

NOM : Groupe

TP probabilités : le jeu des trois portes

Utilisation d'une simulation informatique en langage python

Première simulation

Un premier programme en python simule une partie de jeux des trois portes.

Un exemple de sorties (résultats) du programme est montré ci-dessous :

```
-----Simulation d'une partie du jeux des trois portes-----  
    Jeu de la télévision américaine présenté par Monty Hall  
La voiture est cachée derrière la {'porte3'}  
Le joueur choisit la {'porte2'}  
Monty Hall ouvre la {'porte1'}  
Les {'porte2', 'porte3'} sont fermées.  
Monthy Hall demande au joueur s'il veut changer son choix:  
    Non, le joueur ne change pas son choix, on ouvre la {'porte2'}  
        le joueur perd  
    Oui: le joueur change son choix, on ouvre la {'porte3'}  
        le joueur gagne la voiture
```

Compléter les sorties du code simulant une partie du jeux des trois portes :

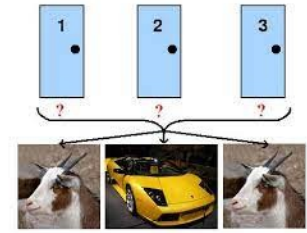
```
-----Simulation d'une partie du jeux des trois portes-----  
    Jeu de la télévision américaine présenté par Monty Hall  
La voiture est cachée derrière la {'porte2'}  
Le joueur choisit la {'porte3'}  
Monty Hall ouvre la {'      '}  
Les {'      ', '      '} sont fermées.  
Monthy Hall demande au joueur s'il veut changer son choix:  
    Non, le joueur ne change pas son choix, on ouvre la {'      '}  
        le joueur        
    Oui: le joueur change son choix, on ouvre la {'      '}  
        le joueur      
```

Un peu de probabilités

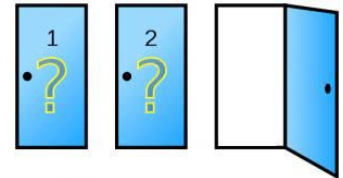
On note :

- V : l'événement *gagner la voiture*
- $P(V)$: la *probabilité de gagner la voiture*

Quelle est la probabilité $P(V)$ de gagner la voiture ? On donnera le résultat exacte sous forme d'une fraction, puis d'une valeur approchée au centième :



$$P(V) = \frac{\dots}{\dots}$$



Créé avec l'application **BDF** développée par la BNF

Problème :

Le présentateur ouvre une porte (ici la porte 3) et demande à un charmant petit couple qui participe au jeu, s'ils veulent changer leur premier choix.

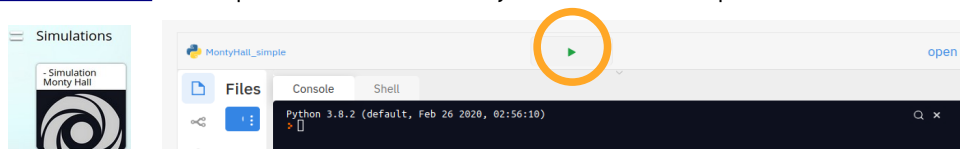
Athéna et Ancélade ne semblent pas d'accord sur la stratégie à tenir.

- Athéna pense qu'il faut changer son choix.
- Ancélade pense que non.

Pourrez-vous à l'aide d'une simulation du jeu écrite en python aider Athéna et Ancélade à choisir une stratégie ?

Usage de la première simulation

Cette première simulation est disponible en cliquant <https://replit.com/talk/share/Monty-Hall-Game-simulation/130522> ou depuis PearlTrees. Pour jouer une fois cliquer sur la flèche au milieu :



Noter vos résultats ci-dessous sous forme d'une fraction, puis d'une valeur décimale approchée

Nombre de jeux	Fréquence Gains (ne change pas)	Fréquence Gains (change d'avis)
5		
10		
20		

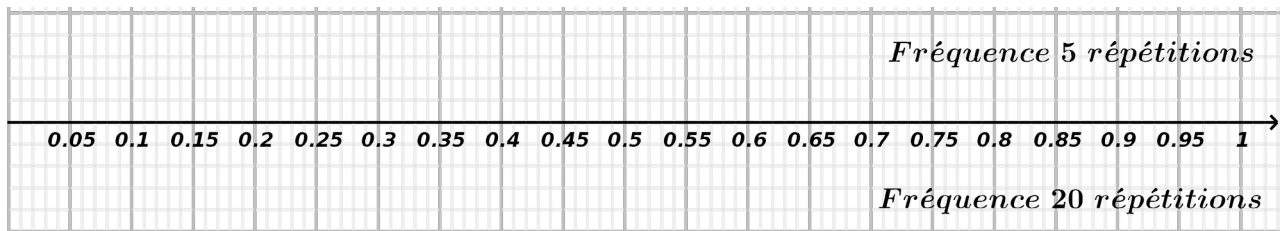
Rappel: $f_{gain} = \frac{\text{nbre gains}}{\text{nbre parties}}$

Analyse de la première simulation :

Noter pour tous les groupes les fréquences de gains pour chaque situation :

Nombre de répétitions du jeu	$f_{gainsAncélade}$ Ancélade ne change pas son premier choix	$f_{gainsAthéna}$ Athéna change son premier choix
5		
20		

Placer les fréquences de gains (avec changement du choix) sur l'axe ci-dessous :



Première conclusion :

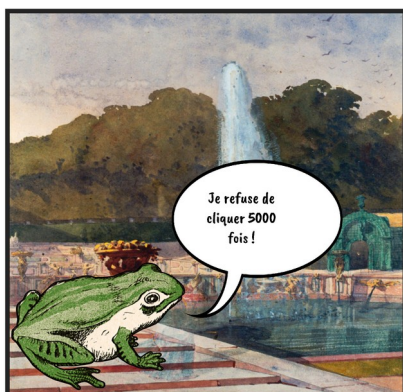
Lorsque ond'avis, onplus souvent, c'est donc qui a raison etqui est la

Toutes les fréquences calculées pour un même nombre de jeux sont-elles identiques ?

..... On dit que les fréquences des gains fluctuent (elles changent autour d'une valeur

Remarque :

Lorsque le nombre de répétitions du jeu augmente, les fréquences des gains (avec changement ou non du choix) tendent vers La fluctuation des gains diminue.



Que faire pour que la valeur des fréquences se rapproche de la probabilité de gagner en changeant son choix ?

.....

.....

.....

.....

Deuxième simulation



Créé avec l'application **Storyboard** développée par la BnF

Pensez-vous aussi que la chance de gagner soit, après ouverture d'une porte, de une chance sur deux ? Après tout, il reste deux portes fermées.

Une deuxième simulation permet de calculer la fréquence des gains automatiquement. Une partie du programme de la deuxième simulation est montrée ci-dessous :

```

9  """
10 # random : module permettant de faire des tirages aléatoires
11 import time
12 import random
13
14 print(time.strftime(' %d /%m/%Y %H:%M:%S'))
15 gain = 0
16 #Nombre de répétition du jeu
17 N_jeux = 5
18
19 print('le joueur ne change pas son choix')
20 for jeux in range(N_jeux):
21     #liste contenant le numéro de chacune des trois portes
22     portes = ['porte_1', 'porte_2', 'porte_3']
23     # On choisit une des trois portes
24     voiture = random.choice(portes)
    
```

le joueur ne change pas son choix

le joueur gagne : 2 sur 5

le joueur change son choix

En changeant son choix, le joueur gagne : 2 sur 5

Un exemple de résultat de cette simulation est donnée ci contre

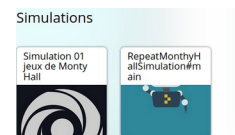
Quelle ligne faut-il modifier dans le programme pour augmenter le nombre de répétitions ?

.....

Ouvrir la simulation et compléter le tableau ci-dessous

<https://replit.com/@jeanpat/RepeatMonthyHallSimulation#main.py> ou avec depuis Pearltrees

Nombre répétitions				
Fréquence gain avec modification du choix				



Quelle est selon vous la probabilité de gagner en changeant le choix de la porte ?

.....

.....