Loi binomiale, cours, terminale STMG

1 Loi de Bernoulli

Définition :

Soit p un nombre réel tel que $p \in [0;1]$. Soit X une variable aléatoire. On dit que X suit une loi de Bernoulli de paramètre p si :

- X prend pour seules valeurs 1 (« succès ») et 0 (« échec ») ;
- $P(X=1) = p$ et $P(X=0) = 1-p$.

Propriété :

Soit X une variable aléatoire qui suit la loi de Bernoulli de paramètre p. Alors son *espérance* est

$$E(X) = p$$

Preuve :

D'une part, $E(X) = 0 \times P(X=0) + 1 \times P(X=1) = 0 \times (1-p) + 1 \times p = p$.

Algorithmique :

Algorithme de simulation d'une épreuve de Bernoulli de paramètre p.

Données : p : nombre décimal entre 0 et 1 ;
Début traitement
  t prend une valeur aléatoire décimale entre 0 inclus et 1 exclu ;
  si $t < p$ **alors**
    Afficher "Succès" ;
  fin
  sinon
    Afficher "Échec" ;
  fin
Fin

Exemple :

TI :

```
Prompt P
NbreAleatoire() ▷ T
If T < P
Then
Disp "SUCCES"
Else
Disp "ECHEC"
```

Casio :

```
"P"? → P
Rand# → T
If T < P
Then "SUCCES"
Else "ECHEC"
```

XCas :

```
saisir("Entrer p : ",p);
t:=alea(0,1);
si (t<p) alors
  afficher("Succès");
sinon
  afficher("Echec");
fsi;
```

Python :

```
from random import *
def simulation_Bernoulli(p):
    t=random()
    if (t<p):
        return "Succès"
    else:
        return "Echec"
```

2 Schéma de Bernoulli

Définition :

Deux expériences sont dites indépendantes si le résultat de l'une n'influence pas le résultat de l'autre.

Exemple :

il y a indépendance lorsqu'on lance deux fois de suite une pièce de monnaie.

Définition :

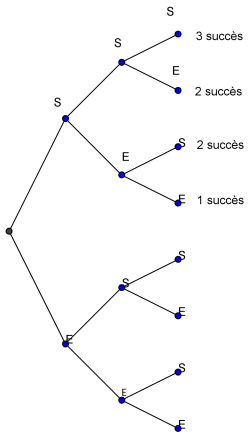
La répétition d'une expérience aléatoire suivant une loi de Bernoulli de paramètre p, ceci n fois de manière indépendante, constitue *un schéma de Bernoulli* de paramètres n et p.

Propriété :

Dans un schéma de Bernoulli, la probabilité d'une liste de résultats est le produit des probabilités de chaque résultat

Exemple de savoir faire :

- **[Construire un arbre représentant un schéma de Bernoulli de paramètres donnés]**
  On contrôle la qualité d'un produit sur une chaîne de production. On prélève 3 produits au hasard. On suppose que les prélèvements sont indépendants. Statistiquement, chaque produit a une probabilité $p = 0,05$ d'être défectueux.
  On a donc un schéma de Bernoulli de paramètres $p = 0,05$ et $n = 3$.



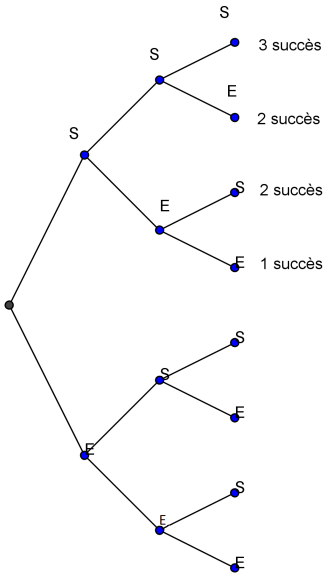
- **[Calculer la probabilité d'un événement représenté par un chemin sur un arbre pondéré]**
  Sur l'arbre ci-dessus représentant un schéma de Bernoulli de paramètres $n = 3$ et $p = 0,05$, la probabilité d'avoir les deux premières expériences qui donnent un succès et la dernière qui donne un échec est $P(SS\bar{S}) = 0,05^2 \times 0,95 \approx 0,002$
  soit 0,2 % de chances d'avoir les deux premiers produits défectueux et le dernier en bon état sur les trois prélevés.

3 Loi binomiale

Définition et propriété :

Soient n et k deux entiers naturels avec $k \leq n$. On note $\binom{n}{k}$ (on dit « k parmi n ») le nombre de manières d'obtenir k succès et $n - k$ échecs pour n répétitions indépendantes de la même expérience de Bernoulli.

Exemple :



$\binom{3}{3} = 1$: il y a une seule manière d'obtenir 3 succès lors de la répétition de 3 épreuves identiques indépendantes.
$\binom{3}{2} = 3$: il y a trois manières d'obtenir 2 succès et un échec lors de la répétition de 3 épreuves identiques indépendantes (SSE ; SES ; ESS).

Définition :

On considère une épreuve de Bernoulli de paramètre $p \in [0;1]$. On répète n fois ($n \geq 1$) cette expérience indépendamment et on note X la variable aléatoire qui compte le nombre de succès. On dit alors que la variable aléatoire X suit une loi *binomiale* de paramètres n et p et on note $X \sim \mathcal{B}(n;p)$.

Propriété :

Si X est une variable aléatoire qui suit une loi binomiale de paramètres n et p, alors la probabilité d'obtenir k succès avec $k \in \{0;1;2;...;n\}$ est

$$P(X=k) = \binom{n}{k} p^k (1-p)^{n-k}$$

.

Preuve :

Il y a $\binom{n}{k}$ manières d'obtenir k succès dans n répétitions d'expériences identiques et indépendantes. La probabilité de chacune de ces événements qui sont évidemment incompatibles est $p^k(1-p)^{n-k}$ d'où le résultat.

Exemple de savoir faire :
[Calculer la probabilité de $P(X = k)$ où X suit une loi binomiale à partir d'un arbre pondéré]
On considère le problème précédent de test des produits d'une chaîne de production. Les prélèvements étant supposés indépendants les uns des autres, l'expérience constitue un schéma de Bernoulli de paramètres $n = 3$ et $p = 0,05$. La variable aléatoire X qui compte le nombre de succès suit la loi binomiale de paramètres $n = 3$ et $p = 0,05$.
On a $P(X = 2) = P(SS\bar{S} \cap S\bar{S}S \cap \bar{S}SS)$
car trois chemins permettent d'obtenir deux succès c'est à dire deux objets défectueux.
D'où $P(X = 2) = P(SS\bar{S}) + P(S\bar{S}S) + P(\bar{S}SS)$
donc $P(X = 2) = 0,05^2 \times 0,95 + 0,05 \times 0,95 \times 0,05 + 0,95 \times 0,05^2 = 3 \times 0,05^2 \times 0,95 = 0,007$
soit une probabilité très faible de 0,007 d'avoir deux produits défectueux.

Remarques :

- On a $P(X < k) = P(X \leq k - 1)$
- pour calculer $P(X > k)$, on calcule $1 - P(X \leq k)$.

Exemples :

- $P(X < 2) = P(X \leq 1) = P(X = 0) + P(X = 1)$.
  Comme $P(X = 0) = 0,95^3 \approx 0,857$
  et $P(X = 1) = P(S\bar{S}\bar{S}) + P(\bar{S}S\bar{S}) + P(\bar{S}\bar{S}S) = 3 \times 0,95^2 \times 0,05 \approx 0,135$
  On obtient $P(X < 2) \approx 0,857 + 0,135 = 0,992$.
- $P(X > 1) = 1 - P(X \leq 1)$
  On a vu que $P(X \leq 1) \approx 0,992$
  on obtient $P(X > 1) \approx 1 - 0,992 = 0,008$

Propriété :
Soit X une variable aléatoire qui suit une loi binomiale de paramètres n et p. Alors :
L'espérance mathématique notée $E(X)$ est égale à

$$E(X) = np$$

L'espérance mathématique donne le nombre moyen de succès sur un grand nombre de simulations de n essais.

Preuve :
Admise

Exemple :
Dans l'exemple du chapitre,
$E(x) = np = 3 \times 0,05 = 0,15$
soit en moyenne, 0,15 objets défectueux sur 3 tests sur un grand nombre de tests donc 15 objets défectueux en moyenne sur 300 tests.

Algorithmique :
Algorithme de simulation d'une loi binomiale de paramètres n et p.

Données : p : nombre décimal entre 0 et 1 ; n : entier naturel ;
Début traitement
  c prend la valeur 0 ;
  pour k *de 1 jusque* n **faire**
    t prend une valeur aléatoire décimale entre 0 inclus et 1 exclu ;
    si $t < p$ **alors**
      c prend la valeur de $c + 1$;
    fin
  fin
Fin
Sorties : c

Exemple :

TI :

```
Prompt N
Prompt P
0 ▷ C
For(K,1,N) NbreAleatoire() ▷ T
If T < P
Then
C + 1 ▷ C
End
End
Disp C
```

Casio :

```
"P"? → P
"N"? → N
0 → C
For 1 → K To N Step 1
Rand# → T
If T < P
Then C + 1 → C
IfEnd
Next
"C=" :C
```

XCas :

```
saisir("Entrer p : ",p);
saisir("Entrer n : ",n);
c:=0;
pour k de 1 jusque n faire
  t:=alea(0,1);
  si t<p alors
    c:=c+1;
  fsi;
fpour
afficher("Nombre de succès : "+c);
```

Python :

```
from random import*
p=float(raw_input("Entrer p : "))
n=int(raw_input("Entrer n : "))
c=0
for k in range(1,n+1):
  t=random()
  if t<p:
    c=c+1
print("Nombre de succès",c)
```

Algorithmique :

Algorithme de simulation d'obtention de N échantillons de lois binomiales de paramètres n et p.

Données : n, N : nombres entiers ;p : nombre décimal entre 0 et 1 ;
Début traitement
  s est une liste vide ;
  pour k *de 0 jusque* n **faire**
    $s[k]$ prend la valeur 0 ;
  fin
  pour m *de 1 jusque* N **faire**
    c prend la valeur 0 ;
    pour k *de 1 jusque* n **faire**
      t prend une valeur aléatoire décimale entre 0 inclus et 1 exclu ;
      si $t < p$ **alors**
        c prend la valeur de $c + 1$;
      fin
    fin
    $s[c]$ prend la valeur de $s[c] + 1$;
  fin
  Afficher s ;
Fin

Exemple :

XCas :

```
saisir("Entrer p : ",p);
saisir("Entrer n : ",n);
saisir("Entrer N : ",N);
s:=[];
pour k de 0 jusque n faire
  s[k]:=0;
fpour;
pour m de 1 jusque N faire
  c:=0;
  pour k de 1 jusque n faire
    t:=alea(0,1);
    si t<p alors
      c:=c+1;
    fsi;
  fpour;
  s[c]:=s[c]+1;
fpour;
pour k de 0 jusque n faire
  afficher("Avec "+k+" succès : "+s[k]);
fpour;
```

Python :

```
from random import*
p=float(raw_input("Entrer p : "))
n=int(raw_input("Entrer n : "))
N=int(raw_input("Entrer N : "))
s=[]
for k in range(0,n+1):
  s.append(0)
for m in range(1,N+1):
  c=0
  for k in range(1,n+1):
    t=random()
    if t<p:
      c=c+1
  s[c]=s[c]+1
for k in range(0,n+1):
print "Nombre avec "+k+" succès : "+s[k]
```

4 Coefficients binomiaux

Propriétés :

- $\binom{n}{k} = \binom{n}{n-k}$;
- $\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$

Triangle de Pascal :

Le triangle de Pascal, du nom de Blaise Pascal, mathématicien français du XVII^{e} siècle qui le« redécouvra » en Occident (car il était connu avant en Orient et au Moyen-Orient) est une disposition permettant de visualiser et de calculer les coefficients binomiaux et qui s'appuie sur la formule précédente.

$\binom{0}{0} = 1$					
$\binom{1}{0} = 1$	$\binom{1}{1} = 1$				
$\binom{2}{0} = 1$	$\binom{2}{1} = 2$	1			
$\binom{3}{0} = 1$	$\binom{3}{1} = 3$	3	1		
1	4	6	4	1	
1	5	10	10	5	1

Explication de la construction : le nombre de la ligne n et de la colonne k est le coefficient binomial $\binom{n-1}{k-1}$. Il est obtenu en ajoutant le nombre situé au dessus (ligne $n-1$ et colonne k) au nombre de la colonne et de la ligne précédente (ligne $n-1$ et colonne $k-1$).
Par exemple, $\binom{3}{1} = 3$ est la somme de $\binom{2}{1} = 2$ et de $\binom{2}{0} = 1$.

1					
1	1				
1	2	1			
1	3	3	1		
1	4	6	4	1	
1	5	10	10	5	1

Algorithmique :
Algorithme de construction du triangle de Pascal.

```
Données : n : entier naturel ;
Début traitement
    a est un tableau vide ;
    pour k de 0 jusque n faire
        b est un tableau vide ;
        pour p de 0 jusque k faire
            b[p] prend la valeur 1 ;
        fin
        a[k] prend la valeur de b ;
    fin
    pour k de 1 jusque n faire
        pour p de 1 jusque k − 1 faire
            a[k][p] prend la valeur de a[k-1][l]+a[k-1][l-1] ;
        fin
        Afficher a[k] ;
    fin
Fin
```