P1n – Les problèmes de Mérée

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

Mot-clé : Probabilité.

1. Objectifs

Introduire les probabilités en Première en utilisant deux problèmes soumis à Pascal par le chevalier de Mérée.
Montrer le lien qui existe entre les statistiques et les probabilités, notamment dans la justification d'un modèle probabiliste.

2. Les problèmes

Le premier problème, traité dans le paragraphe 3, peut se résumer ainsi : dans le cas d'une partie comportant des jeux de pur hasard, quelles sont les chances pour un joueur de gagner la partie dès qu'il a gagné trois jeux, sachant qu'il mène deux jeux à un ?

Le deuxième problème, traité dans le paragraphe 4, peut se résumer ainsi: en pariant de sortir un « as » en quatre lancers de dé au plus, je gagne plus souvent que je ne perds, mais combien dois-je proposer de lancers de deux dés pour avoir un double « as » ?
Le chevalier de Mérée avait proposé 24 lancers, car « 24 est à 36 ce que 4 est à 6 », mais il s'aperçut qu'il perdait plus souvent qu'il ne gagnait d'où son courroux lorsqu'il se plaint à Pascal de l'infidélité des dés par rapport à son raisonnement mathématique.

3. Problème du 2 – 1

Le problème est soumis aux élèves avant tout cours sur les probabilités ; ceux-ci, libres de toute théorie, s'expriment en donnant 50 %, 2 chances sur 3, 75 % voire 100 % de chances pour les plus optimistes. Ces résultats sont tout aussi acceptables que ceux proposés par des contemporains de Pascal !

Se lance alors une discussion dans laquelle chaque élève essaye de persuader les autres (professeur compris) parfois par conviction, quelques fois par intimidation.

Le professeur – arbitre propose alors de faire 100 parties pour voir le bien-fondé des propositions, avec la TI-Nspire.

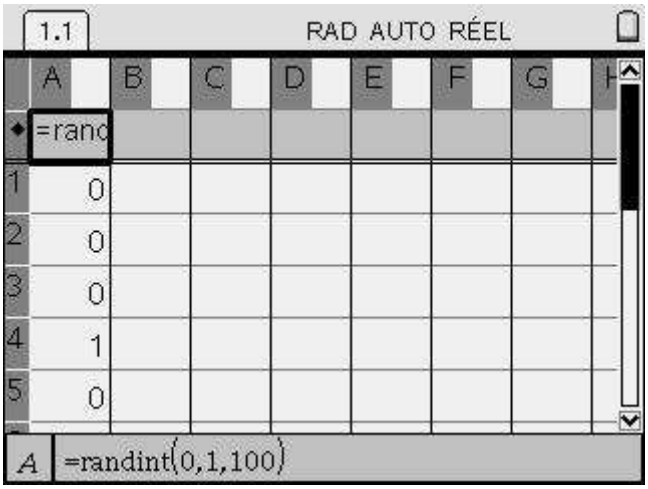
Le résultat du premier jeu est obtenu par un tirage aléatoire de 0 ou 1, la sortie du 1 signifiant que le joueur Un qui mène 2 – 1 gagne le jeu, et donc la partie.

Dans la zone de définition de la colonne A taper :

=randint(0,1,100)

qui remplit la colonne A de 100 valeurs 0 ou 1 choisies aléatoirement.

N.B. On peut signaler à cette occasion qu'il s'agit, en fait, de tirages pseudo-aléatoires.

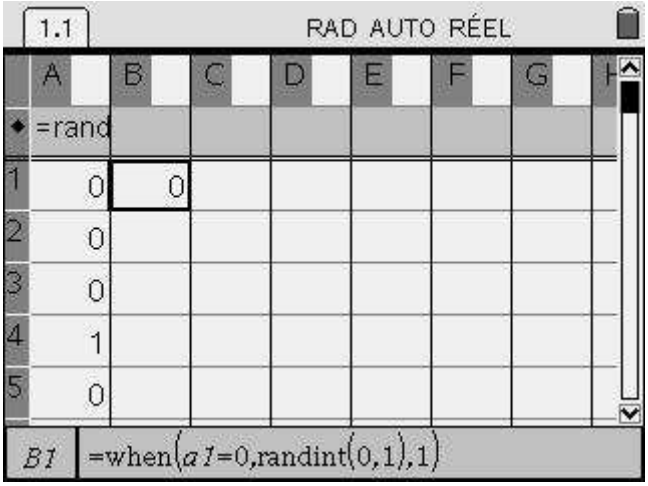


Lorsque le joueur Un a gagné le premier jeu, il gagne la partie, sinon il faut jouer un deuxième jeu. Pour le simuler, on utilise la fonction **Si** *j'ai perdu* **alors** *je rejoue* **sinon** *j'ai gagné*. Ce qui ce traduit en tapant dans la case B1 :

=when(a1=0,randint(0,1),1)

La fonction when(*condition*,*résultat1*,*résultat2*) retourne *résultat1*, si la *condition* est vraie, et *résultat2* sinon.

Dans notre cas, si a1 = 0, alors dans la case B1 s'écrit un nombre aléatoire choisi entre 0 et 1. Si a1 = 1, il s'écrit un 1.

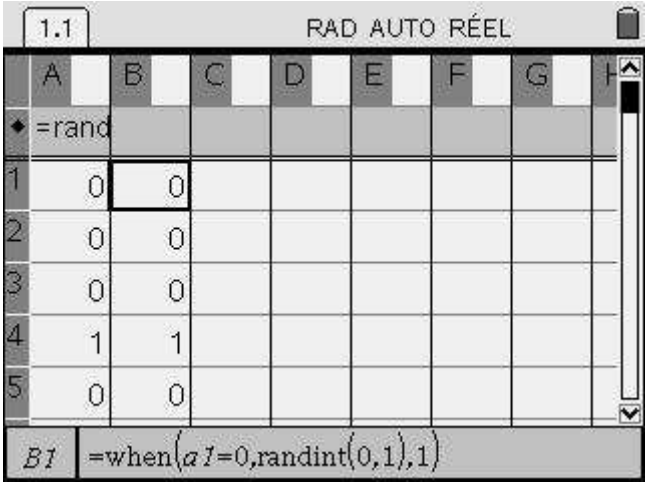


Reste à « recopier » vers le bas 99 fois cette formule par :

ctrl menu **6 : Saisie rapide**

puis descendre avec la flèche basse jusqu'à la cellule B100.

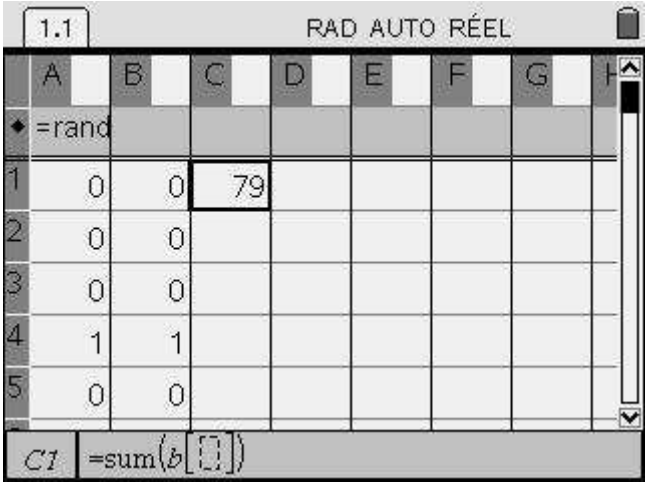
Le nombre de 1 présents dans la colonne B correspond alors au nombre de fois où le joueur Un a gagné son troisième jeu avant l'autre joueur, et donc a gagné la partie.



Pour compter le nombre de 1, il suffit d'utiliser la fonction somme, par exemple dans la cellule C1, taper :

=sum(b[])

qui effectue la somme des 100 éléments de la colonne B, c'est-à-dire le nombre de 1 présents dans la colonne, c'est-à-dire le nombre de fois où B a gagné la partie.



La discussion peut reprendre avec les élèves, d'autant plus que l'on dispose d'un résultat statistique sur 100 × nombre d'élèves, ce qui permet de se rapprocher des 75 % théoriques.

Théorie qu'il est facile de mettre en place, les élèves peuvent s'apercevoir que le joueur Un perd si l'autre joueur gagne deux jeux consécutifs, c'est à dire s'il y a deux 0 sur la même ligne des colonnes A et B. Ce qui correspond à la probabilité d'obtenir deux fois piles dans un jeu de pile ou face…

Résultat que les élèves ne sont pas peu fiers de découvrir lorsqu'on leur apprend qu'il a fallu deux ans à Pascal pour le résoudre.

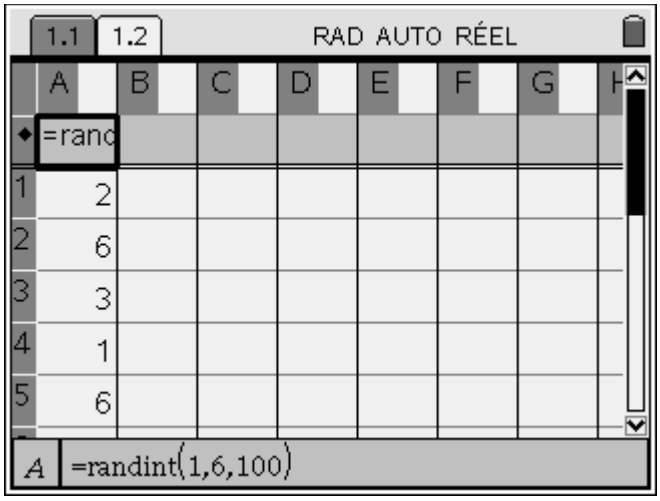
4. Problème de dés

On prend la même méthode que pour le problème précédent appliqué à un choix aléatoire d'un entier entre 1 et 6 pour simuler le résultat du lancer d'un dé.

Dans la zone de définition de la colonne A, taper :

=randint(1,6,100)

qui remplit la colonne A de 100 valeurs entières choisies aléatoirement entre 1 et 6.



Lorsque l'as est sorti, le joueur a gagné son pari, sinon il a droit à un deuxième lancer. Pour le simuler, on utilise encore la fonction *when* dans la case B1

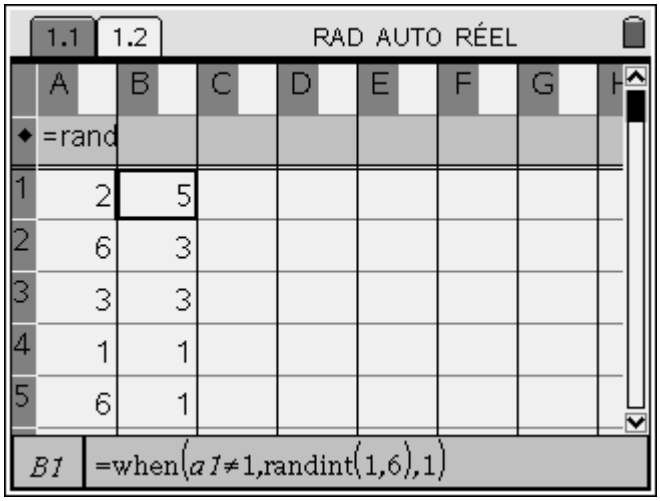
=when(a1≠1,randint(1,6),1)

Dans ce cas, si a1 ≠ 1, alors dans la case B1 s'écrit un nombre entier aléatoire choisi entre 1 et 6. Si a1 = 1, il s'écrit un 1.

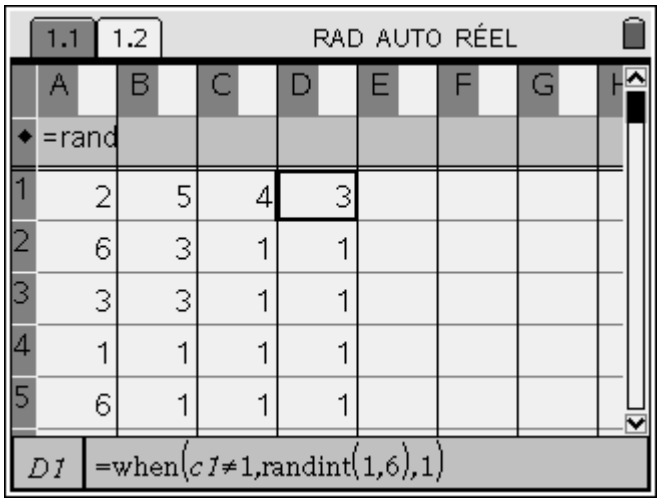
Reste à « recopier » vers le bas 99 fois cette formule par :

ctrl menu **6 : Saisie rapide**

puis descendre avec la flèche basse jusqu'à la cellule B100.



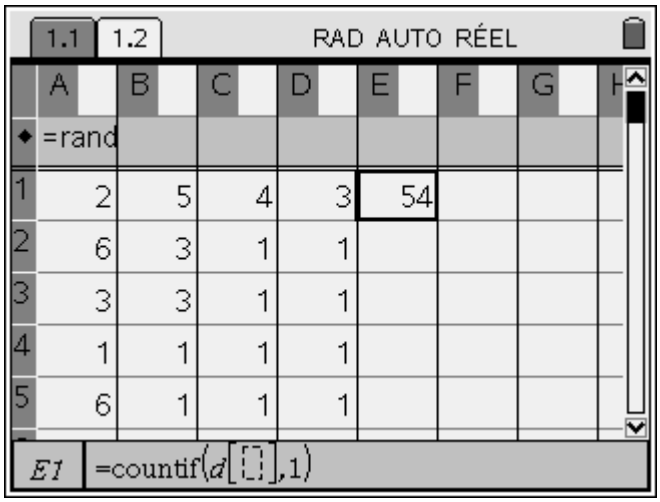
S'il n'y a pas d'as dans la colonne B, il faut recommencer dans la colonne C, puis dans la colonne D



Le nombre de 1 dans la colonne D représente le nombre de fois où le joueur a gagné son pari. Reste à compter le nombre de 1 en tapant dans une cellule libre :

=**countif(d[],1)**

où la fonction *countif*, suivant la syntaxe countif(*liste*,*valeur*), retourne le nombre de fois où la *valeur* est présente dans la *liste*.



Là, le débat peut reprendre avec les élèves dans l'analyse du jeu de lancer d'un seul dé. Sur l'écran, le joueur a gagné 54 parties sur 100, quelques élèves auront un résultat contraire, mais cela permet de revenir sur la notion de *fluctuation d'échantillonnage*.

Et le débat peut se poursuivre avec le lancer de deux dés, mais il semble fastidieux de remplir 24 colonnes. Il vaut mieux programmer une fonction qui retourne true (vrai) si le double « as » est sorti en au plus *n* lancers de deux dés et retourne false (faux) sinon.

Ouvrir une page de **Calculs**.

menu **9 : Fonctions & programme 1 : Editeur de programme 1 : Nouveau**

S'ouvre alors une fenêtre, dans laquelle est demandé le nom (taper **meree** par exemple), le type (choisir fonction) et l'accès à la bibliothèque (laisser le choix par défaut : Aucun).

L'écran de calcul est partagé en deux, la partie de droite étant l'écran de l'éditeur de programme, dans lequel se trouve

```
Define meree()=
Func

EndFunc
```

Que l'on complète par

```
Define meree(n)=
Func
Local i
i:=1
```

```
While i≤n
      If (randint(1,6)=1) and (randint(1,6)=1)
            Return true
      i:=i+1
EndWhile
Return false
EndFunc
```

La fonction **meree** retourne *true* si le double « as » est obtenu en *n* fois au plus (dans la boucle while, i ≤ *n*), et *false* sinon (à la sortie de boucle i = *n* + 1).

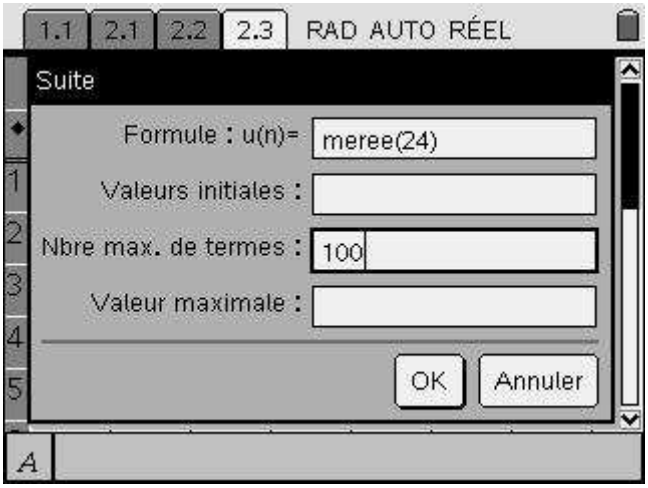
On enregistre la fonction programmée comme une variable dans le fichier en cours :

menu **2 : Vérifier la syntaxe et enregistrer** **1 : Vérifier la syntaxe et enregistrer**

Ouvrir une page **Tableur & listes**. Se placer dans la colonne A et taper :

menu **3 : Données** **1 : Générer une suite**

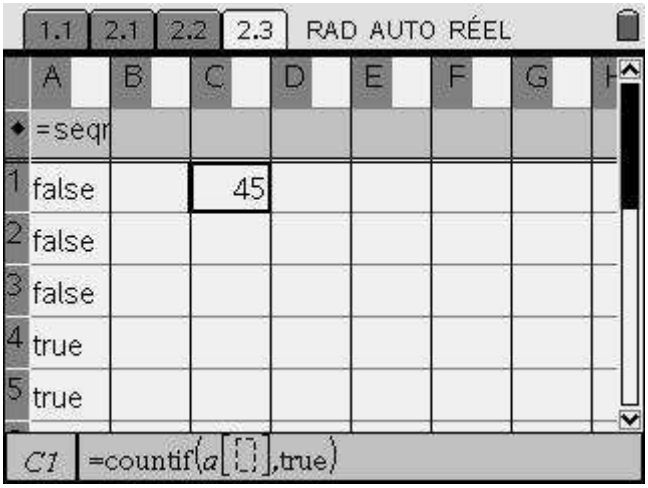
Une fenêtre de définition de suite s'ouvre alors. Compléter la formule u(n) par **meree(24)** et le nombre maximum de termes par **100**.



Reste à compter le nombre de *true* de la colonne A par

=countif(a[],true)

placé dans une cellule libre.



Avec cet écran, le pari est perdu, ce qui est peut-être normal, suivant l'expérience de Mérée mais qui est loin d'être évident sur un échantillon de l'ordre de 100. On peut modifier le nombre d'expériences en tapant 200 ou plus dans la zone de définition de la colonne A à la place de 100 ou augmenter le nombre d'élèves par classe.

5. Compléments

La solution théorique peut se mener avec les élèves, en prolongeant le raisonnement tenu pour le problème 1.

- Dans le lancer d'un seul dé, le Chevalier de Mérée perd la partie s'il n'y a pas de « 1 » sur une ligne obtenu sur l'écran, il y a $5\times5\times5\times5$ cas sur les $6\times6\times6\times6$ possibles donc sa probabilité de gagner est de $1-\left(\frac{5}{6}\right)^4\approx0{,}518$.

- L'utilisation de la fonction **meree** dans le cas de deux dés cache un peu le raisonnement. Il faut alors extrapoler celui du cas d'un dé : sur un lancer il y a 36 cas possibles, et parmi eux 35 cas pour lequel le pari n'est pas gagné donc la probabilité de gagner sur 24 lancers est de $1-\left(\frac{35}{36}\right)^{24}\approx0{,}491$. En revanche, pour 25 lancers, $1-\left(\frac{35}{36}\right)^{25}\approx0{,}506$; c'est le nombre de lancers qu'a proposé Pascal, mais le chevalier devait être un joueur acharné pour avoir senti la différence !