

Énoncé

Voici la répartition en France des différents groupes sanguins selon les différents rhésus :

	O	A	B	AB
Rhésus +	36%	37%	9%	3%
Rhésus -	6%	7%	1%	1%

- On note p la probabilité qu'une personne prise au hasard en France soit du groupe O. Donner la valeur de p .
- On choisit au hasard un groupe de 10 personnes. On note X la variable aléatoire qui dénombre les personnes ayant un groupe O. Quelle loi suit X ?
- Afin de calculer des probabilités autour de X , on va construire le triangle de Pascal grâce à un script Python. Que faut-il taper en console pour obtenir les coefficients du triangle de Pascal permettant de calculer les valeurs de $p(X = k)$?
- En déduire la probabilité que 6 personnes aient un groupe O sur les 10 personnes choisies au hasard.
- La fonction suivante permet de déterminer la loi de X . Coder cette fonction **loiproba** en Python et l'exécuter pour obtenir la loi de X .

```
Fonction loiproba() :
    c est la liste des coefficients de la ligne 10 du triangle de Pascal
    p prend la valeur 0,42 et a prend la valeur 0,58
    l est une liste vide
    Pour i allant de 0 à 10
        u prend la valeur c[i]*pi*q10-i
        ajouter u à la liste l
    renvoyer l
```

On l'exécutera puis on complétera la tableau ci-dessous à 10^{-2} près.

k	0	1	2	3	4	5	6	7	8	9	10
$p(X = k)$											



Crédit photo : www.pexels.com - Artem Podrez

```
def newline(l):
    a=[1]
    n=len(l)
    for i in range(n-1):
        c=l[i]+l[i+1]
        a.append(c)
    a.append(1)
    return a
```

```
def pascal(n):
    l=[1]
    for i in range(n):
        l=newline(l)
    return l
```

1. Français de groupe O

Il y a 36% des français avec un groupe O+ et 6% avec un groupe O- cela représente donc un total de 42%. Donc $p=0,42$.

2. Loi de X

X suit une loi binomiale de paramètre $n=10$ et $p=0,42$.

3. Triangle de Pascal

Le fonction `newline` permet de construire la ligne suivante du triangle de Pascal à partir de la ligne courante. La fonction `pascal` qui prend comme paramètre un entier `n` construit toutes les lignes du triangle de Pascal jusqu'à la ligne `n` qu'elle renvoie. Il fallait donc entrer en console `pascal(10)` car X suit une loi binomiale de paramètre $n=10$.

On peut vérifier ces résultats en sortant du mode Python, avec notre calculatrice en appuyant sur  **PROB** **Combinaison**. On constate bien que les premiers coefficients correspondent bien.

4. Calcul de $p(X = 6)$

D'après le résultat de la question précédente $\binom{10}{6} = 210$.

On obtient ainsi $p(X = 6) = 210 \times 0,42^6 \times 0,58^4 \approx 0,13$.

5. Fonction `loiproba`

Pour obtenir une valeur approchée à 10^{-3} près on a utilisé l'instruction `round(u,3)` lorsqu'on ajoute `u` à la liste `l`.

Voici le tableau de la loi de X :

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de BINOME
>>> from BINOME import *
>>> loiproba()
[0.004, 0.031, 0.102, 0.196, 0.249, 0.216, 0.13, 0.054, 0.015, 0.002, 0.0]
>>> |
  
```

Remarque : Cette fonction peut facilement se généraliser en ajoutant deux paramètres : `n` pour obtenir la loi binomiale de paramètre `n` et `p`. On a appelé cette fonction `loiproba2` dont voici le script ci-contre.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de BINOME
>>> from BINOME import *
>>> pascal(10)
[1, 10, 45, 120, 210, 252, 210, 120, 45, 10, 1]
  
```

```

HISTORIQUE
.....252
10C0 .....1
.....10
10C1 .....45
.....120
10C2 .....
10C3 .....
  
```

```

ÉDITEUR : BINOME
LIGNE DU SCRIPT 0026
def loiproba():
    c=pascal(10)
    p,q=0.42,0.58
    l=[]
    for i in range(11):
        u=c[i]*(p**i)*(q**(10-i))
        l.append(round(u,3))
    return l
  
```

```

ÉDITEUR : BINOME
LIGNE DU SCRIPT 0035
def loiproba2(n,p):
    c=pascal(n)
    q=1-p
    l=[]
    for i in range(n+1):
        u=c[i]*(p**i)*(q**(n-i))
        l.append(round(u,3))
    return l
  
```