toy_notebook_fr

July 15, 2020

1 A propos du calcul de π

1.1 En demandant à la lib maths

Mon ordinateur m'indique que π vaut *approximativement*

[1]:
```
from math import *
print (pi)
```

```
3.141592653589793
```

1.2 En utilisant la méthode des aiguilles de Buffon

Mais calculé avec la **méthode** des aiguilles de Buffon, on obtiendrait comme **approximation** :

[2]:
```
import numpy as np
np.random.seed(seed=42)
N = 10000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

[2]:
```
3.128911138923655
```

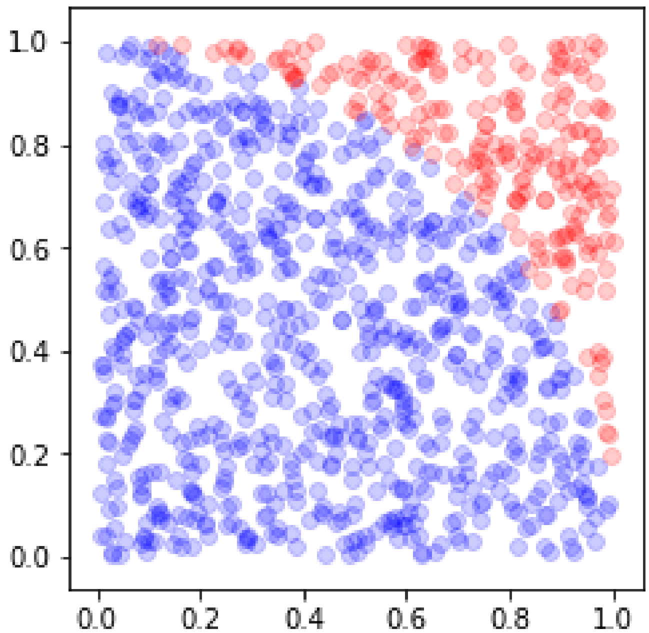
1.3 Avec un argument "fréquentiel" de surface

Sinon, une méthode plus simple à comprendre et ne faisant pas intervenir d'appel à la fonction sinus se base sur le fait que si $X \sim U(0,1)$ et $Y \sim U(0,1)$ alors $P[X^2 + Y^2 \leq 1] = \pi/4$ (voir méthode de Monte Carlo sur Wikipedia). Le code suivant illustre ce fait :

[3]:
```
%matplotlib inline
import matplotlib.pyplot as plt
np.random.seed(seed=42)
N = 1000
x = np.random.uniform(size=N, low=0, high=1)
y = np.random.uniform(size=N, low=0, high=1)
```

```
accept = (x*x+y*y) <= 1
reject = np.logical_not(accept)

fig, ax = plt.subplots(1)
ax.scatter(x[accept], y[accept], c='b', alpha=0.2, edgecolor=None)
ax.scatter(x[reject], y[reject], c='r', alpha=0.2, edgecolor=None)
ax.set_aspect('equal')
```



Il est alors aisé d'obtenir une approximation (pas terrible) de π en comptant combien de fois, en moyenne, $X^2 + Y^2$ est inférieur à 1 :

[4]:
```
4*np.mean(accept)
```

[4]: 3.112