

Document initial dans son environnement

JupyterLab interface showing a document titled "example_pi.ipynb".

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement*

```
In [1]: from math import *\nprint(pi)\n3.141592653589793
```

Mais calculé avec la méthode des aiguilles de Buffon (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtiendrait comme approximation :

```
In [2]: import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(sum((x+np.sin(theta))>1)/N)\nOut[2]: 3.1437198684098765
```

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).

```
In [3]: %matplotlib inline\nimport matplotlib.pyplot as plt\n\nmu, sigma = 100, 15\nx = mu + sigma*np.random.randn(10000)\n\nplt.hist(x,40)\nplt.grid(True)\nplt.show()
```



A histogram showing the distribution of 10,000 random samples generated from a normal distribution with mean 100 and standard deviation 15. The x-axis ranges from 40 to 160, and the y-axis (frequency) ranges from 0 to 800. The distribution is bell-shaped and centered around 100.

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

3.141592653589793

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

```
import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(sum((x+np.sin(theta))>1)/N)
```

3.1437198684098765

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et

des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).

