1. Startup workspace, loading libraries and global variables.
2. For each model, run\_model.R:
   1. Fit empirical data
      1. Load empirical data
      2. Load models into workspace
      3. Run long chains w/ thinning to get independent samples
      4. Ensure platforms are equivalent and make plots
      5. Run across range of platforms and tuning parameters (run.chains)
      6. Save meta data locally and make interim plots
   2. Fit simulated data; for each model size:
      1. Define model variables and MCMC arguments
      2. Generate data
      3. Load models into workspace
      4. Run across range of platforms and tuning parameters (run.chains)
      5. Save meta results to local folder
      6. Make interim plots

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

* input:
  + Compiled models with their data, inits, and output parameters
  + Vector of seeds to run across
  + 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, Nwarmup, Nthin