

# Software Livre científico: O que temos?

(De onde viemos? Para onde vamos?)

Renan Birck Pinheiro  
[[renan.ee.ufsm@gmail.com](mailto:renan.ee.ufsm@gmail.com)]

# O autor

- Formado em Engenharia Elétrica pela UFSM;
  - Curso onde o MATLAB é quem manda, na prática
- Hardware/software/sistemas embarcados;
- Tarefas diversas de simulação, análise de dados etc...;

# Problemas das Ferramentas Proprietárias


## Pricing and Licensing

 Contact sales  Request a Quote

 Standard

 Education

 Home

 Student

MATLAB  
**\$2,650**

*Individual License*

[Request a quote](#)

[—View another product—](#) ▼

### Individual License

Select this license option if you are an end user and you want to operate, install, and administer the software yourself.

This license option:

- Offers choice between two activation types: Standalone Named User and Designated Computer
- Allows installation on the following platforms: Windows, Linux, and Macintosh

[» Learn more](#)

US\$ 2650 x dólar a R\$ 3,80 = **R\$ 10200**  
(e isso pelo MATLAB sem nenhuma *toolbox*)

# Outros problemas

Além do custo:

- Cada ferramenta usa uma linguagem proprietária;
- Ferramentas criadas especificamente para aplicações científicas, dificultando outras tarefas comuns (ex.: criação de GUIs ou servidores, conexão com banco de dados etc...);
- Difícil (senão impossível) usar como backend de uma ferramenta mais complexa;
- Algoritmos fechados prejudicam a reprodutibilidade (necessária ao método científico).

# Algumas ferramentas

- Octave, Scilab
- Sage
- Python + NumPy + SciPy + matplotlib + outras



# Octave / Scilab

- <http://www.octave.org> E <http://www.scilab.org>
- Duas ferramentas que tentam se aproximar do MATLAB
  - Vantagem: compatibilidade de código
  - Porém:
    - Linguagem obsoleta (pouco suporte a OO, namespaces etc...);
    - Tiveram que copiar todos os vícios e bugs do MATLAB;
    - Na prática são clones incompletos.

# Sage

- <http://www.sagemath.org>;
- Vários pacotes matemáticos na mesma ferramenta, com uma interface *notebook*;
- Grande, confuso e ruim de instalar;
- Voltado à matemática pura/aplicada.

Então, vamos ao Python!



# Por quê?

- Bibliotecas abrangentes para diversas aplicações (científicas ou não);
- Linguagem simples e moderna;
- Comunidade forte (Stack Overflow, GitHub, redes sociais, etc...).

- Dica: Anaconda

- <http://continuum.io/downloads>
- Instalação fácil do ambiente Python + bibliotecas de uso científico
  - Mesmo que você não seja root
  - Mesmo que você esteja usando Windows
  - Não afeta o Python do SO (ótimo para distros *stable*).

# Por que não?

- Problemas com desempenho;
- Ambiente menos integrado;
- Falta de bibliotecas para aplicações específicas.

# NumPy

- Permite trabalhar com matrizes, números complexos etc... de forma nativa, além de fornecer outras funções

# SciPy

- Funções para integração numérica, processamento de sinais e de imagens, otimização etc...
- Usa a NumPy como framework



# matplotlib

- Criação de gráficos 2D/3D;
- Duas formas de uso:
  - Programação imperativa (similar ao MATLAB)
  - OO (cria-se um objeto gráfico e daí se constrói a imagem)

# ipython / Jupyter

- Permite trabalhar com Python (e com diversas outras linguagens) de forma interativa, dentro do browser.
- Na prática é um “editor de texto” no qual se mistura texto e código/resultados.

Podemos usar as ferramentas do NumPy e SciPy, por exemplo, definindo matrizes. A propósito, podemos escrever expressões em LaTeX:

$$A = \begin{pmatrix} 3 & 2 & 4 \\ 6 & 8 & 1 \\ 5 & 5 & 9 \end{pmatrix}$$

```
In [6]: A = np.matrix([[3, 2, 4],
                      [6, 8, 1],
                      [5, 5, 9]])
det(A)
```

Out[6]: 62.999999999999993

E também podemos plotar gráficos. Vejamos como isso será feito para um gráfico exemplo:

```
In [7]: fig = figure(figsize=(12, 9)) # Criar figura grande
years = [2010, 2011, 2012, 2013, 2014, 2015] # Vetores de dados para exemplo
sales_A = [100, 150, 200, 300, 100, 150]
sales_B = [150, 220, 300, 330, 75, 55]

plot(years, sales_A, color='red', linestyle='-', marker='*', markerfacecolor='red')
plot(years, sales_B, color='black', linestyle='-', marker='*', markerfacecolor='black')
xlabel('Ano')

# Definir limites dos eixos
xlim([2010, 2015])
ylim([50, 350])

ax = axes()
ax.set_xticklabels(years)
ylabel('Vendas (R$ mil)')
legend(['Carro A', 'Carro B'])
grid()
```



# pandas

- Trabalhar com arquivos CSV, Excel etc... de forma simples e eficiente.

# Alguns exemplos de aplicação



# Exemplo 1: FFT

- Transformada rápida de Fourier
  - Passar sinal do tempo para frequência
  - Uma das ferramentas mais usadas em EE, processamento de sinais etc...

# Octave

# Python

## Exemplo 2: filtrar sinal

- Sinal (senoide) com mistura de frequências
- Queremos pegar apenas uma delas

# Octave



# Python

# OBRIGADO PELA ATENÇÃO!

Slides e materiais dessa apresentação:  
<http://github.com/renanbirck/sfdsm-2015>

Contatos:

- <http://github.com/renanbirck>
- <http://fb.com/renanbirck>
- <http://twitter.com/birckrenan>