Installation Instructions

January 9, 2016

1 Install Julia

1. Download julia-0.4.2 from `http://julialang.org/downloads/`.
2. Open up a julia terminal.
3. Install several packages using the following code, line by line:

   Pkg.add("JuMP")
   Pkg.add("DataFrames")
   Pkg.add("Gadfly")
   Pkg.add("Cairo")

4. Make sure the following code works:

   using JuMP
   using DataFrames
   using Gadfly
   using Cairo

5. At this point, you have the Julia programming language successfully installed on your computer! In the terminal prompt, you can experiment by typing in some math and then "enter". For example, you can try: "2+2", "3^2", and "12*10".

2 Install Sublime Text 2

1. Download Sublime Text 2 from `http://www.sublimetext.com/2`.

2. Go to the webpage `https://packagecontrol.io/installation#Manual` and copy the code block under "Sublime Text 2".

3. Open up Sublime Text 2 and click on "View → Show Console".

4. Paste the code block and hit "enter".

5. Restart Sublime Text 2.

6. Press "ctrl + shift + p" (Windows) or "cmd + shift + p" (Mac) to open up a search bar. Begin typing "Install Package" and click on "Package Control: Install Package".

7. Begin typing "Julia" and click on "Julia". The package for Julia syntax highlighting will automatically install.

8. Open up "jump_start.jl". Click on "View → Syntax → Julia" to view the code with syntax highlighting.

3 Set your Julia working directory

1. Create a new folder on your desktop where you can save all of your Julia programs. Name it something simple like "optimization".

2. Create a new file in Sublime Text 2 named ".juliarc.jl".

3. In this file, type: cd("*path_here*"). In place of *path_here*, put the file path on the computer to the "optimization" folder. For example, it might be:

   cd ("C:\\Users\\colin\\Documents\\optimization").

4. Open up a new Julia terminal and type "homedir()" to determine the home directory folder.

5. Save ".juliarc.jl" in the home directory folder, which is the output of the "homedir()" command run in the previous step.

6. To test that the working directory is set up correctly, open up a new Julia terminal and type in **;** and then **pwd**. Check that the folders match up.

4 Command line tips

In the Julia terminal, use **;** to switch to shell commands, and **backspace** to switch back[1]. These commands allow you to navigate to different directories in the computer. Here are a few of the most useful ones:

ls lists the files in the current directory.

cd stands for "change directory". **cd** *folder* opens up *folder* in the present directory, and **cd ..** goes up one level.

pwd stands for "pass working directory". This tells you which directory you are currently in.

Test out these commands below.

1. (Optional) Open up the Julia terminal, and type in **;**, then **ls**. The files in your current folder, "optimization", should pop up.
2. (Optional) Pop out of the folder by typing in **;**, then **cd ..**.
3. (Optional) Type in **;**, then **ls** to see all of the folders in that directory.
4. (Optional) Return to the directory by typing in **;**, then **cd optimization**.

To run a julia file, use the command **include("filename.jl")**. You can use this command as long as "filename.jl" is in the current directory.

5 Julia tips

1. (Optional) Visit the webpage `http://docs.julialang.org/en/release-0.4/` for an introduction to the Julia programming language.
2. (Optional) For a condensed list of many useful Julia commands, see "Julia-cheatsheet.pdf".
3. (Optional) When you are typing in code in the Julia terminal, you can use "tab" to auto-complete certain phrases. This is especially useful for shell commands.

[1]These are also called "command line" or "BaSH" commands.

4. **Experiment and have fun!** You learn programming languages by trying things out yourself, looking stuff up on the internet, and writing your own code.