Julien Clancy
Silliman 2015

Random Simulations of the Prefectures of Japan
with Eric Weese

After its economic collapse Japan began merging municipalities within its prefectures in an effort to cut fixed costs. Municipalities were given the option to engage in any contiguous merger involving any number of other municipalities, and would afterwards be legally recognized as a new prefecture. Professor Weese is in the process of writing a paper that explains the behavior of the municipalities (i.e. which mergers they undertook and which they did not) and thus its citizens, but needed simulation data to run his model on. The idea was to use some of the model itself to generate random prefectures (with semi-random polygonal municipalities) using certain parameters, then use the real model to attempt to recover those parameters from the randomly generated prefecture. My original task was to write code in R to generate these random prefectures. This took approximately three weeks. As the data had mergers that could not be accounted for by the model, it was necessary to introduce error terms, or epsilons, to cause mergers that by the model should not have happened to happen, and keep other mergers that should have happened from happening (should the model be run on actual premerger Japan, it should produce the observed results). The remaining three weeks of my stay in Japan was spent on two approaches to this problem. I also wrote a short piece detailing ways to create prefectures with arbitrarily bad utility that would still choose not to merge (and thus improve their utility) under a democratic system, a novel result in two dimensions (the other known case was in one dimension).

I originally took the original problem to be one of graph theory, but it turned out to be one of planar tessellation and other (in this case planar) geometric techniques such as Voronoi tessellation and Lloyd relaxation. Generation of the prefectures required some input and random data looks too harsh, so I implemented a technique called Simplex-Driven Poisson Disk Sampling, a commonly used algorithm in the video game industry for object placing and texture generation. The second problem was largely one of nonlinear optimization, and I became familiar with a suite called SCIP (for constrained mixed-integer programming) and with the languages used in optimization in general.

My knowledge in the areas of planar geometry and optimization was expanded greatly, and I learned the statistical language R from scratch for the project. I also read a great deal about graph theory and learned the Boost Graph Library's C++ interface (though I ultimately opted for a different system, based in R). Visiting Japan was of course also a great experience. I had never worked closely with a professor before, and I enjoyed my work greatly. I will likely continue to work on this project this fall as well. Overall, for me the SRO experience was a great one.