Algorithmes randomisés

Nicolas Gast

Inria – Univ. Grenoble Alpes

2020. Maths701, Master MEEF.

Plan du cours

1. Rappel de Probabilités
   - Événements et espaces probabilisés
   - Variables aléatoires
   - Inégalités de concentration, intervales de confiance
2. Génération d'aléatoire et simulation
   - Comment générer de l'aléatoire?
   - Générer selon une loi donnée
3. Algorithmes randomisés
   - Algorithmes "Las Vegas": Quicksort / Quickselect
   - Algorithmes "Monte Carlo"
   - Algorithmes décentralisés / apprentissage par renforcement
4. Conclusion

Espace probabilisé

Un espace probabilité est $(\Omega, \mathcal{F}, \mathbb{P}[])$, où

- Ω est l'ensemble des possibles.
- $\mathcal{F} \subset 2^\Omega$ est l'ensemble des événements *mesurables*.
- $\mathbb{P}[] : \mathcal{F} \to [0,1]$ est une mesure de probabilité.

$$\mathbb{P}[E] = \text{ probabilité de l'événement } E.$$

Espace probabilisé

Un espace probabilité est $(\Omega, \mathcal{F}, \mathbb{P}[])$, où

- Ω est l'ensemble des possibles.
- $\mathcal{F} \subset 2^\Omega$ est l'ensemble des événements *mesurables*.
- $\mathbb{P}[] : \mathcal{F} \to [0,1]$ est une mesure de probabilité.

$$\mathbb{P}[E] = \text{ probabilité de l'événement } E.$$

Exemple: on lance deux dés:

$$\Omega = \{$$

Indépendance et probabilité conditionnelle

Soit $A, B \in \mathcal{F}$ deux événements. A et B sont indépendents si $\mathbb{P}[A \cap B] = \mathbb{P}[A]\,\mathbb{P}[B]$.

Ex:
Si $A \in \mathcal{F}$ is un événement de probabilité non nulle, alors:

$$\mathbb{P}[B \mid A] = \frac{\mathbb{P}[B \cap A]}{\mathbb{P}[A]}$$

Ex: lancé de deux dés.

Règle de Bayes

Bayes rule

$$\mathbb{P}[B|A] = \frac{\mathbb{P}[A|B]\,\mathbb{P}[B]}{\mathbb{P}[A]}.$$

Exercice

Example: Une maladie a touché 10% de la population. Le laboratoire Testex a créé un nouveau test qu'il permet de savoir si on a été infecté ou non. Ce test est censé être fiable à 95% (c'est à dire qu'il donne un résultat juste 95% du temps.

Question – Un individu est testé positif par le test. Quelle est la probabilité qu'il soit malade?

Table des matières

Variables aléatoire

Soit $\mathcal{X}$ un sous-ensemble de $\mathbb{R}$ (typiquement, $\mathcal{X} = \mathbb{R}$ ou $\mathcal{X}$ est fini).

Une variable aléatoire à valeur dans $\mathcal{X}$ est une fonction mesurable, $X : \Omega \to \mathcal{X}$. On note

$$\mathbb{P}[X \in A] = \mathbb{P}[\{\omega : X(\omega) \in A\}].$$

En règle général, on n'écrira pas $X(\omega)$ mais uniquement X.

Exemple: lancés de dés.

Espérance, variance

Soit X une variable aléatoire à valeur dans un ensemble fini $\mathcal{X}$.
L'espérance de X, notée $\mathbb{E}[X]$ est:

$$\mathbb{E}[X] =$$

La variance de X est:

Loi usuelles

	Définition	Espérance	Variance
Loi de Bernoulli			
Loi géométrique			
Loi binomiale			
Loi de Poisson			

Variables aléatoires indépendantes

Deux variables sont indépendantes si

$$\mathbb{P}[X \in A \cup Y \in B] = \mathbb{P}[X \in A]\,\mathbb{P}[X \in B]\,.$$

Exemple: lancés de dés.

Espérance, variance et indépendance

Soit X et Y deux variables aléatoires:

- $\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$.
- Si X et Y sont indépendantes, alors $\mathrm{var}[X+Y] = \mathrm{var}[X] + \mathrm{var}[Y]$.

Exemple: loi de Bernoulli.

Table des matières

Inégalité de Markov et de Chebychev

Soit X une variable aléatoire.

$$\mathbb{P}[X \geq a] \leq \frac{1}{a}\mathbb{E}[|X|]$$

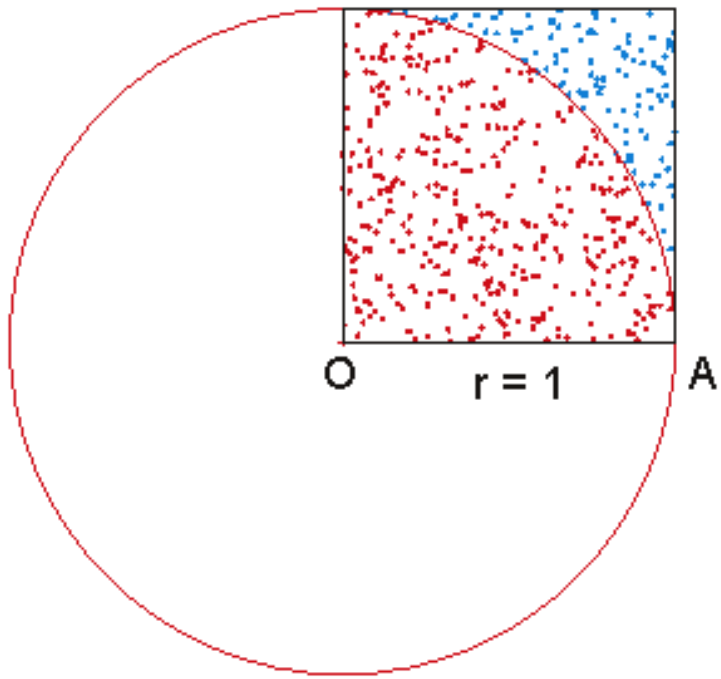
$$\mathbb{P}[|X - \mathbb{E}[X]| \geq a] \leq \frac{1}{a^2}\mathrm{var}[|X|]$$

Application: loi des grands nombres

Soit $X_1, X_2, \dots$ une suite de variable aléatoire indépendantes et à valeurs dans $\mathcal{X}$ fini. Soit $S_n = \frac{1}{n}\sum_{k=1}^{n} X_k$. Alors:

$$\forall \epsilon > 0 : \qquad \lim_{n\to\infty} \mathbb{P}\left[|S_n - \mathbb{E}[X]| \geq \varepsilon\right] = 0.$$

Illustration: Estimation de π.



Confiance?

On dispose d'une pièce biaisée $\mathbb{P}[X_i = 1] = 1 - \mathbb{P}[X_i = 0] = p$.

- On cherche à estimer p.

Combien de tirages sont nécessaire?

Confiance?

On dispose d'une pièce biaisée $\mathbb{P}[X_i = 1] = 1 - \mathbb{P}[X_i = 0] = p$.

- On cherche à estimer p.

Combien de tirages sont nécessaire?

$X = (0,$

Confiance?

On dispose d'une pièce biaisée $\mathbb{P}[X_i = 1] = 1 - \mathbb{P}[X_i = 0] = p$.

- On cherche à estimer p.

Combien de tirages sont nécessaire?

$X = (0, 1, 0, 1, 0, 1,$

Confiance?

On dispose d'une pièce biaisée $\mathbb{P}[X_i = 1] = 1 - \mathbb{P}[X_i = 0] = p$.

- On cherche à estimer p.

Combien de tirages sont nécessaire?

$X = (0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,$

Confiance?

On dispose d'une pièce biaisée $\mathbb{P}[X_i = 1] = 1 - \mathbb{P}[X_i = 0] = p$.

- On cherche à estimer p.

Combien de tirages sont nécessaire?

$X = (0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1,$
$1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0,$
$1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1,$
$0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1,$
$0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, \ldots)$

Confiance?

On dispose d'une pièce biaisée $\mathbb{P}[X_i = 1] = 1 - \mathbb{P}[X_i = 0] = p$.

- On cherche à estimer p.

Combien de tirages sont nécessaire?

$X = ($0, 1, 0, 1, 0, 1,1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1,. . .)

	1	3	5				
$\frac{1}{n}\sum_{i=1}^{n} X_i$	0.0	0.333. . .	.4				

Confiance?

On dispose d'une pièce biaisée $\mathbb{P}[X_i = 1] = 1 - \mathbb{P}[X_i = 0] = p$.

- On cherche à estimer p.

Combien de tirages sont nécessaire?

$X = ($0, 1, 0, 1, 0, 1,1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1,. . .)

	1	3	5	10	20	100	
$\frac{1}{n}\sum_{i=1}^{n} X_i$	0.0	0.333. . .	.4	0.7	0.7	0.57	

Confiance?

On dispose d'une pièce biaisée $\mathbb{P}[X_i = 1] = 1 - \mathbb{P}[X_i = 0] = p$.

- On cherche à estimer p.

Combien de tirages sont nécessaire?

$X = (0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, \ldots)$

	1	3	5	10	20	100	1000
$\frac{1}{n}\sum_{i=1}^{n} X_i$	0.0	0.333...	.4	0.7	0.7	0.57	0.606

Confiance?

On dispose d'une pièce biaisée $\mathbb{P}[X_i = 1] = 1 - \mathbb{P}[X_i = 0] = p$.

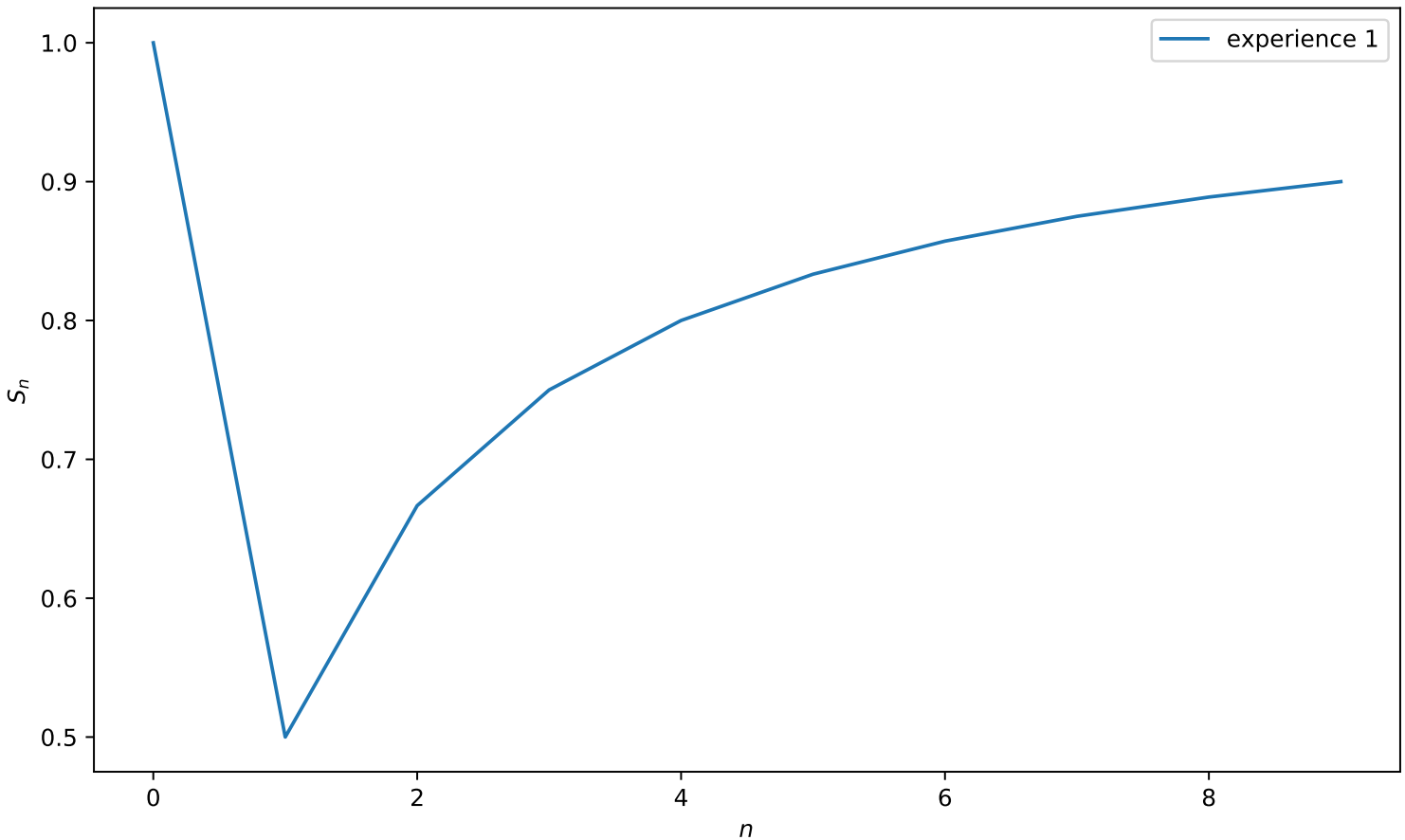
- On cherche à estimer p.

Combien de tirages sont nécessaire?

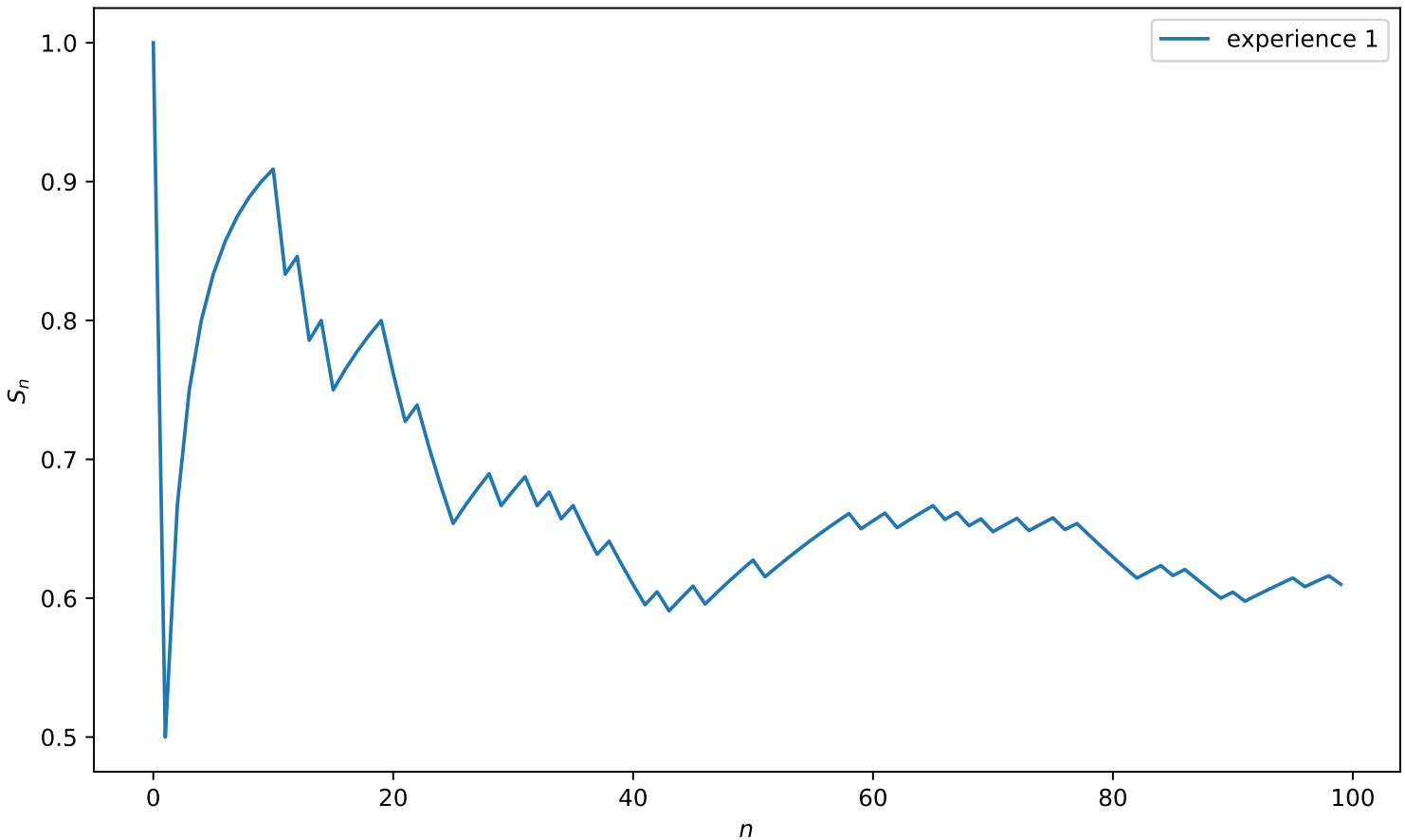
$X = (0, 1, 0, 1, 0, 1,1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1,$
$1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0,$
$1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,0, 0, 1, 1, 1, 1, 0, 0,$
$1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1,$
$0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1,$
$0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1,\ldots)$

	1	3	5	10	20	100	1000
$\frac{1}{n}\sum_{i=1}^{n} X_i$	0.0	0.333...	.4	0.7	0.7	0.57	0.606
	0.0	0.666...	0.8	0.7	0.65	0.71	0.62
	1.0	0.666...	0.6	0.4	0.55	0.6	0.597
	0.0	0.0	0.2	0.3	0.45	0.53	0.574
	0.0	0.333...	0.2	0.5	0.6	0.59	0.592
	0.0	0.333...	0.4	0.6	0.55	0.6	0.623

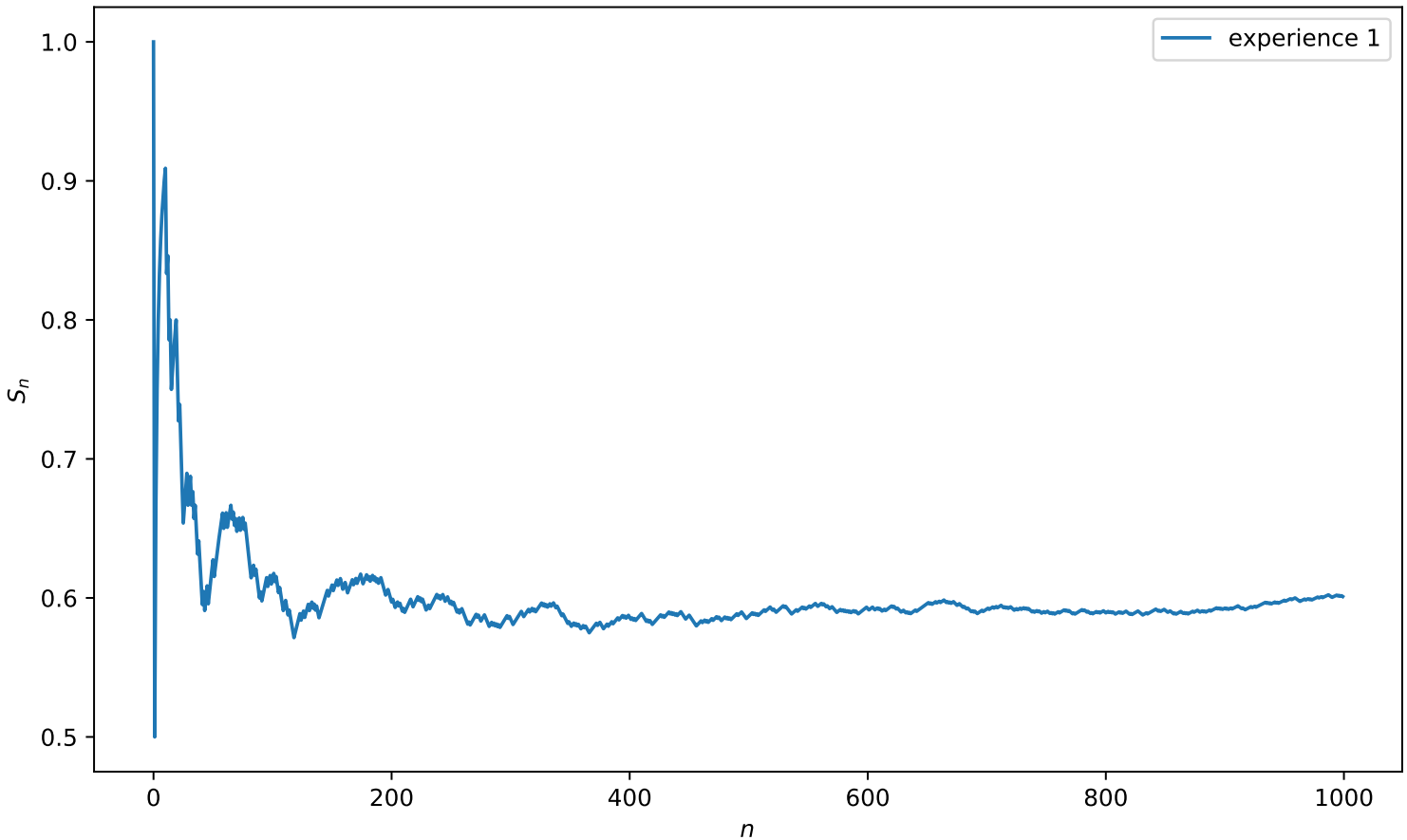
Étudions S_n



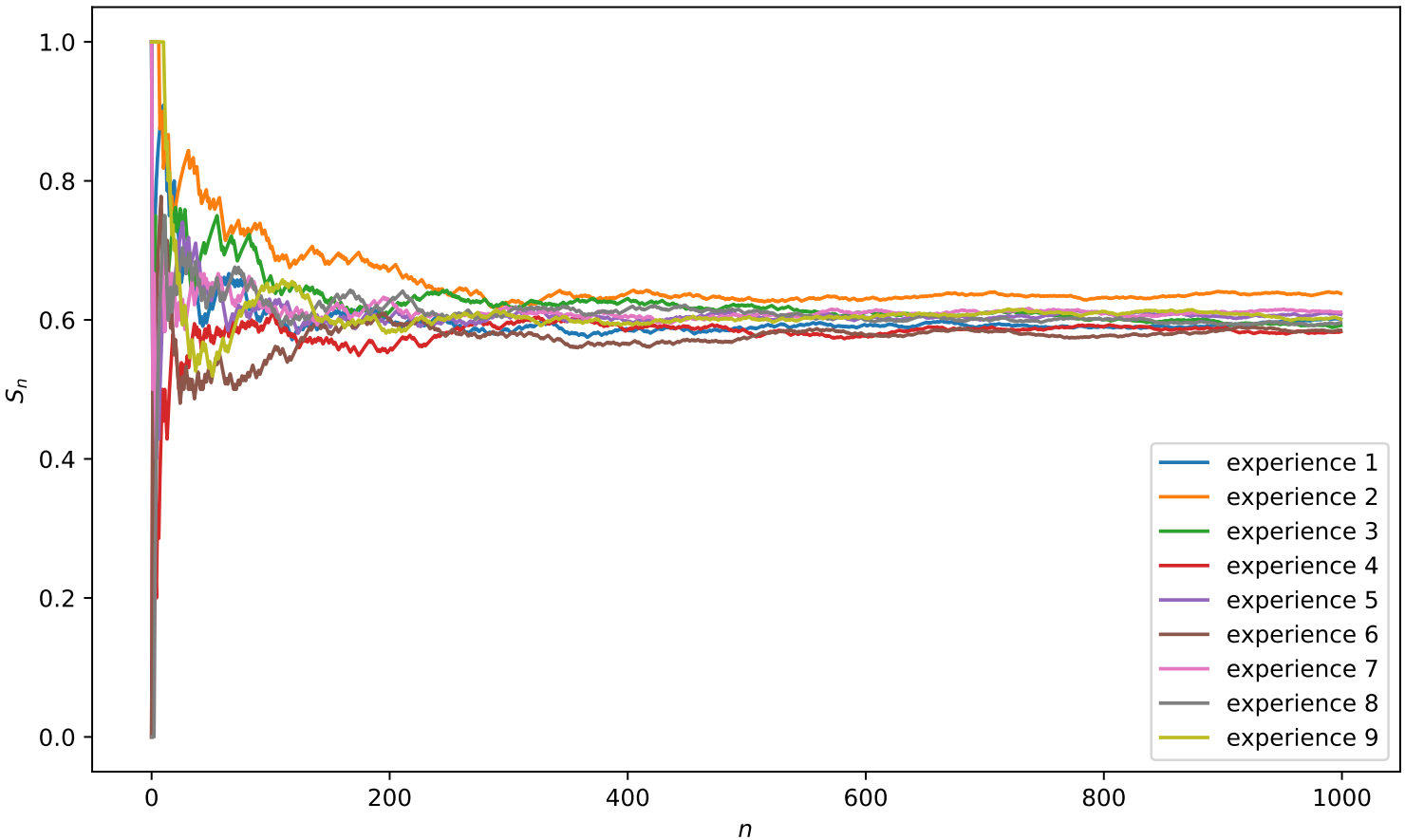
Étudions S_n



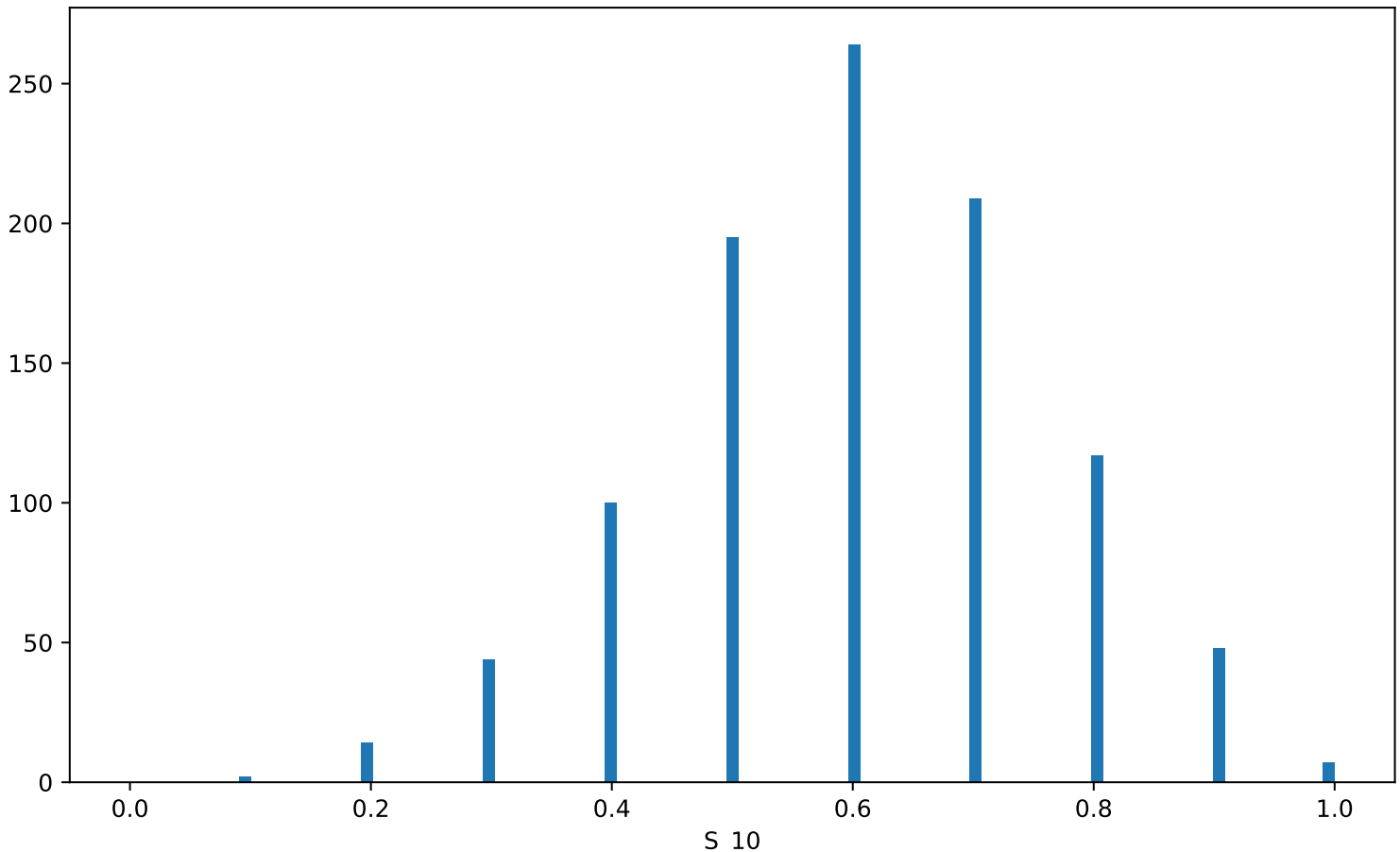
Étudions S_n



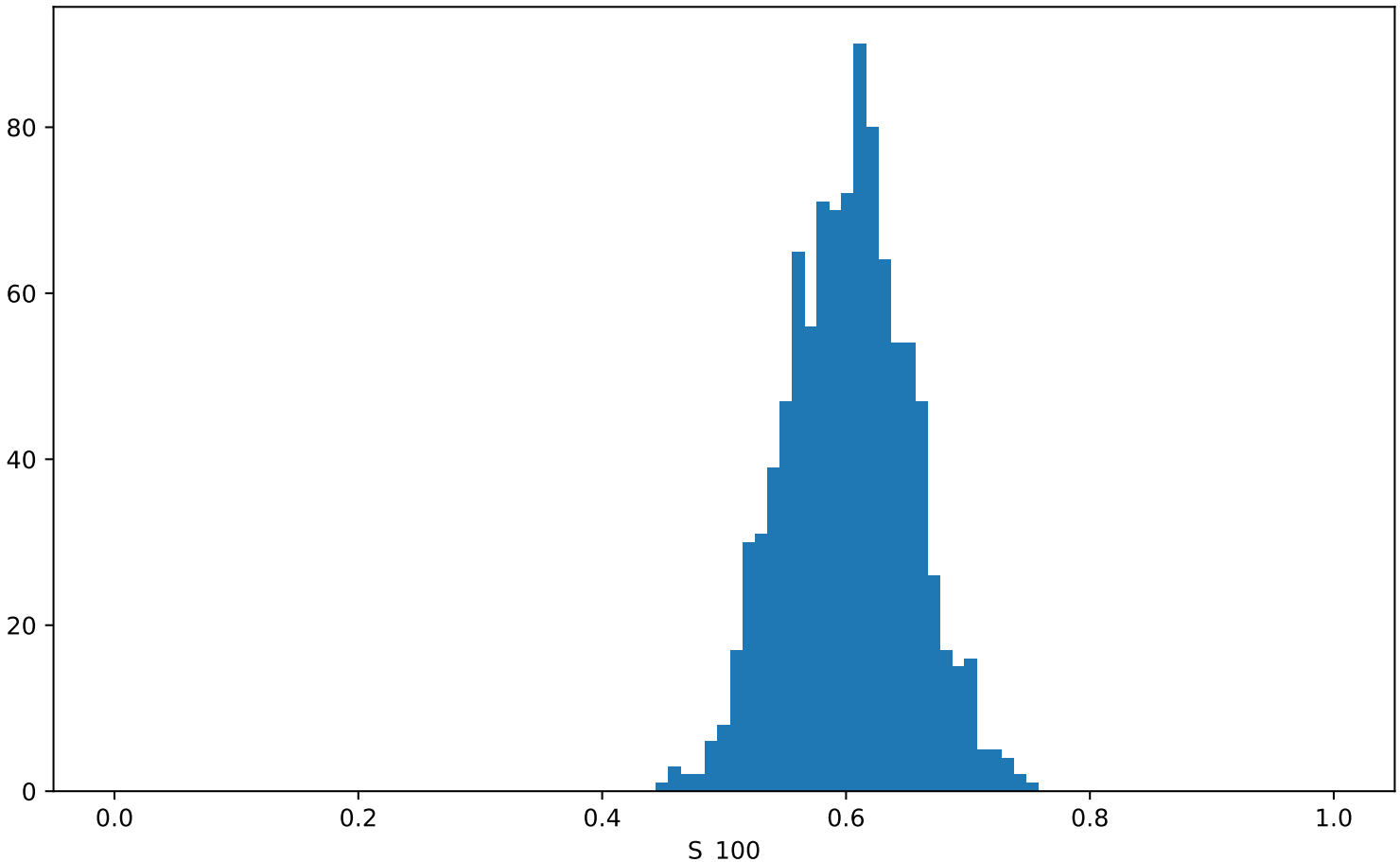
Étudions S_n


1.0
0.8
0.6
0.4
0.2
0.0
S_n
0
200
400
600
800
1000
n
experience 1
experience 2
experience 3
experience 4
experience 5
experience 6
experience 7
experience 8
experience 9

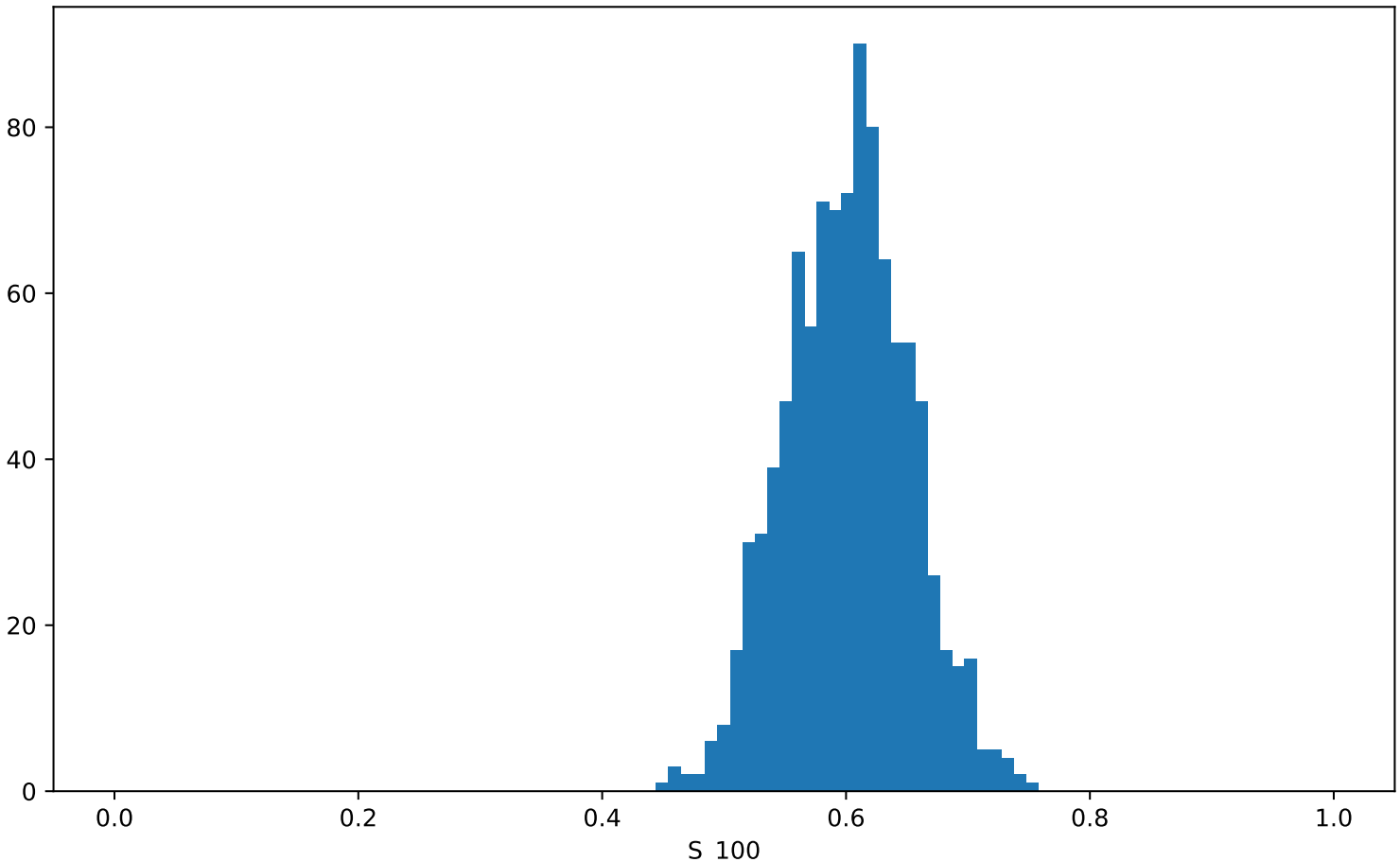
Distribution de S_n pour $n \in \{10, 100, 1000\}$



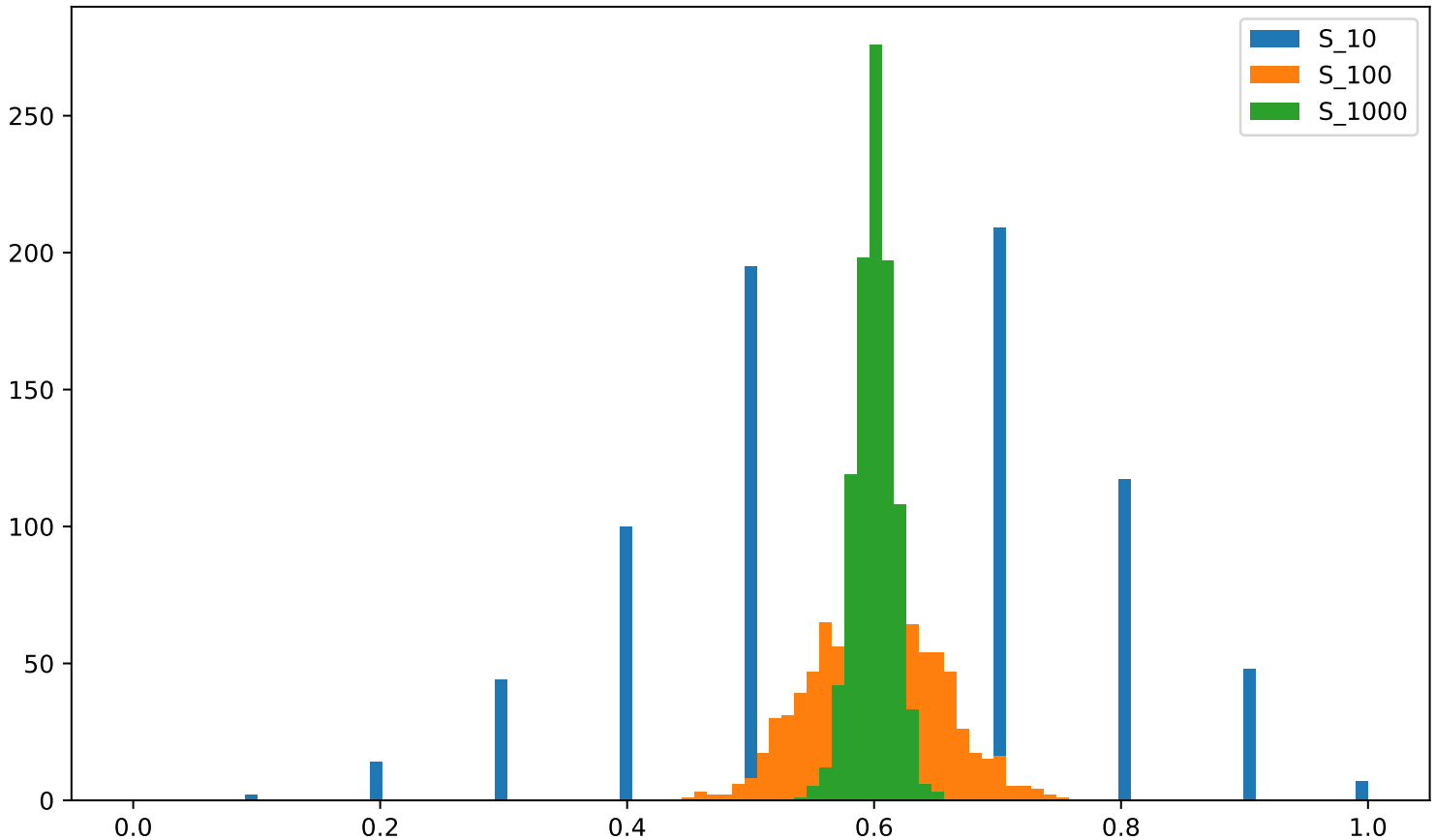
Distribution de S_n pour $n \in \{10, 100, 1000\}$



Distribution de S_n pour $n \in \{10, 100, 1000\}$

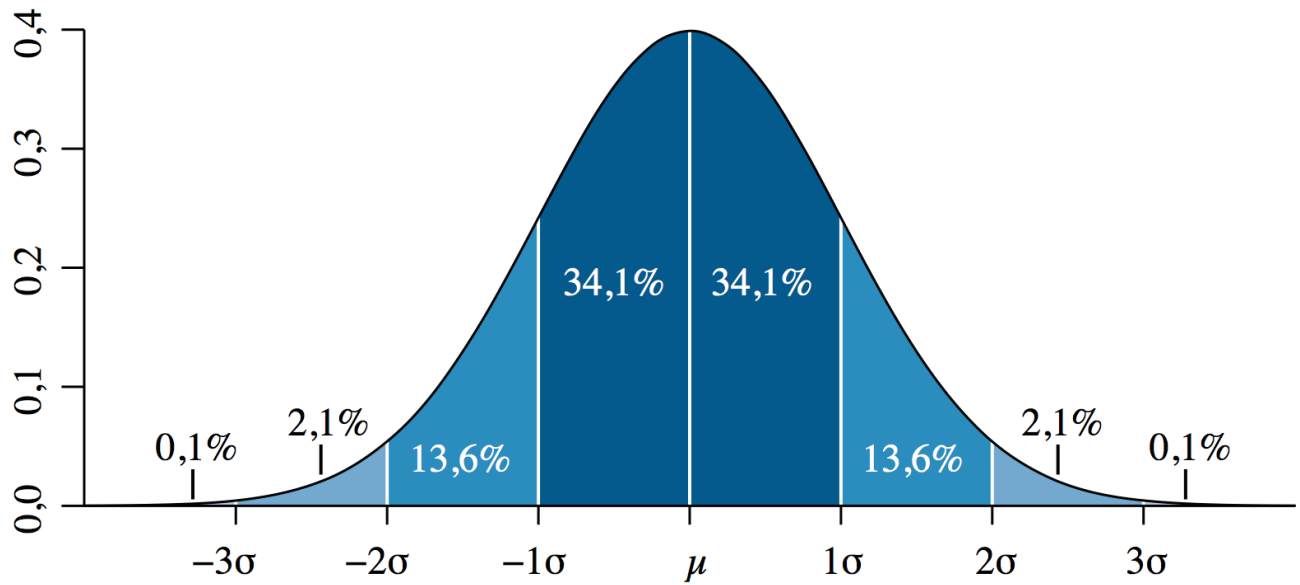


Distribution de S_n pour $n \in \{10, 100, 1000\}$



Loi normale

Densité: $f(x) = \frac{1}{\sigma\sqrt{2\pi}}\ \mathrm{e}^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$.



Théorème central limite

Soit $X_1, X_2, X_3, \ldots$ une suite de variable aléatoires *i.i.d.* de moyenne μ et de variance σ^2. Alors:

$$\frac{1}{\sigma\sqrt{n}} \sum_{k=1}^{n} (X_k - \mu) \to N(0,1) \qquad \text{en loi}$$

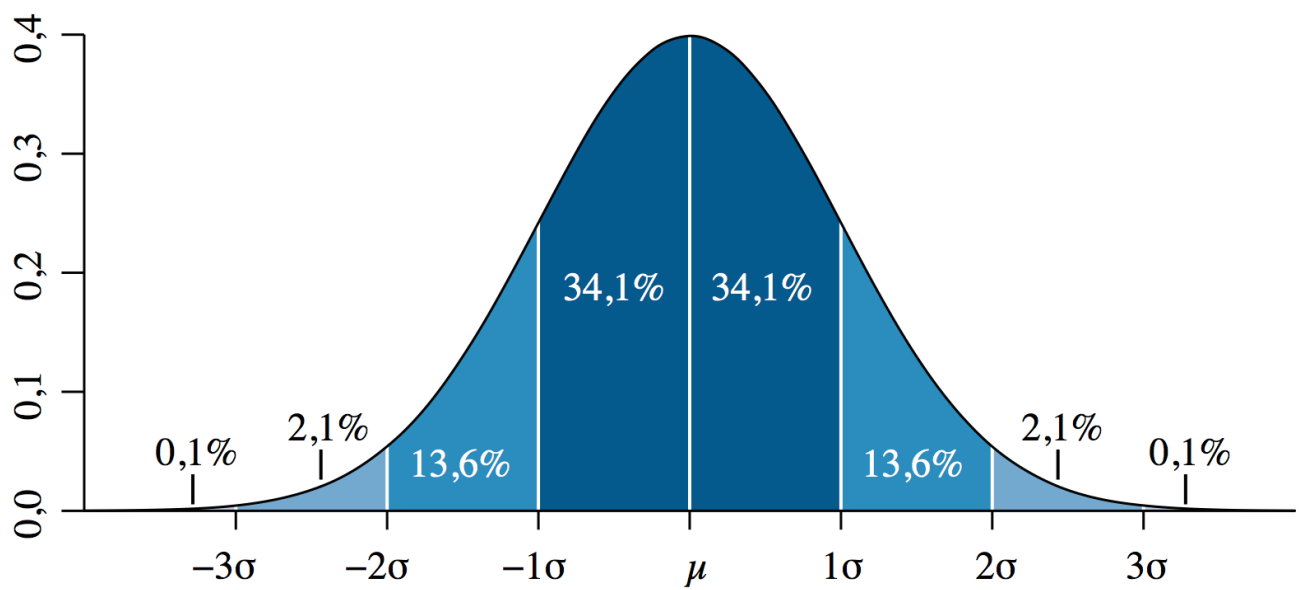
Historique de la démonstration: de Moivre 1733 (pour une pièce non biaisée), Laplace 1812 (pour les pièces biaisées), Pólya 1920 (cas général)

Intervalle de confiance

$$\frac{1}{n}\sum_{k=1}^{n} X_k \approx \mu + \frac{1}{\sqrt{n}}\sigma.$$

En particulier:

$$\mathbb{P}\left[\frac{1}{n}\sum_{k=1}^{n} X_k \in [\mu - \frac{2}{\sqrt{n}}\sigma; \mu + \frac{2}{\sqrt{n}}\sigma]\right] \geq 95\%.$$



Quel σ utiliser? Cas des variables sont de Bernoulli

- $\sigma = \sqrt{p(1-p)} \leq 1/2$. Application: sondage et confiance à 3 points.

Quel σ utiliser? Cas des variables sont de Bernoulli

- $\sigma = \sqrt{p(1-p)} \leq 1/2$. Application: sondage et confiance à 3 points.

- Que faire si $\bar{\sigma} = 0$? Vous testez n personnes et aucune n'est positive. Pouvez vous estimer la proportion de positifs?

Quel σ utiliser? Cas des variables sont de Bernoulli

- $\sigma = \sqrt{p(1-p)} \leq 1/2$. Application: sondage et confiance à 3 points.

- Que faire si $\bar{\sigma} = 0$? Vous testez n personnes et aucune n'est positive. Pouvez vous estimer la proportion de positifs?
  Réponse, avec grande probabilité:

$$0 = \bar{\mu} \geq p - \frac{2}{\sqrt{n}}\sqrt{p(1-p)} \approx p - 2\sqrt{\frac{p}{n}}.$$

  Ce qui donne la règle des $4n$ (parfois appelée la règle est $3n$ ou règle des $3.7n$)

$$p \leq \frac{4}{n}.$$

Cas plus complexe: Fiabilité d'un vaccin

Pfizer annonce un vaccin efficace à 90% puis Moderna a 94.4% puis Pfizer à 95%. Qui gagne et de combien?

Cas plus complexe: Fiabilité d'un vaccin

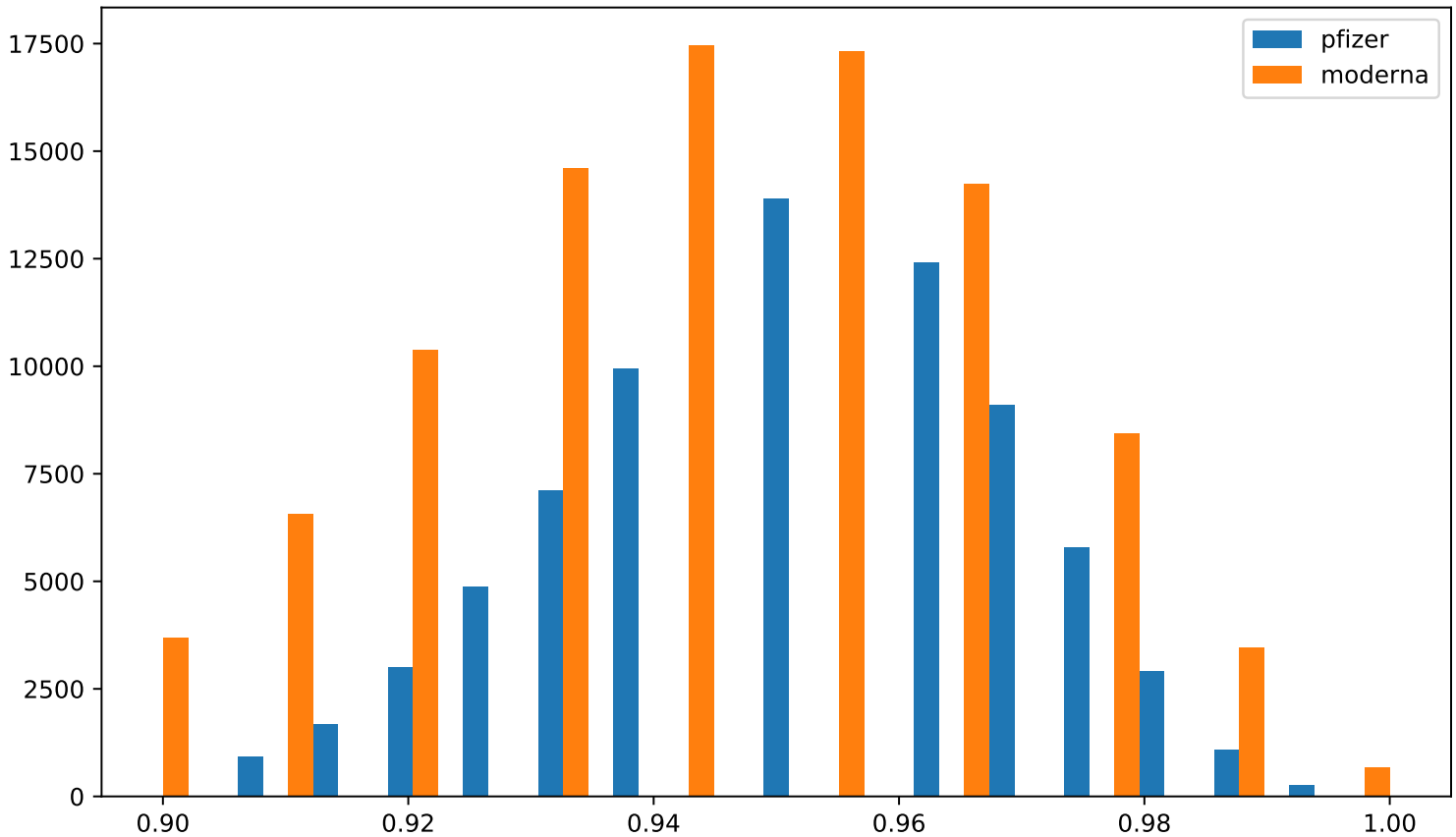
Pfizer annonce un vaccin efficace à 90% puis Moderna a 94.4% puis Pfizer à 95%. Qui gagne et de combien?

	Placebo	Vaccinés
Pfizer	8/20500	162/20500
Moderna	5/30000	90/30000

Idée: $1 - 8/162 \approx 0.951$, $1 - 5/90 \approx 0.944$.
Ces chiffres sont ils estimés de façon précises?

Un peu de simulation sur l'efficacité des vaccins

Un peu de simulation sur l'efficacité des vaccins



Confidence intervals:

	90%	95%
Moderna	0.900-0.978	0.889-0.989
Pfizer	0.920-0.975	0.914-0.981

Plan du cours

1. Rappel de Probabilités
   - Événements et espaces probabilisés
   - Variables aléatoires
   - Inégalités de concentration, intervales de confiance
2. Génération d'aléatoire et simulation
   - Comment générer de l'aléatoire?
   - Générer selon une loi donnée
3. Algorithmes randomisés
   - Algorithmes "Las Vegas" : Quicksort / Quickselect
   - Algorithmes "Monte Carlo"
   - Algorithmes décentralisés / apprentissage par renforcement
4. Conclusion

Quelle source d'aléatoire?

On utilise une suite de variables aléatoire *i.i.d.* (indépendantes et identiquement distribuées).

Questions:

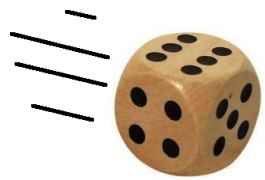
- Comment générer une suite de v.a. $X_i \sim \mathrm{Unif}[0,1]$ indépendantes?
- Comment utiliser des variables uniformes pour en générer d'autres.

Un ordinateur est-il aléatoire?

"*Dieu ne joue pas aux dés*"

Albert Einstein

Le hasard sert à modéliser ce qu'on ne connaît pas exactement.



`/dev/random` v.s. `/dev/urandom`)

En pratique, on n'utilise des générateurs pseudo-aléatoires

Exemple: congruence linéaire

$$x_0 = \text{graine}$$
$$x_{n+1} = a x_n \bmod m$$

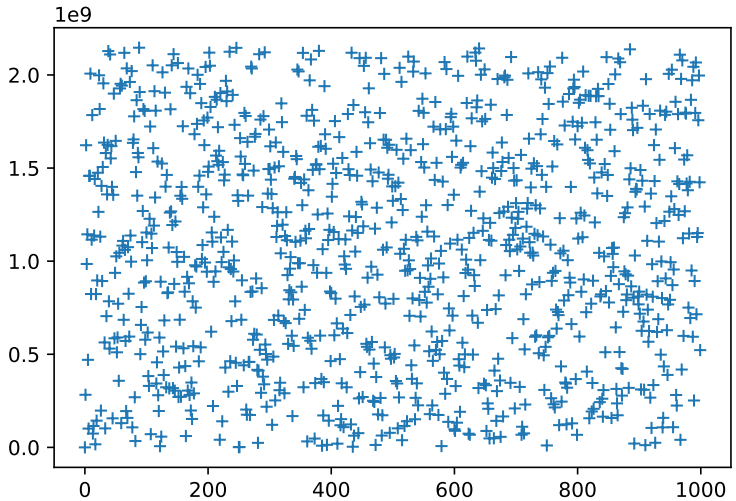
Sur ma machine (fonction $\mathrm{rand}()$ de la libc): $a = 16807, m = 2^{31} - 1$.

Est-ce que cela a l'air aléatoire?

Tests basics

```cpp
#include<cstdlib>
#include<iostream>

int main() {
  srand(1);
  for(int i=0;i<1000;i++)
    std::cout << rand() << "\n";
}
```



Simple plot : OK

Autocorrélation:

OK, répartition: OK.

Période d'un générateur pseudo-aléatoire

Cette source a l'air aléatoire mais:

$$x_{n+1} = a x_n \bmod m$$

$x_n = x_0$ implique que $x_{n+1} = x_1$, etc.

On appelle le plus petit n telle que $x_n = x_0$ la **période** du générateur aléatoire.

```
#include<cstdlib>
#include<iostream>

#define TWO_TO_31 2147483648

int main() {
  srand(42);
  std::cout << rand() << "\n";
  for (long long int i=0;i<TWO_TO_31-3;i++) rand();
  std::cout << rand() << "\n";
}
```

Après < 15secondes:

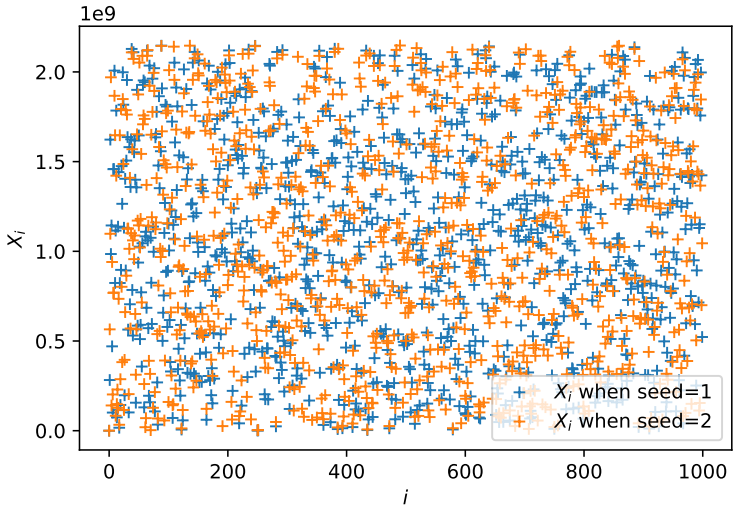
```
705894
705894
```

Problème des graines parallèles

$x_0 =$ graine

$x_{n+1} = ax_n \bmod m$

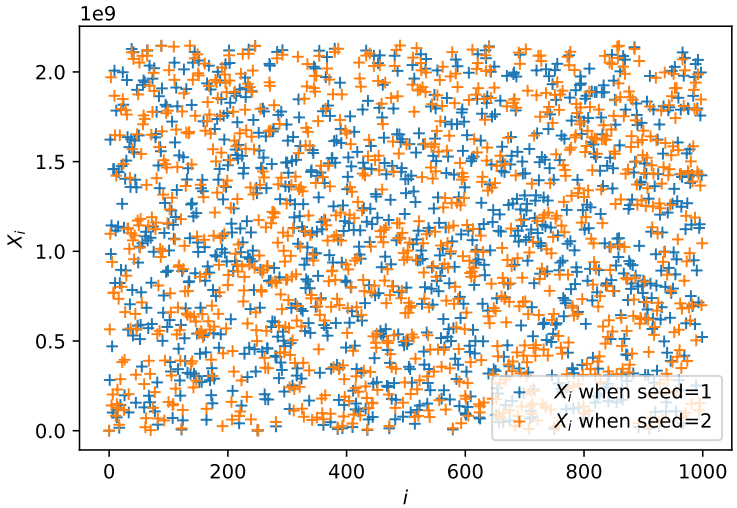
Graine=1
Graine=2



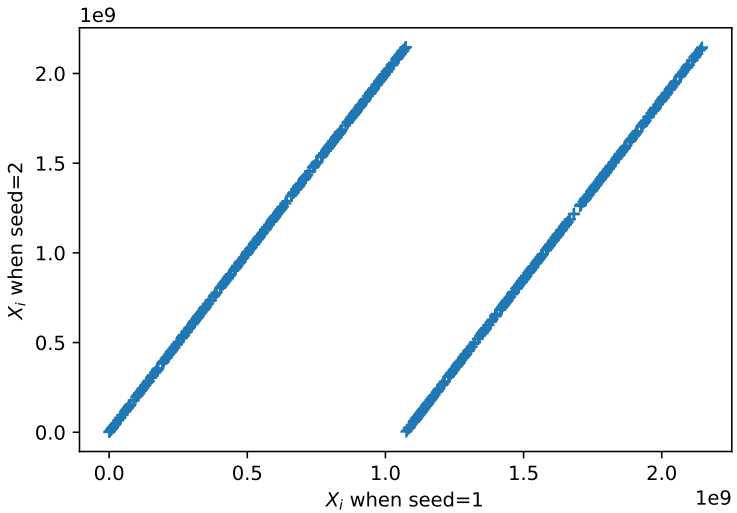
Problème des graines parallèles

$x_0 =$ graine $\qquad x_{n+1} = ax_n \bmod m$

Graine=1
Graine=2



Graine=2



Graine=1

Take-home message

Lors de l'utilisation d'un générateur pseudo-aléatoire:

- Attention à la période
- Le générateur utilise une graine.
  - Toujours sauver la graine (pour pouvoir reproduire la même simulation)
  - Pour des réplications indépendantes, utiliser des graines différentes.

Table des matières

Comment générer une v.a. selon une distribution donnée?

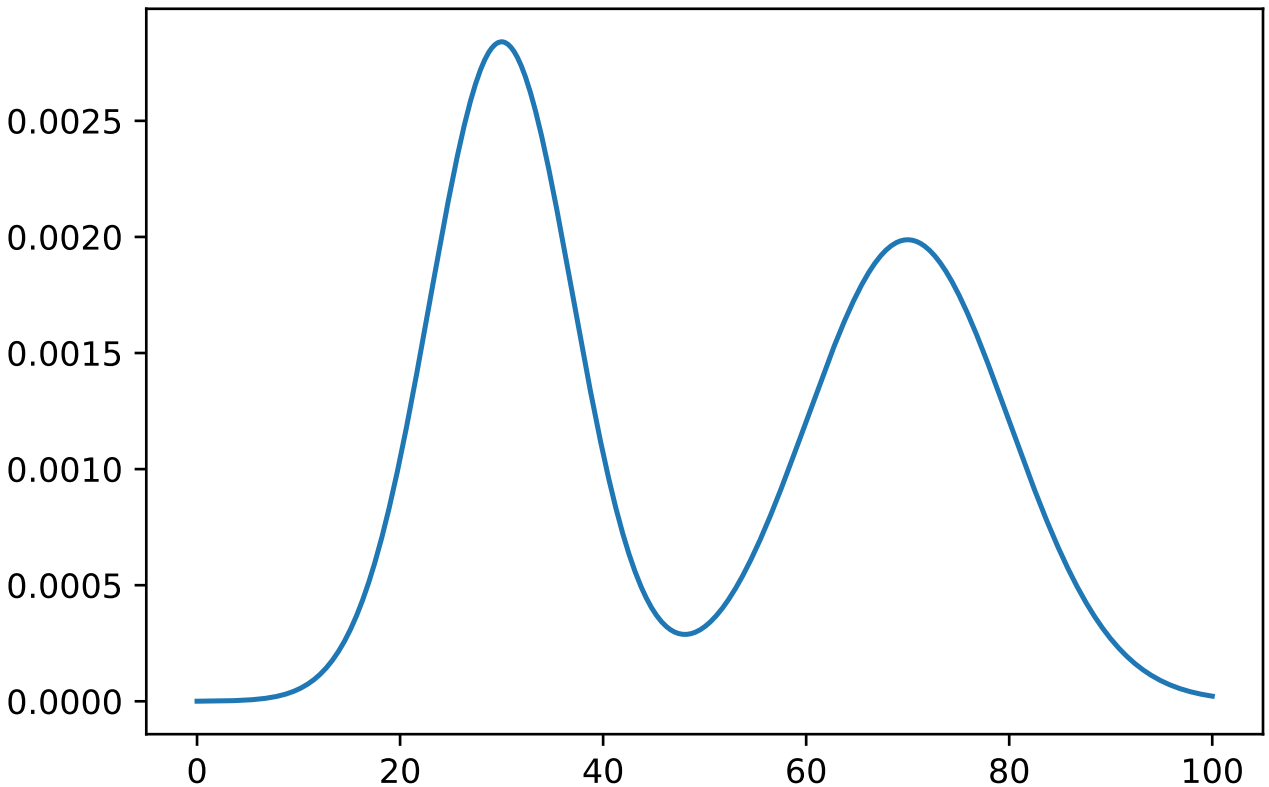
On se donne $U \sim \mathit{Unif}([0,1])$.

Exemple 1: Comment générer X tel que $\mathbb{P}[X = i] = \pi_i$, avec $\pi_1 = 6/20, \pi_2 = 4/20, \pi_3 = 3/20, \pi_4 = \pi_5 = 2/20, \pi_6 = \pi_7 = \pi_8 = 1/20$

Comment générer une v.a. selon une distribution donnée?

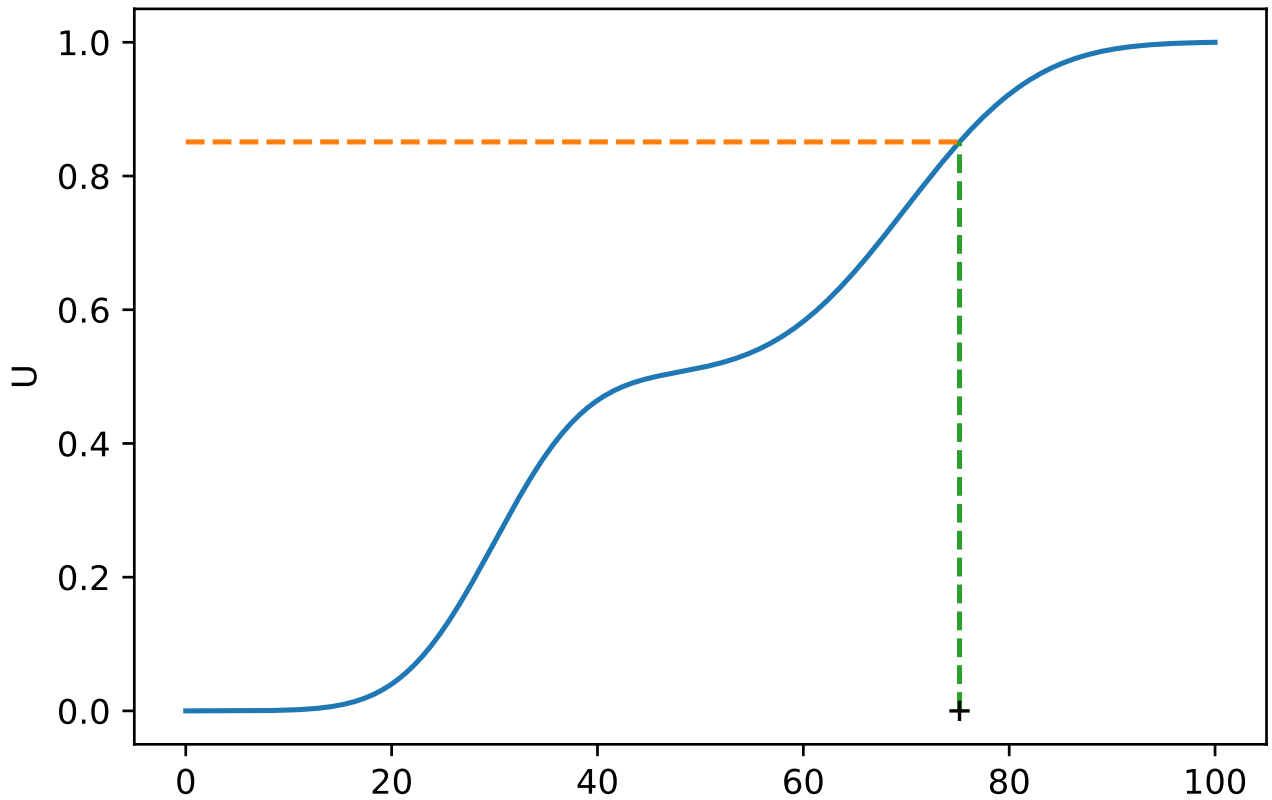
On se donne $U \sim Unif([0,1])$.

Exemple 2:



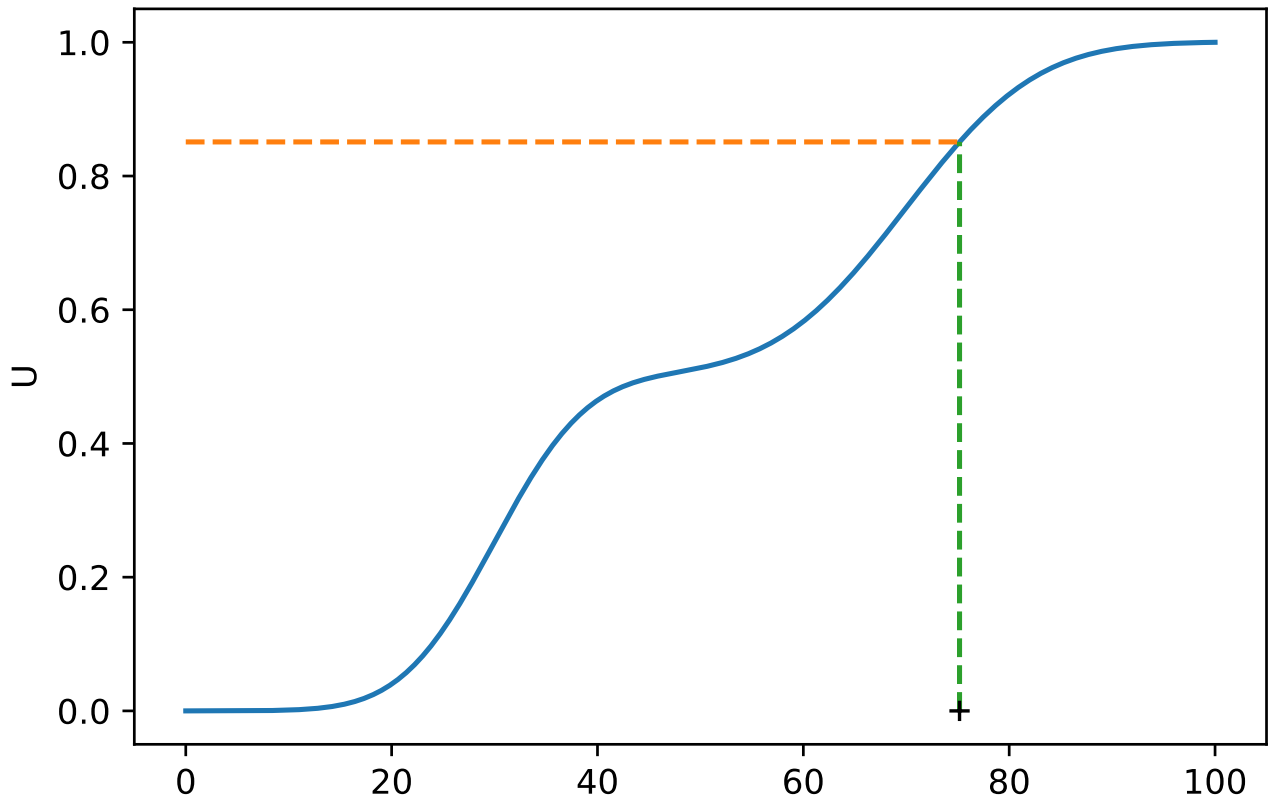
Méthode de la fonction inverse

Rappel, la fonction cumulatrice de distribution F est une fonction $F : \mathbb{R} \to [0, 1]$ telle que $F(x) = \mathbb{P}[X \leq x]$.



Méthode de la fonction inverse

Rappel, la fonction cumulatrice de distribution F est une fonction $F : \mathbb{R} \to [0, 1]$ telle que $F(x) = \mathbb{P}[X \leq x]$.

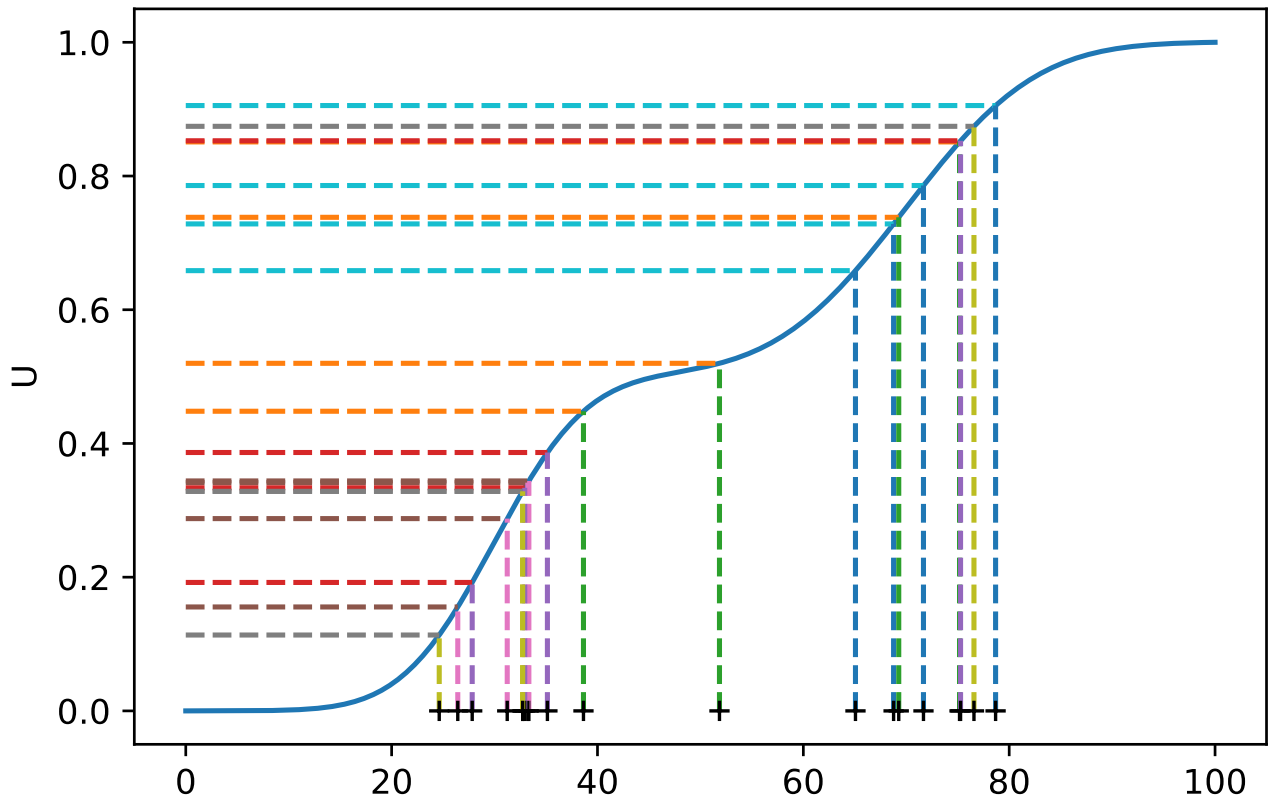


Algorithme: Générer $U \sim Unif([0, 1])$ et rendre

$$F^{-1}(U) = \sup\{x \mid F(x) \leq U\}.$$

Méthode de la fonction inverse

Rappel, la fonction cumulatrice de distribution F est une fonction $F : \mathbb{R} \to [0, 1]$ telle que $F(x) = \mathbb{P}[X \leq x]$.



Algorithme: Générer $U \sim Unif([0, 1])$ et rendre

$$F^{-1}(U) = \sup\{x \mid F(x) \leq U\}.$$

Quiz

Qu'est-ce que l'algorithme suivant génère?

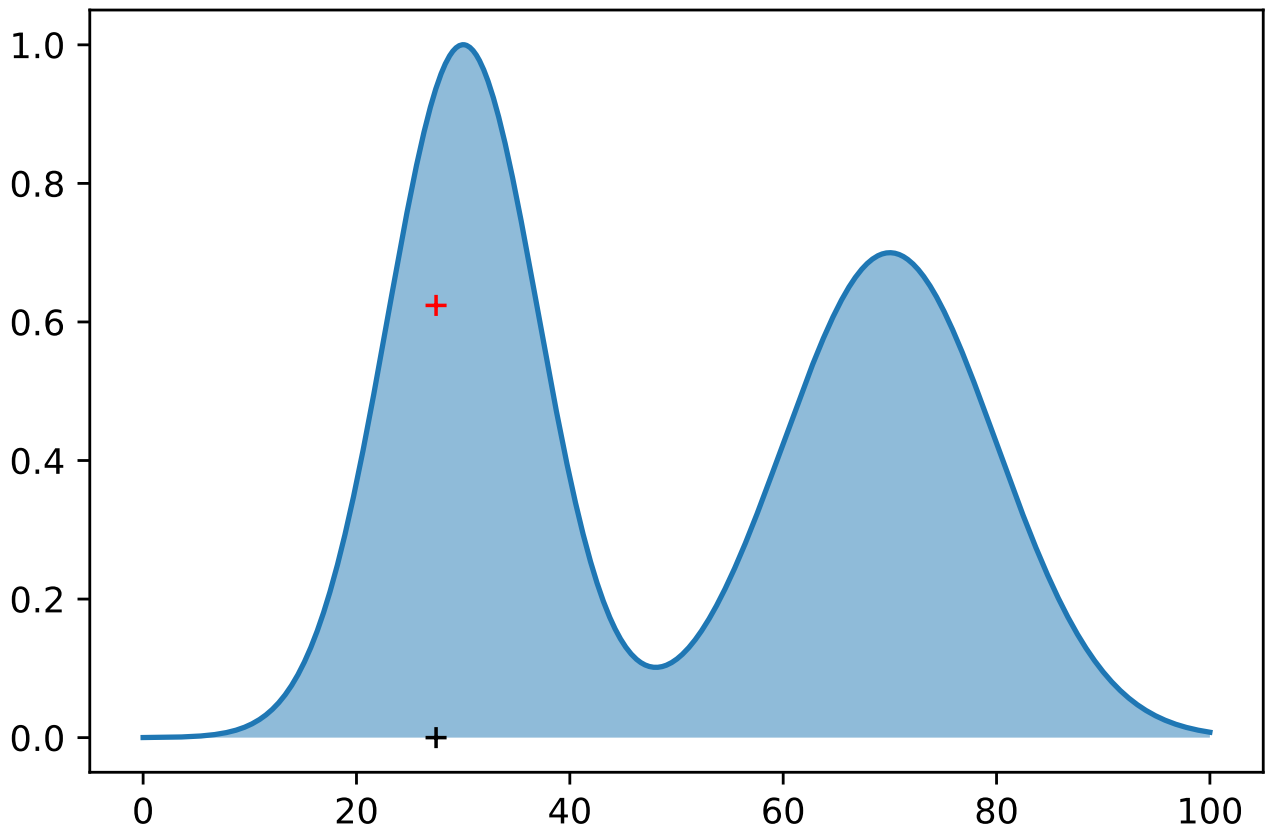
```
u = rand()
if u < 0.3:
    return ( 0 )
elif( u < 0.5):
    return ( 1 )
else:
    return ( 2 )
```

1. 0 avec probabilité 0.3, 1 avec probabilité 0.5 and 2 avec probabilité 1
2. 0 avec probabilité 0.3, 1 avec probabilité 0.2 and 2 avec probabilité 0.5
3. 0 avec probabilité 0.7, 1 avec probabilité 0.2 and 2 avec probabilité 0.1
4. Je ne sais pas.

Méthode du rejet

Soit X une variable aléatoire à valeur dans $\mathcal{X}$ et $f(x) = \mathbb{P}[X = x]$:

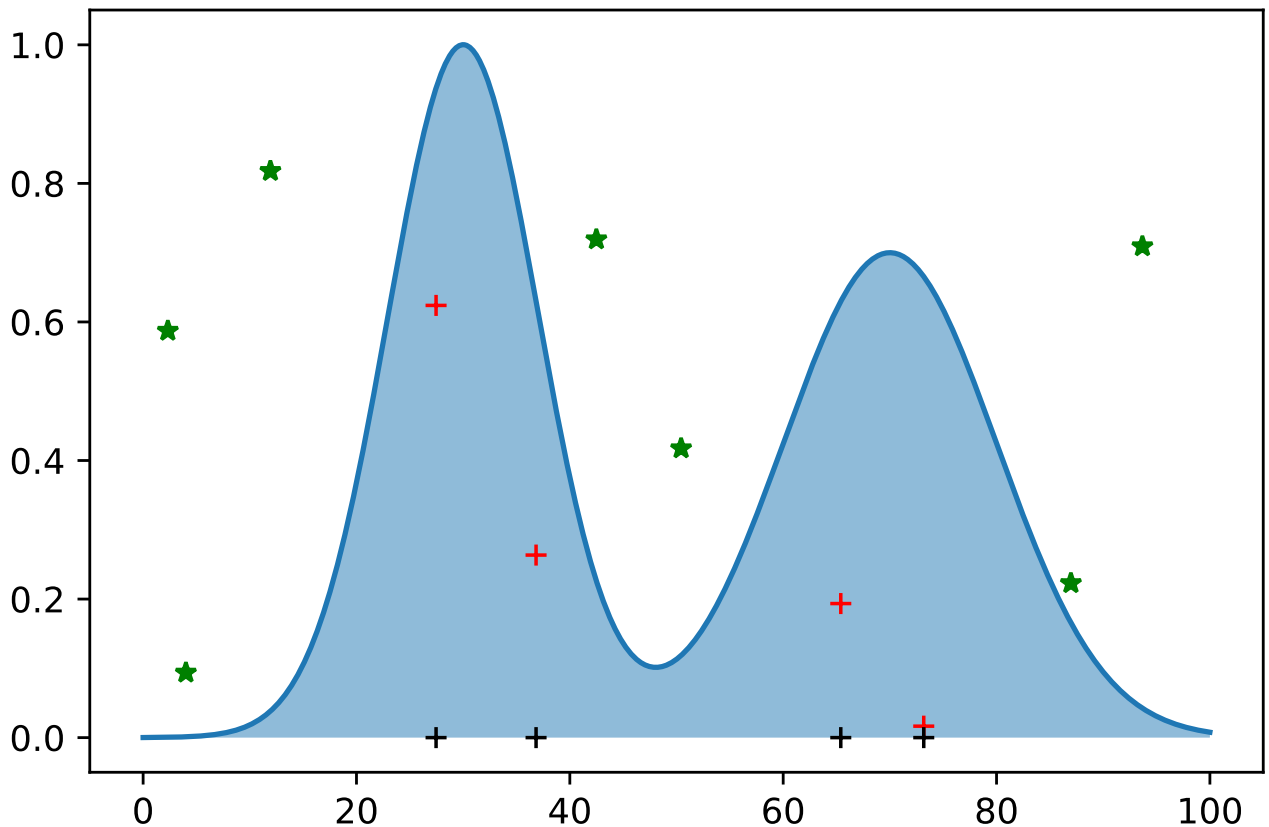
1. Générer U_1 sur $\mathcal{X}$ (uniformément).
2. Générer $U_2 \in [0, \max_x f(x)]$ (uniformément).
3. Si $U_2 \leq f(U_1)$, retourner U_1. Sinon revenir à l'étape 1.



Méthode du rejet

Soit X une variable aléatoire à valeur dans $\mathcal{X}$ et $f(x) = \mathbb{P}[X = x]$:

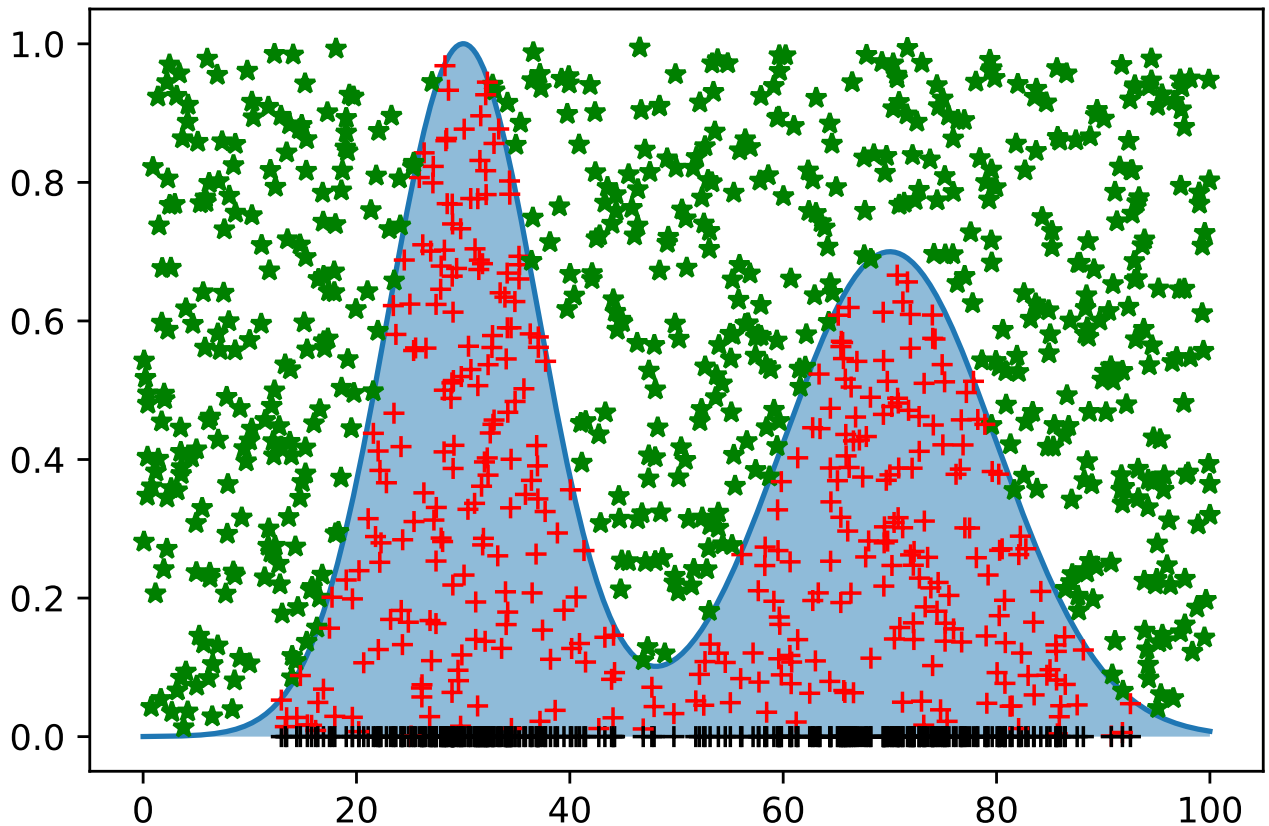
1. Générer U_1 sur $\mathcal{X}$ (uniformément).
2. Générer $U_2 \in [0, \max_x f(x)]$ (uniformément).
3. Si $U_2 \leq f(U_1)$, retourner U_1. Sinon revenir à l'étape 1.



Méthode du rejet

Soit X une variable aléatoire à valeur dans $\mathcal{X}$ et $f(x) = \mathbb{P}[X = x]$:

1. Générer U_1 sur $\mathcal{X}$ (uniformément).
2. Générer $U_2 \in [0, \max_x f(x)]$ (uniformément).
3. Si $U_2 \leq f(U_1)$, retourner U_1. Sinon revenir à l'étape 1.

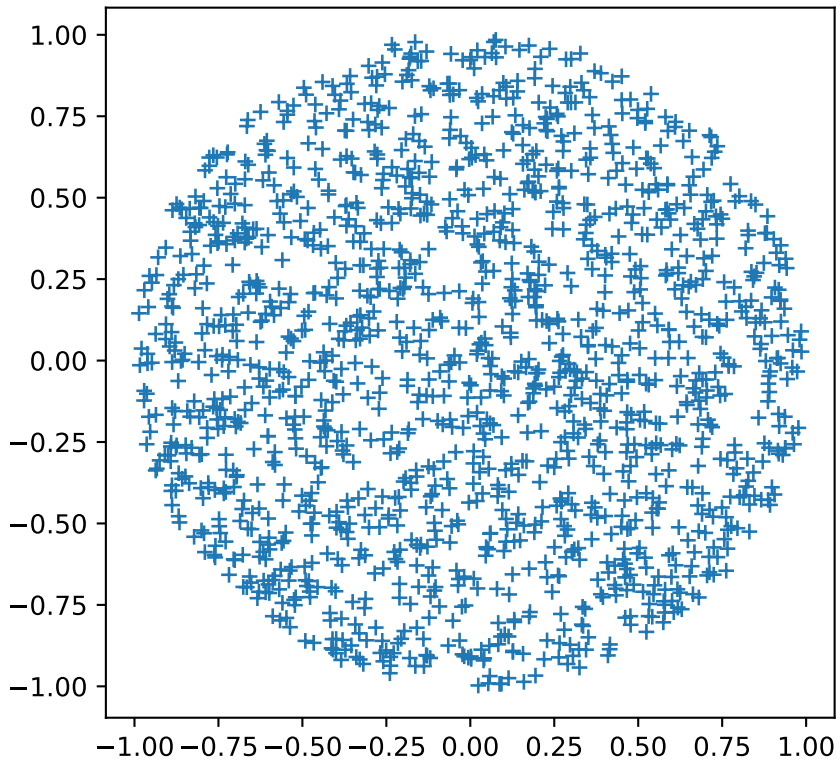


Quiz

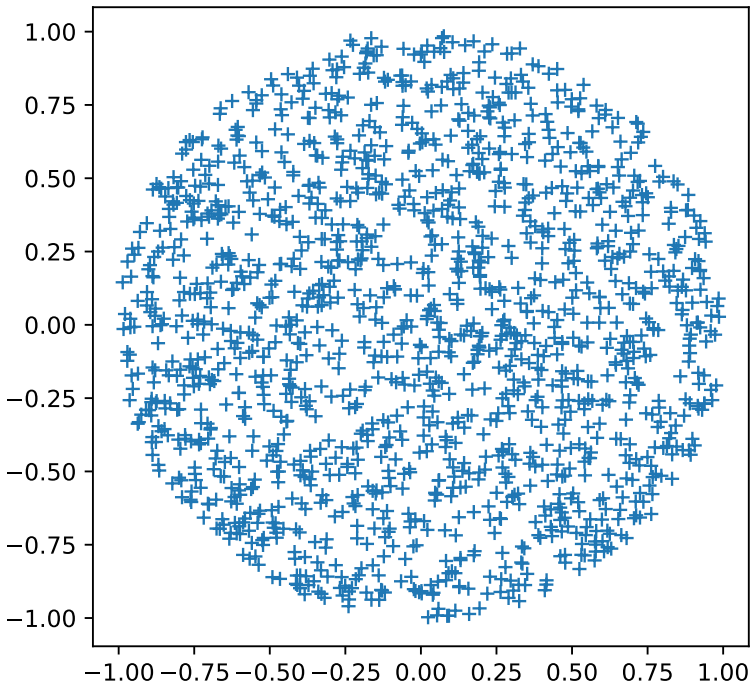
Exemple 2. On génère un couple (X, Y) tiré uniformément sur le cercle centré en 0 et de rayon 1.

Les variables X et Y sont elles indépendantes?

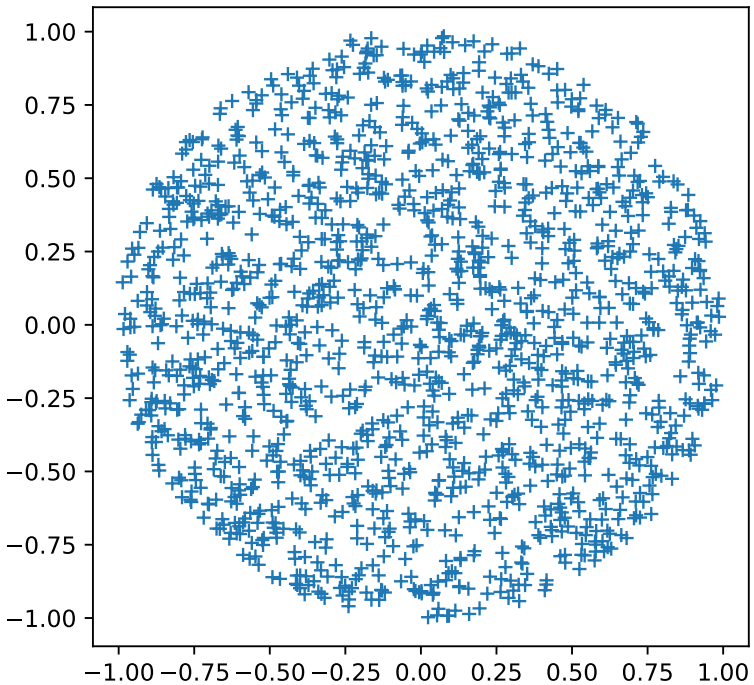
1. Oui
2. Non
3. Je ne sais pas.

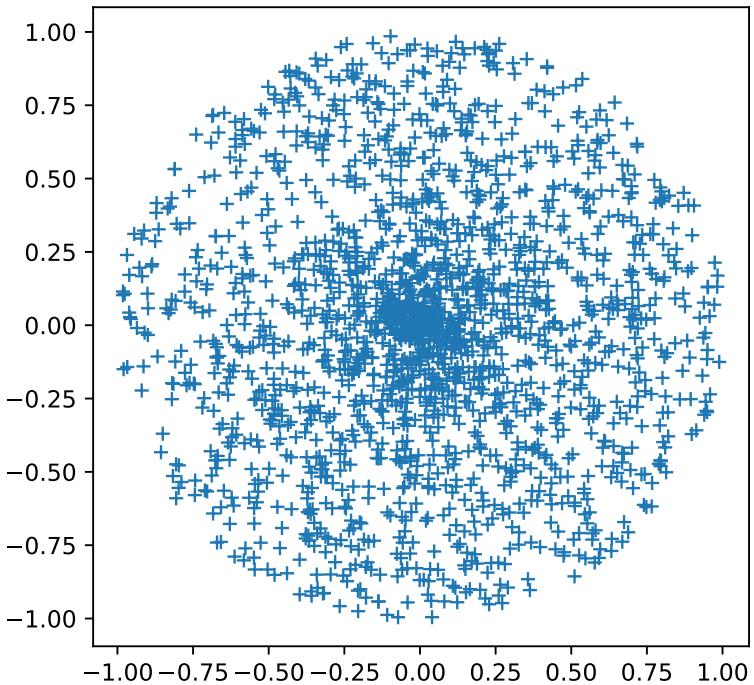


Application du rejet: comment générer une variable sur un cercle



Application du rejet: comment générer une variable sur un cercle





Rejet:
Répéter

- Générer X, Y
- Tant que $X, Y \notin \text{Circle}$

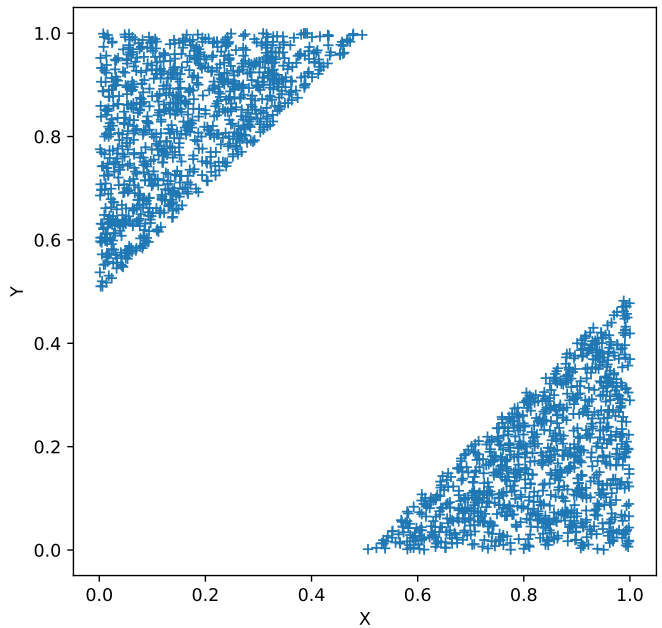
Méthode naïve (fausse):
Générer le rayon et l'angle uniformément.

Quiz

On veut générer un couple de nombre (X, Y) uniformément dans la zone bleu (*i.e.* telle que $|X - Y| > 0.5$).
L'algorithme suivante est-il correct?

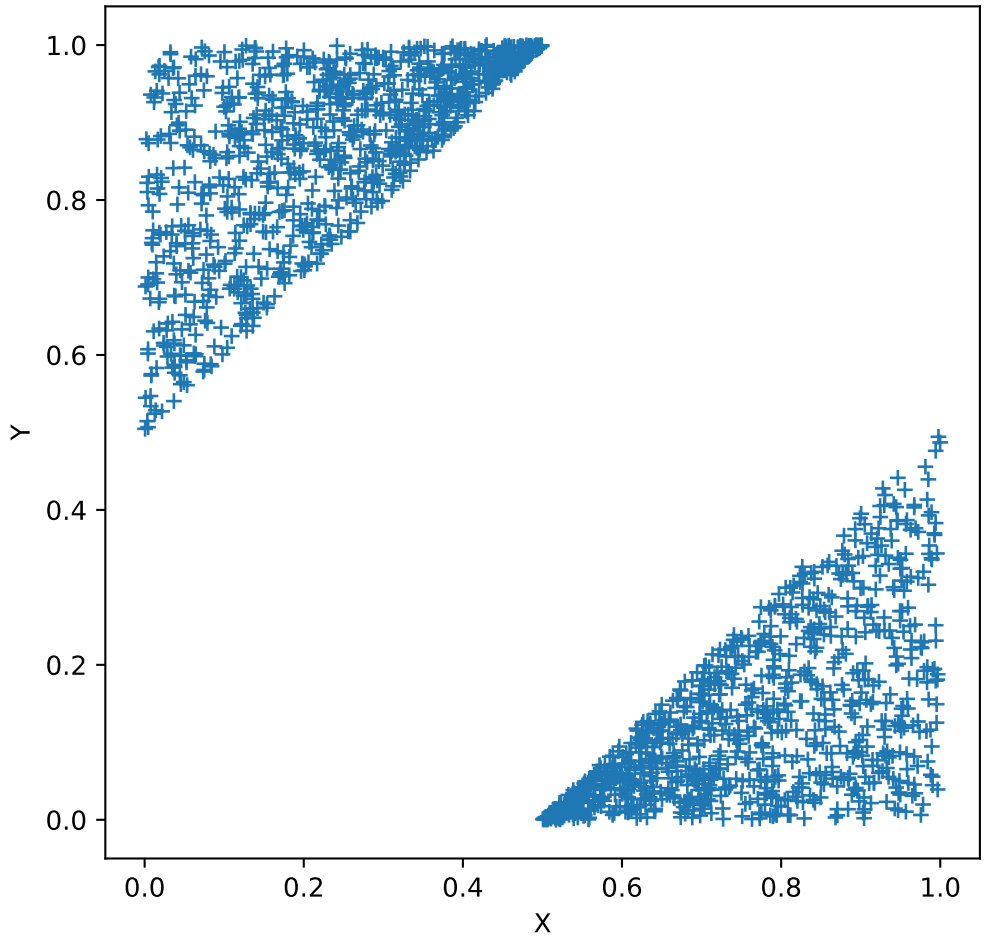
- Générer $X \sim$Unif$[0, 1]$
- Faire:
- Générer $Y \sim$Unif$[0, 1]$
- Tant que $|X - Y| \leq 0.5$

1. Oui
2. Non
3. Je ne sais pas.



Réponse: NON

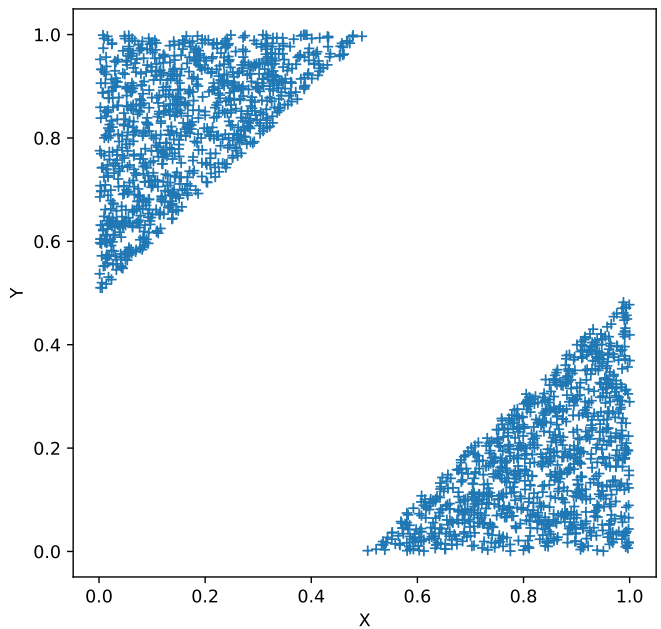
Sur une simulation (10000 tirage):

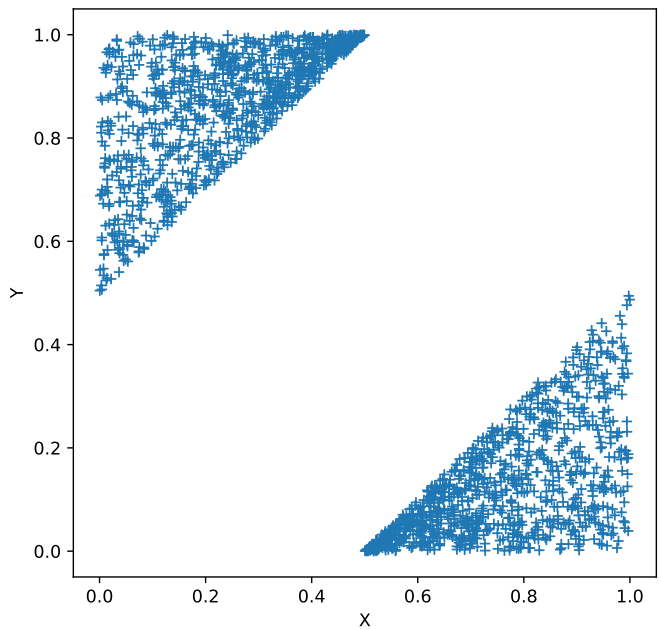


Ce n'est pas uniforme.

Pour cet exemple

Algorithme Correct	Algorithme incorrect
• Faire:	• Générer $X \sim \text{Unif}[0,1]$
• Générer $X \sim \text{Unif}[0,1]$	• Faire:
• Générer $Y \sim \text{Unif}[0,1]$	• Générer $Y \sim \text{Unif}[0,1]$
• Tant que $\lvert X - Y \rvert \leq 0.5$	• Tant que $\lvert X - Y \rvert \leq 0.5$





A retenir

- Attention, tous les générateurs pseudo-aléatoires ne sont pas bons.
- Les méthodes de rejet marchent bien.
- Il y a beaucoup de bibliothèques pour générer des nombres aléatoires: ne réinventez pas la roue.

```python
import numpy.random as rd
rd.exponential(2)             # Distribution exponentielle
rd.poisson(3)                 # Loi de poisson
rd.choice(3,p=[0.3,0.2,0.5]) # Rend 0 avec proba 0.3, 1 avec proba 0.2, 2 avec proba 0.5
```

https://docs.scipy.org/doc/numpy-1.14.0/reference/routines.random.html

Des librairies existent aussi en C, java, R, ...

Table des matières

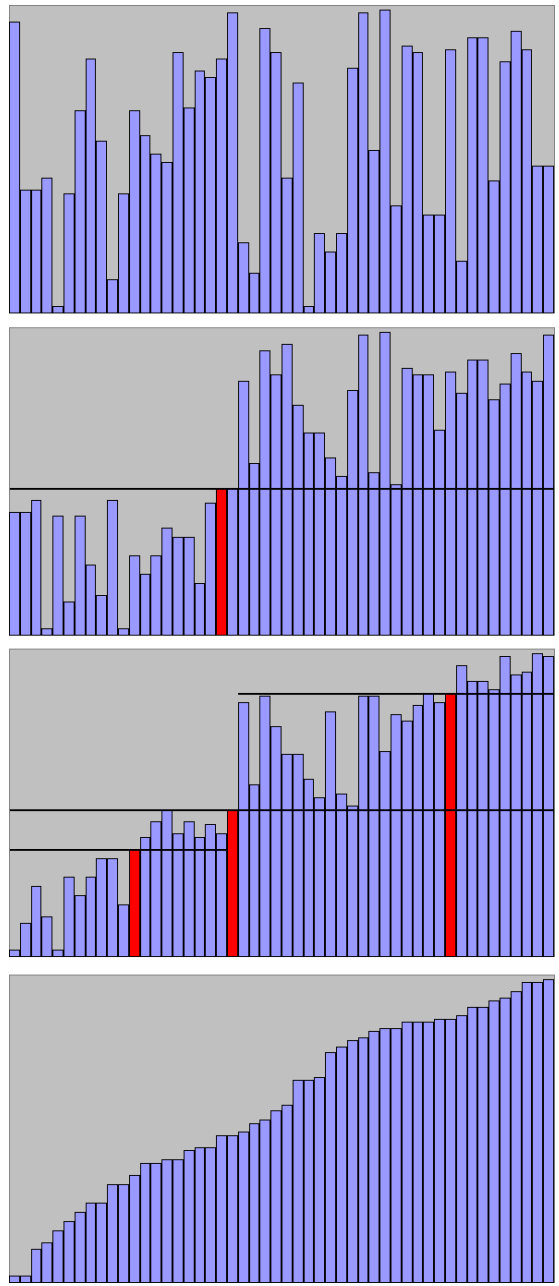
Les algorithmes randomisés sont nombreux (et variés)

- Algorithmes “Las Vegas” : Complexité probabiliste
  - Quicksort a une **complexité moyenne** de $O(n \log n)$.
  - Les tables de hachages sont très efficaces en pratique.
- Algorithmes “Monte Carlo”: Résultat probabiliste
  - Miller-Rabin
  - Coupe minimale
- Autres:
  - Algorithmes distribués
  - Apprentissage en ligne

Table des matières

Exemple d'algorithmes randomisés : **Quicksort**

- Choisir un pivot (aléatoirement)
- Partitionner
- Recommencer récursivement
  - Dans les deux tas (pour quicksort)
  - Dans un des deux (pour quickselect)

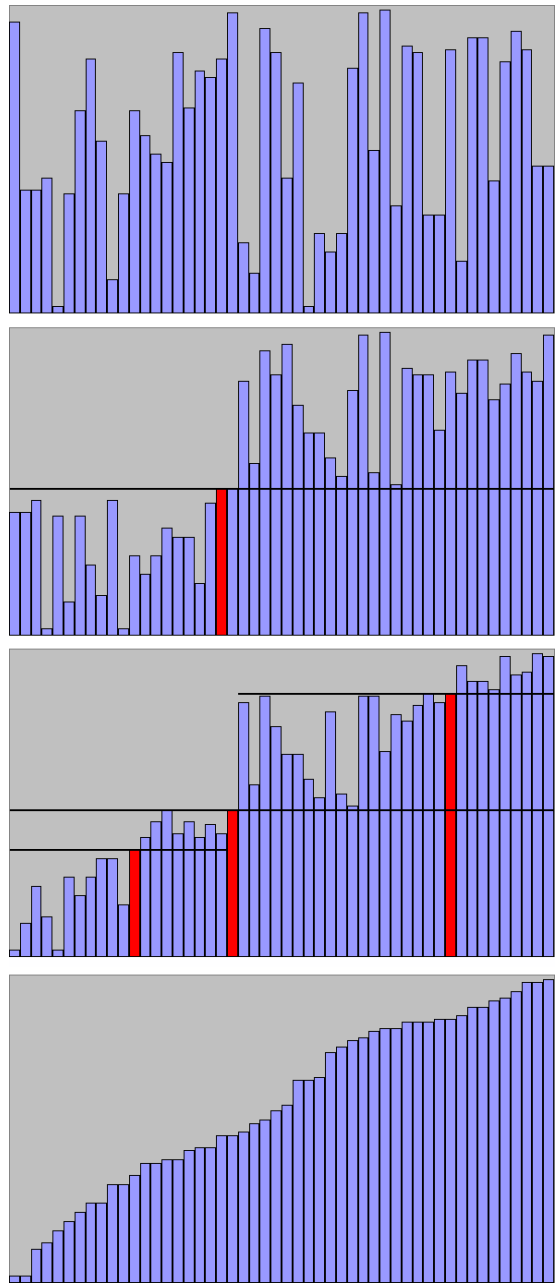


Exemple d'algorithmes randomisés : **Quicksort**

- Choisir un pivot (aléatoirement)
- Partitionner
- Recommencer récursivement
  - Dans les deux tas (pour quicksort)
  - Dans un des deux (pour quickselect)

Complexité:

- Partitionner $O(n)$
- Pire cas : $P_n : O(n^2)$.
- En moyenne : $O(n \log n)$.



Analyse de complexité: quicksort et quickselect

Analyse de complexité: quicksort et quickselect

- Quicksort : En moyenne :

$$\begin{aligned} C_n &= n + \frac{1}{n}\sum_{i=0}^{n-1} C_i + C_{n-i-1} \\ &= n + \frac{2}{n}\sum_{i=0}^{n-1} C_i \\ &= O(n \log n) \end{aligned}$$

Analyse de complexité: quicksort et quickselect

- Quicksort : En moyenne :

$$\begin{aligned} C_n &= n + \frac{1}{n}\sum_{i=0}^{n-1} C_i + C_{n-i-1} \\ &= n + \frac{2}{n}\sum_{i=0}^{n-1} C_i \\ &= O(n \log n) \end{aligned}$$

- Quickselect :

$$\begin{aligned} C_n &= n + \frac{1}{n}\sum_{i=0}^{n-1} \max(C_i, C_{n-i-1}) \\ &= n + \frac{2}{n}\sum_{i=n/2}^{n-1} C_i \\ &= O(n) \end{aligned}$$

L'aléatoire intervient aussi dans les structures de donnée

- Table de hachage : on insère $n = \alpha m$ éléments dans une table de m éléments.
  - L'insertion se fait en temps $O(\alpha)$.
  - La recherche se fait en temps $O(\alpha)$.
  - Après n insertions, $\mathbb{P}$[une case a k éléments] $\approx \alpha^k e^{-\alpha}/k!$.

L'aléatoire intervient aussi dans les structures de donnée

- Table de hachage : on insère $n = \alpha m$ éléments dans une table de m éléments.
  - L'insertion se fait en temps $O(\alpha)$.
  - La recherche se fait en temps $O(\alpha)$.
  - Après n insertions, $\mathbb{P}[\text{une case a } k \text{ éléments}] \approx \alpha^k e^{-\alpha}/k!$.

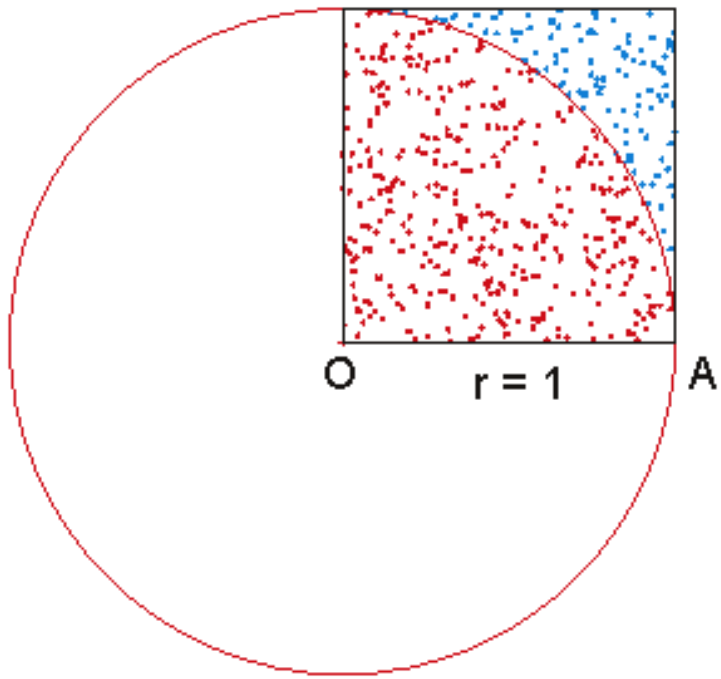
- Soit X_n la hauteur d'un arbre binaire de recherche de taille n
  - On note $Y_n = \mathbb{E}\left[2^{X_n}\right]$. On a:

$$Y_n \leq 2\sum_{i=0}^{n-1} \frac{1}{n}(Y_i + Y_{n-i-1}) \leq 4\sum_{i=0}^{n-1} Y_i = O(n^3).$$

  - D'ou: $\mathbb{E}[X_n] \leq \log_2 \mathbb{E}\left[2^{X_n}\right] \leq 3\log_2 n + O(1)$ (Jensen)

Table des matières

Estimation de π



Exemple d'algorithmes randomisés: **Miller-Rabin**

Let $2^s d = n - 1$, avec d impair. Choisir $a \in \{1, \ldots, n-1\}$. Si

$$(a^d \neq 1 \mod n) \text{ et } (a^{2^r d} \neq -1 \mod n \text{ pour tout } 0 \leq r \leq s-1)$$

alors retourner "n n'est pas premier". Sinon, retourner "n est premier".

Exemple d'algorithmes randomisés: **Miller-Rabin**

Let $2^s d = n - 1$, avec d impair. Choisir $a \in \{1, \ldots, n-1\}$. Si

$$(a^d \neq 1 \mod n) \text{ et } (a^{2^r d} \neq -1 \mod n \text{ pour tout } 0 \leq r \leq s-1)$$

alors retourner "n n'est pas premier". Sinon, retourner "n est premier".

Analyse:

- Si n est premier, aucun a ne vérifie cette propriété
- Si n est composé, au moins $3/4$ des a vérifient cette propriété.

Exemple d'algorithmes randomisés: **Miller-Rabin**

Let $2^s d = n - 1$, avec d impair. Choisir $a \in \{1, \ldots, n-1\}$. Si

$$(a^d \neq 1 \mod n) \text{ et } (a^{2^r d} \neq -1 \mod n \text{ pour tout } 0 \leq r \leq s-1)$$

alors retourner "n n'est pas premier". Sinon, retourner "n est premier".

Analyse:

- Si n est premier, aucun a ne vérifie cette propriété
- Si n est composé, au moins $3/4$ des a vérifient cette propriété.

Note : Le *test de fermat* $a^{n-1} = 1 \mod n$ n'est qu'un test heuristique car $2^{340} = 1 \mod 341$ (nombre de Carmichael)

Tests de primalité

- Miller-Rabin (1976 - 80) : $\texttt{PRIMES} \in \mathbf{co-RP}$.
  Voir aussi : Solovay–Strassen (1977)
- Adleman–Huang algorithm (1992) : $\texttt{PRIMES} \in \mathbf{RP}$

On a donc $\texttt{PRIMES} \in \mathbf{ZPP}$.

- AKS primality test (2002) : $\texttt{PRIMES} \in \mathbf{P}$.

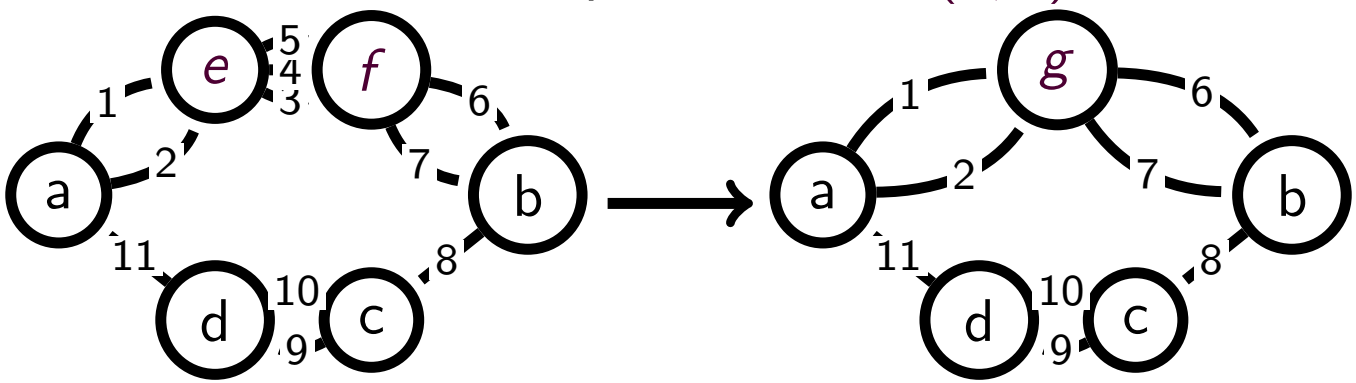
Coupe de taille minimale (algorithme de Krager)

Une coupe est un ensemble d'arêtes telles que si on les enlèves, le graphe est déconnecté.

Algorithme de Krager:

- Répéter $n-2$ fois: Choisir une arête uniformément dans le multigraphe G et "conctracter le graphe" par cette arête.

Contraction par une arête (e, f).



Analyse de l'algorithme de Krager : comment garantir l'erreur

Soit k la taille de la coupe minimale et C_{min} une coupe min. Montrer que:

- Le graphe a au moins $kn/2$ arêtes (indices: sommets isolés)

Analyse de l'algorithme de Krager : comment garantir l'erreur

Soit k la taille de la coupe minimale et $C_{\min}$ une coupe min. Montrer que:

- Le graphe a au moins $kn/2$ arêtes (indices: sommets isolés)
- Au premier tirage, on tombe sur une arête de $C_{\min}$ avec probabilité $\leq k/(kn/2) = 2/n$.
- À la $i$ème itération, le graphe a i arêtes et on tombe sur une arête de $C_{\min}$ avec probabilité $\leq 2/(n-i+1)$

Analyse de l'algorithme de Krager : comment garantir l'erreur

Soit k la taille de la coupe minimale et $C_{\min}$ une coupe min. Montrer que:

- Le graphe a au moins $kn/2$ arêtes (indices: sommets isolés)
- Au premier tirage, on tombe sur une arête de $C_{\min}$ avec probabilité $\leq k/(kn/2) = 2/n$.
- À la $i$ème itération, le graphe a i arêtes et on tombe sur une arête de $C_{\min}$ avec probabilité $\leq 2/(n-i+1)$
- D'où:

$$\mathbb{P}\left[\text{jamais choisir } C_{\min}\right] \geq 2/n^2.$$

- Comment garantir une probabilité d'erreur ε?

Analyse de l'algorithme de Krager : comment garantir l'erreur

Soit k la taille de la coupe minimale et $C_{\min}$ une coupe min. Montrer que:

- Le graphe a au moins $kn/2$ arêtes (indices: sommets isolés)
- Au premier tirage, on tombe sur une arête de $C_{\min}$ avec probabilité $\leq k/(kn/2) = 2/n$.
- À la $i$ème itération, le graphe a i arêtes et on tombe sur une arête de $C_{\min}$ avec probabilité $\leq 2/(n-i+1)$
- D'où:

$$\mathbb{P}\left[\text{jamais choisir } C_{\min}\right] \geq 2/n^2.$$

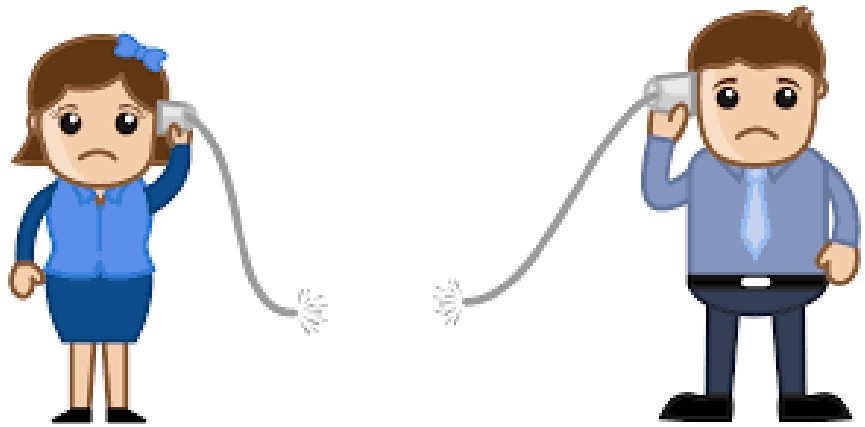
- Comment garantir une probabilité d'erreur ε?
  - Solution: répéter l'algorithme $Kn^2/2$ fois donne une erreur

$$(1-2/n^2)^{Kn^2/2} \leq e^{-K}$$

Table des matières

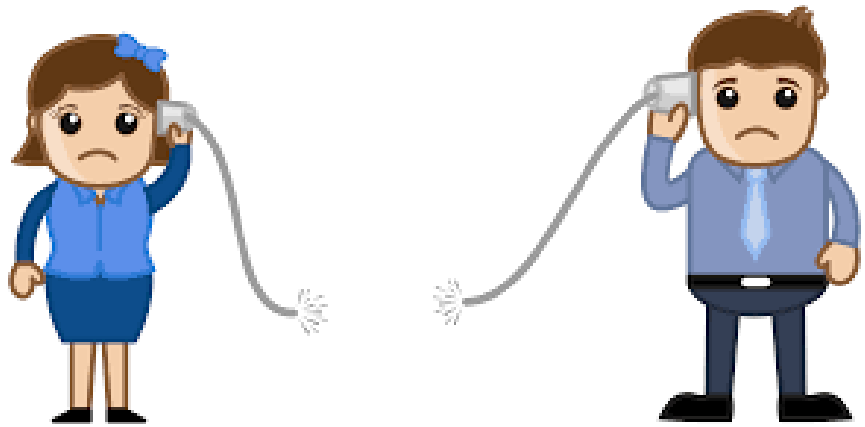
(Dé)synchronisation

Problème: comment décider qui parle?



(Dé)synchronisation

Problème: comment décider qui parle?



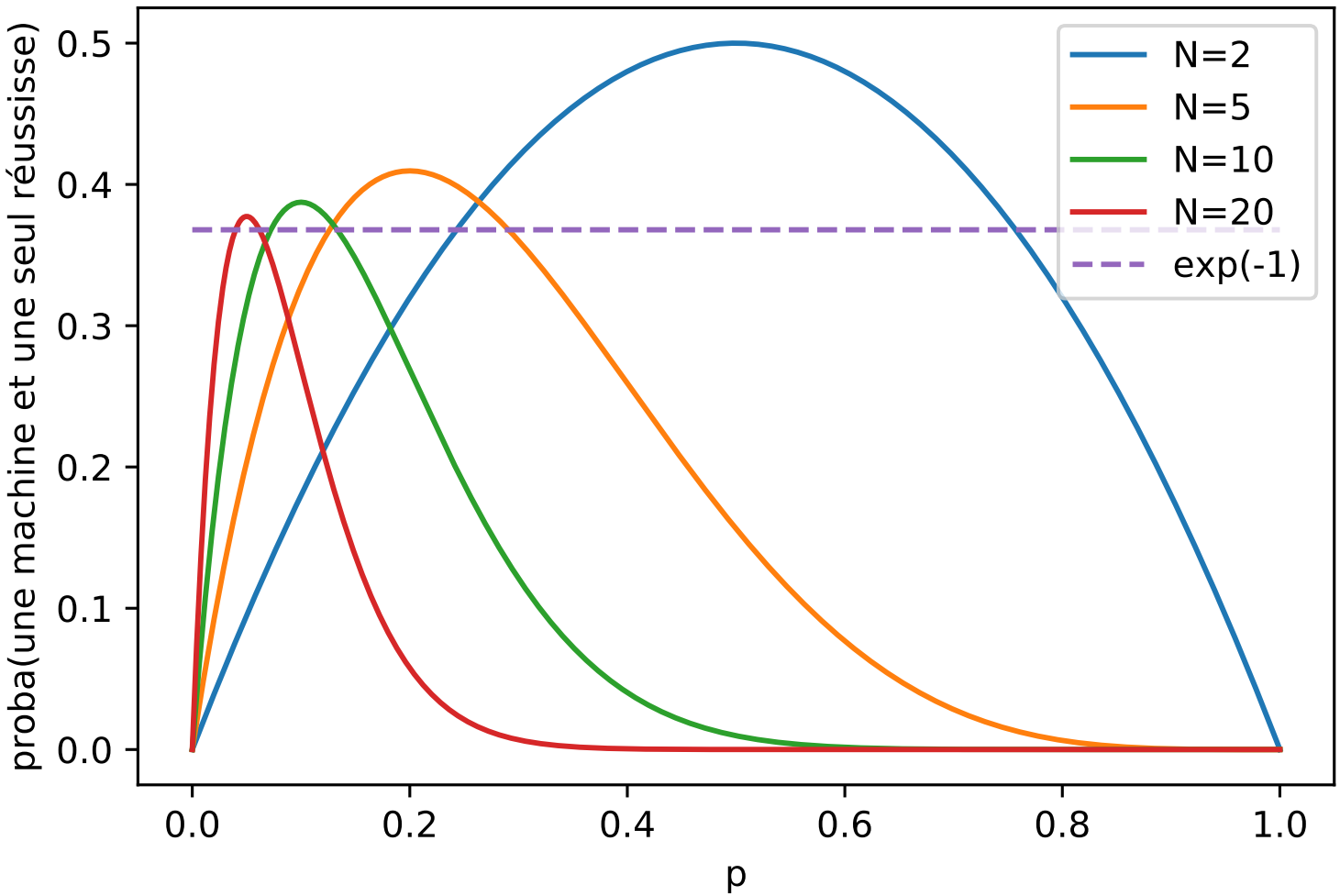
Algorithmes (Slotted Aloha)

- N machines veulent communiquer sur un cannal.
- Le temps discret: $t \in \{0, 1, 2, \dots\}$.
- Au temps t, chaque machine transmet avec probabilité p.
- Si deux machines transmettent en même temps, le paquet est perdu.

Question: quel est le meilleur p?

Analyse de "Slotted Aloha"

Analyse de "Slotted Aloha"



Apprentissage en ligne

Motivation

- **Tests cliniques**

μ_1 μ_2 μ_3 μ_4




  - Choisir un traitement A_t pour le patient t
  - Observer la réponse $X_t \in \{0, 1\}$ with $P(X_t = 1) = \mu_{A_t}$.
  - But: maximiser le nombre de guérisons.

Motivation

- **Tests cliniques**

μ_1 μ_2 μ_3 μ_4




  - Choisir un traitement A_t pour le patient t
  - Observer la réponse $X_t \in \{0, 1\}$ with $P(X_t = 1) = \mu_{A_t}$.
  - But: maximiser le nombre de guérisons.

- **Publicité en ligne**, *par exemple*, trouver le meilleur titre pour un article de presse:

Titre	Proba. clique
"Deux corps sans vie trouvés dans la rue Goriot"	μ_1
"Meurtres de la rue Goriot: un possible serial-killer?"	μ_2

  - Choisir un titre A_t à montrer à l'internaute t
  - Observer la réponse $X_t \in \{0, 1\}$ (clique ou non).
  - But: maximiser le nombre de cliques.

Quelques idées de politique

- **Aléatoire** – Choisir A_t aléatoirememement (et uniformément).
  - Exploration

Quelques idées de politique

- **Aléatoire** – Choisir A_t aléatoirememement (et uniformément).
  - Exploration

- **Glouton:** Toujours choisir le meilleur choix:

$$A_{t+1} = \arg\max_{a \in \{1 \dots n\}} \hat{\mu}_a(t)$$

  - Exploitation

Quelques idées de politique

- **Aléatoire** – Choisir A_t aléatoiremement (et uniformément).
  - Exploration
- **Glouton:** Toujours choisir le meilleur choix:

$$A_{t+1} = \arg\max_{a \in \{1 \dots n\}} \hat{\mu}_a(t)$$

  - Exploitation
- ε-**glouton** : "glouton" avec probabilité $1 - \varepsilon$ et "aléatoire" sinon.
  - Exploration and exploitation.

Analyse de la politique ϵ-glouton pour $\epsilon > 0$

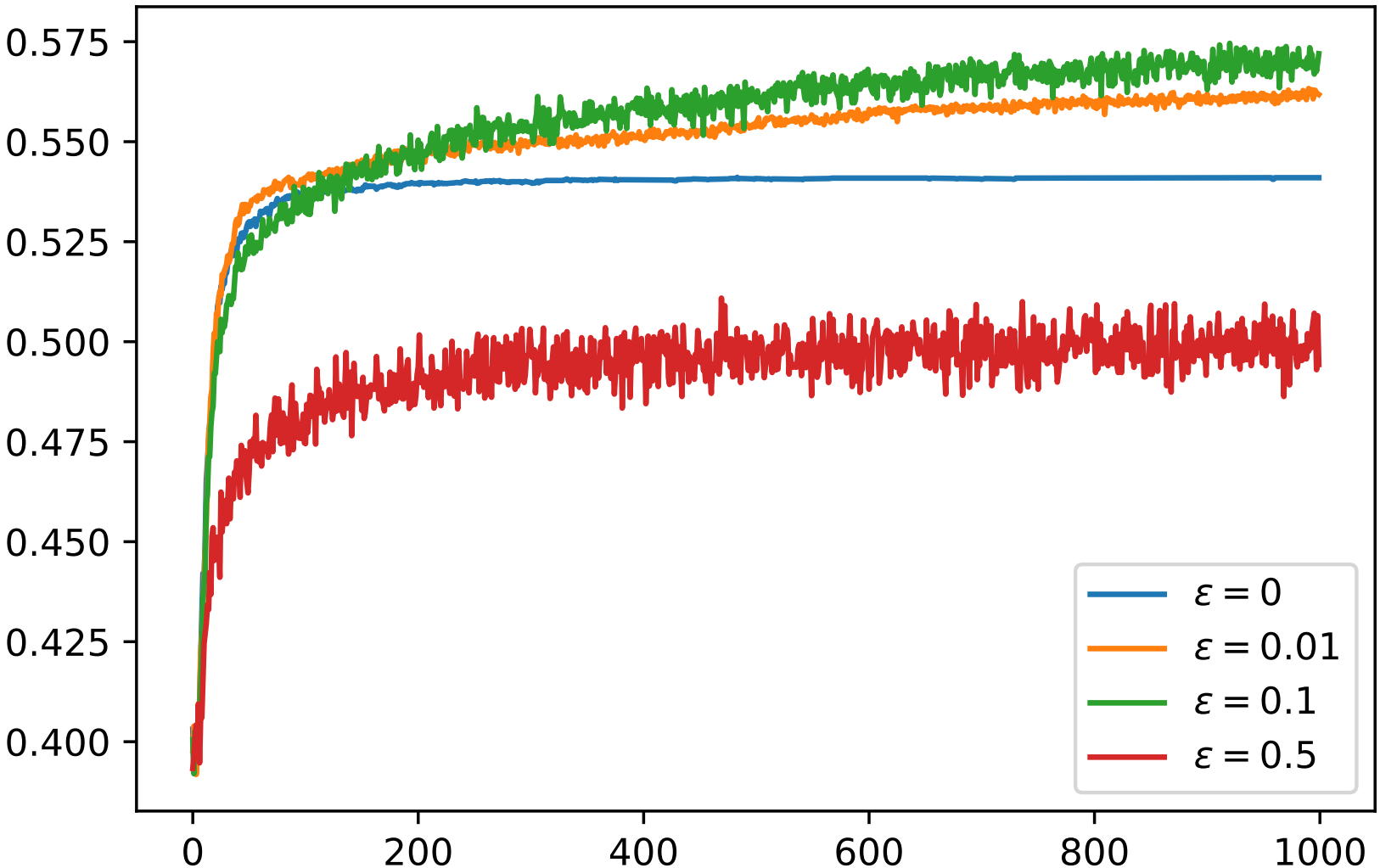
Analyse de la politique ϵ-glouton pour $\epsilon > 0$

- Comme $\epsilon > 0$, tous les choix vont être faits un nombre infini de fois.
- Par la loi des grands nombres: $\lim_{t\to\infty} \hat{\mu}_a(t) = \mu_a(t)$.
- Le gain par épisode converge donc vers

$$(1-\varepsilon)\max_a \mu_a + \frac{\varepsilon}{n}\sum_a \mu_a$$

ε-glouton : un ϵ plus petit ou plus grand n'est pas forcément mieux.

Simulation: 5 choix avec probabilité $\mu = [0.5, 0.3, 0.6, 0.4, 0.2]$. Le gain en fonction du temps est



L'algorithme UCB

L'algorithme UCB

L'idée de UCB est de calculer une borne de confiance $UCB_a(t)$ telle que $\mu_a(t) \leq UCB_a(t)$ avec grande probabilité:

$$UCB_a(t) = \hat{\mu}_a(t) + \sqrt{\frac{\alpha \log t}{2N_a(t)}}.$$

- Choisit $A_{t+1} \in \underset{a \in \{1 \dots n\}}{\arg\max}\, UCB_a(t)$ (optimisme).

L'algorithme UCB

L'idée de UCB est de calculer une borne de confiance $UCB_a(t)$ telle que $\mu_a(t) \leq UCB_a(t)$ avec grande probabilité:

$$UCB_a(t) = \hat{\mu}_a(t) + \sqrt{\frac{\alpha \log t}{2N_a(t)}}.$$

- Choisit $A_{t+1} \in \underset{a \in \{1 \dots n\}}{\arg\max}\, UCB_a(t)$ (optimisme).

Table des matière

Conclusion

L'aléatoire se retrouve partout. Dans se cours nous avons pu voir (ou revoir):

- Rappels de probabilité, probabilités bayésiennes et intervalles de confiance.
- Générations aléatoires
- Nombreux algorithmes randomisés.