

# Sts-Ciel 2 Probabilités conditionnelles

Le chikungunya est une maladie virale transmise d'un être humain à l'autre par les piqures de moustiques femelles infectées. Un test a été mis au point pour le dépistage de ce virus. Le laboratoire fabriquant ce test fournit les caractéristiques suivantes :

- La probabilité qu'une personne atteinte par le virus ait un test positif est de 0.98 ;
- La probabilité qu'une personne non atteinte par le virus ait un test positif est de 0.01.



On utilise ce test de dépistage dans une population « cible » dans laquelle 5 % des personnes sont atteintes par le chikungunya. Un individu est choisi au hasard dans cette population. On appelle :

- M l'évènement : « L'individu choisi est atteint du chikungunya »
- T l'évènement : « Le test de l'individu choisi est positif »

Dans cette activité, on se propose de répondre à la question « *Quelle est la probabilité que l'individu choisi soit atteint du chikungunya, sachant qu'il a un test positif* » en utilisant une technique de simulation. Celle-ci consiste à créer virtuellement une population conforme aux critères énoncés précédemment pour ensuite compter dans celle-ci, les individus qui sont ciblés par la question. Ce travail a été divisé en 3 parties.

## PARTIE 1. : SIMULER UNE POPULATION DANS LAQUELLE 5 % DES PERSONNES SONT ATTEINTES PAR LE CHIKUNGUNYA

Un individu est choisi au hasard dans cette population. La probabilité que cet individu soit malade est  $p(M) = 0.05$ . On peut simuler cette population atteinte ou non par le chikungunya, en utilisant la fonction `random()` de python :

⇒ Ouvrir un nouveau fichier sur Pyzo et l'enregistrer sous le nom `chikungunya_monNom.py`.

⇒ Exécuter dans la console la ligne suivante qui permet d'importer la fonction `random()` de la librairie `random` : `>>> from random import random`

⇒ Toujours dans la console, exécuter plusieurs fois la fonction `random()` :  
..... à chaque exécution, cette fonction renvoie un nombre réel aléatoire compris entre 0 et 1 inclus.  
« Ce nombre a ainsi 5 % de chance d'être compris dans l'intervalle  $[0 ; 0.05]$  et 95 % de chance d'être compris dans l'intervalle  $[0.05 ; 1]$  »

```
>>> random()
0.7556828862637028

>>> random()
0.666188032253304

>>> random()
0.14368965723399107
```

⇒ On peut vérifier ce propos en testant le code donné ci-contre (à écrire dans votre fichier `chikungunya_monNom.py`)

```
from random import random

#Fonctions

def test(p) :
    for i in range(100) :
        x = random()
        if x < p :
            print("Je suis malade, j'ai le chikungunya ")
        else : print("Je ne suis pas malade")
```

et en exécutant cette fonction plusieurs fois dans la console : `>>> test(0.05)`

on peut aussi réaliser l'exécution suivante pour voir la différence : `>>> test(0.5)`

## PARTIE 2. : PRISE EN COMPTE DES TESTS DE DEPISTAGE :

Le laboratoire fabriquant ce test fournit les caractéristiques suivantes :

- La probabilité qu'une personne atteinte par le virus ait un test positif est de 0.98 ;
- La probabilité qu'une personne non atteinte par le virus ait un test positif est de 0.01.

On se propose ici de créer une population virtuelle de  $n$  individus qui auront tous été testés. Chacun possède 2 attributs :

- o « M » si cet individu est malade du chikungunya ou « pas M » s'il ne l'est pas,
- o « T » si son test est positif ou « non T » s'il est négatif.

Numériquement, chaque individu sera un élément d'une liste nommé *population* de taille  $n$ . Cet élément sera lui-même une liste de 2 valeurs qui pourront être :

- o `['M', 'T']` si l'individu est malade et a été testé positif,
- o `['Non M', 'T']` si l'individu n'est pas malade mais a été testé positif,
- o `['M', 'Non T']` si l'individu est malade et a été testé négatif,
- o `['Non M', 'Non T']` si l'individu n'est pas malade mais a été testé négatif,

La fonction *genererPopulation()* donnée ci-contre est incomplète. Elle ne prend pas de paramètre et renvoie la liste *population* décrite ci-dessus.

⇒ Ecrire ce code en le complétant, dans votre fichier *chikungunya\_monNom.py* et exécuter cette fonction dans le programme principal afin d'y créer et retourner la liste *population*.

```
def genererPopulation() :
    population = []
    for i in range(n) :
        x = random()
        if x < 0.05 :
            population.append(["M",None])
        else : population.append(["Non M",None])

    for i in range(n) :
        x = random()
        if population[i][0] == "M":
            if x < 0.98 : population[i][1] = "T"
            else : population[i][1] = "Non T"
        else :

    return population

# Main
n = 1000
population = genererPopulation()
```

A compléter

⇒ Exécuter dans la console la ligne suivante afin d'afficher séparément tous les individus qui composent cette liste *population* :

```
>>> for individu in population : print(individu)
```

### PARTIE 3. : RECENSEMENT DES ATTRIBUTS DANS CETTE LISTE POPULATION :

La fonction *recensement()* donnée ci-dessous est incomplète. Elle possède 1 paramètre, la liste *population* du paragraphe précédent. Cette fonction renvoie un dictionnaire nommé *nombreIndividus* qui possède 8 clés qui sont données ci-contre. Les valeurs liées à ces clés sont égales au nombre d'individus malades pour la clé 'M', non malades et testés négatifs pour la clé 'Non M et Non T', etc ....

```
M
Non M
T
Non T
M et T
M et Non T
Non M et T
Non M et Non T
```

```
def recensement(population) :
    nombreIndividus = {
        "M":0,
        "Non M":0,
        "T":0,
        "Non T":0,
        "M et T":0,
        "M et Non T":0,
        "Non M et T":0,
        "Non M et Non T":0
    }
    for i in range(len(population)) :
        if population[i][0] == "M" : nombreIndividus["M"] += 1
        if population[i][0] == "Non M" : nombreIndividus["Non M"] += 1
        if population[i][1] == "T" : nombreIndividus["T"] += 1
        if population[i][1] == "Non T" : nombreIndividus["Non T"] += 1
        if population[i][0] == "M" and population[i][1] == "T" : nombreIndividus["M et T"] += 1

    return nombreIndividus

# Main
n = 1000000
population = genererPopulation()
nombreIndividus = recensement(population)
for cle in nombreIndividus :
    print(f"{cle} : ----> {nombreIndividus[cle]}")
```

*A compléter*

⇒ Ecrire ce code en le complétant, dans votre fichier *chikungunya\_monNom.py* et l'exécuter dans le programme principal, par exemple le résultat affiché par une exécution sur une population de  $n = 100\ 000$  individus était lors de la rédaction de ce document, celui-ci-dessous. Mais avec une autre exécution, on trouverait des résultats peu différents, mais différents tout de même car la composition de la population est déterminée aléatoirement.

```
>>> (executing file "chikungunia.py")
M : ----> 50054
Non M : ----> 949946
T : ----> 58430
Non T : ----> 941570
M et T : ----> 49089
M et Non T : ----> 965
Non M et T : ----> 9341
Non M et Non T : ----> 940605
```

## EXERCICE 4. : BILAN ET CONCLUSION :

Réaliser votre exécution et reporter ces résultats dans le tableau incomplet de probabilité ci-dessous :

	$M$	$\bar{M}$	Total
$T$	$card(T \cap M)$ 49089	$card(T \cap \bar{M})$	$Card(T)$
$\bar{T}$	$card(\bar{T} \cap M)$	$card(\bar{T} \cap \bar{M})$	$card(\bar{T})$ 941570
Total	$Card(M)$	$card(\bar{M})$	

Finalement, à la question « *Quelle est la probabilité que l'individu choisi soit atteint du chikungunya, sachant qu'il a un test positif* », on peut répondre ainsi :

On calcule la probabilité conditionnelle suivante :  $p_T(M) = \frac{card(T \cap M)}{card(T)}$  ... pour cela on vous demande d'écrire le script de la fonction *question()* suivante :

```
def question(nombresIndividus) :
    
# Main
n = 1000000
population = genererPopulation()
nombresIndividus = recensement(population)
for cle in nombresIndividus :
    print(f"{cle} : ----> {nombresIndividus[cle]}")
p = question(nombresIndividus)
print("La probabilité que l'individu choisi soit atteint du chikungunya, sachant qu'il a un test positif est : ",p)
```

Le résultat trouvé sera à peu près celui donné ci-dessous :

```
La probabilité que l'individu choisi soit atteint du chikungunya, sachant qu'il a un test positif est : 0.8362342995584117
```