Lois de probabilités – Introduction

Ce classeur Nspire et ce manuel ont été créés par Laurae@TI-Planet.

Conditions nécessaires et obligatoires pour utiliser le classeur :
- Nspire avec CAS (Clickpad CAS, Touchpad CAS, CX CAS)
- OS 3.1 ou supérieur

Ce qui n'est pas nécessaire :
- Ndless : jailbreak de la TI-Nspire

Ce qui est optionnel mais recommandé :
- Ndless : jailbreak de la TI-Nspire
- Nover : permet d'overclocker la calculatrice (@150 MHz pour une Clickpad par exemple au lieu de 90 MHz)

Notes

Le classeur est très gourmand en ressources. Il est conseillé d'overclocker la calculatrice afin de l'ouvrir plus rapidement et d'exécuter plus rapidement les calculs sur la loi normale qui peuvent prendre plusieurs minutes. Par ailleurs, il est recommandé de rebooter régulièrement la calculatrice (faire " "+" " sur une page de calcul pour Nspire Clickpad) afin de vider la mémoire vive et donc de pouvoir utiliser encore plus longtemps le classeur et ses programmes.

Ce classeur, créé par Laurae, est destiné pour les étudiants en gestion qui ont a leur programme les différentes lois de probabilités récurrentes (Bernoulli, binomiale, Poisson, normale/Laplace-Gauss) sans trop approfondir dans les détails. Il est également utilisable pour les étudiants en Terminale Scientifique, que vous soyez en Spécialité Mathématiques ou non.

Tous les calculs sont réalisés pas-à-pas, comme par exemple pour :
- l'espérance mathématique d'une variable aléatoire (2 fonctions)
- la variance d'une variable aléatoire (2 fonctions)
- l'écart-type d'une variable aléatoire (2 fonctions)
- les conséquences d'une transformation affine d'une variable aléatoire (9 fonctions)
- processus de Bernoulli (1 fonction)
- loi binomiale (9 fonctions)
- loi de Poisson (8 fonctions)
- loi normale (12 fonctions)

Toutes ces fonctions affichent tous les calculs pas-à-pas nécessaires si c'est possible à la résolution du problème qui correspond à la fonction que vous utilisez.

Attention : les calculs présentés dans la page 6 du classeur sont généralement justes, mais ils ont été réalisés avec ma vieille version des programmes, que j'ai réactualisés le lendemain (j'ai voulu préserver le sens des calculs en les calquant dans l'ordre de présentation des programmes). Il se peut donc que des détails soient faux, mais le résultat est toujours celui escompté. Si vous trouvez des bugs, veuillez signaler Laurae sur TI-Planet sur le forum ou via message privé.

Sommaire (cliquer pour accèder)

Liste des fonctions statistiques pas-à-pas détaillée

praxb(a,b) : Calcule p(a<X<b).

prxa(a) : Calcule 1-p(X<a).

esp({x1,x2...},{p(X=x1),p(X=x2)...} : Calcule l'espérance de {(x1,p(X=x1)),(x2,p(X=x2))...}.

var1({x1,x2...},{p(X=x1),p(X=x2)...} : Calcule la variance de {(x1,p(X=x1)),(x2,p(X=x2))...} en utilisant la première formule.

var2({x1,x2...},{p(X=x1),p(X=x2)...} : Calcule la variance de {(x1,p(X=x1)),(x2,p(X=x2))...} en utilisant la seconde formule.

etype1(variance) : Calcule l'écart−type en introduisant directement la variance.

etype2({x1,x2...},{p(X=x1),p(X=x2)...} : Calcule l'écart−type de {(x1,p(X=x1)),(x2,p(X=x2))...} en utilisant la première formule.

etype3({x1,x2...},{p(X=x1),p(X=x2)...} : Calcule l'écart−type de {(x1,p(X=x1)),(x2,p(X=x2))...} en utilisant la seconde formule.

affe(a,b,esp) : Calcule l'espérance de Y du changement affine Y=aX+b en connaissant l'espérance esp de X.

affv(a,b,var) : Calcule la variance de Y du changement affine Y=aX+b en connaissant la variance var de X.

afft(a,b,et) : Calcule l'écart−type de Y du changement affine Y=aX+b en connaissant l'écart−type et de X.

sume(a,b) : Calcule l'espérance de Z qui est la somme des variables indépendantes X et Y avec a=E(X) et b=E(Y).

sumv(a,b) : Calcule la variance de Z qui est la somme des variables indépendantes X et Y avec a=V(X) et b=V(Y).

sumt(a,b) : Calcule l'écart−type de Z qui est la somme des variables indépendantes X et Y avec a=σ(X) et b=σ(Y).

suse(a,b) : Calcule l'espérance de Z qui est la soustraction des variables indépendantes X et Y avec a=E(X) et b=E(Y).

susv(a,b) : Calcule la variance de Z qui est la soustraction des variables indépendantes X et Y avec a=V(X) et b=V(Y).

sust(a,b) : Calcule l'écart−type de Z qui est la soustraction des variables indépendantes X et Y avec a=σ(X) et b=σ(Y).

Liste des fonctions Bernoulli et loi binomiale pas−à−pas détaillé

bern(p) : Processus de Bernoulli, calcule les probabilités des deux issues possibles (p étant la probabilité de réussite), l'espérance, la variance, et l'écart−type.

binof(n,p) : Formule de la loi binomiale B(n,p) avec n répétitions et p probabilité de réalisation de l'évévnement, calcule l'espérance, la variance, et l'écart−type.

binoc(n,p,x) : Formule de la loi binomiale B(n,p) avec n répétitions et p probabilité de réalisation de l'évévnement, calcule p(X=x).

bino1(n,p,x) : Formule de la loi binomiale B(n,p) avec n répétitions et p probabilité de réalisation de l'évévnement, calcule p(X≤x) directement.

bino1f(n,p,x) : Similaire à bino1(n,p,x),calcule p(X≤x) en brute force.

bino2(n,p,x) : Formule de la loi binomiale B(n,p) avec n répétitions et p probabilité de réalisation de l'évévnement, calcule p(X≥x) directement.

bino2f(n,p,x) : Similaire à bino2(n,p,x),calcule p(X≥x) en brute force.

binos(n1,n2,p) : Calcule la loi binomiale qui est la somme des variables aléatoires suivant les lois binomiales X1→B(n1,p) et X2→B(n2,p).

binoap(n,p) : Détermine l'approximation de la loi binomiale X→B(n,p) par une loi de Poisson.

binoan(n,p) : Détermine l'approximation de la loi binomiale X→B(n,p) par une loi normale.

Liste des fonctions loi de Poisson pas−à−pas détaillé

poisf(m) : Formule de la loi de Poisson de paramètre m, calcul de l'espérance, la variance, et l'écart−type.

poisc(m,x) : Formule de la loi de Poisson de paramètre m, calcul de la probabilité p(X=x).

pois1(m,x) : Formule de la loi de Poisson de paramètre m, calcule p(X≤x) directement.

pois1f(m,x) : Similaire à pois1(m,x),calcule p(X≤x) en brute force.

pois2(m,x) : Formule de la loi de Poisson de paramètre m, calcule p(X≥x) directement.

pois2f(m,x) : Similaire à pois2(m,x),calcule p(X≥x) en brute force.

poiss(m1,m2) : Calcule la loi de Poisson qui est la somme des variables aléatoires suivant les lois de Poisson X1→P(m1) et X2→P(m2).

poisan(m) : Détermine l'approximation de la loi de Poisson X→P(m) par une loi normale.

Liste des fonctions loi normale pas−à−pas détaillé

normf(m,σ) : Formule de la loi normale de paramètre E(X)=m et σ(X)=σ, calcul de l'espérance, la variance, et l'écart−type.

normcr(m,σ) : Centrage et réduction de la loi normale N(m,σ) tel que T→N(0,1).

normcr1(m,σ,x) : Centrage, réduction, résolution de p(X≤x) avec X→N(m,σ).

normcr2(m,σ,x) : Centrage, réduction, résolution de p(X≥x) avec X→N(m,σ).

normcr3(m,σ,x,y) : Centrage, réduction, résolution de p(z≤X≤y) avec X→N(m,σ).

norms1(m,σ,p) : Centrage, réduction, résolution de x pour p(Y≤x)=p avec Y→N(m,σ).

norms2(m,σ,p) : Centrage, réduction, résolution de x pour p(Y≥x)=p avec Y→N(m,σ).

norms3(m,σ,p) : Centrage, réduction, résolution de x pour p(z≥Y≥x)=p avec Y→N(m,σ).

normsu(m1,σ1,m2,σ2) : Calcule la loi normale qui est la somme des variables aléatoires suivant les lois normales X1→N(m1,σ1) et X2→N(m2,σ2).

normso(m1,σ1,m2,σ2) : Calcule la loi normale qui est la soustraction des variables aléatoires suivant les lois normales X1→N(m1,σ1) et X2→N(m2,σ2).

normaf(m,σ,a,b) : Calcule la loi normale qui est le changement affine de la variable aléatoire suivant la loi normale X→N(m,σ).

normin1(m,σ,p,n) : Centrage, réduction, résolution de x pour p(Y≤x+n)=p avec Y→N(m,σ).

normin2(m,σ,p,n) : Centrage, réduction, résolution de x pour p(Y≥x+n)=p avec Y→N(m,σ).

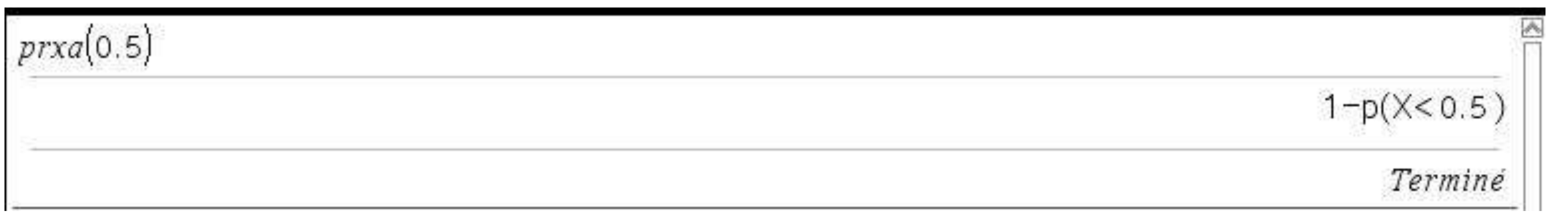
Statistiques pas-à-pas

L'objectif principal de cette série de programmes est de déterminer rapidement l'espérance, la variance, et l'écart-type d'une variable aléatoire, qui peut être simple, résultante d'une fonction affine, résultante d'une somme de variables aléatoires, ou résultante d'une soustraction de variables aléatoires. Plusieurs formules différentes peuvent être utilisées, aboutissant au même résultat. La maitrise basique de l'utilisation des listes sur Nspire est nécessaire pour pouvoir utiliser la plupart des fonctions de cette catégorie. Savoir créer une liste est suffisant, savoir utiliser des variables contenant des listes est encore mieux pour ne pas avoir à retaper chacune des listes une par une, caractère par caractère.

1 – praxb(a,b)

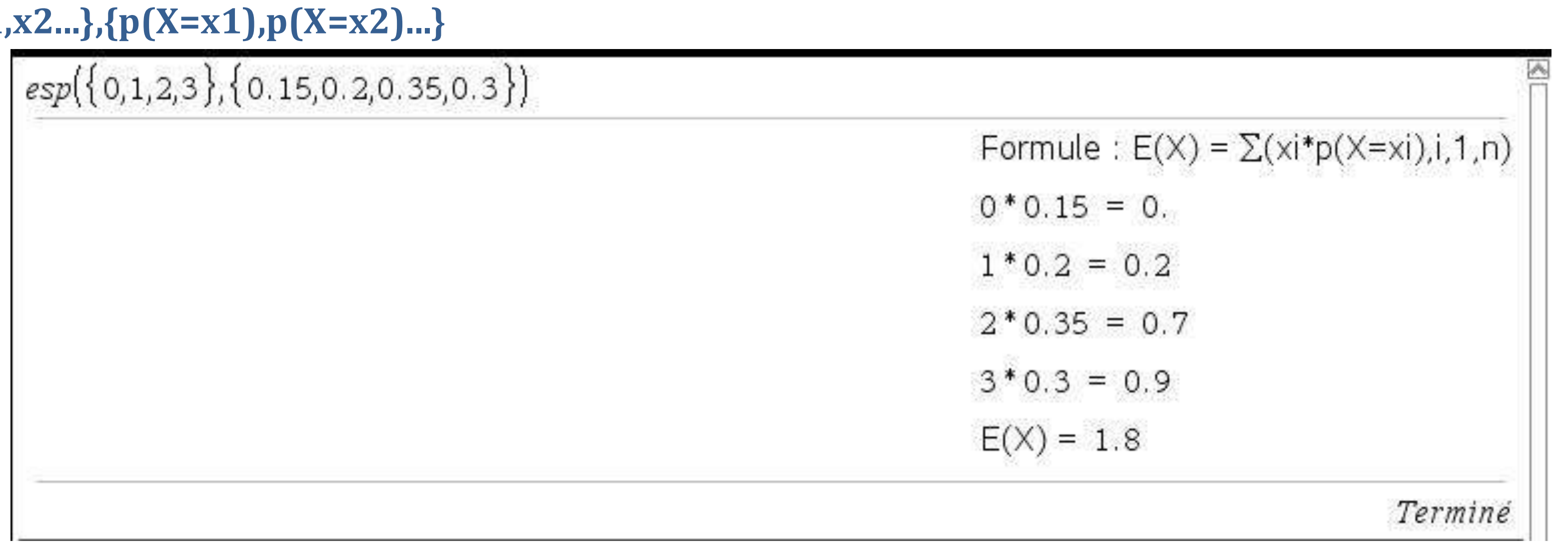
La fonction *praxb(a,b)* (pour « Probabilité de a<x<b ») permet d'éclater le calcul de p(a<X<b) en le transformant dans un calcul possible (pour les lois de probabilités continues) en *p(X<b)-p(X<a)*.

2 – prxa(a)



La fonction *praxa(a)* (pour « Probabilité de x>a ») permet de transformer ce calcul en calcul utilisable immédiatement par exemple pour les lois normales, en transformant *p(X>a)* en *1-p(X<a)*.

3 – esp({x1,x2…},{p(X=x1),p(X=x2)…}



La fonction *esp({x1,x2…},{p(X=x1,p(X=x2)…)* (pour « Espérance ») permet de calculer l'espérance d'une variable aléatoire, en spécifiant sous forme de listes les éventualités et les fréquences associées à ces éventualités, en affichant la formule, et en calculant pas-à-pas *E(X)*.

Formule de l'espérance : $E(X) = \sum_{i=0}^{n}(x_i * p(X = x_i))$

Dans cet exemple, nous avons ce tableau ci-contre :

Eventualité i *(xi)*	0	1	2	3	
Fréquence i *(p(X=xi))*	0.15	0.20	0.35	0.30	
*xi*p(X=xi)*	0.00	0.20	0.70	0.90	*E(X)* = 1.80

Et par conséquent, l'espérance par application de la formule est :
*E(X) = 0*0.15 + 1*0.20 + 2*0.35 + 3*0.30 = 0.00 + 0.20 + 0.70 + 0.90 = 1.80*

On pourrait par exemple l'interpréter comme cela par un exercice : soit une loterie où l'on peut gagner jusqu'à trois lots, on a 15% de chance de ne rien gagner, 20% de chance de gagner 1 lot, 35% de chance de gagner 2 lots, et 30% de chance de gagner 3 lots.

Il y a donc 4 éventualités possibles :
- gagner 0, 1, 2, ou 3 lots : donc mettre *{0,1,2,3}*
- avec des fréquences respectives de 15%, 20%, 35%, et 30% : donc mettre *{0.15,0.2,0.35,0.3}*
Et donc pour obtenir pas-à-pas l'espérance, il faut faire *esp({0,1,2,3},{0.15,0.2,0.35,0.3})*.

4 – var1({x1,x2…},{p(X=x1),p(X=x2)…}

```
var1({0,1,2,3},{0.15,0.2,0.35,0.3})
                    Formule : V(X) = Σ[((xi−E(X))^2)*p(X=xi)]
                    E(X) = 1.8
                    [(0−1.8)^2]*0.15 = 0.486
                    [(1−1.8)^2]*0.2 = 0.128
                    [(2−1.8)^2]*0.35 = 0.014
                    [(3−1.8)^2]*0.3 = 0.432
                    V(X) = 1.06
                                                    Terminé
```

La fonction *var1({x1,x2…},{p(X=x1,p(X=x2)…)* (pour « Variance, première méthode ») permet de calculer la variance d'une variable aléatoire, en spécifiant sous forme de listes les éventualités et les fréquences associées à ces éventualités, en affichant la première formule de la variance, et en calculant pas-à-pas *V(X)*.

Première formule de la variance : $V(X) = \sum_{i=0}^{n}((x_i - E(X))^2 * p(X = x_i))$

Dans cet exemple, nous avons ce tableau ci-contre :

Eventualité i *(xi)*	0	1	2	3	
Fréquence i *(p(X=xi))*	0.15	0.20	0.35	0.30	
*xi*p(X=xi)*	0.00	0.20	0.70	0.90	*E(X)* = 1.80
*(xi-E(X))²*p(X=xi)*	0.486	0.128	0.014	0.432	*V(X)* = 1.06

Et par conséquent, la variance par application de la première formule est :
*V(X) = (0-1.8)²*0.15 + (1-1.8)²*0.20 + (2-1.8)²*0.35 + (3-1.8)²*0.30 + 0.486 + 0.128 + 0.014 + 0.432 = 1.06*

5 – var2({x1,x2…},{p(X=x1),p(X=x2)…}

```
var2({0,1,2,3},{0.15,0.2,0.35,0.3})
                    Formule : V(X) = Σ[(xi^2)*p(X=xi)]−(E(X))^2
                    E(X) = 1.8
                    [E(X)]^2 = 3.24
                    (0^2)*0.15 = 0.
                    (1^2)*0.2 = 0.2
                    (2^2)*0.35 = 1.4
                    (3^2)*0.3 = 2.7
                    V(X) = 4.3 − 3.24
                    V(X) = 1.06
                                                    Terminé
```

La fonction *var2({x1,x2…},{p(X=x1,p(X=x2)…)* (pour « Variance, seconde méthode ») permet de calculer la variance d'une variable aléatoire, en spécifiant sous forme de listes les éventualités et les fréquences associées à ces éventualités, en affichant la seconde formule de la variance, et en calculant pas-à-pas *V(X)*.

Seconde formule de la variance : $V(X) = \sum_{i=0}^{n}(x_i^2 * p(X = x_i)) - E(X)^2$

Dans cet exemple, nous avons ce tableau ci-contre :

Eventualité i *(xi)*	0	1	2	3	
Fréquence i *(p(X=xi))*	0.15	0.20	0.35	0.30	
*xi*p(X=xi)*	0.00	0.20	0.70	0.90	*E(X)* = 1.80
*xi²*p(X=xi)*	0.00	0.20	1.40	2.70	*V(X)* = 1.06

Et par conséquent, la variance par application de la seconde formule est :
*V(X) = (0²*0.15 + 1²*0.2 + 2²*0.25 + 3²*0.30) – 1.80² = (0.00 + 0.20 + 1.40 + 2.70 = 4.30) – 3.24 = 1.06*

6 – etype1(variance)

```
etype1(1.06)
                    Formule : σ(X) = √V(X)
                    σ(X) = √(1.06) = 1.02956
                                                    Terminé
```

La fonction *etype1(variance)* (pour « Ecart-type, première méthode ») calcule l'écart-type en appliquant directement la formule de la variance.
Il suffit tout simplement de rentrer la variance.

Première formule de l'écart-type : $\sigma(X) = \sqrt{V(X)}$

Dans l'exemple précédent, on peut reprendre *V(X) = 1.06* pour calculer directement l'écart-type qui est la racine carrée de la variance. Ce qui nous donne *1.02956* en valeur approchée.

Eventualité i *(xi)*	0	1	2	3	
Fréquence i *(p(X=xi))*	0.15	0.20	0.35	0.30	
*xi*p(X=xi)*	0.00	0.20	0.70	0.90	*E(X)* = 1.80
*xi²*p(X=xi)*	0.00	0.20	1.40	2.70	*V(X)* = 1.06
					σ(X) = 1.02956

7 – etype2({x1,x2…},{p(X=x1),p(X=x2)…}

```
etype2({0,1,2,3},{0.15,0.2,0.35,0.3})
                         Formule : σ(X) = √ (Σ[((xi−E(X))^2)*p(X=xi)])
                         E(X) = 1.8
                         [( 0 − 1.8 )^2]* 0.15 = 0.486
                         [( 1 − 1.8 )^2]* 0.2 = 0.128
                         [( 2 − 1.8 )^2]* 0.35 = 0.014
                         [( 3 − 1.8 )^2]* 0.3 = 0.432
                         V(X) = 1.06
                         σ(X) = √ ( 1.06 ) = 1.02956
                                                              Terminé
```

La fonction *etype2({x1,x2…},{p(X=x1,p(X=x2)…)* ((pour « Ecart-type, seconde méthode ») calcule l'écart-type en appliquant la seconde formule de l'écart-type. Cette fonction est similaire à la fonction *var1({x1,x2…},{p(X=x1,p(X=x2)…)* (pour « Variance, première méthode »). La seule chose supplémentaire est le calcul de l'écart-type après avoir automatiquement calculé la variance en utilisant la première formule de la variance.

Seconde formule de l'écart-type : $\sigma(X) = \sqrt{\sum_{i=0}^{n}((x_i - E(X))^2) * p(X = x_i)}$

Dans l'exemple précédent, on peut reprendre *E(X) = 1.80* et *V(X) = 1.06* pour calculer l'écart-type.

Eventualité i *(xi)*	0	1	2	3	
Fréquence i *(p(X=xi))*	0.15	0.20	0.35	0.30	
*xi*p(X=xi)*	0.00	0.20	0.70	0.90	*E(X)* = 1.80
*(xi-E(X)²)*p(X=xi)*	0.486	0.128	0.014	0.432	*V(X)* = 1.06
					σ(X) = 1.02956

On a donc bien de nouveau comme écart-type 1.02956.

8 – etype3({x1,x2…},{p(X=x1),p(X=x2)…}

```
etype3({0,1,2,3},{0.15,0.2,0.35,0.3})
                       Formule : σ(X) = √ (Σ[(xi^2)*p(X=xi)]−(E(X))^2)
                       E(X) = 1.8
                       [E(X)]^2 = 3.24
                       ( 0 ^2)* 0.15 = 0.
                       ( 1 ^2)* 0.2 = 0.2
                       ( 2 ^2)* 0.35 = 1.4
                       ( 3 ^2)* 0.3 = 2.7
                       V(X) = 4.3 − 3.24
                       V(X) = 1.06
                       σ(X) = √ ( 1.06 ) = 1.02956
                                                              Terminé
```

La fonction *etype2({x1,x2…},{p(X=x1,p(X=x2)…)* ((pour « Ecart-type , troisième méthode ») calcule l'écart-type en appliquant la seconde formule de l'écart-type. Cette fonction est similaire à la fonction *var2({x1,x2…},{p(X=x1,p(X=x2)…)* (pour « Variance, seconde méthode »). La seule chose supplémentaire est le calcul de l'écart-type après avoir automatiquement calculé la variance en utilisant la seconde formule de la variance.

Troisième formule de l'écart-type : $\sigma(X) = \sqrt{\sum_{i=0}^{n}(x_i^2 * p(X = x_i) - E(X)^2}$

Dans l'exemple précédent, on peut reprendre *E(X) = 1.80* et *V(X) = 1.06* pour calculer l'écart-type.

Eventualité i *(xi)*	0	1	2	3	
Fréquence i *(p(X=xi))*	0.15	0.20	0.35	0.30	
*xi*p(X=xi)*	0.00	0.20	0.70	0.90	*E(X)* = 1.80
*xi²*p(X=xi)*	0.00	0.20	1.40	2.70	*V(X)* = 1.06
					σ(X) = 1.02956

On a donc bien de nouveau comme écart-type 1.02956.

9 – affe(a,b,esp)

affe(2,3400,3500)

E(Y) = E(aX+b) = E(2 X+ 3400)

E(Y) = a*E(X)+b = 2 * 3500 + 3400 = 10400

Terminé

La fonction *affe(a,b,esp)* (pour « Espérance du changement affine d'une variable aléatoire ») permet de calculer l'espérance de la variable aléatoire *Y* issue du changement affine *aX+b* de la variable aléatoire *X* connaissant l'espérance *esp* de la variable aléatoire *X*.

La formule est la suivante : $E(Y) = aE(X) + b$

Dans l'exemple ci-contre, on a *esp = 3500*, *a = 2*, et *b = 3400*. On a donc *E(Y) = E(2X+3400) = 2*3500+3400 = 10400*.

On peut également utiliser des arguments négatifs, tels que b = -3400 :

affe(2,-3400,3500)

E(Y) = E(aX+b) = E(2 X+ -3400)

E(Y) = a*E(X)+b = 2 * 3500 + -3400 = 3600

Terminé

Ici, *a* est resté constant, ainsi que *esp*. On a juste pris l'opposé de *b*, qui devient *-3400*.

10 – affv(a,b,var)

affv(2,-3400,360000)

V(Y) = V(aX+b) = V(2 X+ -3400)

V(Y) = V(X)*a^2 = 4 * 360000 = 1440000

Terminé

La fonction *affv(a,b,var)* (pour « Variance du changement affine d'une variable aléatoire ») permet de calculer la variance de la variable aléatoire *Y* issue du changement affine *aX+b* de la variable aléatoire *X* connaissant la variance *var* de la variable aléatoire *X*.

La formule est la suivante : $V(Y) = a^2 * V(X)$

Dans l'exemple ci-contre, on a *var = 360000*, *a = 2*, et *b = -3400*. On a donc *V(Y) = V(2X-3400) = 2²*360000 = 1440000*.

11 – afft(a,b,et)

afft(2,-3400,600)

σ(Y) = σ(aX+b) = σ(2 X+ -3400)

σ(Y) = a*σ(X) = 2 * 600 = 1200

Terminé

La fonction *afft(a,b,et)* (pour « Ecart-type du changement affine d'une variable aléatoire ») permet de calculer l'écart-type de la variable aléatoire *Y* issue du changement affine *aX+b* de la variable aléatoire *X* connaissant l'écart-type *et* de la variable aléatoire *X*.

La formule est la suivante : $\sigma(Y) = a * \sigma(X)$

Dans l'exemple ci-contre, on a *et = 360000*, *a = 2*, et *b = -3400*. On a donc *σ (Y) = σ (2X-3400) = 2*600 = 1200*.

12 – sume(a,b)

sume(3600,3600)

E(Z) = E(X+Y)

E(Z) = E(X)+E(Y) = 3600 + 3600 = 7200

Terminé

La fonction *sume(a,b)* (pour « Espérance de la somme de deux variables aléatoires ») permet de calculer l'espérance *E(Z)* issue de la somme de deux variables aléatoires *X* et *Y*.

La formule est la suivante : $E(Z) = E(X) + E(Y)$

Dans l'exemple ci-contre, on a *E(X) = a = 3600* et *E(Y) = b = 3600*, ce qui donne évidemment *E(Z) = E(X)+E(Y) = 7200*.

13 – sumv(a,b)

```
sumv(360000,360000)
                    V(Z) = V(X+Y)
                    V(Z) = V(X)+V(Y) =  360000 + 360000  =  720000
                                                          Terminé
```

La fonction *sumv(a,b)* (pour « Variance de la somme de deux variables aléatoires ») permet de calculer la variance *V(Z)* issue de la somme de deux variables aléatoires *X* et *Y*.

La formule est la suivante : $V(Z) = V(X) + V(Y)$

Dans l'exemple ci-contre, on a *V(X) = a = 360000* et *V(Y) = b = 360000*, ce qui donne évidemment *E(Z) = E(X)+E(Y) = 720000*.

14 – sumt(a,b)

```
sumt(600,600)
            σ(Z) = σ(X+Y)
            σ(Z) = √ [(σ(X))^2+(σ(Y))^2] = √ ( 600 ^2+ 600 ^2) =  600·√2
                                                          Terminé
```

La fonction *sumt(a,b)* (pour « Ecart-type de la somme de deux variables aléatoires ») permet de calculer l'écart-type *σ (Z)* issue de la somme de deux variables aléatoires *X* et *Y*.

La formule est la suivante : $\sigma(Z) = \sqrt{\sigma(X)^2 + \sigma(Y)^2}$

Dans l'exemple ci-contre, on a *σ (X) = a = 600* et *σ (Y) = b = 600*, ce qui donne évidemment *σ (Z) = σ (X)+ σ (Y) = 600* √ 2 = 848.528*.

Pour obtenir le résultat sous forme approchée, utilisez soit :
- un nombre à virgule
- faites Ctrl+Enter au lieu de Enter
- écrivez approx(...) et remplacez … par la fonction que vous utilisez

15 – suse(a,b)

```
suse(3600,3600)
                         E(Z) = E(X−Y)
                         E(Z) = E(X)−E(Y) =  3600 − 3600  =  0
                                                          Terminé
```

La fonction *suse(a,b)* (pour « Espérance de la soustraction de deux variables aléatoires ») permet de calculer l'espérance *E(Z)* issue de la soustraction de deux variables aléatoires *X* et *Y*.

La formule est la suivante : $E(Z) = E(X) - E(Y)$

Dans l'exemple ci-contre, on a *E(X) = a = 3600* et *E(Y) = b = 3600*, ce qui donne évidemment *E(Z) = E(X)-E(Y) = 0*.

16 – susv(a,b)

```
susv(360000,360000)
                    V(Z) = V(X−Y)
                    V(Z) = V(X)+V(Y) =  360000 + 360000  =  720000
                                                          Terminé
```

La fonction *susv(a,b)* (pour « Variance de la soustraction de deux variables aléatoires ») permet de calculer la variance *V(Z)* issue de la soustraction de deux variables aléatoires *X* et *Y*.

La formule est la suivante : $V(Z) = V(X) + V(Y)$

Dans l'exemple ci-contre, on a *V(X) = a = 360000* et *V(Y) = b = 360000*, ce qui donne évidemment *V(Z) = V(X)+V(Y) = 720000*.

17 – sust(a,b)

```
sust(600,600)
            σ(Z) = σ(X−Y)
            σ(Z) = √ [(σ(X))^2+(σ(Y))^2] = √ ( 600 ^2+ 600 ^2) =  600·√2
                                                          Terminé
```

La fonction *sust(a,b)* (pour « Ecart-type de la soustraction de deux variables aléatoires ») permet de calculer l'écart-type σ *(Z)* issue de la soustraction de deux variables aléatoires *X* et *Y*.

La formule est la suivante : $\sigma(Z) = \sqrt{\sigma(X)^2 + \sigma(Y)^2}$

Dans l'exemple ci-contre, on a σ *(X) = a = 600* et σ *(Y) = b = 600*, ce qui donne évidemment σ *(Z) =* σ *(X)+* σ *(Y) = 600** √ *2 = 848.528*.

Bernoulli et loi binomiale pas-à-pas

Bernoulli et la loi binomiale vont toujours ensemble. L'objectif de cette série de programmes est de maitriser le processus de Bernoulli, puis d'en étudier un cas général résultant de l'application à la chaine du processus de Bernoulli, ce qui aboutit à la loi binomiale. Il faut maitriser le processus de Bernoulli afin d'étudier la loi binomiale qui en découle selon les énoncés. Les programmes vous permettront ensuite de réaliser toutes sortes de calculs que vous pouvez retrouver en quelques secondes. La connaissance des formules n'est pas exigée pour utiliser ces programmes, mais il vaut mieux les connaitre pour ne pas avoir de mauvaises surprises (peut être que certains calculs ne sont pas assez détaillés). L'approximation de la loi binomiale par une autre loi nécessite la maitrise des autres lois en question. Mieux vous maitrisez la loi binomiale, plus les programmes que vous utiliserez seront compréhensibles. En effet, il faut rentabiliser le temps que vous utiliserez à trouver la fonction à utiliser selon votre énoncé. L'usage de ces programmes est de ne pas gaspiller votre temps, mais de vous assister.

1 – bern(p)

```
bern(0.05)
                                   p(X=x1) = 0.05
                                   p(X=x2) = 0.95
                                   E(X) = 0.05
                                   V(X) = 0.05*(0.95) = 0.0475
                                   σ(X) = √V(X) = √0.0475 = 0.217945
                                                                 Terminé
```

La fonction *bern(p)* (pour « Processus de Bernoulli ») permet de calculer la fréquence des deux issues possibles, l'espérance de votre processus, sa variance, et son écart-type, le tout pas-à-pas.
Il suffira de rentrer la probabilité *p* de l'événement qui peut se réaliser.

Par exemple, une pomme à 5% de chance de tomber sur votre tête. La probabilité donc qu'une pomme tombe sur votre tête est de *p = 0.05*.

Formule de l'espérance : $E(X) = p$
Formule de la variance : $V(X) = p * (1 - p)$
Formule de l'écart-type : $\sigma(X) = \sqrt{V(X)} = \sqrt{p * (1 - p)}$

2 – binof(n,p)

```
binof(100,0.05)
              Formule : B(100,0.05) ↔ p(X=x) = nCr(100,x)*(0.05^x)*0.95^(100-x)
              nCr(n,p) = (n!)/(p!(n-p)!)
              E(X) = 100*0.05 = 5.
              V(X) = 100*0.05*0.95 = 4.75
              σ(X) = √4.75 = 2.17945
                                                                 Terminé
```

La fonction *binof(n,p)* (pour « Formule de la loi binomiale ») permet de déterminer la formule de la loi binomiale selon le nombre de répétitions du processus de Bernoulli incriminé par la fréquence de réalisation de ce processus. En outre, l'espérance, la variance, et l'écart-type de la loi binomiale de paramètre *n* et *p* sont calculés de manière détaillée.

Par exemple, vous passez 100 fois au-dessous d'un arbre. A chaque fois que vous passez sous un arbre, il y a 5% de chance qu'une pomme vous tombe dessus. Par conséquent, le nombre de répétition du processus de Bernoulli de fréquence *p = 0.05 (5%)* est de *100*.
En moyenne, 5 pommes vous tomberont dessus.

Formule de l'espérance : $E(X) = n * p$
Formule de la variance : $V(X) = n * p * (1 - p)$
Formule de l'écart-type : $\sigma(X) = \sqrt{V(X)} = \sqrt{n * p * (1 - p)}$

3 – binoc(n,p,x)

```
binoc(100,0.05,5)
   Formule : B(100,0.05) ↔ p(X=x) = nCr(100,x)*(0.05^x)*0.95^(100-x)
   p(X=5) = nCr(100,5)*(0.05^5)*((0.95)^95) = 75287520*3.125E-7*0.007651 = 0.180018
   binomPdf(100,0.05,5) = 0.180018
                                                                 Terminé
```

La fonction *binoc(n,p,x)* (pour « Calcul d'une probabilité de la loi binomiale ») permet de déterminer la formule de la loi binomiale selon le nombre de répétitions du processus de Bernoulli incriminé par la fréquence de réalisation de ce processus, et permet de calculer l'événement *p(X=x)*.

Ici en reprenant l'exemple précédent, on veut déterminer la probabilité que sur les *n=100* passages en-dessous d'un arbre, qu'exactement *x=5* pommes nous tombent dessus sachant qu'une pomme à *p=0.05* de chance de nous tomber dessus. La probabilité est ici de 18%.
La fonction utilisable (*binomPdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

4 – bino1(n,p,x)

```
bino1(100,0.05,1)
        Formule : B( 100 , 0.05 ) ↔ p(X=x) = nCr( 100 , x )*( 0.05 ^ x )* 0.95 ^( 100−x )
        p(X≤ 1 ) = binomCdf( 100 , 0.05 , 0 , 1 ) =  0.037081
                                                                          Terminé
```

La fonction *bino1(n,p,x)* (pour « Calcul de la probabilité p(X≤x) de la loi binomiale ») permet de déterminer la formule de la loi binomiale selon le nombre de répétitions du processus de Bernoulli incriminé par la fréquence de réalisation de ce processus, et permet de calculer l'événement *p(X≤x)*.

Ici en reprenant l'exemple précédent, on veut déterminer la probabilité que sur les *n=100* passages en-dessous d'un arbre, que moins de *x=1* pomme nous tombe dessus sachant qu'une pomme à *p=0.05* de chance de nous tomber dessus. La probabilité est ici de *3.71%*.
La fonction utilisable (*binomCdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

5 – bino1f(n,p,x)

```
bino1f(100,0.05,1)
        Formule : B( 100 , 0.05 ) ↔ p(X=x) = nCr( 100 , x )*( 0.05 ^ x )* 0.95 ^( 100−x )
        Méthode brute force
        p(X= 0 ) =  0.005921
        p(X= 1 ) =  0.031161
        p(X≤ 1 ) =  0.037081
        p(X≤ 1 ) = binomCdf( 100 , 0.05 , 0 , 1 ) =  0.037081
                                                                          Terminé
```

La fonction *bino1f(n,p,x)* (pour « Calcul de la probabilité p(X≤x) de la loi binomiale, méthode brute force ») permet de déterminer la formule de la loi binomiale selon le nombre de répétitions du processus de Bernoulli incriminé par la fréquence de réalisation de ce processus, et permet de calculer l'événement *p(X≤x)* en calculant toutes les probabilités de 0 à x, puis en faisant leur somme.

L'exemple est le même. La probabilité est de *3.71%*, issue *de p(X=0) = 0.59%* et *p(X=1) = 3.12%*.
La fonction utilisable (*binomCdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

6 – bino2(n,p,x)

```
bino2(100,0.05,1)
        Formule : B( 100 , 0.05 ) ↔ p(X=x) = nCr( 100 , x )*( 0.05 ^ x )* 0.95 ^( 100−x )
        p(X≥ 1 ) = 1−p(X< 1 ) = 1−binomCdf( 100 , 0.05 , 0 , 0 ) = 1− 0.005921  =  0.994079
                                                                          Terminé
```

La fonction *bino2(n,p,x)* (pour « Calcul de la probabilité p(X≥x) de la loi binomiale ») permet de déterminer la formule de la loi binomiale selon le nombre de répétitions du processus de Bernoulli incriminé par la fréquence de réalisation de ce processus, et permet de calculer l'événement *p(X≥x)*.

Ici en reprenant l'exemple précédent, on veut déterminer la probabilité que sur les *n=100* passages en-dessous d'un arbre, que au moins de *x=1* pommes nous tombent dessus sachant qu'une pomme à *p=0.05* de chance de nous tomber dessus. La probabilité est ici de *99.41%*.
La fonction utilisable (*1-binomCdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

7 – bino2f(n,p,x)

```
bino2f(100,0.05,1)
        Formule : B( 100 , 0.05 ) ↔ p(X=x) = nCr( 100 , x )*( 0.05 ^ x )* 0.95 ^( 100−x )
        Méthode brute force
        p(X= 0 ) =  0.005921
        p(X< 1 ) =  0.005921
        p(X≥ 1 ) = 1−p(X< 1 ) =  0.994079
        p(X≥ 1 ) = 1−p(X< 1 ) = 1−binomCdf( 100 , 0.05 , 0 , 0 ) = 1− 0.005921  =  0.994079
                                                                          Terminé
```

La fonction *bino2f(n,p,x)* (pour « Calcul de la probabilité p(X≥x) de la loi binomiale, méthode brute force ») permet de déterminer la formule de la loi binomiale selon le nombre de répétitions du processus de Bernoulli incriminé par la fréquence de réalisation de ce processus, et permet de calculer l'événement *p(X≥x)* en calculant toutes les probabilités de 0 à x-1, puis en faisant leur somme.

L'exemple est le même. La probabilité est de *99.41%*, issue *de p(X<1) = p(X=0) = 0.59%*.
La fonction utilisable (1-*binomCdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

8 – binos(n1,n2,p)

```
binos(100,100,0.05)
                    X1→B( 100 , 0.05 ) et X2→B( 100 , 0.05 )
                    Si X1 et X2 indépendantes : X1+X2→B( 200 , 0.05 )
                                                                 Terminé
```

La fonction *binos(n1,n2,p)* (pour « Somme de deux variables aléatoires binomiales ») permet de calculer la loi binomiale B issue de la somme de deux variables aléatoires binomiales *X1* et *X2*.

La formule est la suivante : $B(n,p) = \begin{cases} n = n1 + n2 \\ p = p \end{cases}$

Dans l'exemple ci-contre, on a *n1 = 100* et *n2 = 100 et p = 0.05*, ce qui donne évidemment *B(100+100,0.05) = B(200,0.05)*.

9 – binoap(n,p)

```
binoap(120,0.01)
                    Approximation de la loi binomiale par une loi de Poisson
                    Paramètres : n = 120 et p = 0.01
                    n≥30
                    np*(1-p)<5 ↔ 1.188 <5
                    p≤0.1
                    Les 3 conditions sont vérifiées
                    X→B( 120 , 0.01 ) →P( 120 * 0.01 ) = P( 1.2 )
                                                                 Terminé
```

La fonction *binoap(n,p)* (pour « Approximation de la loi binomiale par une loi de Poisson ») permet de déterminer si la loi binomiale de paramètres *n* et *p* peut être approchée par une loi de Poisson.

Critères de l'approximation de la loi binomiale par une loi de Poisson : $Si \begin{cases} n \geq 30 \\ n * p * (1-p) < 5 \\ p \leq 0.1 \end{cases}, alors\ B(n,p) \to P(n*p)$

Ici, on tente une approximation de la loi binomiale de paramètres *n = 120* et *p = 0.01*. On a *n≥30*, *np(1-p)<5*, et *p≤0.1*. Par conséquent, on peut approximer la loi binomiale par la loi de Poisson de paramètre *120*0.01 = 1.2*.

10 – binoan(n,p)

```
binoan(120,0.3)
        Approximation de la loi binomiale par une loi de normale
        Paramètres : n = 120 et p = 0.3
        n≥30
        np*(1-p)≥5 ↔ 25.2 ≥5
        Les 2 conditions sont vérifiées
        X→B( 120 , 0.3 ) →N([m= 120 * 0.3 ],[σ= √ ( 120 * 0.3 * 0.7 ]) = N(m= 36. ,σ= 5.01996 )
                                                                 Terminé
```

La fonction *binoan(n,p)* (pour « Approximation de la loi binomiale par une loi normale ») permet de déterminer si la loi binomiale de paramètres *n* et *p* peut être approchée par une loi normale.

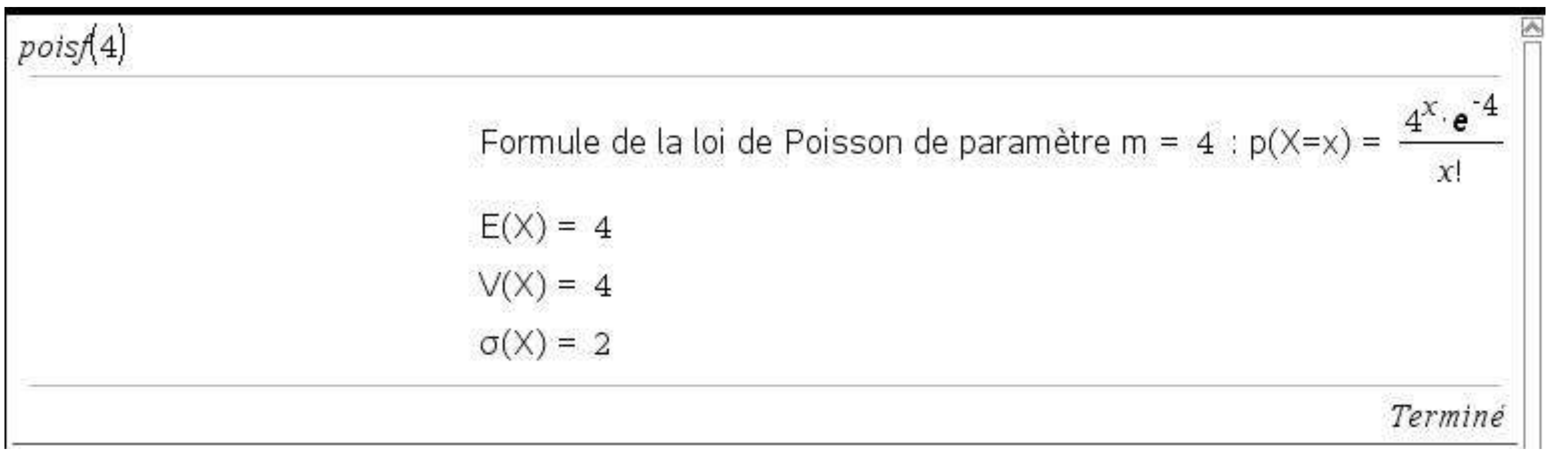
Critères de l'approximation de la loi binomiale par une loi normale : $Si \begin{cases} n \geq 30 \\ n * p(1-p) \geq 5 \end{cases}, alors\ B(n,p) \to N \begin{cases} m = n * p \\ \sigma = \sqrt{n * p * (1-p)} \end{cases}$

Ici, on tente une approximation de la loi binomiale de paramètres *n = 120* et *p = 0.30*. On a *n≥30*, *np(1-p) ≥5.*Par conséquent, on peut approximer la loi binomiale par la loi normale de paramètres *m = 120*0.3 = 36* et *p = √(120*0.3*0.7) = 5.02*.

Loi de Poisson pas-à-pas

L'objectif de cette série de programmes est de maitriser la loi de Poisson. Les calculs possibles sont identiques à la loi binomiale, il ne restera qu'à connaitre les formules à utiliser, qui vous sont rappelées lors du calcul par les programmes. La maitrise de la formule de la loi de Poisson est essentielle, et il faut savoir l'utiliser parfaitement si vous voulez l'utiliser en dehors du programme. De ces formules découlent des calculs de probabilités aussi simples que celles de la loi binomiale.

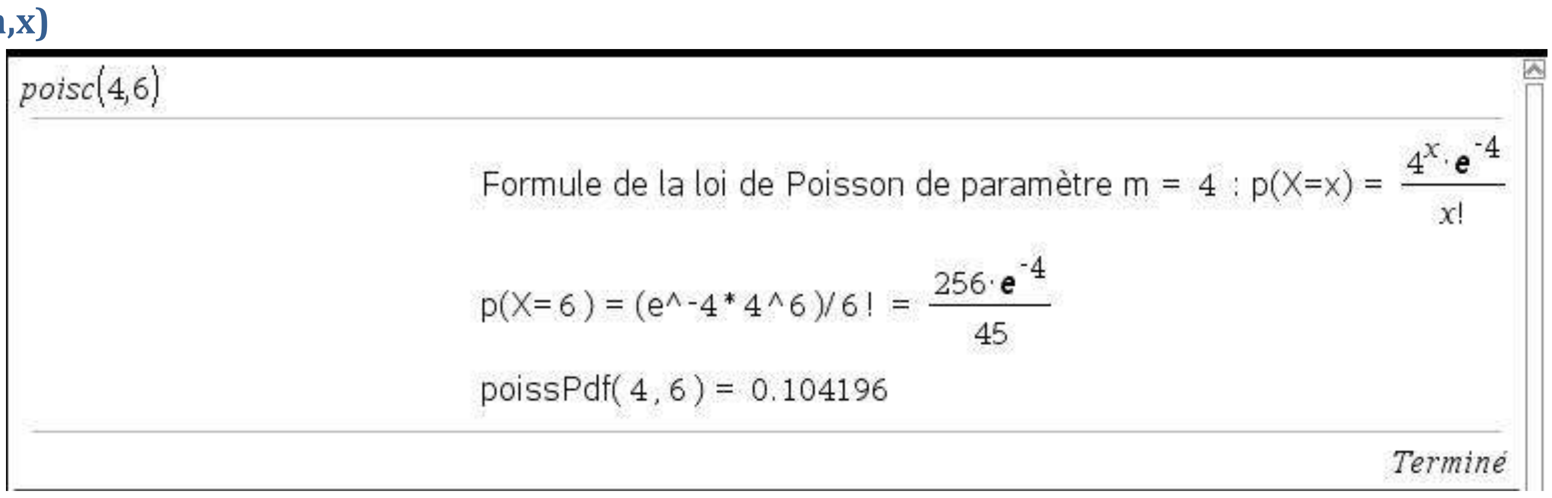
1 – poisf(m)



La fonction *poisf(m)* (pour « Formule de la loi de Poisson ») permet de calculer la formule de la loi de Poisson de paramètre *m*, son espérance, sa variance, et son écart-type.
Il suffira de rentrer le paramètre *m* de la loi de Poisson en question.

Formule de la loi de Poisson : $p(X = x) = \frac{e^{-m} m^{x}}{x!}$
Formule de l'espérance : $E(X) = m$
Formule de la variance : $V(X) = m$
Formule de l'écart-type : $\sigma(X) = \sqrt{V(X)} = \sqrt{m}$

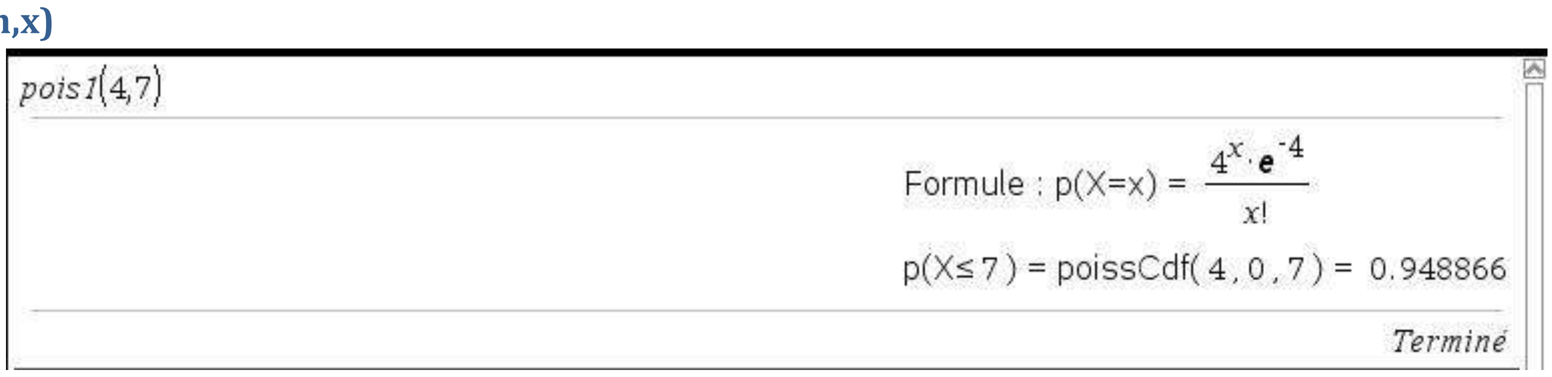
2 – poisc(m,x)



La fonction *poisc(m,x)* (pour « Calcul d'une probabilité de la loi de Poisson ») permet de déterminer la formule de la loi de Poisson selon le paramètre *m*, et permet de calculer l'événement *p(X=x)*.

La fonction utilisable (*poissPdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

3 – pois1(m,x)



La fonction *pois1(m,x)* (pour « Calcul de la probabilité p(X≤x) de la loi de Poisson ») permet de déterminer la formule de la loi de Poisson selon le paramètre *m*, et permet de calculer l'événement *p(X≤x)*.

La fonction utilisable (*poissCdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

4 – pois1f(m,x)

pois1f(4,7)

$$\text{Formule : p(X=x)} = \frac{0.018316\cdot(4.)^{x}}{x!}$$

Méthode brute force

p(X= 0.) = 0.018316

p(X= 1.) = 0.073263

p(X= 2.) = 0.146525

p(X= 3.) = 0.195367

p(X= 4.) = 0.195367

p(X= 5.) = 0.156293

p(X= 6.) = 0.104196

p(X= 7.) = 0.05954

p(X≤ 7.) = 0.948866

p(X≤ 7.) = poissCdf(4. , 0. , 7.) = 0.948866

Terminé

La fonction *pois1f(m,x)* (pour « Calcul de la probabilité p(X≤x) de la loi de Poisson, méthode brute force ») permet de déterminer la formule de la loi de Poisson selon le paramètre *m*, et permet de calculer l'événement *p(X≤x)* en calculant toutes les probabilités de 0 à x, puis en faisant leur somme.

La fonction utilisable (poiss*Cdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

Il est recommandé aux utilisateurs de calculer les résultats sous forme approchée.

5 – pois2(m,x)

pois2(4,4)

$$\text{Formule : p(X=x)} = \frac{4^{x}\cdot e^{-4}}{x!}$$

p(X≥ 4) = 1−p(X< 4) = 1−poissCdf(4 , 0 , 3) = 0.56653

Terminé

La fonction *pois2(m,x)* (pour « Calcul de la probabilité p(X≥x) de la loi de Poisson ») permet de déterminer la formule de la loi de Poisson selon le paramètre *m*, et permet de calculer l'événement *p(X≥x)*.

La fonction utilisable (*poissCdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

6 – pois2f(m,x)

pois2f(4,4)

$$\text{Formule : p(X=x)} = \frac{0.018316\cdot(4.)^{x}}{x!}$$

Méthode brute force

p(X= 0.) = 0.018316

p(X= 1.) = 0.073263

p(X= 2.) = 0.146525

p(X= 3.) = 0.195367

p(X< 4.) = 0.43347

p(X≥ 4.) = 1−p(X< 4.) = 0.56653

p(X≥ 4.) = 1−p(X< 4.) = 1−poissCdf(4. , 0. , 3.) = 0.56653

Terminé

La fonction *pois2f(m,x)* (pour « Calcul de la probabilité p(X≥x) de la loi de Poisson, méthode brute force ») permet de déterminer la formule de la loi de Poisson selon le paramètre *m*, et permet de calculer l'événement *p(X≥x)* en calculant toutes les probabilités de 0 à x-1, puis en faisant leur somme.

La fonction utilisable (poiss*Cdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

Il est recommandé aux utilisateurs de calculer les résultats sous forme approchée.

7 – poiss(m1,m2)

```
poiss(2,3)
                    X1→P( 2 ) et X2→P( 3 )
                    Si X1 et X2 indépendantes : X1+X2→P( 5 )
                                                      Terminé
```

La fonction *poiss(m1,m2)* (pour « Somme de deux variables aléatoires suivant une loi de Poisson ») permet de calculer la loi de Poisson P issue de la somme de deux variables aléatoires suivant une loi de Poisson *X1* et *X2*.

La formule est la suivante : $P(m) = P(m1 + m2)$

Dans l'exemple ci-contre, on a *m1* = 2 et *m2 = 3*, ce qui donne évidemment *P(2+3)=P(5)*.

8 – poisan(m)

```
poisan(20)
                    Approximation de la loi de Poisson par une loi normale
                    Paramètres : m = 20
                    m≥15
                    La condition est vérifiée
                    X→P( m ) →N([m= 20 ],[σ= √ ( 20 ]) = N(m= 20 ,σ= 2·√5 )
                                                      Terminé
```

La fonction *poisan(n,p)* (pour « Approximation de la loi de Poisson par une loi normale ») permet de déterminer si la loi de Poisson de paramètre *m* peut être approchée par une loi normale.

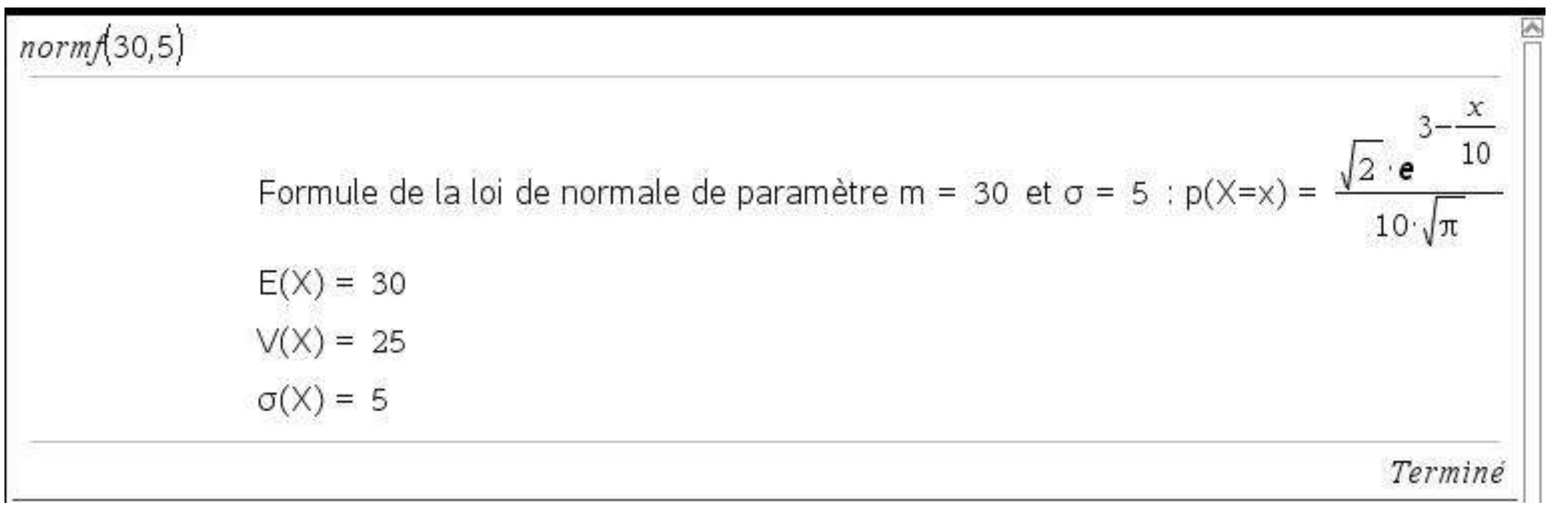
Critère de l'approximation de la loi de Poisson par une loi normale : $Si\ m \geq 15\ , alors\ N(m, \sigma) \to N\begin{cases} m = m \\ \sigma = \sqrt{m} \end{cases}$

Ici, on tente une approximation de la loi binomiale de paramètres *m = 20*. On a *m≥30.* Par conséquent, on peut approximer la loi de Poisson par la loi normale de paramètres *m = 20* et *p = √(20) = 2√5*

Loi normale

L'objectif de cette série de programmes est de maitriser et parfaire ses connaissances en applications de la loi normale. Les calculs sont beaucoup plus compliqués que ceux pour la loi binomiale ou de Poisson, mais restent accessible à la majorité des étudiants vu que ce n'est que l'application pure et simple d'algorithmes à connaitre inconsciemment. En effet, les calculs réalisables sur la loi normale sont toujours les mêmes. Il faut également savoir maitriser les changements de probabilités permettant de résoudre le problème posé par l'énoncé. Ensuite, il suffit d'appliquer les algorithmes qu'on connait (il est vrai que des étudiants savent réaliser le calcul eux-mêmes, mais sont incapables d'en écrire l'algorithme étape par étape). Par ailleurs, certaines fonctions nécessitent de lourds calculs, ce pourquoi dans la note de début, est inscrit qu'il est recommandé de faire le jailbreak de votre calculatrice Nspire CAS et d'utiliser Nover pour overclocker votre calculatrice afin d'accélérer les calculs qui peuvent durer plus de 3 minutes sur une Nspire CAS Clickpad overclockée à 150 MHz. Les fonctionnalités nécessitant de lourds calculs sont mentionnées ici. Les fonctions concernées sont *norms1*, *norms2*, *norms3*, *normin1*, et *normin2*.

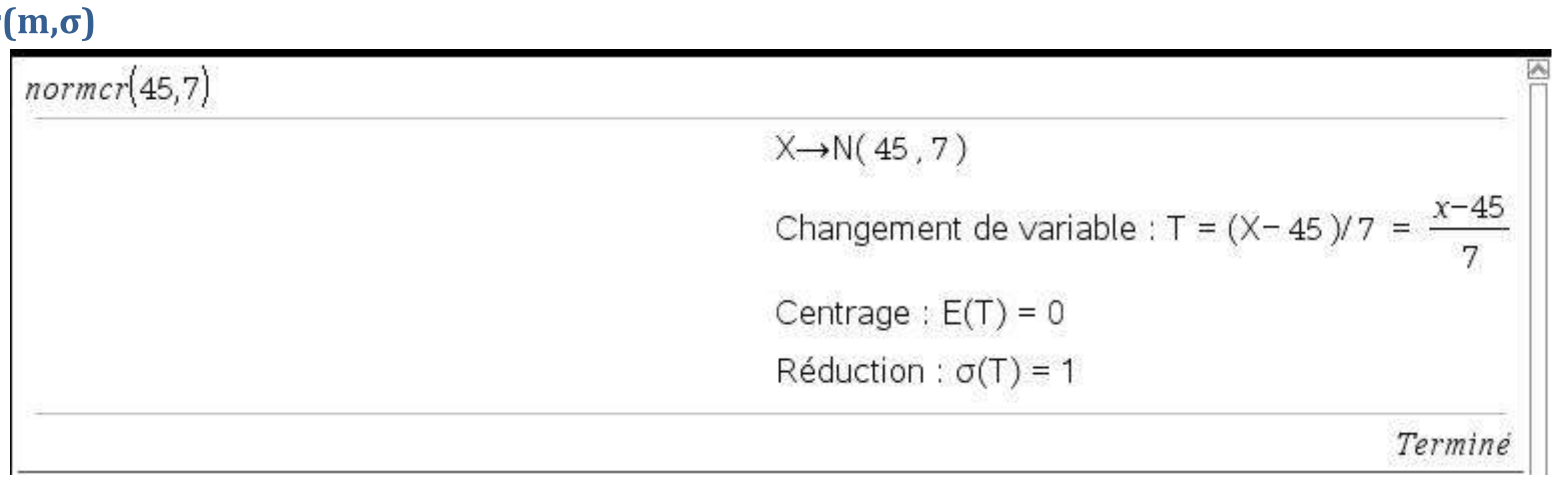
1 – normf(m,σ)



La fonction *normf(m, σ)* (pour « Formule de la loi normale ») permet de calculer la formule de la loi normale de paramètres *m* et *σ*, son espérance, sa variance, et son écart-type.
Il suffira de spécifier les paramètres *m* et σ.

Formule de la loi normale : $p(X = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{X-m}{\sigma}\right)^2}$
Formule de l'espérance : $E(X) = m$
Formule de la variance : $V(X) = \sigma^2$
Formule de l'écart-type : $\sigma(X) = \sigma$

2 – normcr(m,σ)

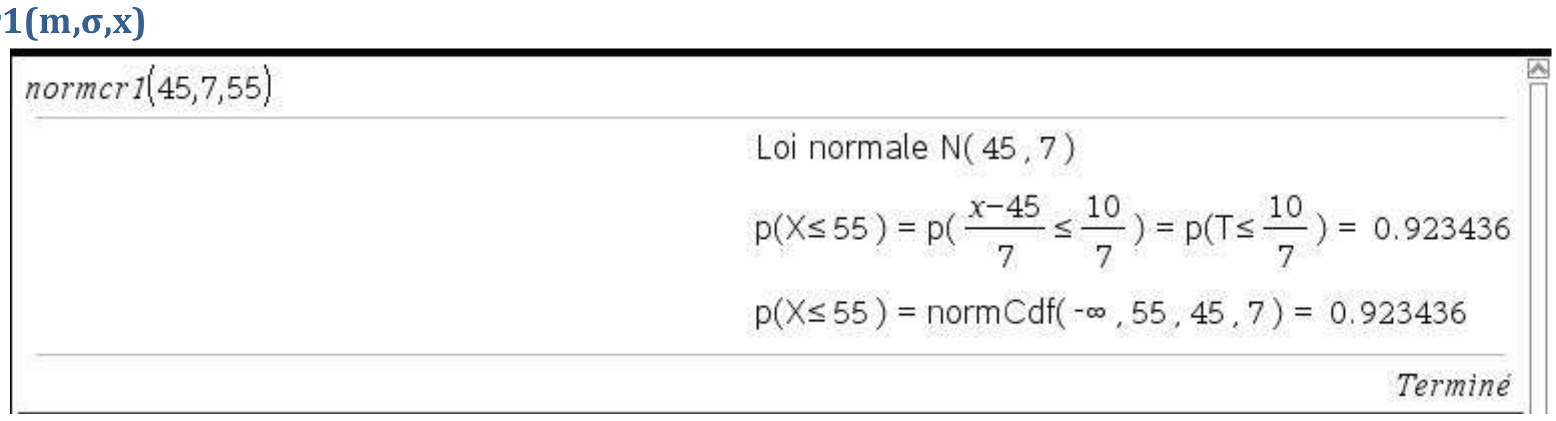


La fonction *normcr(m, σ)* (pour « Loi normale centrée réduite ») permet de calculer la formule de la loi normale centrée réduite de paramètres *m* et *σ*.

Le centrage et la réduction ont pour propriétés suivantes :
Formule de la loi normale centrée réduite : $p(T = x) = \frac{X-m}{\sigma}$
Formule de l'espérance : $E(X) = 0$
Formule de la variance : $V(X) = 1$
Formule de l'écart-type : $\sigma(X) = 1$

Le centrage et la réduction permet de résoudre plus aisément des problèmes de probabilités utilisant la loi normale.

3 – normcr1(m,σ,x)



La fonction *normcr1(m,σ,x)* (pour « Calcul de la probabilité p(X≤x) de la loi normale ») permet de calculer l'événement *p(X≤x)* connaissant la loi normale N(m, σ). La probabilité de l'événement est calculée après le centrage et la réduction de la loi normale, permettant des calculs concrets sans passer par de lourdes formules.

Le principe est le suivant :
- centrer et réduire
- adopter un changement de variable pour le premier membre de l'inéquation
- lire sur une table de valeurs de la loi normale centrée réduite le résultat

La fonction utilisable (*normCdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

normcr1(45,7,35)

Loi normale N(45 , 7)

$p(X\leq 35) = p(\frac{x-45}{7} \leq \frac{-10}{7}) = p(T\leq \frac{-10}{7}) = p(T> \frac{-10}{7}) = 1-p(T\leq \frac{-10}{7}) = 1-0.923436 = 0.076564$

$p(X\leq 35) = normCdf(-\infty, 35, 45, 7) = 0.076564$

Terminé

normcr1(45,7,35)

Loi normale N(45. , 7.)

$p(X\leq 35.) = p(0.142857\cdot(x-45.)\leq -1.42857) = p(T\leq -1.42857) = p(T> -1.42857) = 1-p(T\leq -1.42857) = 1-0.923436 = 0.076564$

$p(X\leq 35.) = normCdf(-\infty, 35., 45., 7.) = 0.076564$

Terminé

Ces deux exemples montrent deux choses différentes.

La première montre que le programme sait s'adapter à un cas où on ne peut lire la probabilité dans une table de la loi normale centrée réduite.
La seconde montre qu'il est plus intéressant d'opter pour des valeurs approchées (Ctrl+Enter au lieu de Enter) pour pouvoir immédiatement lire la probabilité dans une table de la loi normale centrée réduite. Cela n'impacte pas le résultat du calcul. En revanche, les divisions deviennent des multiplications, ce qui est peut être rapidement génant.

4 – normcr2(m,σ,x)

normcr2(45,7,30)

Loi normale N(45 , 7)

$p(X\geq 30) = p(\frac{x-45}{7} \geq \frac{-15}{7}) = p(T\geq \frac{-15}{7}) = p(T< \frac{15}{7}) = 0.983938$

$p(X\geq 30) = normCdf(30, \infty, 45, 7) = 0.983938$

Terminé

La fonction *normcr2(m,σ ,x)* (pour « Calcul de la probabilité p(X≥x) de la loi normale ») permet de calculer l'événement *p(X≥x)* avec *N(m, σ)*.. La probabilité de l'événement est calculée après le centrage et la réduction de la loi normale, permettant des calculs concrets sans passer par de lourdes formules.

Le principe est le même :
- centrer et réduire
- adopter un changement de variable pour le premier membre de l'inéquation
- lire sur une table de valeurs de la loi normale centrée réduite le résultat

La fonction utilisable (*normCdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

normcr2(45,7,60)

Loi normale N(45 , 7)

$p(X\geq 60) = p(\frac{x-45}{7} \geq \frac{15}{7}) = p(T\geq \frac{15}{7}) = p(T< \frac{-15}{7}) = 1-p(T< \frac{15}{7}) = 1-0.983938 = 0.016062$

$p(X\geq 60) = normCdf(60, \infty, 45, 7) = 0.016062$

Terminé

normcr2(45,7,60)

Loi normale N(45. , 7.)

$p(X\geq 60.) = p(0.142857\cdot(x-45.)\geq 2.14286) = p(T\geq 2.14286) = p(T< -2.14286) = 1-p(T< 2.14286) = 1-0.983938 = 0.016062$

$p(X\geq 60.) = normCdf(60., \infty, 45., 7.) = 0.016062$

Terminé

De même que pour la fonction précédente, les valeurs approchées permettent de contourner le problème « d'illisibilité » des fractions telles que $\frac{15}{7}$ par exemple, mais élimine les divisions au profit des multiplications. Il est donc conseillé de faire à la fois le calcul exact et le calcul approché, puis de les recouper à l'écrit pour répondre à l'énoncé. Il est certainement préférable de prendre la première partie de la résolution du problème en calcul exact, puis de réaliser le reste des calculs en valeur approchée (car passer de $p\left(T < \frac{15}{7}\right)$ à 0.983938 peut être étrange sans l'étape intermédiaire $p(T < -2.14286)$, sauf si vous appliquez la formule avec l'intégrale permettant de faire tout le calcul, c'est-à-dire en effet la fonction de répartition suivante : $F(x) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x} e^{-\frac{1}{2}*\frac{(t-m)^2}{\sigma^2}}$).

5 – normcr3(m,σ,x,y)

normcr3(45,7,60,65)

Loi normale N(45 , 7)

$p(60 \leq X \leq 65 = p(X \leq 65) - p(X \leq 60) = p(\frac{x-45}{7} \leq \frac{20}{7}) - p(\frac{x-45}{7} \leq \frac{15}{7}) = p(T \leq \frac{20}{7}) - p(T \leq \frac{15}{7}) =$

$0.997863 - 0.983938 = 0.013925$

p(60 ≤X≤ 65) = normCdf(60 , 65 , 45 , 7) = 0.013925

Terminé

La fonction *normcr3(m*,σ,x,y*)* (pour « Calcul de la probabilité p(x≤X≤y) de la loi normale ») permet de calculer l'événement *p(x≤X≤y)* avec *N(m, σ)*. La probabilité de l'événement est calculée après le centrage et la réduction de la loi normale, permettant des calculs concrets sans passer par de lourdes formules.

Le principe est presque le même :
- éclater la probabilité en deux probabilités différentes
- centrer et réduire chacune des deux inéquations
- adopter un changement de variable pour le premier membre de l'inéquation de chaque probabilité
- lire sur une table de valeurs de la loi normale centrée réduite les deux résultats nécessaires

La fonction utilisable (*normCdf*) pour calculer directement cette probabilité est mentionnée, avec les arguments nécessaires.

normcr3(45,7,40,50)

Loi normale N(45 , 7)

$p(40 \leq X \leq, 50 = p(X \leq 50) - p(X \leq 40) = p(\frac{x-45}{7} \leq \frac{5}{7}) - p(\frac{x-45}{7} \leq \frac{-5}{7}) = p(T \leq \frac{5}{7}) - p(T \leq \frac{-5}{7}) =$

$p(T \leq \frac{5}{7}) - (1-p(T \leq \frac{5}{7})) = 0.762475 - (1 - 0.762475) = 0.52495$

p(40 ≤X≤ 50) = normCdf(40 , 50 , 45 , 7) = 0.52495

Terminé

normcr3(45,7,30,35)

Loi normale N(45 , 7)

$p(30 \leq X < 35 = p(X \leq 35) - p(X \leq 30) = p(\frac{x-45}{7} \leq \frac{-10}{7}) - p(\frac{x-45}{7} \leq \frac{-15}{7}) = p(T \leq \frac{-10}{7}) - p(T \leq \frac{-15}{7})$

$= (1-p(T \leq \frac{10}{7})) - (1-p(T \leq \frac{15}{7})) = (1 - 0.923436) - (1 - 0.983938) = 0.060502$

p(30 ≤X≤ 35) = normCdf(30 , 35 , 45 , 7) = 0.060502

Terminé

2/99

Ces deux derniers exemples montrent que le programme est capable de s'adapter aux cas les plus compliqués que vous pouvez avoir sous la forme *p(x≤X≤y)*. En effet, les probabilités opposées sont bien comprises, ainsi que le cas où les deux membres de probabilités sont opposés, comme par exemple dans le dernier exemple, passer de $p\left(T \leq \frac{-10}{7}\right) - p(T \leq \frac{-15}{7})$ à $\left[1 - p\left(T \leq \frac{10}{7}\right)\right] - [1 - p\left(T \leq \frac{15}{7}\right)]$. Cela ne pose aucun problème au programme, mais peut porter la confusion à ceux qui ne maitrisent pas les oppositions qui sont nécessaires.

6 – norms1(m,σ,p)

norms1(45,7,0.1)

Loi normale N(45 , 7)

$p(y \leq x) = 0.1 \leftrightarrow p(\frac{y-45}{7} \leq \frac{x-45}{7}) \leftrightarrow p(T \leq t) = 0.1 \leftrightarrow p(T \leq -t) = 0.9 \leftrightarrow -t = 1.28155 \leftrightarrow t =$

$-1.28155 \leftrightarrow \frac{x-45}{7} = -1.28155 \leftrightarrow x = 36.0291$

p(y≤ 36.0291) = 0.1

Terminé

La fonction *norms1(m,σ,p)* (pour « Résolution de p(Y≤x)=p de la loi normale ») permet de déterminer la variable x en résolvant *p(Y≤x)=p* avec *N(m, σ)*. La probabilité de l'événement est calculée après le centrage et la réduction de la loi normale, permettant des calculs concrets sans passer par de lourdes formules.

Le principe est le suivant :
- centrer et réduire
- adopter un double changement de variables pour le deux membres de l'inéquation
- lire sur une table de valeurs de la loi normale centrée réduite le résultat t

Puis en inverse, le principe est le suivant :
- on utilise le résultat t
- on remonte à la première équation
- on compare cette équation au résultat
- on résout cette nouvelle équation

La dernière ligne de calcul permet de vérifier si le programme a fait le bon cheminement.

norms1(45,7,0.975)

Loi normale N(45 , 7)

p(y≤x) = 0.975 ↔ p($\frac{y-45}{7} \leq \frac{x-45}{7}$) ↔ p(T≤t) = 0.975 ↔ t = 1.95996 ↔ $\frac{x-45}{7}$ = 1.95996 ↔ x = 58.7197

p(y≤ 58.7197) = 0.975

Terminé

Pour information, voilà comment fonctionne le programme pour ce cas là (qui n'est pas la meilleure méthode mais la plus simple) :
- résolution de l'équation suivante pour x : $p = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}*\frac{(t-m)^2}{\sigma^2}} dt$
- chemin inverse du raisonnement que l'on fait normalement (donc on remonte du résultat au début)

7 – norms2(m,σ,p)

norms2(45,7,0.05)

Loi normale N(45 , 7)

p(y≥x) = 0.05 ↔ p($\frac{y-45}{7} \geq \frac{x-45}{7}$) ↔ p(T≥t) = 0.05 ↔ p(T<t) = 0.95 ↔ t = 1.64485 ↔ $\frac{x-45}{7}$ = 1.64485 ↔ x = 56.514

p(y≥ 56.514) = 0.05

Terminé

La fonction *norms2(m,σ,p)* (pour « Résolution de p(Y≥x)=p de la loi normale ») permet de déterminer la variable x en résolvant *p(Y≥x)=p* avec *N(m, σ)*. La probabilité de l'événement est calculée après le centrage et la réduction de la loi normale, permettant des calculs concrets sans passer par de lourdes formules.

Le calcul est similaire que pour la fonction *norms1(m,σ,p)*. Il y a juste une opposition à faire au milieu de calcul (passage de $p(T \geq t)$à $p(T < t)$).

Notons aussi que $p(T \geq t)$et $p(T > t)$ sont équivalents, ainsi que $p(T \leq t)$et $p(T < t)$.

8 – norms3(m,σ,p)

norms3(45,7,0.95)

Loi normale N(45 , 7)

p(z≤y≤x) = 0.95 ↔ p($\frac{z-45}{7} \leq \frac{y-45}{7} \leq \frac{x-45}{7}$) ↔ p(t1≤T≤t2) = 0.95 ↔ p(T≤t) = 0.975 ↔ t = 1.95996 donc $\frac{x-45}{7}$ = 1.95996 ↔ x = 58.7197 et aussi $\frac{z-45}{7}$ = 1.95996 ↔ z = 31.2803

p(58.7197 ≤y≤ 31.2803) = 0.95

Terminé

La fonction *norms3(m,σ,p)* (pour « Résolution de p(z≥Y≥x)=p de la loi normale ») permet de déterminer les variables x et z équidistants de y en résolvant *p(z≥Y≥x)=p* avec *N(m, σ)*. La probabilité de l'événement est calculée après le centrage et la réduction de la loi normale, permettant des calculs concrets sans passer par de lourdes formules.

Le principe est le suivant :
- centrer réduire
- changement de variables par t1, T, et t2 des trois membres de l'inéquation
- par symétrie, t1 et t2 peuvent se rassembler en une unique variable t, ce qui change la probabilité
- lecture sur la table de la loi normale la probabilité
- chemin inverse pour x
- résolution de x
- chemin inverse pour z
- résolution de z

Il est conseillé de faire le calcul à la fois en valeur exacte et en valeur approchée.

9 – normsu(m1,σ1,m2,σ2)

normsu(90000,4500,96000,6000)

X1→N(90000 , 4500) et X2→N(96000 , 6000)

Si X1 et X2 indépendantes : X1+X2→N(90000 + 96000 √ (4500 ^2+ 6000 ^2)) ↔ N(186000 , 7500)

Terminé

La fonction *normsu(m1,σ1,m2,σ2)* (pour « Somme de deux variables aléatoires suivant une loi normale ») permet de calculer la loi normale *N(m, σ)* issue de la somme de deux variables aléatoires suivant une loi normale *X1* de paramètres *m1 et σ1*,et et *X2* de paramètres *m2 et σ2*.

La formule est la suivante : $N\begin{cases} m \\ \sigma \end{cases} = N\begin{cases} m = m1 + m2 \\ \sigma = \sqrt{\sigma 1^2 + \sigma 2^2} \end{cases}$

10 – normso(m1,σ1,m2,σ2)

normso(90000,4500,96000,6000)

X1→N(90000 , 4500) et X2→N(96000 , 6000)

Si X1 et X2 indépendantes : X1+X2→N(90000 − 96000 √ (4500 ^2+ 6000 ^2)) ↔ N(-6000 , 7500)

Terminé

La fonction *normso(m1,σ1,m2,σ2)* (pour « Soustraction de deux variables aléatoires suivant une loi normale ») permet de calculer la loi normale *N(m, σ)* issue de la soustraction de deux variables aléatoires suivant une loi normale *X1* de paramètres *m1 et σ1*,et et *X2* de paramètres *m2 et σ2*.

La formule est la suivante : $N\begin{cases} m \\ \sigma \end{cases} = N\begin{cases} m = m1 - m2 \\ \sigma = \sqrt{\sigma 1^2 + \sigma 2^2} \end{cases}$

11 – normaf(m,σ,a,b)

normaf(600,30,150,-170000)

X→N(600 , 30)

N(Y) = N(aX+b) 600 X+ 30)

N(Y) = a*N(X)+b = N(150 * 600 + -170000 , 150 * 30 = N(-80000 , 4500)

Terminé

La fonction *normaf(m1,σ1,a,b)* (pour « Variable aléatoire affine suivant une loi normale ») permet de calculer la loi normale *N(m, σ)* tel que X suive un ajustement affine de type *aX+b*.

La formule est la suivante : $N\begin{cases} m \\ \sigma \end{cases} = N\begin{cases} m = a * m + b \\ \sigma = |a| * \sigma \end{cases}$

12 – normin1(m,σ,p,n)

normin1(22000,8000,0.98,15000)

Loi normale N(22000 , 8000)

p(y≤p+ 15000) = 0.98 ↔ p($\frac{y-22000}{8000} \leq \frac{p-7000}{8000}$) ↔ p(T≤t) = 0.98 ↔ t = 2.05375 ↔ $\frac{p-7000}{8000}$ = 2.05375 ↔ p = 23430.

p(y≤ 23430. + 15000 = p(y≤ 38430.) = 0.98

Terminé

La fonction *normin1(m,σ,p,n)* (pour « Résolution de p(Y≤p+n)=p de la loi normale ») permet de déterminer la variable *p* en résolvant *p(Y≤x+n)=p* avec *N(m, σ)*. La probabilité de l'événement est calculée après le centrage et la réduction de la loi normale, permettant des calculs concrets sans passer par de lourdes formules.

Via ces calculs, on s'intéresse au changement linéaire causé par la variable *n*, qui fait que résoudre l'équation semble plus difficile. Hors, cela n'est pas le cas, et introduit uniquement un calcul supplémentaire : le calcul de *p*.

normin1(22000,8000,0.02,15000)

Loi normale N(22000 , 8000)

p(y≤p+ 15000) = 0.02 ↔ p($\frac{y-22000}{8000} \leq \frac{p-7000}{8000}$) ↔ p(T≤t) = 0.02 ↔ p(T≤-t) = 0.98 ↔ -t = 2.05375 ↔ t = -2.05375 ↔ $\frac{p-7000}{8000}$ = -2.05375 ↔ p = -9429.99

p(y≤ -9429.99 + 15000 = p(y≤ 5570.01) = 0.02

Terminé

Le programme gère également les cas légèrement plus compliqués (oppositions, calculs supplémentaires).

13 – normin2(m,σ,p,n)

normin2(22000,8000,0.02,15000)

Loi normale N(22000 , 8000)

$p(y \geq p+15000) = 0.02 \leftrightarrow p\left(\frac{y-22000}{8000} \geq \frac{p-7000}{8000}\right) \leftrightarrow p(T \geq t) = 0.02 \leftrightarrow p(T < t) = 0.98 \leftrightarrow t =$

$2.05375 \leftrightarrow \frac{p-7000}{8000} = 2.05375 \leftrightarrow p = 23430.$

$p(y \geq 23430. + 15000 = p(y \geq 38430.) = 0.02$

Terminé

La fonction *normin2(m,σ,p,n)* (pour « Résolution de p(Y≥p+n)=p de la loi normale ») permet de déterminer la variable *p* en résolvant *p(Y≥x+n)=p* avec *N(m, σ)*. La probabilité de l'événement est calculée après le centrage et la réduction de la loi normale, permettant des calculs concrets sans passer par de lourdes formules.

Via ces calculs, on s'intéresse au changement linéaire causé par la variable *n*, qui fait que résoudre l'équation semble plus difficile. Hors, cela n'est pas le cas, et introduit uniquement un calcul supplémentaire : le calcul de *p*.

normin2(22000,8000,0.98,15000)

Loi normale N(22000 , 8000)

$p(y \geq p+15000) = 0.98 \leftrightarrow p\left(\frac{y-22000}{8000} \geq \frac{p-7000}{8000}\right) \leftrightarrow p(T \geq t) = 0.98 \leftrightarrow p(T < t) = 0.02 \leftrightarrow p(T < {}^{-}t) =$

$0.98 \leftrightarrow {}^{-}t = 2.05375 \leftrightarrow t = {}^{-}2.05375 \leftrightarrow \frac{p-7000}{8000} = {}^{-}2.05375 \leftrightarrow p = {}^{-}9429.99$

$p(y \geq {}^{-}9429.99 + 15000 = p(y \geq 5570.01) = 0.98$

Terminé

Le programme gère également les cas légèrement plus compliqués (oppositions, calculs supplémentaires).