Julia/IJulia/JuMP Installation Guide

Contacts: kqureshi@mit.edu

This setup guide is for students who want to install Julia/IJulia/JuMP in their local computer and run large-scale optimization problems (possibly for projects). We recommend install JuliaBox for students to complete problem sets. Please see the other setup guide for JuliaBox.

I. Purpose

We will use Julia to model and solve several different types of problems involving linear programs, integer programs, nonlinear programs.

Here we set up

1. Julia, the programming language
2. IJulia, an environment for writing Julia code
3. JuMP, a Julia package for modeling optimization problems

II. Install Julia

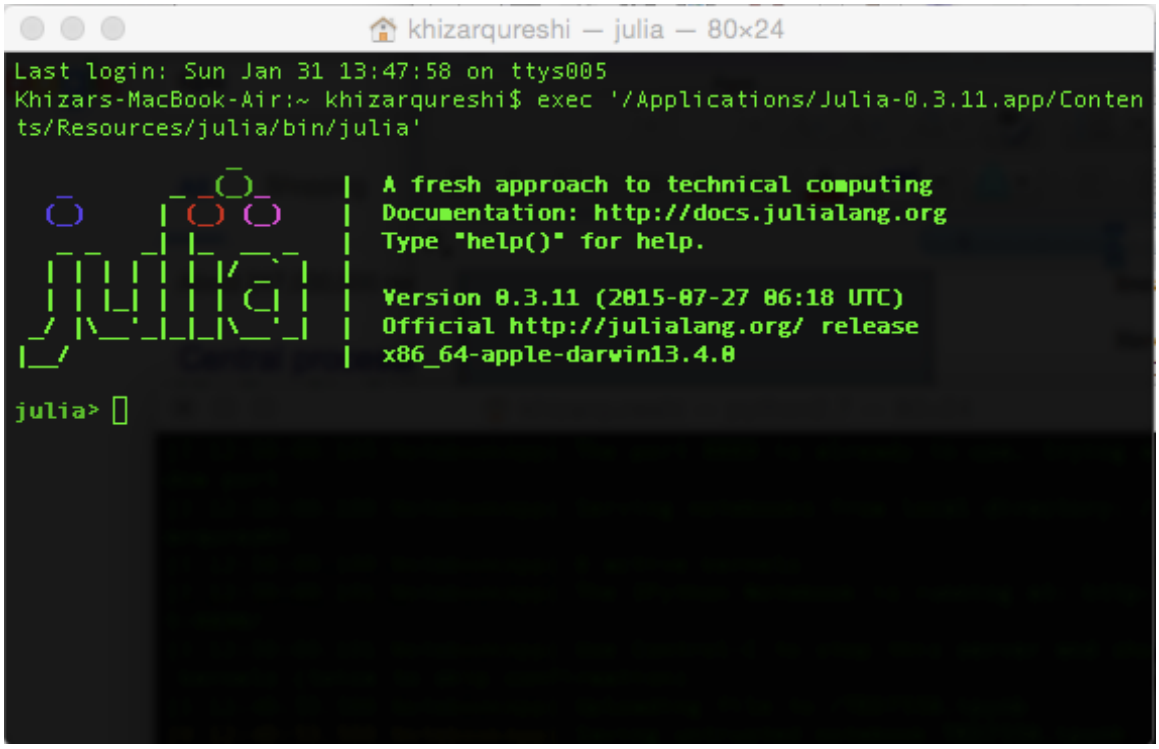
1. Download **Julia v0.4.0** from the official downloads page, http://julialang.org/downloads/

Windows: Go to system information under the start menu to see whether your computer runs 64-bit or 32-bit. Download the corresponding version

MacOS: Julia requires OS X 10.7 or higher

Please ensure that Julia 0.4.0 is installed. To confirm, launch Julia, and the version should be listed above the prompt.

The image below shows Julia 0.3.11 installed, which is incorrect.



III. Install IJulia

1. Install Anaconda Python from http://continuum.io/downloads
    a. Check 'add anaconda to the system path' in the installation window
    b. Check 'register anaconda as default' in the installation window
2. Open Julia
    a. Windows: using the command window or start menu
    b. MacOS: using terminal.app or applications
3. Wait for the kernel to start, as indicated by Julia>
4. Enter and run Pkg.add("IJulia")
5. Enter and run the following to test installation
    using IJulia
    IJulia.notebook()
6. *If the above fails, run the following in Julia*
    Pkg.checkout("IJulia")
    Pkg.build("IJulia")
7. *If the above fails, run the following in terminal*
    pip install jupyter
    jupyter notebook

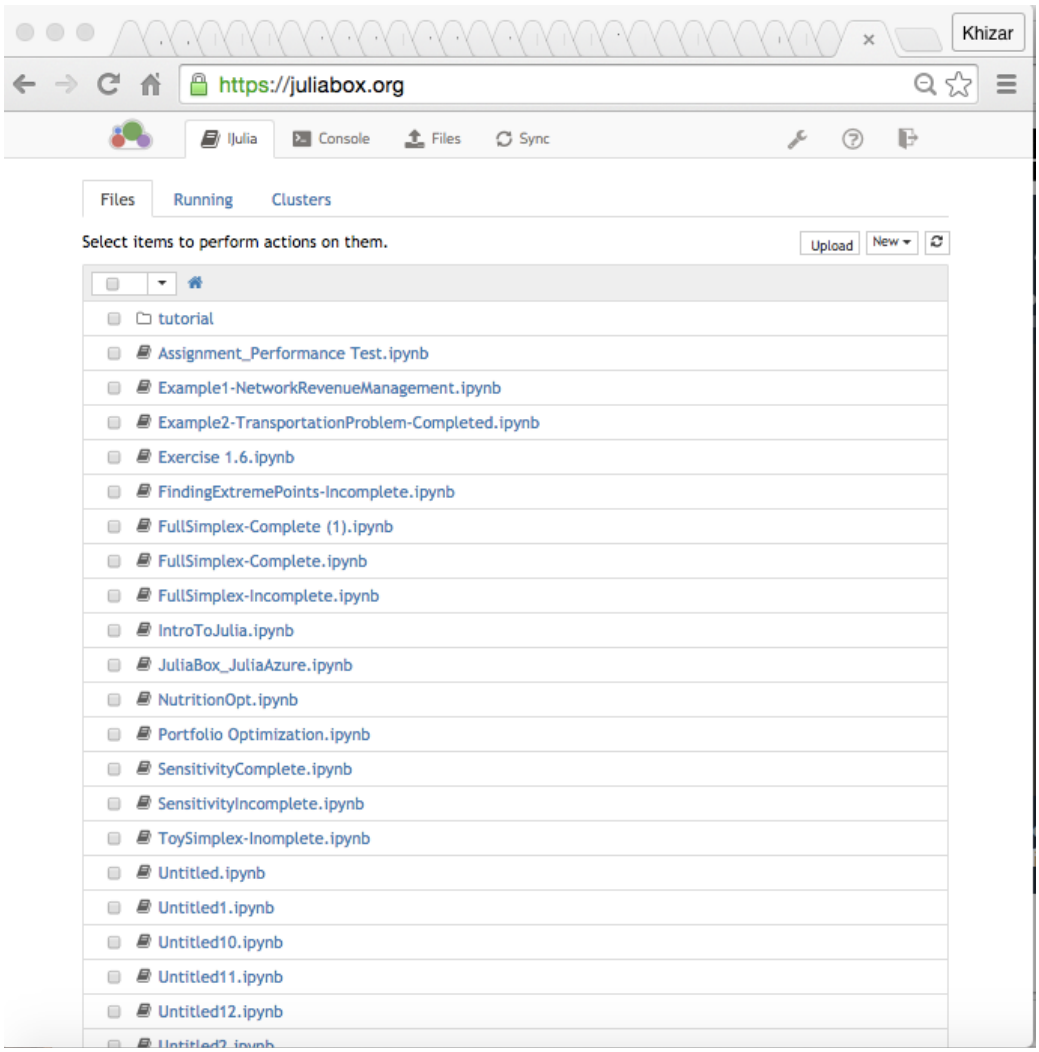
IV. JuliaBox

JuliaBox is run on servers donated by Amazon web services (AWS). It has similar features as a local Julia 0.4.0, but with limitations on directory access and run time.

1. Sign up for an account at https://juliabox.org/
2. To install packages, repeat the steps for local Julia above

JuliaBox has an interface very similar to Julia. However, store accessible files, we must use the upload function, rather than changing into a working directory.

JuliaBox performance is dependent on an AWS, rather than CPU, so run time is longer. We will see this for larger scale problems.



V. Adding Julia packages

We will use Julia packages for (i) creating a model, (ii) solving a model, and (iii) data management.

1. Open Julia, and enter the following
   ```
   Pkg.add("JuMP")
   Pkg.add("Clp")
   Pkg.add("Cbc")
   Pkg.add("Ipopt")
   Pkg.add("DataFrames")
   ```

We will use JuMP to model both linear and nonlinear programs. We will use Cbc and Clp to solve linear programs. We will use Ipopt to solve nonlinear programs. We will use DataFrames for data management.

VI. References

1. Julia: http://Julialang.org
2. JuMP: http://github.com/JuliaOpt/JuMP.jl