

# Lancers de pièces ("pile ou face")

On considère des lancers de pièces, "pile ou face" ("[Coin flipping](#)", "coin tossing", or "heads or tails" en anglais), en faisant l'hypothèse d'une probabilité égale d'occurrence des 2 possibilités.

- expérimenter à l'aide de pièces, par exemple faire des séries de 10 lancers (ou lancers de 10 pièces) en comptabilisant les nombres de "pile" et de "face"
- comptabiliser (tableau, distribution)
- aborder le problème théoriquement (distribution binomiale et triangle de Pascal)
- tenter une comparaison
- Amélioration de la statistique :
  - effectuer beaucoup plus d'expériences (temps nécessaire ?)
  - passer à une simulation ?

## Simulation

Voici un programme permettant de simuler et traiter des séries de lancers de pièces. Le programme peut être complété par une analyse statistique des séries (moyennes et écart-types de "séries de séries") et par des représentations graphiques des résultats avec comparaison à la distribution "théorique" (binomiale).

```
<sxh python; title : coin_tossing-01.py> #!/usr/bin/env python # -*- coding: utf-8 -*- """ Statistics on coin flipping series of given number of coins. Comparison with infinite mean solution : binomial distribution and Pascal's triangle. """
```

```
import numpy as np import matplotlib import matplotlib.pyplot as plt import random import collections
```

```
def nheads(n):
```

```
    """
    return number of heads for n equally likely outcomes coin flipping
    """
    return sum([random.choice(values) for i in range(n)])
```

```
values = [0,1] # tail or head nflips = 10 # Pascal's triangle : pt = [1],[1,1] for i in range(len(pt),nflips+1):
```

```
    pt.append([1]+[pt[i-1][j-1] + pt[i-1][j] for j in range(1,i)]+[1])
```

```
print pt[nflips], sum(pt[nflips]) # theoretical distribution
```


```
nb = 102400 heads = [nheads(nflips) for j in range(nb)] #print heads c = collections.Counter(heads)
for i in range(nflips+1):
```

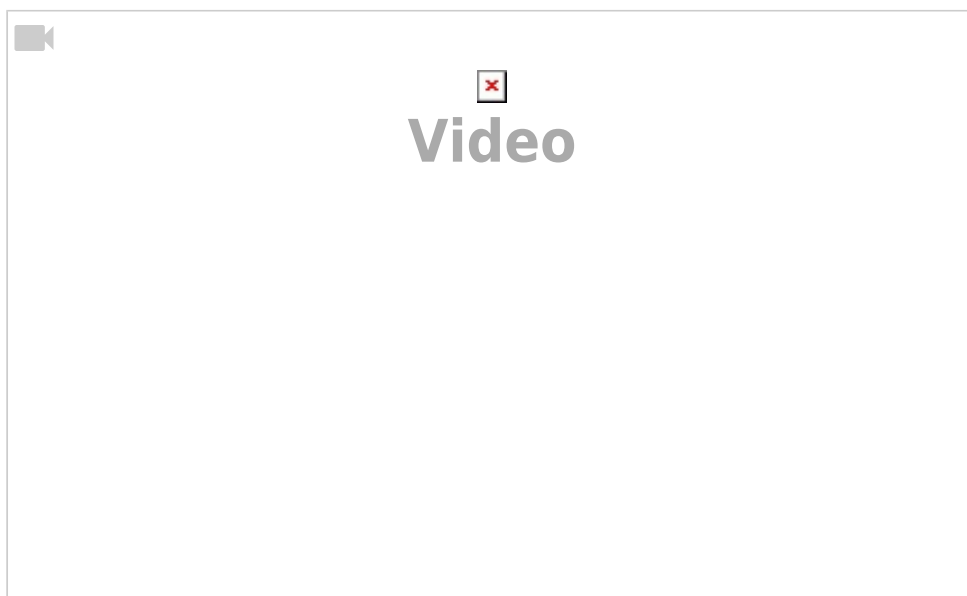
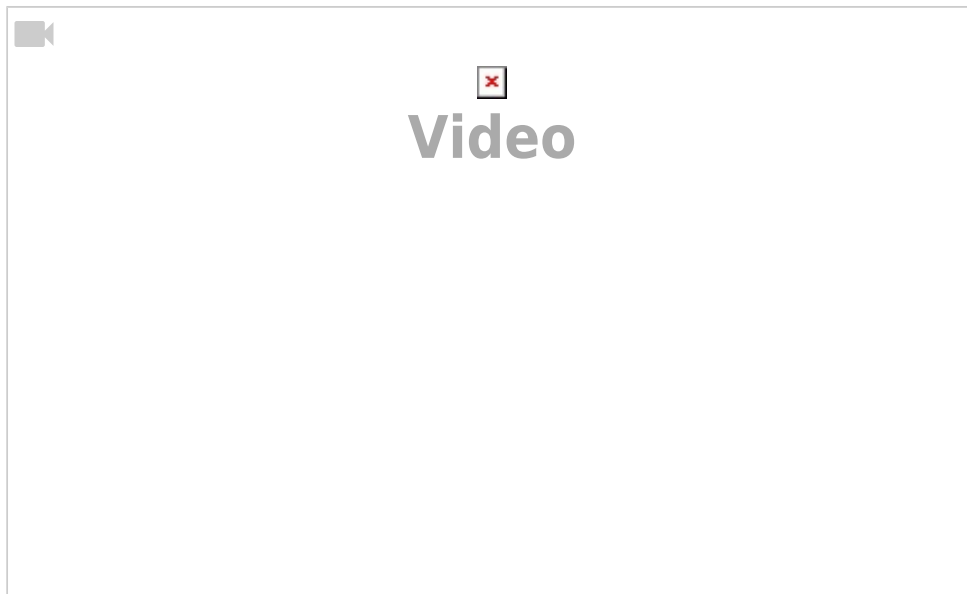
```
    print i, c[i], pt[nflips][i]
```

```
</sxh>
```

Exécution en accès public : [ici](#)

## Binomiale expérimentale : Galton board

-  [Bean machine](#) ou « Galton board »



From:  
<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:  
[https://dvillers.umons.ac.be/wiki/teaching:exos:lancer\\_pieces?rev=1489012434](https://dvillers.umons.ac.be/wiki/teaching:exos:lancer_pieces?rev=1489012434)

Last update: **2017/03/08 23:33**

