

HW 5

Nick Strayer

November 10, 2014

1. We have two sets to compare: Mk1s and MK2s....

	Mk1	Mk2
n	26,751	27,079
kills	183	222

We know that each BHK interaction occurred with one BHK and it isn't possible to lose more than one warrior per battle. There is a constant probability (p) of losing a warrior in a battle.

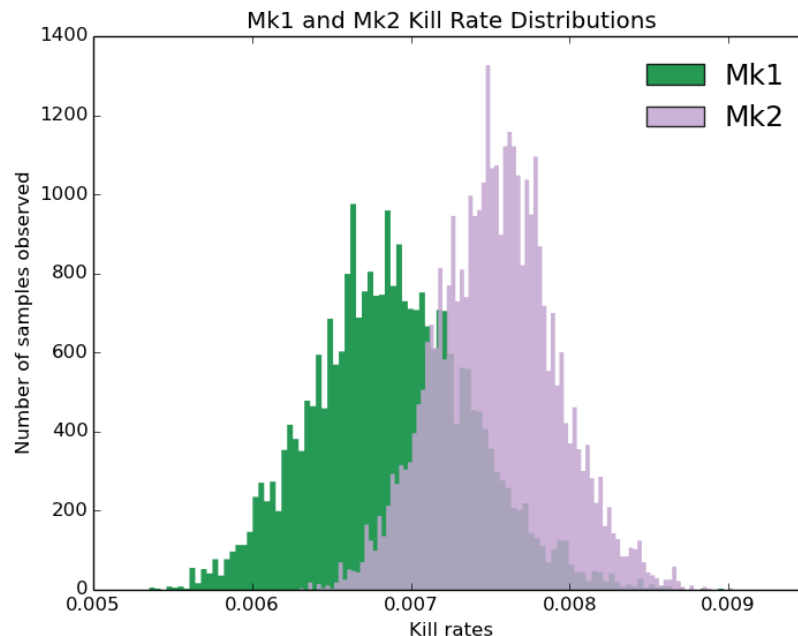
Part a

A binomial distribution characterizes this data. A binomial distribution is the result from n successive independent Bernoulli trials. A Bernoulli trial is characterized by a dichotomous result. In this case the warrior died, or the warrior lived, the chance of a success (or in this case of the warrior dying) is given by p .

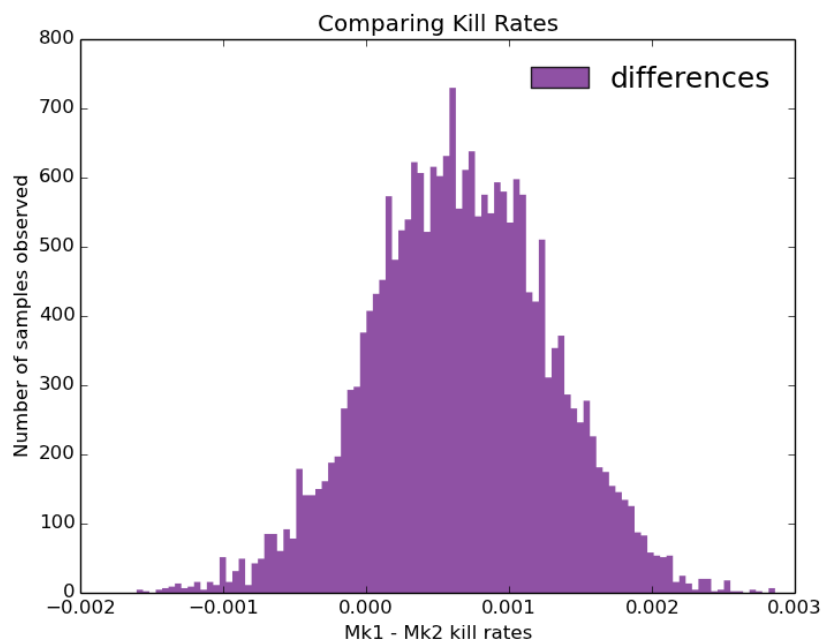
Part b

To the MCMCs!

Using MCMC per the in class coin flip problem we can compute the posterior distributions of the Mk1 and Mk2 kill rates (Problem.1.py):

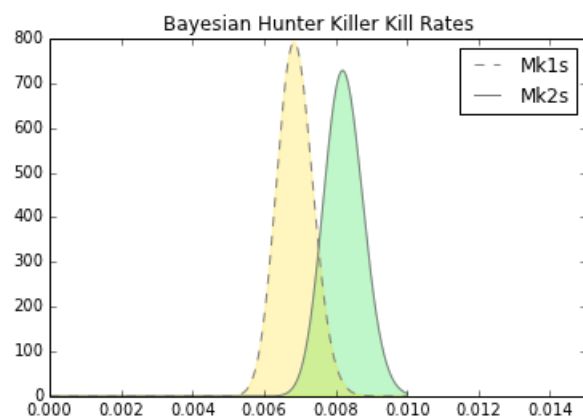


From this it is immediately apparent that there is a good amount of overlap between the two distributions. To further investigate this similarity we take the difference of the values given to us by MCMC in python (Problem_1.py line 28). From this data we plot and check the middle 95% of it to see if 0 is or isn't present:



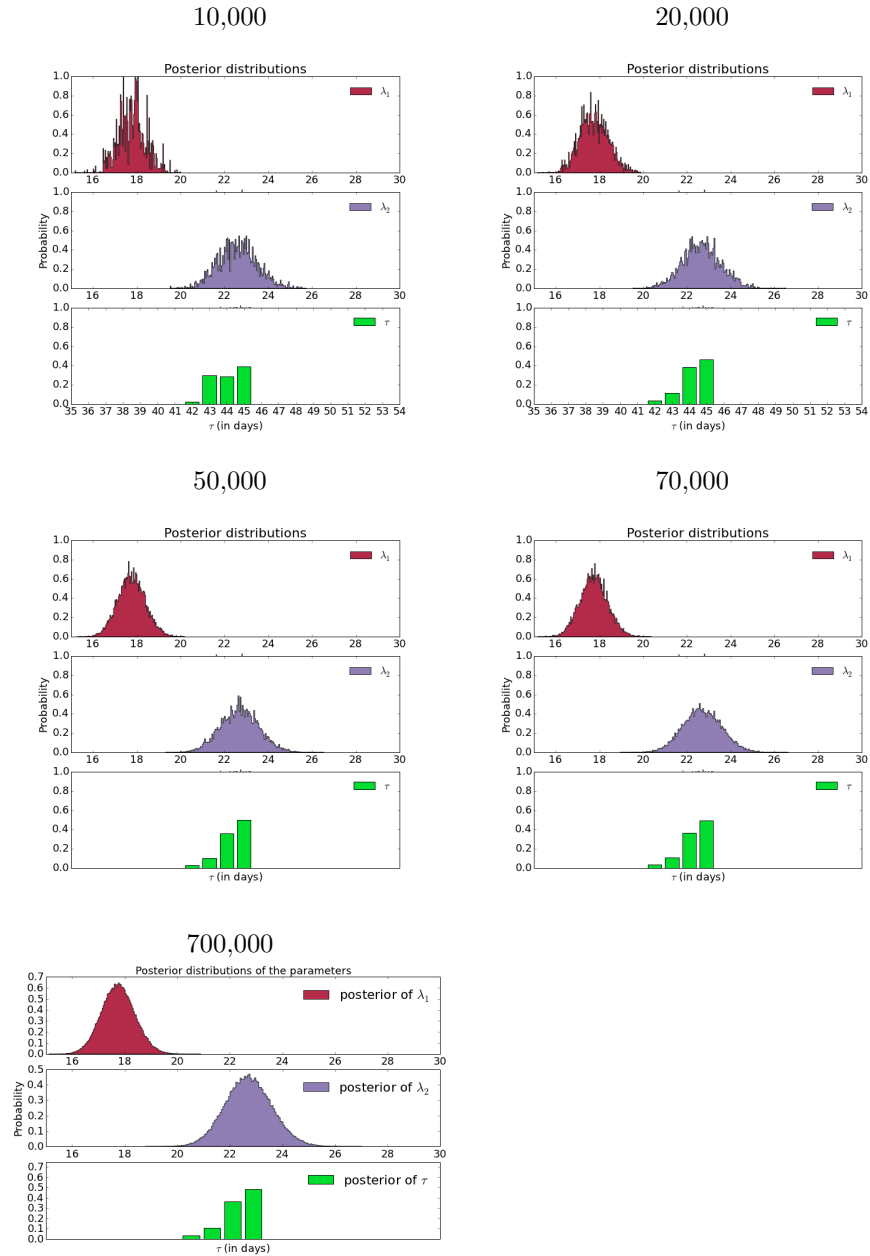
The middle 95% for this given run was $[-0.000573233022814, 0.00192015873444]$ of which 0 is firmly entrenched. Furthermore, 0 sits at a percentile of 13.8 or a p-value of .138. Therefore we fail to reject the null that there is no difference in rates.

Bonus exact posterior plot!



2. Part a

We observe convergence to the underlying distribution using MCMC by comparing plots of different amounts of iterations using the pymc algorithm (Problem_2a.py):



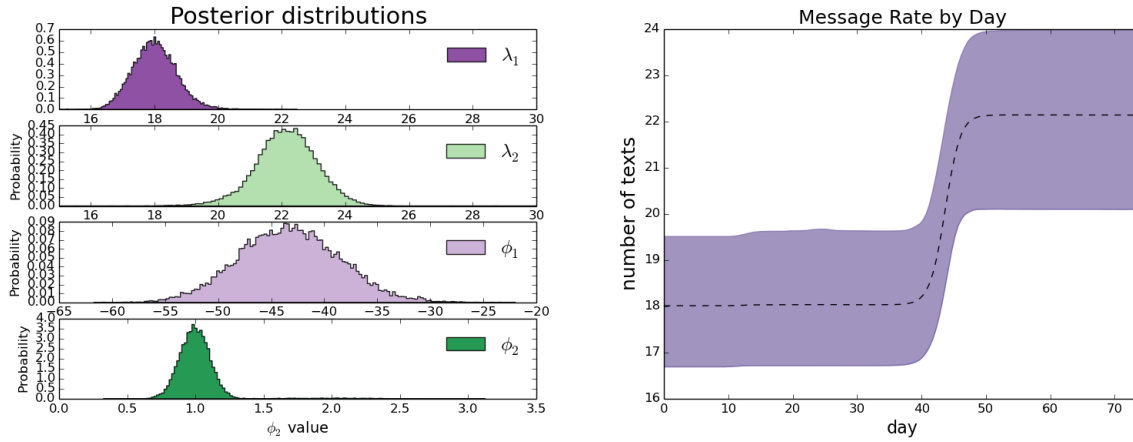
We can see from these plots that quickly the posteriors are converging to their underlying posteriors in a remarkably smooth way.

Part b

$$\begin{aligned}\Delta\lambda &= \lambda_2 - \lambda_1 \\ f(t) &= \frac{\Delta\lambda}{1 + e^{-(\phi_1 + \phi_2 \cdot t)}} + \lambda_1\end{aligned}$$

$f(t)$ is a sigmoid function. I chose this because it fits with the instructions of "smoothly changes the poisson rate" very well. The λ s in this case represent the respective texting rates on either side of the transition. The ϕ s combined to capture the behavior of the transition itself. The ratio $-\frac{\phi_1}{\phi_2}$ determines the actual point of transition and ϕ_2 controls the steepness of the transition.

Upon updating the code from part a/class with the newly derived sigmoid function (see lines 26-34 in Problem_2b.py) with priors for the ϕ s being normals (lines 19,20) we run MCMC.



We can plot the expected poisson rate for the texts over time (right figure) and arrive at the following figure (purple represents 95% confidence interval):

Looking at results from the new model the shift to the higher rate happens very abruptly, even with the confidence intervals applied. This seems to imply that the new model is not necessarily better than the old. It is more complicated but most likely unnecessarily so.