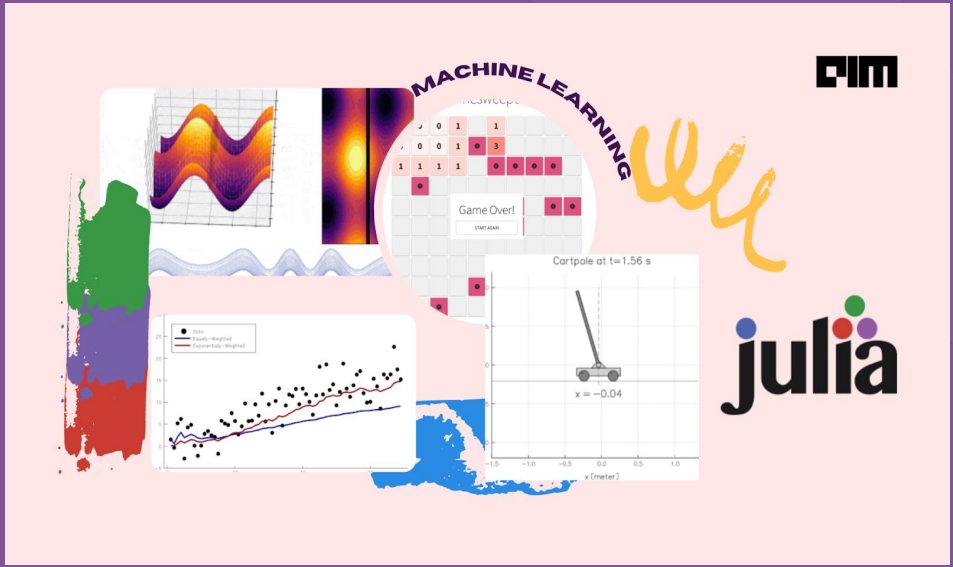
Sloka Sudhin, Althea Pamintawan, &
Zhizheng Wang

Table of Contents



Introduction

Julia is a high-level, high-performance, dynamic programming language designed specifically for technical and scientific computing. It was created to address the limitations of other programming languages, particularly when it comes to numerical and scientific computing tasks.

Why Julia?

- **High-Performance:** Julia offers C/Fortran-like performance with easy syntax. Dynamic d packages.
- **Dynamic Typing:** No need to declare variable types.
- **Multiple Dispatch:** Functions adapt based on argument types.
- **Math Libraries:** Rich libraries for math, statistics, and data analysis.
- **Computing:** Built-in support for parallel and distributed computing.
- **Open Source:** Freely available and supported by a growing community.
- **Ease of Use:** Clean syntax, similar to Python and MATLAB. Package
- **Ecosystem:** Extensible with a wide range of community-developed packages.

History

Julia started in 2009 and first released in 2012

C: late 1978, Java: 1995, Python: 1991

Julia 1.9.0 was released on 7 May 2023

Jeff Bezanson

Stefan karpinski

Viral Shah

Alan Edelman

How to install Julia

1. Download

https://julialang.org/downloads/

2. IDE

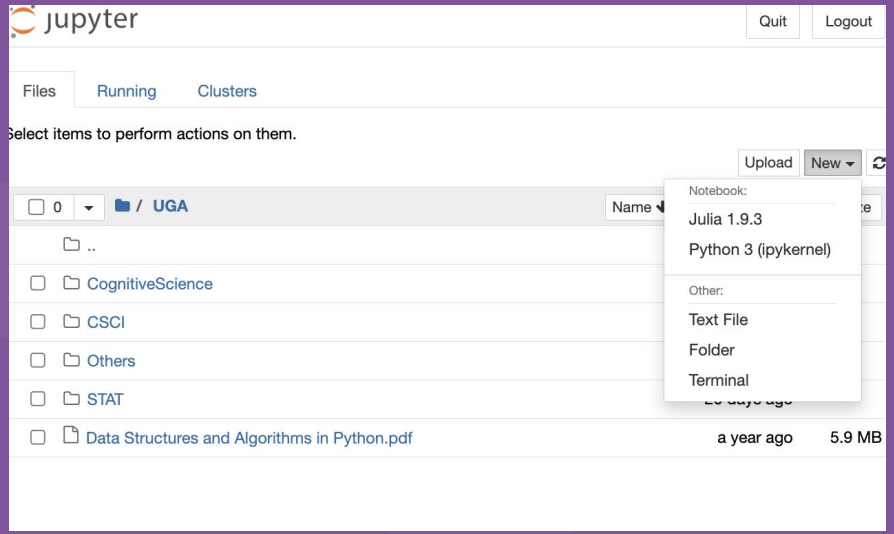
Jupyter NoteBook:

1. Open julia in terminal
2. julia> using pkg
   julia> Pkg.add("IJulia")

Visual Studio Code:

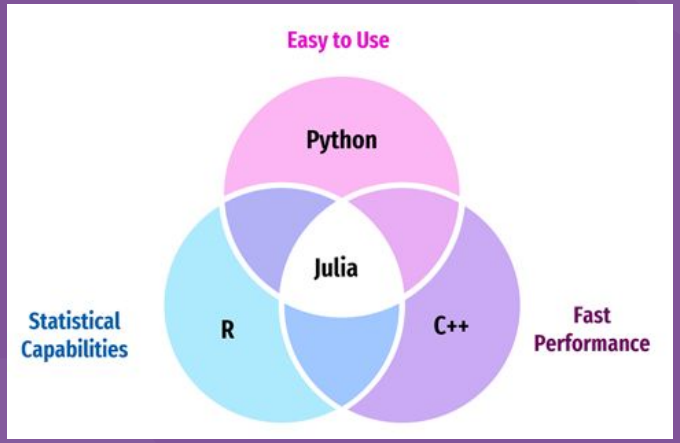
1. Install julia extension
2. _____.jl

How to install Julia



Syntax

https://docs.julialang.org/en/v1/manual/variables/




```
C:\Users\Admin>julia
               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.4.2 (2020-05-23)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> print("Hello World")
Hello World
julia>
```

Libraries

- LinearAlgebra: https://docs.julialang.org/en/v1/stdlib/LinearAlgebra/
  - Provides fundamental linear algebra operations and functionalities including: matrix operations, vector operations, eigenvalue/eigenvector calculations, etc.
- Plots.jl: https://docs.juliaplots.org/stable/install/
  - Provides a user-friendly and consistent interface for creating a wide range of high-quality plots and visualizations.
- DifferentialEquations.jl: https://juliapackages.com/p/differentialequations
  - Good for solving differential equations of various types, including ordinary differential equations (ODEs), partial differential equations (PDEs), and stochastic differential equations (SDE)
- Turing.jl:https://juliapackages.com/p/turing
  - Bayesian statistics with probabilistic inference. Great for probabilistic data science.

Importing is fairly easy, particularly via CLI.

Applied Example

- Most personal projects are uploaded as libraries onto GitHub
- Still a “newer” language, so people mostly are expanding on documentation and available libraries
- Great resource for starter datasets to play around with: https://juliabyexample.helpmanual.io/#DataFrames

- We’ll do a super simple example with the “iris” dataset (commonly used in R)
- Much of the functionality in older examples has been rewritten in recent years (lots of changes post-COVID to the language)

```julia
using CSV
using DataFrames
using Plots

# Load the iris dataset from a CSV file (assuming the file is named "iris.csv")
iris_df = CSV.File("iris.csv") |> DataFrame

# Display the first few rows of the dataset
first(iris_df, 5)

# Summary statistics of the dataset
describe(iris_df)

# Create a scatter plot of Sepal Length vs. Sepal Width
scatter(
    iris_df.SepalLength,
    iris_df.SepalWidth,
    group = iris_df.Species,
    xlabel = "Sepal Length",
    ylabel = "Sepal Width",
    title = "Iris Dataset",
    legend = true
)

# Save the plot as an image (optional)
savefig("iris_scatter_plot.png")
```

Presentation Resources

- https://docs.julialang.org/en/v1/manual/variables/
- https://code.visualstudio.com/docs/languages/julia
- https://docs.julialang.org/en/v1/stdlib/LinearAlgebra/
- https://docs.juliaplots.org/stable/install/

Thanks!