Key idea here is to structure the code so that it is as easy as possible to plug in new models. Thus everything that is generic is written as a function. Data, inits, MCMC parameters will differ but everything else should stay the same.

1. Startup workspace, loading libraries and global variables.
2. For each model, run\_model.R:
   1. Fit empirical data (fit.empirical())
      1. Load empirical data
      2. Run long chains w/ thinning to get independent samples (verify.models)
      3. Ensure platforms are equivalent and make plots (plot.model.comparisons)
      4. Run across range of platforms and tuning parameters (run.chains)
      5. Save meta data and make interim plots locally
   2. Fit simulated data; for each model size:
      1. Generate data and inits into global workspace using seed
      2. Run JAGS and NUTS with defaults only (run.chains)
      3. Save meta results to folder
      4. Make interim plots locally

Function run.chains is used for empirical and simulated data, and should:

* input:
  + Path to models; their data, inits, and output parameters
  + Vector of seeds to run across
  + Vectors of tuning parameters (delta, lambda, metric)
  + MCMC arguments (warmup, thinning, etc.)
  + Control arguments for printing
* Output list consisting of
  + (1) Adaptation information from Stan (adapt.list)
    - Meta data about run, and get\_sampler\_params returns – different for HMC and NUTS.
  + (2) Performance information from all platforms (perf.list)
    - Model name, platform, seed, Npar, Nsims, time.warmup, time.sampling, time.total, minESS, which.minESS, minESS.coda, final adapted eps, target adapt\_delta, pct.diverged, medianRhat,

Function Plot.model.results should

* Input perf and adapt lists
* Output: Write plots to local directory
  + Compare performance metrics
  + Check adaptation of eps
  + Check NUTS metrics

Function Plot.model.checks should

* Input long chains from each software platform, dims of grids (varies with Npar)
* Print grids of qqplots to local directory. Print ESS for each software and parameters

Naming conventions:

* Niter=2\*(Nout\*Nthin), Nwarmup is default where possible, Nthin is 1 where possible