Marche aléatoire - Casino.

Enoncé

On souhaite simuler les gains (et les pertes…) d'un joueur à la roulette.
Le joueur utilise la stratégie suivante : Il mise systématiquement 1€ sur le rouge.

1. Il y a 18 numéros rouges (qui feront gagner 1€ au joueur), 18 numéros noirs (le joueur perd sa mise) et un vert : le 0 où le joueur perd aussi.
Soit X la variable aléatoire représentant le gain (ou la perte) du joueur. Donner la loi de X en complétant le tableau ci-dessous :

x	-1	1
$p(X = x)$		

Quelle est l'espérance de X ? Que signifie ce résultat ?

2. Compléter la fonction Python **jeu** qui renvoie le gain du joueur lors d'un lancer de roulette.

3. On souhaite maintenant simuler **n** fois l'expérience aléatoire précédente et on nomme G le gain total lors de ces **n** expériences.

Compléter la fonction Python **gain** qui prend comme argument **n**, le nombre de jeux, et qui renvoie le gain total (le gain peut-être négatif si le joueur a perdu de l'argent). Lancer la fonction **gain** avec **n=500**.

4. Afin de visualiser l'évolution du gain dans le temps, on va représenter graphiquement le numéro du jeu en abscisse et le gain (cumulé) en ordonnée.
Compléter la fonction Python **graphe** qui prend en argument **n** le nombre total de jeux et qui affiche le nuage de point récherché. Lancer la simulation avec **n=500** plusieurs fois. Quelle tendance semble suivre ces graphiques ?

5. Soit Y la variable aléatoire représentant le nombre de fois où le joueur gagne lors des n jeux.

a) Quelle loi suit Y ?

b) En remarquant que $p\left(Y > \frac{n}{2}\right)$ représente la probabilité d'avoir un gain positif lors des **n** jeux, calculer cette probabilité pour $n = 500$ et $n = 2000$.

Crédit photo : www.pexels.com – Javon Swaby

```
ÉDITEUR : ROULETTE
LIGNE DU SCRIPT 0005
from random import *
def jeu():
  a=randint(1,37)
  if a<=18:
    return ...
  else:
    return ...
```

```
def gain(n):
  g=0
  for i in range(n):
    g=g+....
  return g
```

```
def graphe(n):
  g,xi,yi=0,[],[]
  for i in range(n):
    xi.append(i)
    yi.append(g)
    g=g+jeu()
  plt.cls()
  plt.auto_window(xi,yi)
  plt.axes("on")
  plt.plot(xi,yi,".")
  plt.show_plot()
```

1. Loi de probabilité

Il y a 37 numéros (de 0 à 36). Il y a 18 numéros rouges donc la probabilité de gagner 1€ est donc $p(X = 1) = \frac{18}{37}$ et on en déduit que $p(X = -1) = \frac{19}{37}$

x	-1	1
$p(X = x)$	$\frac{19}{37}$	$\frac{18}{37}$

$E(X) = 1 \times p(X = 1) + (-1)p(X = -1) = 1 \times \frac{18}{37} - 1 \times \frac{19}{37} = -\frac{1}{37} \approx -0{,}027.$
Cela signifie qu'en moyenne à chaque jeu le joueur va perdre 0,027€.

2. Fonction `jeu`

```
ÉDITEUR : ROULETTE
LIGNE DU SCRIPT 0007
from random import *
def jeu():
  a=randint(1,37)
  if a<=18:
    return 1
  else:
    return -1
```

a est un entier aléatoire compris entre 1 et 37. Il y a 18 numéros rouges, donc **`a<=18`** représente l'événement « le joueur gagne », il faut donc renvoyer **`1`** dans ce cas et dans le cas contraire, on renvoie **`-1`**.

3. Fonction `gain`

```
def gain(n):
  g=0
  for i in range(n):
    g=g+jeu()
  return g
```

La variable **`g`** représente le gain qui est initialisé à **`0`**. A chaque tour de boucle, il faut ajouter à **`g`** le résultat d'un jeu de roulette, ce qui se traduit par l'instruction **`g=g+jeu()`**.

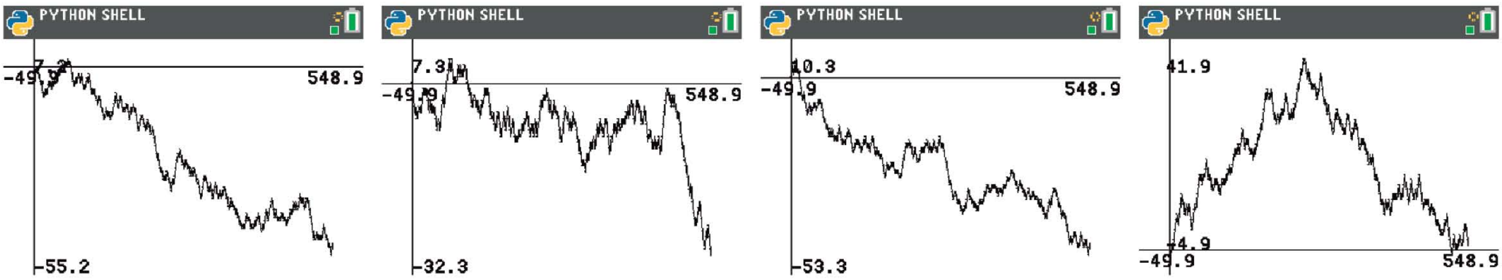
```
PYTHON SHELL
>>> gain(500)
32
>>> gain(500)
-8
>>> gain(500)
-16
>>> gain(500)
-16
>>> gain(500)
-18
>>> |
Fns… | a A # | Outils | Éditer | Script
```

On exécute le script et on lance la fonction gain plusieurs fois. On peut rappeler le nom de la fonction facilement en appuyant sur var .

On constate que le joueur a plus souvent de gains négatifs que de gains positifs.

4. Représentation graphique

On lance **`graphe(500)`** plusieurs fois et on obtient ce type de graphique :



5. Variable gain

a) Y suit une loi binomiale de paramètres n et $p = \frac{18}{37}$.

b) Pour avoir un gain positif il faut gagner plus de parties qu'en perdre ! Cela correspond donc à $p\left(Y > \frac{n}{2}\right)$. Pour utiliser la calculatrice on doit effectuer une transformation : $p\left(Y > \frac{n}{2}\right) = 1 - p\left(Y \leq \frac{n}{2}\right)$.

Pour $n = 500$, on calcule $p(Y \geq 250)$ grâce à 2nde var (distrib) et **`binomFrep`**. On complète la boite de dialogue, de même pour $n = 2000$. On obtient les résultats ci-contre :

```
NORMAL FLOTT AUTO RÉEL RAD MP
1-binomFRép(500,18/37,250)
                0.2580167309
1-binomFRép(2000,18/37,10▸
                0.1090832139
```