

PYTHON



1 – Linguagem python aplicada Data Science

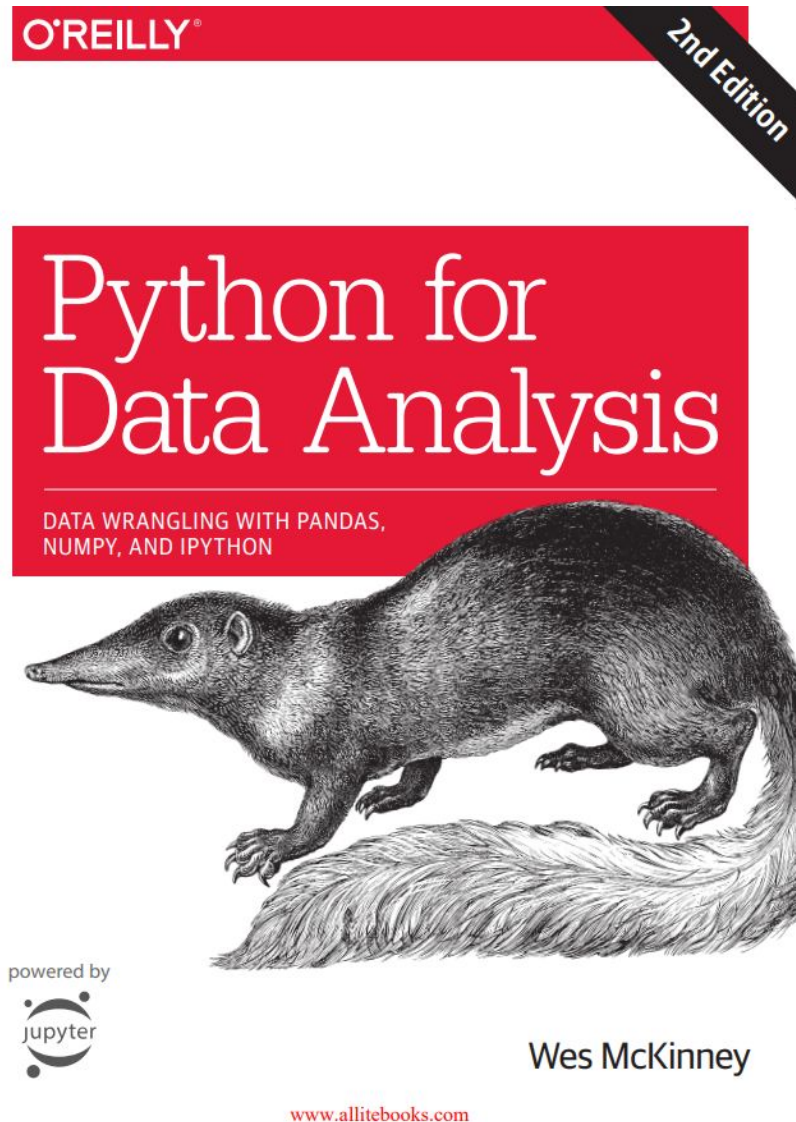
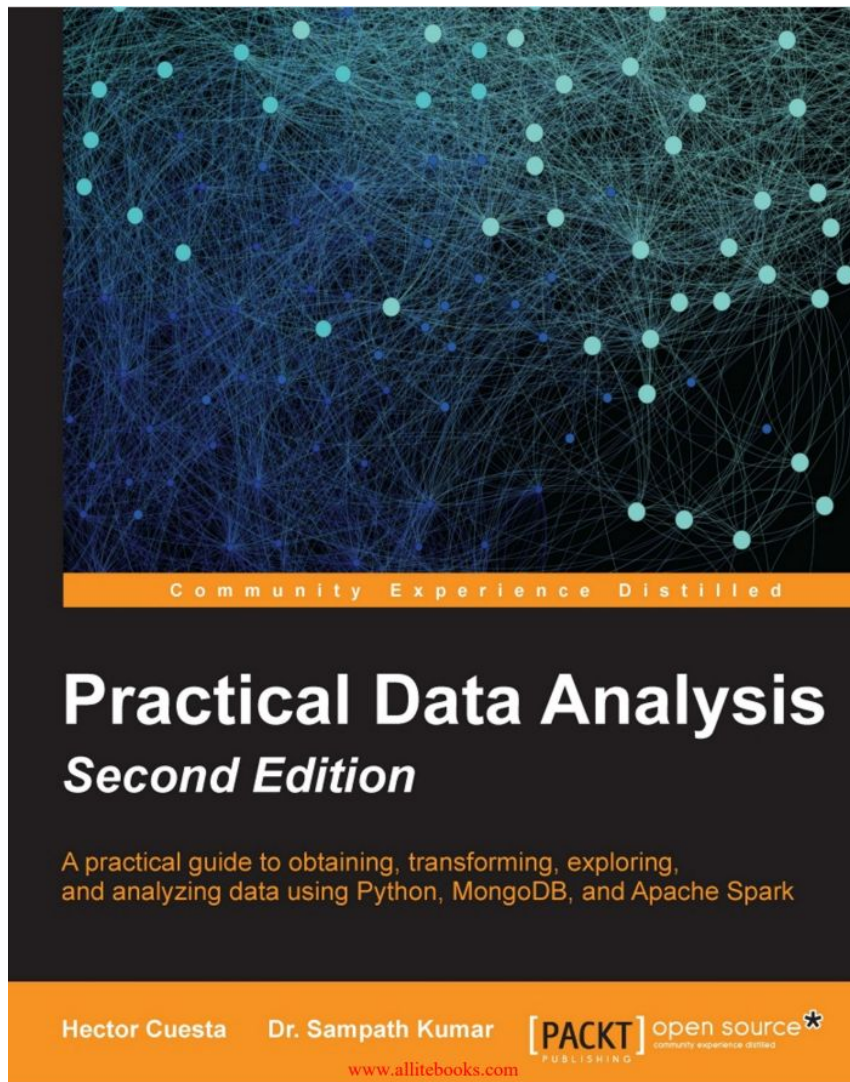
- Introdução na Linguagem Python
- Jupyter: Interface e Notebooks: Colab google
- Operações matemáticas
- Variáveis
- Programando com Python
- Importação e limpeza de dados em Python
- Databases em Python
- Arrays e Matrizes com NumPy
- Estruturas e Operações com Pandas
- Manipulação de dados com NumPy

2 - Visualização de Dados

- Matplotlib
- Seaborn

3 - Exploração e Análise de Dados

- Distribuições
- Histogramas
- Box-plots, quartiles, scatter plots, heat maps and others
- Aplicando regressão linear e estatística



Tendência de Tíquetes

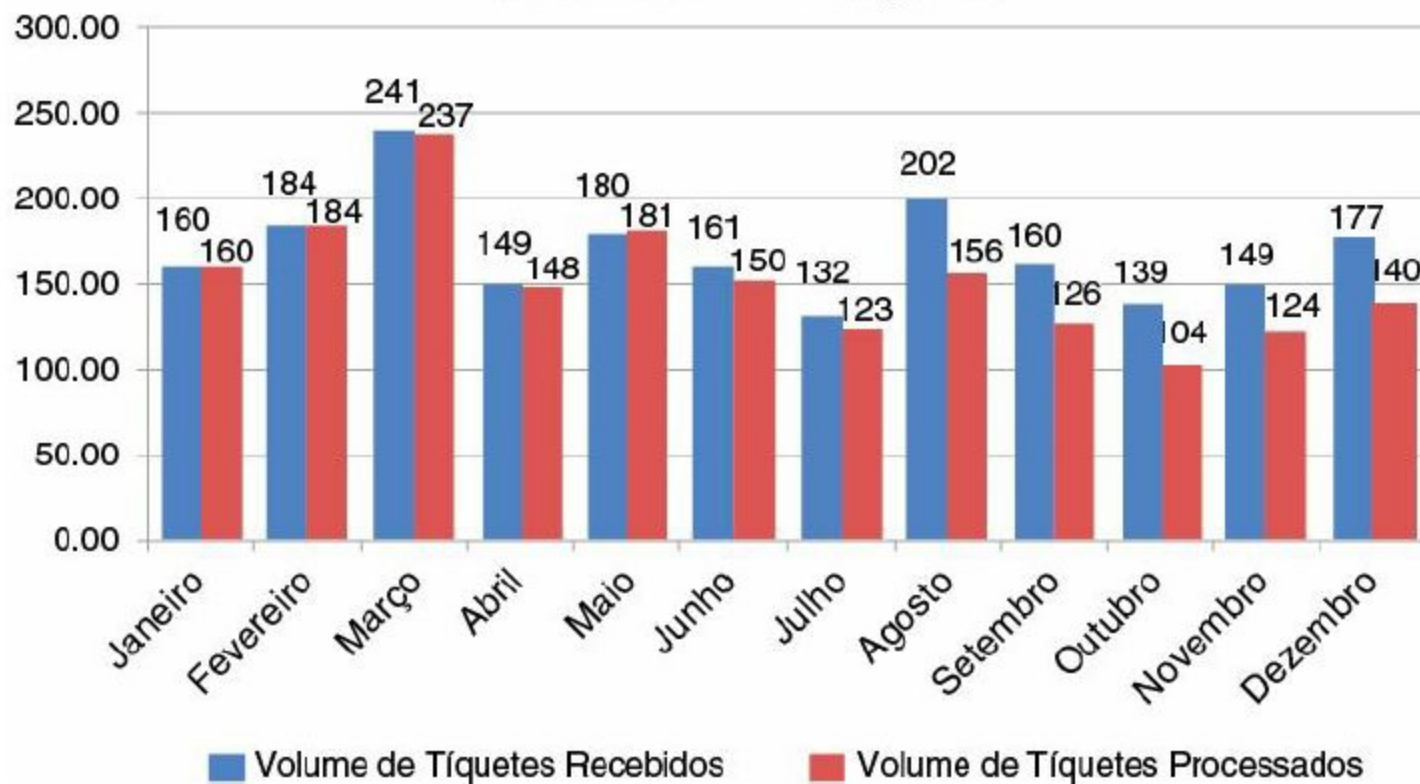
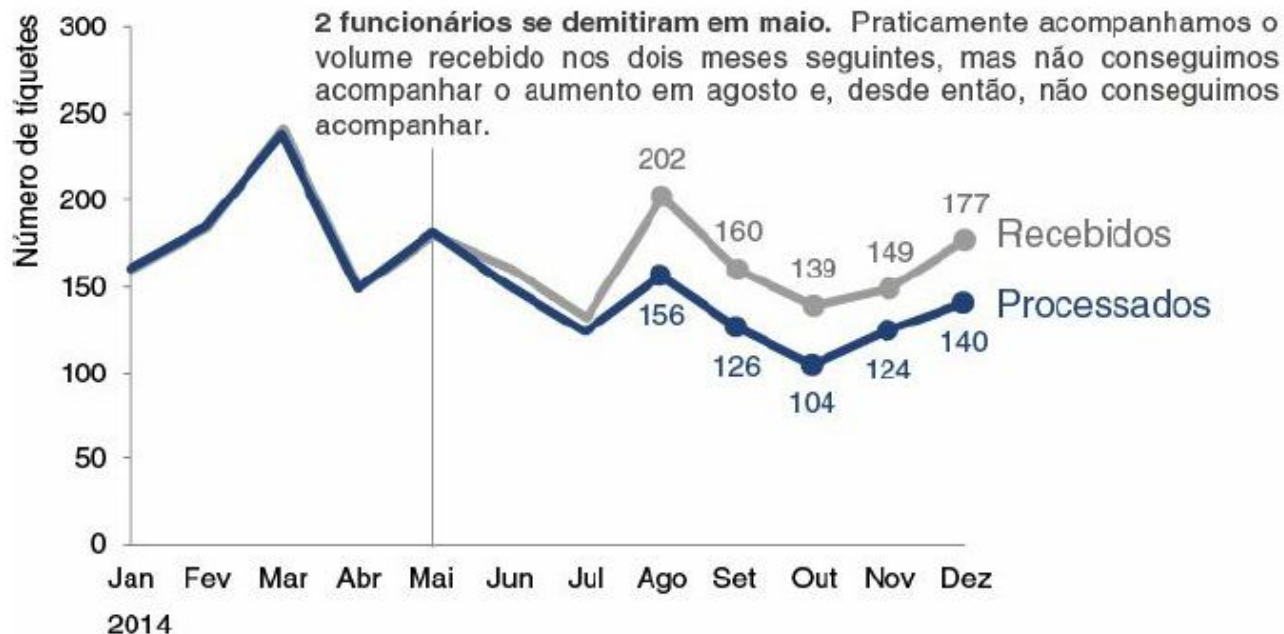


FIGURA 0.2 Exemplo 1 (antes): apresentação de dados

Favor aprovar a contratação de 2 ETIs

para repor aqueles que se demitiram no ano passado

Volume de tíquetes ao longo do tempo



Fonte dos dados: XYZ Dashboard, em 31/12/2014 | Foi feita uma análise detalhada dos tíquetes processados por pessoa e meses para resolver os problemas para informar este pedido. Se necessário, pode ser fornecida.

FIGURA 0.3 Exemplo 1 (depois): storytelling com dados

Resultados da Avaliação

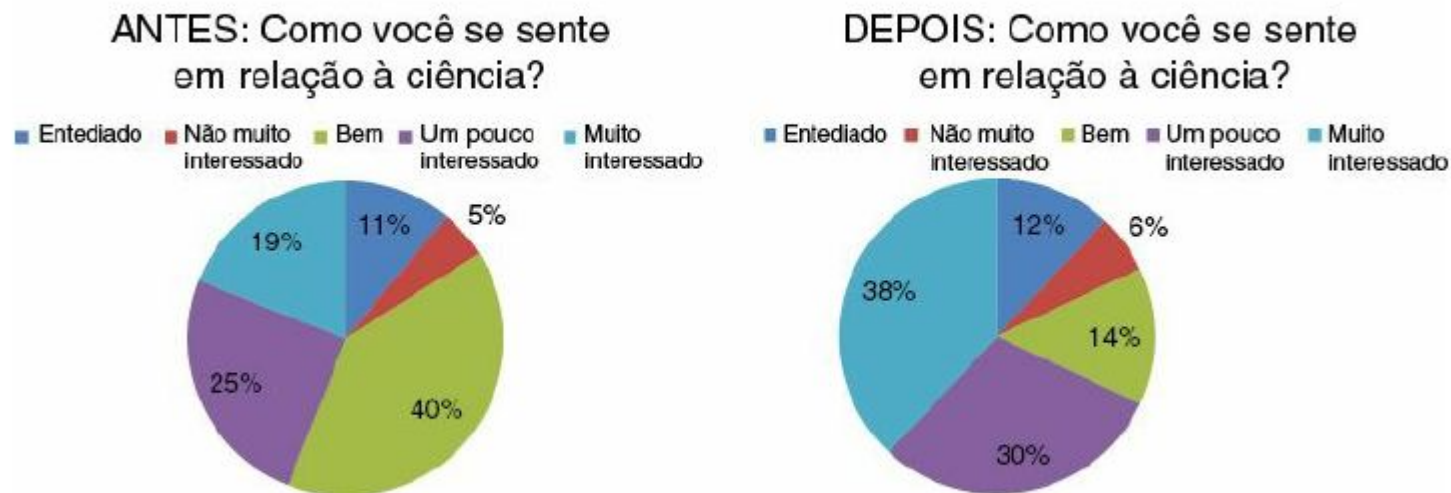


FIGURA 0.4 Exemplo 2 (antes): apresentação de dados

O programa-piloto foi um sucesso

Como você se sente em relação à ciência?

ANTES do programa,
a maioria das crianças
se sentia **BEM** em
relação à ciência.



DEPOIS
do programa,
mais crianças tinham
algum interesse e
muito interesse
em ciências.

Baseado na avaliação de 100 alunos, realizada antes e depois do programa-piloto (taxa de respostas de 100% nas duas avaliações)

FIGURA 0.5 Exemplo 2 (depois): storytelling com dados

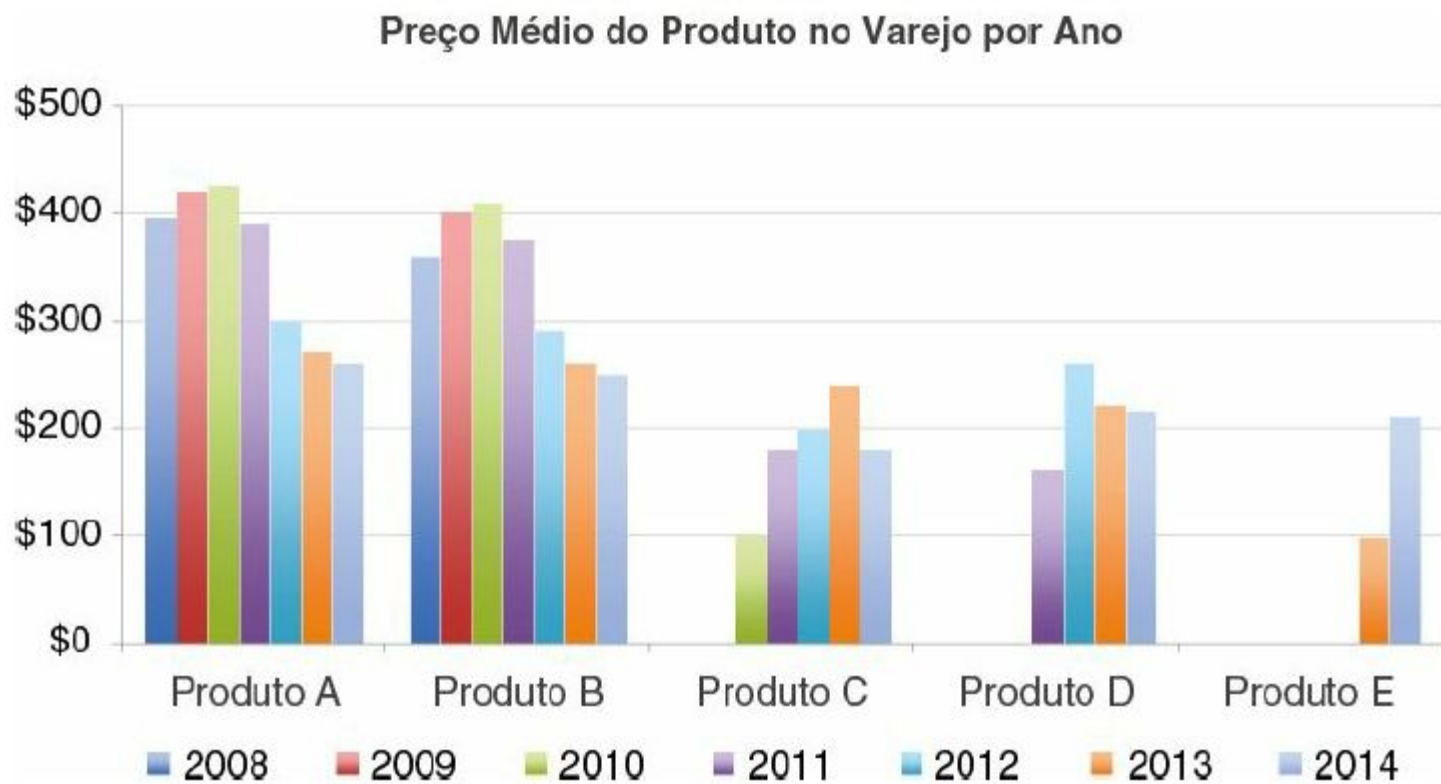


FIGURA 0.6 Exemplo 3 (antes): apresentação de dados

Para sermos competitivos, recomendamos introduzir nosso produto abaixo do preço médio de \$223, **na faixa de \$150 a \$200**

Preço médio ao longo do tempo por produto

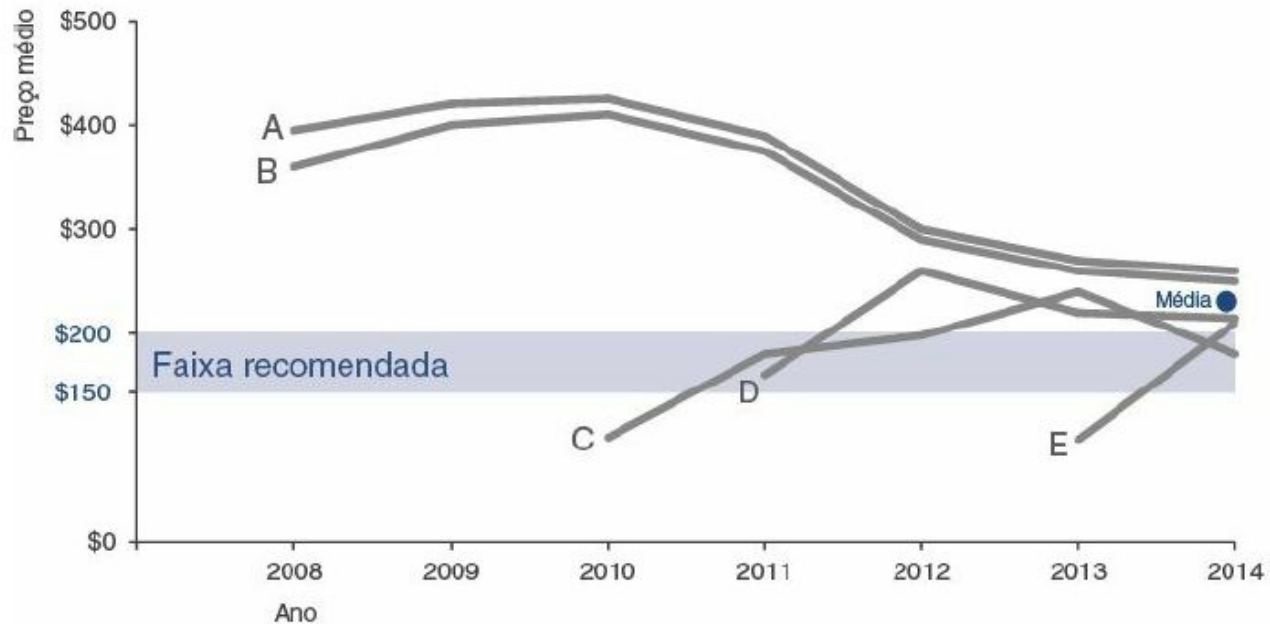


FIGURA 0.7 Exemplo 3 (depois): storytelling com dados

Linha de base diferente de zero: conforme originalmente representada

SE OS CORTES DE IMPOSTOS DE BUSH EXPIRAREM
MAIOR ALÍQUOTA DE IMPOSTO



Linha de base zero: conforme deveria ser representada

SE OS CORTES DE IMPOSTOS DE BUSH EXPIRAREM
MAIOR ALÍQUOTA DE IMPOSTO



FIGURA 2.13 OS GRÁFICOS DE BARRA DEVEM TER UMA LINHA DE BASE ZERO



🔒 anaconda.com



Products ▼

Pricing

Solutions ▼

Resources ▼

Blog

Company ▼

Get Started

Data science technology for human sensemaking.

A movement that brings together millions of data science practitioners,
data-driven enterprises, and the open source community.

<https://www.anaconda.com/>



colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Machine Learning Examples
- Section

+ Code + Text Copy to Drive

Connect Editing

What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
```

<https://colab.research.google.com/notebooks>



Untitled6.ipynb ☆



Comentário



Compartilhar



Arquivo Editar Ver Inserir Ambiente de execução F



+ Código + Texto

Conectar ▼



Editar



{x}



+ Código

+ Texto



|

Adicionar célula de texto

Linguagem Python





Google Hello_world.ipynb ☆

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas



Comentário



Compartilhar



+ Código + Texto



RAM
Disco



Editar



Hello, world!



```
print("Hello, World!")
```

✓
0s

Hello, World!

Operações:

- ✓ [2] # Adição:
4+2
- ✓ [3] # Subtração:
5-2
- ✓ [4] # Multiplicação:
3*2
- ✓ [5] # Divisão:
8/2
- ✓ [6] # Potência:
2**3
- ✓ [7] # Raiz quadrada:
3**(1/2)
- ✓ [8] # Raiz:
27**(1/3)
- ✓ [9] # uso da função de potência `pow(a,b)=a**b`
`pow(4,0.5)`

Tipos de variáveis numéricas e funções pow(), round() e abs():



✓ [10] # tipo inteira: int()
int(5)

✓ [11] # tipo decimal: float()
float(3.8)

✓ [12] # tipo decimal: float()
float(3) #Podemos converter o número de inteiro para decimal

✓ [13] # exemplo int
int(4.5)

✓ [14] # função: Arredondar Números Decimais Em Python
round(3.6)

✓ [15] # exemplo float
float(3.5)

✓ [16] # A função abs() retorna o valor absoluto
abs(-6)

As variáveis em Python são utilizadas para armazenar valores.

```
✓ [1] # Atribuindo um valor à variável var  
    var = 3
```

```
✓ [2] # imprimindo a variável var  
    print(var)
```

```
✓ [6] # Atribuindo um valor à variável var e imprimindo:  
    var = 5  
    print(var)
```

```
✓ [4] # Para realizar múltiplas atribuições,  
  
    var1, var2, var3 = 20, 32, 45  
    print("var 1 = ",var1)  
    print("var 2 = ",var2)  
    print("var 3 = ",var3)
```

```
✓ [5] #Operações com Variáveis:  
    a=2  
    b=5  
    c=a+b  
    print(" c = a+b \n c = ",c)
```

▼ Funções:

✓ [7] # A estrutura para escrever uma função está representada abaixo:

0s

```
def funcao():  
    print("Estou começando a entender Python!")
```

✓ [8] funcao()

✓ [10] # Uma função que soma dois valores

0s

```
def func(a, b):  
    """  
    Add two numbers together  
    Returns the_sum : type of arguments  
    """  
    c = a+b  
    print(c)
```

✓ [11] func(3,2)

✓ [13] func?

0s

Expressões Lambda:

```
[14] # Vamos definir uma expressão com lambda  
      # para elevar números à terceira potência (cubo)  
  
      cubo = lambda x: pow(x, 3)
```

✓
0s [15] cubo(2)

8

✓
0s [16] cubo(5)

125

▼ Listas:

```
[ ] # Criando uma lista
```

```
    lista = [10, 25, 20, 35, 40]  
    print(lista)
```

```
[ ] # Verificando o tipo de variável
```

```
    type(lista)
```

```
[ ] # Podemos acessar valores individuais da lista, utilizando os índices  
    lista[0]
```

```
[ ] # Podemos acessar valores individuais da lista, utilizando os índices  
    lista[0]
```

```
[ ] # Podemos adicionar valores individuais da lista  
    lista.append(56)
```

```
[ ] print(lista)
```

```
[ ] # Podemos deletar valores individuais da lista  
    del lista[4]  
    print(lista)
```

```
[ ] # Podemos, ainda, atualizar os valores das listas
    lista[2]=21
    print(lista)
```

```
[ ] #Podemos concatenar (juntar) duas listas, utilizando o símbolo de +
    lista_1=[1,2,3,4]
    lista_2=[5,1,7]
    lista_3=lista_1+lista_2
    print(lista_3)
```

```
▶ # Para verificar o tamanho da lista utilizamos o método len.
    len(lista_3)
```

```
[ ] #Para encontrar o valor máximo, utilizamos max
    max(lista_3)
```

```
[ ] # mínimo
    min(lista_3)
```

```
[ ] #soma
    sum(lista_3)
```

```
▶ # a frequencia de um elemento
    lista_3.count(1)
```