**Simulation de la loi binomiale sur calculatrice**

**Exercice**

Dans une urne, il y a : .

1. On tire successivement, **avec remise**, 3 jetons de l’urne.

On appelle succès, l’évènement *B*=« Obtenir une jeton blanc »

1. Construire l’arbre pondéré qui décrit cette expérience.
2. On appelle *X* la variable aléatoire qui compte le nombre de succès.

Quelle est la loi suivie par *X* ? Préciser ses paramètres.

1. Construire la table de probabilité de *X*.
2. On recommence l’expérience, mais cette fois, le tirage s’effectue **sans remise**.
3. Construire l’arbre pondéré qui décrit cette expérience.
4. On appelle *Y* la variable aléatoire qui compte le nombre de succès.

Pourquoi *Y* ne suit-elle pas une loi binomiale ?

1. Construire la table de probabilité de *Y*.
2. Algorithmique
3. Écrire un algorithme qui simule un schéma de Bernoulli de paramètres *n*=3 et *p*=0,7
4. Traduire cet algorithme pour une calculatrice.

**Solution**

1. Tirages avec remise
2. arbre



1. Quelle est la loi suivie par *X* ?

L’expérience est un schéma de Bernoulli car, à chaque tirage, on a 2 issues :

* le jeton est blanc (succès de probabilité 0,7)
* le jeton est noir (échec de probabilité 0,3)

Les 3 tirages sont identiques et indépendants

Donc la variable *X* qui compte le nombre de succès suit la loi binomiale *B*

1. Table de la loi de *X*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Résultat | *BBB* | *BB* | *BB* | *B* | *BB* | *B* | *B* |  |
| *k*=nb succès | 3 | 2 | 2 | 1 | 2 | 1 | 1 | 0 |
| prob. | =0,343 | ×0,3  =0,147 | ×0,3  =0,147 | 0,7×  =0,063 | ×0,3  =0,147 | 0,7×  =0,063 | 0,7×  =0,063 | =0,027 |

En résumé :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nb succès : *k* | 0 | 1 | 2 | 3 |
| *p*(*X*=*k*) | 0,027 | 3×0,063=0,189 | 3×0,147=0,441 | 0,343 |

1. Le tirage s’effectue **sans remise** cette fois.
2. Arbre



1. Pourquoi la variable *Y* ne suit-elle pas une loi binomiale ?

Elle ne suit pas une loi binomiale car nous ne sommes pas en présence d’un schéma de Bernoulli :

* on a bien une répétition de 3 épreuves de Bernoulli (succès-échec)
* les épreuves à 2 issues ne sont pas identiques (la probabilité d’un succès varie)
* les épreuves à 2 issues ne sont pas indépendantes (le résultat d’une épreuve influence le résultat de la suivante)

1. Table de probabilité de la loi suivie par *Y* :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Résult. | *BBB* | *BB* | *BB* | *B* | *BB* | *B* | *B* |  |
| *k*=nb succès | 3 | 2 | 2 | 1 | 2 | 1 | 1 | 0 |
| prob. | = | = | = | = | = | = | = | = |

En résumé :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nb succès : *k* | 0 | 1 | 2 | 3 |
| *p*(*X*=*k*) | ó0,023 | =0,1875 | =0,455 | ó0,334 |

1. Algorithmique
2. Écrire un algorithme qui simule un schéma de Bernoulli de paramètres *n*=3 et *p*=0,7

Début

Initialisations : *n*=3 et *p*=0,7

Mettre à zéro les *n* valeurs de la liste 1

Pour *k* de 1 à *n* :

Choisir un nombre *x* au hasard dans l’intervalle

Si *xÂp* alors la *k*e valeur de la liste 1 reçoit 1 (succès)

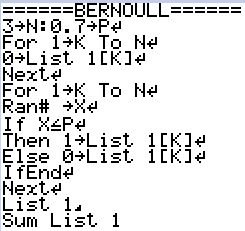
sinon la *k*e valeur de la liste 1 reçoit 0 (échec)

Fin pour

Afficher la liste 1 (contenant les *n* tirages)

Afficher la somme de la liste 1 (contenant le nombre de succès)

Fin

1. Programme sur Casio ci-contre :

Utilisation :

On obtient d’abord la liste des 3 tirages (ici 0-0-1 c’est-à-dire echec-échec-succès) :

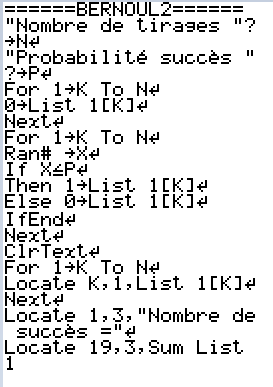


Puis on obtient le nombre de succès (ici 1) :



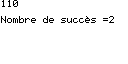
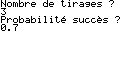
1. Améliorations du programme

On peut demander à l’utilisateur de saisir les valeurs de N et de P et afficher la liste des résultats sur une seule ligne :



U

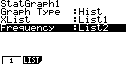
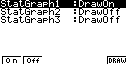
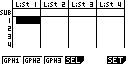
Utilisation :

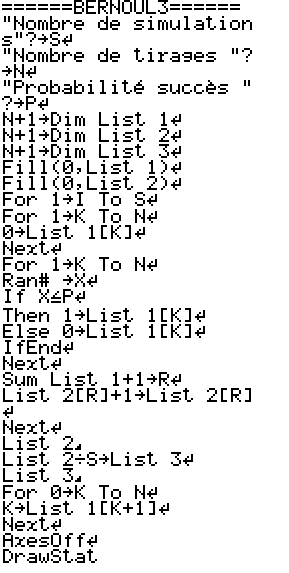


Pour améliorer encore le programme, on peut demander l’affichage d’un histogramme.

Il faut alors commencer par faire les réglages dans le menu statistiques comme ceci :

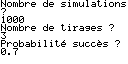
Menu STAT F1 SEL Exit/SET





On utilise alors le programme BERNOUL3 donné ci-contre :

Utilisation (avec 1000 simulations) :



on appuie sur EXE pour obtenir la répartition des 1000 simulations (conservée dans la Liste 2) :



Les 4 nombres de la liste 2 représentent respectivement :

* le nombre de simulations avec *X*=0 (ici 20)
* le nombre de simulations avec *X*=1 (ici 171)
* le nombre de simulations avec *X*=2 (ici 469)
* le nombre de simulations avec *X*=3 (ici 340)

On obtient ensuite (en appuyant sur EXE) les fréquences (conservée dans la Liste 3) :



Enfin, en appuyant sur EXE, on obtient l’histogramme

