

Métodos Computacionais da Física A

Mateus Cruz Rossetto - Matrícula: 00314722
IF-UFRGS

21 de setembro de 2021

Resumo

Estudaremos o lançamento de dois dados "N" vezes e "observando" as soma das ocorrências, repetições de ocorrência, para obtermos a média e desvio padrão destes, a plotagem de um histograma, com alguns destes resultados, como também verificar a tendência deles. Iremos utilizar a linguagem de programação python para observarmos um valor de "N = 1000", onde, relativamente, seria trabalhoso manualmente.

1 Introdução

Utilizaremos a linguagem Python para esse projeto, para que poupemos o trabalho manual, usaremos em Python recursos como funções, arrays, operações lógicas, matemáticas, loops entre outros para execução deste projeto, assim como algumas bibliotecas, que nos permitiram acesso a recursos previamente não instalados dependendo da sua IDE, neste caso em particular estaremos utilizando o "Pycharm", onde é necessário a inclusão dos mesmos.

2 Método

Primeiramente, iremos importar três bibliotecas específicas para este projeto (numpy, random e matplotlib.pyplot), onde numpy nos fornecera a manipulação de arrays, random para gerar números aleatórios (neste caso os números voltados para cima ao lançar os dados), e matplotlib.pyplot para plotar um histograma. Devemos fazer a inclusão da seguinte forma:

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

Desta maneira o "as" é para renomear a biblioteca, Assim, quando "chamarmos" uma função desta bibliotecas escrevermos apenas o apelido. Utilizaremos para os cálculos as equações da média (1) e desvio (2), onde são as seguintes:

$$\langle Y \rangle = \frac{1}{N} \sum y_i \quad (1)$$

$$Y = \sqrt{\frac{1}{N} \sum (y_i - \langle Y \rangle)^2} \quad (2)$$

Agora usaremos três funções para gerar a combinação de números de dois dados, o cálculo da média e o cálculo do desvio, respectivamente.

```
. . .
def lançamentosDados(n):
    x1 = np.array([rand.randint(1, 6) for i in range(n)])
    # gerando numeros aleatórios em x1 n vezes.
    x2 = np.array([rand.randint(1, 6) for i in range(n)])
    # gerando numeros aleatórios em x2 n vezes.
    y = x1 + x2 # possível apenas com array's.
    return y

def média(y):
    soma = float(0)
    for i in range(len(y)):
        soma += y[i]
    return float(soma/len(y))

def desvio(m, y):
    soma = 0
    for i in range(len(y)):
        soma += float((y[i]-m)**2)
    d = (soma/len(y))*0.5
    return d
. . .
```

Em Python "#" indicam ao programa algo que não deve ser executado (um comentário, que no caso foi usado para explicar o que é o "x1" e "x2"). Note que aqui usamos a biblioteca random (renomeada rand). Agora já temos nossas funções prontas, então podemos fazer nossa linha de execução, definiremos primeiro nosso "N = 1000", depois "chamaremos" as funções com os

parâmetros necessários e colocaremos os retornos ("return" na linha de código anterior) em variáveis, para que possamos manipular os resultados.

```
. . .
N=1000
Y = lançamentosDados(N) #lista com os valores das somas dos dados.
média = média(Y)
desvio = desvio(média, Y)
print(f"Minha média = {média}, meu desvio = {desvio}")
. . .
```

Podemos então, fazer a contagem de quantas vezes aparece cada número de dois a doze na soma dos dados e colocá-los em uma lista "cont", onde o número de cada "Y" tem seu próprio índice em "cont", como no código abaixo:

```
. . .
    cont = (np.zeros(11)) #gerando uma lista com seis números iguais a zero
for i in Y:
    cont[i-2] += 1
. . .
```

Tendo todos os dados, podemos fazer um histograma de quantas vezes aparece cada número, criando uma lista "numeros" para formar o eixo "X", assim montando o histograma (numeros X cont):

```
. . .
numeros = np.arange(2,13) #colocando os valores que irão no eixo x
no gráfico do histograma
plt.bar(numeros, cont)
plt.title("Lançamento De Dois Dados")
plt.xlabel("Combinação numérica")
plt.ylabel("Repetições")
plt.show()
. . .
```

Podemos analisar que, as equações da média e desvio demonstradas e montamos em python, existem dentro do python("np.mean()" e "np.std()"), calculando diretamente tais resultados, podemos usá-las da seguinte forma:

```
. . .
    print(f"Média da biblioteca numpy = {np.mean(Y)}, desvio da
    biblioteca numpy = {np.std(Y)}")
. . .
```

3 Resultados

```
Minha média = 7.129, meu desvio = 2.443841034110031
```

Figura 1: Print - média e desvio

```
Média da biblioteca numpy = 7.129, desvio da biblioteca numpy = 2.443841034110034
```

Figura 2: print - média e desvio (biblioteca numpy)

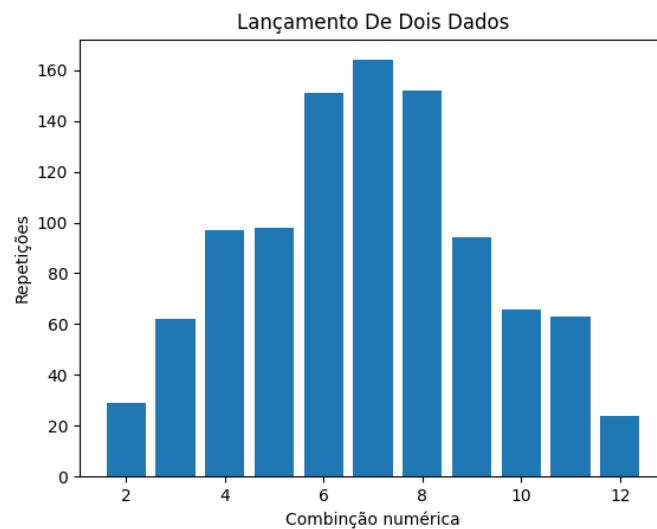


Figura 3: Histograma (numeros X cont)

Notamos que o histograma se assemelha a uma distribuição normal por volta do número sete (onde podemos notar pela média complementarmente). Isso se dá pelo fato de que o número sete tem mais combinações possíveis que os demais (mais casos favoráveis), além de que quanto mais afastamos

do número sete, para as extremidades, diminuimos o número de combinações possíveis que formam tal número. Chegando em dois e doze, nota-se que cada um tem apenas uma combinação possível (dois = um+um, doze = seis+seis). Quando tendemos o valor de "N" à infinito o resultado tende a sete e o histograma será cada vez mais semelhante a distribuição normal.

4 Conclusões

Podemos ver que usando uma linguagem de programação (Python, por exemplo) para estudar este evento é mais eficiente, pois podemos usar um número "N" qualquer, sem todo o procedimento mecânico que poderia levar até dias para um número suficientemente grande. Podemos executar o programa diversas vezes com números distintos em minutos. Assim, podendo analisar a modificação da curva do histograma para diferentes "N's". Conseguimos analisar também que para números com casas decimais suficientemente grandes, podemos ter diferenças em processos "caseiros" (como fizemos com a média e desvio) dentro do programa em relação a uma função já existente em uma biblioteca, mas que são números com bastante casas decimais iguais relativamente. Com isso, dependendo da aplicação poderiam ser considerado arredondamentos e retornariam números iguais (arredondamento do desvio para duas casas decimais, por exemplo, desvio = 2,44).