



Mini-Curso: SELMAT
XXXVI Semana da Licenciatura em Matemática

Aventuras matemáticas com a linguagem JULIA

Prof. M.e Daniel Cassimiro Fernandes

Profa. Dra. Sônia Cristina Poltroniere

24 de setembro de 2024



unesp

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"



Conteúdos

- ▶ Sobre a linguagem Julia
 - Aplicativos
- ▶ Pacotes
- ▶ Controles de fluxo
- ▶ Julia como calculadora
 - Funções e constantes elementares
 - Matrizes



A linguagem de programação Julia foi desenvolvida visando atender aos requisitos da computação numérica de alto desempenho e científica.

Seus criadores quiseram reunir na linguagem as principais características de outras linguagens:

- Ruby
- Matlab
- R
- Julia
- C.



A linguagem de programação Julia é uma linguagem moderna e poderosa que foi criada para atender aos requisitos da computação numérica de alto desempenho e científica. Seus criadores quiseram reunir na linguagem as principais (melhores) características de outras linguagens:

- Ruby;
- Matlab;
- R;
- Julia;
- C.

Julia é uma linguagem de programação de alto nível, interpretada e multi-paradigma.

- Tipagem Dinâmica: suas variáveis podem receber qualquer tipo de dado e sua sintaxe se aproxima mais da linguagem humana do que da linguagem de máquina.
- Multiparadigma: suporta diversos paradigmas de programação, como orientação a objetos e programação funcional.
- Alto Nível: possui uma sintaxe expressiva e amigável.
- Gratuita e Open Source: Julia é distribuída sob a licença MIT.
- Suporte a Unicode e UTF-8: permite o uso de símbolos matemáticos durante a escrita de programas.
- Gerenciador de pacotes práticos: facilita a instalação e atualização de pacotes.

- Data Science: Julia é usada para análise de dados e descoberta de conhecimento a partir de grandes conjuntos de dados.
- Machine Learning: possui pacotes para criação de modelos de aprendizado de máquina.
- Computação Científica: ideal para construir modelos matemáticos e soluções numéricas.
- Desenvolvimento Geral: pode ser aplicada no desenvolvimento de aplicações web, desktop e outras áreas.
- Em resumo, Julia é uma linguagem versátil que combina alta performance com uma sintaxe amigável, tornando-a uma escolha popular entre cientistas de dados, desenvolvedores e entusiastas da programação.



Aplicativos que serão utilizados neste Mini-curso



Linguagem Julia



ANACONDA®

Ambiente de trabalho Anaconda



Visual Studio

Editor de código-fonte VS Code



Aplicativos que serão utilizados neste Mini-curso

README

Linguagem JULIA

Tutorial de instalação:

1. [Instalação do Julia pelo windows powershell](#)
2. [Instalação do Anaconda pelo windows powershell](#)
3. [Instalação do Vs Code pelo windows powershell](#)

4. [Instalação do Julia pelo modo gráfico](#)
5. [Instalação do Anaconda pelo modo gráfico](#)

5. [Julia "Hello World"](#)
6. [Integração da linguagem Julia no jupyter-notebook](#)
7. [Exemplo de otimização](#)

Julia + Jupyter Notebook + VS Code

Para versões \emph{Windows}, utilize o Prompt de comando \emph{PowerShell} para inserir o comando de instalação a seguir:

```
winget install julia -s msstore
winget install --id=Anaconda.Anaconda3 -e
winget install -e --id Microsoft.VisualStudioCode
```

Para versões \emph{Linux}, \emph{Ubuntu} ou \emph{Debian}, basta utilizar os comandos a seguir e esperar a instalação:

```
sudo apt install julia -y
sudo apt install wget
wget https://repo.anaconda.com/archive/Anaconda3-2024.06-1-Linux-x86_64.sh
bash Anaconda3-2024.06-1-Linux-x86_64.sh
sudo snap install --classic code
```

Tutorial de instalação



Conteúdos

- ▶ Sobre a linguagem Julia
 - Aplicativos
- ▶ Pacotes
- ▶ Controles de fluxo
- ▶ Julia como calculadora
 - Funções e constantes elementares
 - Matrizes

Assim como acontece com a maioria das linguagens de programação modernas, o ecossistema de pacotes criados por outras pessoas facilita diversas tarefas. Alguns exemplos são:

- Pacotes para análises estatísticas: `DataFrame`, `StatsBase`
- Pacote de operações avançadas em Álgebra Linear: `LinearAlgebra`
- Pacotes gráficos: `Plots`, `Luxor`
- Pacotes de Otimização (solvers): `Gurobi`, `CPLEX`

No link <https://juliapackages.com/> existe uma listagem desses pacotes.

Para instalarmos os pacotes deveremos abrir o terminal Julia:

Com o terminal aberto, aperte a tecla] (fechar o colchetes), o que mudará o seu console de `julia>` para `version pkg>`.

Para adicionar um pacote, incluiremos:

```
1 > add nome-do-pacote
```

Para remover um pacote, incluiremos:

```
1 > rm nome-do-pacote
```

Para listar os pacotes listados, incluiremos:

```
1 > status
```

Para voltar para o modo normal do Julia, aperte ← backspace

Para carregar um pacote instalado, digite:

```
1 > using nome-do-pacote
```

Por exemplo, para instalar o pacote Luxor, no terminal:

```
1 julia> ] # Altera o terminal para o modo pacotes
2 v pkg> add Luxor # Adiciona o pacote Luxor
3 v pkg> backspace # Voltar ao terminal do modo de interpretação
4 julia> using Luxor # Comando para iniciar o uso do pacote, podendo ser
    incluso em scripts
```



Conteúdos

- ▶ Sobre a linguagem Julia
 - Aplicativos
- ▶ Pacotes
- ▶ **Controles de fluxo**
- ▶ Julia como calculadora
 - Funções e constantes elementares
 - Matrizes

Em Julia, o valor lógico verdadeiro é escrito como `True` e o valor lógico falso como `False`. Temos os seguintes operadores lógicos disponíveis:

```
1 && # e lógico
2 || # ou lógico
3 ! # negação
4 == # igualdade
5 != # diferente
6 < # menor que
7 > # maior que
8 <= # menor ou igual que
9 >= # maior ou igual que
```

Avaliação Condicional

```
1 if x < y
2     println("x menor y")
3 elseif x > y
4     println("x maior y")
5 else
6     println("x igual y")
7 end
```

Estrutura de repetição: while

```
1 i = 1;
2
3 while i <= 3
4     println(i)
5     global i += 1
6 end
7 1
8 2
9 3
```


Estrutura de repetição: for

```
1 for i = 1:3
2     println(i)
3 end
```

```
1  
2 function nome(variável)  
3     # instruções  
4 end
```



Conteúdos

- ▶ Sobre a linguagem Julia
 - Aplicativos
- ▶ Pacotes
- ▶ Controles de fluxo
- ▶ Julia como calculadora
 - Funções e constantes elementares
 - Matrizes

Em Julia, os operadores matemáticos elementares são os seguintes:

```
1  + # adição
2  - # subtração
3  * # multiplicação
4  / # divisão de a por b
5  \ # divisão de b por a
6  % # Resto da divisão euclidiana
7  ÷ # Quociente inteiro da divisão euclidiana (alt + 246)
8  (ou \div + tab)
9  ^ #potenciação
10 // # Frações
11 exp() #potenciação de base e
12 Log() #Logarítimo Neperiano
13 Log10() #Logarítimo de base 10
```

1

$$2//3+3//4$$



Funções e constantes elementares

Várias funções e constantes elementares estão disponíveis nativamente na linguagem Julia. Por exemplo:

```
1 julia>  $\pi$  = pi (\pi + tab)
2  $\pi$  = 3.1415926535897...
3
4 julia> cos( $\pi$ )
5 -1.0
6
7 julia> exp(1)
8 2.718281828459045
9
10 julia> log(exp(1))
11 1.0
```

1 $f(x) = x^2 - 2$

```
1 using Plots
2 plot(f,-2,4,framestyle = :origin)
3 plot!(g)
```



```
1 1:1:10
```

1

v.+1

```
1 A = [ 1 2;  
2      3 4]
```



Obtendo dados de uma matriz

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

Jeff Bezanson, Stefan Karpinski, Viral B. Shah e Alan Edelman Documentação Julia. <https://docs.julialang.org/en/v1/>

Guia introdutório <https://juliadatascience.io/pt/>

JuMP style guide. (2021).

<https://jump.dev/JuMP.jl/v0.21/developers/style/#using-vs.-import>

Guia Julia <https://www.juliasos.com/>



Aventuras matemáticas com a linguagem JULIA

Obrigado!