}{N}$.

Le théorème centrale limite ne permet jamais de borner l'erreur (support d'une gaussienne est égale à \mathbb{R} en entier). On présente souvent l'erreur de la méthode de Monte-Carlo, soit en donnant l'écart-type de l'erreur ε_N , soit en donnant un intervalle de confiance à 95% pour le résultat. L'intervalle de confiance de la méthode à 95% est :

$$[I_1, I_2] = \left[I_N - 1.96 \frac{\sigma}{\sqrt{N}}, I_N + 1.96 \frac{\sigma}{\sqrt{N}} \right]$$

En général le calcul de la valeur de σ est approximé par la méthode de Monte Carlo :

$$\frac{1}{N} \sum_{i=1}^N [g(X_i)]^2 - \left[\frac{1}{N} \sum_{i=1}^N g(X_i) \right]^2 \longrightarrow \sigma^2$$

La vitesse de convergence est en $\frac{1}{\sqrt{N}}$, ce qui n'est pas très rapide, mais c'est parfois la seule méthode abordable, de plus cette vitesse ne change pas si on est en grande dimension, et elle ne dépend pas de la régularité de la fonction.

1.4.5 Application du Théorème Central Limite

Une application immédiate du théorème central limite donne une méthode très simple de g en ération de variables aléatoires normales. On sait que si on a n variable aleatoire i.i.d. d'esperance μ et de variance σ^2 , en notant S leur somme, on a :

$$\frac{S - n\mu}{\sqrt{n}\sigma} \xrightarrow{\text{loi}} \mathcal{N}(0, 1).$$

Dans de nombreux cas, cette convergence est assez rapide et permet d'assez bonnes approximations pour $n \simeq 10$).

On utilise ici les propriétés des variables aléatoires de loi $\mathcal{U}_{[0;1]}$:

$$\mathbb{E}(U) = \frac{1}{2}, \quad \text{Var}(U) = \frac{1}{12}$$

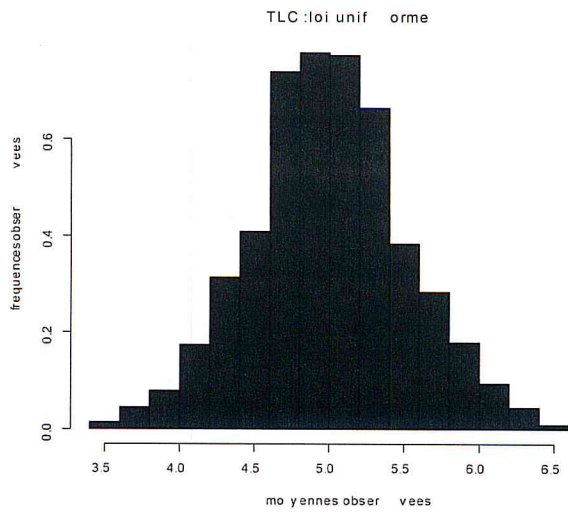
et on simule 12 variables aléatoires indépendantes uniformes sur $[0; 1]$ et on en fait la somme. La variance de cette somme vaut 1 et il faut ensuite la centrer en lui retranchant son esperance, c'est a dire :

$$X = \sum_{i=1}^{12} U_i - 6$$

La loi de cette variable est très proche de la loi normale $\mathcal{N}(0; 1)$.

code R :

```
a<-0
b<-10
m=(a+b)/2
sigma<-(b-a)/sqrt(12)
Nbsimul<-1000
taille<-30
Result<-NULL
for(i in 1:Nbsimul)
{
tirage<-runif(taille,0,10)
Result[i]<-mean(tirage)
}
hist(Result,freq=FALSE,col="blue",
ylim=c(0,dnorm(m,m,sigma/sqrt(taille))),
main = "TLC :loi uniforme",xlab="moyennes observees",ylab="frequences observees",
curve(dnorm(x,m,sigma/sqrt(taille)),add=TRUE,col="red",lty=3,lwd=2)
```



Chapitre 2

Méthodes de Simulation

2.1 Introduction

Ce chapitre a pour objet de présenter certaines des méthodes permettant de simuler la génération de telles variables aléatoires.

2.2 Méthode d'inversion de la fonction de répartition

Les transformations de variables nous permettront de simuler des échantillons fictifs d'une variable aléatoire à partir d'un ensemble de nombres aléatoires uniformément distribués sur $[0, 1]$.

Dans certains cas, par exemple quand on peut expliciter complètement la fonction de répartition, pour atteindre ce but il suffira d'appliquer simplement une formule de transformation.

La méthode de la transformation inverse peut s'utiliser pour générer n'importe quelle variable aléatoire à condition que l'on connaisse analytiquement la fonction de répartition inverse $F^{-1}(x)$.

Elle est basée sur le théorème suivant :

Théorème 2.1 (*Théorème de la transformation inverse*) Soit X une variable aléatoire et $F(x)$ sa fonction de répartition. Si $F(x)$ est continue et strictement croissante, alors $Y = F(X)$ est une variable aléatoire dont la distribution est uniforme sur $[0; 1]$. Dit autrement, si U est une variable aléatoire uniforme sur $[0; 1]$, la variable $F^{-1}(U)$ est une variable aléatoire dont F est sa fonction de répartition.

Preuve: On a

$$\begin{aligned}y &= F(x) = \Pr(X \leq x) \\ &= \Pr(F(X) \leq F(x)) \\ &= \Pr(Y \leq y) = F_Y(y)\end{aligned}$$

ce qui prouve le théorème. ■

Une application directe de ce théorème permet, à partir d'une loi uniforme, de générer n'importe quelle autre loi. Ainsi, pour générer un échantillon fictif issu d'une variable aléatoire X , il faut connaître $F(x)$ et avoir une suite de nombres y_1, y_2, \dots, y_n issus d'une variable $\mathcal{U}([0, 1])$. L'égalité $x = F^{-1}(y)$ permet d'obtenir l'échantillon

$$x_1 = F^{-1}(y_1), x_2 = F^{-1}(y_2), \dots, x_n = F^{-1}(y_n)$$

issu d'une population distribuée selon F . La seule difficulté est celle de calculer F^{-1} en connaissant F . Dans certains cas cela n'est pas possible de manière explicite, et dans ces cas d'autres techniques, que l'on verra plus loin, permettent de simuler un échantillon de X .

2.2.1 Algorithme de la transformation inverse

1. Générer U de distribution $\mathcal{U}([0, 1])$.
2. Calculer $X = F^{-1}(U)$.

Exemple 2.1 (*Loi exponentielle*)

Rappelons qu'une v. a. exponentielle de paramètre λ , de fonction de répartition

$$F(x) = 1 - \exp(-\lambda x), x \geq 0.$$

Calculons sa fonction réciproque (F étant clairement bijective).

$$F(x) = u \iff 1 - \exp(-\lambda x) = u \iff x = \frac{-\ln(1-u)}{\lambda}.$$

Ainsi pour générer un n-échantillon selon une $\text{Exp}(\lambda)$ on pose :

$$(X_1, \dots, X_n) = \left(\frac{-\ln U_1}{\lambda}, \dots, \frac{-\ln U_n}{\lambda} \right),$$

Remarquons que si U est une v. a. uniforme, $(1-U)$ est une aussi v. a. uniforme. Donc, si U est une v. a. uniforme, alors $X = -\log U$ est une v. a. de loi exponentielle négative (de moyenne 1).

Voici maintenant une implémentation en R.

```

fexp <-function(n,lamda)return(-log(runif(n))/lamda)
> ##simulation d'une exp(5)
n=500
lamda<-5
x<-fexp(n,lamda)
##comparaison histogramme/densit
hist(x,freq=FALSE)##freq=FALSE pour aire=1
curve(dexp(x,lamda),xlim=c(0,max(x)),col="red",add=TRUE)

```

Notons au passage que pour que cette méthode soit efficace (au sens informatique), il faut que F^{-1} soit rapidement calculable.

La méthode d'inversion prend une forme particulière lorsque le domaine de définition de variable aléatoire X à simuler est un ensemble fini $\{x_1, \dots, x_k\}$, $k \in \mathbb{N}$. Sans perte de

généralité, nous pouvons supposer que $x_1 < x_2 < \dots < x_k$. Ainsi on a donc

$$X = F^{-1}(U) = \begin{cases} x_1, & \text{si } U \leq p_1 \\ x_j, & \text{si } \sum_{l=1}^{j-1} p_l < U \leq \sum_{l=1}^j p_l \\ x_k, & \text{si } \sum_{l=1}^{k-1} p_l < U \leq \sum_{l=1}^k p_l = 1, \end{cases}$$

avec

$$p_j = Pr(X = x_j), j = 1, \dots, k.$$

Exemple 2.2 (Loi de Bernoulli)

Rappelons que la loi de Bernoulli de paramètre $p \in (0, 1)$ et notée $Ber(p)$ est définie par

$$Pr(X = j) = \begin{cases} p & \text{si } j = 1 \\ 1 - p & \text{si } j = 0 \end{cases}$$

Ainsi un n -échantillon distribué selon une $Ber(p)$ est générée selon

$$(X_1, \dots, X_n) = (1_{\{U_1 \leq p\}}, \dots, 1_{\{U_n \leq p\}}), U_1, \dots, U_n \overset{iid}{\sim} \mathcal{U}(0, 1).$$

Cette méthode peut être lente car elle peut nécessiter beaucoup de test de la forme $\sum p_l < U \leq \sum p_l$ et il peut donc être rentable de réordonner les x_j par valeurs de probabilités décroissantes—on augmente ainsi la probabilité que $U \leq p_1$ par exemple.

Remarque 2.1 La fonction *runif* permet de générer des séries de nombres aléatoires, tirés d'une loi uniforme ;

On a donc un générateur de nombres pseudo aléatoires classique.

Définition 2.1 *On définit un nombre aléatoire comme la réalisation d'une variable aléatoire distribuée uniformément dans l'intervalle $[0, 1]$. Une suite de nombres aléatoires sera dans la suite de ce mémoire un échantillon issu d'une population distribuée selon la loi $\mathcal{U}([0, 1])$ dont les éléments sont indépendants les uns des autres.*

Définition 2.2 *Une suite de nombres pseudo-aléatoires est une suite de nombres possédant les mêmes propriétés qu'une suite de nombres aléatoires, mais générée à travers une procédure déterministe. Les nombres pseudo-aléatoires sont donc générés par des fonctions mathématiques ou par des algorithmes.*

On pourrait ainsi créer un vecteur de nombres au hasard prenant comme valeur 0 ou 1, en écrivant par exemple :

```
> x<-runif(100)
> for (i in 1 : length(x)){
  if (x[i]<= 0.5) x[i]=0
  else x[i]=1
}
```

2.3 La méthode de Box-Muller

La méthode de Box et Muller (1958) est une autre méthode pour obtenir des réalisations de variables dont la distribution est normale. Cette méthode est fondée sur la transformation des coordonnées polaires en coordonnées cartésiennes.

Théorème 2.2 (Théorème de Box et Muller) *Soient U_1 et U_2 deux variables aléatoires uniformes et indépendantes sur l'intervalle $[0; 1]$. Les variables*

$$\begin{aligned}Z_1 &= \sqrt{-2 \log(U_1)} \cos(2\pi U_2) \\Z_2 &= \sqrt{-2 \log(U_1)} \sin(2\pi U_2)\end{aligned}$$

sont alors deux variables aléatoires normales centrées réduites indépendantes.

Preuve: La démonstration est une application instructive de ce que nous avons vu à propos des transformations de variables continues. Soient Z_1 et Z_2 deux variables aléatoires indépendantes avec $Z_i \rightsquigarrow \mathcal{N}(0, 1), i = 1, 2$:

$$f_{z_1}(z_1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z_1^2\right), \quad f_{z_2}(z_2) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z_2^2\right)$$

Leur densité conjointe est donnée par :

$$f(z_1, z_2) = \frac{1}{2\pi} \exp\left(-\frac{1}{2}(z_1^2 + z_2^2)\right).$$

Dans le plan \mathbb{R}^2 , le point (z_1, z_2) est situé à une distance $r = \sqrt{z_1^2 + z_2^2}$ de l'origine, et la droite passant par $(0; 0)$ et (z_1, z_2) forme un angle avec l'axe horizontal, tel que $\tan \theta = \frac{z_2}{z_1}$, le passage des coordonnées cartésiennes $(z_1; z_2)$ aux coordonnées polaires s'effectue par la transformation :

$$z_1 = r \cos \theta, \quad z_2 = r \sin \theta$$

dont le jacobien J est égal à :

$$J = \begin{vmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{vmatrix} = r.$$

Ainsi :

$$f(r, \theta) = |r| f(z_1, z_2) = r \cdot \frac{1}{2\pi} \exp\left(-\frac{1}{2}(z_1^2 + z_2^2)\right) = \frac{1}{2\pi} \cdot r \exp\left(-\frac{r^2}{2}\right).$$

Cela montre que les variables aléatoires $R = \sqrt{Z_1^2 + Z_2^2}$ et Θ (la variable aléatoire définie à partir de Z_1 et Z_2 comme la fonction qui associe à Z_1 et Z_2 la seule valeur dans $[0, 2[$ dont $\cos \Theta$ a la même signe que Z_1 ; $\sin \Theta$ a la même signe que Z_2 et $Z_1 \sin \Theta = Z_2 \cos \Theta$) sont des variables indépendantes dont les densités respectives sont :

$$\begin{aligned} f_R(r) &= r \exp\left(-\frac{r^2}{2}\right), \text{ pour } -\infty < r < +\infty \\ f_\Theta(\theta) &= \frac{1}{2\pi} \text{ pour } 0 \leq \theta < 2\pi. \end{aligned}$$

En posant $U = \exp(-R^2/2)$, on a, pour $u > 0$:

$$\begin{aligned} \Pr(U \leq u) &= \Pr(\exp(-R^2/2) \leq u) = \Pr(-R^2/2 \leq \log u) \\ &= \Pr\left(R \geq \sqrt{-2 \log u}\right) = \int_{\sqrt{-2 \log u}}^{\infty} r \exp\left(-\frac{r^2}{2}\right) dr \\ &= u. \end{aligned}$$

On a ainsi transformé (Z_1, Z_2) , deux variables aléatoires normales indépendantes, en une paire de variables uniformes indépendantes. ■

Les transformations inverses permettent donc de simuler deux échantillons indépendants d'une loi normale à partir de deux échantillons indépendants issus d'une loi uniforme.

Remarque 2.2 *Dans le cas de la simulation d'une loi normale, la méthode de l'inversion est d'application difficile, car l'inverse de ne s'explique pas par des fonctions élémentaires, et la fonction de répartition de la loi normale centrée réduite s'exprime seulement à travers des fonctions élémentaires.*

2.3.1 Algorithme de la méthode Box-Muller

- Soit V, U deux v.a. indépendantes uniformes sur $[0; 1]$
- $\Theta = 2\pi U$ (suit une loi uniforme sur $[0, 2\pi]$).
- $R = \sqrt{-2 \log V}$, d'où :

$$X = \sqrt{-2 \log(V)} \cos(2\pi U),$$

$$Y = \sqrt{-2 \log(V)} \sin(2\pi U),$$

est un couple gaussiennes centrées réduites indépendantes.

Le code R suivant permet de générer 100 observations d'une variable aléatoire gaussienne centrée réduite en utilisant la méthode de Box-Muller :

```
N=100
X<-numeric(N)
for(i in 1:N){U<-runif(1)
V<-runif(1)
X[i]=sqrt(-2*log(v))*cos(2*pi*U)}
X
qqnorm(X,xlab="Q.Nom",ylab="Q.Theo",main="QQ-plot")
qqline(X)
```

2.4 Simulation par la méthode de rejet

La méthode de rejet peut s'utiliser pour la génération de n'importe quel type de variable aléatoire. Elle consiste à générer des données suivant une distribution de fonction de densité proche de celle désirée et ensuite éliminer une certaine proportion de ces données de manière à se ramener à des données qui suivent la distribution attendue.

La méthode de rejet est utilisée pour engendrer indirectement une variable aléatoire X , de densité de probabilité f lorsqu'on ne sait pas simuler directement la loi de densité de probabilité f .

Soit (Y, U) un couple de variables aléatoires indépendantes tirées selon une loi uniforme, i.e. (Y, U) est un point tiré uniformément dans le carré unité. On peut alors montrer que la distribution de X est la loi conditionnelle de X sachant l'événement : $M = \{U \leq f_X(Y)\}$.

Autrement dit, $f_X(x) = f_Y(x/M)$.

Pour simuler une suite de variables aléatoires réelles $(X_n)_{n \geq 1}$ de distribution identique à celle de X , il suffit donc, dans une suite de tirages de couples (Y_i, U_i) uniformes indépendants, de sélectionner les Y_i correspondant aux tirages vérifiant (Y_i, U_i) et de rejeter les autres.

2.4.1 Algorithme

On voudrait simuler une variable aléatoire réelle X de densité de probabilité f . On suppose qu'il existe une autre densité de probabilité g telle que le ratio $\frac{f}{g}$ soit borné, disons par c (i.e. $f \leq cg$), et qu'on sache simuler Y de densité g .

La version basique de la méthode de rejet prend la forme suivante :

1. Tirer Y de densité g ,
2. Tirer U selon la loi uniforme $U(0; 1)$, indépendamment de Y ,
3. Tant que $U > \frac{f(Y)}{cg(Y)}$, reprendre en 1,
4. Accepter Y comme un tirage aléatoire de densité de probabilité f .

Remarque 2.3 *On remarque que l'algorithme comporte une boucle dont la condition porte sur des variables aléatoires. Le nombre d'itérations, notons-le N est donc lui-même aléatoire. On peut montrer que N suit la loi géométrique de paramètre $\frac{1}{c}$.*

Par la suite, l'espérance de N , c-à-d le nombre moyen d'itérations à effectuer avant d'obtenir une réalisation de la densité f vaut c . On a donc tout intérêt à choisir c le plus petit possible. En pratique, une fois la fonction g choisie, le meilleur choix de c est donc la plus petite constante qui majore le ratio $\frac{f}{g}$, c'est-à-dire :

$$c = \sup \frac{f(x)}{g(x)}.$$

Notons que, soit c est supérieur strict à 1, soit $f \neq g$, la deuxième alternative étant assez théorique : en effet, comme $cg - f \geq 0$. On a donc intérêt à

choisir c le plus proche de 1 possible, pour que le nombre d'itérations moyen soit proche de 1 lui aussi. De plus, le choix de g est primordial :

- le tirage de la loi g doit être facile ;
- l'évaluation de $f(x)/g(x)$ doit être aisée ;
- la constante c doit être la plus petite possible ;
- la fonction cg doit majorer la densité f .

2.4.2 Simulation d'une variable normale

Pour simuler une variable aléatoire normale centrée réduite X par la méthode du rejet, on choisira $g(y) = e^{-y}$ pour $y > 0$ et $c = \sqrt{\frac{e}{2\pi}} \simeq 0.6577$.

A noter qu'ici $c < 1$. Puisque f , la fonction de densité de X , est symétrique, on simulera seulement des valeurs positives d'une distribution normale, pour ensuite les étaler de manière aléatoire de part et d'autre de l'axe de symétrie. Cette variante de la méthode entraîne seulement $c > 1/2$.

Algorithme :

1. Générer U selon la loi uniforme $\mathcal{U}(0, 1)$,
2. Générer Y de densité g ,
3. Tester si $U \leq f(Y)/cg(Y)$ c'est-à-dire si :

$$U \leq \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} / \sqrt{\frac{e}{2\pi}} * e^{-y}$$

4. Si cette condition n'est pas vérifiée retourner à l'étape 1, sinon continuer,
5. Si $U < 0.5$ mettre un signe positif à Y , sinon le garder négatif.

Une simulation complète est détaillée dans l'exemple 2.3.

Exemple 2.3 Soit $f(x)$ la densité d'une distribution normale centrée réduite, et $g(x) = e^{-x}$ pour $x > 0$. Une application de la méthode du rejet est utilisée pour simuler des réalisations de la loi normale en accord avec la procédure décrite plus haut.

2.4.3 Calcul d'intégrale par la méthode de rejet

L'estimation d'une intégrale peut également se faire en utilisant les méthodes de simulation de loi uniforme dans certaines parties de \mathbb{R}^2 . Afin de simplifier les notations, on ne supposera ici que $g(x) \geq 0$. Il est connu que l'intégrale d'une fonction est l'aire sous la courbe de celle-ci. Estimer l'intégrale

$I = \int g(x)f(x)dx$ revient donc à estimer l'aire de la partie $D = \{(x, u) \in \mathbb{R}^2, 0 \leq u \leq g(x)f(x)\}$ de \mathbb{R}^2 :

$$I = \int_{\mathbb{R}} g(x)f(x)dx = \lambda(D).$$

Afin d'estimer cette aire, on peut procéder de la façon suivante. Choisir un domaine D' contenant D , tirer N points uniformément dans D' et calculer la proportion f_n d'entre eux qui tombe dans D : cette quantité estime $I/\lambda(D')$. Finalement, on estime donc I par $\lambda(D') \cdot f_n$.

Choix de la constante c

La constante c à utiliser doit être telle que l'exponentielle soit toujours plus grande que la courbe normale mais la plus proche possible (jusqu'à la toucher) pour maximiser l'efficacité des tirages.

Pour que la courbe $cg(x)$ soit toujours plus grande que $f(x)$ il faut, dans le cas présent que :

$$c \times e^{-x} \geq \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \forall x \geq 0$$

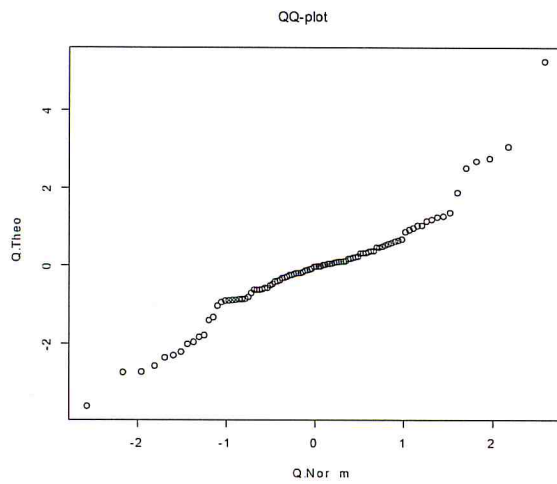
soit

$$c \geq \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2-2x}{2}} = \frac{\sqrt{e}}{\sqrt{2\pi}} e^{-\frac{x^2-2x+1}{2}} = \frac{\sqrt{e}}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}}, \forall x \geq 0$$

Le terme de droite de cette expression prend sa valeur maximum en $x = 1$ et vaut alors $\frac{\sqrt{e}}{\sqrt{2\pi}}$ la valeur minimum à donner à c .

Code R pour la méthode de rejet

```
# Simulation de loi normale centree reduite
N=100
repeat{u<-runif(N)
Y<-rexp(N)
I=function(y){(1/sqrt(2*pi))*exp(-y^2/2)}
J=function(y){exp(-y)}
c<-sqrt(exp(1)/2*pi)
if(u<=I(Y)/c*J(Y))
{break}
}
X=numeric(N)
for(i in 1:N){
X[i]=ifelse(u[i]<0.5,Y[i],-Y[i])}
qqnorm(X,xlab="Q.Norm",ylab="Q.Theo",main="QQ-plot")
```



Chapitre 3

Méthode de Monté Carlo (MC)

3.1 Introduction

Le terme méthode de Monte-Carlo désigne toute méthode visant à calculer une valeur numérique en utilisant des procédés aléatoires, c'est à dire des techniques probabilistes. Le nom de cette méthode fait allusion aux jeux de hasard pratiqués à Monte-Carlo. Il n'y a pas de définition précise de ce qu'est une technique de type Monte-Carlo, mais la description la plus habituelle consiste à dire que les méthodes de ce type se caractérisent par l'utilisation du hasard pour résoudre des problèmes centrés sur le calcul d'une valeur numérique. La réponse fournie sera une réponse statistique.

La méthode de Monte Carlo a été préférée dans de très nombreux secteurs scientifiques et technologiques. La puissance développée des ordinateurs a permis à ces méthodes de devenir opérationnelles et de s'étendre dans des domaines différents tels que la finance, les mathématiques, la physique, la biologie moléculaire et génétique, les télécommunications, les réseaux, la recherche opérationnelle et bien d'autres encore. D'une façon générale, l'utilisation de cette méthode recouvre tous les domaines où l'utilisation des méthodes scientifiques se heurte à des difficultés. Dans ce contexte, on distingue deux grands domaines où la méthode de Monté Carlo peut être utilisée avec succès :

Problème déterministes : Ce sont des problèmes de nature déterministe faisant appel aux calculs numériques. On cite comme des exemples à ces problèmes :

- Estimation des surfaces.
- Calculs des intégrales multiples.
- Résolution des équations différentielles.
- Résolution des systèmes des équations algébriques.

- Résolution des problèmes d'optimisation combinatoire.

Phénomènes et processus aléatoires : on cite comme des exemples à ces problèmes :

- Mouvement des particules.
- Systèmes stochastiques de la gestion ou de la production.
- Reconnaissance de formes (analyse d'images, de paroles, ...).
- Systèmes de commande décrits par des équations différentielles ordinaires ou des équations aux différences.

3.2 Histoire de MC

Bien qu'il y ait eu auparavant une utilisation isolée de la technique, telle qu'une estimation de π en jetant des aiguilles sur une surface plane en bois avec des lignes parallèles équidistantes Asaph, H., 1873 est une pierre angulaire de l'histoire de la méthode de Monte Carlo.

Au cours du projet Manhattan, le CMC a été utilisé sous sa forme moderne pour simuler la diffusion aléatoire de neutrons dans une matière fissile. Comme tout le projet était secret, la technique portait le nom de code «Monte Carlo» d'après le casino de Monte Carlo, à Monaco, où un parent du scientifique dépenserait l'argent qu'il avait emprunté à sa famille. Cette technique a été bénéfique à la naissance de l'analyseur d'intégrateur numérique électronique et de l'ordinateur a.k.a. ENIAC, le premier ordinateur électronique à exécuter le calcul détaillé par ailleurs impossible.

Cette nouvelle praticabilité autorisée a suscité un vif intérêt pour la technique, qui s'est développée parallèlement au boom de l'informatique, et est devenue, l'un des dix algorithmes les plus utilisés au XXe siècle, principalement pour ses caractéristiques techniques très abordables.

3.3 Principe de la méthode de Monté Carlo

L'une des procédures pour calculer une quantité par la méthode de Monte-Carlo est de la mettre tout d'abord sous la forme d'une espérance, à l'issue de cette étape, il reste à calculer cette quantité par une espérance $\mathbb{E}(X)$ de la variable aléatoire X . Pour ce calcul, il convient de savoir simuler une variable aléatoire selon la loi de X . On dispose alors d'une suite $(X_i)_{1 \leq i \leq N}$ de N réalisations de la variable aléatoire X . On approxime alors $\mathbb{E}(X)$ par :

$$\mathbb{E}(X) \approx \frac{1}{N}(X_1 + \dots + X_N)$$

3.3.1 Calcul d'intégrale par la méthode de Monte Carlo

Intégration unidimensionnelle

Les méthodes de Monte Carlo reposent sur une approximation probabiliste et non déterministe. En ce sens, on ne résout pas l'objet mathématique mais on cherche à l'approcher moyennant la loi forte des grands nombres. Cet objet peut être une intégrale comme c'est le cas dans le présent paragraphe. Ici, nous traiterons les intégrales se présentant sous la forme :

$$I = \int_a^b f(x) dx$$

où : f une fonction intégrable sur $[a, b]$.

La méthode de Monte Carlo pour l'intégration consiste à trouver une variable aléatoire Z telle que : $I = \mathbb{E}(Z)$. Ce qui permet d'estimer I en utilisant la loi forte des grands nombres. L'erreur commise est contrôlée par le théorème central limite dès que Z est de carré intégrable.

Si aucune primitive de $f(x)$ n'est connue, l'intégrale ne peut pas être calculée analytiquement. Mais si $f(x)$ peut être facilement calculée en tout point de l'intervalle $[a, b]$, on peut obtenir une bonne approximation de la valeur de cette intégrale par des méthodes numériques.

Il existe de nombreuses méthodes d'intégration numérique. La plus simple consiste à diviser l'intervalle par N rectangles adjacents. La hauteur de chaque rectangle est égale à la valeur de $f(x)$ pour x pris au milieu de la base du rectangle. La somme des aires de ces rectangles est une approximation de l'aire sous la courbe représentant $f(x)$, c'est à dire l'intégrale I recherchée :

$$I \simeq \sum_{t=1}^N h f(x_t) = h \sum_{t=1}^N f(x_t) = \frac{b-a}{N} \sum_{t=1}^N f(x_t).$$

Si $f(x_i)$ a un comportement suffisamment régulier, et si les rectangles sont suffisamment étroits, alors l'aire ainsi calculée sera une bonne approximation de la valeur de l'intégrale.

Proposition 3.1 *On écrit :*

$$I = \int_a^b \frac{g(x)}{f_X(x)} f_X(x) dx = \mathbb{E} \left(\frac{g(X)}{f_X(X)} \right)$$

où X est une variable aléatoire de densité f_X et on note que $\frac{g(x)}{f_X(x)} = Z$ est bien définie car $f_X(X) \neq 0$ presque sûrement. Par ailleurs Z est dans

$\mathbb{L}^1(a, b)$, donc, d'après la loi forte des grands nombres :

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N \frac{g(X_i)}{f_X(X_i)} \xrightarrow{p.s} I \text{ quand } N \rightarrow \infty.$$

\hat{I}_N est un estimateur sans biais de I si les X_i sont indépendantes de densité f_X , on a

$$\frac{\hat{I}_N - I}{\sigma} \xrightarrow{D} \mathcal{N}(0, 1), \text{ quand } N \rightarrow \infty$$

où $\mathcal{N}(0, 1)$ est la loi normale centrée réduite et \xrightarrow{D} désigne la convergence en loi à condition $\sigma^2 < \infty$ est la variance de Z .

$$\lim_{N \rightarrow \infty} P \left(\hat{I}_N - z_\alpha \frac{\hat{\sigma}_N}{\sqrt{N}} \leq I \leq \hat{I}_N + z_\alpha \frac{\hat{\sigma}_N}{\sqrt{N}} \right) = 1 - 2\alpha$$

$\hat{\sigma}_N$ est un estimateur sans biais de la variance de Z :

$$\hat{\sigma}_N^2 = \frac{1}{N-1} \sum_{i=1}^N \left(\frac{g(X_i)}{f_X(X_i)} - \hat{I}_N \right)^2.$$

Exemple 3.1 Calcul d'intégral unidimensionnel par la méthode de Monte Carlo. Pour calculer l'intégral :

$$I = \int_1^{10} \exp(-x + 2) dx$$

On note

$$\psi(x) = \exp(-x + 2)$$

Pour l'écriture de la fonction, on utilise le syntax "function" du langage R comme suit :

le nom de la fonction <- fonction(liste des arguments) {le corps de la fonction}

1) **Le code de R ci-dessous :**

L'écriture de la fonction : $\psi(x)$

fo=function(x){exp(-x+2)} # Est ce qu'elle fonctionne?

fo(0); fo(3); fo(pi/2)

Il faut charger le package qui contient le syntaxe utilisé et ça par deux méthodes. La 1ère : on clique sur "**packages**" ensuite "**charger package**"

puis sélectionne le nom de package puis clique sur "ok". La 2ème : est plus simple, on écrit dans le console R : *Library (le nom de package)*.

– Utilisation de syntaxe "integrate" : Il faut charger le package "stats" pour utiliser le

syntaxe "integrate". On utilise le syntaxe "integrate" du langage R sous forme :

le nom d'intégrale <- integrate(le nom de la fonction, la borne inférieure d'intégrale, la borne supérieure d'intégrale)

Le programme R

Simulation par méthode de Monte-Carlo

Valeur théorique de l'intégrale :

$$\int_1^{10} e^{2-x} dx$$

```
library(stats) # charger le package "stats"
```

```
fo=function (x){exp(-x + 2)}
```

```
fo2=function (x){(exp(-x + 2))^2}
```

```
I=integrate(fo,1,10)$v
```

```
I
```

```
[1] 2.717946
```

```
## Integration par simulation de Monte-Carlo
```

```
N = 10000
```

```
u=runif(N,1,10)
```

```
x=seq(1 : N)
```

```
fu=fo(u)
```

```
In = 9*mean(fu)
```

```
In = 2.719585
```

```
## Loi Forte des Grands Nombres
```

```
par(bty="l") #pour eliminer les axes haut et droite
```

```
plot(1 : N,9*cumsum(fu)/1 : N,type="l",main="L.F.G.Nbrs",xlab="x")
```

```
abline(h=I,col="red")
```

3.3.2 Vers l'évaluation de l'erreur

Le but de cette partie est de commencer à étudier l'erreur commise par une méthode de Monte Carlo. Pour cela, on considère la quantité $err_N = I_N - I$, d'après l'exemple précédent : `plot(1 : N, I - 9*cumsum(fu)/1 : N, type="l")`

Remarque 3.1 *La syntaxe "integrate" qui donne la valeur de I avec l'erreur pour ça on ajoute \$v (obtenue sa valeur sans erreur).*

d'après ce qu'on a vu on remplit ce tableau selon l'exemple précédent :

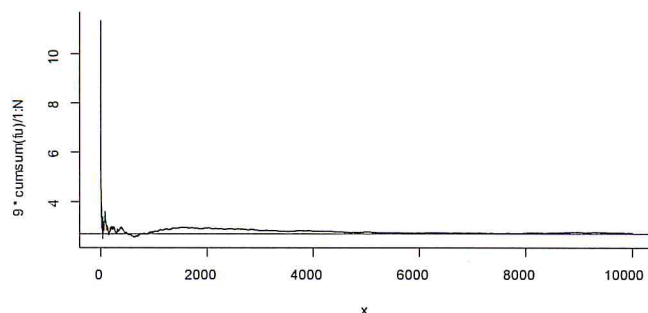


FIG. 3.1 – Convergence de l'intégrale simulé par la méthode de Monte-Carlo.

Intégrale multiple

Que se passe-t-il si nous utilisons le même approche dans le cas d'une intégrale multiple ?

Nous rencontrons deux difficultés :

1. D'abord, la région d'intégration n'est plus définie par une paire de nombres comme précédemment, mais pas une hyper-surface fermée dont la forme peut être très compliquée même pour des problèmes simples, et impossible à décrire analytiquement.
2. La deuxième difficulté est encore plus grave. Revenons un instant à l'intégrale simple, et supposons que nous ayons décidé d'utiliser 500 rectangles. Puis envisageons une intégrale multiple à 500 variables, et décidons de conserver sur chaque axe la même résolution que dans le cas de l'intégrale simple. Nous devons alors définir 500×500 hyper-rectangles, un nombre au-delà des capacités des ordinateurs les plus rapides.

Cette difficulté est absolument universelle, et se retrouve dans toute technique locale, qu'elle soit déterministe ou probabiliste. Notez que le nombre de rectangles est choisi essentiellement sur la base considérations portant sur la rapidité avec laquelle la fonction varie dans la région d'intégration. Il n'est donc pas possible de réduire arbitrairement le nombre de rectangles jusqu'à une valeur compatible avec des temps de calcul raisonnables, sous peine de perdre tellement d'information sur la fonction que l'approximation obtenue devienne grossièrement fausse.

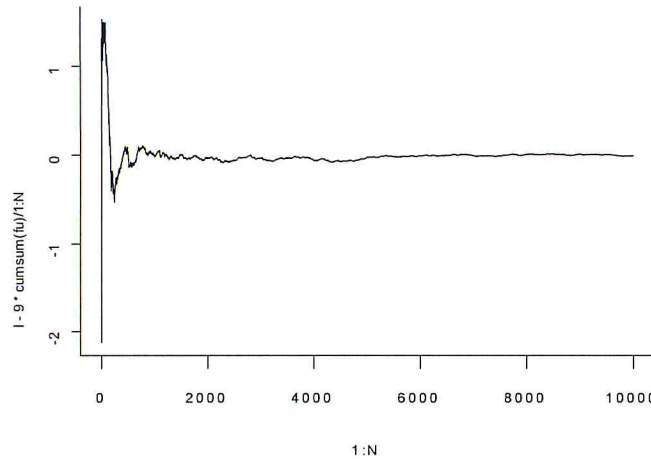


FIG. 3.2 – Convergence de la valeur de l'intégrale

3.3.3 Intégration par simulation de Monte-Carlo

C'est alors qu'intervient une idée à la fois simple et efficace. Nous avons remarqué qu'une intégrale n'est, à peu de chose près, que la somme des valeurs de la fonctions prises sur des points régulièrement répartis dans la région d'intégration. Dans cette phrase, remplaçons "**régulièrement répartis**" par "**répartis selon une distribution de probabilité uniforme**", et nous avons notre première simulation de Monte-Carlo.

Donc, dans sa version la plus simple, le calcul d'une intégrale sous la forme :

$$I = \iiint \dots \int_{\text{région}} f(x_1, \dots, x_n) dx_1 \dots dx_n$$

par simulation de Monte-Carlo procède ainsi :

1. Se servir d'une distribution uniforme à n dimensions pour tirer N points dans la région d'intégration (dont nous notons le volume V).
2. Additionner tous les $f(p_i)$, multiplier le résultat par V/N .
3. Le résultat est une estimation de la valeur de l'intégrale I :

$$\hat{I} = \frac{V}{N} \sum_{i=1}^N f(p_i).$$

Remarque 3.2 *L'intérêt principal de l'intégration par simulation de Monte-Carlo réside dans son comportement en grande dimension. Cette question est difficile mais la réponse est simple. Considérons une méthode quelconque d'intégration numérique déterministe reposant sur le calcul de valeurs de la fonction sur les nœuds d'une grille dans la région d'intégration. Si nous comparons les performances de cette méthode et de celle de la méthode de Monte-Carlo pour des dimensions de plus en plus grandes, il se trouvera une dimension d au-delà de laquelle la méthode de Monte-Carlo sera plus efficace que la méthode déterministe pour un nombre N donné de tirages. Ceci veut-dire que les valeurs de l'intégrale trouvées par la méthode de Monte-Carlo seront, le plus souvent, plus proches de la valeur vraie de l'intégrale que celle trouvée par l'approximation déterministe.*

Exemple 3.2 *calcul d'intégral multiple par la méthode de Monte Carlo.*

Pour calculer l'intégrale

$$I = \int_0^1 \int_0^1 \int_0^1 (x_1 + x_2 + x_3) dx_1 dx_2 dx_3$$

sa valeur théorique est : $\frac{3}{2} = 1.5$

On note

$$f(x) = (x_1 + x_2 + x_3) = \sum_{i=1}^3 x_i$$

Pour cela :

1. générer x_1 à x_3 (dans l'intervalle d'intégration),
2. calculer $f(x_1; x_2; x_3)$ et en déduire \hat{I} ,
3. réitérer N fois ($N = 2, 4, 8, \dots, 8192$) les étapes 1 et 2, calculer $\langle \hat{I}_N \rangle$: la moyenne des N évaluations de \hat{I} .

le code de R

```
N=
for(i in 1:N)
{x=runif(3,0,1)
```

```

f[i]=sum(x)
S=sum(f)
I=S/N #estimation de I

\begin{equation*}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
$N$
& $100$
& $1000$
& $10000$
& $10500$
& $15000$
\\ \hline
$\hat{I}$
& $1.485416$
& $1.496255$
& $1.496663$
& $1.498879$
& $1.500948$
\\ \hline
\end{tabular}%
\end{equation*}

```

on remarque que à chaque fois on augmente le N on approche à la valeur théorique.

3.3.4 Calcul d'aire (Approximation de π)

La Méthode de Monte-Carlo permet la résolution de certains problèmes numériques déterministes. On résout les problèmes de façon approchée avec une simulation.

Dans ce paragraphe nous présentons brièvement l'approximation du nombre π par l'estimation de la surface d'un quart de disque de forme circulaire de rayon 1. Dans ce cas, on tire au hasard des coordonnées x et y , chacune dans l'intervalle $[0, 1[$. Si $x^2 + y^2 < 1$ alors le point P de coordonnées (x, y) appartient au quart de disque D de centre $(0, 0)$ et de rayon 1. La probabilité que P appartienne à D est $\frac{\pi}{4}$ (rapport de l'aire de D et du carré l'englobant). Donc, si on tire au hasard n points, et si p d'entre eux appartiennent à D , on s'attend à avoir : $\frac{p}{n} \approx \frac{\pi}{4}$. On en tire donc une approximation de π égal à $\frac{4p}{n}$.

Le code de R

```
n=  
x=runif(n)  
y=runif(n)  
Z=x^2+y^2  
T=Z[Z<=1]  
p=length(T)  
k=p/n  
pi=4*k
```

d'après ce code on voit les résultats suivantes :

n	100	1000	10000	100000	1000000
$\hat{\pi}$	3.24	3.156	3.1616	3.13588	3.142728

On remarque que à chaque fois on augmente le N l'approximation de π approche de la valeur théorique.

3.4 Techniques de réduction de la variance

3.4.1 Introduction

Dans cette partie, il est très important de rappeler que lors d'une estimation d'un paramètre, un bon estimateur est celui qui rend faible à la fois le biais au carré et la variance.

Plusieurs méthodes de Monte Carlo ont en commun le fait de ne pas introduire de biais dans l'estimation. Ainsi, lorsque le biais est nul, on peut se donner comme objectif de réduire au maximum la variance lors de l'estimation des paramètres.

Objectif : réduire la variance de l'estimateur de Monte Carlo classique :

$$\mathbb{E} \left[\left(\hat{I}_N - I \right)^2 \right] = \frac{1}{N} \text{Var} (f(X))$$

Les méthodes :

- Échantillonnage préférentiel
- Variables de contrôle
- Variables antithétiques
- Stratification
- Entropie croisée

3.4.2 Echantillonnage préférentiel (importance sampling)

Observation : la représentation de I comme espérance n'est pas unique. Si X est à densité p :

$$I = \mathbb{E}_p[f(X)] = \int f(x)p(x)dx = \int \frac{f(x)p(x)}{q(x)}q(x)dx = \mathbb{E}_q \left[\frac{f(X)P(X)}{q(X)} \right]$$

L'utilisateur est libre dans le choix de la densité q .

- **Idée** : Dans la situation où on sait que $f(X)$ est surtout sensible à certaines valeurs de X , au lieu de tirer les $X(k)$ selon la densité originale $p(x)$ de X , on les tire selon une densité "biaisée" $q(x)$ qui favorise les valeurs de X dans la zone d'importance.
- En considérant la représentation

$$I = \mathbb{E}_p[f(X)] = \mathbb{E}_q \left[f(X) \frac{p(X)}{q(X)} \right]$$

On propose l'estimateur :

$$\hat{I}_N = \frac{1}{N} \sum_{k=1}^N f(X^{(k)}) \frac{p(X^{(k)})}{q(X^{(k)})}$$

où $(X^{(k)})_{k=1, \dots, N}$ est un N -échantillon tiré selon

- Estimateur non-biaisé : $\mathbb{E}_q[\hat{I}_N] = I$
- Variance de l'estimateur :

$$Var(\hat{I}_N) = \frac{1}{N} Var_q \left(f(X) \frac{p(X)}{q(X)} \right) = \frac{1}{N} \left(\mathbb{E}_q \left[f(X)^2 \frac{p(X)}{q(X)} \right] - \mathbb{E}_q [f(X)]^2 \right)$$

En choisissant bien q , on peut fortement réduire la variance. En fait, en choisissant (en supposant $f \geq 0$)

$$q_{opt}(x) = \frac{f(x)p(x)}{I}$$

on trouve

$$Var(\hat{I}_N) = 0$$

Mais $q_{opt}(x)$ dépend de I ! (\leftrightarrow méthodes séquentielles / adaptatives).

- Points pratiques importants pour pouvoir implémenter la méthode : Il faut savoir simuler X de loi q et il faut savoir calculer le rapport de vraisemblance $\frac{p(x)}{q(x)}$.

Exemple 3.3 On veut estimer

$$I = \mathbb{E}[f(X)]$$

avec $X \rightsquigarrow \mathcal{N}(0, 1)$ et $f(x) = [x - 3]^+ = \begin{cases} x-3 & \text{si } x \geq 3 \\ 0 & \text{sinon} \end{cases}$

$$I = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} (x-3)^+ e^{-\frac{x^2}{2}} dx = \frac{1}{\sqrt{2\pi}} \int_3^{\infty} (x-3) e^{-\frac{x^2}{2}} dx \simeq 3.82 \times 10^{-4}$$

Monte Carlo :

$$\hat{I}_N = \frac{1}{N} \sum_{k=1}^N (X_k - 3)^+, X_k \rightsquigarrow \mathcal{N}(0, 1)$$

on a $Var(\hat{I}_N) = \frac{1}{N} 2.03 \cdot 10^{-4}$

Code R

```
N=
x=rnorm(N,0,1)
S=sum(x-3)
I=S/N
I
var=2.03*10^(-4)/N
```

N	100	1000	10000	100000
\hat{I}_N	3.108827×10^{-4}	3.018107×10^{-4}	3.001175×10^{-4}	3.002061×10^{-4}
$Var(\hat{I}_N)$	$2.03 \cdot 10^{-6}$	$2.03 \cdot 10^{-7}$	$2.03 \cdot 10^{-8}$	$2.03 \cdot 10^{-9}$

Table : approximation de I

Echantillonnage préférentiel : On tire $X(k)$ selon la loi $\mathcal{N}(3, 1)$.

$$\hat{I}_N = \frac{1}{N} \sum_{k=1}^N (X - 3)^+ \frac{e^{-\frac{X^2}{2}}}{e^{-\frac{(X-3)^2}{2}}} = \frac{1}{N} \sum_{k=1}^N (X - 3)^+ e^{-3X_k + \frac{9}{2}}, \quad X_k \rightsquigarrow \mathcal{N}(3, 1)$$

on a $Var(\hat{I}_N) = \frac{1}{N} 2.47 \times 10^{-7}$.

Il faut 1000 fois moins de simulations avec la méthode *IS* pour atteindre la même précision !

Exemple 3.4 : On veut estimer

$$I = \mathbb{E}[f(X)]$$

avec $X \rightsquigarrow \mathcal{N}(0, 1)$ et $f(x) = \exp(x)$.

$$I = \frac{1}{\sqrt{2\pi}} \int e^x e^{-\frac{x^2}{2}} dx = e^{\frac{1}{2}}$$

Ce sont les grandes valeurs de X qui sont importantes.

Echantillonnage préférentiel : on tire $X^{(k)}$ selon la loi $\mathcal{N}(\mu, 1)$, $\mu > 0$.

$$\hat{I}_N = \frac{1}{N} \sum_{k=1}^N f(X^{(k)}) \frac{e^{-\frac{[X^{(k)}]^2}{2}}}{e^{-\frac{[X^{(k)}-\mu]^2}{2}}} = \frac{1}{N} \sum_{k=1}^N f(X^{(k)}) e^{-\mu X^{(k)} + \frac{\mu^2}{2}}$$

$$\text{Var}(\hat{I}_N) = \frac{1}{N} (e^{\mu^2 - 2\mu + 2} - e^1)$$

Monte Carlo : $\mu = 0$ et $N = 10000$: $\text{Var}(\hat{I}_{10000}) = \frac{1}{10000} (e^2 - e^1) = 4.6708 \times 10^{-4}$.

Echantillonnage préférentiel optimal : $\mu = 1$ et $\forall N \in \mathbb{N} : \text{Var}(\hat{I}_N) = 0$.

Exemple 3.5 (Le jeu de pile ou face) Question : sur une suite de 100 lancers de pile ou face, quelle est la probabilité B que plus de 70 lancers donnent pile ?

Réponse : On note S le nombre de piles obtenus. La probabilité d'obtenir k piles est

$$P(S = k) = p_S(k) = C_k^{100} \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{100-k} = C_k^{100} 2^{-100}$$

On cherche

$$B = P(S > 70)$$

Donc

$$B = P(S > 70) = \sum_{k=71}^{100} p_S(k) \simeq 1.6 \times 10^{-5}.$$

Imaginons maintenant qu'on ne sache pas faire le calcul.

Méthode de Monte Carlo.

On réalise N expériences indépendantes.

Expérience i : $\begin{cases} \text{On lance 100 pièces} \\ \text{On note } S^{(i)} \text{ le nombre de piles obtenus} \end{cases}$

On compte le nombre de fois où on a eu plus de 70 piles, et on divise par N

$$B'_N = \frac{1}{N} \sum_{i=1}^N 1_{S^{(i)} > 70}$$

On a ainsi un estimateur non-biaisé et convergent de B . Mais :

$$\text{Var}[B'_N] = \mathbb{E}[(B'_N - B)^2] = \frac{1}{N}(B - B^2)$$

L'erreur relative est donc :

$$\text{Erreur} = \frac{\sqrt{\text{Var}[S'_N]}}{S} = \frac{1}{\sqrt{N}} \left(\frac{1}{S} - 1 \right)^{1/2} \simeq \frac{1}{\sqrt{N}\sqrt{S}}$$

Il faut $N \geq 100S^{-1}$ pour avoir une erreur relative de moins de 10%.

Echantillonnage préférentiel : on utilise une pièce biaisée, dont la probabilité de faire pile est $q > 1/2$. Alors la probabilité de faire k piles est :

$$P^{IS}(S = k) = p_S^{IS}(k) = \binom{100}{k} q^k (1 - q)^{100 - k}$$

On réalise n expériences indépendantes, pendant lesquelles on lance 100 pièces biaisées. On note $S^{(i)}$ le nombre de piles obtenus lors de l'expérience i .

Estimateur non-biaisé convergent :

$$\begin{aligned} B'_N &= \frac{1}{N} \sum_{i=1}^N 1_{S^{(i)} > 70} \frac{p_S(S^{(i)})}{p_S^{IS}(S^{(i)})} \\ &= \frac{1}{N} \sum_{i=1}^N 1_{S^{(i)} > 70} \left(\frac{1}{2q} \right)^{S^{(i)}} \left(\frac{1}{2(1-q)} \right)^{100 - S^{(i)}} \end{aligned}$$

Remarque 3.3 *Un rapport de vraisemblance peut être plus facile à calculer qu'une vraisemblance. On connaît tout ici, on peut donc tout calculer et vérifier analytiquement.*

$$\begin{aligned} \text{Var}_{IS}[B'] &= \frac{1}{N} \left\{ \mathbb{E}_{IS} \left[1_{S^{(i)} > 70} \frac{p(S^{(i)})^2}{p^{IS}(S^{(i)})^2} \right] - B^2 \right\} \\ &= \frac{1}{N} \left\{ \sum_{k=0}^{100} 1_{k > 70} \binom{100}{k} \left(\frac{1}{4q} \right)^k \left(\frac{1}{4(1-q)} \right)^{100-k} - B'^2 \right\} \end{aligned}$$

La variance de l'estimateur dépend explicitement du choix de q !

$$Erreur_{IS}[B'_N] = \frac{\sqrt{Var[B']}}{B} = \frac{1}{\sqrt{N}} \times \begin{cases} 249 & \text{si } q = 0.5 \\ 8.86 & \text{si } q = 0.6 \\ 2.06 & \text{si } q = 0.7 \\ 6.78 & \text{si } q = 0.8 \end{cases}$$

En choisissant bien q , l'erreur peut être fortement réduite.

- Il faut pousser q , pour que l'événement "plus de 70 piles" devienne de probabilité d'ordre 1.
- Il ne faut pas trop pousser, car sinon les rapports de vraisemblance deviennent grands et amplifient les erreurs.

Remarque 3.4 Si on choisit

$$q_{opt}(x) = \frac{f(x)p(x)}{I}$$

alors :

$$Var(\hat{I}) = 0.$$

Différentes méthodes (séquentielles, adaptatives, ...) existent pour "trouver" la densité $q_{opt}(x)$:

- Analyse théorique (style grandes déviations).
- Utilisation d'un modèle réduit

$$f_r(x) \rightarrow q_{opt}(x) = \frac{f_r(x)p(x)}{I_r}, \text{ avec } I_r = \int f_r(x)p(x)dx$$

- Méthodes paramétriques : choisir une famille de lois biaisées, estimer ses paramètres à partir des premières simulations, puis tirer les dernières simulations pour approcher la loi estimée.
- Méthodes non-paramétriques : idem mais par des méthodes d'estimation à noyaux ou autres.
 - **Contraintes** : il faut savoir simuler q et calculer les rapports de vraisemblance.
 - **Pas facile** : il faut trouver comment biaiser l'entrée X pour que la sortie $f(X)$ ait le comportement désiré.

3.4.3 Variables de contrôle

L'idée principale de cette technique de réduction de la variance lors d'une simulation consiste à produire par simulation une variable auxiliaire qui est en corrélation positive avec la variable d'intérêt.

Soit X la variable aléatoire dont on veut estimer la moyenne $\mu = \mathbb{E}(X)$, Y la variable d'intérêt obtenue par simulation et $\theta = \mathbb{E}(Y)$ sa moyenne. Ici, Y est appelée variable de contrôle.

Construisons l'estimateur de μ de la façon suivante : $X_a = X - a(Y - \theta)$ avec $a \in \mathbb{R}$. Ici

$$\begin{aligned}\mathbb{E}(X_a) &= \mathbb{E}(X - a(Y - \theta)) \\ &= \mathbb{E}(X) - a(\mathbb{E}(Y - \theta)) \\ &= \mathbb{E}(X) - a\mathbb{E}(Y) + a\theta \\ &= \mathbb{E}(X) - a\theta + a\theta \\ &= \mathbb{E}(X) = \mu.\end{aligned}$$

Donc X_a est un estimateur sans biais de μ , $\forall a \in \mathbb{R}$.

On obtient

$$\begin{aligned}\text{Var}(X_a) &= \text{Var}(X - a(Y - \theta)) \\ &= \text{Var}(X) + \text{Var}(a(Y - \theta)) - 2\text{Cov}(X, a(Y - \theta)) \\ &= \text{Var}(X) + a^2\text{Var}(Y - \theta) - 2a\text{Cov}(X, (Y - \theta)) \\ &= \text{Var}(X) + a^2\text{Var}(Y) - 2a\text{Cov}(X, Y).\end{aligned}$$

Donc $\text{Var}(X_a) < \text{Var}(X)$ si $a^2\text{Var}(Y) - 2a\text{Cov}(X, Y) < 0$, c'est-à-dire que

$$0 < a < 2\frac{\text{Cov}(X, Y)}{\text{Var}(Y)}.$$

Alors $\text{Var}(X_a)$ sera minimisée par la valeur optimale a^* de a

$$a^* = \frac{\text{Cov}(X, Y)}{\text{Var}(Y)}.$$

Ainsi la valeur optimale de $\text{Var}(X_a)$ est $\text{Var}(X_a^*)$ avec

$$\begin{aligned}\text{Var}(X_a^*) &= \text{Var}(X) + \left(\frac{\text{Cov}(X, Y)}{\text{Var}(Y)}\right)^2 \text{Var}(Y) - 2\left(\frac{\text{Cov}(X, Y)}{\text{Var}(Y)}\right) \text{Cov}(X, Y) \\ &= (1 - \rho_{XY}^2) \text{Var}(X).\end{aligned}$$

Ici

$$\rho_{XY} = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$$

et ρ_{XY} est appelé le coefficient de corrélation entre X et Y , $-1 \leq \rho_{XY} \leq 1$.

- Dans le meilleur des cas possibles, si le coefficient de corrélation est égal à -1 ou 1 , la variance est réduite à zéro.
 - Dans le pire des cas, quand il n'y a pas de corrélation, $\rho_{XY} = 0$ et la variance reste inchangée, c'est-à-dire $Var(X^*c) = Var(X)$.
 - Dans la situation où on dispose d'un modèle réduit $f_r(X)$.
- Supposons qu'on connaisse $I_r = \mathbb{E}[f_r(X)]$. En considérant la représentation

$$I = \mathbb{E}[f(X)] = I_r + \mathbb{E}[f(X) - f_r(X)]$$

on propose l'estimateur :

$$\hat{I}_N = I_r + \frac{1}{N} \sum_{k=1}^N f(X^{(k)}) - f_r(X^{(k)})$$

Peut réduire beaucoup la variance.

Exemple 3.6 On souhaite estimer

$$I = \int_0^1 e^x dx = E(f(x))$$

avec $X \rightsquigarrow \mathcal{U}([0, 1])$, et $f(x) = \exp(x)$.

Réponse :

- **Résultat théorique :** $I = e - 1 \simeq 1.72$.
- **Monte Carlo :**

$$\hat{I}_N = \frac{1}{N} \sum_{k=1}^N \exp[X^{(k)}]$$

Variance de l'estimateur $MC = \frac{1}{N}(2e - 1) \simeq \frac{1}{N}4.44$

- **Variable de contrôle :** Modèle réduit : $f_r(x) = 1 + x$ (ici $I_r = \frac{3}{2}$).
Estimateur VC :

$$\hat{I}_N = I_r + \frac{1}{N} \sum_{k=1}^N \{\exp[X^{(k)}] - 1 - X^{(k)}\}$$

Variance de l'estimateur

$$VC = \frac{1}{N} \left(3e - \frac{e^2}{2} - \frac{53}{12} \right) \simeq \frac{1}{N} 0.044$$

Il faut donc 100 fois moins de simulations avec l'estimateur VC !

Exemple 3.7 (Méthodes de Romberg statistiques) Pour l'estimation de

$$I = \mathbb{E}[f(X)].$$

On dispose d'un code léger f_r en plus du code lourd f . Le rapport du coût calcul entre un appel à f et un appel à f_r est $q > 1$. Estimateur

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N_r} f_r(\tilde{X}^{(i)}) + \frac{1}{N} \sum f(X^{(i)}) - f_r(X^{(i)})$$

avec $N_r \gg N$

Allocation entre appels au code lourd et appels au code léger à optimiser sous la contrainte $\frac{N_r}{q} + N(1 + \frac{1}{q}) = N_{tot}$

Compromis classique entre erreur d'approximation et erreur d'estimation.

Utilisé dans le cas où $f(X)$ est la solution d'une équation différentielle stochastique discrétisée finement, avec $f_r(X)$ la solution avec un schéma de discrétisation grossier.

3.4.4 Échantillonnage stratifié

Le principe général de cette méthode est de partitionner l'intervalle $(0, 1)$ en plusieurs sous-intervalles appelés strates, voir Hammersley & Handscomb (1964).

Considérons la partition suivante : $0 = \alpha_0 < \alpha_1 < \dots < \alpha_k = 1$ et appliquons la méthode de Monte Carlo élémentaire séparément sur chaque strate. On cherche à estimer l'intégrale suivant :

$$\theta = \int_0^1 f(x)dx = \mathbb{E}(f(x)), X \in D.$$

On choisit l'échantillon à respecter exactement les proportions théoriques dans certaines "strates".

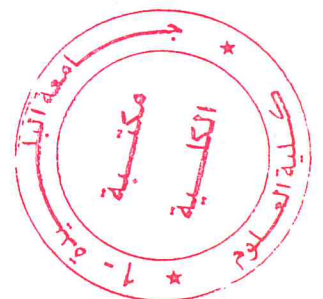
- Deux aspects :

i) Une partition de l'espace D : $D = \cup_{i=1}^m D_i$. On connaît $p_i = P(X \in D_i)$

ii) Formule des probabilités totales :

$$\mathbb{E}[f(X)] = \sum_{i=1}^m \mathbb{E}[f(X) | X \in D_i] P(X \in D_i)$$

on note $J_i = \mathbb{E}[f(X) | X \in D_i]$, $p_i = P(X \in D_i)$



– **Estimation :**

1) On estime J_i par N_i tirages de Monte Carlo :

$$\hat{J}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} f(X^{(j)}), \quad X^{(j)} \sim \mathbb{E}(X \mid X \in D_i).$$

2) On forme l'estimateur $\hat{I}_N = \sum_{i=1}^m \hat{J}_i p_i$

$$\hat{I}_N = \sum_{i=1}^m \hat{J}_i p_i, \quad \hat{J}_i = \sum_{j=1}^{N_i} f(X^{(j)}), \quad X^{(j)} \sim \mathbb{E}(X \mid X \in D_i)$$

Le nombre de simulations total est $N = \sum_{j=1}^{N_i} N_i$.

– L'estimateur est non-biaisé, convergent (si $N_i \rightarrow \infty$) et sa variance est

$$\text{Var}(\hat{I}_N) = \sum_{i=1}^m p_i^2 \text{Var}(\hat{J}_i) = \sum_{i=1}^m p_i^2 \frac{\sigma_i^2}{N_i},$$

avec

$$\sigma_i^2 = \text{Var}(f(X) \mid X \in D_i).$$

L'utilisateur est libre de choisir les allocations N_i (sous la contrainte $\sum_{i=1}^m N_i = N$ fixé).

i) **Stratification proportionnelle :** $N_i = p_i N$. Alors

$$\text{Var}(\hat{I}_N)_{SP} = \frac{1}{N} \sum_{i=1}^m p_i \sigma_i^2,$$

Or (inégalité de convexité pour $x \mapsto x^2$) :

$$\text{Var}(\hat{I}_N)_{MC} = \frac{1}{N} \text{Var}(f(X)) \geq \frac{1}{N} \sum_{i=1}^m p_i \sigma_i^2 = \text{Var}(\hat{I}_N)_{SP}$$

Cependant, l'allocation proportionnelle n'est pas optimale ! L'allocation optimale est celle qui minimise la variance

$$\text{Var}(\hat{I}_N) = \sum_{i=1}^m p_i^2 \frac{\sigma_i^2}{N_i}$$

C'est la solution du problème de minimisation sous contrainte : trouver $(N_i)_{i=1,\dots,m}$ minimisant

$$\sum_{i=1}^m p_i^2 \frac{\sigma_i^2}{N_i} \text{ avec } \sum_{i=1}^m N_i = N$$

Solution (allocation optimale) :

$$N_i = N \frac{p_i \sigma_i}{\sum_{l=1}^m p_l \sigma_l}$$

et alors

$$Var(\hat{I}_N)_{SO} = \frac{1}{N} \left(\sum_{i=1}^m p_i \sigma_i \right)^2$$

On a :

$$Var(\hat{I}_N)_{SO} \leq Var(\hat{I}_N)_{SP} \leq Var(\hat{I}_N)_{MC}$$

Problème : σ_i inconnu. \leftrightarrow stratégie adaptative.

Exemple 3.8

$$I = \int_{-1}^1 e^x dx = \mathbb{E}[f(X)]$$

avec : $X \rightsquigarrow \mathcal{U}([-1, 1])$ de densité $g(x) = \frac{1}{2}$, et $f(x) = \exp(x)$.

Résultat :

$$I = \int_{-1}^1 e^x dx = 2 \sinh(1) \simeq 2.3504$$

Monte Carlo :

$$\hat{I}_N = \frac{2}{N} \sum_{k=1}^N \exp[X^{(k)}]$$

Variance de l'estimateur :

$$MC = \frac{1}{N} \left(\frac{1}{2} - \frac{e^{-2}}{2} \right) \simeq \frac{1}{N} 0.43$$

Code R

```
N=
x=runif(N, -1, 1)
e=exp(x)
S=2*sum(e)
I=S/N
```

N	100	1000	10000	100000
\hat{I}_N	2.256 4	2.368	2.348 6	2.354 8
$Var(\hat{I}_N)$	0.0043	0.00043	4.3×10^{-5}	4.310^{-6}

Approximation de la valeur de l'intégrale I en fonction de N

On remarque que à chaque fois on augmente le N l'approximation de I approche de la valeur théorique et la Var déminue. Pour $N = 10000$

$$MC = \frac{1}{10000} \left(\frac{1}{2} - \frac{e^{-2}}{2} \right) \simeq \frac{1}{10000} 0.43 = 4.3 \times 10^{-5}.$$

ii) **Stratification proportionnelle** : On tire

- $X^{(1)}, \dots, X^{(N/2)}$ selon $\mathcal{U}(-1, 0)$,
- $X^{(N/2+1)}, \dots, X^{(N)}$ selon $\mathcal{U}(0, 1)$.

$$\hat{I}_N = \frac{1}{N} \sum_{k=1}^{N/2} \exp[X^{(k)}] + \frac{1}{N} \sum_{k=N/2+1}^N \exp[X^{(k)}] = \frac{1}{N} \sum_{k=1}^N \exp[X^{(k)}]$$

Variance de l'estimateur $SP \simeq \frac{1}{N} 0.14$.

Il faut 3 fois moins de simulations avec l'estimateur SP.

iii) **Stratification non-proportionnelle** : On tire

- $X^{(1)}, \dots, X^{(N/4)}$ suivant $\mathcal{U}([-1, 0])$,
- $X^{(N/4+1)}, \dots, X^{(N)}$ suivant $\mathcal{U}([0, 1])$.

$$\hat{I}_N = \frac{1}{N} \sum_{k=1}^{N/4} \exp[X^{(k)}] + \frac{1}{N} \sum_{k=N/4+1}^N \exp[X^{(k)}]$$

Variance de l'estimateur $\simeq \frac{1}{N} 0.048$. Il faut 9 fois moins de simulations.

3.4.5 Variables antithétiques

Historiquement le nom des variables antithétiques a été introduit par Hammersley & Mauldon (1956). Contrairement à la méthode des variables de contrôle, la méthode des variables antithétiques introduit une corrélation négative entre les variables.

On cherche à calculer

$$I = \int_{[0,1]^d} f(x) dx$$

Monte Carlo avec un échantillon i.i.d. $X^{(1)}, \dots, X^{(N)}$ de loi $U([0, 1]^d)$:

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^N f(X^{(i)})$$

$$\mathbb{E}[(\hat{I}_N - I)^2] = \frac{1}{N} \text{Var}(f(X)) = \frac{1}{N} \left(\int_{[0,1]^d} f^2(x) dx - I^2 \right)$$

On a aussi

$$I = \int_{[0,1]^d} f(1-x) dx \quad \text{et} \quad I = \int_{[0,1]^d} \frac{f(x) + f(1-x)}{2} dx$$

Monte Carlo avec un échantillon i.i.d. $(X^{(1)}, \dots, X^{(N/2)})$ de loi $U([0, 1]^d)$:

$$\tilde{I}_N = \frac{1}{N} \sum_{i=1}^{N/2} f(X^{(i)}) + f(1 - X^{(i)})$$

Estimateur de type Monte Carlo avec un échantillon

$$\left(\tilde{X}^{(1)}, \dots, \tilde{X}^{(N)} \right) = \left(X^{(1)}, \dots, X^{(N/2)}, 1 - X^{(1)}, \dots, 1 - X^{(N/2)} \right)$$

non indépendant :

$$\tilde{I}_N = \frac{1}{N} \sum_{i=1}^N f(\tilde{X}^{(i)})$$

La fonction f est appelée N fois.

$$\mathbb{E} \left[\left(\tilde{I}_N - I \right)^2 \right] = \frac{1}{N} \left(\text{Var}(f(x)) + \text{Cov}(f(x), f(1-X)) \right) < 0$$

(vrai si f monotone).

Exemple 3.9

$$I = \int_0^1 \frac{1}{1+x} dx$$

La valeur exacte est $I = \ln 2$.

Monte Carlo :

$$\tilde{I}_N = \frac{1}{N} \sum_{i=1}^N \frac{1}{1 + X^{(i)}}$$

$$\text{Var}(\hat{I}_N) = \frac{1}{N} \left(\left(\int_0^1 (1+x)^{-2} dx - \ln 2 \right)^2 \right) = \frac{1}{N} \left(\frac{1}{2} - \ln 2^2 \right) \simeq \frac{1}{N} 1.95 \times 10^{-2}$$

Variables antithétiques :

$$\tilde{I}_N = \frac{1}{N} \sum_{i=1}^{N/2} \left(\frac{1}{1 + X^{(i)}} + \frac{1}{2 - X^{(i)}} \right)$$

$$\text{Var}(\hat{I}_N) = \frac{1}{N} \left(\int_0^1 \left(\frac{1}{2(1+x)} + \frac{1}{2(2-x)} \right)^2 dx - \ln 2^2 \right) \simeq \frac{1}{N} 1.2 \times 10^{-3}.$$

Il faut trois fois moins de simulations avec la méthode. Plus généralement : il faut trouver un couple (X, \tilde{X}) tel que $f(X)$ et $f(\tilde{X})$ ont la même espérance et variance et $\text{Cov}(f(X), f(\tilde{X})) < 0$

Monte Carlo avec un échantillon i.i.d. $\left((X^{(1)}, \tilde{X}^{(1)}), \dots, (X^{(N/2)}, \tilde{X}^{(N/2)}) \right)$:

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N/2} f(X^{(i)}) + f(\tilde{X}^{(i)})$$

$$\mathbb{E} \left[\left(\tilde{I}_N - I \right)^2 \right] = \frac{1}{N} \left(\text{Var}(f(X^{(i)})) + \text{Cov}(f(X), f(\tilde{X})) \right).$$

Comment obtenir Y_1 et Y_2 de même loi que Y avec $\text{Cov}(Y_1, Y_2) < 0$?

Proposition 3.2 Soit $Y = g(U)$ avec U uniforme sur $[0, 1]$. Si g est une fonction monotone, alors

$$\text{Cov}(g(U), g(1 - U)) < 0.$$

3.4.6 Entropie Croisé (CE)

La méthode CE a été introduite par Rubinstein (1999 ; 2001), prolongeant des travaux antérieurs sur la minimisation de la variance (Rubinstein 1997). À l'origine, la méthode CE a été développée pour calculer les probabilités d'événements rares. c'est-à-dire de très petites probabilités - disons moins de 10^{-4} .

L'estimation MMC d'une telle probabilité nécessite un effort de simulation important, inversement proportionnel à l'ampleur de la probabilité d'événement rare. La méthode CE repose sur deux idées.

- La première idée consiste à estimer la probabilité d'intérêt en modifiant progressivement la distribution d'échantillonnage, de la distribution d'origine à une distribution pour laquelle l'événement rare est beaucoup plus susceptible de se produire. Pour éliminer le biais d'estimation, un échantillonnage par importance est utilisé.
- La deuxième idée consiste à utiliser la distance CE pour construire la séquence des distributions d'échantillonnage. Cela simplifie considérablement le calcul numérique à chaque étape et fournit des algorithmes rapides et efficaces faciles à mettre en œuvre par les praticiens.

Le but général de la méthode CE concerne l'estimation d'une attente θ de la forme

$$\theta = \mathbb{E}_f [H(X)] = \int H(x)f(x)dx$$

où H est une fonction à valeurs réelles et f est une densité de probabilité d'une v. a. X .

Le cas où $H(x) = 1_{\{S(x) > \gamma\}}$ est particulièrement important, où S est une autre fonction à valeur réelle.

L'estimateur de Monte Carlo (CMC) brut de θ est

$$\hat{\theta}_{CMC} = \frac{1}{N} \sum_{i=1}^N H(x_i),$$

où X_1, \dots, X_N sont i.i.d. avec une densité f . Il n'est pas difficile de voir que l'erreur relative de l'estimateur CMC est donnée par

$$RE_{CMC} = \frac{\sqrt{\text{var}(\hat{\theta}_{CMC})}}{\theta} = \frac{1 - \theta}{N\theta} \approx \sqrt{\frac{1}{\theta N}}.$$

L'idée sous-jacente à l'échantillonnage préférentiel consiste à échantillonner le $\{X_i\}$ à partir d'un pdf différent g , sous lequel l'événement rare peut être plus probable, tout en compensant le biais ainsi introduit. Soit en particulier

g un pdf pour lequel $H(x)f(x) \neq 0$ pour chaque x . L'espérance θ peut être écrite comme

$$\theta = \int H(x) \frac{f(x)}{g(x)} g(x) dx = \mathbb{E}_g \left[H(x) \frac{f(x)}{g(x)} \right],$$

où l'indice g indique que l'espérance est prise vis-à-vis de g plutôt que de f . Par conséquent, si $X_1, \dots, X_N \sim_{iid} g$, alors

$$\hat{\theta} = \frac{1}{N} \sum_{i=1}^N H(x_i) \frac{f(x_i)}{g(x_i)},$$

est un estimateur non biaisé de θ . Le quotient $W(X) \stackrel{\text{déf}}{=} f(X)/g(X)$ est appelé le rapport de vraisemblance de X . L'estimation θ avec un tel estimateur par échantillonnage préférentiel peut être bénéfique, en termes d'erreur relative plus petite pour le même effort de simulation, si le préférentiel de l'échantillonnage pdf g est choisi de manière appropriée. Notez qu'en prenant $f = g$, on obtient l'estimateur CMC.

L'idée de la méthode CE est de choisir la densité d'importance de l'échantillonnage pdf g dans une classe spécifiée de fonctions pdfs telle que la distance de Kullback-Leibler entre l'échantillonnage d'importance optimale pdf g^* et g est minimal.

La distance de Kullback-Leibler entre deux pdfs g et h est donnée par

$$\begin{aligned} D(g, h) &= \mathbb{E}_g \left[\ln \frac{g(X)}{h(X)} \right] = \int g(x) \ln \frac{g(x)}{h(x)} dx \\ &= \int g(x) \ln g(x) dx - \int g(x) \ln h(x) dx \end{aligned}$$

Nous supposons désormais que la fonction H est positive, et que le pdf nominal f est paramétré par un vecteur de dimension finie u ; c'est-à-dire que $f(x) = f(x, u)$ et que le préférentiel échantillonnage pdf est $f(\cdot, v)$ pour un paramètre v . La procédure de minimisation CE implique la recherche d'un vecteur de paramètre de référence optimal, v^* disons, par minimisation d'entropie croisée :

$$\begin{aligned} v^* &= \arg \min_v D(g^*, f(\cdot, v)) \\ &= \arg \min_v \int H(x) f(x, u) \ln f(x, v) dx \\ &= \arg \min_v \mathbb{E}_u H(x) \ln f(X, v) \\ &= \arg \min_v \mathbb{E}_w H(x) W(X, u, w) \ln f(X, v) \end{aligned} \tag{3.1}$$

où w est un paramètre de référence. Ce v^* peut être estimé via la contrepartie stochastique de (3.1) :

$$v^* = \arg \min_v \frac{1}{N} \sum_{i=1}^N H(x_i) W(X_i, u, w) \ln f(X_i, v) \quad (3.2)$$

où $X_1, \dots, X_N \sim_{iid} f(\cdot, w)$.

- Algorithm 3.1**
1. Choisir un paramètre de référence w , e.g., $w = u$, et générer $X_1, \dots, X_N \sim_{iid} f(\cdot, w)$.
 2. Estimer CE paramètre optimal \hat{v} par le programme stochastique de (3.2).
 3. Générer $X_1, \dots, X_{N_1} \sim_{iid} f(\cdot, \hat{v})$ et estime θ par la technique d'échantillonnage préférentiel.

3.5 Application de la méthode de réduction de la variance

Nous avons vu différentes méthodes pour accélérer la convergence de la méthode de Monte Carlo brute dans la section précédente. Bien sûr, les questions les plus intéressantes sont de savoir quelles méthodes utiliser et quelle est la meilleure méthode.

Il n'y a pas de réponse claire à ces questions. Cependant, nous essayons de formuler quelques conseils simples pour aider à appliquer les méthodes de Monte Carlo :

- S'il existe un bon variable de contrôle, utilisez-le. Souvent, dans cette situation, l'utilisation de variables antithétiques est bénéfique avant l'application de la méthode de la variable de contrôle.
- Pour estimer l'espérance des événements rares, l'échantillonnage préférentiel est préférable, parfois la seule qui fonctionne.
- S'il n'y a pas d'argument évident en faveur d'une méthode de réduction de la variance, soyez prudent lorsque vous appliquez une telle méthode car elle peut également entraîner une perte de temps de calcul.

Chapitre 4

Méthodes de Monte Carlo pour le risque de crédit d'un portefeuille

4.1 Introduction

La crise financière de 2007-2009 a commencé avec un échec majeur sur les marchés du crédit. Les causes de cet échec s'étendent bien au-delà de la modélisation mathématique inadéquat (voir Donnelly et Embrechts [2010] et Brigo et al.[2009] pour des discussions détaillées du point de vue de la finance mathématique). Néanmoins, il est clair que certains des modèles de risque de crédit les plus populaires imparfait. Beaucoup de ces modèles étaient populaires parce qu'ils sont mathématiquement traitables, permettant un calcul facile de diverses mesures de risque. Des modèles plus réalistes (et plus complexes) à un coût de calcul important, nécessitant souvent des méthodes de Monte Carlo pour estimer les quantités d'intérêt .

Le but de ce chapitre est d'étudier les techniques de Monte Carlo utilisées en modélisation du risque de crédit d'un portefeuille. Nous discutons différentes approches pour modéliser les dépendances entre composantes individuelles d'un portefeuille et se concentrer sur deux mesures de risque principales : (Value-At-Risk) Valeur à risque (**VaR**) et (Expected Shortfall) le déficit attendu (**ES**).

L'estimation efficace des mesures du risque de crédit est souvent coûteuse en termes de calcul, comme ça implique l'estimation de petits quantiles.

Techniques de simulation d'événements rares telles que l'importance d'échantillonnage peut réduire considérablement la charge de calcul, mais le choix d'une bonne importance de la distribution d'échantillonnage peut être un

problème mathématique difficile.

Des techniques de simulation récentes telles que la méthode de l'entropie croisée [Rubinstein et Kroese, [2004] ont considérablement amélioré l'applicabilité des techniques d'échantillonnage d'importance en adaptant de choisir la distribution d'échantillonnage d'importance, basé sur des échantillons des modèle de la simulation originale.

La suite de ce chapitre est organisée comme suit. Dans la section 2, nous décrivons le modèle de cadre pour la perte de portefeuille de crédit. La section 3 traite de l'échantillonnage d'importance approches pour estimer les mesures de risque par la méthode de Monte Carlo . Diverses applications à des modèles spécifiques (y compris les modèles de mélange de Bernoulli, les modèles de facteurs) sont donnés à la section 4. Beaucoup de ces modèles capturent les caractéristiques empiriques du risque de crédit, comme le clustering par défaut, qui ne sont pas capturés par les modèles gaussiens standard. Finalement, on présente un exemple d'application pour les modèles de facteurs.

4.2 Modélisation des pertes du crédit d'un portefeuille

Le risque de crédit d'un portefeuille est habituellement évalué dans un contexte statique, où la perte d'un portefeuille est modélisée via une seule variable aléatoire L représentant la somme des pertes subies par les différentes composantes du portefeuille; C'est,

$$L = Loss_1 + \dots + Loss_N.$$

Si les pertes individuelles sont indépendantes, le problème de décrire la distribution de L réduit à le problème de décrire la distribution marginale de chaque perte individuelle. Cependant, en pratique les pertes individuelles ont tendance à dépendance l'une de l'autre. Il est donc important de bien modéliser la dépendance entre les $\{Loss_i\}$.

Les pertes peuvent résulter de changements dans la qualité du crédit ainsi que du défaut. Pour la simplicité, nous nous considérerons que les événements par défaut. Nous écrivons chaque perte individuelle comme le produit de la perte subie si le composant individuel par défaut est une variable aléatoire de Bernoulli (c'est-à-dire, indicateur) qui prend la valeur 1 quand un défaut se produit et 0 sinon. Ainsi, notre modèle est donné par :

$$L = \ell_1 D_1 + \dots + \ell_N D_N$$

où les $\{\ell_i\}$ sont les grandeurs des pertes individuelles et les $\{D_i\}$ sont des variables de Bernoulli modélisant les événements par défaut, les $\{\ell_i\}$ peut être aléatoires ou déterministes. Les preuves empiriques suggèrent une forte relation entre l'ampleur des pertes et le nombre de défauts. Cependant, de nombreux modèles de risque de crédit populaires supposent l'indépendance entre les $\{\ell_i\}$ et $\{D_i\}$. Nous allons se concentrer uniquement sur la modélisation des événements par défaut $\{D_i\}$, si certains des modèles donnés ci-dessous peuvent être modifiés pour incorporer la dépendance entre les pertes et le nombre de défauts.

Le risque de crédit est l'un des risques cruciaux que les instituts financiers doivent gérer. Cela a été souligné de manière dramatique lors de la crise des prêts hypothécaires à risque. Le risque de crédit fait référence à la perte possible due à un défaut ou à une dégradation de la cote de crédit des débiteurs associés à un portefeuille. Une des caractéristiques du risque de crédit est ce que l'on appelle le «cygne noir», qui se produit rarement, mais qui entraîne une perte importante pour les investisseurs, le cas échéant.

Quelques modèles ont été développés pour mesurer le risque de crédit ; le plus largement utilisé est le modèle gaussien à facteur de copule. Dans ce cadre, les risques des débiteurs sont modélisés par un ensemble de facteurs de risque sous-jacents appelés facteurs de risque systémiques et idiosyncratiques. Dans cette section, nous passons en revue cette approche, en suivant les indications du projet de recherche de Han [5]. La notation ci-dessous est utilisée tout au long de ce chapitre :

N = nombre de crédateurs

C = nombre d'états de crédit, numérotés $\{0, \dots, C - 1\}$

S = dimension du facteur de risque du système

L = le pourcentage de perte du portefeuille

$\mathbf{1}_n^c$ = fonction de l'indicateur : 1 si le crédateur n est à l'état de crédit c et 0 sinon

EAD_n = exposition par défaut, c-à-d la valeur perdue si le débiteur n est par défaut

LGC_n^c = pourcentage de perte/gain si le crédateur n passe à l'état de crédit c

Y_n = indice de solvabilité du crédateur n

P_η^γ = la probabilité de sortir de l'état η de crédit à l'état γ de crédit

$c(n)$ = état du crédit initial pour le crédateur n

Le pourcentage de perte du portefeuille total peut être exprimé comme suit :

$$\begin{aligned} L &= \frac{\sum_{n=1}^N EAD_n \left(\sum_{c=0}^{C-1} LGC_n^c \mathbf{1}_n^c \right)}{\sum_{n=1}^N EAD_n} \\ &= \sum_{n=1}^N w_n \left(\sum_{c=0}^{C-1} LGC_n^c \mathbf{1}_n^c \right) = \sum_{n=1}^N w_n L_n. \end{aligned}$$

où

$$w_n = \frac{EAD_n}{\sum_{n=1}^N EAD_n}$$

est le poids d'exposition du débiteur n et

$$L_n = \sum_{c=0}^{C-1} LGC_n^c \mathbf{1}_n^c$$

est le taux de perte du créancier n .

4.3 Mesures de risques

La distribution de L souvent appelée la distribution de perte et notée F_L est l'objet central de la modélisation du risque de crédit. F_L généralement n'est pas disponible sous forme fermée. Au lieu de cela, certaines mesures de risques sont utilisées pour décrire ses principales caractéristiques, en particulier son comportement en queue.

Définition 4.1 *La mesure du risque le plus largement utilisé dans le risque de crédit est la Valeur à risque (VaR), qui décrit les quantiles de la distribution des pertes. Par exemple, la VaR à 99% d'un portefeuille est la valeur au de-la variable de perte L telle qu'une perte plus importante ne se produirait que 1% du temps. La VaR de niveau de confiance α est donnée par*

$$v_\alpha = F_L^{-1}(\alpha) \tag{4.1}$$

où F_L^{-1} est l'inverse généralisée de F_L :

$$F_L^{-1}(\alpha) = \inf\{\ell : F_L(\ell) \geq \alpha\}. \tag{4.2}$$

Les valeurs communes pour α sont 0.95, 0.99, 0.995 et 0.999.

L'utilisation de la **VaR** comme mesure de risque a fait l'objet de critiques significatives (voir Bluhm et autres [2010] et McNeil et autres [2005] pour des discussions). En particulier, il a la caractéristique contre-intuitive qu'il n'est pas sous-additif : La **VaR** de deux portefeuilles pourrait être supérieure à la somme des **VaR** des différents portefeuilles. Autrement dit, la **VaR** d'un portefeuille n'est pas nécessairement réduite par la diversification. Ce qui a conduit Artzner et al. [1999] de proposer une classe de mesures de risque cohérentes, qui satisfait certaines «naturelles» exigences, y compris la sous-additivité.

Définition 4.2 *L'un des plus populaires d'entre eux est le manque à gagner attendu (ES), également connu sous le nom de valeur conditionnelle à risque (CVaR). Le déficit attendu α est donné par*

$$c_\alpha = \mathbb{E}[L \mid L \geq v_\alpha]. \quad (4.3)$$

Le déficit attendu est également un exemple de mesure du risque spectral, voir Bluhm et al.[2010].

4.3.1 Dépendance de modélisation

Le défi principal dans la modélisation du risque de crédit du portefeuille consiste à décrire la relation entre événements par défaut. Les défauts ne se produisent pas indépendamment les uns des autres, mais ont plutôt tendance à se regrouper. Ces grappes par défaut peuvent apparaître en raison de conditions spécifiques au secteur, telles qu'un ralentissement dans un secteur ou un marché particulier, ou en raison de facteurs macroéconomiques plus larges.

Un défaut important dans les modèles de crédit dans la crise financière de 2007 – 2009 était qu'ils ont échoué à modéliser la possibilité qu'un grand nombre de défauts puisse se produire simultanément . Afin de discuter de cette limitation, nous devons introduire un certain nombre de mesures de dépendance différentes qui décrire la relation entre les variables aléatoires.

Le mesure de dépendance la plus simple entre deux variables aléatoires par paire X et Y est donnée par leurs coefficient de corrélation linéaire

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}.$$

Son analogue multivarié est la matrice de corrélation. La structure de dépendance du vecteur aléatoire X est complètement spécifiée par sa matrice de corrélation si et seulement si X a une distribution elliptique, voir McNeil et

al.[2005]. Les cas spéciaux importants sont les distributions t-Student multivariées et normales multivariées. Un inconvénient de la corrélation linéaire (et d'autres mesures de corrélation, telles que la corrélation de rang) est qu'il décrit le comportement conjoint moyen des variables aléatoires. Dans la gestion des risques, il est un événement extrême, plutôt que des événements typiques, qui sont d'un intérêt primordial. Plus précisément, donné deux variables aléatoires X et Y , avec des distributions F_X et F_Y , nous définissons le coefficient de la dépendance supérieure de la queue comme

$$\lambda_\mu = \lim_{q \uparrow 1} P(Y > F_Y^{-1}(q) \mid X > F_X^{-1}(q)), \quad (4.4)$$

et le coefficient de dépendance inférieure de la queue comme

$$\lambda_d = \lim_{q \downarrow 0} P(Y \leq F_Y^{-1}(q) \mid X \leq F_X^{-1}(q)). \quad (4.5)$$

Ces mesures décrivent la relation entre les variables dans les queues des distributions. On dit que la distribution a une indépendance de queue supérieure (inférieure) si $\lambda_\mu = 0$ ($\lambda_d = 0$).

Certains des modèles plus populaires de risque de crédit, en particulier, les différents modèles de copules gaussiennes-queue de l'exposition de l'indépendance dans les deux queues. Ce n'est pas clairement qu'une caractéristique souhaitable dans les modèles de risque, comme empirique la preuve tend à indiquer que les défauts et les facteurs de risque tendent à devenir plus corrélés dans des conditions extrêmes. À l'exception des modèles canoniques gaussiens, tous les modèles décrits dans les sections suivantes possèdent la dépendance de la queue.

4.4 Estimation des mesures de risque via Monte Carlo

Pour une distribution générale des pertes F_L , calcul analytique des différentes mesures de risque décrites dans la dernière section est généralement impossible. Souvent, la seule approche possible consiste à estimer ces mesures de risques, en utilisant des méthodes de Monte Carlo. Pour continuer, nous avons besoin d'une méthode d'échantillonnage d'une suite de v.a indépendantes et identiquement distribués (iid) de la variable aléatoire L et une méthode d'estimation des mesures de risque, étant donné un échantillon iid L_1, \dots, L_N . La méthodologie pour estimer les mesures de risque est largement indépendante du modèle, qui est l'objet de cette section.

L'estimation Monte Carlo de la **VaR** s'avère un peu plus difficile que le traditionnel problème d'estimation d'une attente. En particulier, les estimateurs **VaR** sont des fonctions non linéaires de l'échantillon. De nombreuses méthodes classiques Monte Carlo ne peuvent pas être appliquées à l'estimation de la **VaR** ou besoin d'être modifié pour bien fonctionner. En outre, il est généralement difficile de trouver des intervalles de confiance pour les estimateurs **VaR**.

Estimateurs Monte Carlo bruts (cruds)

L'estimateur Monte Carlo brut (**CMC**) de la **VaR** est l'estimateur du quantile des statistiques classiques ; voir Van der Vaart [1998] pour une discussion de ses propriétés dans un contexte statistique. Il remplace la fonction de distribution inconnue F_L de L , dans la définition de **VaR** en (4.1) avec la fonction empirique de distribution \hat{F}_L .

$$\hat{v}_\alpha = \inf \left\{ \ell : \hat{F}_L(\ell) \geq \alpha \right\},$$

où

$$\hat{F}_L(\ell) = \frac{1}{N} \sum 1(L_i \leq \ell) \quad (4.6)$$

est la fonction de distribution empirique de l'échantillon iid L_1, \dots, L_N . Notez que \hat{F}_L est une fonction en escalier, par conséquent l'estimateur quantile **CMC** peut être facilement obtenu en ordonnant les $\{L_i\}$ comme $L_1 \leq L_2 \leq \dots \leq L_N$ et trouver la plus grande valeur de $\lceil \alpha N \rceil$ ième.

Algorithm 4.1 (Estimateur CMC de VaR)

1. Générer un échantillon iid L_1, \dots, L_N .
2. Ordonner l'échantillon du plus petit au plus grand $L_1 \leq \dots \leq L_N$
3. Trouver $\hat{v}_\alpha = L_{(\lceil \alpha N \rceil)}$.

L'estimateur **CMC** pour **ES** est plus simple, car **ES** est simplement une valeur moyenne.

L'estimateur est donné par :

$$\hat{c}_\alpha = \frac{1}{N(1-\alpha)} \sum_{i=1}^N L_i 1_{(L_i \geq \hat{v}_\alpha)}.$$

La variance de l'estimateur **VaR** est difficile à évaluer, car l'estimateur n'est pas une moyenne des variables aléatoires iid. Cependant, les théorèmes

de limite centrale suivants, donnés avec références dans Hong et Liu [2011], montrent que les estimateurs **VaR** et **ES** ont asymptotiquement distributions normales.

1.

$$\sqrt{N}(\hat{v}_\alpha - v_\alpha) \xrightarrow{D} \frac{\sqrt{\alpha(1-\alpha)}}{f_L(v_\alpha)} Z_1,$$

2.

$$\sqrt{N}(\hat{c}_\alpha - c_\alpha) \xrightarrow{D} \frac{\sqrt{\text{Var}(L1_{(L>v_\alpha)})}}{1-\alpha} Z_2,$$

où Z_1 et Z_2 sont des variables aléatoires normales et \xrightarrow{D} indique une convergence en distribution.

Exemple 4.1 (*application de l'algorithme d'estimateur CMC de VaR et ES*)

L'échantillon suit la loi exponentiel pour $\lambda = 2$ dans notre exemple : comme

$$F(v_\alpha) = 1 - \exp(-\lambda v_\alpha) = \alpha$$

donc

$$F^{-1}(\alpha) = \frac{-1}{\lambda} \log(1 - \alpha) = v_\alpha$$

TAB.4.1-Approximation de **VAR** et **ES** par **MCC** pour un modèle exponentiel avec $N=1000$

$N=10^4$	v_α	\hat{v}_α	std \hat{v}_α	c_α	\hat{c}_α	std \hat{c}_α
CMC($\alpha = 0.95$)	1.497866	1.490578	-0.728821	1.997519	1.945584	5.19355
CMC($\alpha = 0.99$)	2.302585	2.259239	-4.334621	2.770251	2.493226	27.70251

4.4.1 Echantillonnage préférentiel (Importance Sampling (IS))

Les estimateurs **CMC** de **VaR** et **ES** nécessitent généralement des échantillons de très grande taille, pour un niveau acceptable de précision. C'est parce que les estimateurs se concentrent sur des événements relativement «rare». Il existe un important corpus de théorie consacré aux méthodes de Monte Carlo efficaces pour les événements rares. Cette théorie a été principalement développée dans le contexte de l'estimation des probabilités des événements rares de la forme $\ell = P(S(X) > \gamma)$ pour une fonction à valeur réelle S , seuil γ , et vecteur aléatoire X . La discussion suivante supposera la

familiarité avec ces concepts. L'approche de l'échantillonnage préférentiel de l'estimation du quantile a été suggérée dans Glynn [1996]. Nous remplaçons l'estimateur **CMC** de la fonction de distribution empirique par l'estimateur **IS**

$$\hat{F}_L^{IS}(\ell) = 1 - \frac{1}{N} \sum_{i=1}^N W(L_i) 1_{(L_i > \ell)},$$

où les $\{L_i\}$ sont tirés de la densité **IS** g et $W(l) = \frac{f_L(l)}{g(l)}$ est le rapport de vraisemblance.

Notez que cet estimateur se concentre sur la queue droite de la distribution, (voir Glynn [1996]). Cela conduit ensuite à l'estimateur **VaR** de **IS**

$$\hat{v}_\alpha^{IS} = \inf \left\{ \ell : \hat{F}_L^{IS}(\ell) \geq \alpha \right\}. \quad (4.7)$$

L'estimateur **ES** correspondant est

$$\hat{c}_\alpha^{IS} = \frac{1}{N(1-\alpha)} \sum_{i=1}^N W(L_i) L_i 1_{(L_i \geq \hat{v}_\alpha^{IS})} \quad (4.8)$$

où les L_i sont tirés de g . Si g est choisi de telle sorte qu'il tire de la queue droite de L plus fréquemment, cet estimateur pourrait fournir des performances nettement meilleures que l'estimateur **CMC**. En pratique, l'estimateur **IS** de **VaR** est calculé comme suit.

Algorithm 4.2 (*Estimation IS de VaR*)

1. Générer L_1, \dots, L_N de la densité **IS** de g .
2. Calculer les rapports de vraisemblance $W(L_1), \dots, W(L_N)$.
3. Ordonner l'échantillon du plus petit au plus grand $L_1 \leq \dots \leq L_N$
4. Trouver $N^* = \sup \left\{ n : \frac{1}{N} \sum_{i=n}^N W(L_{(i)}) \geq 1 - \alpha \right\}$.
5. Trouver $\hat{v}_\alpha^{IS} = L_{[\alpha N^*]}$

Jusqu'à présent, nous avons pris g comme donné. Le théorème centrale limite suivant, donnés dans Hong et Liu [2011] et Sun et Hong [2010], suggèrent un bon choix de g .

Théorème 4.3 (Théorème central limite pour l'estimateur **IS** de **VaR** et **ES**)

Si L a une densité positive et différentiable f_L dans un voisinage de v_α et qu'il existe un $\varepsilon > 0$ tel que $W(\ell)$ est borné pour tout $\ell \in (v_\alpha - \varepsilon, v_\alpha + \varepsilon)$ et $\mathbb{E}_g 1_{(L \geq v_\alpha - \varepsilon)} (W(L))^p$ est fini pour $p > 2$, alors que $N \rightarrow \infty$.

1.

$$\sqrt{N} (\hat{v}_\alpha^{IS} - v_\alpha) \xrightarrow{D} \frac{\sqrt{\text{Var}_g(W(L)1(L \geq v_\alpha))}}{f_L(v_\alpha)} Z_1.$$

2.

$$\sqrt{N} (\hat{c}_\alpha^{IS} - c_\alpha) \xrightarrow{D} \frac{\sqrt{\text{Var}_g(W(L)1(L \geq v_\alpha))}}{(1 - \alpha)} Z_2$$

où Z_1 et Z_2 sont des variables aléatoires normales et \xrightarrow{D} indique une convergence en distribution.

Ceci suggère qu'un bon choix de g , au moins asymptotiquement, est celui qui minimise $\text{Var}_g(W(L)1(L > v_\alpha))$. Cela équivaut à trouver la densité g qui minimise la variance de

$$\hat{\ell}^{IS} = \frac{1}{N} \sum_{i=1}^N W(L_i)1(L_i > v_\alpha)$$

où les $\{L_i\}$ sont tirés de g . Ceci est l'estimateur *IS* standard pour

$$\ell = P(L > v_\alpha).$$

Bien sûr, le calcul de $\hat{\ell}^{IS}$ implique v_α , qui est la quantité inconnue que nous cherchons à estimer.

Cependant, une estimation approximative de v_α peut souvent être obtenue, soit par une approximation ou par une simulation initiale en utilisant l'estimateur **CMC** de **VaR**. Estimateurs d'échantillonnage préférentiel pour le **VaR** et le **ES** fourniront souvent de très grands gains d'efficacité, même dans les milieux où le premier estimateur de v_α est assez inexact. Une autre complication est que nous ne connaissons généralement pas f_L , la densité de L . Ainsi, nous ne pouvons pas appliquer l'échantillonnage préférentiel au $\{L_i\}$ directement. Au lieu de cela, nous cherchons à représenter L en tant que fonction S soit d'un vecteur aléatoire X de densité connue f_X , soit d'un processus stochastique vectoriel $X = \{X(t), 0 \leq t \leq T\}$, pour laquelle on peut appliquer un échantillonnage référentiel.

En pratique, la procédure pour appliquer l'échantillonnage préférentiel est la suivante.

Algorithm 4.4 (*Estimation d'échantillonnage d'importance pour la VaR et ES*) Étant donné une représentation $L = S(X)$:

1. Calculer une estimation initiale de v_α , Notée \hat{v}_α .
2. Trouver une densité d'importance appropriée pour estimer $P(L > \hat{v}_\alpha)$.
3. Générer $L_1 = S(X_1), \dots, L_N = S(X_N)$ sous la densité IS et calculer le rapports de vraisemblance $W(X_1), \dots, W(X_N)$.
4. Calculer l'estimation de la **VAR** comme dans 4.7 et l'estimation **ES** comme dans 4.8.

Exemple 4.2 (application de l'algorithme d'estimateur **IS** de **VAR** et **ES**)

L'échantillon suit loi exponentielle dans notre exemple :comme

$$F(v_\alpha) = 1 - \exp(-\lambda v_\alpha) = \alpha$$

TAB.4.2-Approximation de **VAR** et **ES** par **IS** pour un modèle exponentiel avec N=1000

N=10 ³	v_α	\hat{v}_α	std \hat{v}_α	c_α	\hat{c}_α	std \hat{c}_α
CMC($\alpha = 0.95$)	1.49787	1.54110	0.00137	2.01922	2.0596	0.00128
CMC($\alpha = 0.99$)	2.302585	2.10746	0.00617	2.45426	2.6997	0.00776

TAB.4.3-Approximation de **VAR** et **ES** par **IS** pour un modèle exponentiel avec N=10000

N=10 ⁴	v_α	\hat{v}_α	std \hat{v}_α	c_α	\hat{c}_α	std \hat{c}_α
CMC($\alpha = 0.95$)	1.49787	1.49725	0.00016	1.98499	1.98895	0.00039
CMC($\alpha = 0.99$)	2.30259	2.27244	0.00030	2.70429	2.73133	0.00027

TAB.4.4-Approximation de **VAR** et **ES** par **IS** pour un modèle Normal avec N=1000

N=10 ³	v_α	\hat{v}_α	std \hat{v}_α	c_α	\hat{c}_α	std \hat{c}_α
CMC($\alpha = 0.95$)	1.64485	1.58144	0.00201	1.99546	2.03537	0.00126
CMC($\alpha = 0.99$)	2.32635	2.19497	0.00415	2.67967	2.94764	0.00848

TAB.4.4-Approximation de **VAR** et **ES** par **IS** pour un modèle Normal avec N=10000

N=10 ⁴	v_α	\hat{v}_α	std \hat{v}_α	c_α	\hat{c}_α	std \hat{c}_α
CMC($\alpha = 0.95$)	1.64485	1.62441	0.00020	2.05771	2.06182	0.00041
CMC($\alpha = 0.99$)	2.32635	2.31128	0.00015	2.69029	2.71718	0.00027

4.4.2 Échantillonnage préférentiel adaptatif

Parce que les modèles de risque de crédit sont généralement compliqués, il peut être difficile de trouver a priori une bonne densité d'échantillonnage

préférentiel g . Les méthodes d'échantillonnage préférentiel adaptative visent pour éviter les problèmes théoriques et informatiques difficiles en «apprenant» une bonne densité à partir des données. Nous supposons ici que f_L , la densité de L , n'est pas connue et qu'une représentation de la forme $L = S(X)$, où X a une densité f_X , peut être utilisé à la place. Nous appliquons l'échantillonnage d'importance à X . Étant donné une densité IS prédéfinie g_θ paramétré par θ , l'idée est de prélever un échantillon initial X_1, \dots, X_N et essayez d'apprendre les paramètres optimaux en utilisant cet exemple. Si l'échantillon initial X_1, \dots, X_M peut être échantillonné directement à partir de la densité de variance zéro $g^*(x) = f(x|S(x) > v_\alpha)$, alors les paramètres peuvent être choisis soit pour minimiser la distance CE à g^* :

$$\hat{\theta}_{CE}^* = \arg \max_{\theta} \frac{1}{M} \sum_{i=1}^M \log(g_\theta(X_i)),$$

ou pour minimiser la variance de l'estimateur

$$\hat{\theta}_{VM}^* = \arg \min_{\theta} \frac{1}{M} \sum_{i=1}^M W_\theta(X_i).$$

Dans certains cas, g^* est échantillonné à l'aide des méthodes Monte Carlo Chaîne de Markov (MCMC) (voir Kroese et al., [2011] pour une introduction). Cependant, parce que la probabilité d'une perte est supérieure à v_α est pas trop petit, nous pouvons souvent utiliser une méthode d'acceptation-rejet plus directe ici.

Algorithm 4.5 (*Échantillonnage Approximatif à partir de g^**)

1. Générer un échantillon L_1, \dots, L_N .
2. Commandez l'échantillon du plus petit au plus grand en tant que $L_1 \leq \dots \leq L_N$.
3. Choisissez $L_{(\lceil \alpha N \rceil)}, \dots, L_{(M)}$ en tant qu'échantillon approximatif de g^* .

Un très petit échantillon est généralement suffisant pour trouver de très bons paramètres CE ou VM . le coût de calcul supplémentaire de l'essai est généralement faible par rapport aux coûts globaux de la simulation.

En effet, il n'y a pratiquement aucun frais par rapport aux méthodes non adaptatives pour estimation quantile, comme de telles méthodes utilisent des essais pour trouver une estimation initiale de v_α . Un similaire approche adaptative est prise dans Reitan et al. [2010]. Pour une méthode alternative, où les paramètres sont mis à jour pendant la phase d'échantillonnage primaire, voir Egloff et Leippold [2010].

4.5 Modèles spécifiques

Dans cette section, nous discutons deux classes spécifiques de modèle de risque de crédit : modèles de mélange de Bernoulli et modèles de facteurs, bien que chacun de ces modèles soit basé sur le cadre général 4.4, ils utilisent différentes structures mathématiques pour modéliser les dépendances entre les variables par défaut $\{D_i\}$. En conséquence, chaque modèle nécessite une approche différente de Monte Carlo pour estimer efficacement la **VaR** et le **ES**.

4.5.1 Le modèle de mélange de Bernoulli

Les modèles de mélange Bernoulli sont des classes fondamentales des modèles de risque de crédit, car de nombreux crédits de risque peuvent être représentés comme un modèle de mélange. Il est facile d'appliquer l'échantillonnage préférentiel à ces modèles.

Dans un modèle de mélange de Bernoulli, les variables par défaut de Bernoulli D_1, \dots, D_n sont conditionnellement indépendants donné un vecteur de probabilités par défaut $P = (P_1, \dots, P_n)$. On suppose que ces probabilités par défaut sont de la forme $P(\Psi)$, où Ψ est un vecteur aléatoire avec une densité connue f_Ψ conditionnel à P , calculer L réduit à une somme pondérée de queue légère (light-tailed) Variables aléatoires indépendantes.

Il est assez simple d'échantillonner à partir d'un modèle de mélange de Bernoulli.

Algorithm 4.6 (*Échantillonnage d'un modèle de mélange de Bernoulli*)

1. Générer un vecteur de probabilités de succès $P = (P_1, \dots, P_n)$.
2. Étant donné P , générer $D_1 \rightsquigarrow Ber(P_1), \dots, D_n \rightsquigarrow Ber(P_n)$.

Exemple :

4.5.2 Échantillonnage préférentiel en une étape

Il n'est généralement pas possible d'appliquer directement l'échantillonnage d'importance à L , car la distribution de L est souvent indisponible sous forme fermée. Au lieu de cela, nous pouvons appliquer l'échantillonnage d'importance au dessin P où les D_1, \dots, D_n conditionnelles à P . Il est plus simple d'appliquer l'échantillonnage d'importance dans le second cas. Si nous supposons que ℓ_1, \dots, ℓ_n sont des constantes, alors

$$L = \ell_1 D_1 + \dots + \ell_n D_n$$

est la somme des variables aléatoires indépendantes, la i ème variable prenant la valeur ℓ_i avec probabilité P_i et 0 autrement. Nous torsions exponentiellement chacune de ces variables de sorte que la probabilité de défaut pour la i ème composante est donnée par

$$\tilde{P}_i = \frac{P_i \exp(\theta \ell_i)}{P_i \exp(\theta \ell_i) + 1 - P_i}$$

Le choix unique « asymptotiquement efficace » de θ est la solution de $k_n(\theta^* | P) = v_\alpha$, où

$$k_n(\theta | P) = \sum_{i=1}^n \log [P_i \exp(\theta \ell_i) + 1 - P_i]$$

est la fonction génératrice cumulée de $\{\ell_i D_i\}$ conditionnelle à P .

Algorithm 4.7 (*Échantillonnage d'importance en une étape pour un modèle de mélange*)

1. Générer $P = (P_1, \dots, P_n)$.
2. Trouver θ^* , la solution à $k'_n(\theta) = v_\alpha$. (Cette étape doit généralement être effectuée numériquement).
3. Si $\theta^* < 0$, alors $\theta^* = 0$
4. Calculer

$$\tilde{P}_i = \frac{P_i \exp(\theta \ell_i)}{P_i \exp(\theta \ell_i) + 1 - P_i} \quad i = 1, \dots, n$$

5. Étant donnée $\tilde{P}_1, \dots, \tilde{P}_n$, générer $D_i \rightsquigarrow \text{Ber}(\tilde{P}_i)$, $i = 1, \dots, n$
6. Retour $L = \ell_1 D_1 + \dots + \ell_n D_n$ et le rapport de vraisemblance correspondant

$$W(L) = \exp(k_n(\theta^* | P) - \theta^* L).$$

Malheureusement, cette approche ne peut pas donner un estimateur asymptotiquement efficace pour $\ell = P(L > v_\alpha)$.

C'est parce que P peut jouer un rôle essentiel dans la dynamique d'un événement rare. Par exemple, dans le contexte des modèles factoriels gaussiens, Glasserman et Li [2005] montrent que cette efficacité asymptotique ne peut être atteinte que si la corrélation entre les valeurs par défaut diminue (à un certain rythme) que $n \rightarrow \infty$ et $v_\alpha \rightarrow \infty$.

4.5.3 Échantillonnage préférentiel en deux étapes

Un schéma d'échantillonnage d'importance potentiellement plus efficace implique un échantillonnage d'importance dans la génération P ainsi que D_1, \dots, D_n . Nous pouvons décomposer la variance de $\hat{\ell}$ comme

$$z = P(L > v_\alpha | P(\psi))$$

La densité de variance zéro g^* pour un tel problème est donnée par

$$g_\Psi^*(\psi) \propto P(L > v_\alpha | P(\psi)) f_\Psi(\psi)$$

La constante de normalisation est l'inconnue ℓ , donc ce n'est pas une densité *IS* pratique.

Il existe deux approches communes pour trouver une bonne densité *IS*. Une approche utilise une densité g whose mean est égal à la valeur de g_Ψ^* . Ce mode est la solution à un généralement intraitable problème d'optimisation. Étant donné g , le plan d'échantillonnage en deux étapes est résumé comme suit.

Algorithm 4.8 (*Échantillonnage d'importance en deux étapes pour un modèle de mélange*)

1. Générer Ψ à partir de g_Ψ .
2. Générer $P = P(\Psi)$.
3. Trouver θ^* , la solution à $k'_n(\theta) = v_\alpha$
4. Calculer

$$\tilde{P}_i = \frac{P_i \exp(\theta^* \ell_i)}{P_i \exp(\theta^* \ell_i) + 1 - P_i}, i = 1, \dots, n$$

5. Étant donné $\tilde{P}_1, \dots, \tilde{P}_n$, génère $D_i \rightsquigarrow \text{Ber}(\tilde{P}_i), i = 1, \dots, n$
6. Retour $L = \ell_1 D_1 + \dots + \ell_n D_n$ et le rapport de vraisemblance correspondant

$$W(L) = \frac{f_\Psi(\psi)}{g_\Psi(\psi)} \exp(k_n(\theta^* | P) - \theta^* L).$$

Exemple 4.3 *Un modèle de mélange de Bernoulli avec des probabilités bêta. Nous considérons un modèle de mélange de Bernoulli simple pour un portefeuille de $n = 1000$ composantes, avec $\ell_1 = \dots = \ell_n = 1$. Les probabilités par défaut sont toutes égales, avec $P \rightsquigarrow \text{Beta}(0.5, 9)$. Nous considérons trois approches : **CMC**, **CE** et échantillonnage en une étape. L'approche **CE** trouve les résultats de P correspondant aux échantillons $N(1 - \alpha)$ les plus élevés de L . Il calcule ensuite le MLE pour une distribution bêta numériquement.*

Pour l'approche IS, $k_n(\theta|P) = \hat{v}_\alpha$ peut être résolu analytiquement. Cependant, pour ce problème, la dynamique de L est largement pilotée par P . Ainsi, l'estimateur IS fonctionne très mal. Chaque estimateur a été utilisé pour calculer 100 estimations. Les moyens et les écarts-types de ces estimateurs sont indiqués. Pour IS, les premiers 10% de l'échantillon utilisé pour calculer une estimation approximative de \hat{v}_α . Pour CE, les premiers 10% de l'échantillon ont été utilisés pour apprendre les paramètres.

4.5.4 Modèles de facteurs

Dans les modèles factoriels, la i ème composante est par défaut lorsqu'une variable aléatoire correspondante X_i traverse un seuil prédéfini ρ_i . C'est,

$$D_i = 1(X_i > \rho_i), \quad i = 1, \dots, n.$$

La variable X_i peut être parfois considérée comme correspondant à un temps par défaut, comme dans les copules, bien que ce ne soit pas le cas. La relation entre les $\{D_i\}$ est imposé ayant tous les $\{X_i\}$ dépendent d'un vecteur de facteurs communs Ψ . Un modèle avec un facteur est appelé un modèle à facteur unique; un modèle avec un facteur est appelé un modèle multifactoriel. Ces facteurs peuvent correspondre à des facteurs macroéconomiques ou sectoriels, bien qu'ils n'ont pas besoin d'avoir une interprétation économique. Dans le cas le plus simple d'un modèle de facteur linéaire, chaque X_i est une somme pondérée des facteurs et une autre variable aléatoire, ε_i qui représente le risque idiosyncrasique spécifique au composant conditionnel à Ψ , les modèles de facteurs sont les modèles de mélange de Bernoulli.

Les modèles de facteurs les plus populaires sont basés sur les distributions normales et t-Student, nous nous concentrons sur trois modèles de facteurs spécifiques.

Modèle factoriel de Gauss

- **Modèle factoriel Gaussien** : chaque X_i a la représentation

$$X_i = \alpha_{i1}Z_1 + \dots + \alpha_{im}Z_m + \alpha_i\varepsilon_i,$$

où les $\{Z_j\}$ et $\{\varepsilon_i\}$ sont des variables aléatoires normales standard indépendantes et les coefficients sont choisis de sorte que la distribution marginale de chaque X_i soit normale. Ici, conditionnel sur $Z_1 = z_1, \dots, Z_m = z_m$ (donc $\Psi = Z$), la probabilité par défaut pour la i ème

composante est :

$$\begin{aligned} P_i &= P\left(\varepsilon_i > \frac{\rho_i - (\alpha_{i1}z_1 + \dots + \alpha_{im}z_m)}{\alpha_i}\right) \\ &= \phi\left(\frac{(\alpha_{i1}z_1 + \dots + \alpha_{im}z_m) - \rho_i}{\alpha_i}\right) \end{aligned}$$

- **Modèle factoriel t-Student** : chaque X_i est une somme pondérée de variables aléatoires t-Student. Habituellement, le modèle factoriel t-Student est choisi de telle sorte que chaque X_i a la représentation suivante

$$X_i = \sqrt{\frac{r}{V}}(\alpha_{i1}Z_1 + \dots + \alpha_{im}Z_m + \alpha_i\varepsilon_i),$$

- **Modèle factoriel unique** : plus général avec de fortes queues et une dépendance de la queue est introduit à Bassamboo et al. [2008]. C'est une extension des modèles de mélange de moyenne-variance normale décrits à Frey et McNeil [2001]. Ici, chaque X_i est de la forme :

$$X_i = \frac{\alpha_i Z_i + \sqrt{1 - \alpha_i^2} \varepsilon_i}{W}$$

où les $\{\varepsilon_i\}$ sont des variables aléatoires iid indépendantes de la variable aléatoire Z , et W est un variable aléatoire indépendant de Z et du $\{\varepsilon_i\}$, avec une densité f_W qui satisfait

$$f_W(w) = \lambda w^{v-1} + o(w^{v-1}) \text{ lorsque } w \downarrow 0. \quad (4.9)$$

Ce modèle inclut t-Student à facteur unique comme cas particulier, comme la distribution du khi-carré satisfait 4.9. Conditionnel à $Z = z$ et $W = w$ (donc, $\Psi = (Z, W)$) par défaut les probabilités sont

$$P_i = P\left(\varepsilon_i > \frac{w\rho_i - \alpha_i z}{\sqrt{1 - \alpha_i^2}}\right).$$

Il est généralement facile d'échantillonner à partir d'un modèle factoriel.

Algorithm 4.9 (*Échantillonnage à partir d'un modèle factoriel*)

1. Générer les facteurs communs Ψ et les risques idiosyncratiques $\varepsilon_1, \dots, \varepsilon_n$.
2. Calculer X_1, \dots, X_n selon le modèle.
3. Calculer

$$L = \ell_1 1(X_1 > \rho_1) + \dots + \ell_n 1(X_n > \rho_n).$$

4.5.5 Echantillonnage préférentiel

Les modèles de facteurs sont généralement des modèles de mélange de Bernoulli. Ainsi, l'échantillonnage d'importance peut être appliqué comme ci-dessus. Il est généralement nécessaire d'utiliser un plan d'échantillonnage en deux étapes. La difficulté réside dans le choix de g , la densité IS pour les facteurs communs Ψ .

Dans le cas des modèles factoriels gaussiens, où $\Psi = Z$, Glasserman et Li [2005] utilisent une densité normale multivariée $\mathcal{N}(\mu, I)$ avec le vecteur moyen μ fixé égal au mode de g_Z^* , le mode, à son tour, peut être obtenu comme la solution au problème d'optimisation

$$\mu^* = \arg \max_z P(L > v_\alpha \mid Z = z) \exp(z^\top z/2) \quad (4.10)$$

Glasserman et Li suggèrent un certain nombre d'approximations qui simplifient ce problème. Une approche est l'approximation constante, où L est remplacé par $\mathbb{E}[L \mid Z = z]$ et $P(L > v_\alpha \mid Z = z)$ est remplacé par $1(\mathbb{E}[L \mid Z = z] > v_\alpha)$. Dans ce cas, 4.10 devient

$$\arg \min_z \{z^\top z : \mathbb{E}[L \mid Z = z] > v_\alpha\}. \quad (4.11)$$

Une autre approche est l'approximation liée à la queue, qui est montrée asymptotiquement optimale dans le cas d'un portefeuille homogène à facteur unique. Cette approche se rapproche de $P(L > v_\alpha \mid Z = z)$ par sa limite supérieure, et 4.10 devient

$$\arg \max \{k_n(\theta_{v_\alpha} \mid z) - \theta_{v_\alpha} v_\alpha - z^\top z/2\}$$

où $\theta_{v_\alpha} = \theta_{v_\alpha}(z)$ est la solution à $k_n(\theta \mid z) = v_\alpha$ et k_n est donné en ?? .

Dans un contexte multi-facteurs, le problème de trouver une bonne approximation de g^* devient beaucoup plus difficile. C'est parce que plus d'une combinaison de facteurs peut causer une perte plus grande que v_α . Glasserman et al. [2008] proposent une approche qui tente essentiellement de partitionner événement rare $\{L > v_\alpha\}$ en différents sous-événements, chaque sous-événement correspond à un ensemble particulier de facteurs prenant de grandes valeurs, et ils résolvent 4.11 pour chacun de ces événements. Cette approche est montrée être asymptotiquement efficace dans certains contextes.

Dans le modèle t-Student donné ci-dessus, Kang et Shahabuddin [2005] proposent premier échantillonnage V , puis Z_1, \dots, Z_m . Étant donné V , ils procèdent comme dans Glasserman et al. [2008]. Ils proposent une torsion exponentielle de V par un paramètre qui est à nouveau la solution d'un problème d'optimisation. Notez que cette approche est très coûteuse en calcul, car

elle nécessite plusieurs procédures d'optimisation numérique par échantillon. Kang et Shahabuddin [2005] suggèrent en utilisant un plan d'échantillonnage stratifié pour minimiser ce coût.

Pour le modèle général à un seul facteur, Bassamboo et al. [2008] présentent deux méthodes :

- Premièrement, ils proposent une torsion exponentielle W et trouvent un bon paramètre de torsion θ en minimisant la limite supérieure du rapport de vraisemblance. Cette approche donne une erreur relative bornée dans certains conditions techniques.
- Dans la seconde, ils appliquent une torsion du taux de risque à $V = 1/W$, voir Juneja et Shahabuddin [2006] pour une discussion de cette méthode. Encore une fois, ils choisissent le paramètre de torsion minimiser la limite supérieure du rapport de vraisemblance. Dans certaines conditions techniques, le résultat de l'estimateur est démontré asymptotiquement efficace.

Une autre méthode pour appliquer la réduction de la variance aux modèles factoriels t-Student est donnée dans Chan et Kroese [2010]. Dans cette approche, la **VaR** peut être estimée en calculant les attentes de variables aléatoires gamma tronquées.

Exemple 4.4 *Un modèle de facteur gaussien*

Nous considérons un exemple suggéré dans Glasserman et Li [2005]. Dans cet exemple, le portefeuille est de taille $n = 1000$, avec $\ell_i = (\lceil 5i/n \rceil)^2$. Les barrières sont données par

$$\rho_i = \phi^{-1}(1 - P_i),$$

où

$$P_i = 0.01 * (1 + \sin(16\pi * i/n)).$$

Les chargements de facteur $m = 10$, $\{\alpha_{ij}\}$ sont générés uniformément sur $(0, 1/\sqrt{m})$.

Nous calculons la **VaR** et le **ES** en utilisant trois méthodes différentes : **CMC**, Glasserman et Li méthode et Cross-Entropy.

Pour l'algorithme de Glasserman et Li, nous appliquons seulement l'échantillonnage d'importance à la $\{Z_i\}$, comme tordant le $\{D_i\}$ ne fait pas une différence substantielle dans ce cas, et prend considérablement plus de temps. Nous tirons le $\{Z_i\}$ d'une distribution $\mathcal{N}(\mu, I)$, avec μ solution de (4.5) trouvée via la recherche de racine numérique. Dans l'approche *CE*, nous fixons les moyens de $\{Z_i\}$ et la moyenne de $\{\varepsilon_i\}$ égale aux moyennes de l'échantillon de $\{Z_i\}$ et $\{\varepsilon_i\}$ correspondant à les $[N \times (1 - \alpha)]$ valeurs les plus élevées de L .

Chapitre 5

Présentation de R

5.0.6 Généralités

R est un système, communément appelé langage et logiciel, qui permet de réaliser des analyses statistiques, particulièrement, il comporte des moyens qui rendent possible la manipulation des données, les calculs et les représentations graphiques. R a aussi la possibilité d'exécuter des programmes stockés dans des fichiers textes et comporte un grand nombre de procédures statistiques appelées paquets (packages).

Ces derniers permettent de traiter assez rapidement des sujets aussi variés que les modèles linéaires (simples et généralisés), la régression (linéaire et non linéaire), les séries chronologiques, les tests paramétriques et non paramétriques classiques, les différentes méthodes d'analyse des données,... Plusieurs paquets, tel que, **ade4**, **FactoMineR**, **MASS**, **multivariate**, **scatterplot3d** et **rgl** entre autres sont destinés à l'analyse des données statistiques multidimensionnelles.

Il a été initialement créé, en 1996, par *Robert Gentleman* et *Ross Ihaka* du département de statistique de l'Université *d'Auckland* en Nouvelle Zélande. Depuis 1997, il s'est formé une équipe "**R Core Team**" qui développe R. Il est conçu pour pouvoir être utilisé avec les systèmes d'exploitation **Unix**, **Linux**, **Windows** et **MacOS**.

R est devenu un logiciel libre et gratuit en 1995. R est à la fois un langage de programmation et un progiciel de fonctions statistiques. La version de base de R contient déjà un grand nombre de fonctions statistiques et graphiques permettant, par exemple, de calculer une moyenne ou une variance ou de tracer un histogramme.

De nombreux chercheurs ont développé au cours des années des fonctions plus avancées qui sont disponibles à tous les utilisateurs de R.

CRAN :

Le "*Comprehensive R Archive Network*" (CRAN) est un ensemble de sites qui fournit ce qui est nécessaire à la distribution de R, ses extensions sa documentation, ses fichiers sources et ses fichiers binaires. Le site maître du CRAN est situé en Autriche à Vienne ,nous pouvons y accéder par L'URL : `http://cran.r-project.org/`

Installation du logiciel R :

Le logiciel est téléchargé sur le site web officiel de R (`http://www.r-project.org/`), il faut ensuite se diriger dans download CRAN. Choisissez un site miroir proche de chez vous (par exemple : en Algérie), les téléchargements seront probablement plus rapides, vous trouvez ensuite un encadrement légendé download and install R.

Sous Windows XP

Pour Windows, cliquez sur Windows puis base. Cliquez ensuite sur le fichier "exist" (par exemple : R-3.5.1-win.exe). Le programme d'installation est alors téléchargé sur votre ordinateur. Il suffit de cliquer dessus et de suivre les instructions. Un dossier portant le nom de la version de R télécharger (R-3.5.1-win.exe dans ce cas) est créée.

Il est situé, à partir du disque d'ôte C, dans la série de dossiers suivante : program ... les /R.

Dans ce dossier se trouve le dossier library qui comprend les packages base de R. Un autre élément utile doit être localisé : le fichier.R data. Celui-ci n'est pas apparent au début. Il contiendra tous les objets que vous créez et sauvegardez dans R. Sur mon ordinateur, il apparaît, à partir du disque d'ôte C.

5.0.7 Instalation des packages

Un package R se présente sous la forme d'un fichier compressé qui regroupe des fonctions, les documents et peut contenir du package facilitant l'utilisation et le partage de fonction de R.

Autrement dire, un package est une compilation du certains outils déjà présents dans l'installation de base de R. En effet, lors de l'installation de R, un dossier Library s'est crée par défaut, sous Windows fichiers binaires pré-compilés.

Il comprend les packages de base de R. Mais d'autres packages qui vous seront utiles pour réaliser vos analyses statistiques seront à télécharger puis à installer.

Sur le CRAN, le réseau officiel de distribution de code et de documentation R. Plus de 3000 bibliothèques R sont en ce moment (il y'a des mois) disponibles sur le site CRAN.

D'autres bibliothèques sont disponibles sur la page Web de chercheurs ou en annexe de publications scientifiques.

L'installation du module («package») package `ade4` doit se faire une seule fois, lors de la première utilisation. Pour les utilisations subséquentes, il suffira de charger le package après l'ouverture du logiciel R. On suppose qu'on dispose d'une connexion internet pour effectuer ces opérations. On peut ensuite travailler hors-connexion.

1. Ouvrir le logiciel R en cliquant sur l'icône créé sur votre bureau lors du téléchargement. Dans la fenêtre qui apparaît, aller dans menu "packages / Choisir le site miroir "CRAN" (pas trop loin si possible), par exemple : `usthb` Algérie et cliquer sur OK (voir figure 1.1).
2. Menu " packages / Installer des packages", Dans la liste des packages, choisir `ade4` et cliquer sur OK.
3. Menu « packages/ Mettre à jour les packages » , pour obtenir la dernière version des packages déjà installés.

5.0.8 Charger un package déjà installé

Aller dans le menu "packages -Charger le package" puis sélectionner le package ou taper la commande :

`library(nom package)`

`library(nom-de-la-bibliotheque) #` par exemple : `library(MASS)` ou encore :

`require(nom-de-la-bibliotheque) #` par exemple : `require(MASS)`

`library()#` permet aussi de lister les packages installes sur notre machine.

5.0.9 Exemple d'utilisation du package : `rpart`

Pour utiliser le package `rpart` (pour construire des arbres de décision), il faut charger le package `rpart` à chaque ouverture du logiciel R. Utiliser les menus, choisir `rpart` dans la liste proposée et cliquer sur ok.

Exemple 1 ligne de commande :

```
library(rpart)
```

search()

On peut « télécharger » un package (pour éviter les conflits entre fonctions de packages différents par exemple).

5.0.10 Syntaxe générale

Pour une v.a. X suivant une loi notée loi dans R, la syntaxe générale est la suivante :

- pour obtenir "la densité" de X , la commande est : **dloi**, on ajoute la lettre d devant loi,
- pour obtenir la fonction de répartition de X , la commande est : **ploi**, on ajoute la lettre p devant loi,
- pour obtenir le quantile de X , la commande est : **qloi**; on ajoute la lettre q devant loi,
- pour simuler des réalisations de var suivant la même loi que X , la commande est : **rloi**, on ajoute la lettre r devant loi.

Ci-dessous, un tableau grossier récapitulatif :

Les noms loi les plus célèbres sont : norm (pour la loi normale), binom (pour la loi binomiale), unif (pour la loi uniforme), geom (pour la loi géométrique), pois (pour la loi de Poisson), t (pour la loi de Student), chisq (pour la loi du Chi-deux), exp (pour la loi exponentielle), f (pour la loi de Fisher).

Loi : loi	densité	fonction de répartition	quantile	simulation
notations	$f(x); P(X = x)$	$F(x)$	valeur liée à $F(x)$	x_1, \dots, x_n
commandes	dloi	ploi	qloi	rloi

5.0.11 Densité μ

- Pour une var X discrète, on appelle "la densité" de X en x la probabilité $P(X = x)$.
- Pour une var X à densité de densité f_X , on appelle "la densité" de X en x la fonction $f_X(x)$.

5.0.12 Commandes

Si la loi de X dépend d'un ou de plusieurs paramètres, disons par1 et par2, alors la densité de X en x est donnée par les commandes : dloi(x,par1,par2)
Quelques exemples sont décrits ci-dessous

Loi	binomiale	géométrique	exponentielle	normale
paramètres	$n \in \mathbb{N}; p \in]0; 1[$	$p \in]0; 1[$	$\lambda > 0$	$\mu \in \mathbb{R}, \sigma > 0$
$X \rightsquigarrow$	$\mathcal{B}(n; p)$	$\mathcal{G}(p)$	$\varepsilon(\lambda)$	$\mathcal{N}(\mu, \sigma^2)$
$X(\Omega)$	$\{0, \dots, n\}$	\mathbb{N}	$[0, \infty[$	\mathbb{R}
$P(X = x)$	$\binom{n}{x} p^x (1-p)^{n-x}$	$p(1-p)^x$	$\lambda e^{-\lambda x}$	$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
commandes	<code>dbinom(x, n, p)</code>	<code>dgeom(x, p)</code>	<code>dexp(x, lambda)</code>	<code>dnorm(x, mu, sigma)</code>

Pour compléter, voir : `help("dgamma")`, `help("dt")`, `help("dchisq")` et `help("df")`.

Calculs

On fait : `dbinom(4, 8, 0.3)`

[1] 0.1361

Ainsi, on a calculé la densité d'une v.a. $X \rightsquigarrow \mathcal{B}(8, 0.3)$ en $x = 4$:

$$P(X = 4) = \binom{8}{4} 0.3^4 (1 - 0.3)^{8-4}$$

On peut vérifier cela en faisant :

`choose(8, 4) * 0.3^4 * (1 - 0.3)^(8 - 4)`

[1] 0.1361

`dnorm(1.7, 2, 0.12)`

[1] 0.1460692

Ainsi, on a calculé la densité d'une var $X \rightsquigarrow \mathcal{N}(2, (0.12)^2)$ en $x = 1.7$:

$$f_x(1.7) = \frac{1}{\sqrt{2\pi * 0.12^2}} e^{-\frac{(1.7-2)^2}{2*0.12^2}}$$

On peut vérifier cela en faisant :

`(1/sqrt(2*pi*(0.12)^2))*exp(-(1.7-2)^2/(2*(0.12)^2))`

[1] 0.1460692

Pour calculer la densité en plusieurs valeurs, on prend pour x le vecteur ayant pour éléments ces valeurs. On peut faire de même avec un ensemble de paramètres et les arguments correspondants.

On fait :

`dbinom(c(4, 6), 8, 0.3)`

[1] 0.13613670 0.01000188

On a ainsi calculé la densité d'une v.a. $X \rightsquigarrow \mathcal{B}(8, 0.3)$ pour $x \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$.
On fait :

```
vec=dexp(2, c(1, 2, 3))  
vec  
[1] 0.135335283 0.036631278 0.007436257
```

Représentation graphique On peut représenter le graphe de la densité d'une v.a. X discrète avec la commande plot et l'option type= h.

On fait :

```
plot(0:5,dbinom(0:5,5,0.2),type= "h",ylab="P(X=x)")
```

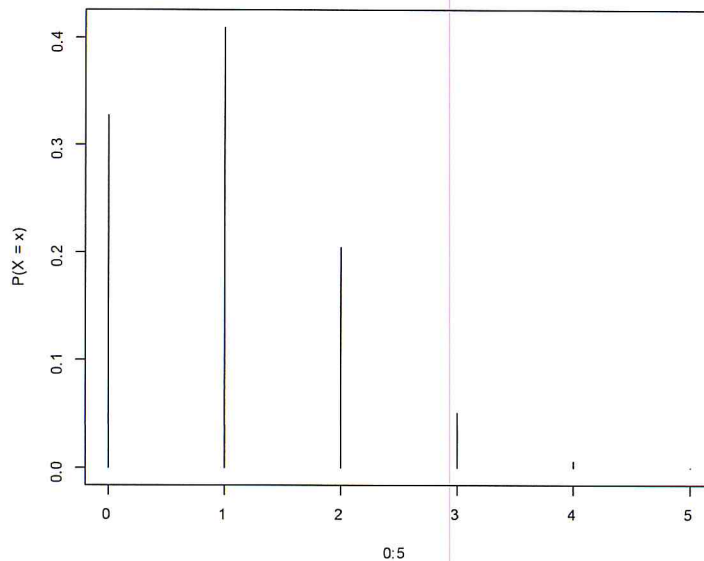


FIG. 5.1 – Diagramme en bâtons d'une v.a. binomiale

On a ainsi représenté le graphe de la densité d'une v.a. $X \rightsquigarrow \mathcal{B}(5, 0.2)$.

On peut représenter le graphe de la densité d'une v.a. X à densité avec la commande curve.

On fait :

```
curve(dnorm(x, 5, 1.5), 0.5, 9.5, ylab="fX(x)")
```

On a ainsi représenté le graphe de la densité d'une v.a. $X \rightsquigarrow \mathcal{N}(5, 1.5^2)$.

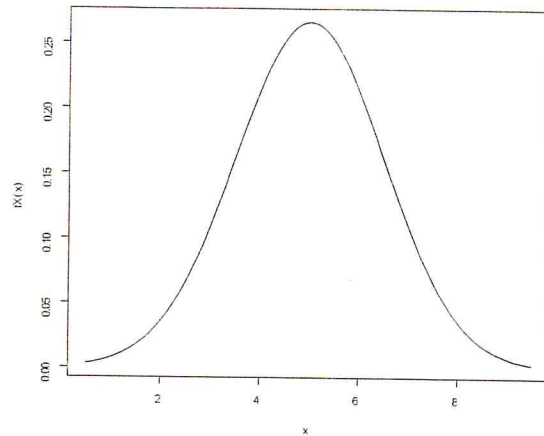


FIG. 5.2 – Graphe d’une densité de la loi normale $N(5, 2.25)$

5.0.13 Commandes

Si la loi de X dépend d’un ou de plusieurs paramètres, disons `par1` et `par2`, alors la fonction de répartition de X en x est donnée par les commandes :

`ploi(x,par1,par2)`

On peut calculer :

$$P(X > x) = 1 - F_X(x),$$

en faisant :

`ploi(x,par1,par2,lower.tail=FALSE)`

Calculs

`pbinom(4,8,0.3)`

0.9420324

Ainsi, on a calculé la fonction de répartition d’une v.a. $X \rightsquigarrow \mathcal{B}(8, 0.3)$ en $x = 4$:

$$F_X(4) = P(X \leq 4) = \sum_{k=0}^4 P(X = k)$$

On peut vérifier le calcul précédent en faisant :

```
sum(dbinom(0:4,8,0.3))
```

```
[1] 0.9420324
```

```
pnorm(12,9,2)
```

```
[1] 0.9331928
```

Ainsi, on a calculé la fonction de répartition d'une var $X \rightsquigarrow \mathcal{N}(9, 2^2)$ en $x = 12$:

$$F_X(12) = P(X \leq 12) = \int_{-\infty}^{12} \frac{1}{\sqrt{2\pi * 2^2}} e^{-\frac{(t-9)^2}{2 * 2^2}} dt$$

```
pexp(2, 3, lower.tail = FALSE)
```

```
[1] 0.002478752
```

Ainsi, on a calculé : $P(X > 2)$, $X \rightsquigarrow \mathcal{E}(3)$:

$$P(X > 2) = \int_2^{\infty} f_X(x) dx = \int_2^{\infty} 3e^{-3x} dx = [-e^{-3x}]_2^{\infty} = e^{-3 \times 2} = e^{-6}.$$

On peut vérifier le calcul précédent en faisant :

```
exp(-6)
```

```
[1] 0.002478752
```

Représentation graphique On peut représenter le graphe de la fonction de répartition d'une v.a. X discrète avec la commande `stepfun`.

```
plot(stepfun(0:15, c(0, pbinom(0:15, 15, 0.6))), ylab="FX(x)", main="title").
```

On a ainsi représenté le graphe de la fonction de répartition d'une var $X \rightsquigarrow \mathcal{B}(15, 0.6)$.

On peut représenter le graphe de la fonction de répartition d'une v.a. X à densité avec la commande `curve`.

```
curve(pnorm(x, 5, 1.5), 0.5, 9.5, ylab="FX(x)")
```

On a ainsi représenté le graphe de la fonction de répartition d'une v.a. $X \rightsquigarrow \mathcal{N}(5, 1.5^2)$.

5.0.14 Quantiles

Soit $p \in]0, 1[$ et X une v.a..

– Si X est discrète, on appelle p -ème quantile de X l'entier x_p défini par :

$$x_p = \inf\{k \in \mathbb{Z} : F_X(k) \geq p\}.$$

– Si X est à densité, on appelle p -ème quantile de X le réel x_P tel que $F_X(x_P) = p$.

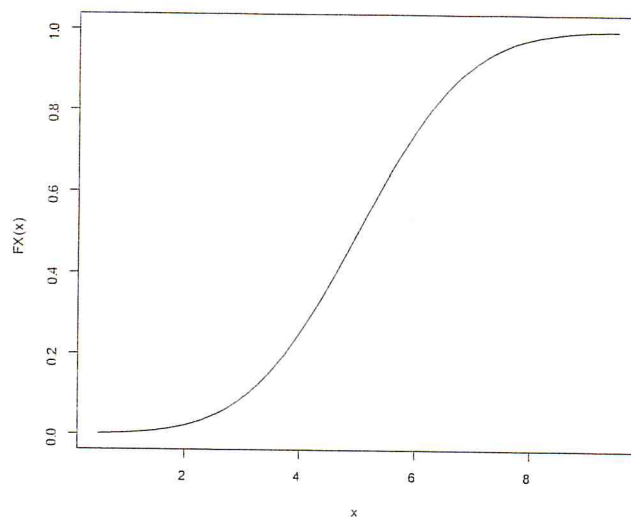


FIG. 5.3 – Fonction de répartition d'une lois normale

5.0.15 Commandes

Si la loi de X dépend d'un ou de plusieurs paramètres, disons par1 et par2, alors le p -ème quantile de X est donné par les commandes :

`qloi(p,par1,par2)`

5.0.16 Calculs

```
qbinom(0.25, 5, 0.6)
[1] 2
```

Ainsi, on a calculé le p -ème quantile avec $p = 0.25$ d'une variable $X \rightsquigarrow \mathcal{B}(5, 0.6)$:

$$x_{0.25} = \inf\{k \in \mathbb{Z}; F_X(k) \geq 0.25\}.$$

On peut vérifier ce résultat en cherchant toutes les valeurs de $F_X(x)$ pour $x \in \{0, 1, 2, 3, 4, 5\}$:

```
pbinom(0:5, 5, 0.6)
[1] 0.01024 0.08704 0.31744 0.66304 0.92224 1.00000
```


On constate alors que

$$F_X(1) = 0,08704 < 0,25 \leq 0,31744 = F_X(2),$$

donc $x_{0,25} = 2$

```
qnorm(0.975,0,1)
[1] 1.959964
```

Ainsi, on a calculé le p -ème quantile avec $p = 0,975$ d'une variable $X \rightsquigarrow \mathcal{N}(0;1)$: le réel $x_{0,975}$ tel que $F_X(x_{0,975}) = 0,975$.

On peut vérifier cela en faisant :

```
pnorm(1.96,0,1)
[1] 0.9750021
```

On aurait aussi pu considérer les commandes : `pnorm(1.96)`, sans spécifier $\mu = 0$ et $\sigma = 1$; la commande prend par défaut les paramètres de la loi centrée et réduite.

5.1 Simulation

5.1.1 La commande `sample`

L'utilisation de base `sample(x)`, où x est un vecteur numérique logique ou chaîne de caractères est un vecteur dont les éléments correspondent au tirage d'un élément de x sans remise.

```
x = 1:7
sample(x)
[1] 2 4 1 7 5 6 3
```

On obtient des résultats différents ; l'ordre de la permutation est due au hasard à chaque fois. On peut conserver le résultat d'une permutation aléatoire dans un vecteur pour le réutiliser ultérieurement :

```
y=sample(c("rouge","vert","bleu","blanc","noir"))
y
[1] "noir" "blanc" "vert" "rouge" "bleu"
```

5.1.2 Options

Il existe des options dans `sample` permettant de modifier le type de tirage. On les active en rajoutant une ou plusieurs commandes dans `sample`.

Par exemple, on fait :

```
sample(1:3,size=2,replace=TRUE,prob=c(25/100,20/100,55/100))
```

Quelques options sont présentées ci-dessous.

Option : size. Les commandes : `size = "n"`, où n est un entier, précise le nombre de fois n que l'on effectuera le tirage. Par défaut, le nombre de tirage est égal à la longueur de x .

On fait :

```
sample(1:10,size=3)
[1] 10 7 3
```

Dans la syntaxe de `sample`, l'argument `size` est en deuxième position. On peut donc se contenter de :

```
sample(1:10, 3).
```

Option : replace. Les commandes : `replace = L`, où L est `TRUE` ou `FALSE`, dont que les tirages sont avec remise si $L = TRUE$ et sans remise sinon.

```
sample(1:5, size = 9, replace = TRUE)
[1] 5 1 4 1 3 1 2 1 3
```

Option : prob. La commande `prob` permet la simulation de v.a. dont le support est l'ensemble des éléments de x , que l'on suppose au nombre de k . Ainsi, la loi commune de ces v.a. est :

$P(X = j\text{-ème élément de } x) = p_j$, avec $j \in \{1, \dots, k\}$, $p_j \in [0; 1]$ et $\sum_{j=1}^k p_j = 1$.

On ajoute donc `prob = vec`, où `vec` est le vecteur des probabilités (p_1, \dots, p_k) . Si `prob` n'est pas indiqué, par défaut $p_j = 1/k$ (tirage équiprobable)

```
y=cc("rouge", "vert", "bleu", "blanc", "noir")
sample(y,2,prob=c(10/100,30/100,10/100,30/100,20/100))
[1] "rouge" "vert"
```

Simulation par des lois préprogrammées

```
rpois(10, 2)
0 2 2 3 3 5 1 3 1 2
```

Ainsi, on a simulé des réalisations de 10 var indépendantes suivant chacune la loi de Poisson $\mathcal{P}(2)$.

```
sum(rbinom(80, 1, 0.02))
[1] 3
```

On a alors simulé une réalisation de la var $\sum_{i=1}^{80} X_i \rightsquigarrow \mathcal{B}(80, 0.02)$, cela revient à simulé

une réalisation d'une variable $Y \rightsquigarrow \mathcal{B}(80, 0.02)$:

```
rbinom(1.80, 0.02).
```

```
x = rnorm(15, 22, 2)
```

Ainsi, on a construit un vecteur dont les éléments sont des réalisations de 15 variables indépendantes suivant chacune la loi normale $\mathcal{N}(22, 2^2)$.

5.1.3 Les lois de probabilité

R fournit un jeu très complet de tables statistiques. Les fonctions fournies nous donnent la densité de probabilité cumulée (ou fonction de répartition), $P(X \leq x)$, la densité de probabilité, le quantile (ou fonction de répartition inverse) et des série de nombres pseudo aléatoires générées suivant la loi de probabilité considérée.

Syntaxes des lois de probabilités sous R

Loi de probabilité	Noms R	Arguments supplémentaires
Beta	beta	shape, shape2, ncp
Binomiale	binom	size, prob
Cauchy	cauchy	location, scale
Khi2	chisq	df, ncp
Exponentielle	exp	rate
Fisher	f	df1, df2, ncp
Gamma	gamma	shape, scale
Géométrique	geom	prob
Hypergéométrique	hyper	m, n, k
Log-normale	lnorm	meanlog, sdlog
Logistique	logis	location, scale
Négative binomiale	nbinom	size, prob
Normal	norm	mean, sd
Poisson	pois	lambda
Student	t	df, ncp
Uniforme	unif	df, ncp
Weibull	weibull	shape, scale

Pour les lois Beta, Khi2, Fisher et Student ncp désigne le paramètre de décentrage.

Quatre préfixes sont possibles pour obtenir les sorties désirées :

d : correspond à la densité de probabilité.

p : correspond à la fonction de répartition (valeur inverse du quantile).

q : correspond au quantile.

r : correspond à la génération aléatoire de nombre.

Le premier argument est 'x'(une valeur) pour dnom-loi, 'q'(un quantile) pour pnom-loi, et 'n'(un nombre) pour rnom-loi.

Exemples avec la loi normale (arguments par défauts : $mean = 0$, $sd = 1$) :

```
dnorm(0.37)
```

```
[1] 0.3725483
```

donne l'ordonnée de la courbe de densité à la valeur 0.37.

```
pnorm(0)
```

```
[1] 0.5
```

donne la probabilité de la fonction de répartition au quantile 0.

```
round(qnorm(0.975), 2)
```

```
[1] 1.96
```



```
# donne le quantile pour la probabilité de 0.975 de la fonction de répartition.
```

```
> rnorm(3) # engendre 3 valeurs de la loi normale  
[1] -1.5651925 1.2417975 0.9014982
```

5.1.4 Aide intégrée au logiciel R

La Commande `help ()`

R contient une aide en ligne (`help` : en anglais) très complète et très bien structurée sur toutes les fonctions que vous pouvez utiliser ainsi que sur les différents symboles du langage. Cette aide accessible au moyen de plusieurs commandes dont la principale est `help ()`.

Attention : il peut arriver que cet alias ne fonctionne pas et il est alors nécessaire d'utiliser la fonction `help()` avec des guillemets.

```
?function # Ne fonctionne pas
```

```
help (function) # Renvoie une erreur
```

```
help("function") # Appel correct
```

Notes et commentaires :

Il est important d'ajouter des notes et des commentaires au fichier texte contenant les commandes sauvegardées, afin de se rappeler plus tard de l'objectif des calculs ou encore

la raison de telle ou telle façon de faire. Une ligne de commentaires commence par le caractère « `#` ».

Lorsqu'on les copie dans la console R, les lignes de commande ou les sections de lignes commençant par « `#` » ne sont pas exécutées par R.

5.1.5 Quelques commandes complémentaires

En plus de la commande principale `help ()`, il existe quelques autres fonctions complémentaires qui peuvent se révéler utiles lorsque l'on cherche de l'aide sur une commande donnée. Nous les présentons ci-dessous :

– `help.start ()` : cette fonction ouvre un navigateur avec plusieurs liens vers des manuels au format HTML, de l'aide sur toutes les fonctions présentes dans tous les packages de R (aussi au format HTML), une *FAQ* (Questions fréquemment posées) qui contient les réponses aux questions les plus fréquemment posées sur R.

Vous y trouvez également un moteur de recherche dans l'aide mots clefs. On y trouve aussi quelques autres documents plus techniques.

– `help .search ()` ou `?? ()` : cette fonction est à utiliser lorsque l'on ne connaît pas le nom d'une commande. Elle renvoie une liste des différentes

fonctions (et des packages dans les quels elles se trouvent) ayant un rapport avec votre requête. Essayer par exemple : **help.search("mean")**.

– **apropos()** : cette instruction renvoie une liste de toutes les fonctions dont le nom contient le paramètre d'appel. Ainsi **apropos("mean")** renvoie toutes les fonctions dont le nom contient le mot mean.

– **Library(help : package)** : cette commande permet de lister toutes les fonctions présentes dans la librairie package. Elle donne les mêmes résultats que la commande **help("package")**. Nous vous conseillons d'essayer les instructions suivantes afin d'obtenir la liste des fonctions principales de R :

```
library(help=base)
```

```
library(help=utils)
```

```
library(help=datasets)
```

```
library(help=stats)
```

```
library(help=graphics)
```

```
library(help=grDevices)
```

La fonction **Library(help = package)** renvoie la liste de toutes les fonctions de librairies packages. Inversement, l'instruction **find("fonction")** permet de savoir à quelle librairie appartient la fonction : **function()**. Par exemple, la fonction **t.test** est dans le package (**stats**) :

```
find("t.test")
```

```
[1] "package : stats"
```

Voici trois autres fonctions qui pourront également se révéler utiles :

– **data()** : cette commande renvoie la liste de tous les jeux de données présents dans R.

– **example()** : cette instruction donne des exemples sur l'utilisation d'une fonction.

Ainsi, **example(mean)** exécute les instructions présentes dans la section exemples du fichier d'aide obtenu quand on tape **help(mean)**.

```
example("mean")
```

```
x <-c(0 :10,50)#50 valeurs entre 0 et 10
```

```
xm <-mean(x)
```

```
c(xm,mean(x,trim =0.10))
```

```
[1] 8.75 5.50
```

– **demo()** : cette instruction est similaire à la précédente, mais n'est accessible que pour un nombre limité de fonctions. Lorsqu'elle est disponible, elle présente un éventuel des possibilités offertes par une fonction. Essayez par exemple : **demo(graphics)**

5.1.6 Créer et manipuler des données

Voici quelques définitions importantes dans la manipulation des données :

Définition (objet) un objet est un espace dans lequel vous pouvez stocker tout ce qui vous intéresse.

Définition (vecteur) un vecteur est un objet d'un même mode pour toutes les valeurs qui le constituent.

Définition (matrice) une matrice est un objet d'un même mode pour toutes les valeurs qui la constituent.

Définition (liste) une liste est un objet permettant de stocker des objets qui peuvent être hétérogènes, c'est-à-dire qui n'ont pas tous le même mode ou la même longueur.

Définition (tableau) un tableau de données, ou `data.frame` en anglais, est une liste particulière dont les composantes sont de même longueur et dont les modes peuvent être différents (est une collection de vecteurs de même longueur). Un `data.frame` est un tableau à double entrée : les lignes sont les individus sur lesquels les mesures sont faites et les colonnes sont les variables.

5.1.7 Les objets : création et types

Création

Simplement par affectation avec les opérateurs `<-`, `->` en lui donnant un nom :

```
b<-sqrt(2) (crée l'objet b) #(racine carrée)
x<-b (x reçoit la valeur b)
x=b (x reçoit la valeur b)
b->x (x reçoit la valeur b)
```

Si un objet n'existe pas l'affectation le crée. Sinon l'affectation écrase la valeur précédente.

Affichage de la valeur d'un objet

```
a (affiche le contenu de a)
q (affiche le contenu de la fonction q qui permet de quitter)
print(a) (affiche le contenu de a)
```

Les différents types d'un objet

- Objets simples : ce sont des objets qui ne comportent que des éléments de même mode.
- Objets composés (ou hétérogènes) : ce sont des objets qui comportent des éléments qui peuvent être de modes différents.

Les principaux modes sont :

- null (objet vide) : NULL
- logical (booleen) : TRUE, FALSE ou T, F
- numeric (valeur numérique) : 1, 2.3222, π , $1e - 10$,
- character (chaîne de caractère) : 'bonjour', "K", "Bebe"
- complex (nombre complexe) : $2 + 0i$, $2i$

5.1.8 Vecteurs

On déclare nos données sous forme matricielle (vecteur ligne ou matrice) dans plusieurs façons : la fonction `c()` pour un vecteur (de type ligne) :

```
c(1,2,3,4)
[1] 1 2 3 4
```

Pour un vecteur colonne, il faut utiliser la fonction `as.matrix()` :

```
as.matrix(c(1,2,3,4))
```

l'opérateur : (prioritaire sur les opérations au sein d'une expression)

```
x=1:10
x
[1] 1 2 3 4 5 6 7 8 9 10
```

La fonction `seq` de différents arguments `seq(from,to)` : `from` le début de la séquence, `to` pour la fin :

```
seq(from=1, to=10)
[1] 1 2 3 4 5 6 7 8 9 10
seq(from=1.5, to=4.7)
[1] 1.5 2.5 3.5 4.5
```

-`seq(from, to, by=)` : la même chose et `by` pour le pas ;

```
seq(from=1, to=10, by=2)
[1] 1 3 5 7 9
```

`seq(from, to, length.out=)` : il crée une séquence de `from` jusqu'au `to` de la longueur `length` :

```
seq(from=1, to=10, length=3)
[1] 1.0 5.5 10.0
```


5.1.9 Matrices

La fonction `matrix()` : remplissage par colonne par défaut :

`data` : les données pour remplir la matrice, `nrow` : le nombre de lignes et `ncol` le nombre de colonnes, `matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)`

`dimnames` : une liste de dimension 2 avec les noms des lignes et/ou colonnes

```
> matrix(data=1:6,nr=2,nc=3,dimnames=list(c("row1","row2"),c("C.1","C.2","C.3
```

5.1.10 Les listes

La liste est un objet hétérogène. C'est donc un ensemble ordonné d'objets qui ne sont pas forcément tous du même mode ni de même longueur. Les objets sont appelés composants de la liste. Remarquons que

- les composants peuvent être de mode \neq ,
- les composants peuvent avoir un nom,
- la liste n'est pas une classe.

5.1.11 Création

```
tablev<- c("Paris", "Lyon")
```

```
list <- list (1:3,ville="tablev") les composants peuvent avoir 1 nom
```

Forme une liste à 5 éléments, qui sont des vecteurs numériques de longueur 1, égaux

respectivement à 1,2,3,4,5 :

```
as.list(1:5), conversion explicite
```

Remarque 5.1 *on peut poser la question à R si l'objet est une liste*

`is.list(x)` et la réponse est TRUE ou FALSE.

Extraction

```
maliste [[2]]
```

```
#1ere composante de la liste ville
```

```
maliste ville
```

```
#2eme composante de la liste (Si la composante possede 1 nom)
```

```
maliste [[1]] [1]
```

```
#1er element de la composante 1
```

```
maliste ville[1]
```

```
#1er element de la composante "ville"
```

5.1.12 Créer un tableau de données sous R

Pour créer un tableau de données sous R, il faut utiliser la fonction `data.frame`. Cette fonction permet de concaténer des vecteurs de mêmes longueur et de modes différents.

Exemple 5.1 Saisissez deux vecteurs notés *mat* et *phy* :

```
mat<-c(19.6,17.6,18.2,16.0)
phy<-c(19.1,17.8,18.7,16.1)
```

puis construisez le tableau de données, notées, à l'aide de la fonction `data.frame` en tapant la ligne de commande suivant :

```
res<-data.frame(mat,phy)
res
```

Il est possible de donner des noms aux lignes du tableau de données avec l'option `row.names` qui doit fournir un vecteur de mode caractère et de longueur égale au nombre de lignes du tableau de données.

5.1.13 Calcul arithmétique et fonctions simples

Les opérations suivantes sont effectuées composante par composante :

- la somme des deux vecteurs avec `+` ;
- le produit avec `*` ;
- le rapport avec `/` ;
- puissance avec `^` ;
- Voici quelques fonction R simple :
 - `sum(x)` : sommes des composantes de `x` ;
 - `prod(x)` : produit des composantes de `x` ;
 - `max(x)` : maximum des composantes de `x` ;
 - `which.max(x)` : retourne l'indice du maximum des composantes de `x` ;
 - `range(x)` : idem que `c(min(x),max(x))` ;
 - `length(x)` : nombre d'éléments dans `x` ;
 - `mean(x)` : la moyenne des éléments dans `x` ;
 - `median(x)` : la médiane des éléments dans `x` ;
 - `var(x)` : la variance (corrigée) des éléments dans `x` ;
 - `cor(x)` : matrice de corrélation si `x` est un `data-frame` et 1 sinon ;
 - `cov(x,y)` : la covariance entre `x` et `y` ;
 - Le résultat est une valeur, sauf pour `range()`, `var()`, `cor()` et `cov()`.

- round(x,n) : arrondi les éléments de x à n chiffres après la virgule ;
- rev(x) : inverse l'ordre de x ;
- sort(x) : trie les éléments de x dans ordre croissante ;
- rank(x) : rangs des éléments de x ;
- cumsum(x) : un vecteur avec les sommes cumulées de composantes de x ;
- cumprod(x) : idem pour le produit ;
- table(x) : retourne un tableau avec les effectifs de différentes valeurs de x ;
- which(x==a) : retourne un vecteur des indices de x pour les quels l'opération de comparaison est vraie ;

```
x=c(2,6,10,60,34)
which(x<35)
[1] 1 2 3 5 #les indices pour les quels on a vrai la comparaison
```

5.1.14 Graphiques

Possibilité de voir des exemples de graphiques avec demo(graphics) ou demo(persp). Lorsqu'une fonction graphique est tapée sur la console, une fenêtre graphique va s'ouvrir avec le graphe demandé.

5.1.15 Partitionner une fenêtre graphique

```
par(mfcol=c(nr,nc)):
```

on partitionne la fenêtre en une matrice de nbr lignes et nbr colonnes et le remplissage est réalisé par colonne.

```
mfrow : idem mais les graphiques sont dessinés en ligne;
layout() : pour des partitions plus complexes
layout(matrix(c(1,2,3,4),2,2)) # pour insérer dans un graphique
layout(matrix(c(1,1,2,1),2,2),c(3,1),c(1,3))
layout.show(2) # et visualiser la partition créée
xlab, ylab : noms des axes, variables caractères entre " " ;
main: variable de type caractère; sub: sous-titre;
points(x,y): ajoute des points ;
lines(x,y): idem mais avec des lignes ;
segments(x0,y0,x1,y1): trace une ligne entre les points (x0,y0) et (x1,y1) ;
abline(a,b): trace une ligne de pente b et ordonnée a l'origine a ;
legend(x,y,legend): ajoute une légende au point de coordonnées (x,y) avec le:
```

5.1.16 Les fonctions

5.1.17 Structure générale pour créer des fonctions

La syntaxe générale de définition d'une fonction est la suivante :

```
nom-fonction<-function(arg1 [=expr1] ,arg2 [=expr2] ...)
```

Les accolades permettent de séparer les instructions par rapport à la signature de la fonction, les crochets, eux permettent de spécifier des valeurs par défaut des arguments de façon facultative.

Exemple 5.2 *exemple définissant la moyenne d'un vecteur :*

```
moyenne.vec <- fonction(x){s<-sum(x); # Somme des'elements de x
+ n<-length(x); # Nombre d'elements de x
+ res<-round(s/n,2); # resultat arrondi
+ return(res)
+ }
```

Par l'intermédiaire d'un éditeur de texte grâce à la fonction `fix(x)` :

```
fix(moyenne.vect)
```

Cette commande lance un éditeur de texte qui travaille avec le système R et qui permet de définir des fonctions. Le code de la fonction est écrit à partir de l'éditeur. Cette méthode est plus confortable et pratique que la première.

La fonction `return()` permet de spécifier le résultat de la fonction, lorsque l'instruction correspondante à `return` n'est pas utilisée, R retourne le résultat de la dernière expression évaluée dans la fonction.

5.1.18 Eléments de programmation

5.1.19 Les instructions de sélection

L'expression `if` :

```
if ( expression logique) < expression 1>
ou
if (expression logique)<expression1> else <expression2>
```

Si l'expression logique prend la valeur vrai (TRUE), l'expression 1 est évaluée, sinon c'est l'expression 2.

5.1.20 L'expression if else :

`ifelse (expression logique, vecteur1, vecteur2)`

Si l'expression logique prend la valeur vrai, l'expression retourne le vecteur 1, sinon elle retourne le vecteur 2. L'expression `switch()` aussi permet de réaliser des sélections.

5.1.21 Les instructions de répétition

L'expression `while` :

`while (expression logique, expression)`

Tant que l'expression logique est vraie, l'expression est exécutée. La valeur finale est celle obtenue lors de la dernière évaluation de l'expression.

5.1.22 L'expression for :

`for (ind in vecteur) expression`

`ind` est la variable boucle, `vecteur` est une expression vectorielle (souvent une séquence comme `1 :30`) et `expression` est souvent un groupe d'expressions qui composent avec la variable `ind`. `expression` est évaluée de manière répétitive tant que `ind` parcourt les valeurs de `vecteur`. L'instruction `repeat` aussi permet de réaliser des répétitions. L'instruction `break`, pour sortir d'une boucle, peut être utilisée dans toutes ces boucles et elle est la seule façon de terminer la boucle `repeat`.

L'instruction `next` peut être utilisée pour sauter une étape et passer à la suivante. Par ailleurs, R autorise la récursivité et il permet à l'utilisateur de créer ses propres opérateurs.

5.1.23 R est merveilleux !

Comme conclusion de ce chapitre, on peut dire :

1. Il y a (presque) toujours un package adapté à nos problèmes.
2. Nous pouvons programmer dans R pour compléter les fonctions.
3. Nous pouvons programmer de nouveaux packages distribuables.

5.2 Conclusion

Le principe de la simulation, au sens commun du terme, est d'utiliser un modèle, c'est-à-dire une représentation abstraite d'un système ou d'un problème, et d'étudier l'évolution de ce modèle sans faire fonctionner le système

réel. C'est par exemple le cas lorsque vous allez à la banque pour un prêt : on vous dit combien vous payeriez chaque mois si vous acceptiez l'offre, sans pour autant avoir à le faire nécessairement. Modéliser consiste donc à poser un problème sur papier ou sur ordinateur afin de l'étudier donc l'objectif de ce travail est de présenter les principales commandes R associées aux lois de probabilités et simulations de variables aléatoires réelles (var).

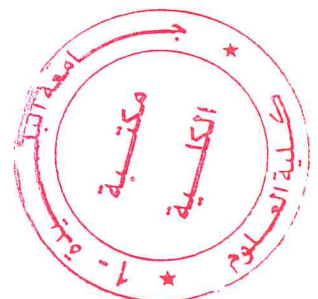
Les méthodes de simulation, conçues pour être utilisées en statistique et en recherche opérationnelle, ont connu et connaissent encore un développement rapide dû l'extraordinaire évolution des ordinateurs. Des applications se rencontrent tant dans l'industrie qu'en économie, ou encore en sciences sociales, en physique des particules, en astronomie et dans de nombreux autres domaines.

La simulation de Monte-Carlo est une méthode d'estimation d'une quantité numérique qui utilise des nombres aléatoires, elle présente le double avantage d'être simple d'utilisation et de pouvoir être appliquée à un très large éventail de problèmes. Elle est utilisée en finance pour déterminer quand lever une option sur un bien financier ; en assurance pour évaluer le montant d'une prime ; en biologie pour étudier les dynamiques intra et intercellulaires ; en physique nucléaire pour connaître la probabilité qu'une particule traverse un écran ; en télécommunications pour déterminer la qualité de service ou de façon générale pour déterminer la fiabilité d'un système, sa disponibilité ou son temps moyen d'atteinte de la défaillance.

Je termine ce mémoire et cette conclusion, on répondant à la question suivante :

Pourquoi faire des simulations ?

- Pour « vérifier » à l'aide de l'ordinateur un résultat mathématique déjà connu.
- Pour « démontrer » à l'aide de l'ordinateur un résultat que l'on ne parvient pas à démontrer mathématiquement.
- Cela peut nous guider dans la démonstration d'un résultat mathématique difficile.
- C'est souvent exigé lorsque l'on essaye de publier un papier dans une revue.



Bibliographie

- [1] Asaph Hall III. (1873). On an experimental determination of π . The Messenger of Mathematics, 2 :113–114.
- [2] Bertrand F., Bertrand M.M. (2010). *Initiation à la Statistique avec R*, Dunod, Paris.
- [3] Caflisch R. E. (1998). *Monte Carlo and quasi-Monte Carlo methods*, Acta Numerica vol. 7, Cambridge University Press, pp. 1-49.
- [4] Han [5]
- [5] P. Glasserman and J. Li. (2005). Importance sampling for portfolio credit risk. Management Science, 51(11) :1643–1656.
- [6] P. W. Glynn. Importance sampling for Monte Carlo estimation of quantiles. In Proceedings of the Second International Workshop on Mathematical Methods in Stochastic Simulation and Experimental Design, pages 180–185, 1996.
- [7] Goga C, Labruère C. (2009). *Introduction au logiciel R*, Ecole Doctorale Dijon. http://math.u-bourgogne.fr/IMB/goga/formation_doc_R_2009.pdf.
- [8] Lafaye de Micheaux P., Drouilhet R., Liqueur B. (2011). *Le logiciel R : Maîtriser le langage, Effectuer des analyses statistiques*. Springer-Verlag, France.
- [9] Lalanne C. Pallier C. (2011). *Introduction aux Statistiques et à l'utilisation du logiciel R*. <http://www.pallier.org/ressources/tp.stats/tp1/tp1.pdf>.
- [10] Legendre P. (2012). *Introduction au langage statistique R*. http://biol09.biol.umontreal.ca/BIO6077/Introduction_R.pdf.
- [11] D. P. Kroese, T. Taimre, and Z. I. Botev. Handbook of Monte Carlo Methods. John Wiley & Sons, New York, 2011.
- [12] Hammersley, J.M. & Handscomb, D.C. (1964), Monte Carlo methods, Wiley, New York.

- [13] Hammersley, J.M. & Mauldon, J.G. (1956), General principles of anti-thetics variates, Proc. Cambridge Philos. Soc., 52, 476-481.
- [14] P Artzner, F Delbaen, JM Eber, D Heath - *Mathematical finance*, 1999.
- [15] R. Korn, E. Korn & G. Kraisandt, (2010) *Monte Carlo Methods and Models in Finance and Insurance*, Taylor & Francis.