

# Exercicio2-Solucao

May 20, 2024

## 1 Data Science Academy

### 1.1 Matemática e Estatística Aplicada Para Data Science, Machine Learning e IA

### 1.2 Lista de Exercícios 2

### 1.3 Resolvendo Equações Quadráticas, Funções e Logaritmos com Python

#### 1.3.1 Instalando e Carregando Pacotes

```
[ ]: # Para atualizar um pacote, execute o comando abaixo no terminal ou prompt de
    ↳ comando:
    # pip install -U nome_pacote

    # Para instalar a versão exata de um pacote, execute o comando abaixo no
    ↳ terminal ou prompt de comando:
    # !pip install nome_pacote==versão_desejada

    # Depois de instalar ou atualizar o pacote, reinicie o jupyter notebook.

    # Instala o pacote watermark.
    # Esse pacote é usado para gravar as versões de outros pacotes usados neste
    ↳ jupyter notebook.
    !pip install -q -U watermark
```

```
[ ]: # Imports
import matplotlib.pyplot as plt
import numpy as np
```

```
[ ]: # Versões dos pacotes usados neste jupyter notebook
%reload_ext watermark
%watermark -a "Data Science Academy"
```

Author: Data Science Academy

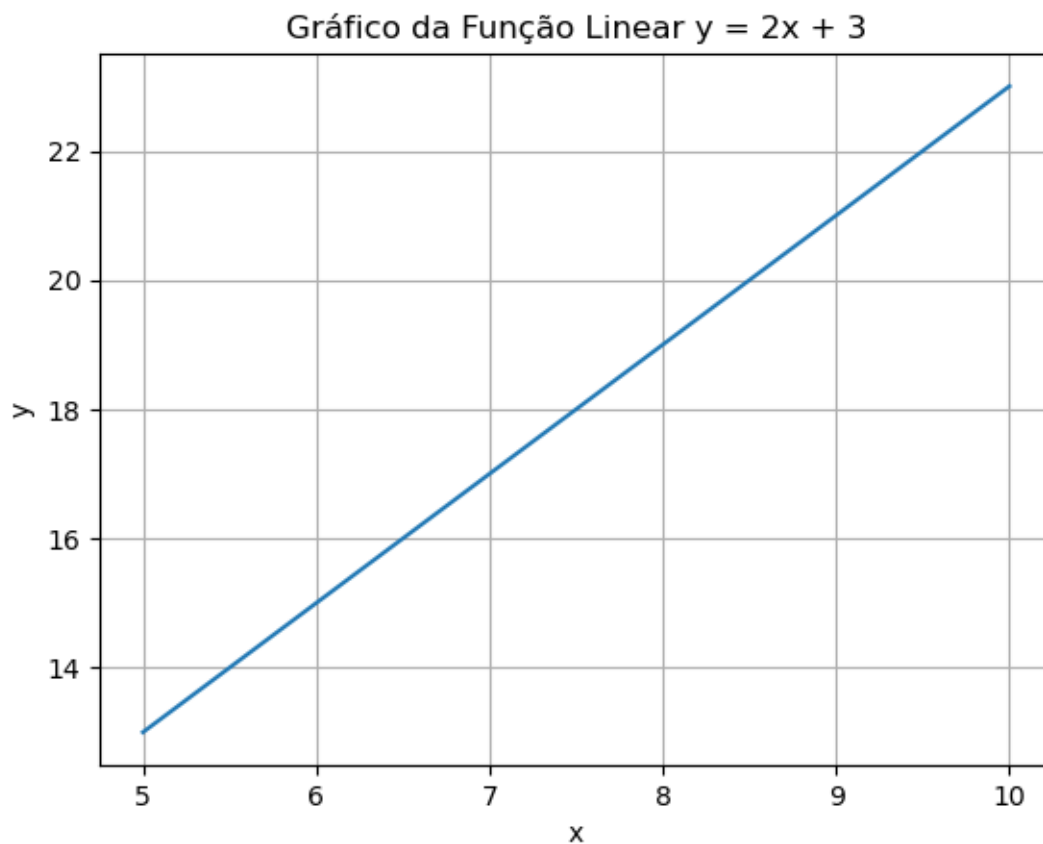
Resolva os exercícios abaixo usando Linguagem Python. Faça pesquisa complementar se necessário.

### 1.3.2 Exercício 1: Gráfico de Uma Função Linear

Crie um gráfico para a função  $y = 2x + 3$  para valores de  $x$  no intervalo de 5 a 10.

```
[ ]: # Gera os dados
x = np.linspace(5, 10, 100)
y = 2 * x + 3

# Plot
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Gráfico da Função Linear y = 2x + 3')
plt.grid(True)
plt.show()
```

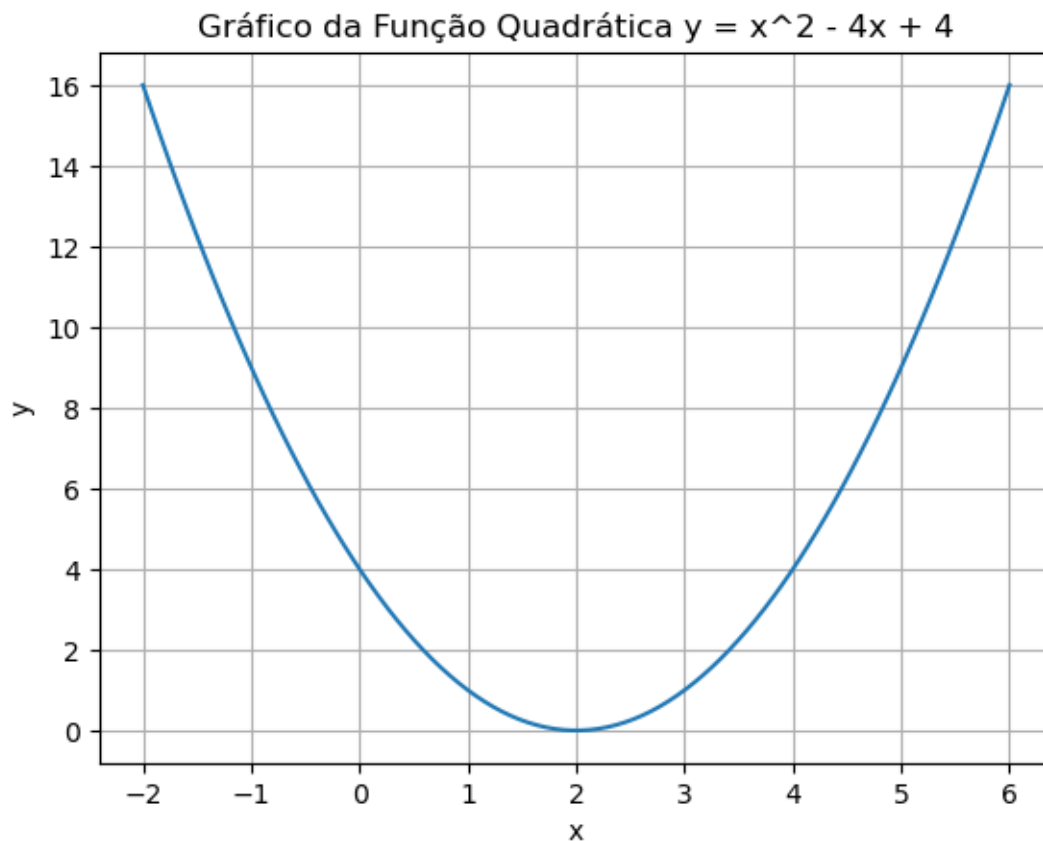


### 1.3.3 Exercício 2: Gráfico de uma Função Quadrática

Desenhe o gráfico da função  $y = x^2 - 4x + 4$ . Crie uma massa de dados de exemplo usando `np.linspace()`.

```
[ ]: # Gera os dados
x = np.linspace(-2, 6, 100)
y = x**2 - 4*x + 4

# Plot
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Gráfico da Função Quadrática y = x^2 - 4x + 4')
plt.grid(True)
plt.show()
```

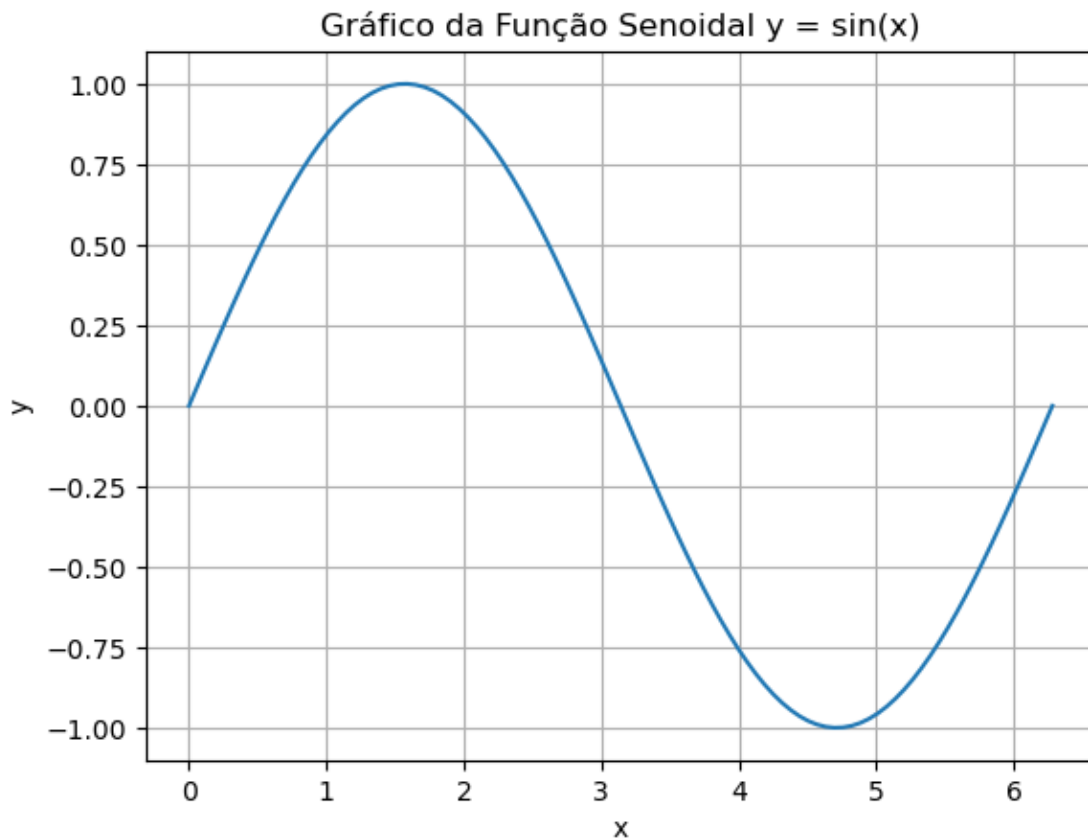


### 1.3.4 Exercício 3: Gráfico de Uma Função Senoidal

Crie um gráfico da função  $y = \sin(x)$  para valores de  $x$  no intervalo de 0 a  $2\pi$  (3.14).

```
[ ]: # Gera os dados
x = np.linspace(0, 2 * np.pi, 100)
y = np.sin(x)
```

```
# Plot
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Gráfico da Função Senoidal y = sin(x)')
plt.grid(True)
plt.show()
```



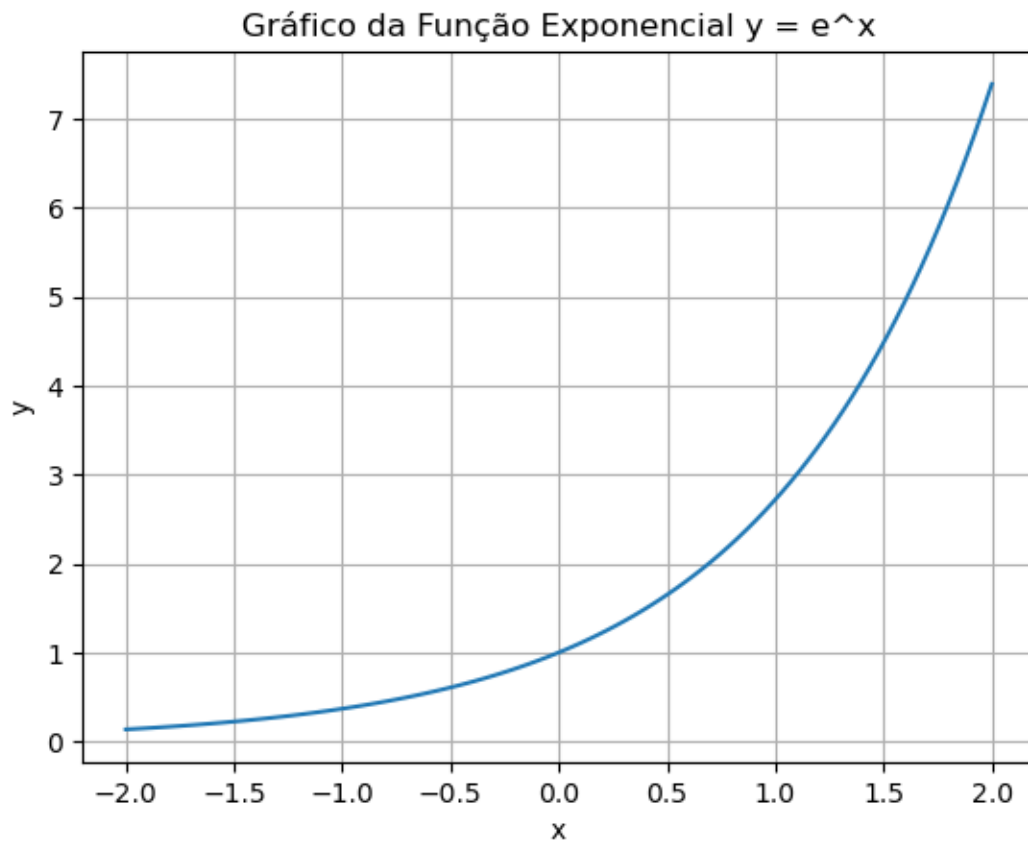
### 1.3.5 Exercício 4: Gráfico de Uma Função Exponencial

Crie um gráfico para a função  $y = e^x$  para valores de  $x$  no intervalo de -2 a 2.

```
[ ]: # Gera os dados
x = np.linspace(-2, 2, 100)
y = np.exp(x)

# Plot
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
```

```
plt.title('Gráfico da Função Exponencial y = e^x')
plt.grid(True)
plt.show()
```

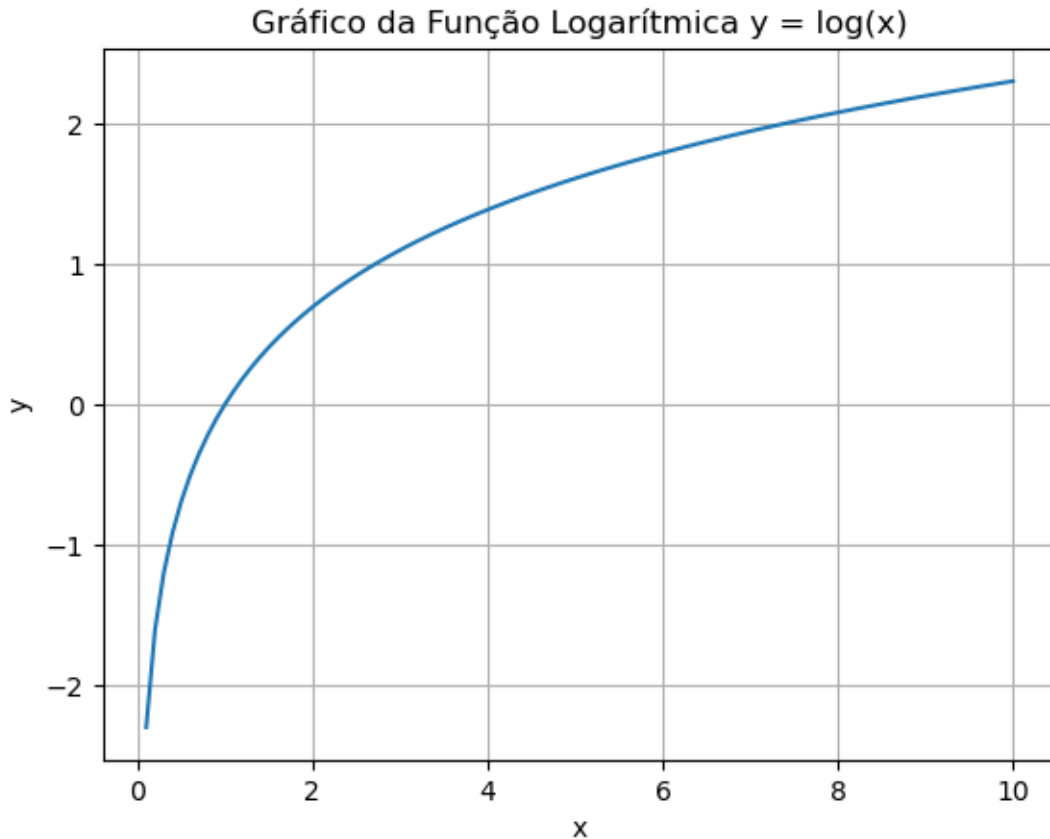


### 1.3.6 Exercício 5: Gráfico de uma Função Logarítmica:

Desenhe o gráfico de  $y = \log(x)$  (logaritmo natural) para valores de x no intervalo de 0.1 a 10.

```
[ ]: # Gera os dados
x = np.linspace(0.1, 10, 100)
y = np.log(x)

# Plot
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Gráfico da Função Logarítmica y = log(x)')
plt.grid(True)
plt.show()
```



### 1.3.7 Exercício 6: Resolver uma Equação Quadrática

Escreva um script em Python que resolve a equação  $x^2 - 5x + 6 = 0$  e imprime as raízes.

```
[ ]: import cmath

a, b, c = 1, -5, 6

d = (b**2) - (4*a*c)
sol1 = (-b-cmath.sqrt(d))/(2*a)
sol2 = (-b+cmath.sqrt(d))/(2*a)

print('As raízes são {0} e {1}'.format(sol1, sol2))
```

As raízes são (2+0j) e (3+0j)

### 1.3.8 Exercício 7: Equação Quadrática com Entrada do Usuário

Modifique o programa anterior para aceitar coeficientes a, b, e c de um usuário e resolver a equação quadrática correspondente.

```
[ ]: a = float(input("Digite o coeficiente a: "))
      b = float(input("Digite o coeficiente b: "))
      c = float(input("Digite o coeficiente c: "))

      d = (b**2) - (4*a*c)

      sol1 = (-b-cmath.sqrt(d))/(2*a)
      sol2 = (-b+cmath.sqrt(d))/(2*a)

      print('As raízes são {0} e {1}'.format(sol1,sol2))
```

```
Digite o coeficiente a: 3
Digite o coeficiente b: 5
Digite o coeficiente c: 8
As raízes são (-0.8333333333333334-1.4043582955293932j) e
(-0.8333333333333334+1.4043582955293932j)
```

### 1.3.9 Exercício 8: Equação Quadrática com Discriminante

Para a equação quadrática  $ax^2 + bx + c = 0$ , calcule o discriminante e determine o número de raízes reais.

O discriminante em uma equação quadrática é uma parte específica da fórmula que é usada para determinar o número e o tipo de soluções (raízes) que a equação possui.

Se o discriminante é positivo: Há duas soluções reais e distintas para a equação. Isto é, a parábola associada à equação quadrática corta o eixo-x em dois pontos distintos.

Se o discriminante é zero: Há exatamente uma solução real para a equação (também conhecida como uma raiz dupla). Neste caso, a parábola toca o eixo-x em apenas um ponto.

Se o discriminante é negativo: Não há soluções reais para a equação. Isso significa que a parábola não cruza o eixo-x, e as soluções são números complexos.

O discriminante é uma ferramenta útil na análise de equações quadráticas, pois fornece uma maneira rápida e direta de entender a natureza das raízes sem necessariamente calcular essas raízes.

```
[ ]: a = float(input("Digite o coeficiente a: "))
      b = float(input("Digite o coeficiente b: "))
      c = float(input("Digite o coeficiente c: "))

      d = (b**2) - (4*a*c)

      if d > 0:
          print("Duas raízes reais e distintas.")
      elif d == 0:
          print("Uma raiz real dupla.")
      else:
          print("Nenhuma raiz real.")
```

Digite o coeficiente a: 3  
Digite o coeficiente b: 5  
Digite o coeficiente c: 2  
Duas raízes reais e distintas.

### 1.3.10 Exercício 9: Cálculo de Logaritmos

Escreva um script que calcula o logaritmo de base 10 de um número fornecido pelo usuário. Certifique-se de tratar o caso em que o logaritmo não pode ser definido (números negativos e zero).

```
[ ]: import math

num = float(input("Digite um número: "))

if num > 0:
    log_value = math.log10(num)
    print("Logaritmo de base 10 de {0} é {1}".format(num, log_value))
else:
    print("Logaritmo não definido para números negativos ou zero.")
```

Digite um número: 5  
Logaritmo de base 10 de 5.0 é 0.6989700043360189

### 1.3.11 Exercício 10: Aplicação de Logaritmos

Utilize logaritmo para calcular o tempo necessário para que um investimento atinja um determinado valor, considerando uma taxa de crescimento contínua.

```
[ ]: # Calculando o tempo necessário para dobrar um investimento com juros compostos

valor_inicial = float(input("Digite o valor inicial do investimento: "))

taxa_juros = float(input("Digite a taxa de juros anual (em decimal): "))

tempo = math.log(2) / math.log(1 + taxa_juros)

print(f"O tempo aproximado para dobrar o investimento é {tempo:.2f} anos.")
```

Digite o valor inicial do investimento: 4000  
Digite a taxa de juros anual (em decimal): 0.3  
O tempo aproximado para dobrar o investimento é 2.64 anos.

```
[ ]: %reload_ext watermark
%watermark -a "Data Science Academy"
```

Author: Data Science Academy

```
[ ]: #%watermark -v -m
```



```
[ ]: ##watermark --inversions
```

## 2 Fim