

## Lancer de pièce et prise de décision.



Sur 35 lancers de pièce on a obtenu plus de 25 fois pile. Est-ce normal ? (On utilisera un niveau de confiance de 95%)

### Dans un script SIMUL1

1°) Ecrire une fonction `pf` qui prend comme argument  $n \in \mathbb{N}^*$  et qui renvoie une liste de  $n$  termes représentant  $n$  lancer d'une pièce.

On représentera face par 0 et pile par 1.

2°) Ecrire une fonction `simul` qui prend comme argument les entiers naturels non nuls  $n$  et  $p$ . On simulera  $n$  lancers d'une pièce et la fonction renvoie `True` si on a obtenu  $p$  (ou plus) fois pile et `False` sinon.

3°) Ecrire une fonction `frequence` qui prend comme arguments  $q, n, p$  des entiers naturels non nuls et qui effectue  $q$  fois  $n$  lancers de la pièce et renvoie la fréquence de l'événement « on a obtenu au moins  $p$  fois pile » lors des  $q$  simulations.

4°) Conclure s'il est normal d'obtenir plus de 25 fois pile lors de 35 lancers d'une pièce.

```
PYTHON SHELL
>>>
>>> from SIMUL1 import *
>>> pf(10)
[0, 0, 0, 1, 1, 0, 0, 1, 1, 1]
>>> pf(20)
[1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0]
```

```
PYTHON SHELL
>>>
>>> from SIMUL1 import *
>>> simul(10,6)
False
>>> simul(10,6)
True
```

```
PYTHON SHELL
>>> frequence(100,30,17)
0.31
>>> frequence(1000,30,17)
0.29
>>> frequence(10000,30,17)
0.2936
```

## Lancer de pièce et prise de décision.

Fonction `pf`

1°) On commence par initialiser la liste `liste` en écrivant :

`liste=[]`. On aurait pu aussi écrire `liste=list()`.

Dans une boucle `for` qui boucle `n` fois, on ajoute à la liste `liste` un entier aléatoire : 0 pour face et 1 pour pile.

On propose un seconde script plus concis réalisable grâce à la syntaxe Python.

```
ÉDITEUR : SIMUL1
LIGNE DU SCRIPT 0015

def pf(n):
    liste=[]
    for i in range(n):
        liste.append(randint(0,1))
    return liste
```

```
ÉDITEUR : SIMUL1
LIGNE DU SCRIPT 0008

from random import *
def pf(n):
    liste=[randint(0,1) for i in range(n)]
    return liste
```

Fonction `simul`

2°) A partir de la liste des faces et piles représentés par des 0 et des 1 dans `liste`, on additionne tous les termes de liste, ce qui correspondra au nombre de 1 présent dans la liste soit le nombre de fois où on a obtenu pile.

Si cette somme est supérieure ou égale à `p` on renvoie `True` et `False` sinon.

```
ÉDITEUR : SIMUL1
LIGNE DU SCRIPT 0021

def simul(n,p):
    liste=pf(n)
    if sum(liste)>=p:
        return True
    else:
        return False
```

Fonction `frequence`

3°) On recommence l'expérience précédente `q` fois. On va compter à chaque fois si on a obtenu plus de `p` piles ou non.

`k` est la variable qui correspond à ce nombre. Si `simul(n,p)` est vrai alors `k` est incrémenté de 1.

On remarque qu'on a écrit `if simul(n,p)`. On aurait pu aussi écrire `if simul(n,p)==True` de façon équivalente car `simul(n,p)` est un booléen.

On renvoie `k/q` qui correspond à la fréquence recherchée.

```
ÉDITEUR : SIMUL1
LIGNE DU SCRIPT 0028

def frequence(q,n,p):
    k=0
    for i in range(q):
        if simul(n,p):
            k=k+1
    return k/q
```

## Conclusion

4°) Conclusion : Il y a environ moins d'un pourcent de chance d'obtenir 25 fois la face pile lors de 35 lancers.

Au niveau de confiance de 95% cela n'est pas normal d'obtenir 25 fois la face pile.

```
PYTHON SHELL

>>> frequence(1000,35,25)
0.008
>>> frequence(1000,35,25)
0.009
>>> frequence(10000,35,25)
0.009299999999999999
>>> frequence(100000,35,25)
0.008710000000000001
>>> frequence(100000,35,25)
0.00808
```

