Calcul des probabilités et théories des erreurs
Rapport projet
Question 6

UMONS
Faculté des Sciences
Sciences Informatiques BAC 3.

Maximilien Charlier

Année académique
2014 - 2015

Table des matières

1 Introduction

Nous avons reçu une question portant sur le cours de calcul de probabilité et théorie des erreurs.
Il nous a était demandé d'y répondre le plus formellement possible et d'expliquer notre démarche.

2 Énoncé

La question étudiée est la suivante[1] :
« Dans le cadre d'une étude portant sur la détection précoce du cancer, pour tester l'efficacité de trois algorithmes de traitement d'images, on a mesuré le nombre de foyers suspects par clichés radiologiques.
Les résultats sont donnés ci-après.
Que pouvez-vous dire sur l'équivalence possible de ces trois algorithmes ? »

1	7	6	4
2	10	6	6
3	4	2	4
4	2	2	2
5	6	15	16
...	...	...	...
998	3	2	5
999	14	12	9
1000	2	1	1

Dans la première colonne on retrouve l'indice du test, et dans les 3 suivantes respectivement les résultats de l'algorithme 1, 2 et 3.

Tout au long de ce rapport, les algorithmes seront appelés par leur placement dans le fichier (les données de la première colonne donnent `algo1`, les données de la seconde donnent `algo2` et enfin la dernière colonne pour `algo3`).

3 Mode de résolution

On va montrer l'équivalence ou non des algorithmes en montrant qu'ils suivent ou non la même loi de probabilité.
Pour cela, 3 histogrammes seront utilisés pour visualiser la "forme" des lois que suivent les échantillons.
Ce premier aperçu permettra d'orienter la recherche de la loi que suivent les données.

Une fois la loi supposée identifiée, les tests de Khi-2 et de Kolmogorov-Smirnov seront utilisés pour montrer formellement que l'échantillon la suit.

Pourquoi montrer qu'ils suivent la même loi ?

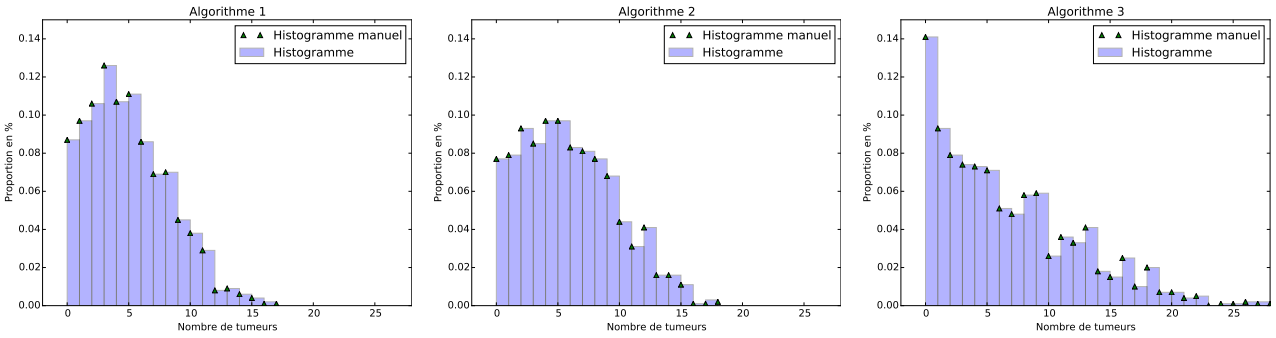
1. Cette question correspond à la question 6.

« Une loi de probabilité décrit le comportement aléatoire d'un phénomène dépendant du hasard. » [2]
Par conséquent, montrer que deux algorithmes suivent la même loi permet de réaliser une comparaison entre eux et de déterminer s'il y a une équivalence.
Par exemple, si une loi gaussienne est identifiée, cette comparaison peut être faite par un test de Fisher (qui permet de montrer que la variance de deux échantillons est la même), et par un test par intervalles de confiance pour montrer que la moyenne entre deux échantillons est la même) si le test de Fisher réussi.

4 Histogramme

Voici les histogrammes des 3 algorithmes :

Algorithme 1 Algorithme 2 Algorithme 3



L'axe X représente le nombre de tumeurs observées sur une radio.
L'axe Y quant à lui, le nombre de fois où ce nombre de tumeur a été détecté.

Premier constat
Les données sont toutes positives, les algorithmes 1 et 2 ressemblent fortement à un gaussienne dont il manque une partie à gauche.
L'algorithme 3 quant à lui ressemble à une exponentielle ou une gaussienne tronquée qui aurait une moyenne négative.

5 Recherche de la (ou les) loi(s) que suivent nos échantillons.

Dans cette section, nous allons détaillé la recherche de la ou les lois que suivent nos 3 échantillons et aussi le prouver.

5.1 Test de conformité à une loi.

Test de Khi-2.

Le test statistique du Khi-2 (χ^2) permet de vérifier si un échantillon (lot de données) suit une loi donnée comme hypothèse (l'équation de la loi).
Il faut calculer une valeur K_n et la comparer avec χ^2 [3].
K_n est donné par [4] :

2. Wikipédia
3. Que l'on trouve dans une table de la loi de Khi-2.
4. Formule issue de la séance 4 de TP

$$K_n = \sum_{i=1}^{r} \frac{(n_i - np_i)^2}{np_i}$$

Où r est le nombre d'événements, i est un événement, n_i est le nombre de répétitions d'un événement i dans notre échantillon, n est le nombre d'éléments dans notre échantillon, et p_i est la probabilité que l'événement i se produise (calculé avec la loi donnée en hypothèse).

On trouve χ^2 dans une table en fonction d'un α et d'un n. Pour nos tests, on choisit $\alpha = 5\%$ comme préconisé dans la littérature [5]. Le n est donné par le nombre d'événements (-1) [6].

On a deux cas possibles, **si $K_n < \chi^2$ l'échantillon suit la loi donnée en hypothèse**, sinon il est plus grand et notre échantillon ne suit pas la loi. χ^2 vaut 27.587 pour l'échantillon 1 (avec 18 évènements), 28.689 pour le deuxième échantillon (avec 19 événements) et 40.113 pour le dernier échantillon avec 27 échantillons.

Test de Kolmogorov-Smirnov.

Tout comme le test de Khi-2, il permet de vérifier si un échantillon respecte une loi, non pas en se basant sur sa densité de probabilité (comme χ^2), mais en se basant sur la fonction de répartition [7] de la loi dont on essaye de montrer l'appartenance.
On doit calculer un D_n donné par [8] :

$$D_n = \frac{Max\,|F_n(x) - F_0(x)|}{n}$$

où $F_n(x)$ est la fonction de répartition de l'échantillon en x et $F_0(x)$ est la fonction de répartition de la loi donnée en hypothèse sur l'échantillon.

Il faut aussi prendre un D_α dans une table [9], comme pour le test de Khi-2 on prend $\alpha = 5\%$. D_α dépend aussi du nombre d'éléments dans les échantillons testés, ici 1000 dans les 3 échantillons qui sont testés.
On a donc D_α qui vaut 0.043.

L'échantillon suit la loi donnée en hypothèse si $D_n < D_\alpha$ et ne la suit pas sinon.

5.2 Une loi normale (Gaussienne)

Nous allons essayer d'approcher une gaussienne à nos échantillons en suivant les formules vues au cours de probabilité.

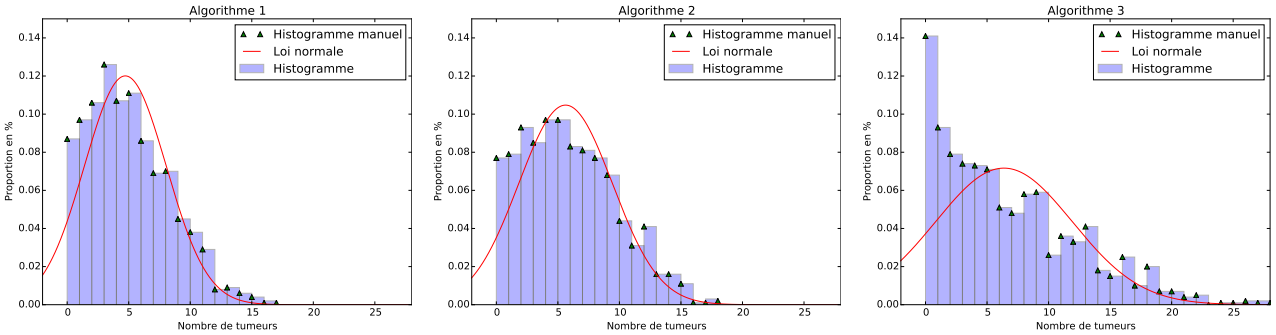
La densité de probabilité de la loi normale est donnée en x par $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\,\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$ où μ est la moyenne et σ est l'écart-type.
La moyenne μ est donnée par la somme des éléments de l'échantillon divisée par le nombre d'échantillons (ici 1000 pour chaque algorithme)
L'écart-type σ est la racine carrée de la variance.
La variance se calcule comme, suit `moyenne([(x-`μ`)`^{2} `for x in tableau])`. Donc la somme de

5. D'après le cours, et Wikipédia.
6. D'après les exercices.
7. L'air sous la courbe de la fonction de probabilité de $-\infty$ à x.
8. Formule issue de la séance 4 de TP
9. Que l'on trouve dans une table de Kolmogorov.

tous les échantillons, auxquels on soustrait la moyenne μ avant de les mettre au carré. Le résultat obtenu est ensuite divisé par le nombre d'échantillons.

Le calcul de la moyenne μ et de l'écart-type σ a donc été automatisé.
On a ensuite ajouté la gaussienne obtenue à chaque histogramme pour déterminer s'il existe une certaine proximité entre eux.

Algorithme 1 Algorithme 2 Algorithme 3



Le premier problème auquel on est confronté est qu'il manque des données sur la gauche (comme précédemment observé). Quand à l'algorithme 3, on constate qu'il ne suit pas du tout une gaussienne, telle que définie dans le cours.

Nous allons réaliser un test de Khi-2 et de Kolmogorov-Smirnov afin de montrer que les algorithmes ne suivent pas une loi normale.
L'α pour Khi-2 et Kolmogorov est fixé à 5% comme préconisé dans la littérature.
Voici les résultats obtenus :

	Khi-2			Kolmogorov-Smirnov		
	Obtenue	Voulue	Acceptable	Obtenue	Voulue	Acceptable
Algo1	112	< 27.587	NON	0.117	≤ 0.043	NON
Algo2	114	< 28.869	NON	0.101	≤ 0.043	NON
Algo3	504	< 40.113	NON	0.201	≤ 0.043	NON

On peut en conclure que les 3 algorithmes ne suive pas de loi normale.
Voici les moyenne et écart-type trouvé avec la formule de la loi normale.

	Moyenne	Écart-type	Médiane	Min	Max
Algo1	4.723	3.322	4	0	17
Algo2	5.625	3.808	5	0	18
Algo3	6.372	5.566	5	0	26

5.3 Une gaussienne tronquée

Suivent-il une loi gaussienne tronquée ?
Une loi gaussienne tronqué est une loi dérivée de la loi normale.
La densité de probabilité de la loi gaussienne tronqué en 0 est donné en x par

$$f(x,\mu,\sigma) = \begin{cases} 0 & \text{si x} < 0, \\ \frac{\frac{1}{\sigma}}{1-\Phi(\frac{0-\mu}{\sigma})}\phi(\frac{x-\mu}{\sigma}) & \text{sinon.} \end{cases}$$

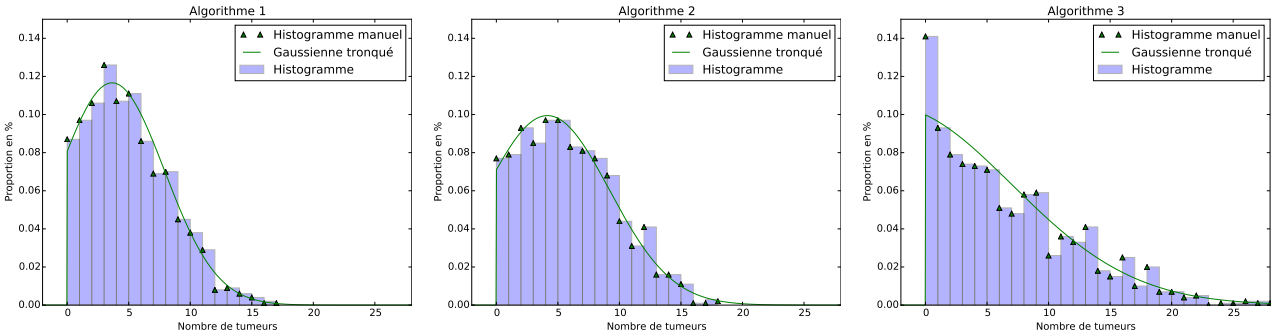
où μ est la moyenne, σ est l'écart type, $\Phi(.)$ est la fonction de répartition et $\phi(.)$ est la densité de la loi normale standard.
Avec $\phi(x) = \frac{1}{\sigma\sqrt{2\pi}}\,\mathrm{e}^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$
et $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\mathrm{e}^{-\frac{1}{2}t^2}\,dt$

$\Phi(x)$ est une intégrale impossible à résoudre analytiquement, on utilise donc Cumulative distribution function (cdf) de la loi normal disponible dans une librairie python [10] pour évaluer sa valeur.

Afin d'approcher le plus possible les valeurs des histogrammes, on a cherché les valeurs de μ et σ qui minimisent les valeurs obtenues avec les test de Khi-2 et de Kolmogorov-Smirnov.
On a obtenu les valeurs suivantes :

	Moyenne	Écart-type
Algo1	3.65	4.25
Algo2	4.15	5.05
Algo3	-3	10

Algorithme 1 Algorithme 2 Algorithme 3



Voici les valeurs minimales obtenues avec le test de Khi-2 et Kolmogorov-Smirnov.

	Khi-2			Kolmogorov-Smirnov		
	Obtenue	Voulue	Acceptable	Obtenue	Voulue	Acceptable
Algo1	14.29	≤ 27.587	OUI	0.0384	≤ 0.043	OUI
Algo2	19.35	≤ 28.869	OUI	0.0317	≤ 0.043	OUI
Algo3	62.79	≤ 40.113	NON	0.0501	≤ 0.043	NON

On vient donc de montrer que les algorithmes algo1 et algo2 suivent une loi normale tronquée.
Ils sont donc par conséquent comparables.
Quant à l'algorithme algo3, il est possible qu'il suive une loi normale tronquée tout comme il lui est possible de suivre une loi exponentielle. On va donc faire le test pour cette loi.

5.4 Algorithme 3 : loi exponentielle

La densité de probabilité d'une loi exponentielle est donnée par :

$$f(x,\lambda) = \begin{cases} 0 & \text{si x} < 0 \\ \lambda\,\mathrm{e}^{-\lambda x} & \text{sinon.} \end{cases}$$

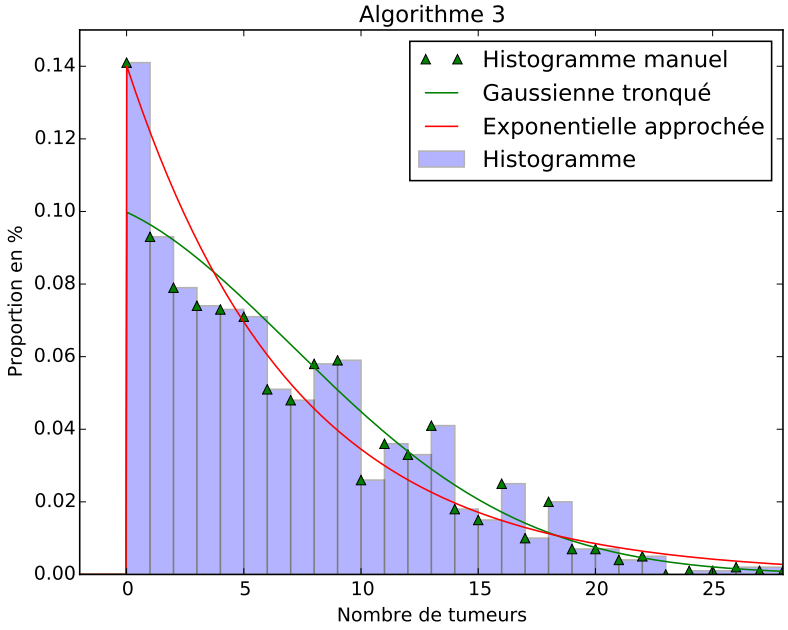
10. La librairie scipy.stats.norm

avec μ l'espérance mathématique valant $\frac{1}{\lambda}$.

La valeur de μ est $7,13$, cette valeur a été choisie pour que l'exponentielle approche le plus possible les valeurs de l'histogramme de l' algorithme algo3.

Voici les valeurs minimales obtenues avec le test de Khi-2 et Kolmogorov-Smirnov pour l'algorithme algo3.

Loi	Khi-2			Kolmogorov-Smirnov		
	Obtenue	Voulue	Acceptable	Obtenue	Voulue	Acceptable
Exponentielle	79.08	≤ 40.113	NON	0.0501	≤ 0.043	NON
Gaussienne tronquée	62.79	≤ 40.113	NON	0.0927	≤ 0.043	NON



On remarque que la loi exponentielle suit encore moins les valeurs de notre échantillon que la loi gaussienne tronquée.
Bien qu'il ait raté le test de Khi-2 et de Kolmogorov pour la loi gaussienne tronquée, elle a des valeurs proches de celles attendues.

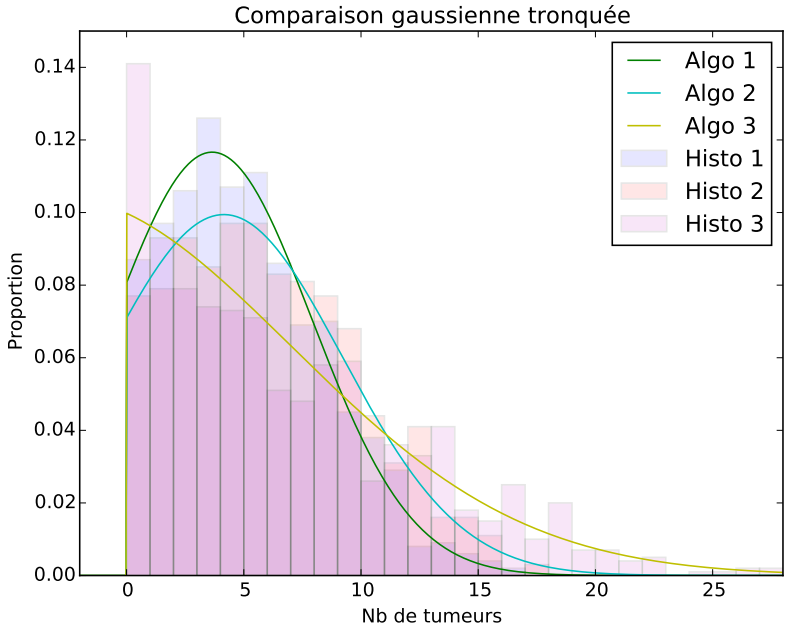
Au vue des résultats obtenus, on peut dire que l'algorithme 3 ne suit pas une loi exponentielle mais suit une loi gaussienne tronquée comme les algorithmes 1 et 2.

5.5 En résumé

Nos **3 algorithmes suivent une loi gaussienne tronquée**, qui est une variante de la loi normale.
On peut donc considérer nos échantillons comme suivant une loi normale avec les μ et σ trouvés pour la loi gaussienne tronquée.

Grâce à cela, on peut faire des tests par intervalles de confiances deux à deux entre les algorithmes.

Voici la comparaison des 3 gaussiennes tronquées trouvées :



6 Équivalence entre les algorithmes

On a montré que les 3 algorithmes suivent une loi gaussienne tronquée, de ce fait, on peut les comparer.
Pour cela nous allons utiliser un test de Fisher et ensuite si le test nous montre que les variances sont égales alors nous utiliserons un test par intervalle de confiance.

6.1 Test de Fisher

Le test de Fisher permet à partir de deux échantillons Gaussien de vérifier si leur variance est égale. Nous faisons ce test en premier car pour faire le test par intervalles de confiance il faut précédemment montrer que les variances sont identiques (d'un point de vue statistique).

Le test se formule comme suit [11]
Soit 2 échantillons gaussiens $X(\mu_x, \sigma_x^2, n_x)$ et $Y(\mu_y, \sigma_y^2, n_y)$ où $\sigma_x^2 > \sigma_y^2$.
On suppose l'hypothèse nulle $H_0 : \sigma_x^2 = \sigma_y^2$ (que les variances sont égales).
Soit

$$T = \frac{\frac{n_x}{n_x - 1} S_x^2}{\frac{n_y}{n_y - 1} S_y^2}$$

où $S_x = \sigma_x$, $S_y = \sigma_y$, n_x est le nombre d'éléments dans l'échantillon X, n_y est le nombre d'éléments dans l'échantillon Y.
$F_{\alpha/2}$ est la valeur de la loi de Fisher-Snedecor à $(n_x - 1, n_y - 1)$ d.d.l. [12] et α l'erreur de première ordre (ici5% comme préconiser dans le cours).
α à 5% signifie que 95% des éléments des deux échantillons X et Y ont une variance très proches (si le test réussi).
Dans notre cas, comme tous les échantillons ont la même taille. Ils ont donc le même degré de liberté, $F_{\alpha/2} \approx 1.132$ [13]

Si $T < F_{\alpha/2}$ alors on accepte l'hypothèse nulle et donc les variances des deux échantillons sont les mêmes.

Note : Une gaussienne tronquée est une variante d'une loi gaussienne et le test peut donc s'appliquer à elle.

Algorithme 1 comparé à l'algorithme 2 :

On prend algo1 comme Y et algo2 comme X (car $\sigma_2^2 > \sigma_1^2$).
On a donc $X(4.15, 5.05^2, 1000)$, $Y(3.65, 4.25^2, 1000)$.
$T = \frac{\frac{1000}{1000-1} 5.05^2}{\frac{1000}{1000-1} 4.25^2}$
$T = \frac{25.5025}{18.0625}$
$T \approx 1.4119$
$F_{\alpha/2} \approx 1.132$
On a que $T > F\alpha/2$, les variances sont donc différentes entre les deux algorithmes. **Il ne sont donc pas équivalents**.

11. Référence : Séance d'exercice 5.
12. d.d.l : degré de liberté, nombre d'éléments (variables aléatoires) d'un échantillon.
13. Valeur issue de la librairie scapi.stats de Python3

Algorithme 2 comparé à l'algorithme 3 :

On prend algo2 comme Y et algo3 comme X (car $\sigma_3^2 > \sigma_2^2$).
On a donc $X(-3, 10^2, 1000)$, $Y(4.15, 5.05^2, 1000)$
$T = \frac{\frac{1000}{1000-1}10^2}{\frac{1000}{1000-1}5.05^2}$
$T = \frac{100}{25.5025}$
$T \approx 3.92$
$F_{\alpha/2} \approx 1.132$
On a que $T > F\alpha/2$, les variances sont donc différentes entre les deux algorithmes. **Il ne sont donc pas équivalents**.

Algorithme 1 comparé à l'algorithme 3 :

On prend algo1 comme Y et algo3 comme X (car $\sigma_3^2 > \sigma_1^2$).
On a donc $X(-3, 10^2, 1000)$, $Y(3.65, 4.25^2, 1000)$
$T = \frac{\frac{1000}{1000-1}10^2}{\frac{1000}{1000-1}4.25^2}$
$T = \frac{100}{18.0625}$
$T \approx 5.536$
$F_{\alpha/2} \approx 1.132$
On a que $T > F\alpha/2$, les variances sont donc différentes entre les deux algorithmes. **Il ne sont donc pas équivalents**.

6.2 Test par intervalle de confiance sur deux échantillons.

Le test par intervalle de confiance permet de vérifier si deux échantillons ont la même moyenne.
Pour pouvoir faire ce test, il faut que les variances des deux échantillons soit les mêmes du point de vue statistique.
Le test est formulé comme suit :

$$|\mu_1 - \mu_2| > Z_{1-\frac{\alpha}{2}}\sqrt{\frac{\mu_1^2}{n_1} + \frac{\mu_2^2}{n_2}}$$

avec μ_1, μ_2 les moyennes respectives des variables aléatoires des échantillons 1 et 2. σ_1, σ_2 l'écart type de la variable aléatoire des échantillons 1 et 2. Et n_1, n_2 les tailles des échantillons (ici 1000).

On choisi α valant 5% comme préconisé dans le cours et donc on a $Z_{1-\frac{\alpha}{2}} = 1.96$ d'après la table correspondante.

Si l'inégalité est vérifiée alors on a que les moyennes sont différentes entre les deux échantillons.

On a montré que les variances des 3 échantillons étaient différentes, nous n'avons donc pas à faire ce test.
Les 3 algorithmes sont différents.

7 Conclusion

Durant cette analyse on a montré que l'algorithme 1 et deux suivent la même loi (une loi normale tronquée) avec certitude.
Quand à l'algorithme 3, il suit approximativement une loi gaussienne tronquée avec une moyenne négative.
Avec le test de Fisher on a montré que les 3 algorithmes avaient chacun une variance statistique différente l'un par rapport à l'autre, et étaient donc tous différent.
Les 3 algorithmes ne sont donc pas équivalents.

8 Annexe

8.1 Implémentation

Pour ce projet, j'ai décidé d'utiliser Python3 comme langage de programmation.
Pour diverses raisons, la première est qu'il est très simple d'usage et demande peu de travail pour arriver à un résultat, la seconde est que ce langage est fort utilisé dans le domaine pédagogique, on y trouve donc beaucoup d'outils qui touchent aux domaines scientifiques.