
TP 14 : Simulation de variables aléatoires discrètes.

Le but de ce TP est de réussir à écrire un programme simulant diverses variables aléatoires (suivant une loi géométrique ou de Poisson ou à partir d'une expérience).

1 Lois usuelles

1.1 La loi géométrique.

- ▷ Rappeler le lien entre une variable aléatoire X suivant une loi géométrique de paramètre p et une expérience de Bernoulli de paramètre p .

- ▷ Rappeler comment on simule une expérience de Bernoulli (2 façons)

- ▷ En déduire un programme `SimuGéom(p)` permettant de simuler une variable aléatoire suivant une loi géométrique de paramètre p et la tester avec le paramètre $\frac{1}{10}$.

- ▷ Comparer avec la commande `rd.geometric(1/10)`

Exercice 1. Bob participe à un jeu dont les règles sont les suivantes :

- Pour pouvoir participer il doit miser une fois 10€
- Ensuite il lance une pièce truquée ayant une probabilité $\frac{1}{4}$ de tomber sur Pile jusqu'à l'obtention d'un Pile.
- Il gagne alors 2€ par lancer effectué.

1. On note X le nombre de lancers effectués par Bob. Quelle est la loi de X ?
2. Soit Y le gain de Bob. Quelle lien y a-t-il entre Y et X ?
3. En déduire un programme permettant de simuler Y .

2 A partir d'une expérience concrète.

Exercice 2 (ECRICOME 2017 voie S).

Pour tout entier n strictement positif, on considère l'expérience suivante : on dispose de n urnes initialement vides, numérotées de 1 à n et on dispose d'un grand stock de boules que l'on dépose une à une dans ces urnes. Pour chaque boule, on choisit au hasard, de façon équiprobable, l'urne dans laquelle la boule est déposée.

On note X_n le rang du premier tirage pour lequel une des urnes contiendra deux boules.

Compléter la fonction Python suivante pour qu'elle simule une réalisation de la variable aléatoire X_n :

```

1 def tirage(n):
2     urnes = np.zeros(n)
3     X = 1
4     choix = 0
5     while .....
6         choix = rd.randint(0,n-1)
7         urnes[choix] = .....
8         X = .....
9     return X

```

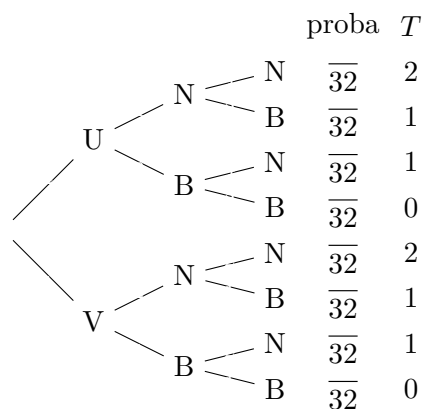
Exercice 3 (ECRICOME 2017 voie T).

1. L'urne U contient toujours 1 boule noire et 3 boules blanches indiscernables au toucher. L'urne V contient 2 boules noires et 2 boules blanches indiscernables au toucher.

On lance une pièce équilibrée. Si elle retombe sur le côté Pile, on tire deux boules successivement et avec remise dans U, et si on obtient Face, on tire deux boules successivement et avec remise dans V.

On note T la variable aléatoire égale au nombre de fois où l'on a pioché une boule noire.

- (a) Que vaut $T(\Omega)$?
- (b) Donner la loi de T . On vérifiera que $P(T = 1) = \frac{7}{16}$. Pour trouver la loi on pourra compléter l'arbre ci-dessous.



t_i			
$P(T = t_i)$			

- (c) On rappelle qu'en langage Python l'instruction `rd.randint(n1,n2)` renvoie un entier au hasard et uniformément compris entre `n1` et `n2`, `n2` exclu. Compléter, sur votre copie, le programme Python suivant afin qu'il affiche une simulation de la variable aléatoire T .

```

1 def simulT():
2     T = .....
3     if rd.randint(1,3) == 1 :
4         for k in range(2):
5             if rd.randint(1,5)<2 :
6                 T = T+1
7     else :
8         .....
9         .....
10        .....
11        .....
12        .....
13    return T

```

2. On procède cette fois-ci dans U (1 boule noire et 3 boules blanches) à une suite de tirages avec remise d'une boule jusqu'à obtenir la boule noire. On appelle Y la variable aléatoire égale au nombre de tirages effectués.

(a) Quelle est la loi de Y ?

On précisera $Y(\Omega)$ et $P(Y = k)$ pour tout $k \in Y(\omega)$.

(b) Compléter le programme Scilab suivant afin qu'il affiche une simulation de la variable aléatoire Y .

```

1 def simulY():
2     Y=1
3     while rd.random()>=.....:
4         Y=.....
5     return Y

```