Stat 361 - Recitation 8

Variance Reduction

Orçun Oltulu

02-03 / 04 / 2020

The Use of Antithetic Variables:

The variance can always be reduced by increasing the number of Monte Carlo replicates, but the computational cost is high. Antithetic variables can bu used to reduce variance. These variables are negatively correlated.

Suppose we are interested in using simulation to estimate $\theta = E(X)$ and suppose we have generated X_1 and X_2, identically distributed random variables having mean θ_1 and θ_2. An unbiased estimate of θ is given by

$$\hat{\theta} = \frac{\hat{\theta}_1 + \hat{\theta}_2}{2}$$

$$Var(\frac{X_1 + X_2}{2}) = \frac{1}{4}[Var(X_1) + Var(X_2) + 2Cov(X_1, X_2)]$$

Hence, it would be advantageous (in the sense that the variance would be reduced) if X_1 and X_2 rather than being independent were negatively correlated.

Question: How can we arrange X_1 and X_2 negatively correlated?

Suppose that X_1 and X_2 are functions of m random numbers;

$$X_1 = h(U_1, U_2, ..., U_m)$$
$$X_2 = h(1 - U_1, 1 - U_2, ..., 1 - U_m)$$

where U is uniformly distributed on (0,1), then so is 1-U.

In addition, since U and 1-U are negatively correlated, we might hope that X_1 and X_2 are also negatively correlated, and indeed that result can be proved in the special case where the function h is a monotone (either increasing or decreasing) function of each of its coordinates.

Question 1:

Based on the function given above

$$f(x) = \sqrt{x + \sqrt{x + \sqrt{x}}} \quad 0 < x < 1$$

Find an approximation to the integral

$$I = \int_0^1 \sqrt{x + \sqrt{x + \sqrt{x}}} dx$$

```
f1 <- function(x) sqrt(x + sqrt(x + sqrt(x)))

a1 <- 0; b1 <- 1
n <- 10^5

x1 <- runif(n,a1,b1)

exp.estimate1 <- mean(sapply(x1,f1)) * (b1-a1)
exp.estimate1
```

```
## [1] 1.219486
```

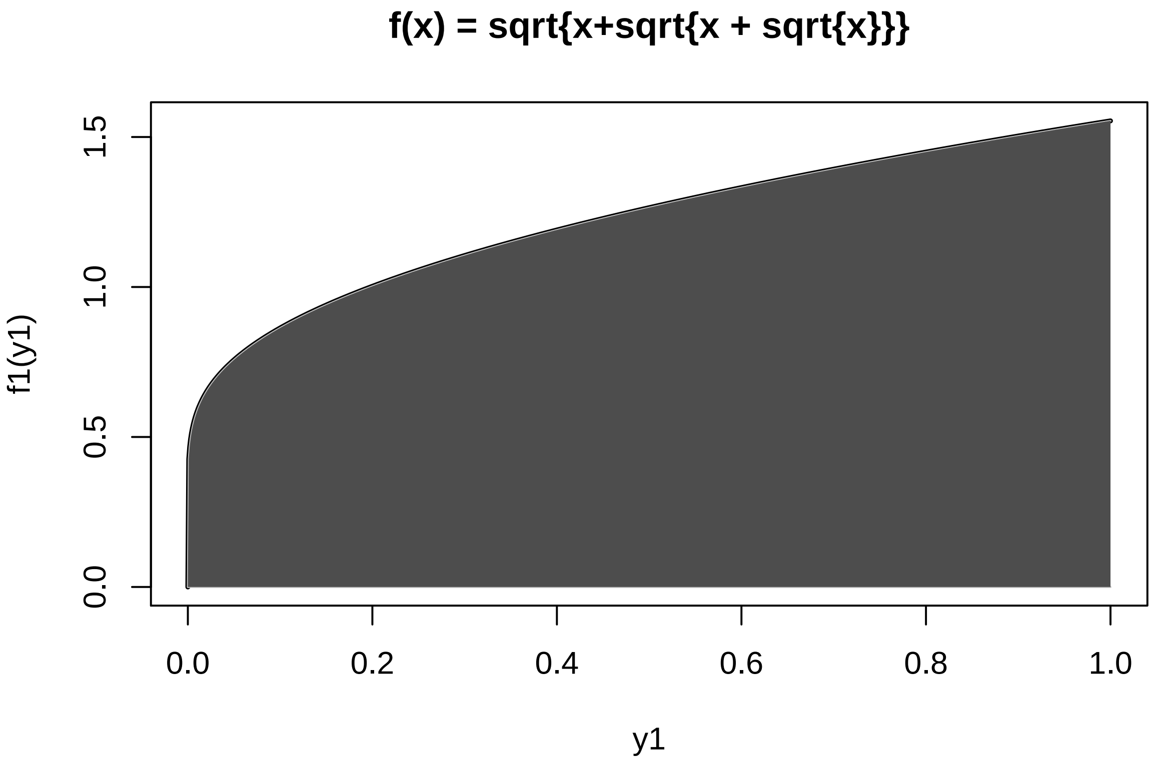
```
var.estimate1 <- mean(sapply(x1,f1)^2) - (mean(sapply(x1, f1)))^2
var.estimate1
```

```
## [1] 0.05923684
```

```
integrate(f1,a1,b1)$value
```

```
## [1] 1.219005
```

```
y1 <- seq(a1,b1,0.001)
y.low1 <- rep(0,times = length(y1))

plot(y1, f1(y1), type = "l",
     main = "f(x) = sqrt{x+sqrt{x + sqrt{x}}}",
     lwd = 2.5)

lines(y1,f1(y.low1), col = 'grey')
lines(y1,f1(y1), col = 'grey')
polygon(c(y1, rev(y1)), c(f1(y1), rev(f1(y.low1))),
     col = "grey30", border = NA)
```




```
f1 <- function(x) sqrt(x + sqrt(x + sqrt(x)))

a1 <- 0; b1 <- 1
n <- 10^5

x1 <- runif(n/2,a1,b1)

theta1.1 <- mean(sapply(x1,f1)) * (b1-a1)
theta1.2 <- mean(sapply(b1-x1,f1)) * (b1-a1)

exp.ant.estimate1 <- (theta1.1 + theta1.2) / 2


z1.1 <- f1(x1)
z1.2 <- f1(b1-x1)

var.ant.estimate1 <- (var(z1.1)+var(z1.2)+2*cov(z1.1,z1.2)) / 4
```

```
result1 <- matrix(c(exp.estimate1,exp.ant.estimate1,
                    var.estimate1, var.ant.estimate1),nrow=2,
                  dimnames = list(c("classical approach","antithetic variables"),
```

```r
                              c("estimated mean","variance")))
result1
```

```
##                            estimated mean    variance
## classical approach               1.219486 0.05923684
## antithetic variables             1.218842 0.00278681
```

```r
var_reduction1 = 100 * ((var.estimate1-var.ant.estimate1)/var.estimate1)

cat("The antithetic variable approach achived",
    round(var_reduction1,2), "%", "reduction in variance.")
```

```
## The antithetic variable approach achived 95.3 % reduction in variance.
```

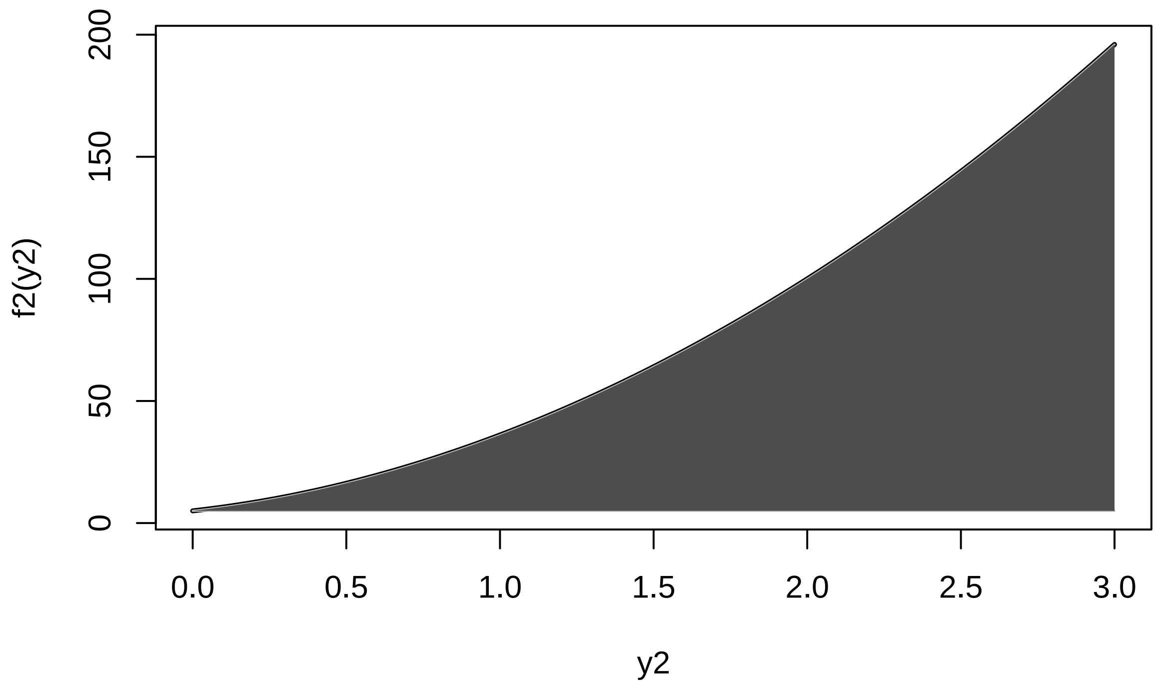
Question 2:

Use Monte Carlo integration with antithetic variables to estimate

$$I = \int_0^3 \frac{e^{-2x}}{1+x^2} + (4x+2)^2 dx$$

```r
start1 <- proc.time()

f2 <- function(x) exp(-2*x) / (1+x^2) + (4*x+2)^2
a2 <- 0; b2 <- 3
n <- 10^5

x2 <- runif(n,a2,b2)

exp.estimate2 <- mean(sapply(x2,f2)) * (b2-a2)
var.estimate2 <- mean(sapply(x2,f2)^2) - (mean(sapply(x2, f2)))^2

run.time.1 <- proc.time() - start1
```

```r
y2 <- seq(a2,b2,0.001)
y.low2 <- rep(0,times = length(y2))

plot(y2, f2(y2), type = "l",
     lwd = 2.5)

lines(y2,f2(y.low2), col = 'grey')
lines(y2,f2(y2), col = 'grey')
polygon(c(y2, rev(y2)), c(f2(y2), rev(f2(y.low2))),
     col = "grey30", border = NA)
```




```
start2 <- proc.time()
f2 <- function(x) exp(-2*x) / (1+x^2) + (4*x+2)^2
a2 <- 0; b2 <- 3
n <- 10^5

x2 <- runif(n/2,a2,b2)

theta2.1 <- mean(sapply(x2,f2)) * (b2-a2)
theta2.2 <- mean(sapply(3-x2,f2)) * (b2-a2)

exp.ant.estimate2 <- (theta2.1 + theta2.2) / 2

z2.1 <- f2(x2)
z2.2 <- f2(b2-x2)

var.ant.estimate2 <- (var(z2.1)+var(z2.2)+2*cov(z2.1,z2.2)) / 4

run.time.ant.1 <- proc.time() - start2
```

```
integrate(f2,a2,b2)$value
```

```
## [1] 228.3989
```

```r
result2 <- matrix(c(exp.estimate2,exp.ant.estimate2,
                    var.estimate2, var.ant.estimate2),nrow=2,
                  dimnames = list(c("classical approach","antithetic variables"),
                                  c("estimated mean","variance")))
result2
```

```
##                      estimated mean  variance
## classical approach         227.7281 3158.4526
## antithetic variables       228.2795  117.2559
```

```r
var_reduction2 = 100 * ((var.estimate2-var.ant.estimate2)/var.estimate2)

cat("The antithetic variable approach achived",
    round(var_reduction2,2), "%", "reduction in variance.")
```

```
## The antithetic variable approach achived 96.29 % reduction in variance.
```

```r
run.time.1
```

```
##    user  system elapsed
##    0.53    0.00    0.53
```

```r
run.time.ant.1
```

```
##    user  system elapsed
##    0.28    0.02    0.31
```

The Use of Control Variates

Again suppose that we want to use simulation to estimate $\theta = E(g(X))$. Suppose that there is a function f, such that $\mu = E(f(X))$ is known, and f(X) is correlated with g(X).

Then for any constant c, it is easy to check that $\hat{\theta}_c = g(X) + c^*(f(Y) - \mu)$ is an unbiased estimator of θ.

The variance

$$Var(\hat{\theta}_c) = Var(g(X)) + c^2 Var(f(X)) + 2cCov(g(X), f(X))$$

is a quadratic function of c. It is minimized at $c = c^*$, where

$$c^* = -\frac{Cov(g(X), f(X))}{Var(f(X))}$$

and minimum variance is

$$Var(\hat{\theta}_{c^*}) = Var(g(X)) - \frac{[Cov(g(X), f(X))]^2}{Var(f(X))}$$

The random variable f(X) is called a *control variate* for the estimator g(X).

From previous equation we can see that Var(g(X)) is reduced by

$$\frac{[Cov(g(X), f(X))]^2}{Var(f(X))}$$

Hence the percent reduction in variance is

$$100\frac{[Cov(g(X), f(X))]^2}{Var(f(X))Var(g(X))} = 100[Cor(g(X), f(X))]^2$$

To compute the constant c^*, we need Cov(g(X), f(X)) and Var(f(X)), but these parameters can be estimated if necessary, from a preliminary Monte Carlo experiment.

Question 3:

Use Monte Carlo integration with control variates to estimate

$$I = \int_0^1 \frac{1}{1+x}dx$$

```
n <- 10^5
a3 <- 0; b3 <- 1

x3 <- runif(n,a3,b3)

f3 <- function(x) 1/(1+x)

exp.estimate3 <- mean(sapply(x3,f3)) * (b3-a3)
exp.estimate3
```

```
## [1] 0.6930724
```

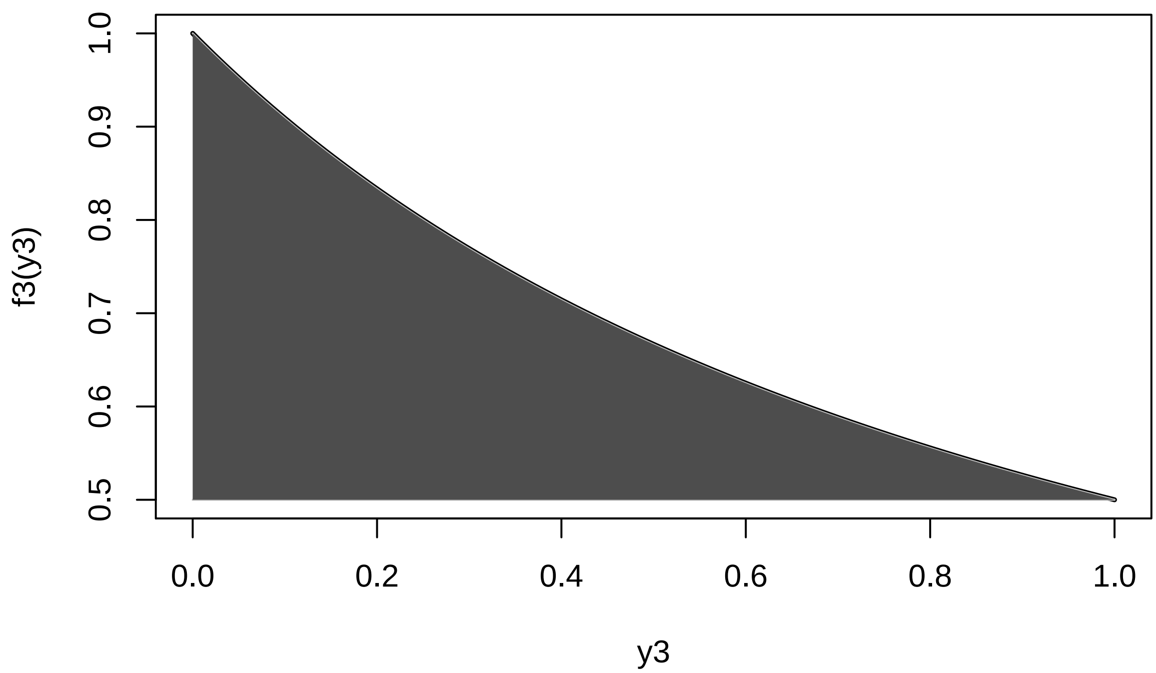
```
var.estimate3 <- mean(sapply(x3,f3)^2) - (mean(sapply(x3, f3)))^2
var.estimate3
```

```
## [1] 0.01949902
```

```
integrate(f3,a3,b3)$value
```

```
## [1] 0.6931472
```

```r
y3 <- seq(a3,b3,0.001)
y.low3 <- rep(1,times = length(y3))

plot(y3, f3(y3), type = "l",
     lwd = 2.5)

lines(y3,f3(y.low3), col = 'grey')
lines(y3,f3(y3), col = 'grey')
polygon(c(y3, rev(y3)), c(f3(y3), rev(f3(y.low3))),
     col = "grey30", border = NA)
```




```r
n <- 10^5
a3 <- 0; b3 <- 1

x3 <- runif(n,a3,b3)

g.cont <- function(x) 1/(1+x)

f.cont <- function(x) 1+x

# est of c*
```

```r
cor(g.cont(x3),f.cont(x3))
```

```
## [1] -0.984053
```

```r
c_star <- -cov(g.cont(x3),f.cont(x3)) / var(f.cont(x3))
c_star
```

```
## [1] 0.4767003
```

```r
expected.fu <- integrate(f.cont,a3,b3)$value

exp.cont.estimate1 <- mean(g.cont(x3) +
                             c_star * (f.cont(x3) - expected.fu)) * (b3 - a3)
exp.cont.estimate1
```

```
## [1] 0.6931249
```

```r
var.cont.estimate1 <- var(g.cont(x3) + c_star * (f.cont(x3) - expected.fu))
var.cont.estimate1
```

```
## [1] 0.0006179094
```

```r
result3 <- matrix(c(exp.estimate3,exp.cont.estimate1,
                    var.estimate3, var.cont.estimate1),nrow=2,
                  dimnames = list(c("classical approach","control variates"),
                                  c("estimated mean","variance")))
result3
```

```
##                    estimated mean     variance
## classical approach      0.6930724 0.0194990217
## control variates        0.6931249 0.0006179094
```

```r
var_reduction3 = 100 * ((var.estimate3-var.cont.estimate1)/var.estimate3)

cat("The control variates approach achived",
    round(var_reduction3,2), "%", "reduction in variance.")
```

```
## The control variates approach achived 96.83 % reduction in variance.
```

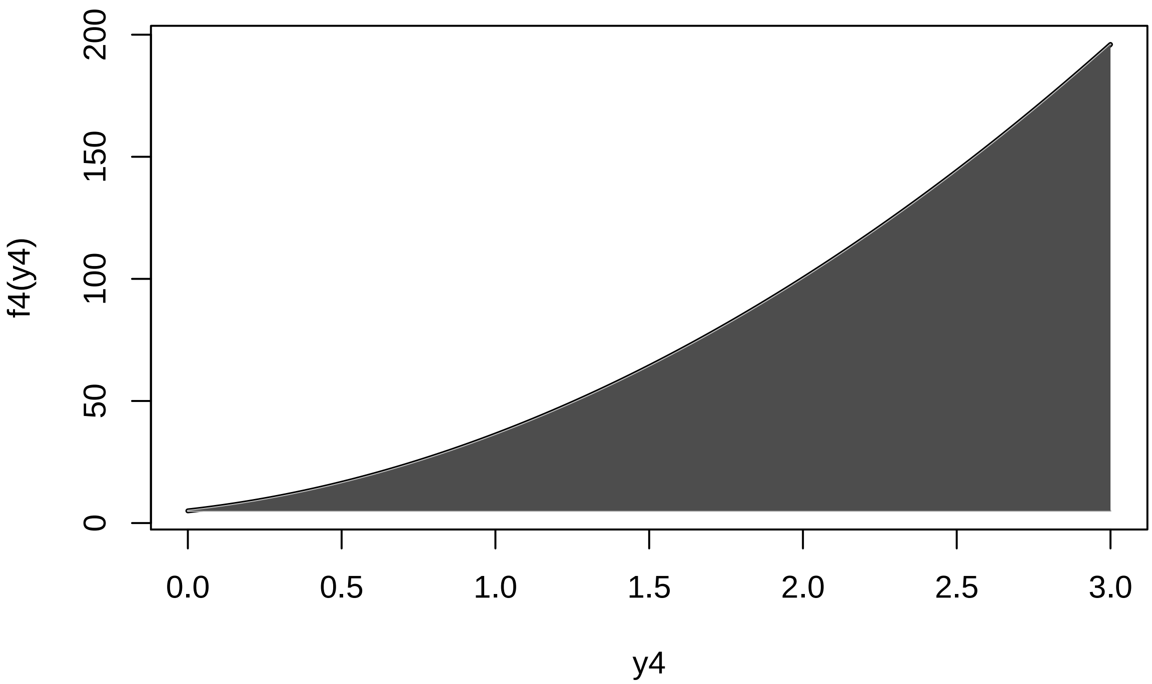
Question 4:

Use Monte Carlo integration with control variates to estimate

$$I = \int_0^3 \frac{e^{-2x}}{1+x^2} + (4x+2)^2 dx$$

```r
start3 <- proc.time()

f4 <- function(x) exp(-2*x) / (1+x^2) + (4*x+2)^2
a4 <- 0; b4 <- 3
n <- 10^5

x4 <- runif(n,a4,b4)

exp.estimate4 <- mean(sapply(x4,f4)) * (b4-a4)
# var(f4(x4))
var.estimate4 <- mean(sapply(x4,f4)^2) - (mean(sapply(x4, f4)))^2

run.time.2 <- proc.time() - start3
```

```r
y4 <- seq(a4,b4,0.001)
y.low4 <- rep(0,times = length(y4))

plot(y4, f4(y4), type = "l",
     lwd = 2.5)

lines(y4,f4(y.low4), col = 'grey')
lines(y4,f4(y4), col = 'grey')
polygon(c(y4, rev(y4)), c(f4(y4), rev(f4(y.low4))),
     col = "grey30", border = NA)
```




```r
start4 <- proc.time()
n <- 10^5
a4 <- 0; b4 <- 3

x4 <- runif(n,a4,b4)

g.cont.2 <- function(x) exp(-2*x) / (1+x^2) + (4*x+2)^2

f.cont.2 <- function(x) 1 / (1+x^2)

# est of c*
cor(g.cont.2(x4),f.cont.2(x4)) # high negative correlation
```

```
## [1] -0.880253
```

```r
c_star.2 <- -cov(g.cont.2(x4),f.cont.2(x4)) / var(f.cont.2(x4))
expected.fu.2 <-  (1/(b4-a4)) * integrate(f.cont.2,a4,b4)$value

exp.cont.estimate2 <- mean(g.cont.2(x4) +
                             c_star.2 * (f.cont.2(x4) - expected.fu.2)) * (b4 - a4)
var.cont.estimate2 <- var(g.cont(x4) + c_star.2 * (f.cont(x4) - expected.fu.2))
```

```r
run.time.cont.var.1 <- proc.time() - start4
```

```r
integrate(f4,a4,b4)$value
```

```
## [1] 228.3989
```

```r
result4 <- matrix(c(exp.estimate2,exp.ant.estimate2, exp.cont.estimate2,
                    round(var.estimate2,6), round(var.ant.estimate2,6),
                    round(var.cont.estimate2,6)),
                  nrow=3, dimnames = list(c("classical approach",
                                  "antithetic variables","control variates"),
                                  c("estimated mean","variance")))
result4
```

```
##                        estimated mean   variance
## classical approach           227.7281  3158.4526
## antithetic variables         228.2795   117.2559
## control variates             228.1966 21517.9905
```

```r
var_reduction4 = 100 * ((var.estimate4-var.cont.estimate2)/var.estimate4)
```

```r
run.time.1
```

```
##    user  system elapsed
##    0.53    0.00    0.53
```

```r
run.time.ant.1
```

```
##    user  system elapsed
##    0.28    0.02    0.31
```

```r
run.time.cont.var.1
```

```
##    user  system elapsed
##    0.07    0.01    0.10
```

Several Control Variates & Regression

The idea of combining unbiased estimators of the target parameter θ to reduce variance can be extended to several control variables. Again we want to use simulation to estimate $\theta = E(g(X))$. The corresponding control variate estimator is

Question 5:

Use Monte Carlo integration with several control variates to estimate

$$\hat{\theta}_c = g(X) + \sum_{i=1}^{k} c_i^*(f_i(X) - \mu_i)$$

where $\mu_i = E[f_i(X)]$, i= 1,...,k and

$$E[\hat{\theta}_c] = E[g(X)] + \sum_{i=1}^{k} c_i^*(f_i(X) - \mu_i) = \theta$$

The controlled estimate $\hat{\theta}_{c^*}$, and estimates for the optimal constants c_i^*, can be obtained by fitting a linear regression model.

$$I = \int_0^1 x^3 + 2x + 5dx$$

```
g5 <- function(x) x^3 + 2*x + 5

n <- 10^5

x5 <- runif(floor(n/3))

f5.1 <- function(x) x
f5.2 <- function(x) x^2
f5.3 <- function(x) x^3


cor(g5(x5),f5.1(x5)) # high negative correlation
```

```
## [1] 0.9909586
```

```
cor(g5(x5),f5.2(x5))
```

```
## [1] 0.9925245
```

```
cor(g5(x5),f5.3(x5))
```

```
## [1] 0.9617377
```

```
mu1 <- integrate(f5.1,0,1)$value
mu2 <- integrate(f5.2,0,1)$value
mu3 <- integrate(f5.3,0,1)$value

L <- lm(g5(x5) ~ f5.1(x5)+ f5.2(x5) +f5.3(x5))

estimate5 <- sum(L$coeff * c(1,mu1,mu2,mu3))
var.est5 <- summary(L)$sigma^2
```

Compare with control variates method

```r
g5 <- function(x) x^3 + 2*x + 5

n <- 10^5

x55 <- runif((n))

f55 <- function(x) x^3

# est of c*

cor(g5(x55),f55(x55))
```

```
## [1] 0.9618977
```

```r
c_star5 <- -cov(g5(x55),f55(x55)) / var(f55(x55))
c_star5
```

```
## [1] -2.863766
```

```r
estimate55 <- mean(g5(x55) +
                             c_star5 * (f55(x55) - integrate(f55,0,1)$value))

var.est55 <- var(g5(x55) + c_star5 * (f55(x55) - integrate(f55,0,1)$value))
```

```r
result5 <- matrix(c(estimate5,estimate55,
                    var.est5, var.est55),
                  nrow=2, dimnames = list(c("Several control variates",
                                            "control variates"),
                                   c("estimate","variance")))
result5
```

```
##                          estimate     variance
## Several control variates 6.250000 4.480496e-28
## control variates         6.250591 5.339270e-02
```