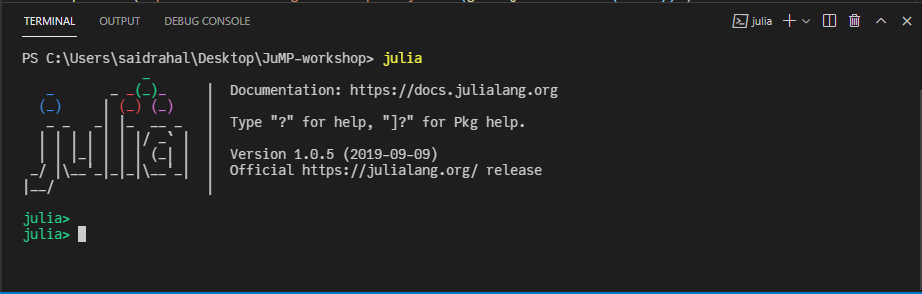
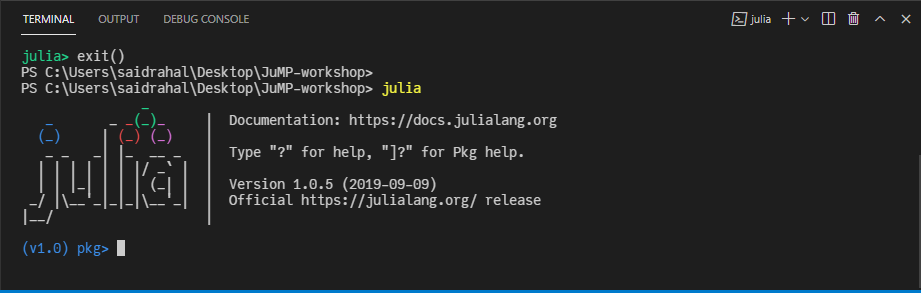
# Note on installing Julia, JuMP, JuMPeR and other Packages

I have written down the steps that will take through the installation of Julia, JuMP and JuMPeR. The main point to note is the compatible versions of each entity.

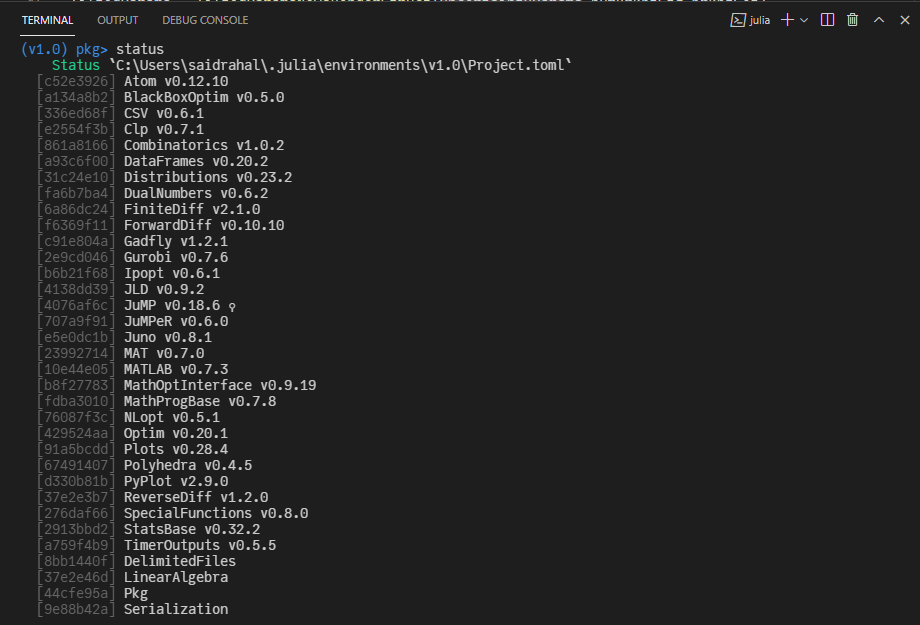
1. Download long-term support Julia Version 1.0.5 using this [link](https://julialang.org/downloads/#long_term_support_release).
2. I use visual studio code as a [Julia editor](https://www.julia-vscode.org/). Previously, the Julia team provided support for using atom as an editor via the [Juno](https://junolab.org/) extension.
3. In the vs code terminal, I call julia by simply typing the name:



1. Use the command exit() to quit.
2. To manage the installed packages (packages are analogous to modules in python), input the **“]”**:



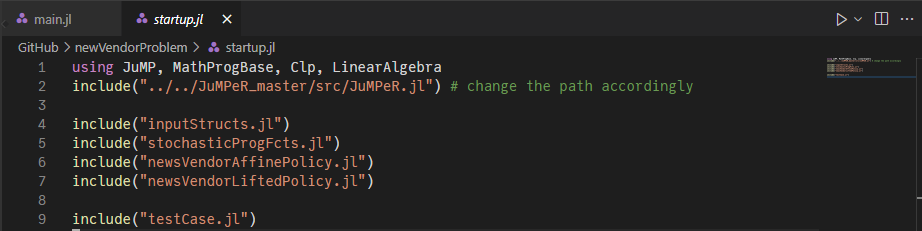
1. The current version of **JuMPeR is v0.6.0** and is compatible with **JuMP v0.18.6**.
2. To install JuMP, type: **add JuMP@0.18.6**.
3. To fix the version of JuMP, type: **pin JuMP**.
4. To install JuMPeR, type: **add JuMPeR**.Note that we do not need to specify the version because we want to install the latest version.
5. The command **status**, lists all the installed packages as in the screenshot below. Note that you will not need all the listed packages. These are the packages which I have used throughout my PhD.



1. The packages you will mostly need are:
   1. A solver (i.e., Clp or Ipopt or Gurobi). If you use CPLEX, you can install it too. I used Clp solver for the newsvendor problem
   2. DataFrames, StatsBase and Linear Algebra packages provide helpful features.
   3. Plots or Pyplot for plotting purposes.
2. The command **update,** updates your listed packages.Personally, I avoided using it because some updated packages other than JuMP may not be compatible with Julia v1.0.5.
3. You remove an installed package using the command **remove.**

# Note on running Julia functions and the Newsvendor Problem

1. To import a Julia package, we use the command **using.** Once you import a package in julia REPL, you can not import again. Julia will throw an error if you so. Hence I include this commands in a separate startup.jl script to compile it once at the start of the Julia REPL.



1. You will see that I compiled JuMPeR using the **include(“”)** command rather than the **using** command. This is because I have slightly modified the JuMPeR source code locally and I need to use the modified version.
2. Speaking about the **include(“”)** command, it is used to compile Julia scripts.
3. In the shared git repository, I included a Code map that will help you navigate the components of each julia script.
4. You can run the code by executing **julia main.jl** from the powershell.