GRADUATE STUDENT STAT 840 A1

Vsevolod Ladtchenko 20895137

Problem 8

(a)

First, note that we must scale by the factor $(b-a)$:

$$
\begin{aligned}
\theta &= \int_{-\infty}^{\infty} f(x)dx \\
&\approx \int_a^b f(x)dx \\
&= (b-a)\int_a^b f(x)\frac{1}{(b-a)}dx \\
&= (b-a)\,\mathbb{E}[f(x)] \\
&\quad \texttt{where } x \sim Unif(a,b) \\
&\approx (b-a)\frac{1}{n}\sum_{i=1}^{n} f(x_i) \\
&= (b-a)\hat{\theta}_{MC}
\end{aligned}
$$

For the finite bounds a and b, we choose 5 because we see that f decreases as $|x|$ increases, and $f(4) = 0.0003$ and $f(5) = .000003$ so in order to hopefully get within 4 decimal places of the real value, we choose this granularity.

```
get_SE = function(x) # return variance of sample mean
{
  n = length(x)
  mu = sum(x) / n
  s_sq = sum((x - mu)^2) / ((n-1)*n) # notes p 30
  return(sqrt(s_sq))
}

get_CI = function(xbar, se, conf=.95)
{
  a = 1 - conf
  L = xbar + qnorm(a/2)*se
  U = xbar + qnorm(1 - a/2)*se
  return(c(L,U))
}

d1 = function(x)
{
  return(exp((-x^2)/2))
```

```r
}

lim_a = -5
lim_b = 5
real = sqrt(2 * pi) / (lim_b - lim_a)

n = 1000
u1 = runif(n, lim_a, lim_b)

# simple MC estimate
deltas1 = d1(u1)
theta_mc = mean(deltas1)

# SE
SE = get_SE(deltas1)

# 95% confidence interval
ci = get_CI(theta_mc, SE)

c(n, theta_mc, SE, ci)
```

```
## [1] 1.000000e+03 2.531078e-01 1.064664e-02 2.322408e-01 2.739749e-01
```

(b)

Note the function $g(x)$ integrates to 5 on the interval $(-5, 5)$.

```
d2 = function(x)
{
  return(1 - abs(x)/5)
}

nn = 50
test2 = rep(0,nn)

for (i in 1:nn)
{
  n = 500

  u = runif(n, lim_a, lim_b)
  d1u = d1(u)
  d2u = d2(u)
  a = -cov(d1u, d2u) / var(d2u)

  deltas2 = d1u + a*(d2u - 5/(lim_b-lim_a)) # integral is 5, /10 for (b-a)

  theta_cv = mean(deltas2)

  test2[i] = theta_cv
}

# average estimate
mean(test2)
```

```
## [1] 0.2500715
```

```
# varaince of the estimate
var(test2)
```

```
## [1] 4.440184e-05
```

(c)

Note the function $h(x)$ integrates to 20/3 on the interval $(-5, 5)$.

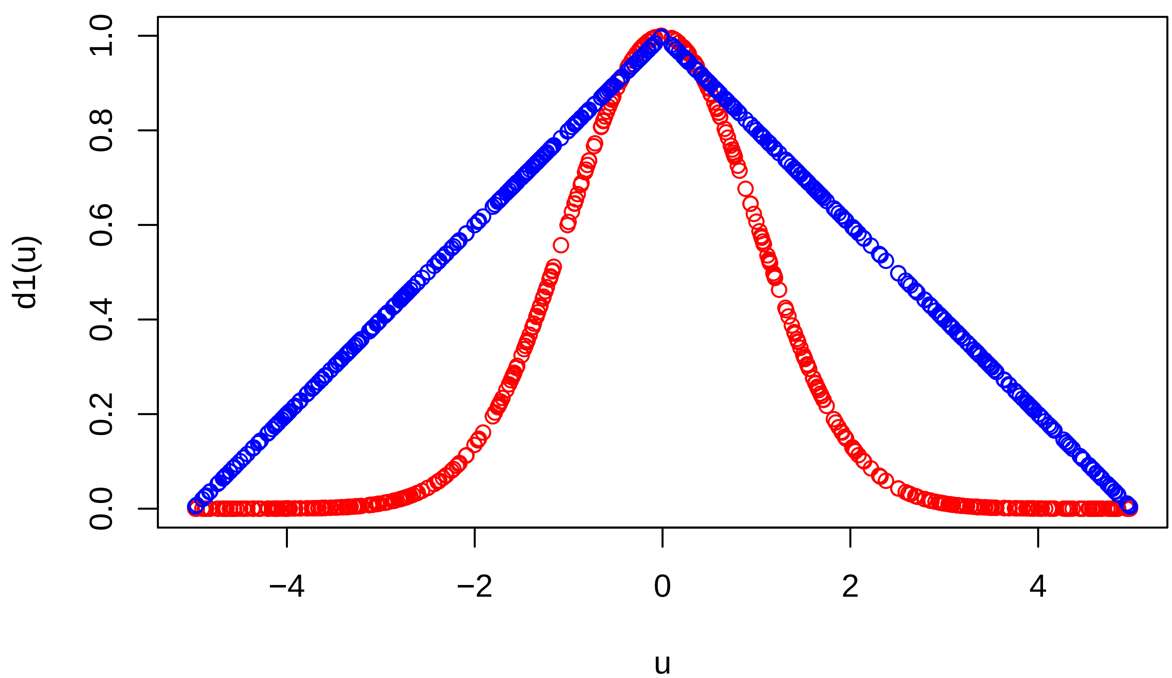
An easy way to tell if it will be better is to see if its graph is more similar to the original function, in which case they will have a higher correlation.

In part b, the function has the same overall slope: up and down. In part c, the function is farther away but has the same convexity In the center. it is hard to tell if it is a better fit.

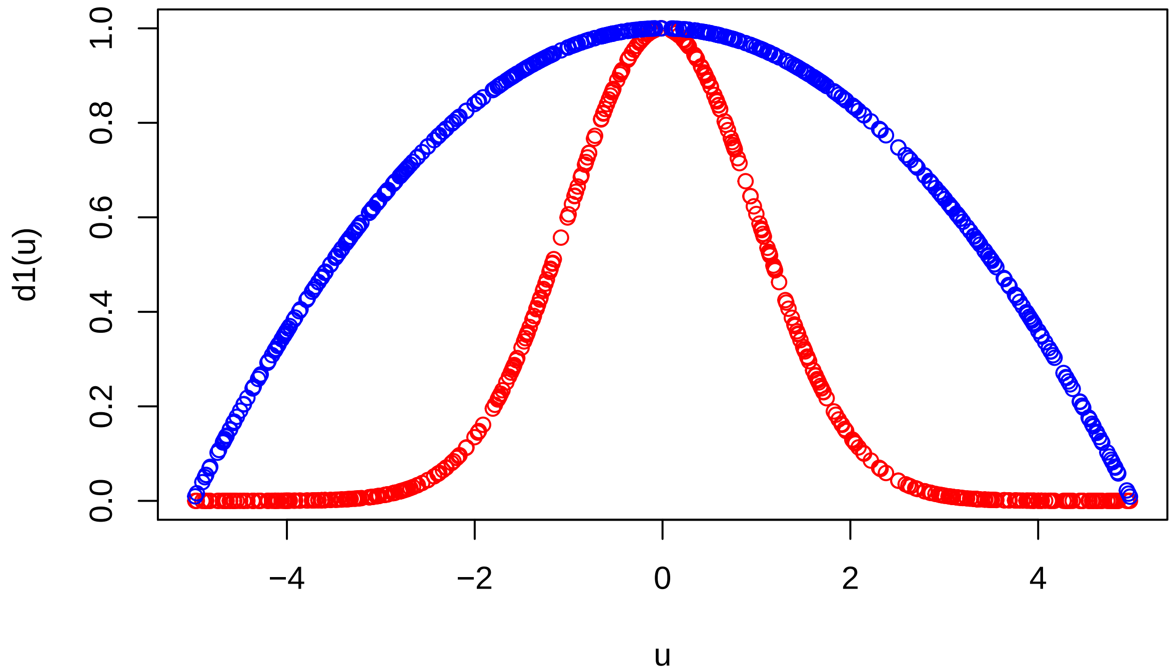
Conclusion: the variance reduction is worse with this function.

```
d3 = function(x)
{
  return(1 - (x^2)/25)
}

plot(u, d1(u), col='red')
points(u, d2(u), col='blue')
```




```
plot(u, d1(u), col='red')
points(u, d3(u), col='blue')
```




```
nn = 50
test2 = rep(0,nn)

for (i in 1:nn)
{
  n = 500

  u = runif(n, lim_a, lim_b)
  d1u = d1(u)
  d2u = d3(u)
  a = -cov(d1u, d2u) / var(d2u)
  deltas2 = d1u + a*(d2u - (20/3)/(lim_b-lim_a)) # integral is 20/3

  theta_cv = mean(deltas2)
  test2[i] = theta_cv
}
mean(test2)
```

```
## [1] 0.2494724
```

```
var(test2)
```

```
## [1] 0.0001258978
```