GRADUATE STUDENT STAT 840 A3

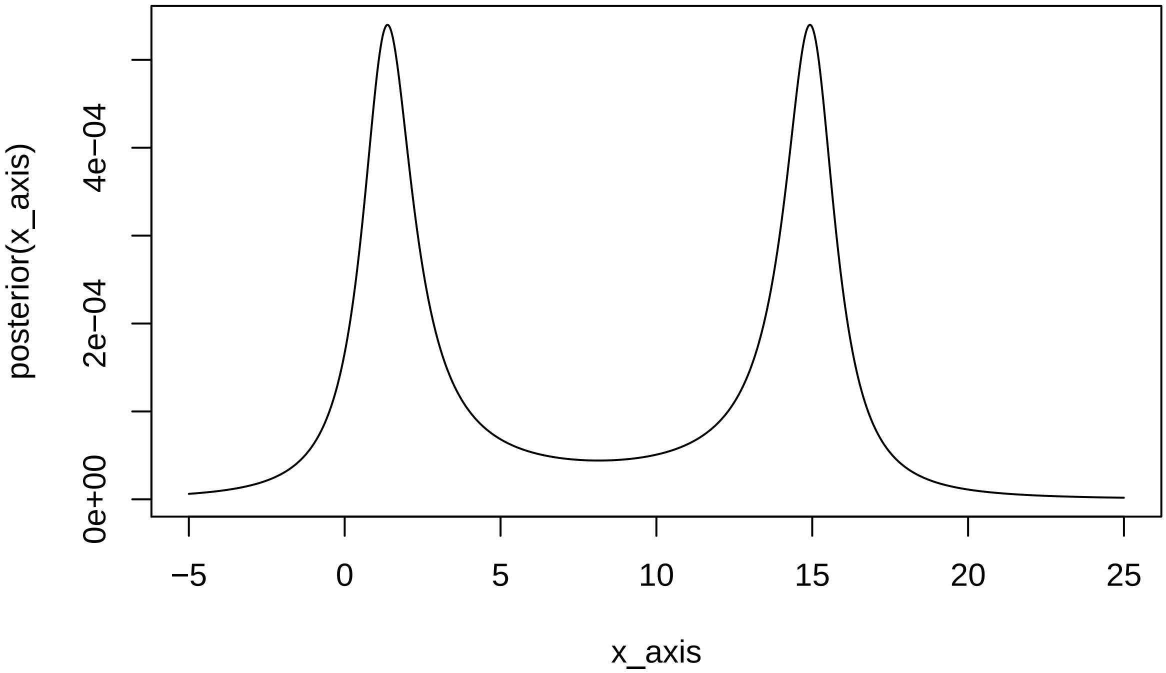
Vsevolod Ladtchenko 20895137

Problem 1

a)

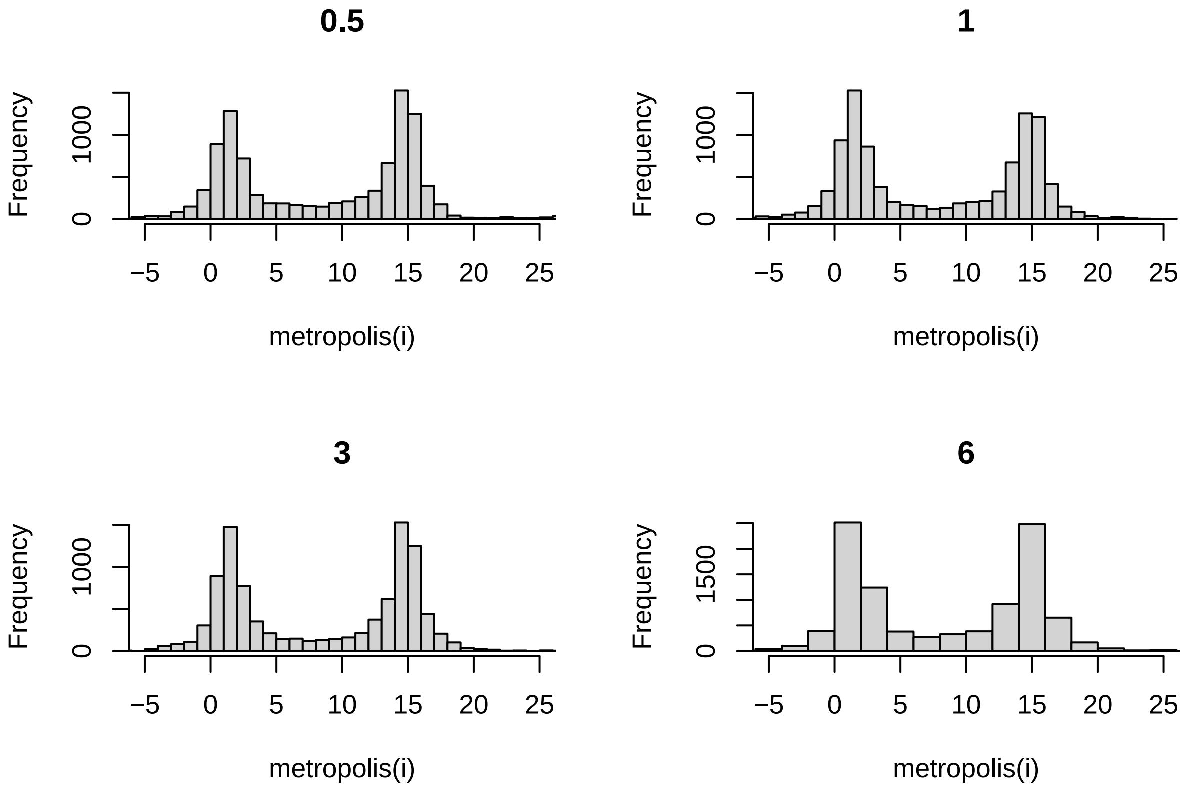
$$
\begin{aligned}
f(\theta) &= 1 \\
f(X \mid \theta) &= \prod_{i=1}^{2} \frac{1}{\pi\big[1+\big(x_i-\theta\big)^2\big]} \\
f(\theta \mid X) &= f(X \mid \theta) f(\theta) \\
&= \prod_{i=1}^{2} \frac{1}{\pi\big[1+\big(x_i-\theta\big)^2\big]}
\end{aligned}
$$

```r
data = c(1.3, 15)
posterior = function(theta) dcauchy(data[1],theta) * dcauchy(data[2],theta)

x_axis = seq(-5,25,0.01)
plot(x_axis, posterior(x_axis),type='l')
```



b)

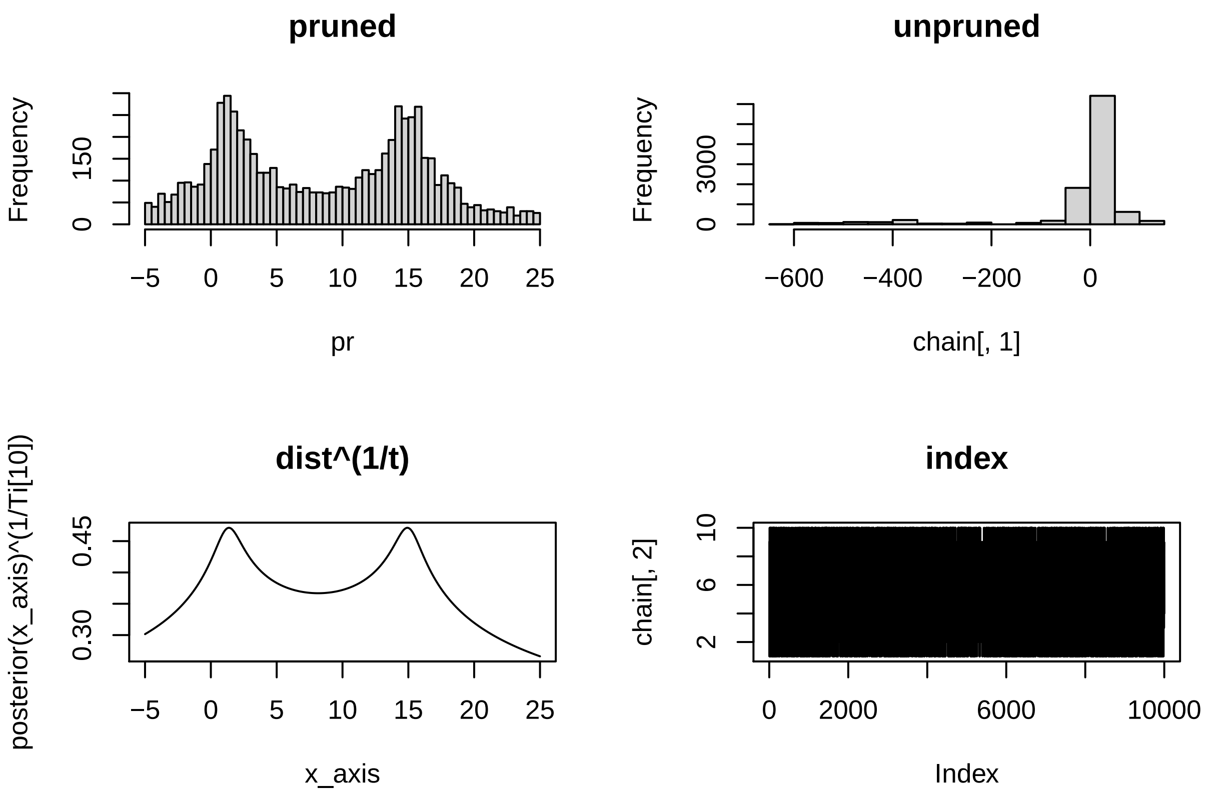
```
accept = function(old,new)
{
  r = posterior(new) / posterior(old)
  return(min(r, 1))
}

metropolis = function(scale)
{
  n = 10000
  chain = rep(NA,n)
  chain[1] = 1

  for (i in 2:n)
  {
    old = chain[i-1]
    new = rcauchy(1, old, scale)

    if (runif(1) <= accept(old, new))
      chain[i] = new
    else
      chain[i] = old
  }
  return(chain)
}

par(mfrow=c(2,2))
for (i in c(0.5, 1, 3, 6))
  hist(metropolis(i),xlim=c(-5,25),breaks=50,main=i)
```



c)

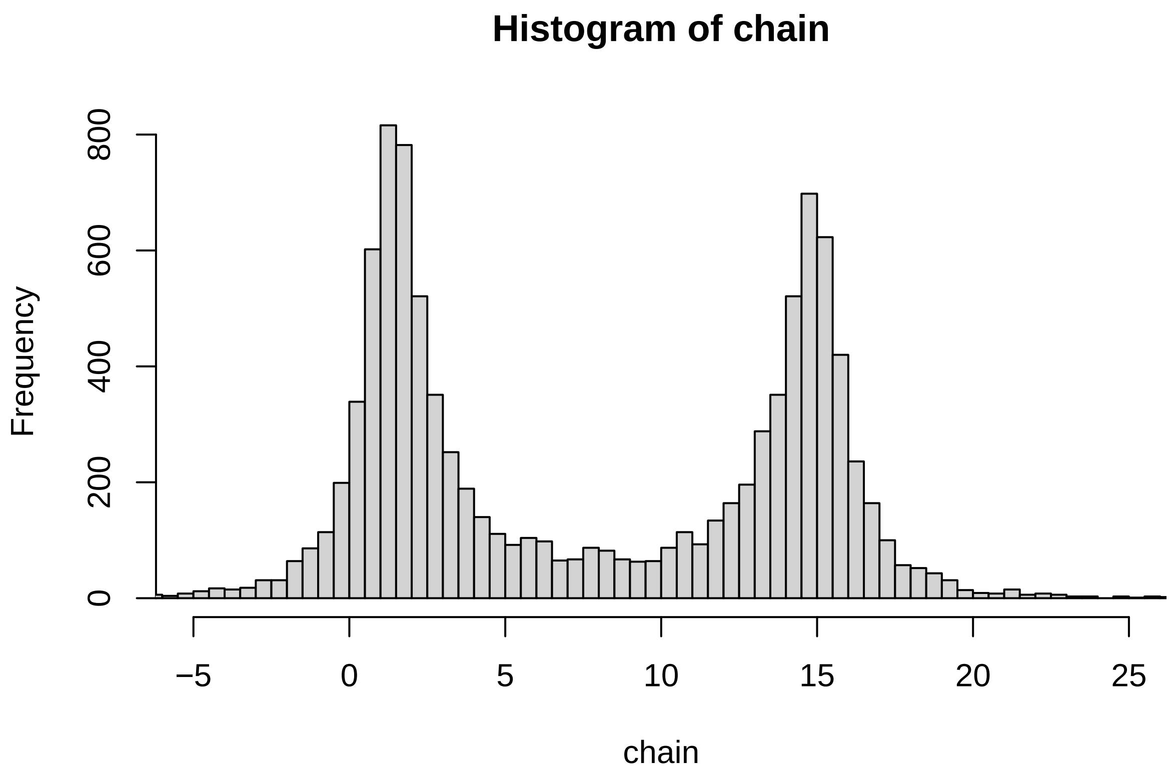
```
Ti = 1/seq(1, 0.1, -0.1)

posterior_i = function(x,i) posterior(x)^(1/Ti[i])

q_iold_inew = function(iold,inew)
{
  # Note: I hard-coded K,K-1 as 10,9 since those are the last possible indices
  # otherwise, the notes suggest K=9 while length(Ti)=10
  if (all(c(iold, inew) == c(1,2)) || all(c(iold, inew) == c(10,9)))
    return(1)
  if ((abs(iold - inew) == 1) && (iold %in% seq(2,9)))
    return(0.5)
  return(0)
}

r_iold_inew_xnew = function(iold,inew,xnew)
```

```
{
  # assume p(iold) == p(inew)
  top = posterior_i(xnew, inew) * q_iold_inew(iold,inew)
  bot = posterior_i(xnew, iold) * q_iold_inew(inew,iold)
  if (bot == 0)
    return(1) # notes 4 p 16
  return(min(top/bot, 1))
}

accept_x = function(old,new,i)
{
  r = posterior_i(new,i) / posterior_i(old,i)
  return(min(r, 1))
}

n = 10000
chain = matrix(ncol=2, nrow=n)
chain[1,] = c(1,1)

for (i in 2:n)
{
  old_x = chain[i-1,1]
  old_i = chain[i-1,2]

  # step 2
  prp_x = rcauchy(1, old_x)
  if (runif(1) <= accept_x(old_x, prp_x, old_i))
    new_x = prp_x
  else
    new_x = old_x

  # step 3,4,5,6
  prp_i = sample(seq(1,10),1)
  if (runif(1) <= r_iold_inew_xnew(old_i, prp_i, new_x))
    new_i = prp_i
  else
    new_i = old_i

  chain[i,] = c(new_x, new_i)
}
par(mfrow=c(2,2))
pr = chain[,1] # pruned
pr = pr[pr > -5]
pr = pr[pr < 25]
hist(pr, xlim=c(-5,25), breaks = 100,main='pruned')
hist(chain[,1], main='unpruned')
plot(x_axis, posterior(x_axis)^(1/Ti[10]),type='l',main='dist^(1/t)')
plot(chain[,2],type='l',main='index')
```



d)

```
invert_cauchy = function(f,x)
{
  # solve f(x) = Cauchy(x,a) for alpha given x,f(x)
  # this is equivalent to alpha being RV, and x is the location
  p = sqrt((1 / (pi*f)) - 1)
  return(c(x-p,x+p))
}

n = 10000
chain = rep(NA,n)
chain[1] = 1

for (i in 2:n)
{
  old = chain[i-1]

  v1 = runif(1) * dcauchy(old, data[1])
  v2 = runif(1) * dcauchy(old, data[2])
  int_1 = invert_cauchy(v1, data[1])
  int_2 = invert_cauchy(v2, data[2])
  l_endpt = max(int_1[1], int_2[1])
  r_endpt = min(int_1[2], int_2[2])
  chain[i] = runif(1, min=l_endpt, max=r_endpt)
}

hist(chain,xlim=c(-5,25),breaks=100)
```



e)

Part (b) with the usual Metropolis and symmetric Cauchy proposal density looks very much like the desired distribution. We do not observe any significant change in varying the scale of the proposal, which may be due to the fact that Cauchy has fat tails to begin with, so getting spread out values is already easy. We note the area between modes is fat here and perhaps the algorithm had no problem transitioning between them, either due to the Cauchy proposal jumping far enough, or due to this separation region not being too low-probability (and thus difficult to get across).

Part (c) uses simulated tempering in order to help MCMC mix faster. The algorithm considers our distribution taken to powers, which makes the modes shorter and the small regions higher in comparison, essentially flattening out the entire graph. This allows for more uniform-like jumps around the distribution. As a result, we see the area between the two modes considerably thicker. This makes sense since the purpose of the algorithm is to allow jumping across these areas of low probability. Although in this case the final histogram looks less like our density, it would come in handy for higher dimensional problems where areas of low probability are far more troublesome, and cannot otherwise be traversed.

Additionally, part (c) is interesting because the initial histogram appears wrong, with wild values, perhaps due to the Cauchy fat tails, and also the fatter tails of the distribution taken to a power. We prune the values outside our range of interest, and see that the histogram actually takes the shape of our density to a power. We plot this density on the bottom left, which is the posterior to the power of 0.1. The histogram on the top left is when we zoom in and discard values outside the range of interest. The histogram on the top right is the initial R output which makes one think the result is erroneous. The bottom right is the sequence of indices, showing us that we get a well mixed sample.

Part (d) also looks like our desired density. We were able to decompose it into a product of densities since it is a likelihood, and invert each density to solve for intervals analytically given the function's height. This is not always feasible due to the function form. But in this case it worked very well.