CODE SWAPP: ANALYZING NICK ULLE'S CODE

CHRISTOPHER CONLEY

1. Reason for not reviewing assigned person

They did not complete the HW1 assignment so I took the liberty of studying Nick Ulle's code; it was written in Python, a new language to me, so it was a good learning experience.

2. Code Description

Nick wrote functions for both Metropolis-Hastings and a Metropolis-within-Gibbs MCMC sampler for Bayesian logistic regression. The two MCMC approaches shared the same log likelihood, retuning procedure, and parameter input. The primary structural code difference was the MWG's had an additional for loop that accounted for conditioning on previous and the current draws in the posterior and had p separate decision rules per iteration, instead of a single decision rule as in the MH function.

3. Readibility

Score: 10 The code had sufficient comments in the key places. Because the variable names and functional structure of the code was so clean, Nick needed very few comments to explain what he was doing; I liked how he explained the "why" not just the "what" in certain comments; this is something Hadley Wickham has emphasized. The code was parsed into very digestible functions with clear naming.

4. Elegance

Score: 10 Since I am not a Python programmer, it is hard to comment on how efficient it is, but he pre-allocated memory for data structures and explained in his comments where things were not currently in vectorized form in python. The code is highly abstracted, with no hard coding to be seen. Because of the functional approach he took, there was no evident redundancy.

5. Kudos

I really liked the way Nick passed in an additional option of using the MWG or MH algorithm within his MCMC sampler. The fact that all his code, including generous white space was less than 300 lines, is a testament to how abstracted his implementation was. I learned that

6. Advice

This is perhaps a matter of taste, but maybe you could rearrange the order of the functions so that before you call them in a script they are defined beforehand. See for example, def autocorr().