**Workshop: “Scientific computing with Julia”
installation instructions**

© 2020 Przemysław Szufel under the terms of MIT License

**What is Julia**

Julia is a new Open Source language designed for science and data analysis. With the stable 1.0 version released in August 2018, an exponential growth of language popularity has been observed and the language is in the top-20 programming languages in IEEE Spectrum ranking and top-10 programming languages developed on GitHub. Julia takes “walks like Python, runs like C” approach and is a perfect replacement for Matlab, Python and R scientific data workflow, yet due to its speed it can be also used to implement computation intensive algorithms that are normally implemented in languages such as Java or C++. Nowadays, the most popular Julia applications include computations related to data science, machine learning, numerical simulation, quantitative economics, applied mathematics, physics, astronomy, chemistry, and bioinformatics.

**Installation instructions**

1. Please download Julia from <https://julialang.org/downloads/> and follow the installation instructions presented at <https://julialang.org/downloads/platform/>. During the workshop we will be using the Stable Release **v1.5.1**. The 64-bit version is recommended.
2. Install Julia packages that will be used throughout the workshop. Once Julia is installed run the Julia console or in the command line (shell console) type **julia**. Once Julia interpreter is running paste the following Julia code (this code can be also copy-pasted to a workshopinstall.jl file and run by command julia workshopinstall.jl:

**using** Pkg

Pkg.add("Agents")

Pkg.add("BandedMatrices")

Pkg.add("BenchmarkTools")

Pkg.add("Cbc")

Pkg.add("Conda")

Pkg.add("CSV")

Pkg.add("DataFrames")

Pkg.add("DataStructures")

Pkg.add("Distributions")

Pkg.add("ForwardDiff")

Pkg.add("FredData")

Pkg.add("FreqTables")

Pkg.add("GLM")

Pkg.add("GLPK")

Pkg.add("GR")

Pkg.add("GraphPlot")

Pkg.add("IJulia")

Pkg.add("Ipopt")

Pkg.add("InteractiveUtils")

Pkg.add("JuMP")

Pkg.add("Juno")

Pkg.add("LightGraphs")

Pkg.add("Logging")

Pkg.add("OpenStreetMapX")

Pkg.add("Optim")

Pkg.add("Parameters")

Pkg.add("PGFPlots")

Pkg.add("Plots")

Pkg.add("PyCall")

Pkg.add("PyPlot")

Pkg.add("ScikitLearn")

Pkg.add("SimpleHypergraphs")

Pkg.add("SparseArrays")

Pkg.add("StatsBase")

Pkg.add("XLSX")

using Conda

Conda.runconda(`install jupyter --yes`)

Conda.runconda(`install matplotlib --yes`)

Conda.runconda(`install folium -c conda-forge`)

[OPTIONAL] Additionally, only if you have Gnu R installed and want to see Julia-R integration:

Pkg.add("RCall")

[OPTIONAL] Additionally, only if you have an NVIDIA GPU hardware installed along with all system drivers and want to see how to do GPU computing with Julia:

Pkg.add("CUDAdrv")

Pkg.add("CUDAnative")

Pkg.add("GPUArrays")

Pkg.add("CUDAapi")

Pkg.add("Flux")

1. Once all packages are installed, while still in the Julia console run the command below to precompile all the packages that you have just installed:

using Pkg

pkg"precompile"

1. The recommended programming environment for the Julia language is Visual Studio Code (<https://code.visualstudio.com/>) with Julia extension. Please follow the steps below:
	1. Download and install VS Code (available at <https://code.visualstudio.com/download/>)
	2. Start VS Code, click View->Command Palette... and type View: Show Extensions to go to the extension manager
	3. In the extension manager search box type “Julia”
	4. On the top of the extension list you should see “Julia Language Support” – click *Install* to install the extension.

If you run into any installation problem, more detailed instructions can be found in <https://github.com/julia-vscode/julia-vscode#getting-started>

1. During the workshop will be mostly working with Julia within Jupyter notebook (this can be also used instead of VS Code)

In order to be able to run Julia inside a Jupyter notebook start the Julia console and run the two following commands:

**using** IJulia

notebook(dir=pwd()) # a new web browser tab should open with Jupyter Notebook

 # showing current Julia working directory

**Literature**

1. Julia Language Manual - <https://docs.julialang.org/en/v1/>
2. The Julia Express - A concise Julia language introductory manual for programmers, <https://github.com/bkamins/The-Julia-Express/>
3. B. Kaminski, P. Szufel, "Julia 1.0 Programming Cookbook – Over 100 numerical and distributed computing recipes for your daily data science workflow", Packt Publishing, 2018