
Nesta aula: introdução à utilização de números
aleatórios na programação

Uso de números pseudo-aleatorios em programação:

- Simulação de acontecimentos probabilísticos
com vista a estimar parâmetros estatísticos - media...
 - lançamento de dados
 - filas de espera (atendimento...)
 - tempos de acesso médio de servidores da internet
 - Metodo de Monte Carlo...
 - Algoritmos aleatorizados
 - Algoritmos eficientes de primalidade
 - ...
 - Criptografia de chave pública
-

1) Geração de aleatórios

Muito difícil obter números verdadeiramente aleatórios nos computadores clássicos...

Pseudo-aleatórios, método linear congruencial

$$n[i+1] = a*n[i] + b \pmod{m}$$

para a, b, m apropriadamente escolhidos

Os pseudo-aleatórios podem ser apropriados para algumas aplicações mas não para outras (criptografia, por exemplo)

Em python

```
import random
>>> import random
>>> random.randint(2,4)
4
>>> random.randint(2,4)
2
>>> random.randint(2,4)
3
>>> random.randint(2,4)
2
>>> random.randint(2,4)
3
```

Só usaremos as funções:

```
randint(a,b) -- inteiro aleatório com igual probabilidade em [a,b]
uniform(a,b) -- 'float' aleatório com igual distribuição de
                probabilidade em [a,b]
```

2) Probabilidade (aproximada) de um acontecimento

Atira-se 5 vezes uma moeda ao ar; qual a probabilidade de saírem 4 ou cinco "coroas"?

a) Mais geralmente, escreva a função

```
coroas(n,c)
```

```
que retorna | True se saíram c ou mais coroas
            | False caso contrario
```

-->

```
from random import *

def coroas(n,c):
    a=0 # conta o numero de coroas saidas
    for i in range(n):
        if randint(1,2)==2:
            a=a+1
    return a==c
```

b) Escreva a função

```
probab(n,c,exp)
```

que chama exp vezes "coroas(n,c)" e retorna a probabilidade (aproximada) de saírem c ou mais coroas

```
(aproximadamente)
= (Num. de vezes em coroas(n,c)==True) / exp
```

-->

```
def probab(n,c,exp):
    s=0
    for i in range(exp):
        if coroas(n,c):
            s=s+1
```

```
return float(s)/exp
```

c) Calcule a probabilidade exacta do caso $n=5$, $c=4$ e compare com `probab(5,4,10000)`

--> cada sequência tem probabilidade $1/2^5 = 1/32$,
Há 5 hipóteses da cara (4 da coroa) -> Resposta: $5/32$

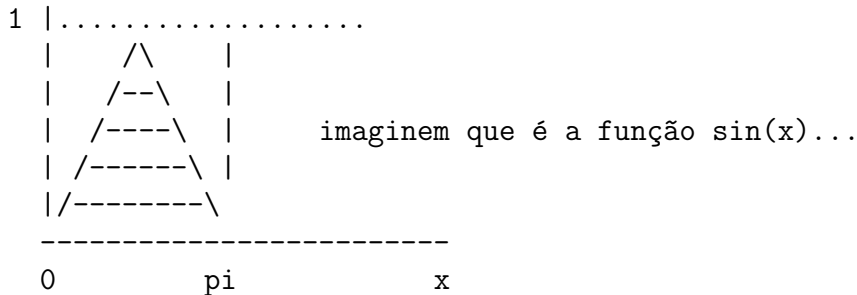
d) Escreva a função

```
distr(n,exp)
```

que retorna numa lista as probabilidades aproximadas calculadas como na alínea b) para $c=0, 1, 2, \dots, n$

```
-->
def distr(n,exp):
    r=[]
    for c in range(n+1):
        r=r+[probab(n,c,exp)]
    return r
```

Pretende-se calcular a area aproximada de uma região...



At = area a tracejado (entre $\sin(x)$ e o eixo xx)

A = area rectangulo = $1 \cdot \pi = \pi$

Gerando um ponto aleatório no rectângulo:

```
x = ...
y = ...
```

n = num. pontos total

s = num de pontos que satisfaz $y < \sin(x)$

At s

-- = (aprox.) -
A t

a) Escreva uma função

area(n)

que calcula pelo, processo indicado, a area a
tracejado...

-->

```
from math import *
def area(n):
    """ só para sin entre 0 e pi """
    s=0
    for i in range(n):
        x=uniform(0,pi)
        y=uniform(0,1)
        if y< sin(x):
            s=s+1
    return pi*float(s)/n
```

b) Generalize! Escreva

areaf(n,f,a,b,max)

onde - $f(x) > 0$ para $a \leq x \leq b$
- max é um majorante de $f(x)$ para $a \leq x \leq b$

'area(n)' (alínea anterior) é equivalente a
'areaf(n,sin,0,pi,1)'