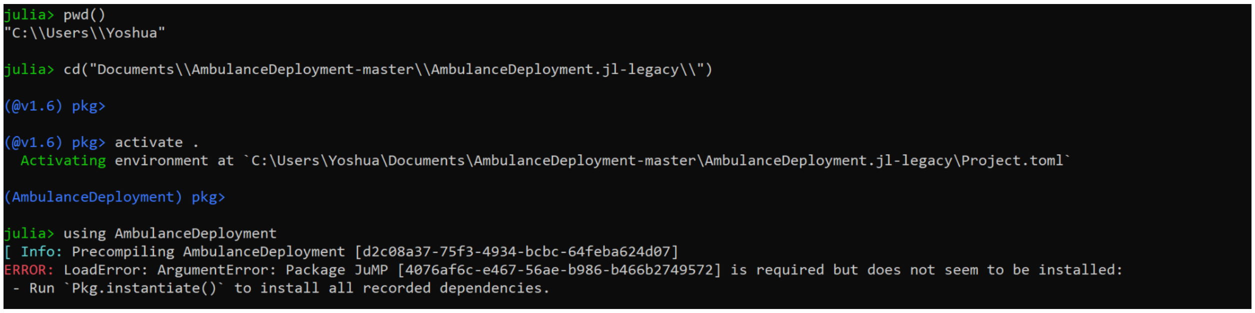
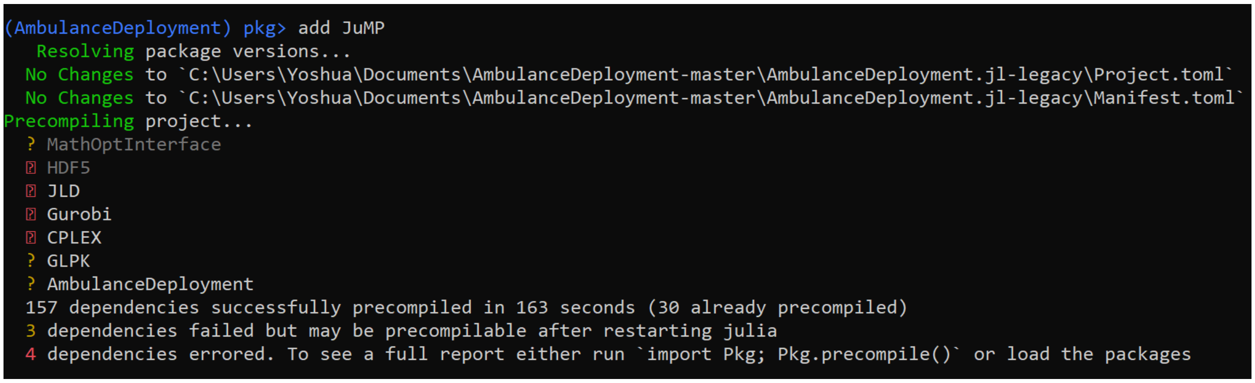
For the first time running Julia let’s make sure 1) we’re in the right directory (AmbulanceDeployment.jl-legacy) 2) we can run all of the packages. To get the terminal to switch to package/ turn blue press “]”.



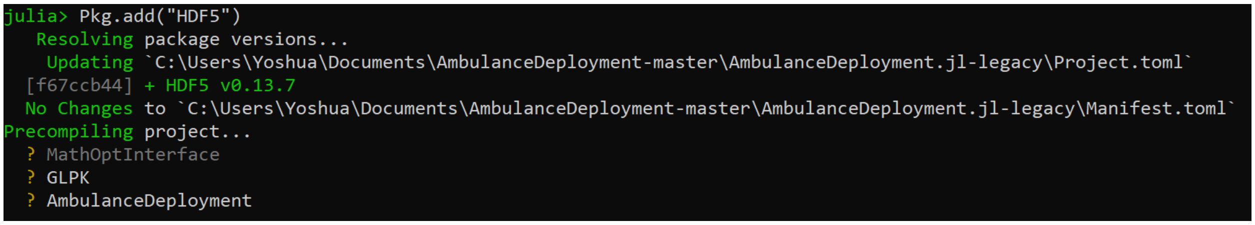
```
julia> pwd()
"C:\\Users\\Yoshua"

julia> cd("Documents\\AmbulanceDeployment-master\\AmbulanceDeployment.jl-legacy\\")

(@v1.6) pkg>

(@v1.6) pkg> activate .
  Activating environment at `C:\Users\Yoshua\Documents\AmbulanceDeployment-master\AmbulanceDeployment.jl-legacy\Project.toml`

(AmbulanceDeployment) pkg>

julia> using AmbulanceDeployment
[ Info: Precompiling AmbulanceDeployment [d2c08a37-75f3-4934-bcbc-64feba624d07]
ERROR: LoadError: ArgumentError: Package JuMP [4076af6c-e467-56ae-b986-b466b2749572] is required but does not seem to be installed:
 - Run `Pkg.instantiate()` to install all recorded dependencies.
```

Since I don’t have Jump I need to add it like so: (make sure you are adding it to the ambulance deployment package).



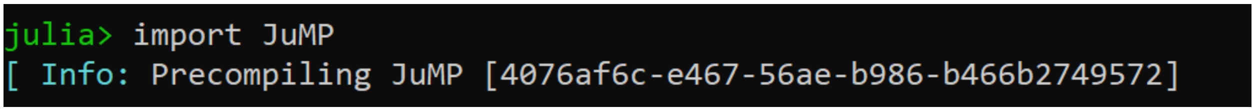
```
(AmbulanceDeployment) pkg> add JuMP
   Resolving package versions...
  No Changes to `C:\Users\Yoshua\Documents\AmbulanceDeployment-master\AmbulanceDeployment.jl-legacy\Project.toml`
  No Changes to `C:\Users\Yoshua\Documents\AmbulanceDeployment-master\AmbulanceDeployment.jl-legacy\Manifest.toml`
Precompiling project...
  ? MathOptInterface
  ⍰ HDF5
  ⍰ JLD
  ⍰ Gurobi
  ⍰ CPLEX
  ? GLPK
  ? AmbulanceDeployment
  157 dependencies successfully precompiled in 163 seconds (30 already precompiled)
  3 dependencies failed but may be precompilable after restarting julia
  4 dependencies errored. To see a full report either run `import Pkg; Pkg.precompile()` or load the packages
```

For the 4 dependencies that won’t install I just had to manually add them.



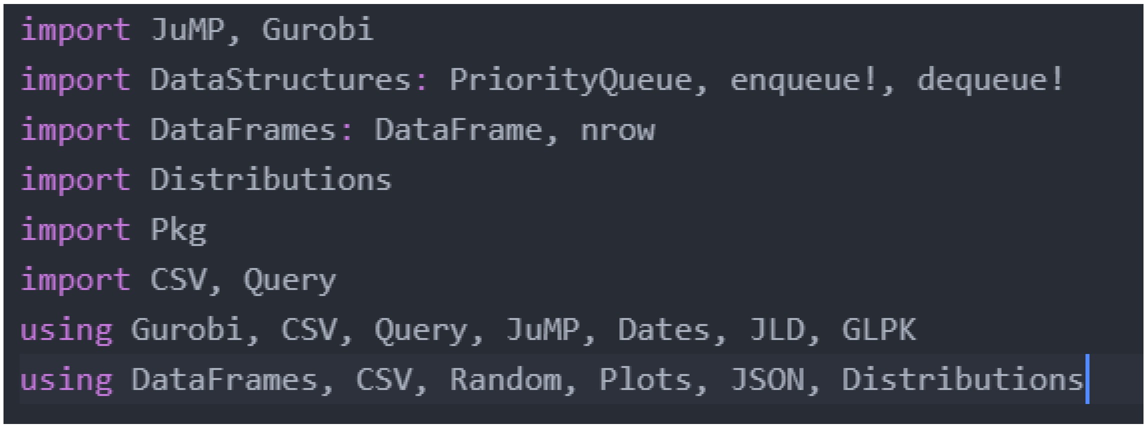
```
julia> Pkg.add("HDF5")
   Resolving package versions...
    Updating `C:\Users\Yoshua\Documents\AmbulanceDeployment-master\AmbulanceDeployment.jl-legacy\Project.toml`
  [f67ccb44] + HDF5 v0.13.7
  No Changes to `C:\Users\Yoshua\Documents\AmbulanceDeployment-master\AmbulanceDeployment.jl-legacy\Manifest.toml`
Precompiling project...
  ? MathOptInterface
  ? GLPK
  ? AmbulanceDeployment
```

Now I can write:



```
julia> import JuMP
[ Info: Precompiling JuMP [4076af6c-e467-56ae-b986-b466b2749572]
```

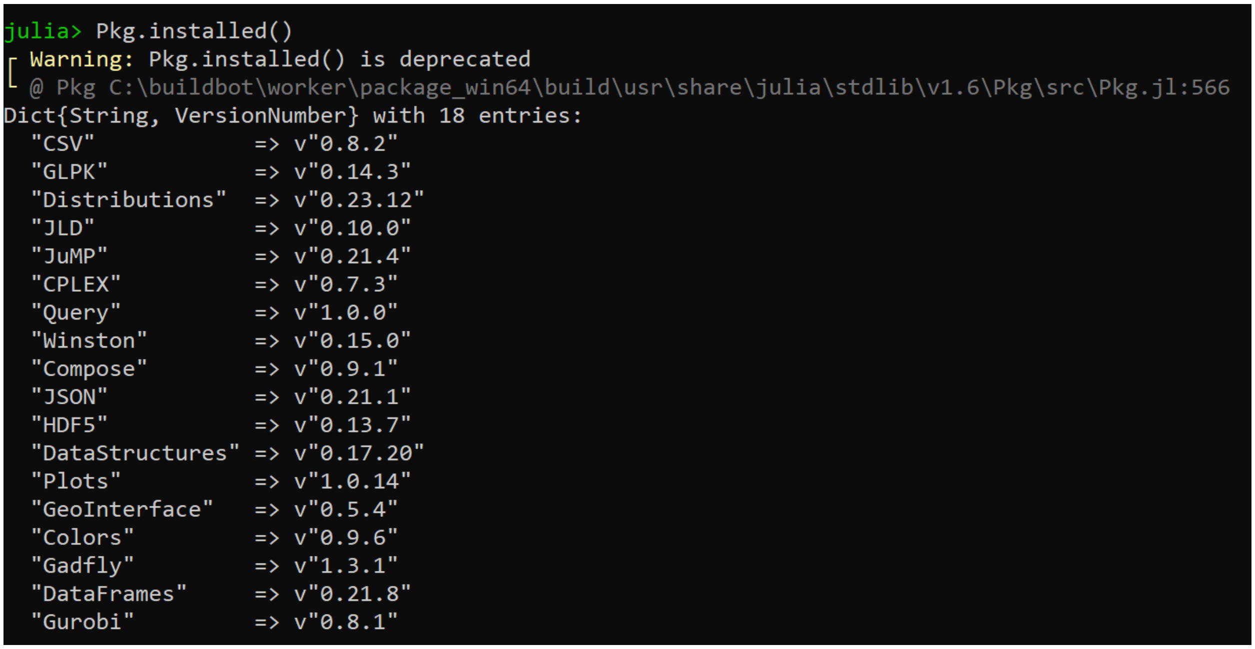
You need to do this once for all new libraries. Here is a list of all of them:



```
import JuMP, Gurobi
import DataStructures: PriorityQueue, enqueue!, dequeue!
import DataFrames: DataFrame, nrow
import Distributions
import Pkg
import CSV, Query
using Gurobi, CSV, Query, JuMP, Dates, JLD, GLPK
using DataFrames, CSV, Random, Plots, JSON, Distributions
```

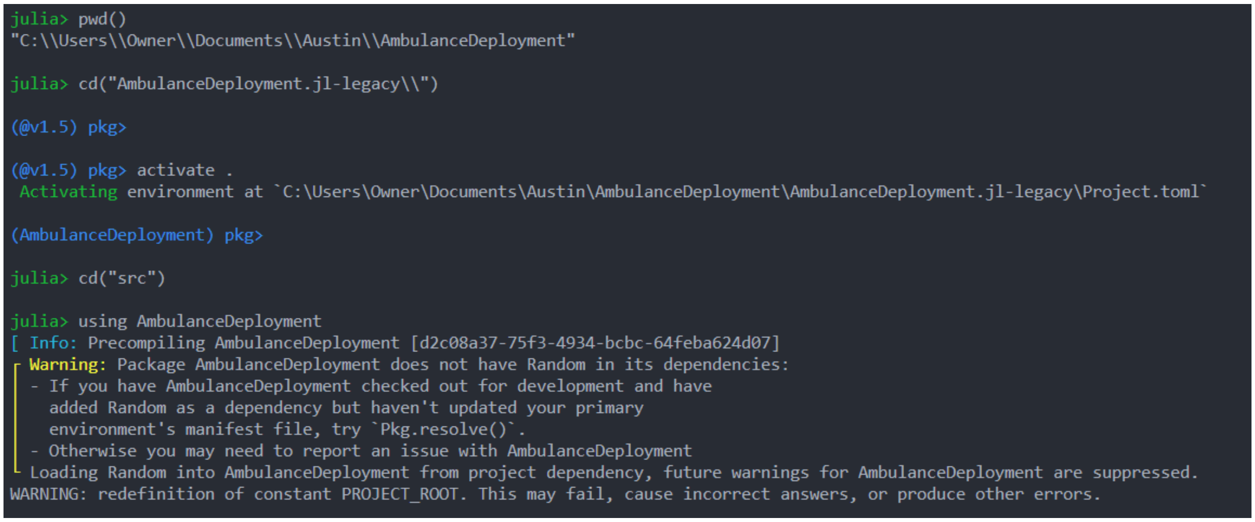
import JuMP, Gurobi
import DataStructures: PriorityQueue, enqueue!, dequeue!
import DataFrames: DataFrame, nrow
import Distributions
import Pkg
import CSV, Query
using Gurobi, CSV, Query, JuMP, Dates, JLD, GLPK
using DataFrames, CSV, Random, Plots, JSON, Distributions

Once you are all done you should be able to run Pkg.installed() and see: (Note your base Pkg and AmbulanceDeployment Pkg can have different libraries)

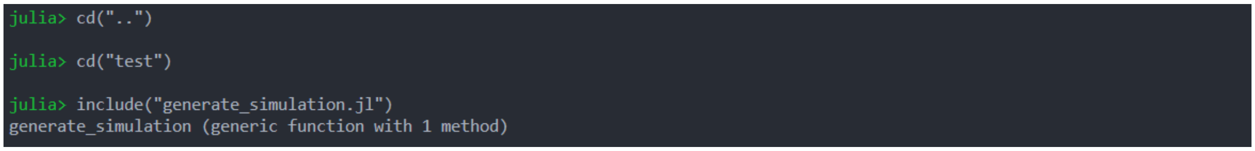


```
julia> Pkg.installed()
┌ Warning: Pkg.installed() is deprecated
└ @ Pkg C:\buildbot\worker\package_win64\build\usr\share\julia\stdlib\v1.6\Pkg\src\Pkg.jl:566
Dict{String, VersionNumber} with 18 entries:
  "CSV"            => v"0.8.2"
  "GLPK"           => v"0.14.3"
  "Distributions"  => v"0.23.12"
  "JLD"            => v"0.10.0"
  "JuMP"           => v"0.21.4"
  "CPLEX"          => v"0.7.3"
  "Query"          => v"1.0.0"
  "Winston"        => v"0.15.0"
  "Compose"        => v"0.9.1"
  "JSON"           => v"0.21.1"
  "HDF5"           => v"0.13.7"
  "DataStructures" => v"0.17.20"
  "Plots"          => v"1.0.14"
  "GeoInterface"   => v"0.5.4"
  "Colors"         => v"0.9.6"
  "Gadfly"         => v"1.3.1"
  "DataFrames"     => v"0.21.8"
  "Gurobi"         => v"0.8.1"
```

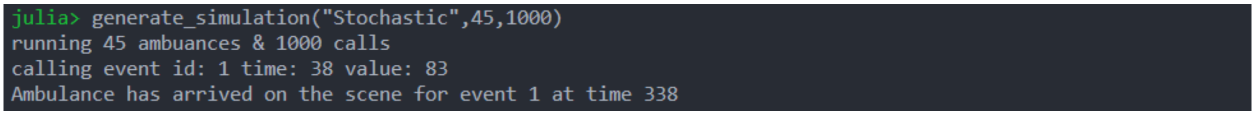
Running generate_simulations.jl once all of the packages have been installed.



```
julia> pwd()
"C:\\Users\\Owner\\Documents\\Austin\\AmbulanceDeployment"

julia> cd("AmbulanceDeployment.jl-legacy\\")

(@v1.5) pkg>

(@v1.5) pkg> activate .
 Activating environment at `C:\Users\Owner\Documents\Austin\AmbulanceDeployment\AmbulanceDeployment.jl-legacy\Project.toml`

(AmbulanceDeployment) pkg>

julia> cd("src")

julia> using AmbulanceDeployment
[ Info: Precompiling AmbulanceDeployment [d2c08a37-75f3-4934-bcbc-64feba624d07]
┌ Warning: Package AmbulanceDeployment does not have Random in its dependencies:
│ - If you have AmbulanceDeployment checked out for development and have
│   added Random as a dependency but haven't updated your primary
│   environment's manifest file, try `Pkg.resolve()`.
│ - Otherwise you may need to report an issue with AmbulanceDeployment
└ Loading Random into AmbulanceDeployment from project dependency, future warnings for AmbulanceDeployment are suppressed.
WARNING: redefinition of constant PROJECT_ROOT. This may fail, cause incorrect answers, or produce other errors.
```



```
julia> cd("..")

julia> cd("test")

julia> include("generate_simulation.jl")
generate_simulation (generic function with 1 method)
```



```
julia> generate_simulation("Stochastic",45,1000)
running 45 ambuances & 1000 calls
calling event id: 1 time: 38 value: 83
Ambulance has arrived on the scene for event 1 at time 338
```