Index

October 23, 2016

1 A Deep Introduction to Julia for Data Science and Scientific Computing

1.1 Introduction

This workshop is put together by Chris Rackauckas as part of the UC Irvine Data Science Initiative. This workshop is made to teach people who are experienced with other scripting languages the relatively new language Julia. Unlike the other Data Science Initiative workshops, this workshop assumes prior knowledge of some form of programming in a language such as Python, R, or MATLAB.

We will start by introducing the basic syntax of the language, and quickly move into the details of how Julia is different from other scripting languages and how to exploit Julia's type system + multiple dispatch to be able to achieve C/Fortran-like performance while maintaining the concise syntax of a scripting language. Large parts of the package ecosystem will be explored and some information on implementation details will be highlighted in order for the participants to learn how to design Julia projects.

The ideal participant is anyone who is interested in Julia. There are many groups of people interested in using Julia. One large fraction come with a strong software development background and C/Fortran knowledge, and are looking to learn Julia as a tool to create packages with enhanced productivity while not losing performance. On the other side, there are users who are interested in the growing package ecosystem of Julia and would like to add Julia to their knowledgebase. And then there's everything in between. One major goal of this workshop is to use Julia's language and syntax to bridge the gap between "package users" and "package developers" in the way that Julia has done.

1.2 Introduction to the Author

Chris is a PhD student in Mathematics from the Mathematical, Computational, and Systems Biology (MCSB) Gateway program and is an active part of the Julia community. He runs the blog, http://www.stochasticlifestyle.com/, where he write tutorials on using the Julia languages covering many topics, such as high-performance and GPGPU computing, package development, and Julia tips. He is part of the JuliaMath and JuliaDiffEq communities for curating the Julia libraries for mathematics and differential equations respectively. Chris is the developer of many Julia packages, the most prominent being DifferentialEquations.jl, a Julia library which become a standard solver for many forms of differential equations.

1.3 Installation and Setup

To get started, see the Tooling, Documentation, and Community notebook.

1.4 Jupyter Notebooks

1.4.1 Introduction

- Tooling, Documentation, and Community
- A Mental Model for Julia
- A Very Quick Introduction to Git/Github for Julia Users

1.4.2 Basics

- A Basic Introduction to Julia
- Why Julia?

1.4.3 Detailed Usage: A Peak Into "the Rabbit Hole"

- Metaprogramming
- Call Overloading
- More Details on Arrays and Matrices - How to get Fortran Speeds in Linear Algebra
- Array and Iterator Interfaces

1.4.4 Packages to Explore

- Overview of the Package Ecosystem
- Data Visualization: Plots.jl
- Differential Equations: DifferentialEquations.jl

1.4.5 Projects / Experiments

- Type-Stability Experiment
- Scoping Experiment
- Parallelism and HPC

Extras

1.5 Slides

1.5.1 Introduction

- Tooling, Documentation, and Community
- A Mental Model for Julia
- A Very Quick Introduction to Git/Github for Julia Users

1.5.2 Basics

- An Introduction to Basic Julia
- Why Julia?

1.5.3 Detailed Usage: A Peak Into "the Rabbit Hole"

- Metaprogramming
- Call Overloading
- Array and Iterator Interfaces
- More Details on Arrays and Matrices - How to get Fortran Speeds from Linear Algebra

1.5.4 Packages to Explore

- Overview of the Package Ecosystem
- An In-Depth Look at Plots.jl

1.5.5 Projects / Experiments

- Type-Stability Experiment
- Scoping Experiment
- Parallelism and HPC

1.6 How These Were Made

This entire repository is made using Jupyter notebooks using the template from the JupyterSite repository.