Relatório Tarefa Probabilidade

José Guilherme De Oliveira Antunes

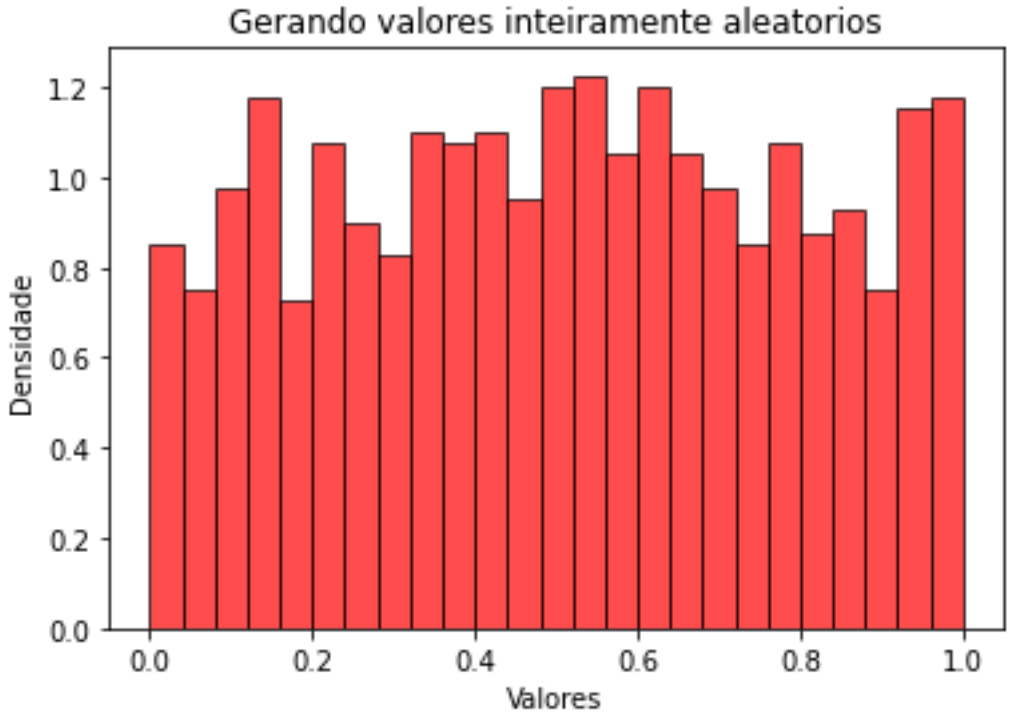
June 7, 2024

Parte 1: Gerando uma distribuição uniforme

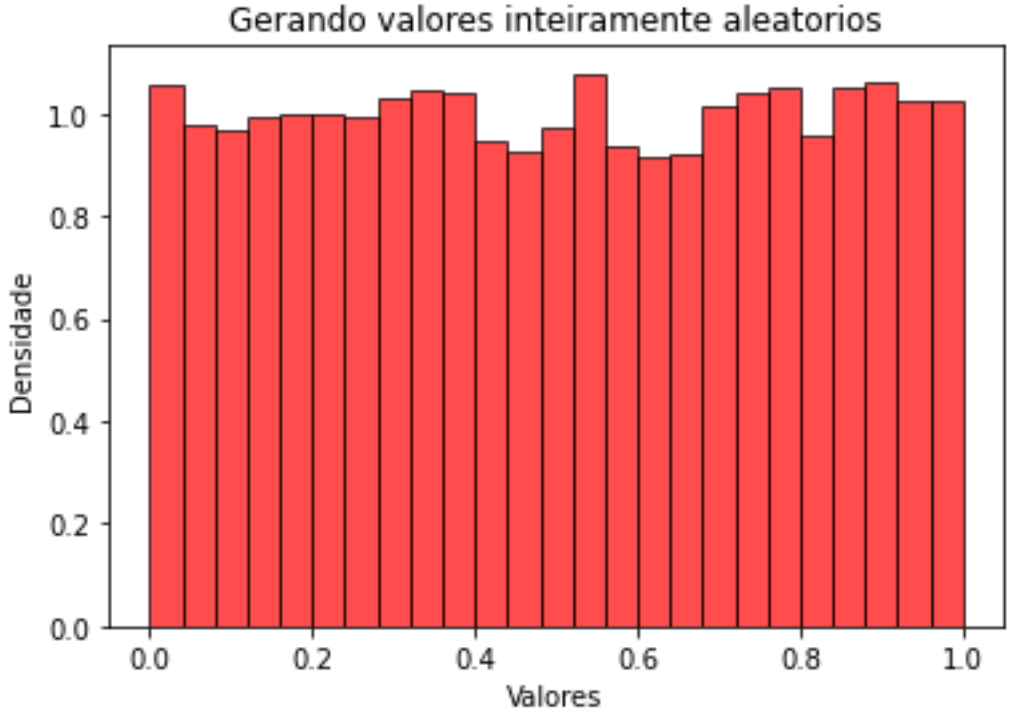
A soma infinita $\sum_{n=1}^{\infty} 2^{-i} X_i$ consiste basicamente em uma forma de representar números entre 0 e 1 utilizando a base binária, como consta em alguns exemplos abaixo:
$0.5 = 1 \cdot 2^{-1} = 0.1$
$0.75 = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 0.11$
$0.625 = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$
Embora os números acima sejam facilmente escritos em base binária, pois são somas finitas de potências de inversas de 2, pode-se escrever qualquer número entre 0 e 1 dessa forma como uma soma infinita de potências inversas de 2. Abaixo exemplos:
$0.3 = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 0 \cdot 2^{-4} + 1 \cdot 2^{-5} + ... = 0.01001...$
$0.763 = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 0 \cdot 2^{-4} + 0 \cdot 2^{-5} + ... = 0.11000...$
$0.347 = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} + ... = 0.01011...$
Desse modo, pode-se ver que, sendo $X_i \sim \text{Bern}\left(\frac{1}{2}\right)$ permite a geração aleatória de números entre 0 e 1. Logo, é uma outra forma de representar $U \sim \text{Unif}(0, 1)$. Para tal, foi feita uma função em python, visível abaixo, que gera um vetor de números aleatórios usando esse método.

```
def vetor_aleatorio(tamanho):
    pot_inv_2 = np.array([1 / (2**i) for i in range(1, 53)])
    vetor_aleatorio = []
    for i in range(1, tamanho+1):
        bern_vector = spt.bernoulli.rvs(p, size = 52)
        num_aleatorio = np.dot(bern_vector, pot_inv_2)
        vetor_aleatorio.append(num_aleatorio)
    vetor_aleatorio = np.array(vetor_aleatorio)
    return vetor_aleatorio
```

Considerando que se gera 1000 valores aleatórios, chega-se em:



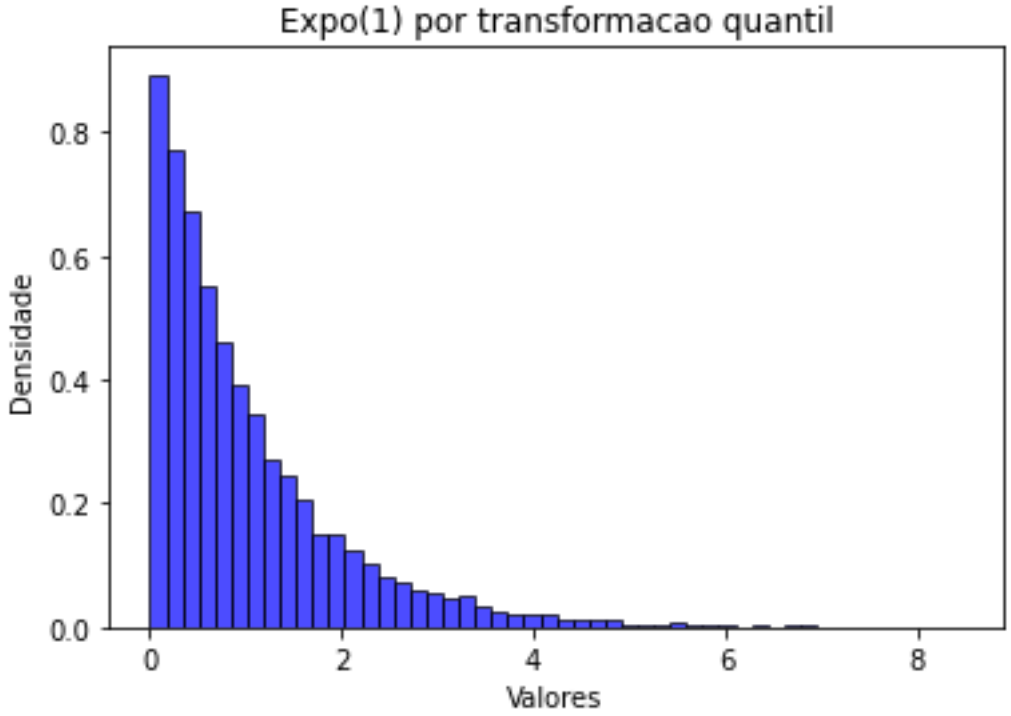
Que ainda está meio disperso, mas gerando 10000 valores, tem-se:



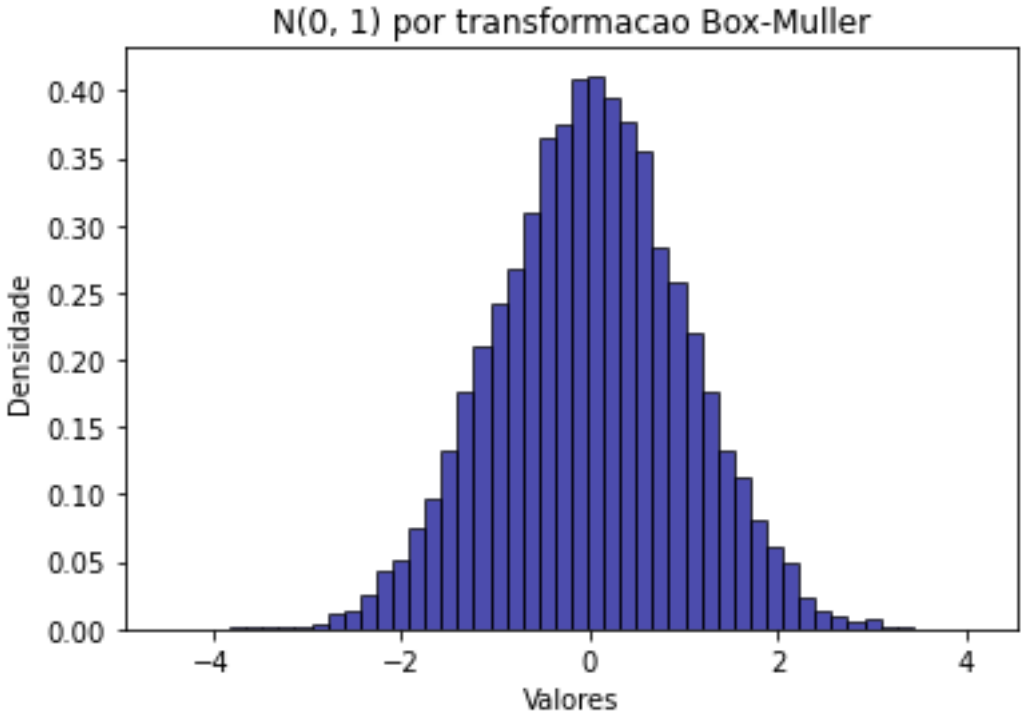
Que de fato parece ser próximo de uma $U \sim \text{Unif}\,(0,1)$

Parte 2: Gerando outras distribuições

A transformação quantil gera, através de uma CDF invertível $F(x)$ e de uma uniforme $U \sim \text{Unif}(0,1)$, uma variável aleatória $X = F^{-1}(U)$ com distribuição $F(x)$. No caso de uma exponencial com $\lambda = 1$, temos que sua CDF é $F(x) = 1 - e^{-x}$. Logo, manipulando algebricamente para encontrar sua inversa, chega-se em $F^{-1}(x) = \ln(\frac{1}{1-x})$. Gerando um vetor aleatório com 10000 uniformes e o utilizando na função inversa, chega-se no seguite histograma:



Que de fato se parece muitíssimo com a pdf de uma distribuição exponencial. Já a transformação de Box-Muller se utiliza, além de uma distribuição uniforme $U \sim \text{Unif}(0,1)$, de uma distribuição exponencial T $\sim$ Expo(1). Assim, podemos gerar uma variável aleatória X com a equação $X = \sqrt{2T} \cdot \cos U$, onde $X \sim \mathcal{N}(0,1)$. Dessa maneira, temos que o histograma da variável criada X, considerando um vetor aleatório com 10000 uniformes e outro com 10000 exponenciais, é:



Que tem uma distribuição bem próxima de uma gaussiana, conforme esperado.