

## TP première ES : Simulation sous Python de la loi de décroissance radioactive

Ce TP a pour but de faire une simulation du phénomène de désintégration de noyaux radioactifs en se servant du langage Python pour effectuer la simulation et d'Excel pour en représenter, sous forme d'une courbe, les résultats

On considère pour cela une population à un instant  $t=0$  pris comme origine des temps.

Pour modéliser le caractère radioactif de chaque noyau, on considère qu'au bout d'une durée  $t$ , il a  $P$  chances sur 100 de se désintégrer, où  $P$  choisi par l'utilisateur du programme, est un entier compris entre 1 et 99. Autrement dit, cela revient, au bout d'une durée  $t$ , pour chaque noyau, à faire tourner une roue avec 100 numéros équiprobables et si la roue affiche un numéro compris entre 1 et  $P$ , considérer le noyau comme désintégré, ce qui peut être simulé dans le langage Python.

Voici les étapes de programmation

### 1<sup>ère</sup> étape :

Pour utiliser Python en ligne, se rendre sur le site : <https://repl.it/languages/Python3>

### 2<sup>ème</sup> étape :

Importer une bibliothèque contenant les fonctions générant des nombres aléatoires. La bibliothèque s'appelle **random** sous Python et on rajoute un astérisque **\*** pour tout importer. La fonction utilisée s'appelle **randrange**

Tester la commande dans Python dans un programme simulant un entier  $M$  aléatoire compris entre 1 et 100 avec le programme suivant:

```
from random import *
M=randrange(100)+1
print("nombre obtenu :",M)
```

question : pourquoi faut il ajouter un 1 dans la formule ?

### 3<sup>ème</sup> étape :

On fait lire le nombre  $N_0$  de noyaux initiaux et la probabilité (en pourcentage entier) de désintégration sur la durée  $t$  avec les commandes :

```
N0= int(input("Nombre de noyaux radioactifs à l'instant initial : "))
P= int(input("probabilité de désintégration sur une durée t (%) : "))
```

Question : pourquoi faut il rajouter la fonction `int()` qui renvoie un nombre entier à la fonction `input()`

#### 4<sup>ème</sup> étape :

Pour effectuer la simulation, on crée une liste appelée **noyaux** contenant un nombre  $N_0$  d'entiers qu'on initialise à 1, 1 signifiant qu'ils ne sont pas encore désintégrés.

Commande : `noyaux=[1 for i in range(N0)]`

On crée également une variable  $N$  qui dénombrera le nombre de noyaux restants à après chaque durée  $t$ .  $N$  est initialisée à  $N_0$ .

On crée une variable **compteur** qui détermine le nombre de durées  $t$  écoulées depuis l'instant origine. Cette variable est initialisée à 0.

#### 5<sup>ème</sup> étape :

On utilise une boucle **tant que** (**while** en anglais) pour effectuer des désintégrations sur une nouvelle durée  $t$  sous la condition qu'il reste des noyaux non désintégrés, soit  $N > 0$

Dans cette boucle, on crée une boucle **for** qui effectue des désintégrations au hasard (tirage d'un entier entre 1 et 100 dans la variable **roue**) sur les noyaux restants pendant la durée  $t$  concernée. Si la variable **roue** donne une valeur comprise entre 1 et  $P$  alors le noyau de rang  $i$  est désintégré, ce que l'on exprime en faisant passer la  $i$ ème valeur du tableau **noyaux** de 1 à 0.

On crée également une boucle qui compte à cette étape le nombre de noyaux restants, c'est-à-dire le nombre d'éléments du tableau **noyaux** qui ont une valeur égale à 1. On affiche le nombre de noyaux restants à cette étape.

Programme complet : pouvant être copié-collé dans la zone de saisie du langage Python :

```
from random import *
N0= int(input("Nombre de noyaux radioactifs à l'instant initial : "))
P= int(input("probabilité de désintégration sur une durée t (%) : "))
print("noyaux restants au bout de :")
noyaux=[1 for i in range(N0)]
N=N0
compteur=0
while N>0:
    compteur=compteur+1
    for i in range(N0):
        roue=randrange(100)+1
        if roue<=P:
            noyaux[i]=0
    N=0
    for i in range(N0):
        if noyaux[i]==1:
            N=N+1
    print(compteur," t : ",N)
```

### Simulation et visualisation des résultats:

Lancer le programme précédent sous Python et entrer les valeurs  $N_0 = 1000$  et  $P = 20$

Ouvrir Excel et dans la première colonne, saisissez en premier  $N_0$ , puis la liste de nombres renvoyés par Python

Avec la commande Insertion, graphique, nuage de points, faire apparaître l'évolution du nombre de noyaux non désintégrés en fonction de la durée écoulée depuis l'instant origine.

Estimer en unités de  $t$ , la valeur de la demi-vie, c'est-à-dire, la durée au bout de laquelle le nombre de noyaux restant est la moitié du nombre de noyaux initial.

Si vous avez du temps, reprendre la simulation toujours avec  $N_0 = 1000$  mais changer la valeur de  $P$  en mettant, 5 puis 35 et comparer les graphiques obtenus ainsi que les demi-vies au cas précédent.