

St8n – LE MOUTON ÉCOSSAIS

Auteur : Isabelle Pazé

TI-Nspire™ CAS

Mots-clés : échantillonnage, simulation, fluctuation, intervalle de confiance.

Fichier associé : mouton.tns.

1. Objectif

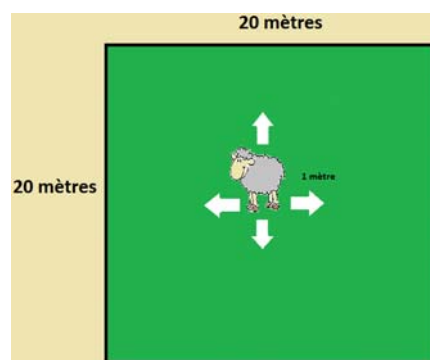
- Résoudre un problème ouvert en simulant le problème à l'aide d'un programme créé sur TI-Nspire.

2. Énoncé¹

Un paisible mouton écossais se trouve chaque matin au centre d'une lande carrée de côté 20 m, bordée d'une route.

Durant la journée, pendant 10 heures, toutes les cinq minutes, le mouton se déplace d'un mètre en choisissant au hasard une direction : soit le nord, soit le sud, soit l'est, soit l'ouest.

Clyde affirme que, la moitié des jours de l'année, il retrouve le mouton, le soir, hors de la lande. Clyde a-t-il raison?



3. Commentaires

Ce problème ouvert peut être appréhendé de plusieurs manières. Dans ce document, il sera traité en simulant les déplacements du mouton à l'aide d'un programme.

Précisons, tout d'abord, l'énoncé et la direction dans laquelle s'orientera le travail :

Ici, l'expérience consiste, dans un premier temps, à mettre le mouton au milieu de sa lande et à observer 10 heures après, s'il en est sorti ou pas.

Clyde affirme que la probabilité qu'un mouton sorte de son enclos lors d'une journée est estimée à 0,5 d'après une statistique établie sur une année, c'est-à-dire d'après une répétition de l'expérience 365 fois. Nous allons donc établir l'intervalle de confiance à partir de fréquences obtenues par simulation et voir si la probabilité estimée par Clyde peut en faire partie ou non.

On se propose donc de traiter le problème en trois parties :

- Simulation du déplacement du mouton sur une journée. (1^{re} étape)
- Simulation du déplacement du mouton sur une année et calcul de la fréquence de sortie de la lande. (2^e étape)
- Calcul de l'intervalle de confiance et conclusion. (3^e étape)

Suivant les exigences de l'enseignant, on peut attendre de l'élève qu'il ne fasse qu'une simulation sur une année (1^{re} et 2^e étapes) et en arrive à une conclusion de rejet ou d'acceptation de l'affirmation donnée dans l'énoncé. La 3^e étape est destinée à des élèves ayant bien assimilé la notion d'intervalle de confiance, ce qui peut être difficile en classe de seconde.

4. Mise en place de la simulation

1) Première étape : déplacement du mouton sur une journée : programme un_jour

Travail préliminaire :

L'élève arrive rapidement à la conclusion qu'un déplacement toutes les cinq minutes pendant dix heures correspond à 120 déplacements du mouton par jour (12 déplacements par heure soit 120 déplacements pour dix heures).

¹ D'après Odyssée, Seconde p. 275, Hatier .

Mise en œuvre :

On considère ici que la lande est munie d'un repère orthonormé d'unité le mètre, d'axes parallèles aux bords de la lande et d'origine le centre du carré.

On se sert de la fonction **randInt** : randInt(1,4) simule le choix aléatoire des 4 directions.

On définit **4 variables** dont **2 locales** :

- *i* sert à la boucle For pour répéter le déplacement aléatoire 120 fois.
- *marche* donne un nombre aléatoire entier compris entre 1 et 4 qui simule le choix aléatoire de direction.
- *x* et *y* correspondent aux coordonnées du mouton que l'on initialise à (0 ; 0) puisque le mouton est placé chaque matin au centre de la lande. Ce sont des variables globales car appelées dans le programme **une_année** de la deuxième étape.

On doit ensuite considérer les 4 cas possibles, à savoir le déplacement aléatoire du mouton à l'est, ouest, nord et sud.

On considère, par exemple, que le mouton se déplace d'un mètre :

- vers l'est si la variable locale *marche* vaut 1,
- vers l'ouest si *marche* vaut 2,
- vers le nord si *marche* vaut 3,
- vers le sud si *marche* vaut 4.

x et *y* sont donc incrémentées en fonction de la valeur de *marche*.

On peut ensuite, pour finir cette première étape, faire afficher les coordonnées du mouton après ses dix heures de marche aléatoire.

Ce qui donne, si on teste ce premier programme, un résultat comme celui de l'écran ci-contre.

```

1.1 1.2 1.3 mouton
un_jour 7/18
Define un_jour()=
Prgm
Local i,marche
x:=0
y:=0
For i,1,120
marche:=randInt(1,4)

```

```

1.1 1.2 1.3 *mouton
un_jour 12/20
If marche=1 Then
x:=x+1
EndIf
If marche=2 Then
x:=x-1
EndIf
If marche=3 Then
y:=y+1
EndIf
If marche=4 Then

```

```

1.1 1.2 1.3 *mouton
un_jour 12/20
If marche=3 Then
y:=y+1
EndIf
If marche=4 Then
y:=y-1
EndIf
EndFor
Disp x
Disp y
EndPrgm

```

```

1.1 1.2 1.3 *mouton
un_jour()
4
0
Terminé
1/99

```

2) Fin de la première étape : test

Une fois ce premier programme terminé, il suffit de lui rajouter une boucle de test pour savoir si, à la fin de la journée, le mouton est ou n'est pas sorti de sa lande : au lieu de faire afficher les coordonnées du mouton, on peut donc faire afficher à la calculatrice un message indiquant la sortie éventuelle de la lande.

```

1.1 1.2 1.3 *mouton
"un_jour" enregistrement effectué
If marche=4 Then
y:=y-1
EndIf
EndFor
If x<-10 or x>10 or y<-10 or y>10 Then
Disp "le mouton est sorti de la lande"
Else
Disp "le mouton est dans la lande"
EndIf
EndPrgm

```

On teste à nouveau le programme **un_jour**.

```

1.1 1.2 1.3 *mouton
un_jour()
le mouton est sorti de la lande
Terminé
un_jour()
le mouton est dans la lande
Terminé
2/99

```

3) Deuxième étape : Simulation sur une année

L'élève doit maintenant simuler ce déplacement sur une année.

Une des façons de faire, consiste à créer le programme **une_année** qui appelle le programme **un_jour** dans une boucle For.

Pour ne pas surcharger l'affichage demandé à la calculatrice on modifie le programme **un_jour** de la manière ci-contre, en enlevant les cinq dernières lignes du programme.

```

1.2 1.3 1.4 *mouton
un_jour 18/18
x:=x-1
EndIf
If marche=3 Then
y:=y+1
EndIf
If marche=4 Then
y:=y-1
EndIf
EndFor
EndPrgm

```

```

1.3 1.4 1.5 *mouton
une_année 9/9
Define une_année()=
Prgm
Local a,s
s:=0
For a,1,365
un_jour()
If x<-10 or x>10 or y<-10 or y>10 Then
s:=s+1
EndIf
EndFor

```

```

1.2 1.3 1.4 *mouton
une_année 1/10
For a,1,365
un_jour()
If x<-10 or x>10 or y<-10 or y>10 Then
s:=s+1
EndIf
EndFor
freq:=s/365.
Disp freq
EndPrgm

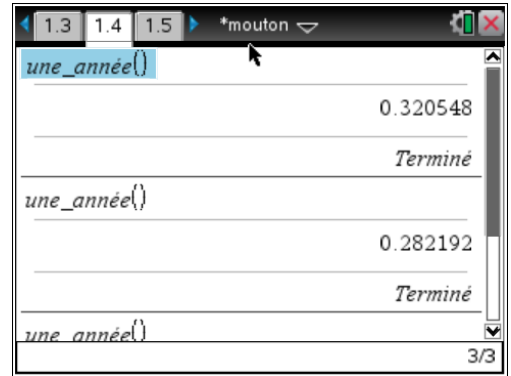
```

Commentaires : la variable *freq* est une variable globale, ce n'est pas la peine de la définir ; l'affichage en valeur décimale approchée est donné par l'instruction $freq := s/365$.

Remarque : ne pas oublier le point après le 365 qui figure au dénominateur. Il impose un affichage décimal. En utilisant l'entier 365 sans le point, on obtiendrait à l'affichage une fraction réduite, peu exploitable dans notre problème.

L'élève teste ensuite le programme **une_année** plusieurs fois pour se rendre compte que la fréquence de sortie est toujours proche de 0,30.

L'élève, en faisant plusieurs essais, peut alors conclure que, même après plusieurs simulations, on ne se rapproche pas de la valeur 0,5 correspondant à l'énoncé et qu'il est donc anormal que le mouton se retrouve la moitié de l'année hors de sa lande ; Clyde a donc tort !



5. Pour aller plus loin : recherche de l'intervalle de confiance et prise de décision

Une dernière étape peut être envisagée pour être complet.

On se propose de calculer l'intervalle de confiance $\left[f - \frac{1}{\sqrt{n}} ; f + \frac{1}{\sqrt{n}} \right]$ où f est la fréquence observée du

nombre de jours où le mouton est retrouvé hors de la lande sur une année et n la taille de l'échantillon.

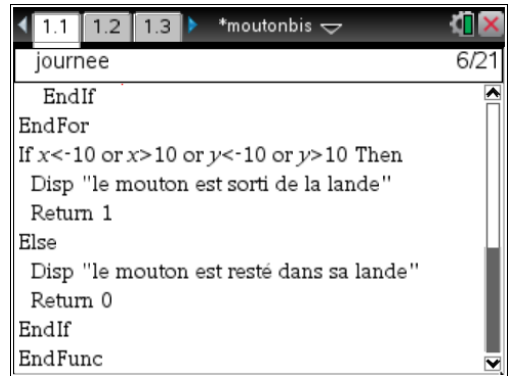
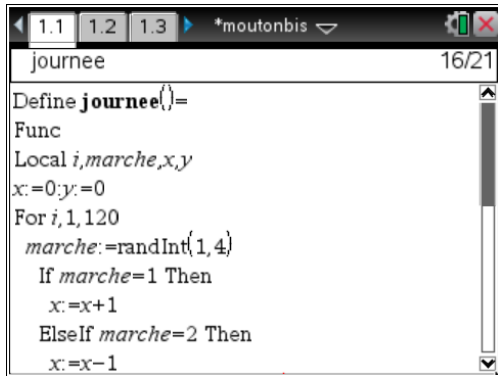
Pour pouvoir utiliser cet intervalle, il faut que $0,2 < p < 0,8$ et que $n > 25$. Ici $n = 365$.

On considère, par exemple, que 0,32 est la fréquence observée de sortie du mouton, on calcule ensuite

l'intervalle de confiance $\left[f - \frac{1}{\sqrt{365}} ; f + \frac{1}{\sqrt{365}} \right] \subset [0,267 ; 0,373]$.

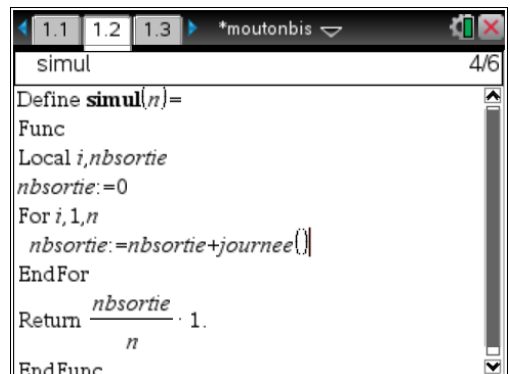
On observe que 0,5 n'appartient pas à l'intervalle de confiance $[0,267 ; 0,373]$; on en conclut donc que Clyde a tort au risque de 5 % d'erreur. (On ne peut pas affirmer qu'il a toujours tort car notre intervalle de confiance l'est au seuil des 95 % !)

Remarque 1: On pouvait utiliser des fonctions au lieu de programmes pour les différentes simulations ; **moutonbis.tns** en propose une version où **journee** renvoie 1 si le mouton est sorti et 0 sinon et où **simul** permet de simuler l'expérience sur 365 jours et renvoie la fréquence observée de sortie du mouton.



À noter : la fonction **journee()** renvoie soit "le mouton est sorti de la lande" suivi de 1, soit "le mouton est resté dans la lande" suivi de 0.

simul(365) renvoie 365 lignes avec l'une des deux phrases ci-dessus, suivies, en dernière ligne, de la fréquence observée.



Remarque 2 : Pour avoir un intervalle de confiance plus précis, on peut simuler cette expérience un grand nombre de fois, par exemple 10 000 fois afin d'avoir un intervalle de confiance plus petit, on utilise le programme **plusieurs_années** avec comme paramètre *a0* qui donne le nombre de jours souhaités.

```

1.2 1.3 1.4 mouton
plusieurs_années 10/11
Define plusieurs_années(a0)=
Prgm
Local a0,i,s
t:=0
s:=0
For i,1,a0
un_jour()
If x<-10 or x>10 or y<-10 or y>10 Then
s:=s+1
EndIf
EndFor

```

```

1.2 1.3 1.4 mouton
plusieurs_années 0/11
For i,1,a0
un_jour()
If x<-10 or x>10 or y<-10 or y>10 Then
s:=s+1
EndIf
EndFor
freq:=s/a0
Disp round(freq)
EndPrgm

```

Dans ce cas, avec comme fréquence observée 0,32, l'intervalle de confiance est alors :

$$\left[0,32 - \frac{1}{\sqrt{10\,000}} ; 0,32 + \frac{1}{\sqrt{10\,000}} \right] = [0,31 ; 0,33].$$

Attention ! Dans ce cas, la fréquence étant calculée pour 10 000 jours, le temps d'exécution du programme est assez long.

```

1.3 1.4 1.5 mouton
plusieurs_années(10000)
0.3231
Terminé
plusieurs_années(10000)
0.3167
Terminé
10/99

```