

## Por que não Excel?

Por muito tempo, linguagens de programação como Python foram vistas como restritas apenas a programadores e desenvolvedores de software, enquanto os demais profissionais utilizavam programas como Excel para suas análises. Hoje, esta barreira não existe, exige-se cada vez mais no mercado de trabalho familiaridade com linguagens de programação. Mesmo onde não há exigência, o profissional que se dedica a aprender programação pode ter ganhos de produtividade e eficiência.

## Principais vantagens do Python

**Automação:** Em sua essência, programação é automação. Ao desenvolver um script ou programa, nós somos obrigados a projetar nossa solução com automatização em mente, afinal, o que escrevemos são instruções de como o computador deve se comportar. Embora também exista este componente no Excel, a interface gráfica nos permite realizar muitas operações manualmente, o que pode economizar tempo no curtíssimo prazo, mas desincentiva o esforço de se planejar para a automação que garantirá eficiência no futuro.

**Reprodutibilidade:** Ao encapsular nossa análise em um programa de computador, garantimos que ela pode ser reproduzida de forma idêntica por nós ou por colegas, sem depender de grandes explicações. Um problema comum com o Excel, é o excesso de processamento manual de dados, que gera a dependência de que devemos sempre garantir que estamos executando exatamente os mesmos passos quando, por exemplo, recebermos dados mais atualizados. O Python garante que os mesmos procedimentos sempre serão feitos.

**Transparência:** Apesar do Excel gerar excelentes análises, nós perdemos ao longo do caminho quais foram os passos para gerá-las. Quais colunas foram deletadas? Existiram valores faltantes que foram preenchidos? Com qual técnica eles foram preenchidos? O código Python é em si uma documentação de todos os passos, que nos permite (e colegas) a auditar e a garantir a integridade da análise.

## Ferramentas que iremos utilizar

### Anaconda

Anaconda é uma distribuição de Python que já vem com os principais pacotes que iremos utilizar no curso. Além disso, junto com a instalação do Anaconda são instaladas ferramentas como o SpyderIDE e o Jupyter Notebook, que usaremos ao longo do curso. O Anaconda pode ser instalado



para Windows, Linux ou Mac e pode ser baixado no link <https://www.anaconda.com/products/individual>. Para instruções detalhadas de instalação, assista o vídeo que disponibilizamos na plataforma Moodle.

### Jupyter Notebook

Jupyter Notebook é um caderno interativo. Ou seja, é um caderno em que podemos escrever e executar código. Toda a parte prática do curso será realizada no Jupyter. Durante as aulas ao-vivo, criaremos em conjunto um caderno que resolve as atividades do dia, programando em tempo real a solução para nossas análises. Ao final da aula, este arquivo será disponibilizado no Moodle e servirá como referência para futuros projetos e empreitadas. Um tutorial de como utilizar a ferramenta também está disponível no Moodle.

### Pacotes Python

O Anaconda já vem com diversos pacotes do Python que usaremos. Abaixo uma breve introdução a cada um deles.

#### Python core

O Python core não é em si um pacote, é o conjunto de funções que vem automaticamente com o Python. O Python nativo por si só já é bastante poderoso, possui estruturas de dados versáteis como listas, tuplas e dicionários. Também conta com declaração de funções, controles de fluxo e outras características comuns a outras linguagens de programação.

**LEITURA RECOMENDADA:** DOCUMENTAÇÃO DO PYTHON, PRINCIPALMENTE SESSÕES 1, 3 E 4  
[DOCS.PYTHON.ORG/PT-BR/3/TUTORIAL/](https://docs.python.org/pt-br/3/tutorial/)

### Numpy

Numpy é o pacote utilizado para programação numérica. Ele nos permite trabalhar com vetores e matrizes de diversas dimensões de forma rápida e eficiente.

Exemplo:

```
a = [[1, 0], [0, 1]]
b = [[4, 1], [2, 2]]
np.dot(a, b)
array([[4, 1],
```



```
[2, 2]])
```

A leitura não é necessária, mas a documentação pode ser acessada em <http://pyscience-brasil.wikidot.com/module:numpy> para maiores detalhes.

## Pandas

Pandas é a biblioteca que fornece funções para gerenciamento e análise de dados, com ela é possível ler dados de planilha Excel, limpá-los, visualizá-los e performar diversas análises com rapidez e escalabilidade. A biblioteca Pandas será a mais utilizada durante o curso.

### Exemplo:

```
In [1]: import pandas as pd  
file_name = "https://people.sc.fsu.edu/~jburkardt/data/csv/homes.csv"  
df = pd.read_csv(file_name)
```

```
In [2]: df.head()
```

Out[2]:

	Sell	"List"	"Living"	"Rooms"	"Beds"	"Baths"	"Age"	"Acres"	"Taxes"
0	142	160	28	10	5	3	60	0.28	3167
1	175	180	18	8	4	1	12	0.43	4033
2	129	132	13	6	3	1	41	0.33	1471
3	138	140	17	7	3	1	22	0.46	3204
4	232	240	25	8	4	3	5	2.05	3613

Todas as funções que podem ser realizadas com Data Frames podem ser encontradas em <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>

## Matplotlib

Matplotlib é a biblioteca que permite a criação de gráficos e visualizações. Exemplos de gráficos que podem ser criados com o módulo podem ser encontrados em <https://matplotlib.org/gallery/index.html>.