Renaud Carpentier
Benoît Dépret

La physique de sup en applications avec Python

MPSI
PCSI
PTSI
MP2I

32 modélisations de sujets contemporains avec corrigés

ellipses

1. Incertitudes par temps de pluie

MOTS-CLÉS : Incertitudes, incertitudes composées, méthode de Monte-Carlo

<u>Capacités numériques</u> (filières PCSI, MPSI, PTSI, MP2I) :
Simuler, à l'aide d'un langage de programmation ou d'un tableur, un processus aléatoire permettant de caractériser la variabilité de la valeur d'une grandeur composée.
Utiliser la fonction `hist` de la bibliothèque `matplotlib.pyplot` (sa spécification étant fournie) pour représenter les résultats d'un ensemble de tirages d'une variable aléatoire.

Il pleut. Quelle excellente occasion de revenir sur la notion d'incertitudes !

On réalise l'expérience simple consistant à mettre deux bassines dehors sous la pluie et à mesurer la quantité d'eau de pluie récupérée après quelques heures de mauvais temps. La bassine 1 est de forme carrée de côté 14,5 cm, tandis que la bassine 2 est circulaire de diamètre 29,0 cm. Ces longueurs sont mesurées à 0,5 cm près.

On note les masses d'eau recueillie par chaque bassine :

Bassine 1	Bassine 2
72 g	222 g

Les masses sont mesurées à 5 g près avec la balance de la cuisine.

1) Montrer que le rapport des masses d'eau dans chaque bassine permet de déterminer une valeur expérimentale du nombre π, notée dans la suite π_{exp}.

On souhaite évaluer l'incertitude-type sur cette mesure. On va pour cela simuler des valeurs expérimentales à l'aide de tirages aléatoires par la méthode de Monte-Carlo.

DOCUMENT : La méthode de Monte-Carlo adapté de « www.wikipedia.fr »

La méthode de Monte-Carlo désigne une famille de méthodes algorithmiques visant à calculer une valeur numérique approchée en utilisant des procédés aléatoires, c'est-à-dire des techniques probabilistes. Le nom de ces méthodes, qui fait allusion aux jeux de hasard pratiqués au casino de Monte-Carlo, a été inventé en 1947 par Nicholas Metropolis et Stanislaw Ulam. Les méthodes de Monte-Carlo sont particulièrement utilisées pour calculer des intégrales en dimensions plus grandes que 1 (en particulier, pour calculer des surfaces et des volumes). Elles sont également couramment utilisées en physique des particules, où des simulations probabilistes permettent d'estimer la forme d'un signal ou la sensibilité d'un détecteur. La comparaison des données mesurées à ces simulations peut permettre de mettre en évidence des caractéristiques inattendues, par exemple de nouvelles particules.

On pourra utiliser l'instruction `np.random.uniform(-1, 1, N)` qui renvoie un tableau de `N` nombres aléatoires compris entre -1 et 1 avec une distribution uniforme. On rappelle que `np.mean(x)` et `np.std(x, ddof=1)` renvoient respectivement la moyenne et l'écart-type des valeurs contenues dans un tableau `x`. L'instruction `plt.hist(x, bins=100)` permet de tracer l'histogramme des valeurs contenues dans `x` réparties en 100 intervalles.

On note m_1 la masse d'eau mesurée dans la bassine 1 et Δm_1 son incertitude. Celle-ci est en fait une estimation raisonnable de la demi-largeur de l'intervalle de valeurs possibles de la masse d'eau recueillie, due à la variabilité de la grandeur étudiée suite au protocole de mesure ou à la précision de l'instrument utilisé pour effectuer la mesure.

2) Créer un tableau de `N` valeurs aléatoires de la masse d'eau contenue dans la bassine 1 choisies uniformément dans l'intervalle $[m_1 - \Delta m_1\,;\, m_1 + \Delta m_1]$ et représenter l'histogramme correspondant. Evaluer l'incertitude-type $u(m_1)$ sur la masse m_1, définie comme l'écart-type de la distribution. Quelle relation simple lie $u(m_1)$ et Δm_1 ? Quel nom courant donne-t-on à ce type dévaluation. On pourra prendre `N = 1 000 000`.

On choisit ainsi aléatoirement tous les paramètres du problème, c'est-à-dire à la fois les masses d'eau recueillie et la taille des bassines, dans chaque intervalle de variabilité.

3) Créer un tableau contenant `N` valeurs expérimentales de π_{exp} obtenues avec `N` valeurs aléatoires des masses d'eau et de la taille des bassines et représenter l'histogramme correspondant. En déduire l'incertitude-type sur la mesure expérimentale de π_{exp}. Comparer la valeur de π_{exp} obtenues à partir des mesures et la moyenne des valeurs aléatoires calculées. Donner la valeur numérique de π_{exp} avec le bon nombre de chiffres significatifs.

Pour valider une mesure expérimentale, les laboratoires de recherche scientifique utilisent la notion d'écart normalisé (ou *Z-score*) par rapport à une valeur de référence défini par :

$$Z = \frac{x - x_{ref}}{u(x)}$$

où x est la valeur mesurée, x_{ref} est la valeur de référence (le plus souvent la moyenne des valeurs obtenues) et $u(x)$ l'écart-type de la distribution des valeurs de x.

Lorsque $|Z| \leq 2$, le résultat de la mesure est acceptable.

Lorsque $2 < |Z| \leq 3$, la mesure nécessite une attention particulière, notamment si cette situation se reproduit souvent.
Lorsque $|Z| > 3$, la mesure doit être rejetée et on recherche alors les causes possibles de cette anomalie.

4) Calculer l'écart normalisé de la mesure expérimentale en prenant comme référence la valeur théorique de π. La mesure obtenue est-elle acceptable ?

Il existe des relations donnant l'incertitude-type sur une grandeur composée connaissant l'incertitude-type sur les grandeurs intermédiaires. On a ainsi par exemple :

Grandeur	Incertitude-type
x, y	$u(x)$, $u(y)$
$z = ax + by$	$u(z) = \sqrt{\left(au(x)\right)^2 + \left(bu(y)\right)^2}$
$z = ax^\alpha y^\beta$	$\frac{u(z)}{z} = \sqrt{\left(\alpha \frac{u(x)}{x}\right)^2 + \left(\beta \frac{u(y)}{y}\right)^2}$

5) Exprimer l'incertitude-type sur π_{exp} en fonction des incertitudes-types sur les différents paramètres de l'expérience. Comparer le résultat numérique à la valeur obtenue par la méthode de Monte-Carlo. Commenter.

La méthode de Monte-Carlo peut aussi permettre de calculer des surfaces. On se propose de simuler la pluie par des tirages aléatoires bidimensionnels pour déterminer la valeur de π.

On considère le disque de rayon R, contenu dans un carré de côté $2R$. Comme les gouttes de pluie récupérées par chaque bassine, on choisit aléatoirement et uniformément des points dans le carré. La probabilité qu'un point soit contenu dans le disque est alors égale au rapport de la surface du disque sur la surface du carré.

6) Exprimer la probabilité théorique qu'un point choisi aléatoirement dans le carré soit contenu dans le disque.

7) Ecrire la fonction Python d'entête :

```
def experience(Ntirages):
```

qui renvoie une valeur expérimentale de π par la méthode de Monte-Carlo après un nombre `Ntirages` de tirages dans le carré.

8) Comparer les mesures obtenues avec `100, 1 000 et 10 000` tirages en traçant les histogrammes représentant la répartition de 1 000 valeurs. Quel est le rôle du nombre de tirages dans l'expérience numérique ainsi réalisée ? Comparer les incertitudes-types relatives à chaque expérience.

On pourra rajouter l'option `range = (2.8, 3.5)` dans `plt.hist()` pour tracer l'histogramme dans l'intervalle fixe `[2.8, 3.5]` afin de mieux comparer les résultats.

Il est d'usage de tracer l'écart normalisé pour les différentes mesures expérimentales par rapport à leur valeur moyenne pour détecter les éventuelles anomalies. On considère une série de 10 000 points de mesures obtenus avec des expériences à 1 000 tirages aléatoires.

9) Tracer l'écart normalisé relatif à ces 10 000 mesures et l'histogramme de la distribution des valeurs entre -5 et 5. Commenter.

10) Déterminer la probabilité d'avoir un *Z-score* $|Z| \leq 2$ et commenter le critère d'acceptabilité d'une mesure expérimentale.

CORRIGÉ

1) Pour déterminer la valeur théorique du rapport $\frac{m_2}{m_1}$ des masses d'eau recueillie par les bassines, on suppose que les gouttes de pluie tombant au sol sont réparties aléatoirement et uniformément en surface. Cela signifie que la quantité d'eau récupérée est en moyenne proportionnelle à la surface de la bassine et à la durée de l'expérience. On pourrait définir un courant surfacique de masse d'eau apportée par la pluie, en moyenne vertical et

uniforme, de norme j_0 (en kg.m^{-2}.s^{-1}), de sorte qu'après la durée Δt la masse d'eau récupérée dans chaque bassine vaut respectivement :

$$m_1 = j_0 S_1 \Delta t \text{ et } m_2 = j_0 S_2 \Delta t$$

où $S_1 = a_1^2$ (surface du carré de côté a_1) et $S_2 = \dfrac{\pi a_2^2}{4}$ (surface du disque de diamètre a_2).

On obtient ainsi :

$$\frac{m_2}{m_1} = \frac{S_2}{S_1} = \frac{\pi a_2^2}{4a_1^2} \text{, c'est-à-dire : } \pi = \frac{4m_2 a_1^2}{m_1 a_2^2}$$

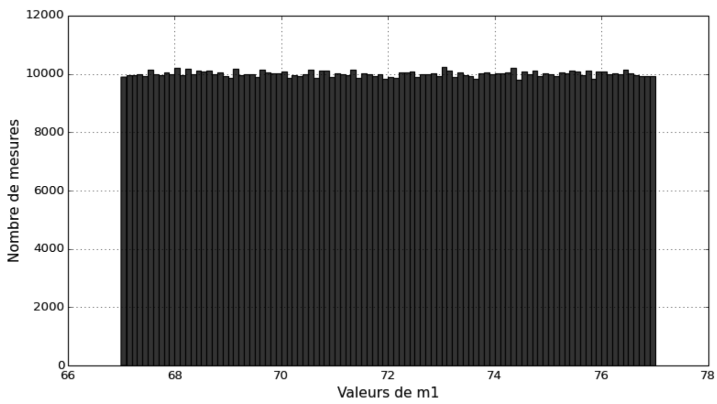
En remarquant que $a_2 = 2a_1$, on obtient simplement :

$$\pi_{exp} = \frac{m_2}{m_1} = 3,08$$

On retrouve un résultat proche de la valeur théorique de π (à 1,3 % près). Il faut cependant évaluer l'incertitude-type associée à cette mesure pour valider le protocole utilisé.

2) L'instruction `np.random.uniform()` permet de créer directement un tableau et on trace l'histogramme montrant la répartition des valeurs dans l'intervalle de variabilité :

```
m1 = 72 # g
delta_m1 = 5 # g
N = 1000000
m1_MC = m1 + delta_m1*np.random.uniform(-1, 1, N) # Valeurs aléatoires
plt.hist(m1_MC, bins=100)
```



S'agissant d'une distribution aléatoire uniforme, les valeurs tirées au hasard sont équiprobables dans l'intervalle choisi.

On obtient l'incertitude-type en calculant l'écart-type de la distribution de valeurs :

```
u_m1 = np.std(m1_MC, ddof=1)
```

Numériquement :

```
u_m1 → 2.8871250630507777 # g
```

On en déduit le rapport :

```
delta_m1/u_m1 → 1.7318266063322461
```

que l'on peut comparer à :

```
np.sqrt(3) → 1.7320508075688772
```

Statistiquement, on peut donc exprimer l'incertitude-type $u(m_1)$ en fonction de l'incertitude expérimentale Δm_1 sous la forme :

$$\boxed{u(m_1) = \frac{\Delta m_1}{\sqrt{3}}}$$

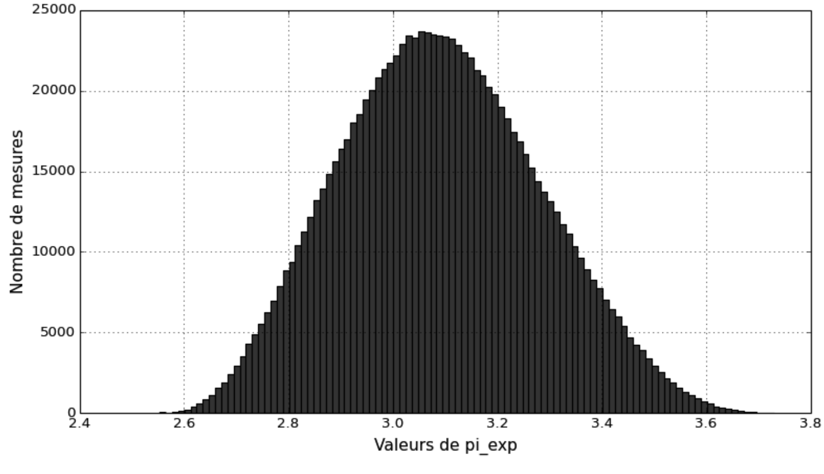
Il s'agit d'une évaluation de type B de l'incertitude-type sur une mesure unique.

3) On calcule `N` valeurs de π_{exp}, sans oublier de tenir compte de la variabilité de toutes les grandeurs en choisissant aléatoirement des masses m_1 et m_2 et des longueurs a_1 et a_2 :

```
m2 = 222 # g
delta_m2 = 5 # g
a1 = 14.5 # cm
delta_a1 = 0.5 # cm
a2 = 29.0 # cm
delta_a2 = 0.5 # cm
m2_MC = m2 + delta_m2*np.random.uniform(-1, 1, N) # Valeurs aléatoires
a1_MC = a1 + delta_a1*np.random.uniform(-1, 1, N) # Valeurs aléatoires
a2_MC = a2 + delta_a2*np.random.uniform(-1, 1, N) # Valeurs aléatoires
pi_MC = 4 * m2_MC * a1_MC**2 / (m1_MC * a2_MC**2) # Calcul de pi_exp
```

Le formalisme des tableaux `np.array` permet de calculer directement le tableau `pi_MC`. On représente l'histogramme des valeurs obtenues :

```
plt.hist(pi_MC, bins=100)
```



La répartition des valeurs obtenues suit apparemment une loi gaussienne caractéristique d'une distribution statistique normale.

On calcule maintenant l'écart-type de cette distribution, correspondant à l'incertitude-type sur chaque valeur expérimentale :

```
u_pi = np.std(pi_MC, ddof=1) → 0.18965148801364812
```

On compare la valeur expérimentale de π_{exp} et la moyenne de la distribution aléatoire :

```
pi_exp = 4 * m2 * a1**2 / (m1 * a2**2) → 3.0833333333333335
pi_moy = np.mean(pi_MC) → 3.0905710203951293
```

Ces valeurs sont proches étant donnée l'incertitude-type obtenue. Dans une approche classique on prendra le plus souvent la valeur issue des mesures expérimentales.

Finalement, un résultat numérique avec deux chiffres significatifs est suffisant :

$$\pi_{exp} = 3{,}1 \text{ et } u(\pi_{exp}) = 0{,}2$$

4) On calcule l'écart normalisé avec la définition fournie :

```
(pi_exp - np.pi)/u_pi → -0.3071914745655307
```

c'est-à-dire $Z = -0{,}31$. Avec $|Z| \leq 2$, la mesure de π obtenue est acceptable.

5) On effectue ici le calcul classique d'incertitudes composées. On a la relation :

$$\pi_{exp} = \frac{4 m_2 a_1^2}{m_1 a_2^2}$$

qui impose l'incertitude-type relative sur π_{exp} :

$$\frac{u(\pi_{exp})}{\pi_{exp}} = \sqrt{\left(\frac{u(m_1)}{m_1}\right)^2 + \left(\frac{u(m_2)}{m_2}\right)^2 + \left(2\frac{u(a_1)}{a_1}\right)^2 + \left(2\frac{u(a_2)}{a_2}\right)^2}$$

en prenant :

$$u(m_1) = \frac{\Delta m_1}{\sqrt{3}},\ u(m_2) = \frac{\Delta m_2}{\sqrt{3}},\ u(a_1) = \frac{\Delta a_1}{\sqrt{3}} \text{ et } u(a_2) = \frac{\Delta a_2}{\sqrt{3}}$$

Numériquement :

```
pi_exp*np.sqrt((u_m1/m1)**2 + (u_m2/m2)**2
               + (2*u_a1/a1)**2 + (2*u_a2/a2)**2)
→ 0.18948793593740834
```

On retrouve la même valeur que celle obtenue avec la méthode de Monte-Carlo, au prix d'un calcul conséquent qu'on ne pourra pas forcément mener jusqu'au bout si la relation est trop compliquée ! La méthode de Monte-Carlo est un outil puissant pour déterminer efficacement l'incertitude-type sur des grandeurs composées.

6) On calcule le rapport des surfaces pour obtenir la probabilité p qu'un point tiré au hasard dans le carré se trouve aussi dans le disque :

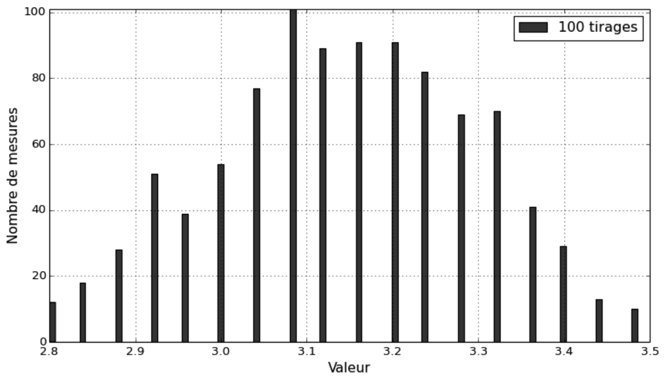
$$p = \frac{\pi R^2}{(2R)^2} = \frac{\pi}{4} = 78{,}5\ \%$$

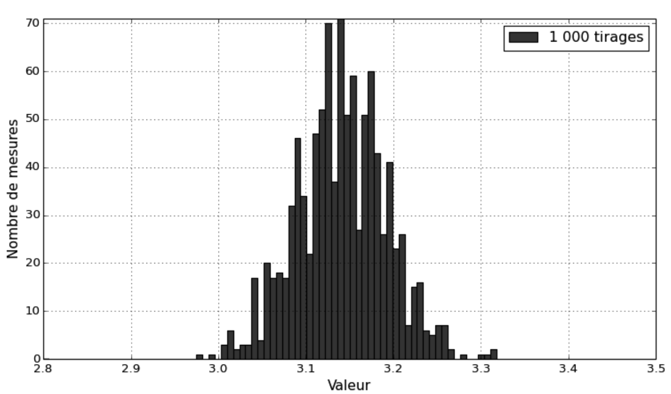
7) On utilise à nouveau la fonction `np.random.uniform(-1,1,N)` pour obtenir aléatoirement les coordonnées (`x,y`) d'un point dans le carré de côté $R = 1$ (les résultats ne dépendent pas de la valeur de R). On teste si les point choisis sont situés dans le disque et on compte le nombre de fois où cela arrive. La probabilité de tirer un point dans le disque est le rapport du nombre de points dans le disque sur le nombre total de tirages. En identifiant à la probabilité théorique, on évalue π.

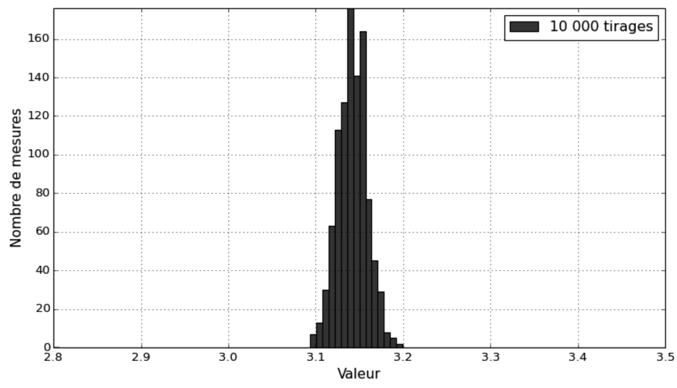
```
def experience(Ntirages):
    x = np.random.uniform(-1, 1, Ntirages) # Tirages aléatoires
    y = np.random.uniform(-1, 1, Ntirages) # Tirages aléatoires
    r = np.sqrt(x**2 + y**2) # Calcul de la distance au centre
    compteur = 0 # Initialisation du compteur
    for k in range(Ntirages):
        if r[k] <= 1: # Le point est situé dans le disque
            compteur = compteur + 1
    return 4 * compteur / Ntirages # Valeur expérimentale de pi
```

8) On crée un tableau contenant les 1 000 mesures de π avec l'expérience numérique, on calcule l'incertitude-type de la distribution de valeurs et on trace l'histogramme correspondant :

```
Ntirages = 100 # Nombre de tirages
N = 1000 # Nombre de mesures
mesures = np.array([experience(Ntirages) for i in range(N)])
u_pi = np.std(mesures, ddof=1) # Incertitude-type
plt.hist(mesures, range=(2.8, 3.5), bins=100)
```







Les valeurs obtenues à chaque expérience sont d'autant plus resserrées autour de la valeur moyenne que le nombre de tirages est important, comme le confirment les valeurs de l'incertitude-type :

Nombre de tirages	Incertitude-type
100	0.168537350277
1 000	0.0512813104044
10 000	0.0163772400774

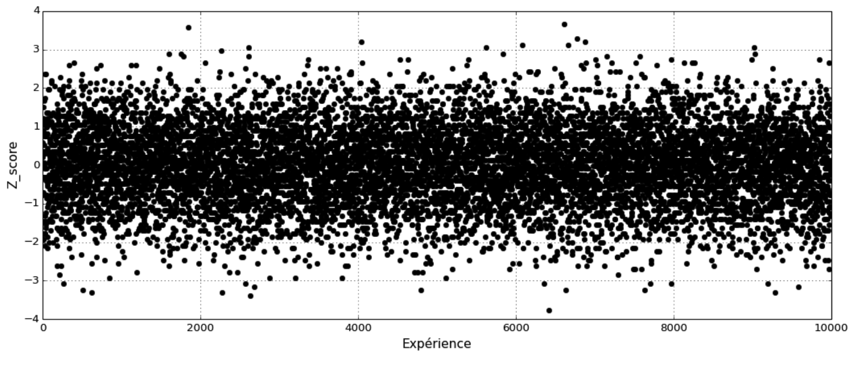
Le nombre de tirages aléatoires dans la simulation numérique impose en fait la variabilité de la grandeur mesurée, que l'on ne peut généralement pas contrôler expérimentalement.

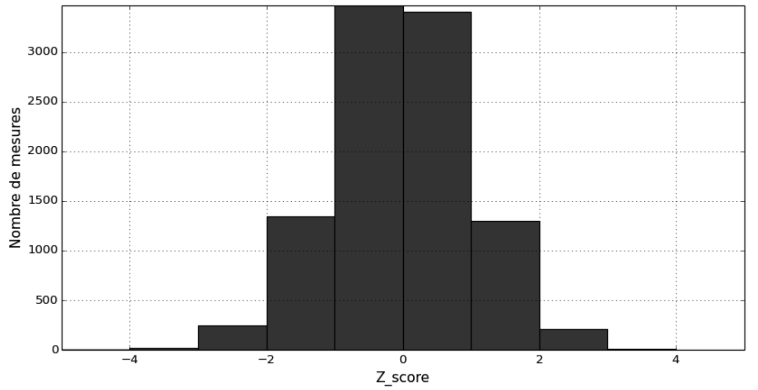
9) On crée le tableau des écarts normalisés pour chaque point de mesure :

```
mesures = np.array([experience(Ntirages) for i in range(N)])
pi_moy = np.mean(mesures) # Moyenne
u_pi = np.std(mesures, ddof=1) # Incertitude-type
z_score = (mesures - pi_moy)/u_pi # Ecart normalisé
```

et on trace à la fois l'écart normalisé et l'histogramme correspondant :

```
plt.plot(z_score, 'ko')
plt.hist(z_score, range=(-5,5), bins=10)
```





Les valeurs de l'écart normalisé sont essentiellement situées entre -2 et 2, ce qui valide globalement les mesures expérimentales.

10) On estime sur l'histogramme qu'il y a environ 500 mesures telles que $|Z| > 2$ (en sommant les valeurs entre -3 et -2 d'une part et entre 2 et 3 d'autre part). Sur 10 000 mesures, cela représente environ 5 % des mesures. On peut confirmer ce résultat en comptant le nombre de mesures telles que $|Z| \leq 2$ avec la fonction suivante :

```
def confiance(z_score, seuil):
    N = len(z_score)
    compteur = 0
    for k in range(N):
        if abs(z_score[k]) <= seuil:
            compteur = compteur + 1
    return compteur / N
```

On en déduit :

```
confiance(z_score, 2) → 0.956
```

Statistiquement, une mesure expérimentale prise au hasard a donc bien environ 95 % de chance d'avoir un écart normalisé tel que $|Z| \leq 2$.

Cette probabilité est encore plus grande si on considère $|Z| \leq 3$:

```
confiance(z_score, 3) → 0.9965
```

La valeur de l'écart normalisé permet donc bien de détecter les anomalies de mesure : si une mesure (ou pire s'il y en a plusieurs) a un écart normalisé supérieur à 3 (en valeur absolue), il y a de fortes chances qu'il s'agisse d'un problème de manipulation ou une anomalie du protocole de mesures plutôt qu'un écart statistique.