St6n – SIMULATION ET FLUCTUATION D'ÉCHANTILLONNAGE

Auteur : Marie-Laurence Brivezac **TI-Nspire™ CAS**

Mots-clés : Série statistique à une variable, échantillon, fluctuation, simulation, algorithmique, tableur.

Fichiers associés : stat_SF.tns (version 3.2)

1. Objectifs

Prendre en mains les outils de simulation et d'échantillonnage sur TI-Nspire CAS.

2. Énoncé

Une urne contient 12 boules blanches et 8 boules noires.
On se propose de simuler le tirage d'une boule de l'urne puis d'observer la fluctuation d'échantillonnage sur des échantillons de taille 100 pour répondre à la question suivante :

D'après le contenu de l'urne, la probabilité de tirer une boule blanche est $\frac{3}{5}$.

Notre simulation est-elle convenable ?

3. Commentaires

Cette activité se place dans le cadre du programme de seconde.
Le générateur aléatoire de la machine nous donne un nombre décimal x dans l'intervalle [0 ; 1[. La proportion de boules blanches dans l'urne est de 12 sur 20, soit 0,6 ; la simulation sera la suivante :

Si $x < 0{,}6$ alors la boule est blanche, sinon elle est rouge.

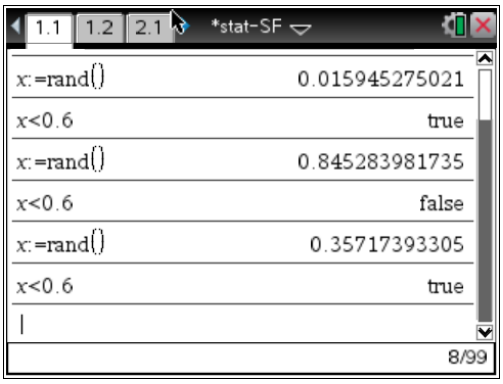
Les manipulations de base de la calculatrice ainsi que les connaissances de seconde en algorithmique sont supposées connues.

4. Conduite de l'activité

1) Simuler l'expérience

Ouvrir une page **Calculs**.
La variable x est affectée d'un nombre aléatoire **x :=rand()**
La valeur obtenue est alors comparée à 0,6 **x<0.6**

Le résultat obtenu est vrai (**true**) ou faux (**false**).



Remarque: Il est possible d'utiliser des simulations différentes, notamment en utilisant une fonction générant un entier aléatoire compris entre 1 et 20 : fonction **randInt(1,20)**.
La simulation pourrait alors être si $x \leq 12$ alors la boule est rouge.

2) Réaliser un échantillon de taille 100

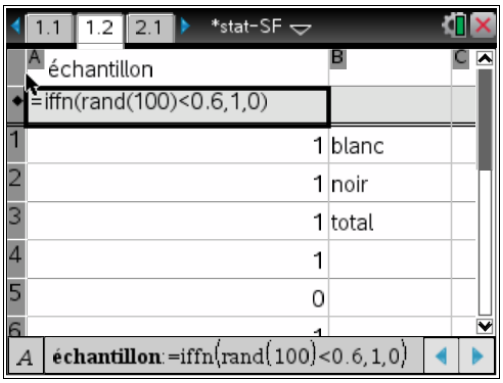
Obtenir cet échantillon revient à effectuer le tirage précédent 100 fois de suite en replaçant la boule obtenue dans l'urne après chaque tirage.

En utilisant les listes, nous allons générer l'échantillon en colonne A.

- Ouvrir une page **Tableurs & Listes**.
- Nommer la colonne A « **échantillon** ».
- Utiliser la fonction rand dans une fonction de test **Iffn** :

=iffn(rand(100)<0.6,1,0)

rand(100) donne 100 décimaux aléatoires,
iffn(condition vraie ou fausse,1,0) donne 1 pour vraie et 0 pour fausse.



- Compter les blancs obtenus revient ensuite à compter les 1 avec une fonction somme (voir **3**) : =**sum(**échantillon**)**.

3) Distribution des fréquences

Calcul des fréquences d'apparition des boules blanches et noires pour l'échantillon réalisé.

On placera les fréquences en colonne C.

- Saisir en colonne B le **texte** entre guillemets,

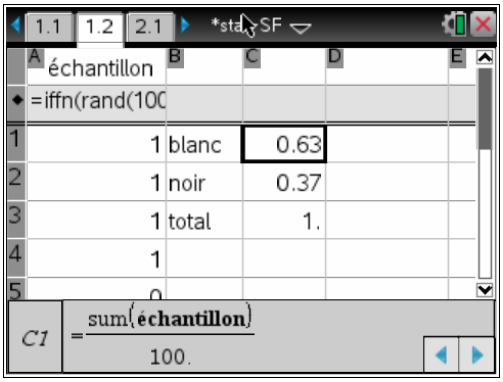
exemple en B1 : "blanc"

- Saisir en colonne C les **formules**.

Pour la fréquence des blancs en C1 :

=**sum**(échantillon)/100.

Remarque : le point après le 100 est volontaire. Il impose un affichage décimal.
En utilisant l'entier 100 sans le point, on obtiendrait à l'affichage une fraction réduite.



4) Fluctuation d'échantillonnage

D'après le contenu de l'urne, la probabilité de tirer une boule blanche est $\frac{3}{5}$.

Notre simulation est-elle convenable ?

On analysera successivement t échantillons de taille $n = 100$, $t \in \{20 ; 50 ; 100 ; 1\,000\}$.

Pour notre problème, l'intervalle de fluctuation au seuil de 95 % est $\left[p - \frac{1}{\sqrt{n}} ; p + \frac{1}{\sqrt{n}} \right] = [0{,}5 ; 0{,}6]$.

Dans cette partie, le travail sur Nspire sera effectué au moyen d'un **programme**[1].

Programme de simulation : freq_blanc(n)

Il calcule la fréquence des tirages blancs d'un échantillon de taille n.

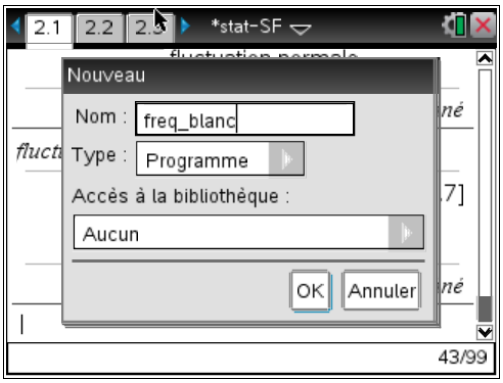
- Ouvrir une nouvelle activité [doc] [4] [1] et choisir une page **Calculs**.
- Passer en éditeur de programmes:

[menu] [9]**: Fonctions & programmes**,

[1]**: Éditeur de programmes**, [1]**: Nouveau**.

- Saisir le nom (freq_blanc) et valider les autres champs.

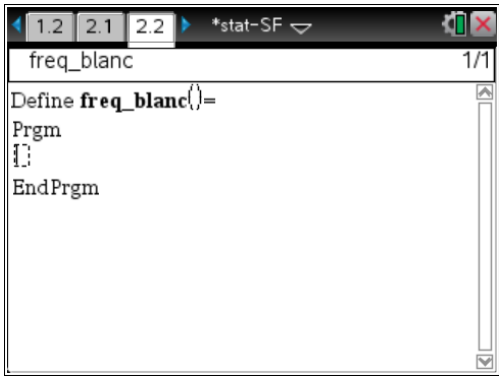
La page calcul est alors partagée en deux dans le sens vertical.



[1] L'utilisation du tableur est une autre solution.

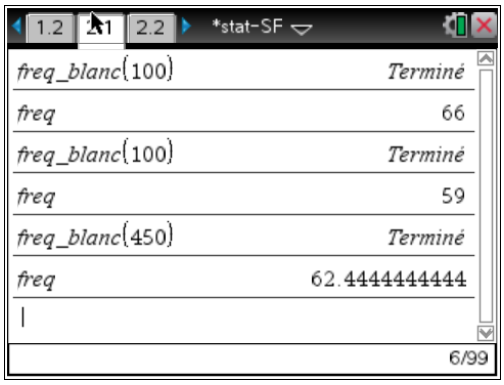
Ce format sur calculatrice n'est pas confortable, nous allons donc dégrouper ces pages avant de continuer.

• Dégrouper les pages avec doc ▾ 5 : **Format de page**, 8 : **Dégrouper**.
On obtient deux pages :
2.1 page calcul,
2.2 page éditeur de programme.



• Saisir le programme.

```
Define freq_blanc(n)=
Prgm
© le paramètre n est la taille de l'échantillon
© freq est une variable globale
Local i,nb
nb:=0
For i,1,n
  If rand()<0.6 Then
    nb:=nb+1
  EndIf
EndFor
freq:=nb/n·1.
EndPrgm
```



• Vérifier la syntaxe et enregistrer : ctrl B.

• Passer sur la page calcul (2.1) pour tester le programme.
La fluctuation d'échantillonnage s'observe déjà.

Programme de fluctuation : fluctuat(*t*,*n*)

On écrit ce programme selon la même procédure.

Il donne le pourcentage d'échantillons dans l'intervalle de confiance et répond à la question de normalité.
Il prépare la représentation des résultats par un nuage de points dans une page **Données & statistiques**.

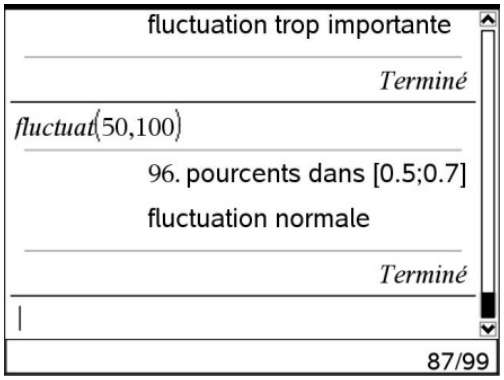
• Deux arguments en entrée : le nombre *t* de simulation et la taille *n* d'une simulation.

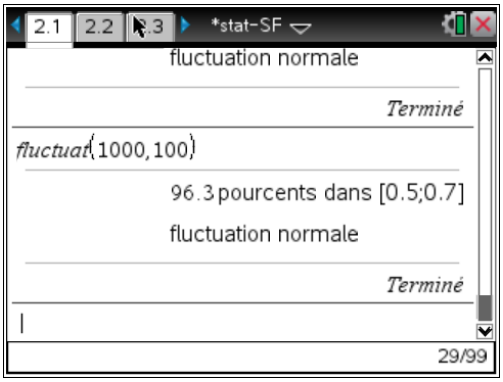
• Ce programme appelle « freq_blanc » qui est donc un sous-programme[2].

• Deux listes en sortie: "echantillon" et "frequence" pour les graphiques.

```
fluctuat
Define fluctuat(t,n)=
Prgm
©paramètre t: le nombre d'échantillons
©paramètre n: taille de l'échantillon
©le sous programme freq_blanc utilise une
©variable globale freq pour donner son résultat
Local i,compteur
echantillon:=newList(t)
frequence:=newList(t)
compteur:=0
For i,1,t
  echantillon[i]:=i
  freq_blanc(n)
  frequence[i]:=round(freq,2)
  If frequence[i]≥0.5 and frequence[i]<0.7 Then
    compteur:=compteur+1
  EndIf
EndFor
freq:=round(compteur/t·100,2)
Disp freq,"pourcents dans [0.5;0.7]"
If freq<95 Then
  Disp "fluctuation trop importante"
Else
  Disp "fluctuation normale"
EndIf
EndPrgm
```

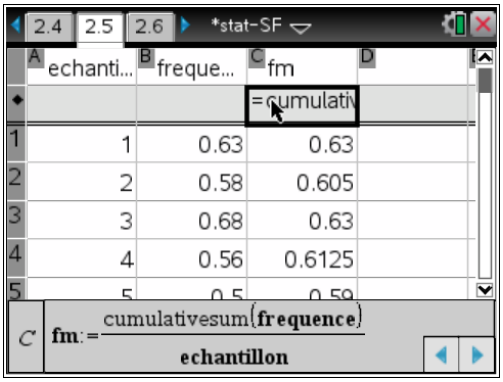
[2] On aurait pu choisir le type « fonction », mais la programmation de fonctions n'est pas au programme de seconde.





Représentations graphiques

Les deux listes de sortie sont maintenant utilisables pour des représentations graphiques. Les listes générées par le programme sont des variables globales donc accessibles dans chaque page de l'activité par leur nom après exécution du programme.



• Ouvrir une page **Tableur & Listes** pour visualiser les listes obtenues.

• Placer en colonne A « echantillon » et en colonne B « frequence ».

• Calculer en colonne C, nommée « fm », la fréquence moyenne :

=cumulativesum(frequence)/echantilllon

La fonction **cumulativesum**(liste) calcule la fréquence Cumulée croissante de la liste indiquée.

Observer la fluctuation des échantillons :

• Ouvrir une page **Données & statistiques**.

• Afficher le nuage de points (echantillon ; frequence).

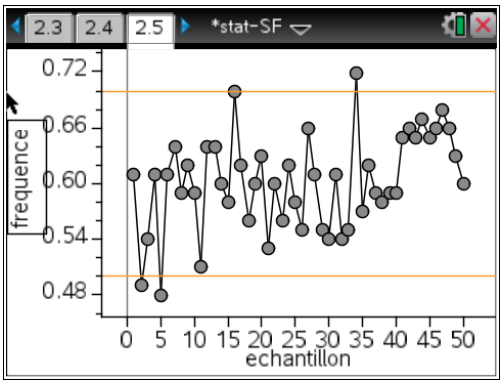
menu 2 5 : **Ajouter la variable X**.

menu 2 8 : **Ajouter la variable Y**.

• Ajouter les droites d'équation $y = 0{,}5$ et $y = 0{,}7$ pour visualiser les échantillons hors intervalle.

menu 4 : **Analyser**, 4 : **Tracer la fonction**.

• Saisir la fonction.



Visualiser la convergence vers la fréquence théorique 0,6 :

• Ouvrir une autre page **Données & statistiques**.

• Afficher le nuage de points (echantillon ; fm).

• Ajouter la droite d'équation $y = 0{,}6$ pour visualiser la convergence vers la fréquence théorique 0,6.