Extremes in Julia

Neal Grantham and Sam Morris

April 21, 2015

Why Julia:

- It's fast
- Open source
  - Transparent and available through Github
- Principal aim is to combine the best elements of R, Python, Matlab, and C at very little cost
  - Easy to learn

Getting started:

- Julia: Command Prompt (http://julialang.org/)
- IDEs:
  - Juno (http://junolab.org)
  - iJulia: iPython interface for Julia
- JuliaOpt: Mathematical Optimization (https://github.com/JuliaOpt/)
- JuliaStats: Statistics and Machine Learning (https://github.com/JuliaStats)
- Curated index of Julia packages by discipline (https://github.com/svaksha/Julia.jl)

Differences from R:

- Julia relies heavily on types
  - Most speedup comes from typing functions
- Easy to create new types
- Minor syntax differences (more similar to Matlab)
- Functions are JIT compiled at first run
- Pass by reference (allows for easier modification in place, better memory management)
- No penalty for for loops
- Published packages listed in `METADATA`

Package installation and usage in Julia

- To install a package from `METADATA`, use the method `Pkg.add()`
  - Distributions: `Pkg.add("Distributions")`
  - RDatasets: `Pkg.add("RDatasets")`
  - Gadfly: `Pkg.add("Gadfly")`
- Can also clone packages from other Github repositories using `Pkg.clone()`
  - `Pkg.clone("https://github.com/sammorris81/ExtremeValueDistributions.jl.git")`
- To update packages, use the method `Pkg.update()`
- To use a package in your code, simply include `using` followed by the package name
  - `using Distributions`

Distributions in Julia:

- They are a type with a common set of methods
- Abstract type:
  - `ContinuousUnivariateDistribution`
  - `DiscreteUnivariateDistribution`
  - `ContinuousMultivariateDistribution`
  - `DiscreteMultivariateDistribution`
- Many standard distributions come with `Distributions.jl`
- Let `d = Normal(0, 1)`
  - `mean(d)`
  - `var(d)`
  - `params(d)`
  - `pdf(d, x)`
  - `rand(d, n)`
  - `fit_mle(Normal, x)`
- Many distributions also include types for sufficient statistics
  - Streamlines parameter estimation and computation

Our main contribution:

- Generalized extreme value distribution
  - `GeneralizedExtremeValue <: ContinuousUnivariateDistribution`
- Generalized Pareto distribution
  - `GeneralizedPareto <: ContinuousUnivariateDistribution`
- Parameter estimation via:
  - MLE
  - MCMC - Adaptive Random Walk Metropolis Hastings