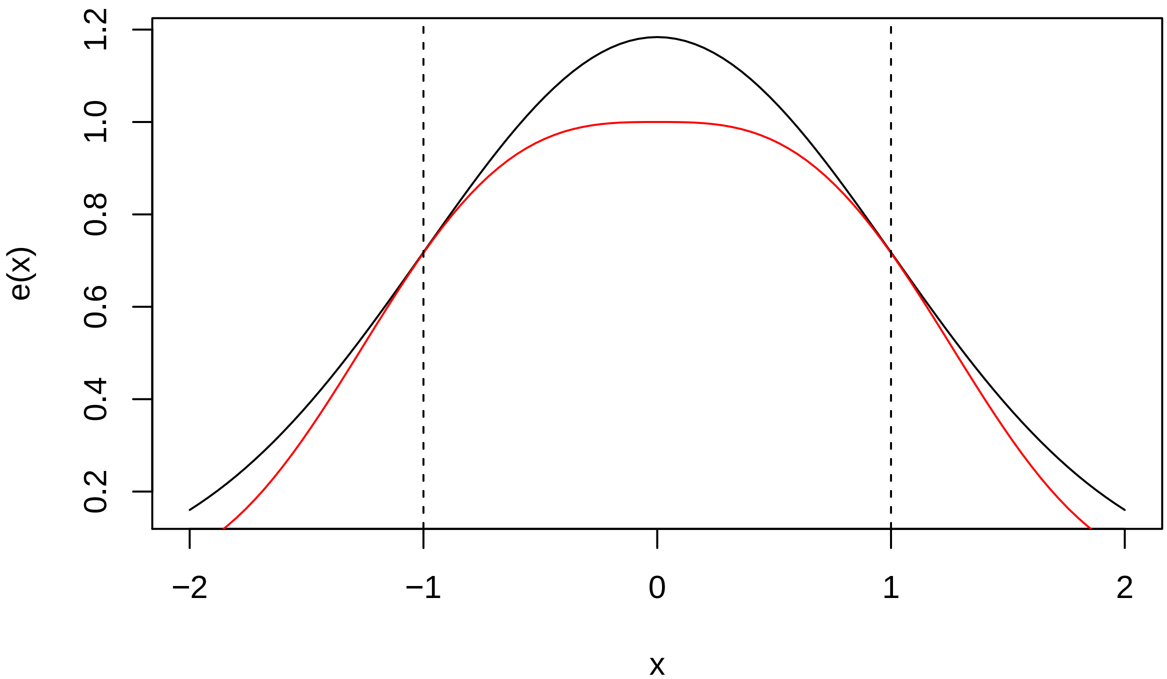
344HW4

Shay Lebovitz

5/2/2020

Regular Monte Carlo

```
e = function(x) {dnorm(x)/0.337}     ## optimal choice of \alpha under this proposaldensity
h = function(x) {exp(-abs(x)^3/3)}
curve(e(x), from=-2, to=2)
curve(h(x), add=T, col="red")
abline(v=c(-1,1), lty=2)
```




```
n = 100000
x = rnorm(n,0,1)
u = runif(n)
y = x[u<h(x)/e(x)]
mu_mc = mean(y^2)
V_mc = var (y^2)/n
```

We see that the variance of the regular Monte Carlo method using rejection sampling is 8.5558x10^{-6}. Using importance sampling, we will try to reduce that. First, I will define all the functions necessary:

```
set.seed(1)
sigma = seq(0.5,5,0.5)
effect = numeric(10)
mumrx = matrix (0, nrow = 10, ncol = 50)
```

```
muvec = numeric(10)
varvec = numeric (10)

q = function (x) {
  exp(-abs(x)^3/3)
}

h = function (x) {
  x^2
}

t = function (x) {
  h(x)*wts
}
```

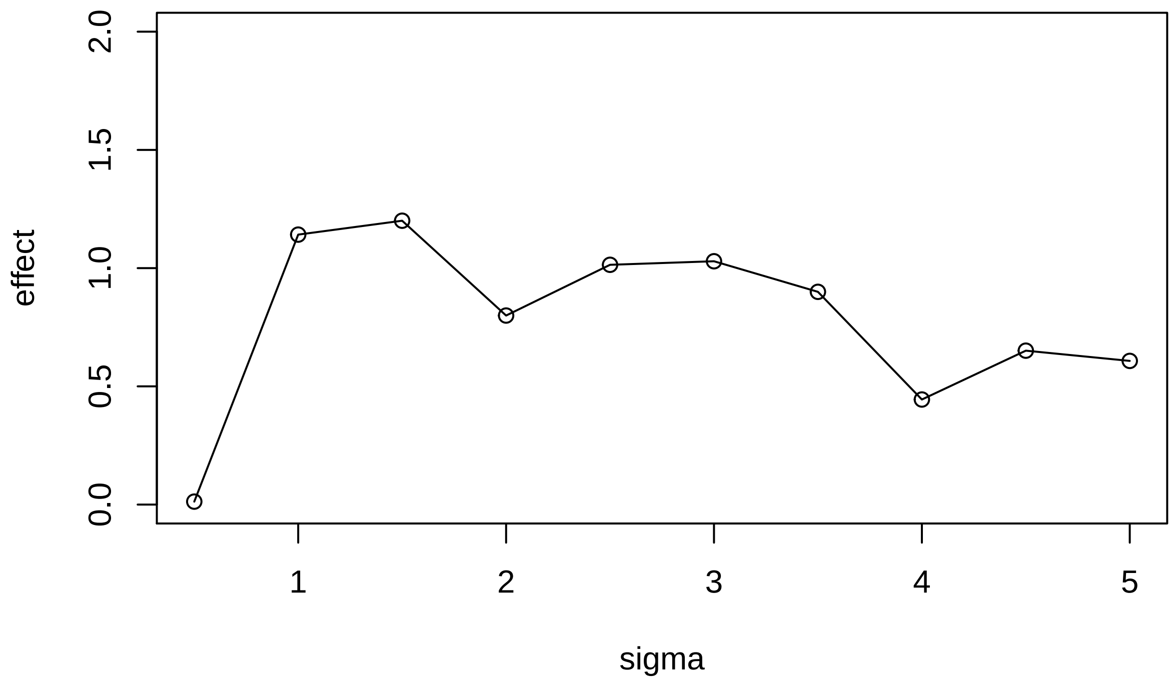
Next, I will run through the values of sigma from 0.5 to 5, finding which standard deviation gives the greatest reduction in variance from the regular Monte Carlo estimate. I will repeat this loop 50 times to get a more accurate analysis. Note that the normalized weights will have to be used, because we do not know the underlying function f(x), only the proportional function q(x).

```
for (j in 1:50) {
  set.seed (j+1)
  for (i in 1:10) {
  x=rnorm(n, 0, sigma[i])

  wts = q(x)/dnorm (x, 0, sigma[i])
  wts_norm = wts/sum(wts)

  mu.is = sum(h(x)*wts_norm)
  mumrx[i,j] = mu.is
  }
}

for(i in 1:10) {
  v1 = var(mumrx[i, 1:50])
  effect[i] = V_mc / v1
}
plot(sigma, effect, type="o", ylim=c(0,2))
```




```
best_sigma = 1.5
red_in_sample_size = effect[3]
red_in_sample_size
```

```
## [1] 1.200473
```

From this plot, we see that the greatest value for "effect" occurs at sigma = 1.5. There is a 1.198x reduction in sample size needed to achieve the same variance. For further analysis, one would calculate the theoretical variance of the importance sampling estimate.