

Matemática Computacional

Capítulo 1

Introdução ao Python
Introdução à estatística

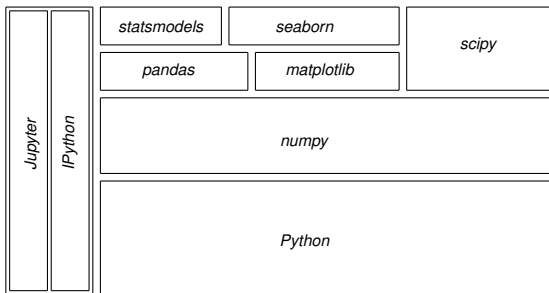
Licenciatura em Engenharia Informática
ISEP
(2024/2025)

1 Introdução ao Python

2 Introdução à estatística

Introdução ao Python

Bibliotecas mais importantes para aplicações de estatística



Para instalar o **Python + Jupyter Notebook**:

- Ir a **anaconda.com**;
- Descarregar e executar;
- Na barra de procura do PC, procurar e executar Jupyter Notebook.

Tipo de variáveis

int float bool str

Tipo e precedência de operadores

Precedência	Símbolo	Significado
1º	()	parênteses
2º	**	potência
3º	* /	multiplicação e divisão
3º	// %	divisão inteira e resto da divisão inteira
4º	+ -	soma(concatena) e subtração

Leitura/Escreita do/no terminal

```
# função input() retorna sempre string  
a = input('Escreva um número:')  
b = input('Escreva um número:')  
print(f'A soma entre o número {a} e o número {b} é {a + b}')
```

input: 5 input: 6

output: A soma entre o número 5 e o número 6 é 56

```
# função input() retorna sempre string  
a = int(input('Escreva um número inteiro:'))  
b = float(input('Escreva um número:'))  
print(f'A soma entre o número {a} e o número {b:.3f} é {a + b:.3f}')
```

input: 5 input: 6.658765856

output: A soma entre o número 5 e o número 6.659 é 11.659

Ciclo: for

```
import numpy as np
for n in range(3, 11, 3):
    print(f'A raiz quadrada do número {n} é {np.sqrt(n):.2f}')
```

output: A raiz quadrada do número 3 é 1.73
 A raiz quadrada do número 6 é 2.45
 A raiz quadrada do número 9 é 3.00

Formato: range(início, fim + 1, passo)

função	Output
<code>range(5)</code>	{0, 1, 2, 3, 4}
<code>range(3, 9)</code>	{3, 4, 5, 6, 7, 8}
<code>range(1, 8, 2)</code>	{1, 3, 5, 7}

while e if

```
c = 1
while c < 10:
    print(c, end=" ")
    c += 1
print('FIM!')
```

output: 1 2 3 4 5 6 7 8 9 FIM!

```
c = 1
while c < 10:
    if c%2 != 0:
        print(c, end=" ")
    elif c == 4:
        print('UAU', end=" ")
    c += 1
print('FIM!')
```

output: 1 3 UAU 5 7 9 FIM!

Definição de funções

```
def soma(a,b):  
    return a+b
```

```
a = float(input('Escreva um número:'))  
b = float(input('Escreva um número:'))  
s = soma(a,b)  
print(f'A soma entre o número {a} e o número {b} é {s}')
```

input: 5 input: 10

output: A soma entre o número 5.0 e o número 10.0 é 15.0

Listas

```
lista1 = [4, 5, 7, 8]  
lista2 = list(range(4,11))  
lista1, lista2
```

output: ([4, 5, 7, 8], [4, 5, 6, 7, 8, 9, 10])

Operações sobre listas

```
# Insere elemento no fim da lista  
lista1.append(6)  
lista1
```

output: [4, 5, 7, 8, 6]

```
# Insere elemento numa posição  
lista1.insert(0,10)  
lista1
```

output: [10, 4, 5, 7, 8, 6]

```
# Retira um elemento da lista  
lista1.remove(8)  
lista1
```

output: [10, 4, 5, 7, 6]

Operações sobre listas

```
# Retira o último elemento da lista  
lista1.pop()  
lista1
```

output: [10, 4, 5, 7]

```
# Ordena os elementos por  
# ordem crescente  
lista1.sort()  
lista1
```

output: [4, 5, 7, 10]

```
# Ordena os elementos por  
# ordem decrescente  
lista1.sort(reverse = True)  
lista1
```

output: [10, 7, 5, 4]

```
# Obtém o número de elementos na lista  
len(lista1)
```

output: 4

Operações sobre listas

```
for c,v in enumerate(lista1):  
    print(f'Na posição {c} encontrei o valor {v}')
```

output: Na posição 0 encontrei o valor 10
Na posição 1 encontrei o valor 7
Na posição 2 encontrei o valor 5
Na posição 3 encontrei o valor 4

```
# Uma lista aceita diferentes  
# tipos de dados  
dados = list()  
dados.append('Pedro')  
dados.append(25)  
dados
```

output: ['Pedro', 25]

```
# Listas de listas  
dados = [[1, 2, 3], [5, 7, 10]]  
dados[1][0]
```

output: 5

```
# Soma de listas
```

```
x = [3, 4, 5]
```

```
y = [4, 9, 7]
```

```
x + y
```

output: [3, 4, 5, 4, 9, 7]

Array da biblioteca Numpy

```
# Soma de arrays
```

```
import numpy as np
```

```
x = np.array([3, 4, 5])
```

```
y = np.array([4, 9, 7])
```

```
x + y
```

output: array([7, 13, 12])

As matrizes são arrays.

Matrizes: Arrays

```
# Matriz de dimensão 2x2
import numpy as np
x = np.array([[1, 2], [3, 4]])
x
```

output : array([[1, 2],
[3, 4]])

Algumas operações com arrays

```
# Matriz de dimensão 2x4
import numpy as np
x = np.array([[1, 2, 5, 9], [3, 4, 6, 8]])
x.ndim, x.shape, x.dtype
```

output: (2, (2, 4), dtype('int32'))

Algumas operações com arrays

```
# Definir o tipo de dados
import numpy as np
x = np.array([[1, 2], [3, 4]], float)
x[0][1]
```

output: 2.0

```
# Redistribuir elementos por um dado número de
# linhas e de colunas
x = np.array([1, 2, 3, 4, 5, 6])
print('beginning x:', x)
xreshape = x.reshape((2,3))
print('reshaped x:\n', xreshape)
```

output: beginning x: [1, 2, 3, 4, 5, 6]
 reshaped x:
 [[1, 2, 3]
 [4, 5, 6]]

Algumas operações com arrays

```
# Diferentes tipos de dados num array
import numpy as np
x = np.array([[1, 2, 5, 9], ['a', 'b', 'c', 'd']])
x[0][1] + x[0][2]
```

output: 25

```
# Ver x
x
```

output : array([['1', '2', '5', '9'],
 ['a', 'b', 'c', 'd']], dtype = '<U11')

Algumas operações com arrays

```
import numpy as np
A = np.array(np.arange(16).reshape(4, 4))
A
```

```
output : array([[ 0,  1,  2,  3],
                [ 4,  5,  6,  7],
                [ 8,  9, 10, 11],
                [12, 13, 14, 15]])
```

```
A[1,2]
```

```
output: 6
```

```
A[[1,3]]
```

```
output : array([[ 4,  5,  6,  7],
                [12, 13, 14, 15]])
```

Algumas operações com arrays

```
array([[ 0,  1,  2,  3],  
       [ 4,  5,  6,  7],  
       [ 8,  9, 10, 11],  
       [12, 13, 14, 15]])
```

```
A[:, [0, 2]]
```

```
output : array([[ 0,  2],  
               [ 4,  6],  
               [ 8, 10],  
               [12, 14]])
```

```
A[[0, 2], :]
```

```
output : array([[ 0,  1,  2,  3],  
               [ 8,  9, 10, 11]])
```

Algumas operações com arrays

```
array([[ 0,  1,  2,  3],  
       [ 4,  5,  6,  7],  
       [ 8,  9, 10, 11],  
       [12, 13, 14, 15]])
```

```
A[[1, 3]][:, [0, 2]]
```

output :

```
array([[ 4,  6],  
       [12, 14]])
```

```
A[[1, 3], [0, 2]]
```

output :

```
array([[ 4, 14]])
```

Leitura de ficheiros e DataFrame

```
import pandas as pd
data = pd.read_csv('state.csv')
data.head(), data['State'].head(), data[:1], data.columns[0]
# Obtém a 4ª e 5ª linhas completas
data.iloc[[3, 4]]
# Obtém a 4ª e 5ª linhas e só os valores da 1ª e 3ª colunas
data.iloc[[3, 4],[0,2]]
# Faz:  $5.6 + 4.4 = 10.0$ 
data['Murder.Rate'].iloc[3] + data['Murder.Rate'].iloc[4]
```

	State	Population	Murder.Rate	Abbreviation
0	Alabama	4779736	5.7	AL
1	Alaska	710231	5.6	AK
2	Arizona	6392017	4.7	AZ
3	Arkansas	2915918	5.6	AR
4	California	37253956	4.4	CA

Algumas operações com DataFrame

```
import pandas as pd
import numpy as np
t = np.arange(0, 10, 0.1)
x = np.sin(t)
y = np.cos(t)
dados = {'Time':t, 'x':x, 'y':y}
df = pd.DataFrame(dados)
print(df)
# Imprime a coluna "Time"
print(df.Time)
print(df['Time'])
# Cria novo DataFrame
data = df[['Time', 'y']]
```

	Time	x	y
0	0.0	0.000000	1.000000
1	0.1	0.099833	0.995004
2	0.2	0.198669	0.980067
3	0.3	0.295520	0.955336
4	0.4	0.389418	0.921061
..
95	9.5	-0.075151	-0.997172
96	9.6	-0.174327	-0.984688
97	9.7	-0.271761	-0.962365
98	9.8	-0.366479	-0.930426
99	9.9	-0.457536	-0.889191

[100 rows x 3 columns]

Introdução à estatística

O que é a Estatística? Como funciona?

- Como é que ajuda a resolver alguns problemas práticos?
 - Que fração de portugueses preferem determinada marca de iogurtes?
 - Que fração de casas rurais na zona norte têm internet?
 - Que fração de peças produzidas numa determinada linha de produção são defeituosas?
- Para todos estes casos:
 - Selecionam-se indivíduos ou peças ao acaso e registam-se as preferências ou observações.
 - Pretende-se tirar conclusões sobre a totalidade da população (finita no caso das pessoas ou casas, infinita no caso da produção de peças), a partir da análise de um subconjunto da mesma.

Algumas definições

- **População** é o conjunto de todos os objetos, eventos ou observações, cuja(s) característica(s) pretendemos estudar. Podem ser **finitas** ou **infinitas**.
- **Amostra** é um subconjunto finito da população, obtido de modo aleatório e em que todos os elementos da população têm igual probabilidade de serem incluídos.
- **Amostragem** é uma técnica de seleção de elementos de uma população para se estimar propriedades e características da população.
- **Censo** é um estudo que inclui todos os elementos da população.
- **Sondagem** é um estudo (da população) efetuado a partir de uma amostra.
- A **teoria de probabilidades** permite a elaboração de previsões acerca dos resultados de fenómenos que dependem da sorte.

Definição de Estatística

Estatística é uma área científica cujo objetivo principal é observar um fenómeno, **recolher**, **analisar** e **interpretar** os dados e auxiliar na formulação de **estimativas** e **tomada de decisões**.

Métodos estatísticos

Os **métodos estatísticos** baseiam-se nas **teoria probabilísticas** e proporcionam-nos os modelos matemáticos para explicar a frequência de **fenómenos de natureza incerta** (não se consegue definir uma relação causa-efeito) e para possibilitar a **previsão** desses fenómenos, **estimando** o fator sorte.

Experiência estatística é qualquer processo que gera um conjunto de resultados que pode ser diferente de cada vez que este é executado, em iguais condições.

Estatística descritiva / Inferência estatística

A estatística é normalmente dividida em duas áreas:

- **Estatística descritiva** é um conjunto de métodos cujo objetivo é sintetizar e representar de forma compreensível a informação obtida dos dados.
- **Inferência estatística** é um conjunto de métodos que permitem elaborar previsões e tirar conclusões sobre uma **população** a partir da informação contida numa **amostra representativa** da população, medindo ou avaliando, o respetivo grau de incerteza.

