Legend: Each bullet point is roughly a lecture (slides, in person lecture, notebooks with code examples and (if desired) homework exercise

- Julia basics (**2-5 days of prep**, depending on depth and audience skill level)
    - Syntax (~1-2 hours)
    - Weird things about Julia if you come from R/Python/Stata/etc. (~1 hour)
    - Popular packages
        - FixedEffectModels.jl, GLM.jl, and StatsModels.jl
        - SciML universe
        - Turing
        - DataFrames.jl
        - CSV/various IO packages
        - Plotting
        - Symbolics and ModelingToolkit
        - Graph theory (the whole Julia graph ecosystem)
- Parallelization for economists (**~3-5 days of prep**)
    - Multithreading/multiprocessing
    - Thread safety
    - Commonly parallelized problems (bootstrapping, monte carlo, etc.)
    - GPU computation
- Optimization & automatic differentiation (**~1 week of prep**)
    - Common packages: Optim.jl, JuMP.jl, and GalacticOptim.jl
    - Automatic differentiation. ForwardDiff.jl, ReverseDiff.jl, and Zygote.jl (or Diffractor.jl if it is advanced enough)
    - Rolling your own GMM estimator, making it fast with AD, adding constraints.
    - Nested optimization
    - Dynamic programming
- Programming & Julia best practices (could be rolled into Julia basics or into the parallelism bucket) (**~1 day of prep**)
    - Git
    - Package management
    - Profiling
    - Using the type system to your advantage
    - Performance tips

  - In-place operations and allocations

- Bayesian inference and probabilistic programming (**3-5 days prep**)
  - Bayes introduction (needed?)
  - What is a probabilistic programming language?
  - Simulating and estimating models
  - Bayesian time series econometrics (Bayesian ARIMA variants or perhaps something like stochastic volatility for the financy folks)
  - Bayesian differential equations
  - Bayesian DSGE models (depends on progress of Cameron’s paper with Jesse Perla)