Nathan Hadfield

CS 677 (Spring 2014)

# MCMC Lab 3: Parameter Learning

*Lots of graphs, not much writing. What writing we do should be pointing out interesting things about the graphs and drawing conclusions.*

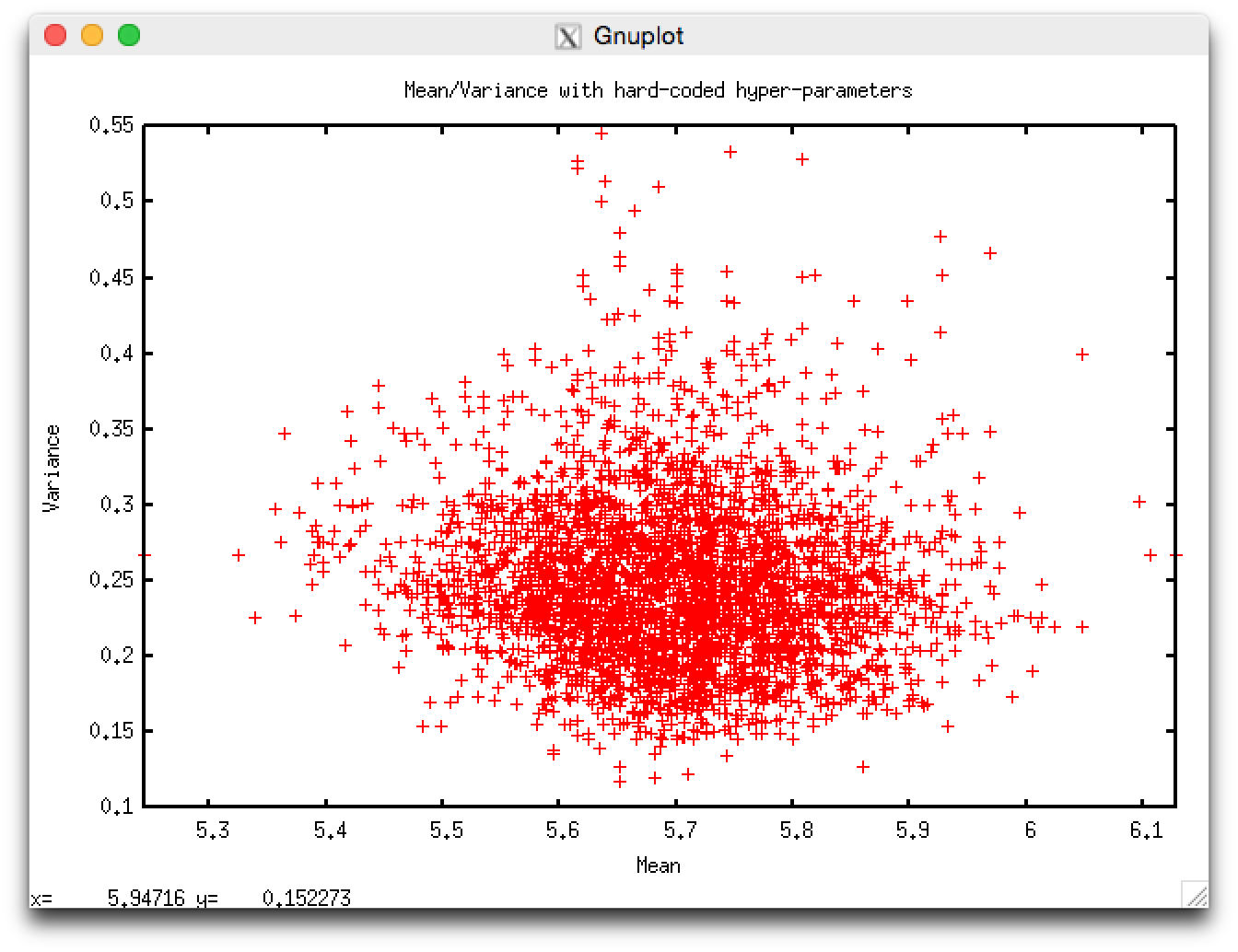
## Faculty Data

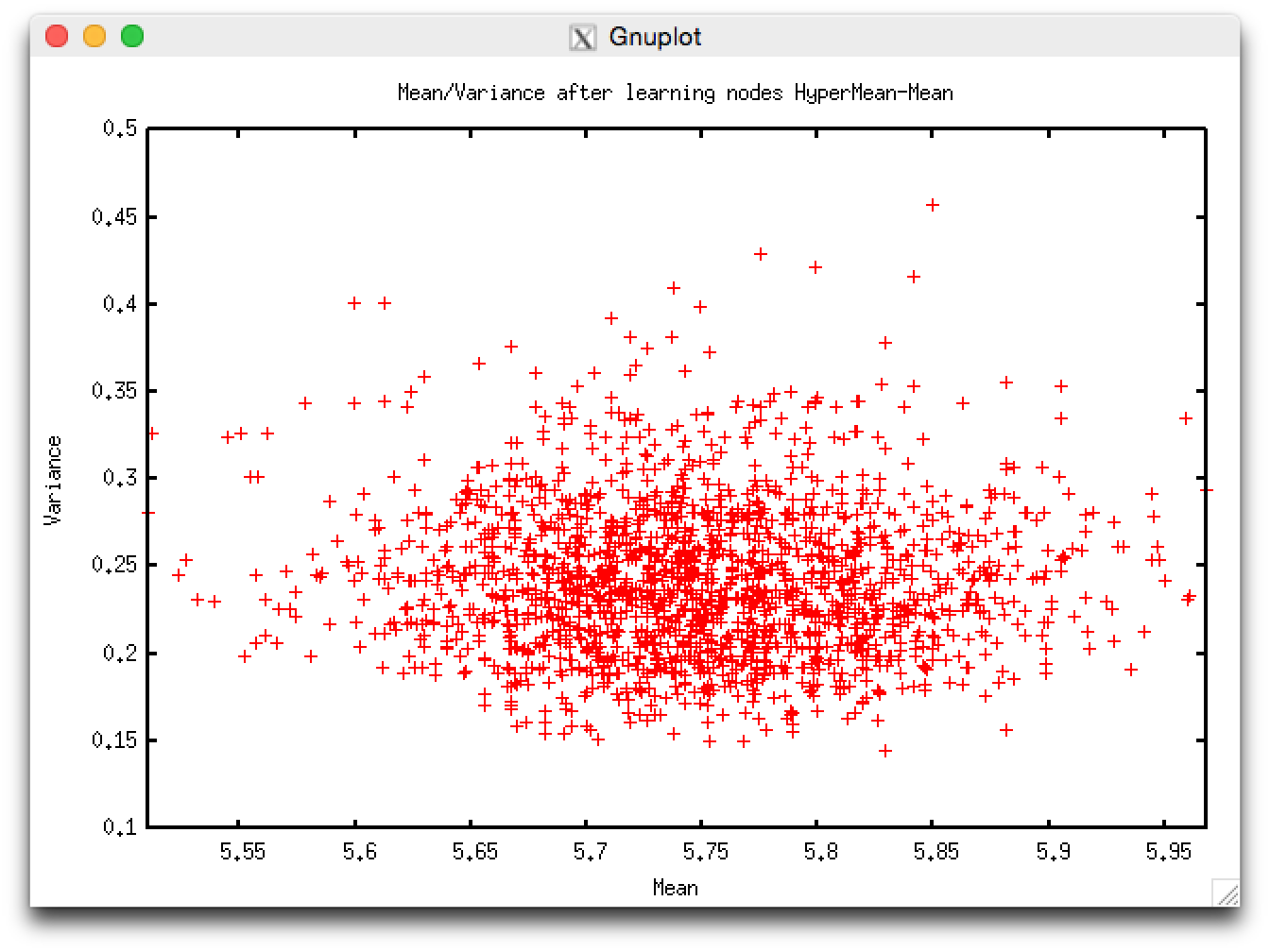
Add hyper-parameters and learn them one-by-one. Compare results as you go.

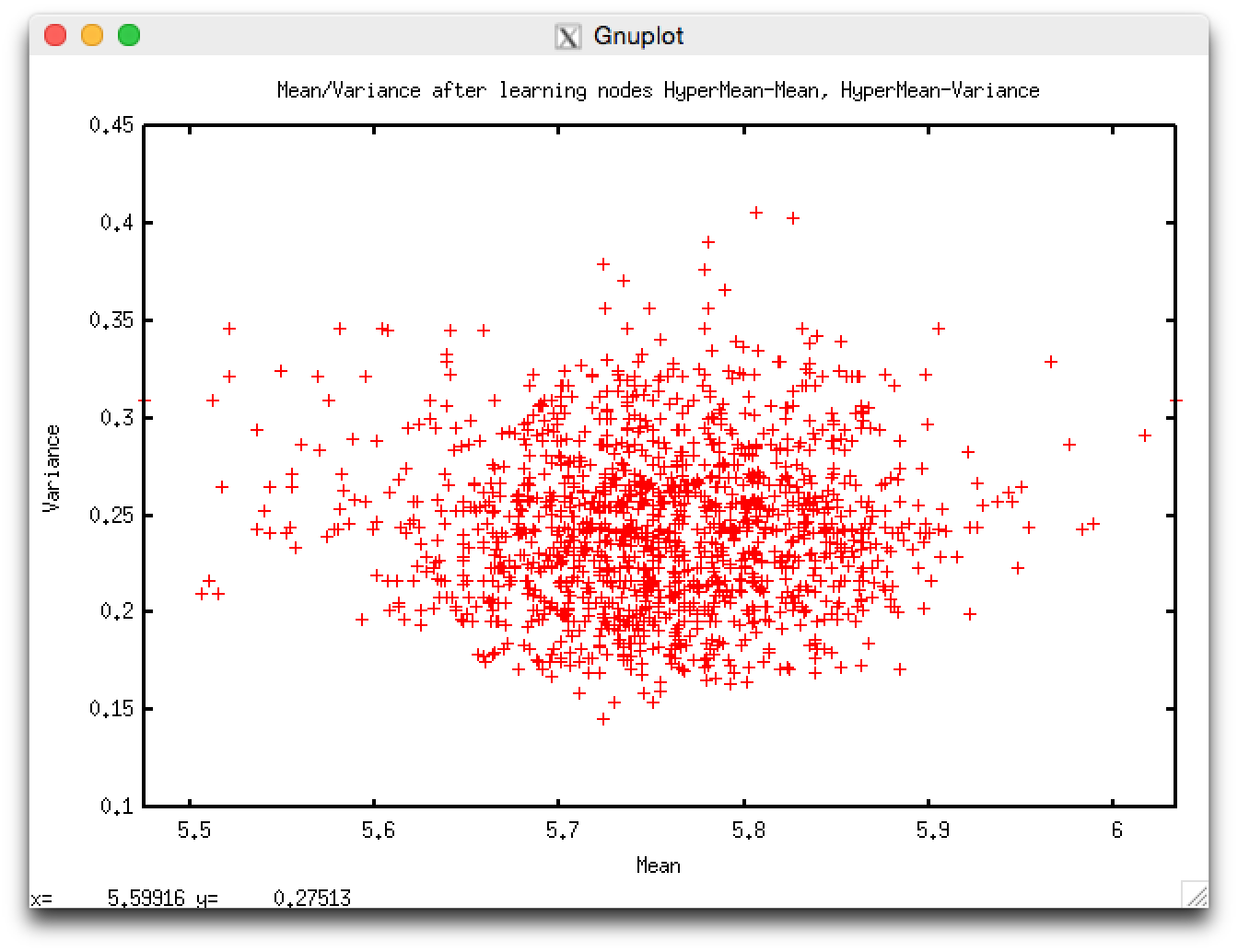


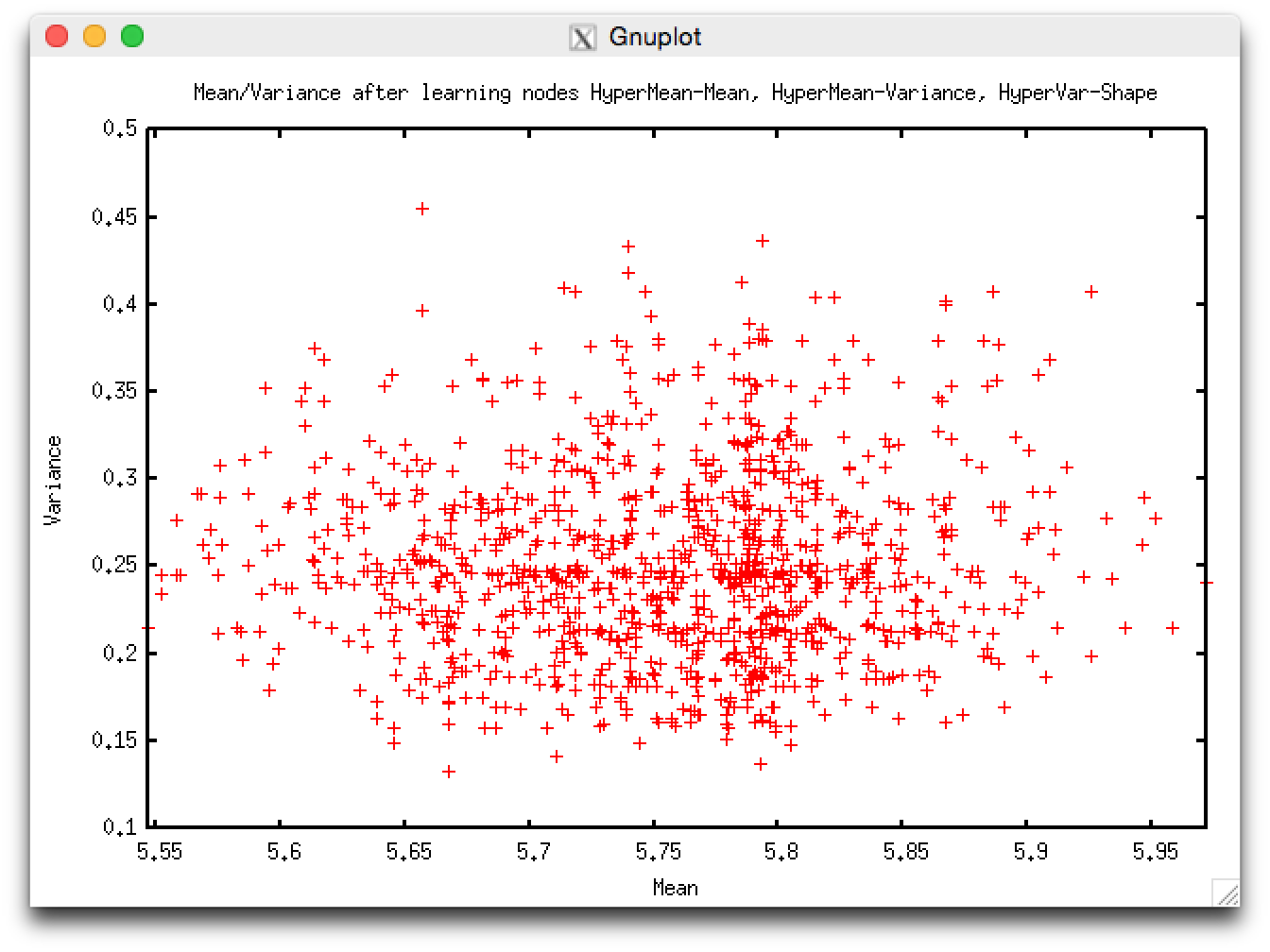
How does learning the hyper-parameters affect the posterior distributions for the mean and variance ( and )? Compare before and after learning.

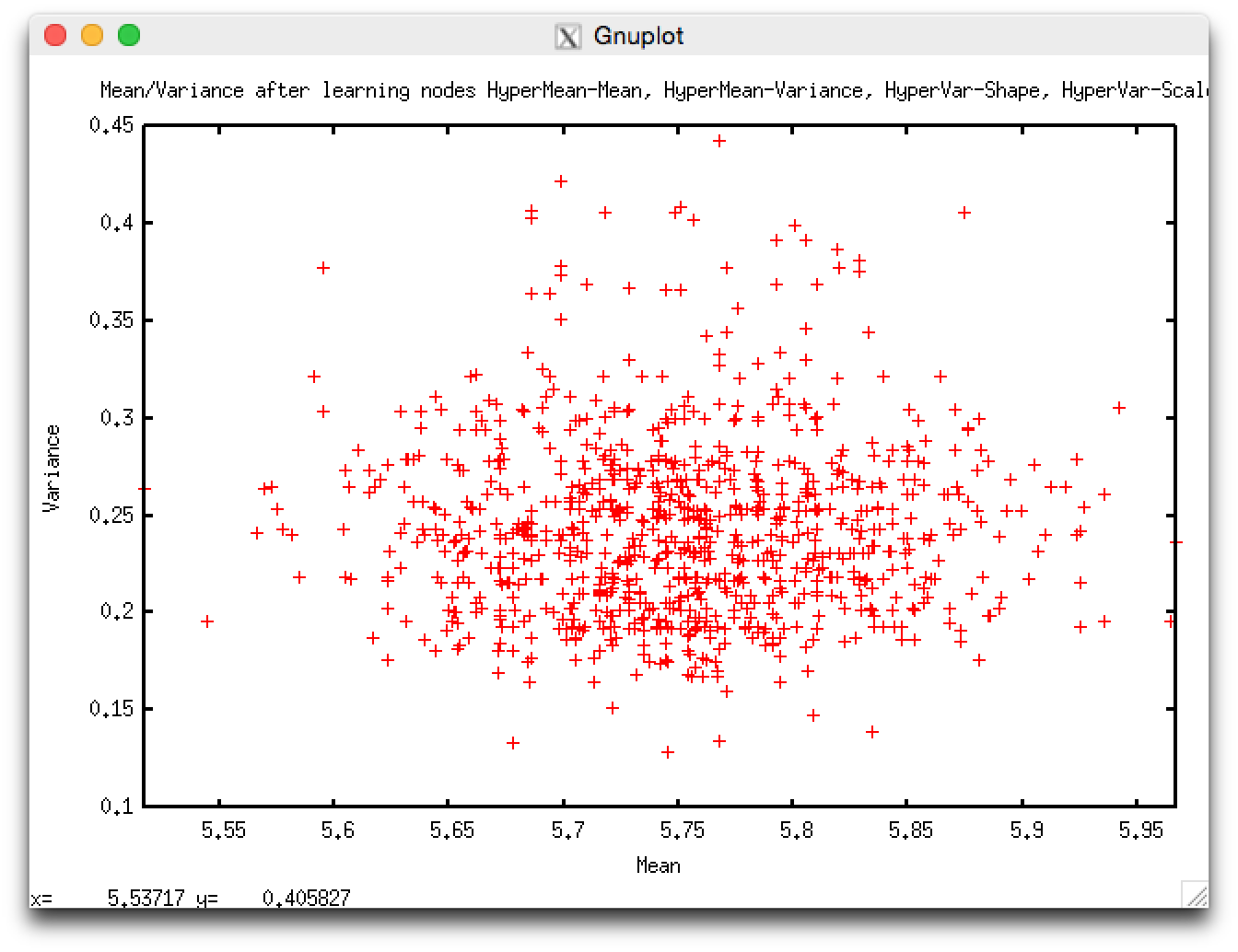
*Mean/Variance as additional hyper-parameters are learned:*



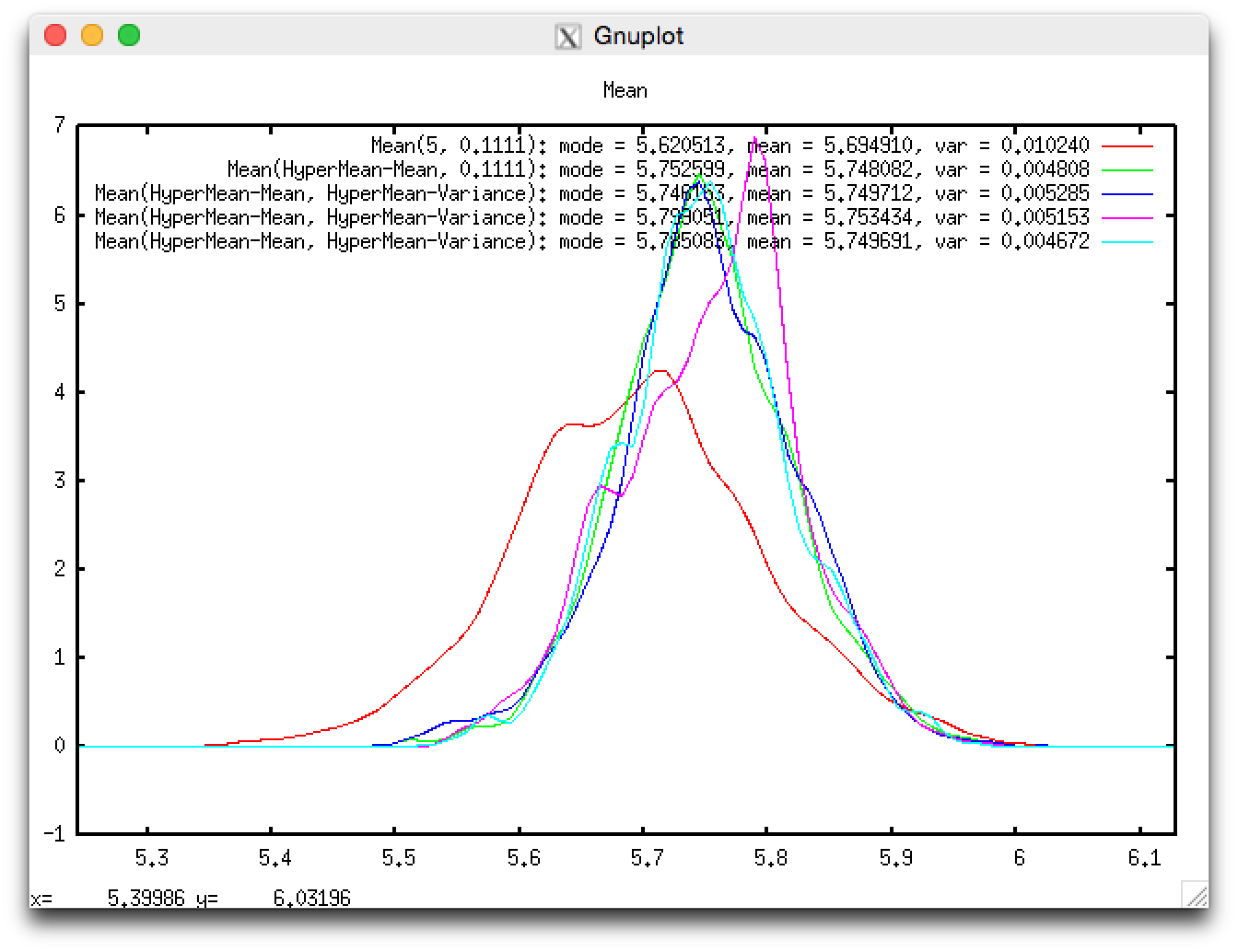


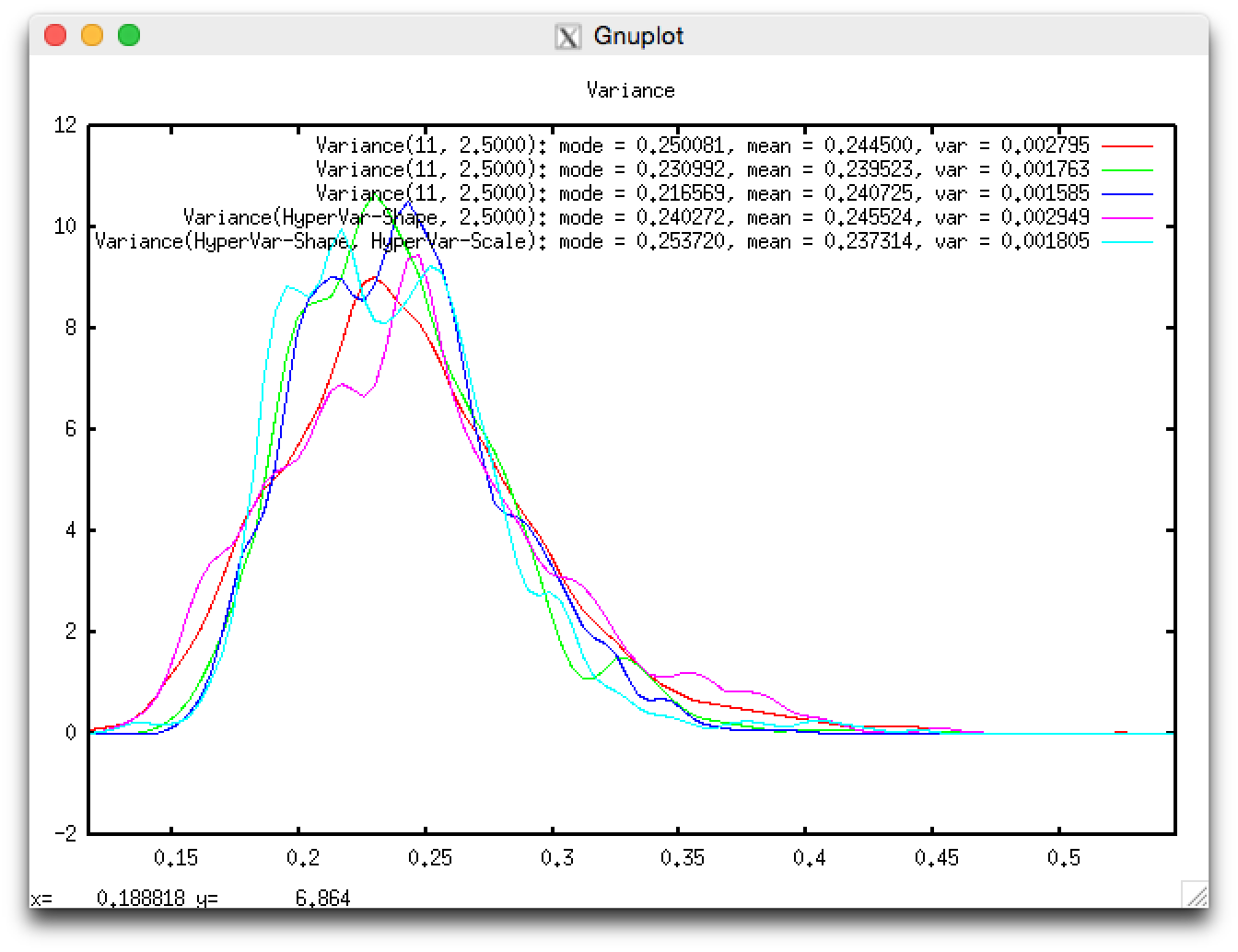


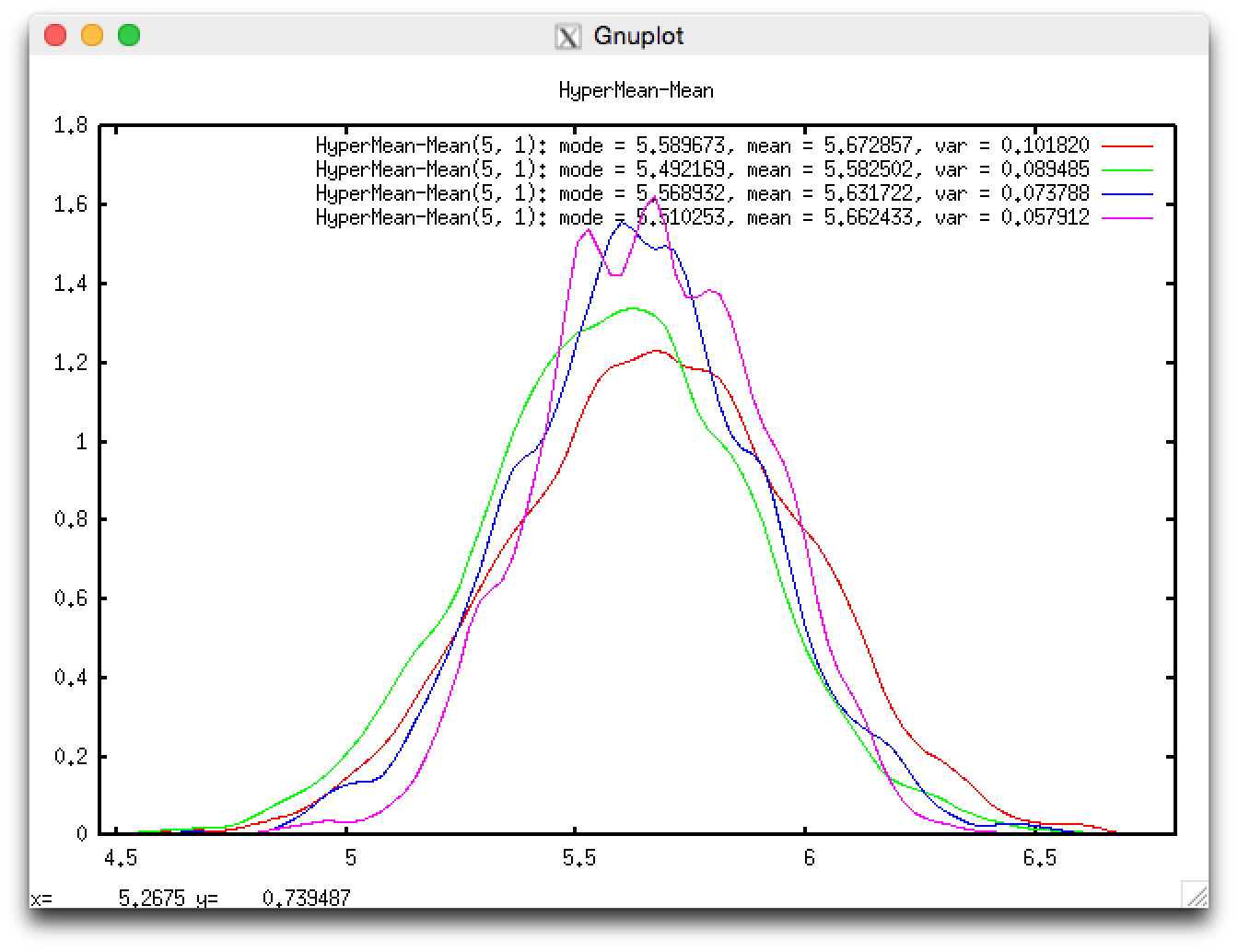


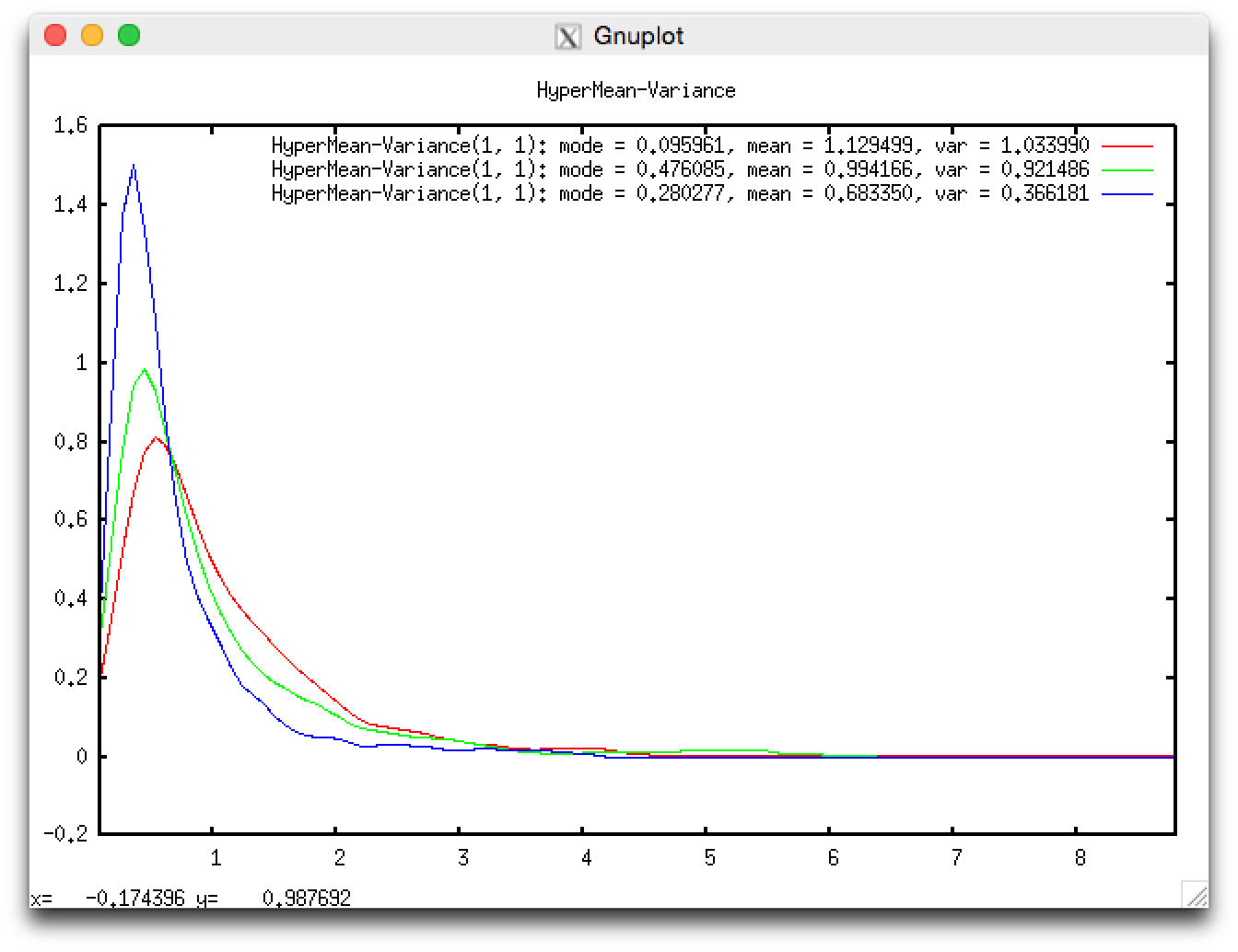


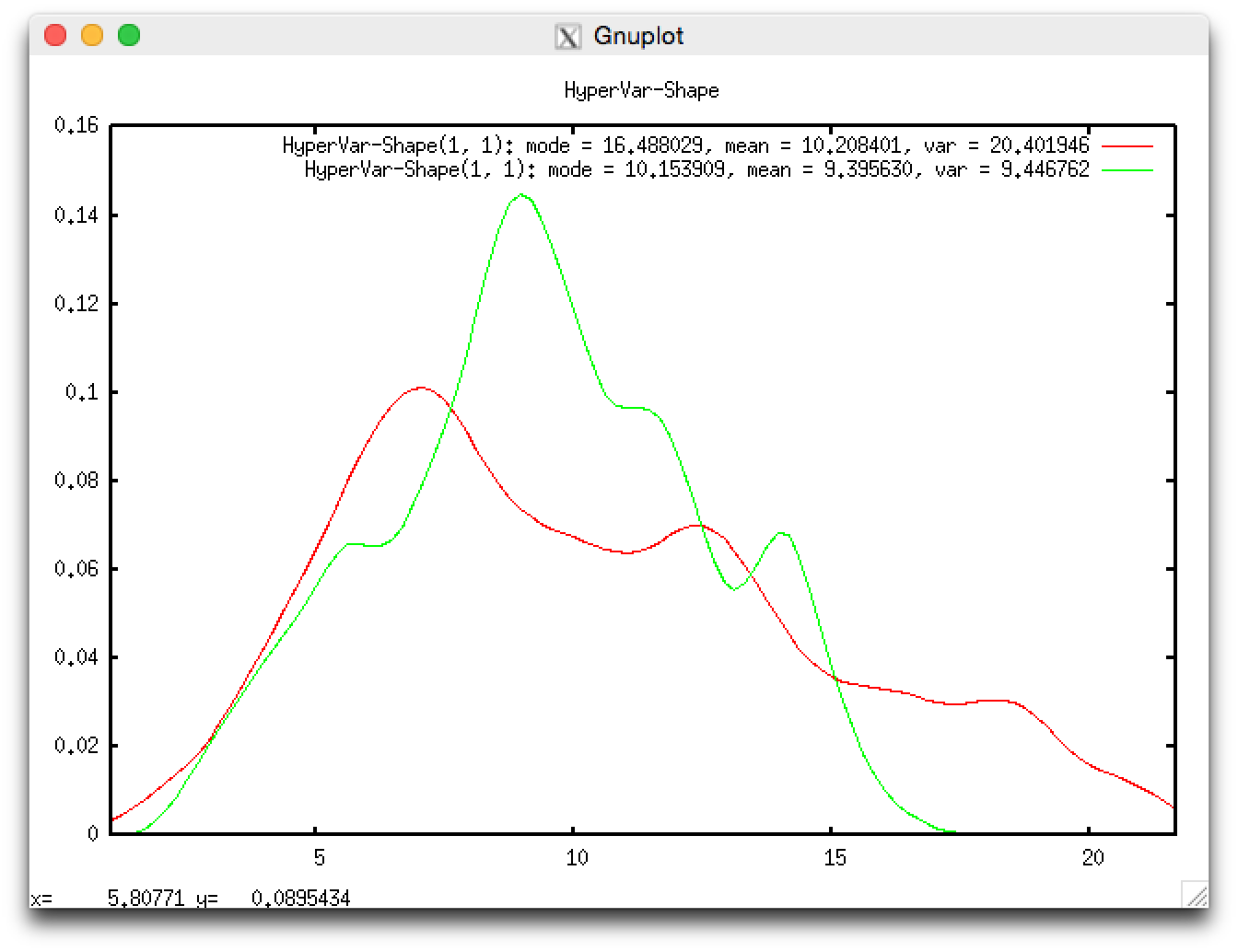
*Distributions of each node:*

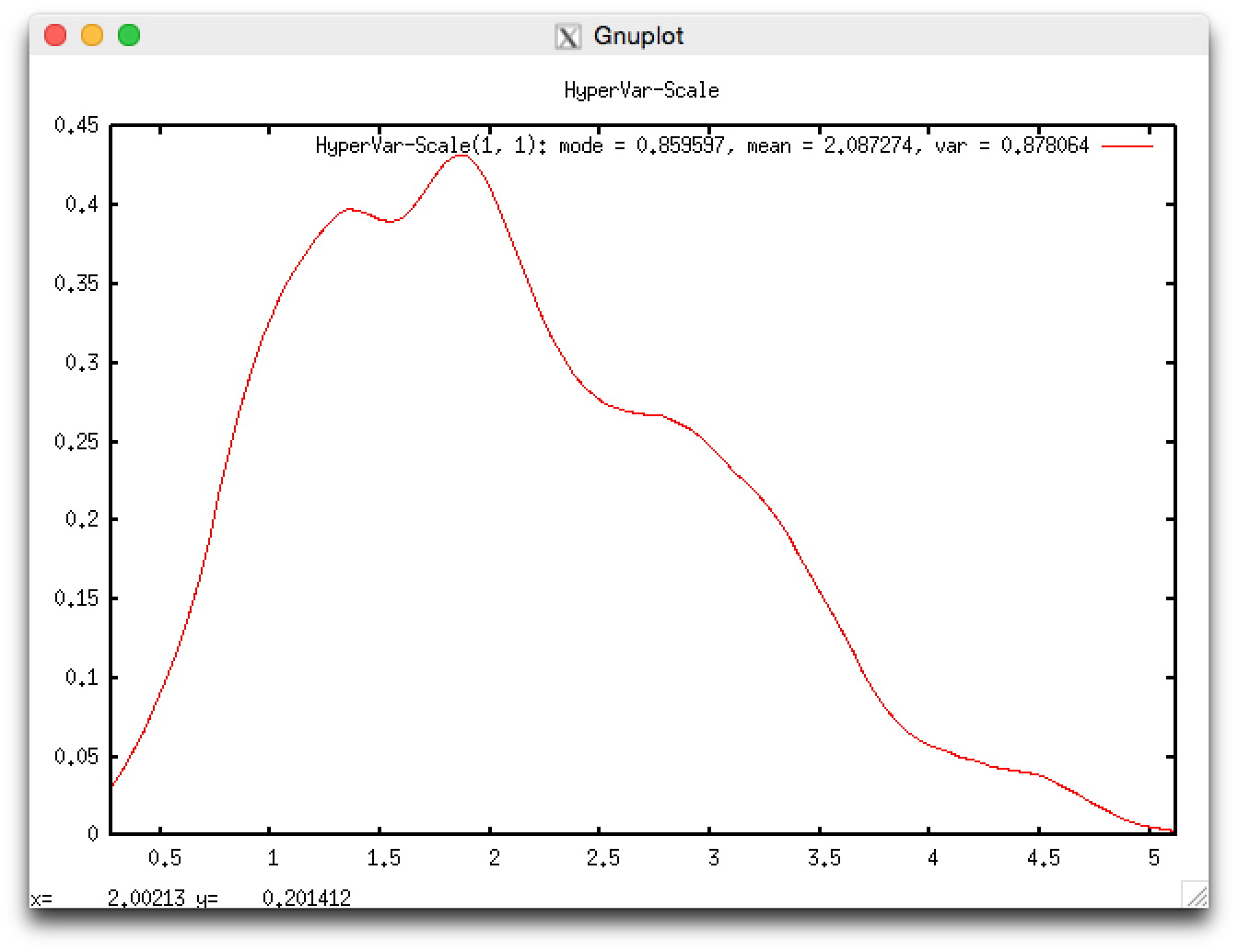






****

****



*Observations:*

Adding the HyperMean-Mean parameter tightened the Mean distribution and shifted it from mode=5.62 to mode=5.75. As subsequent nodes were added, the impact was less significant, but still the distributions of each of the nodes became a bit narrower. This is what we would expect, since that the network should be increasing in precision as it samples the distributions with greater flexibility.

## Alarm Model

Extend the Alarm model from MCMC Lab #1 to allow for multiple observations and parameter learning. Use new probalities:

* P(b)=0.2, P(e) = 0.3 (very unstable neighborhood)
* P(a|~b,~e) = 0.2 (bad alarm)
* P(j|~a)=0.2, P(m|~a) = 0.3 (friendlier neighbors)

Generate your own test data (trade places with knowns [observations] and unknowns).

**Get a feel for how adding hyper-parameters affects your network:** Add three hyper-parameters to learn, then try learning with half of the parameters, then try learning them all. Compare results as you go, and compare to the known true parameters.

**See how the amount of data affect the learning:** Start with just 100 sets of observations (you may need to go even lower). Give plots that show a change in learning as the amount of data changes.

**What in general makes a net harder to learn?** Experiment with the true parameters. E.g., go back to the original parameters—P(b)=0.001, etc.—does this make the system harder or easier to learn? Give a few plots.

**How does adding hyper-hyper-parameters affect your results?** (Add one or two. Give one or two plots.)

#### Putting it all together

**Add some observations with missing data** (removed by hand if easiest). E.g., some rows we don’t know if John Called or not, in others we don’t know if there was a burglary or not. Remove enough data to show a change in learning. Give a few plots.

**Add inference**: What is the posterior probability of a burglary given your training data, and assuming that Mary has called? Give a plot.

**Change the parameters and generate some data (with missing values).** Tell Dr. Seppi what the true parameters are, and give the data to another group.

**Run the learner on data from another group.** Use all hyper parameters, no hyper-hyper parameters. Plot the distributions of hyper parameters for the Burglary and Alarm nodes.