

The course covers both analytical exercises (e.g., deriving the posterior distribution in some early models) and computational exercises.

For computation there are two different approaches to Bayesian computing, one based on macro languages (like R or Matlab) and the other based on specialized packages (like BUGS, JAGS, Stan, NIMBLE). We will definitely use the former (and try to introduce the latter).

- 1) Higher level macro language – Examples include R and Matlab. I assume there are others that people may know/prefer. Note these are not statistics packages just mathematical environments. The capabilities that I am interested in are: matrix operations, random number generation, and the ability to write and save functions/macros.

R is the official language of the course -- this means that I will use R in examples and happily answer questions about R coding. You can use any other language you like as long as you don't expect help from me. If you have access to a machine where you can download software and don't have any of the above packages, then you can easily download R for use in the class. It is available at <http://cran.r-project.org>. The easiest way to proceed is to go to Precompiled Binary Distributions and then choose the appropriate platform from among Windows (choose "base"), Linux, MacOS and MacOS X. Many people find it especially helpful to run R through the RStudio environment (it provides a convenient window system for coding, executing code, plots, etc.). You can download RStudio at [www.rstudio.com](http://www.rstudio.com) (the free version).

- 2) Stan, BUGS, JAGS, Nimble – These are specialty programs focused on Bayesian analysis of statistical models using Markov chain Monte Carlo (MCMC) methods. In the past I've used BUGS. I've been out of touch with the Bayesian computation landscape for a number of years. My intent is to get back in with Stan and especially it's R implementation, RStan. I hope to use this in the course (but it is not a guarantee). If you want to get ahead of the game, you can install RStan (<https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started>) but it is fine with me if you wait on this.