1 Problem 1

Consider the target density

$$f(x) = \frac{1}{1 + \pi^2(x-1)^2}.$$

1. Write a Random walk Metropolis-Hastings algorithm to sample from this density.

2. What about your algorithm in part a can be manipulated for tuning? Describe how you would tune the algorithm, stating clearly what the objective would be in tuning, how you would diagnose having achieved the objective, and what the consequences would be for a poorly tuned or untuned algorithm.

2 Problem 2

In Example 5.16, your book introduced the genetic linkage problem in connection with the EM algorithm. You dealt with a modification of this model for EM in HW 5 and revisited it in HW 7 where you applied a Gibbs Sampler to the model. The original article by Gelfand and Smith considers a slightly more complex situation. Assume there are n = 22 animals categorized as Y = (14, 1, 1, 1, 5) with cell probabilities $p = (\theta/4+1/8, \theta/4, \eta/4, \eta/4+3/8, (1-\theta-\eta)/2)$, so characterized by two unknown parameters θ and η whose sum is smaller than 1. In the augmented data model, we separate out the $\theta/4$ and $1/8$ in the first category and also separate out the $\eta/4$ and $3/8$ in the fourth category, so the model is $X = (X_1, \ldots, X_7) \sim multinomial(22; \theta/4, 1/8, \theta/4, \eta/4, 3/8, (1-\theta-\eta)/2$, so Y augmented through addition of X_1 and X_5.

1. Write out the full joint distribution of X, θ, and η when $Beta(1, 1)$ prior distributions are used on θ and η subject to $\theta + \eta \leq 1$.

2. Deduce the full conditional distributions for θ, η, X_1, and X_5. Note that if Z has a $beta(a, b)$ distribution right-truncated at c, then $f(x) \propto z^{a-1}(1-c-z)^{b-1}$.

3. Describe how to implement the Gibbs Sampler for this model

4. A Gibbs Sampler was run, and the output can be viewed by running the code in gibbsgenetic.R Comment on (and explain) your assessment of the convergence of the sampler based on these plots.

3 Problem 3

The objective is to maximize $h(x) = [cos(20x)+sin(50x)]^2$ on the interval (0,1). The following R code gives a simulated annealing algorithm for the optimization.

```
h=function(x){(cos(20*x)+sin(50*x))^2}
x=runif(1)
hval=hcur=h(x)
diff=iter=1
while(diff>epsilon)
  {
  prop=x[iter]+runif(1,-1,1)*scale
  if (               ){

    x=c(x,prop)
  }
  else{x=c(x,x[iter])}
  hcur=h(x[iter+1])
  hval=c(hval,hcur)
  diff=max(hval)-max(hval[1:(iter/2)])
  iter=iter+1
  }
```

1. Fill in the missing part of the code
2. Comment the code to indicate the structure of the algorithm.