

# **Programação Científica e Ciência de Dados**

# Ciência de Dados

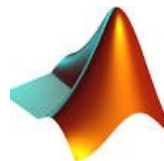
- Objetivos:
  - extração de conhecimento
  - detecção de padrões
  - obtenção de insights
  - tomadas de decisão

# Ciência de Dados

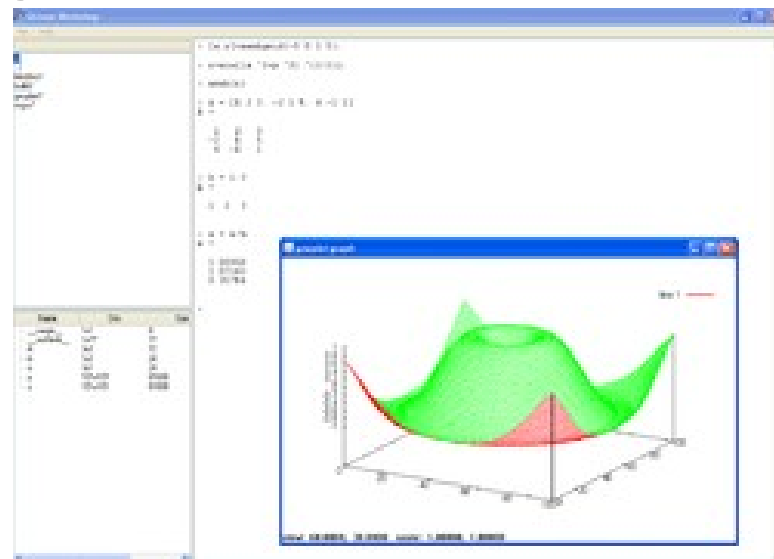
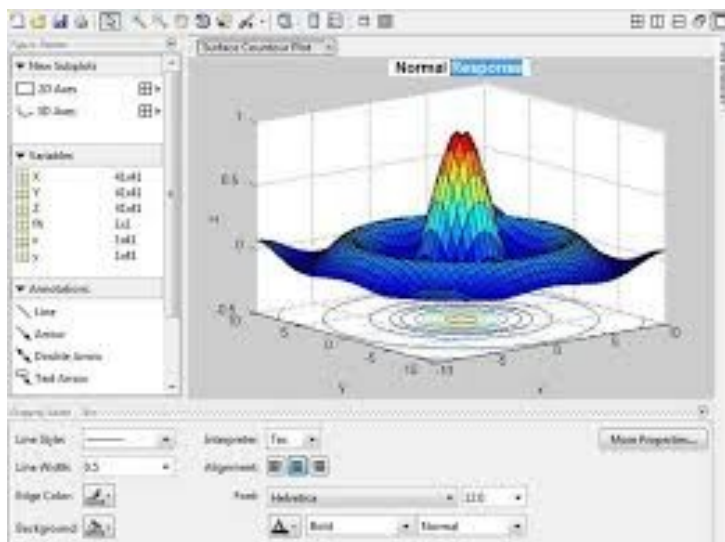
- Ganhou destaque nos últimos anos
  - surgimento e popularização de grandes bases de dados
  - Big data
  - desenvolvimento de áreas como machine learning

# Programação Científica

- Matlab/Octave



- É um poderoso ambiente de trabalho usado para implementar e analisar algoritmos numéricos



# Programação Científica



- R
  - R é uma linguagem de programação multi-paradigma (com ênfase em programação funcional)
  - é largamente usada entre estatísticos e analistas de dados para desenvolver software de estatística e análise de dados

# Programação Científica

- Python



- Existem muitas bibliotecas disponíveis para realizar a análise de dados em Python
- NumPy, SciPy, Matplotlib, IPython, SimPy, Pandas, Scikit-learn



# Programação Científica



- Julia
  - Ganhando popularidade entre os Cientistas de Dados devido sua flexibilidade e facilidade de uso
  - É uma linguagem de bom desempenho, que se aproxima de linguagens estaticamente compiladas como C
  - possui uma extensa lista de bibliotecas e pacotes prontos para uso

# Programação Científica

- Julia X Python

- Julia é rápida
- Sintaxe matemática mais amigável
- Gerenciamento de memória automático
- Paralelismo mais eficiente

- Linguagem madura
- Mais pacotes e bibliotecas
- Milhões de usuários e comunidade ativa
- Ganhando eficiência com melhorias e compilações



# Programação Científica

- Comparações

<https://modelingguru.nasa.gov/docs/DOC-2783>

Table FFT-1.0: Elapsed times to compute the FFT on the Xeon node.

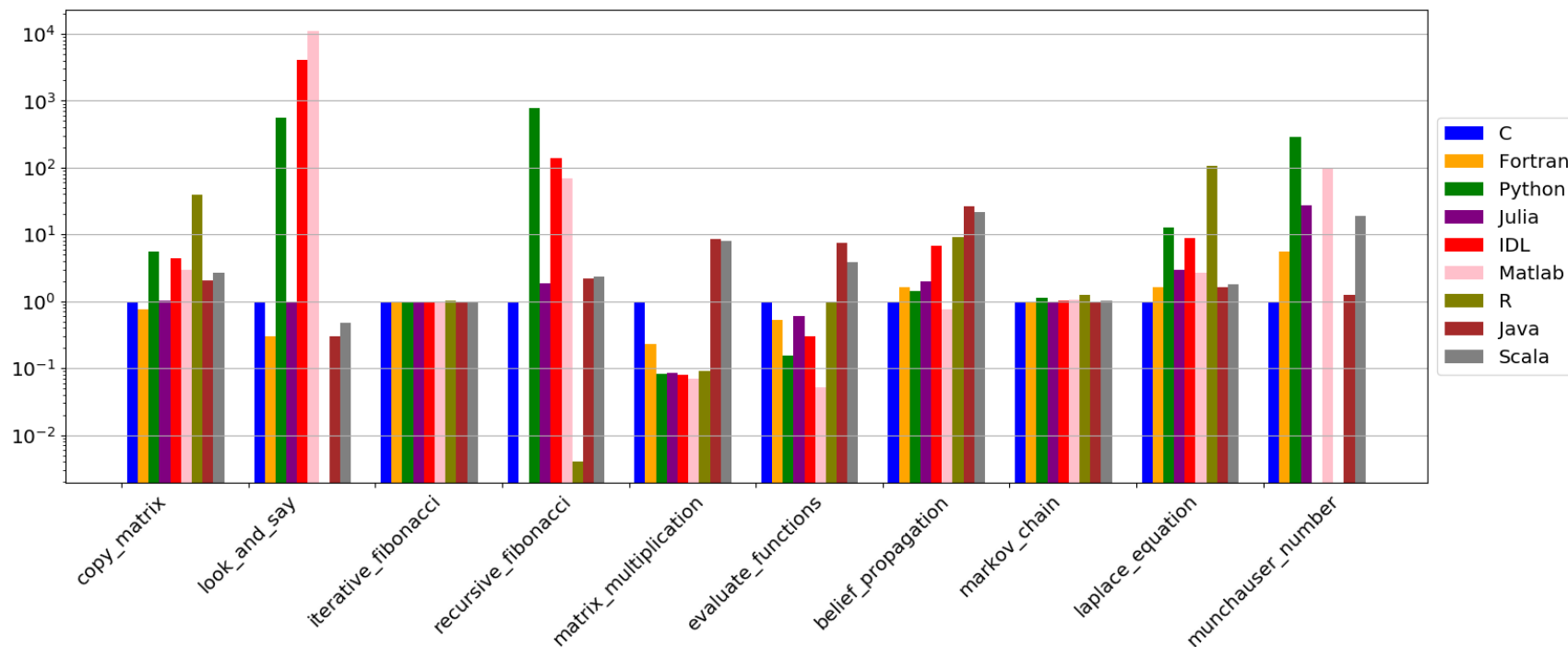
Language	Option	$n=10000$	$n=15000$	$n=20000$
Python	intrinsic	8.0797	19.6357	34.7400
Julia	intrinsic	3.979	11.490	20.751
IDL	intrinsic	16.6699	38.9857	70.8142
R	intrinsic	58.2550	150.1260	261.5460
Matlab	intrinsic	2.6243	6.0010	10.66232

Table MXM-1.0: Elapsed times to multiply the matrices on the Xeon node.

Language	Option	$n=1500$	$n=1750$	$n=2000$
Python	intrinsic	0.1560	0.2430	0.3457
Julia	intrinsic	0.1497	0.2398	0.3507
Java	loop	13.8610	17.8600	32.3370
Scala	loop	9.8380	19.1450	32.1310
R	intrinsic	0.1600	0.2460	0.3620
Matlab	intrinsic	1.3672	1.3951	0.4917
IDL	intrinsic	0.1894	0.2309	0.3258
Fortran	gfortran (loop)	17.4371	31.4660	62.1079
	gfortran -O3 (loop)	3.3282	5.3003	12.1648
	gfortran (matmul)	0.3840	0.6160	0.9241
	gfortran -O3 (matmul)	0.3880	0.6160	0.9161
	ifort (loop)	1.1401	1.8161	2.9282
	ifort -O3 (loop)	1.1481	1.8081	2.9802
	ifort (matmul)	1.1441	1.8121	2.9242
	ifort -O3 (matmul)	0.5160	0.8281	1.2441
	ifort (DGEMM)	0.2160	0.2360	0.3320
C	gcc (loop)	13.2000	20.9800	31.4400
	gcc -Ofast (loop)	1.4500	2.3600	4.0400
	icc (loop)	1.2300	2.1500	4.0500
	icc -Ofast (loop)	1.1500	1.7500	2.5900

# Programação Científica

- Comparações



# Programação Científica

- <https://abandre.github.io/livros/>

