

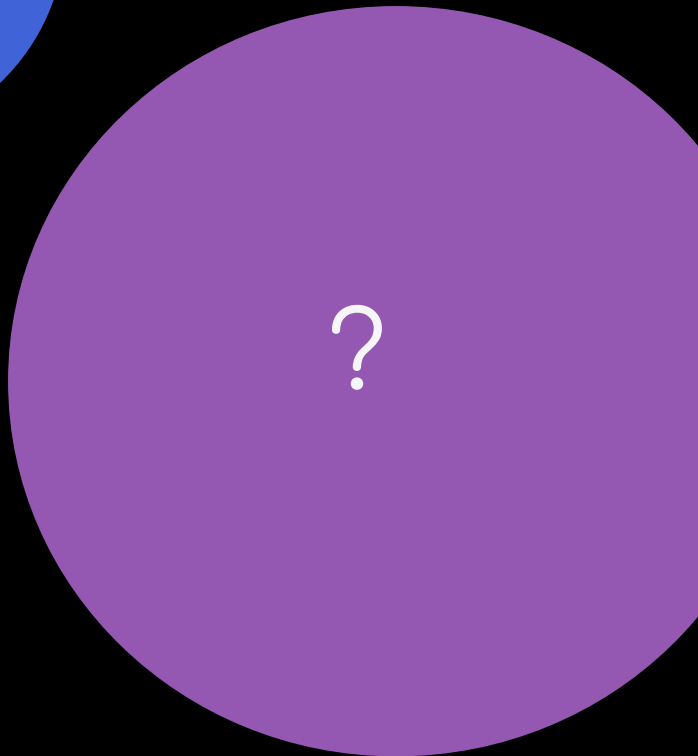
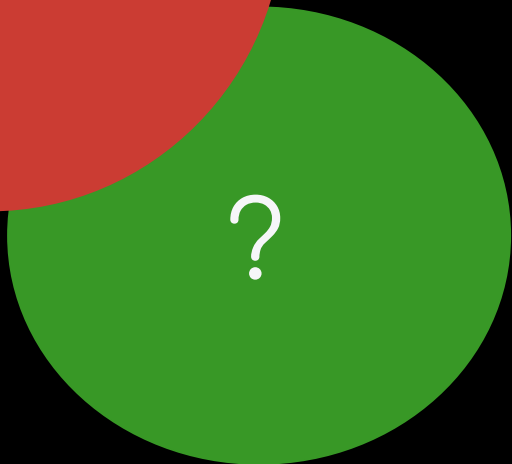
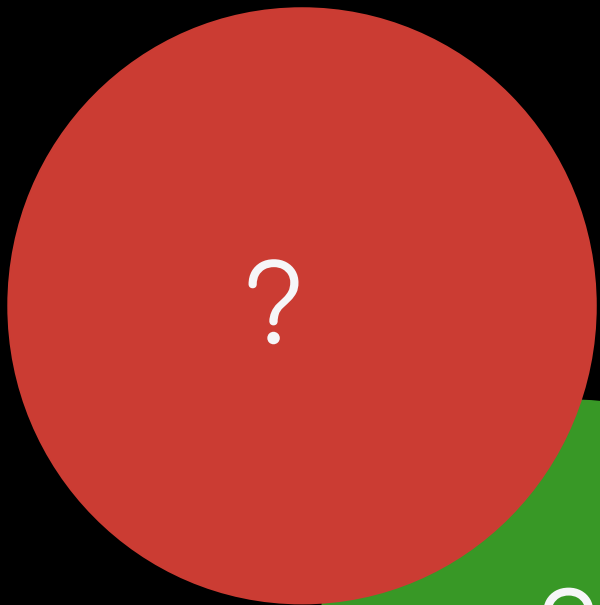


Introdução à Julia para Estatísticxs

04/06/2021

Sobre mim

- Me chamo Augusto (CJ), BECD 020.
- GEPAC (IFSC)
- E o que eu mais gosto em Julia é...



O que é Julia?

- Dinâmica de Alto nível
- Computação de alto desempenho numérico e científico
- Just-Ahead-Of-Time (JAOT)
 - LLVM compiler infra
- Read-eval-print loop (REPL)

Onde surgiu?

- Julia project, MIT, 2009
- Objetivo: Linguagem livre, de alto nível e rápida
- Conseguiram?


“We are greedy: we want more.”
(Bezanson et al, 2012)

- Queremos uma linguagem livre
- Queremos a velocidade de C e o dinamismo de Ruby
- Queremos uma linguagem que seja *homoiconic* com verdadeiros macros como Lisp, mas que tenha a notação matemática do Matlab
- [...]
- Queremos interativo e compilado

O que veremos no workshop :)

- Introdução à linguagem
 - Variáveis, tipagem de dados, operações matemáticas e lógicas, estruturas de controle de fluxo, funções e métodos, broadcasting e piping in Julia
- Bibliotecas de Visualização e Manipulação de Dados
 - DataFrames - Manipulação de dataframes
 - StatsBase - Medidas estatísticas
 - Gadfly - Alguns gráficos bonitos

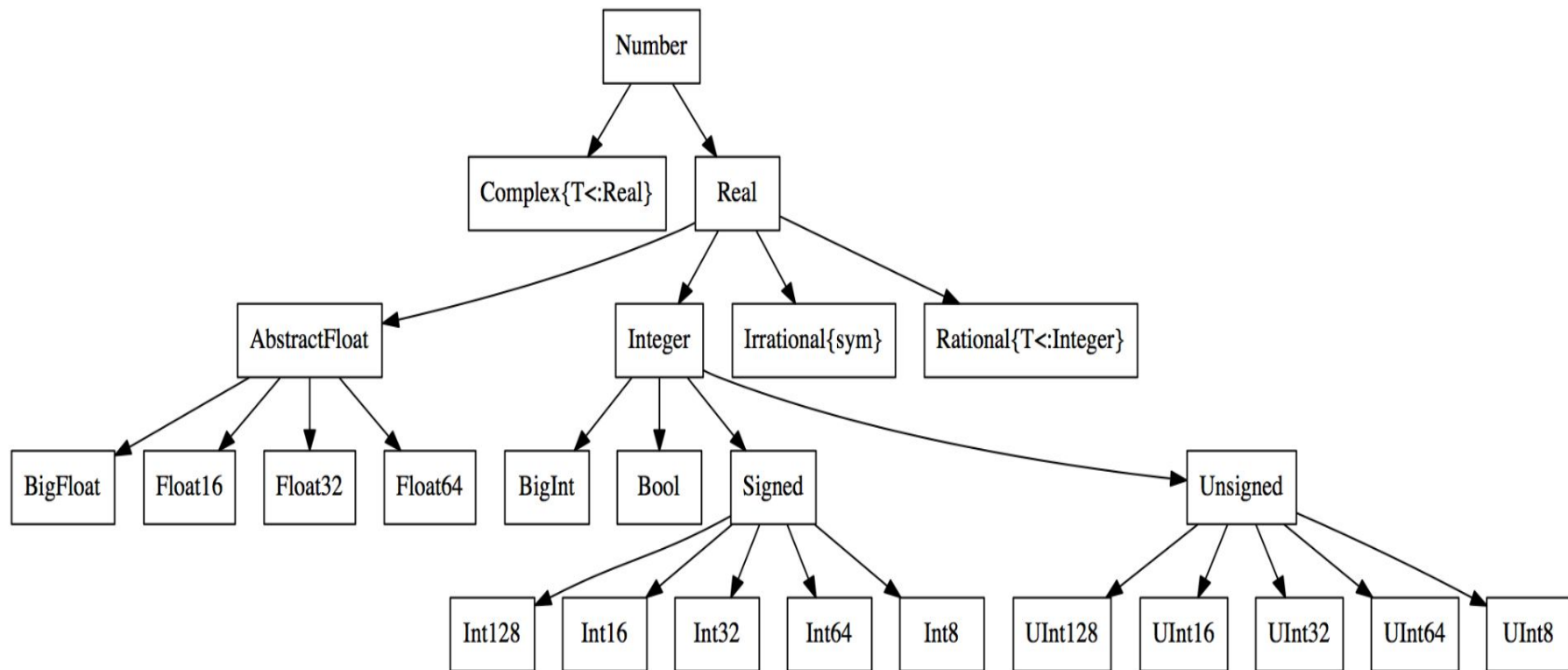
julia

The logo for the Julia programming language. It features the word "julia" in a bold, black, sans-serif font. Above the letter 'j' is a blue circle. Above the letter 'i' is a red circle. Above the letter 'l' is a green circle. Above the letter 'a' is a purple circle. The circles are arranged in a slightly overlapping, triangular pattern.

Introdução à linguagem

Variáveis e tipos de Dados

- O que são variáveis?
- Dinamicamente tipada
- Nomes de variáveis e convenções



Operações lógicas e matemáticas

Expression	Name	Description
<code>+x</code>	unary plus	the identity operation
<code>-x</code>	unary minus	maps values to their additive inverses
<code>x + y</code>	binary plus	performs addition
<code>x - y</code>	binary minus	performs subtraction
<code>x * y</code>	times	performs multiplication
<code>x / y</code>	divide	performs division
<code>x ÷ y</code>	integer divide	x / y , truncated to an integer
<code>x \ y</code>	inverse divide	equivalent to y / x
<code>x ^ y</code>	power	raises x to the y th power
<code>x % y</code>	remainder	equivalent to <code>rem(x, y)</code>

Expression	Name
<code>!x</code>	negation
<code>x && y</code>	short-circuiting and
<code>x y</code>	short-circuiting or
Operator	Name
<code>==</code>	equality
<code>!=, ≠</code>	inequality
<code><</code>	less than
<code><=, ≤</code>	less than or equal to
<code>></code>	greater than
<code>>=, ≥</code>	greater than or equal to

Listas, Dicionários, Tuplas e Estruturas

- Lista = [x1, x2, x3]
- Dicionários = Dict("n1" => x1, "n2" => x2)
- Tuplas = (x1, x2)
 - Listas imutáveis
 - Tuplas nomeadas
- Estruturas
 - define uma lista agrupada de variáveis sob um nome na memória

Funções e métodos

- Funções são objetos que mapeiam uma tupla de valores para um valor retornado
 - `f: x::Any -> y::Any`
- Existem várias maneiras de declaração (unicode :D)
- Funções anônimas
- Métodos !!!1!1!!!
 - É comum utilizar a mesma função para diversos tipos de dados
 - `+` para adição de inteiros e float
 - `*` para multiplicação de inteiros e concatenação de strings
 - Mas como isso funciona? (Spoiler: Multiple dispatch)

Estruturas condicionais

- Executa um bloco de instruções em função de uma condição
- if elseif else
- cond ? (if true) : (if else)
- Short-circuit Evaluation

Estruturas de Repetição

- Laço for
- Laço while
- Keywords
 - continue
 - break

Broadcasting e Piping

- Broadcasting
 - Vetorização de funções
 - `sin(x)` vs `sin.(x)`
- Piping
 - Composição de funções
 - `|>`

Metaprogramação

- Expression
 - Symbols
 - Arguments
- Symbol
- Macros
 - Mapeia uma tupla de argumentos para uma Expression
 - $f : \text{expr} \rightarrow \text{expr}$



[Don't do it](#) (Johnson, JuliaCon 2019)

Ecosystems

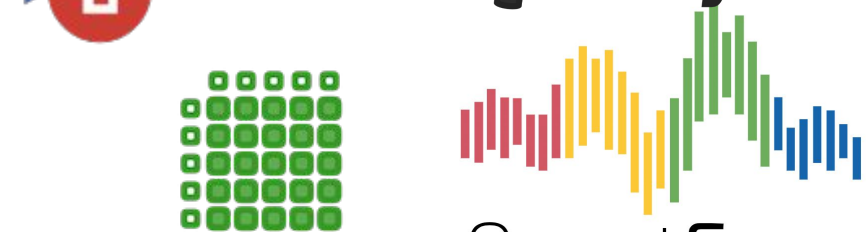
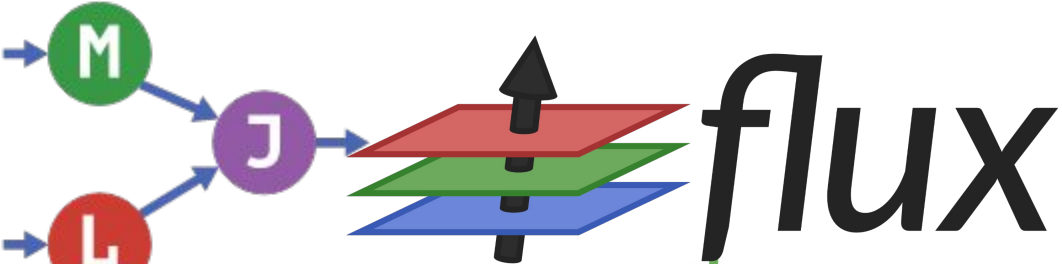
- Visualization: Plots.jl, Makie.jl, Gadfly.jl, UnicodePlots.jl
- Data Science: DataFrames.jl, CSV.jl, JuliaGraphs
- Machine Learning: MLJ.jl, Flux.jl, TextAnalysis.jl, AlphaZero.jl, Turing.jl
- Scientific Domains: Optim.jl, DifferentialEquations.jl, IterativeSolvers.jl, JuliaImages, JuliaAstro, BioJulia, JuliaDynamics...

**BIBLIOTECAS
JULIA**

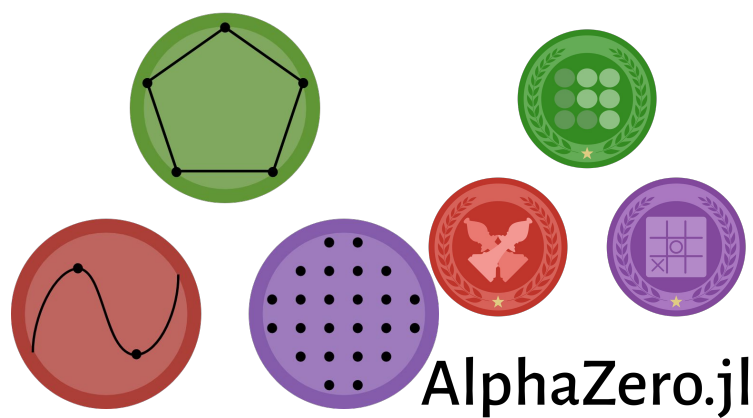
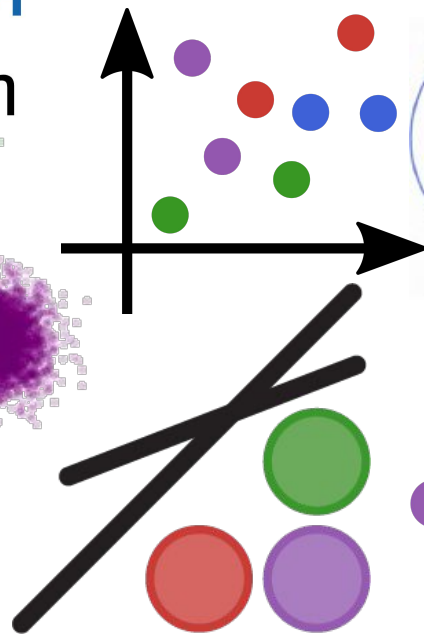
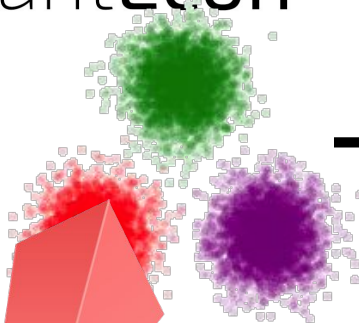
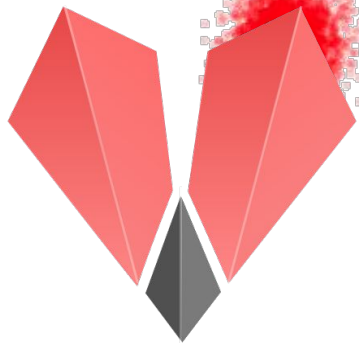
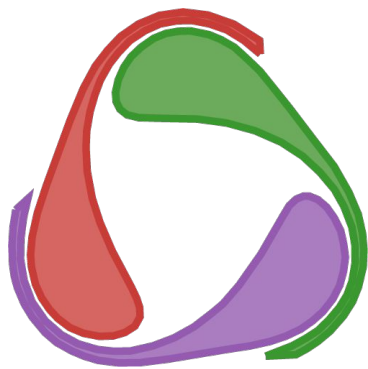
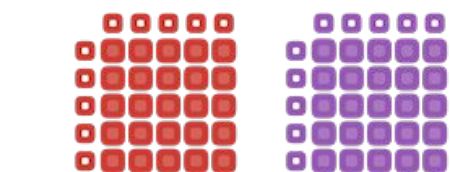


JULIA

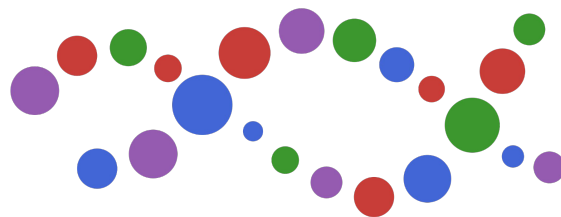
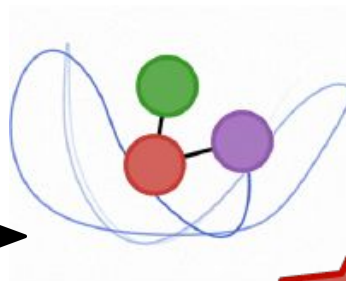


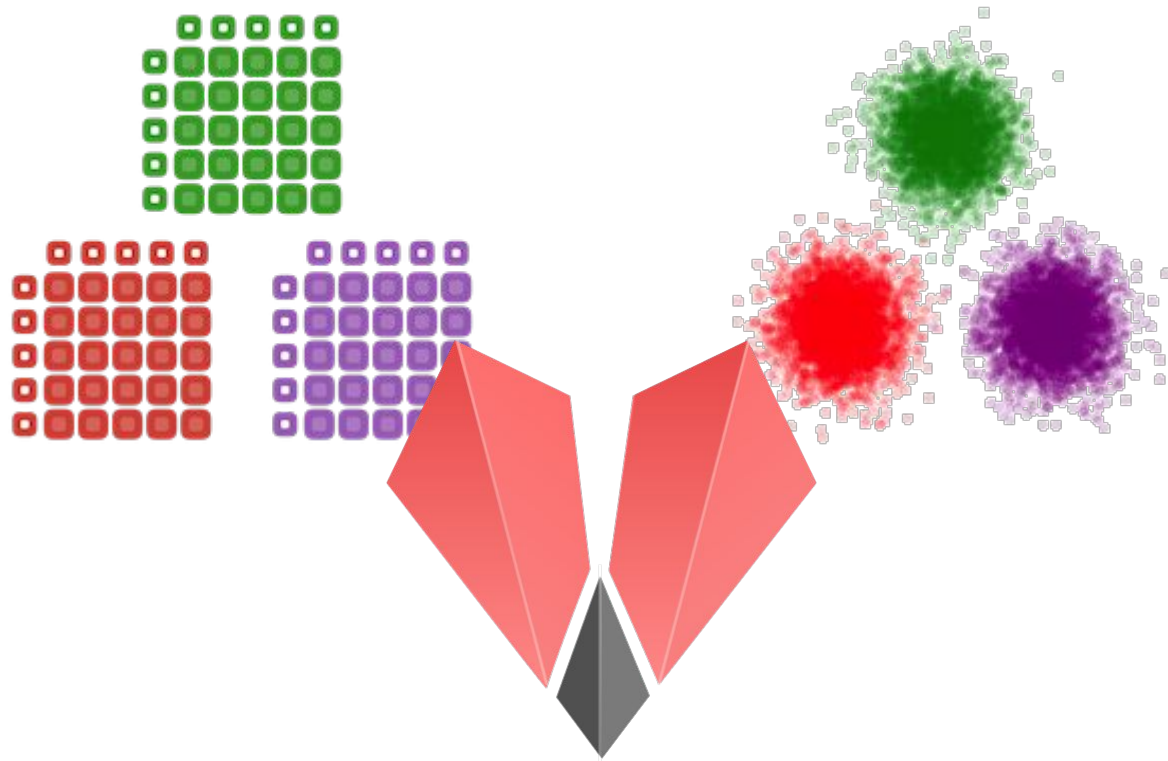


QuantEcon



AlphaZero.jl





Visualização e Manipulação de Dados

Referências

- [Julia.org](https://julialang.org)
- [Julia Docs](https://docs.julialang.org)
- [Gadfly Docs](https://gadfly.julia.dev)
- [DataFrames.jl Docs](https://dataframes.julia.dev)
- [DataFramesMeta.jl Docs](https://dataframesmeta.julia.dev)
- [Distributions.jl Docs](https://distributions.julia.dev)

Outras Sugestões

- [Julia for Data Science](#)
- [Julia for nervous beginners](#)
- [Computational Modelling in Julia](#)
- [Reddit](#)
- [Twitter](#)
- [JuliaCon 2021](#)



Obrigado

- E-mail: augustonunes@usp.br
- GitHub: `augusnunes`
- Telegram: `@augustocj`