St1n – COMBIEN DE FOIS … AVANT DE …

Auteur : Christian Hakenholz **TI-Nspire™ - TI-Nspire™ CAS**

***Avertissement** : ce document a été réalisé avec la version 1.4 ; il est disponible dans sa version la plus récente sur notre site http://education.ti.com/france, menu Ressources pédagogiques.*

Mots-clés : simulation, fluctuation des échantillons.

Fichiers associés : St1n_CombienDeFois_1.tns (*pour le premier problème*),
St1n_CombienDeFois_2.tns (*pour le second problème*).

1. Objectifs

Simuler le lancer d'un dé *n* fois, jusqu'à l'obtention d'un résultat.
Familiariser l'élève avec la notion de fluctuation d'échantillonnage.

2. Les problèmes

Combien de fois faut-il lancer un dé, avant d'obtenir un « 6 » ?
Combien de fois faut-il lancer un dé, avant d'obtenir toutes ses faces ?

3. Problème du « 6 »

On définit une fonction qui compte le nombre de lancers de dés nécessaire pour obtenir un « 6 » ; pour cela, on écrit un petit programme.

Ouvrir un nouveau classeur, puis une page **Calculs** pour définir la fonction :

```
Define six()=Func                 (1)
        Local n                   (2)
        n:=1                      (3)
        While randInt(1,6)≠6      (4)
        n:=n+1                    (5)
        EndWhile                  (6)
        Return n                  (7)
        EndFunc                   (8)
```

Quelques explications :

(1) La fonction **six** n'utilise pas de paramètre, mais ceci doit être signalé par des parenthèses vides, après son nom.

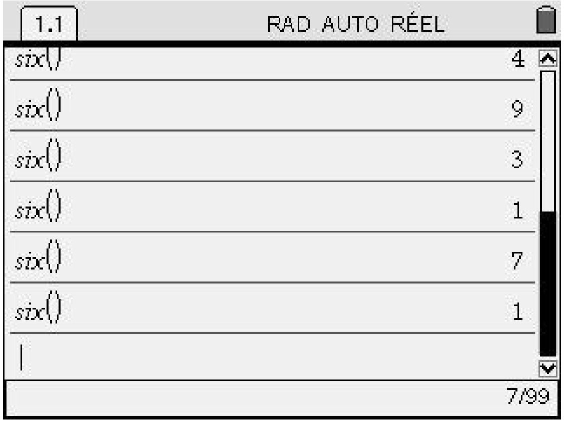
(2) La variable **n** doit être déclaré locale, car une fonction refuserait une affectation comme la ligne (3), sur une variable globale.

(4) La fonction **randInt(1,6)** retourne un nombre entier entre 1 et 6 compris ; elle simule le comportement d'un dé parfait.

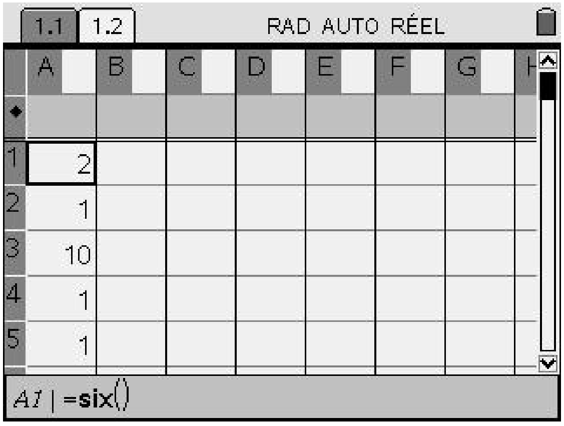
(5) Tant que le dé ne retourne pas un six, on ajoute 1 au nombre *n* de lancers.

(7) On affiche le contenu de *n* lorsque le premier six est sorti.

La fonction s'utilise comme le montre l'écran ci-joint.



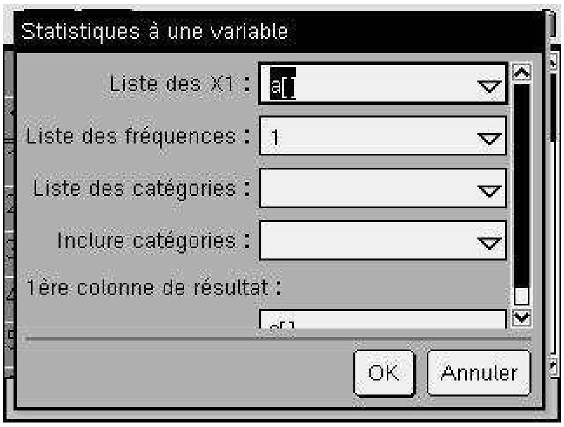
Ouvrir une page **Tableur & listes**, et remplir les 100 premières cellules de la colonne A en tapant dans la cellule A1 **= six()** que l'on copie jusqu'en cellule A100.



Dans la colonne B, on peut placer des résultats statistiques :

Se placer dans l'en-tête de la colonne B ; demander :

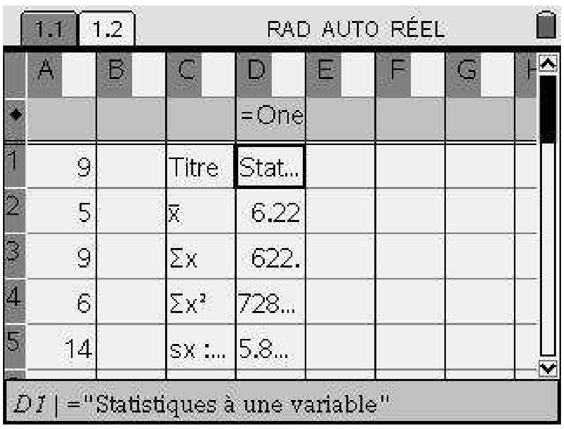
menu **4 : Statistiques**
1 : Calculs statistiques
1 : Statistiques à une variable
Nombre de listes : 1.



Les résultats apparaissent. On pourra noter que la médiane est sensiblement différente de la moyenne (en général, mais la fluctuation des échantillons nous réserve parfois des surprises ….).

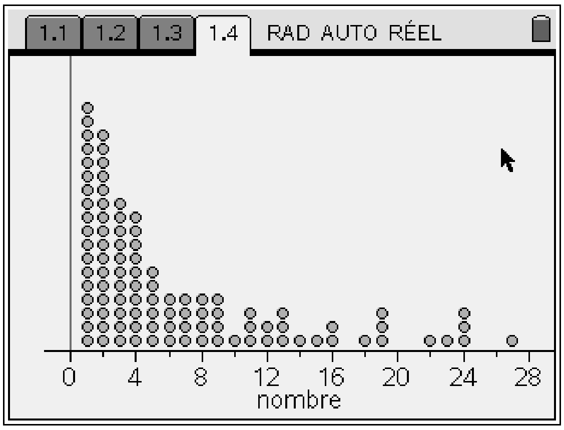
On pourra noter que Σx correspond au nombre de lancers nécessaires pour obtenir 100 fois le « 6 ».

On peut alors lancer le « recalcul » (ctrl R) et faire des observations.



Pour obtenir une illustration de ces résultats, nommer la colonne A, **`nombre`**, par exemple.

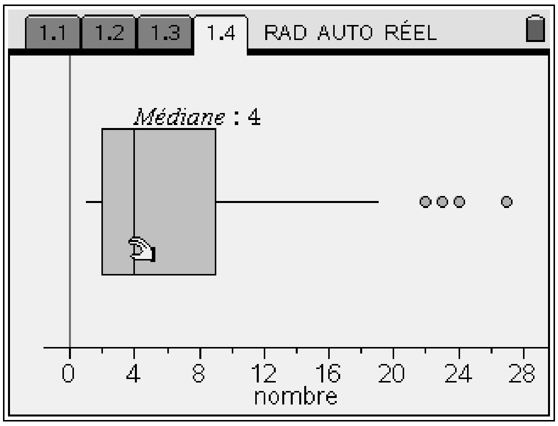
Puis, ouvrir une page **Données & statistiques** et donner en abscisse **nombre**.



On peut également changer le type de tracé, pour une boîte à moustaches, par exemple.

Pour modifier le graphique, appuyer sur ctrl menu et choisir 1 pour la boite à moustaches.

En posant le curseur sur les éléments du graphique, on obtient l'affichage des paramètres de la série, ici la médiane.



Remarque : il est possible de construire rapidement une nouvelle fonction qui affiche la moyenne des valeurs prises par n sur k jeux en procédant comme suit :

```
Approx(mean(seq(six(),a,1,k)))→f(k)
```

Il suffit alors de demander f(10), f(20) *et* f(100) *pour voir à l'écran s'afficher les valeurs qui permettent d'approcher l'espérance mathématique théorique qui est* 6.

Pour aller plus loin : une justification de cette espérance mathématique

Soit X la variable égale au nombre de lancers nécessaires à l'obtention du premier 6 lors du lancer d'un dé parfait.

Il est clair que $P(X = 1) = \frac{1}{6}$ et que $P(X = 2) = \frac{5}{6} * \frac{1}{6}$, en effet si le six sort pour la première fois au second lancer, au premier lancer on a obtenu une autre face que la 6 avec la probabilité $\frac{5}{6}$.

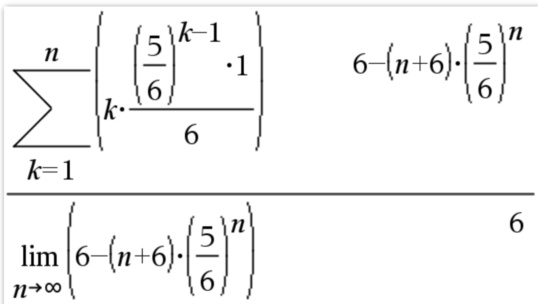
De même, $P(X = k) = \left(\frac{5}{6}\right)^{k-1} \times \frac{1}{6}$.

On dit dans ce cas que la variable aléatoire X suit la loi géométrique de paramètre $\frac{1}{6}$.

L'espérance mathématique de X est donc :

$$E(X) = \sum_{k=1}^{\infty} k.P(X = k) .$$

Le calcul est donné dans l'écran ci-contre.



4. Problème de faces

Voici l'algorithme qui permet de comprendre le programme réalisé ci-dessous :

(1) Créer les variables appelées *n*, *liste* et *d* qui vont respectivement contenir le nombre *n* de lancers effectués, la liste des différentes faces déjà sorties et la nouvelle face qui vient de sortir.

(2) Initialiser la variable *n* à 0, la variable *liste* à {0,0,0,0,0,0}

(3) **Tant que** le produit des termes de la liste est nul
 n augmente de 1
 on lance le dé et on stocke le résultat dans la variable *d*
 on place le résultat du lancer au rang *d* de la liste.
 Fin du tant que

(4) Afficher le contenu de la variable *n*

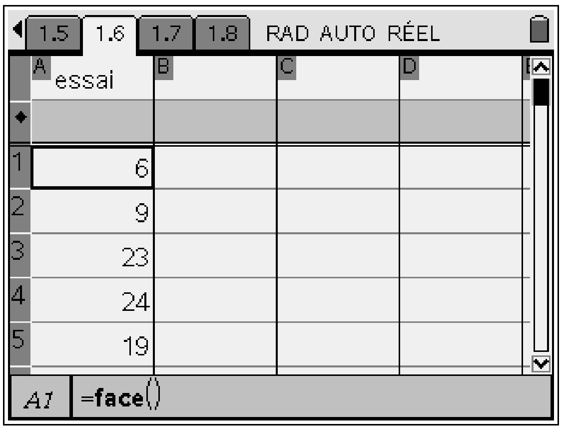
Ouvrir un nouveau classeur, puis une page calcul pour définir la fonction :

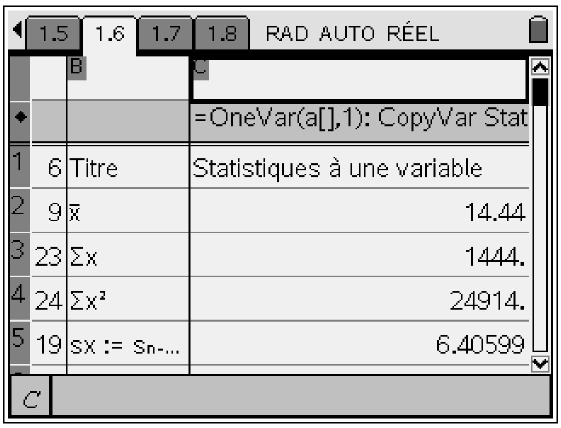
```
Define face()=Func
            Local n,liste,d
            n:=0
            liste:={0,0,0,0,0,0}
            While product(liste)=0
               n:=n+1
               d:=randint(1,6)
               liste[d]:=d
            EndWhile
            Return n
            EndFunc
```

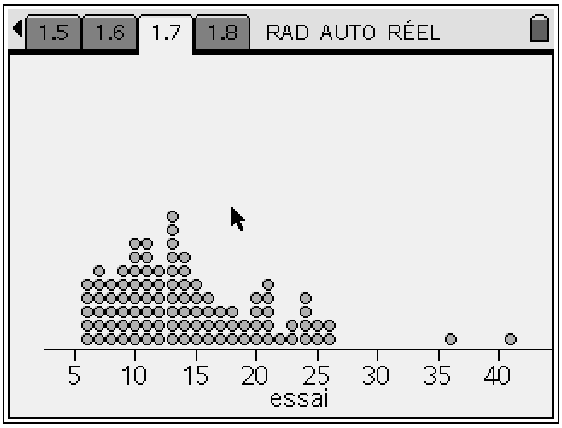
Quelques explications sur les variables :

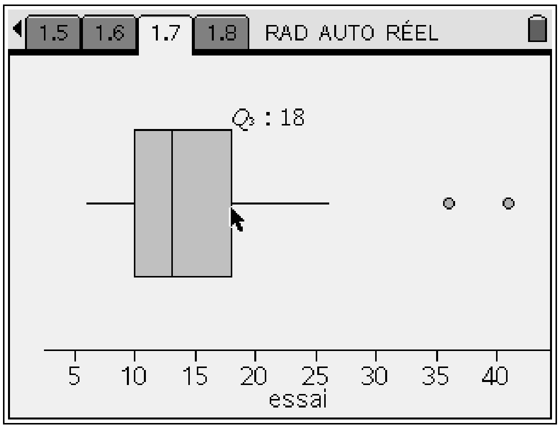
- **n** compte le nombre de fois où le dé est lancé ;
- **liste** est la liste des six faces du dé, elle est initialisée avec 6 zéros au début du jeu car aucune des faces n'est encore sortie.
- **product(liste)**, produit des éléments de la liste, ne sera plus nul dès que la liste sera devenue {1, 2, 3, 4, 5, 6}.

On suit alors le même plan que le problème 1, on obtiendra successivement les écrans suivants :





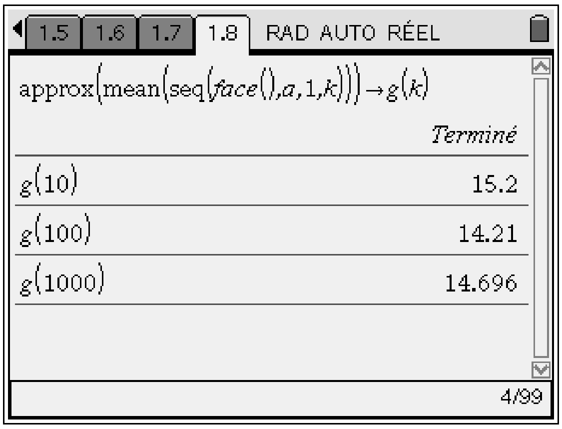




L'observation des diagrammes précédents permet de vérifier la grande variabilité des résultats.

Remarque : il est possible de construire rapidement une nouvelle fonction qui affiche la moyenne des valeurs prises par n sur k jeux en procédant comme suit :

`Approx(mean(seq(face(),a,1,k)))→ g(k)`



Pour aller plus loin : étude théorique de la variable aléatoire.

On appelle Y la variable aléatoire égale au nombre de lancers nécessaires à l'obtention de toutes les faces lorsque on lance un dé cubique parfait.

Soit X_k la variable aléatoire égale au nombre de lancers nécessaires à l'obtention de la $k^{\text{ème}}$ nouvelle face.

X_1 suit la loi géométrique de paramètre 1 ; en effet, au premier lancer, une nouvelle face est sortie.

X_2 suit la loi géométrique de paramètre $\frac{5}{6}$; X_3 une loi géométrique de paramètre $\frac{4}{6}$, …

Plus généralement, X_k suit la loi géométrique de paramètre $\frac{6-k+1}{6}$ qui a pour espérance mathématique $\frac{6}{6-k+1}$.

De plus le nombre total de lancers nécessaires à l'obtention de la série complète est $Y = X_1 + X_2 + \ldots + X_6$.

On a, propriété de l'espérance mathématique, $E(Y) = 1 + \frac{6}{5} + \frac{6}{4} + \frac{6}{3} + \frac{6}{2} + \frac{6}{1} = \frac{147}{10} = 14{,}7$.

Remarque : la simulation précédente pour 1000 *essais est donc très proche de l'espérance mathématique calculée ci-dessus.*