440

```
MC.CI = function(delta){
  n = length(delta)
  se = sqrt((1/n)*(mean(deltax^2)-mean(deltax)^2 ))
  mean(delta)+c(-1,1)*se*1.96
}
```

1 simple monte carlo

```
M=1000
x = matrix(rpois(25*M, 2), nrow = M, ncol = 25)
xbar = rowMeans(x)
deltax = (xbar-2)/(sqrt(2/25)) >= 1.645
# estimate of type 1 error rate
mean(deltax)
```

```
## [1] 0.068
```

```
MC.CI(deltax)
```

```
## [1] 0.05239663 0.08360337
```

importance sampling

```
x = matrix(rpois(25*M, 2.4653), nrow = M, ncol = 25)
xbar = rowMeans(x)
deltax = (xbar-2)/(sqrt(2/25)) >= 1.645
fx = dpois(rowSums(x),2*25)
gx = dpois(rowSums(x),2.4654*25)
deltax = (deltax*fx/gx)
mean(deltax)
```

```
## [1] 0.05995694
```

```
MC.CI(deltax)
```

```
## [1] 0.05488549 0.06502838
```

antithetic

```
hM = round(M/2)
x = matrix(runif(25*hM), nrow = hM, ncol = 25)
x1 = qpois(x,2)
x2 = qpois(1-x,2)

#concatenate 2 groups of data
x = rbind(x1, x2)
xbar = rowMeans(x)
deltax = (xbar-2)/(sqrt(2/25)) >= 1.645
mean(deltax)
```

```
## [1] 0.049
```

```r
MC.CI(deltax)
```

```
## [1] 0.03562036 0.06237964
```

control variate

```r
x = matrix(rpois(25*M, 2), nrow = M, ncol = 25)
xbar = rowMeans(x)
deltax = (xbar-2)/(sqrt(2/25)) >= 1.645

gx = rowSums(x)


cov.hat = sum((deltax-mean(deltax))*(gx-mean(gx))/(M*(M-1)))
var.hat = sum((gx-mean(gx))^2)/(M*(M-1))
vardelta.hat = sum((deltax-mean(deltax))^2)/(M*(M-1))

alpha = -cov.hat/var.hat

real.star=50

delta.cv = deltax + alpha*(gx-real.star)

var.cv = vardelta.hat+alpha^2*var.hat+2*cov.hat*alpha

theta.cv = mean(delta.cv)

theta.cv
```

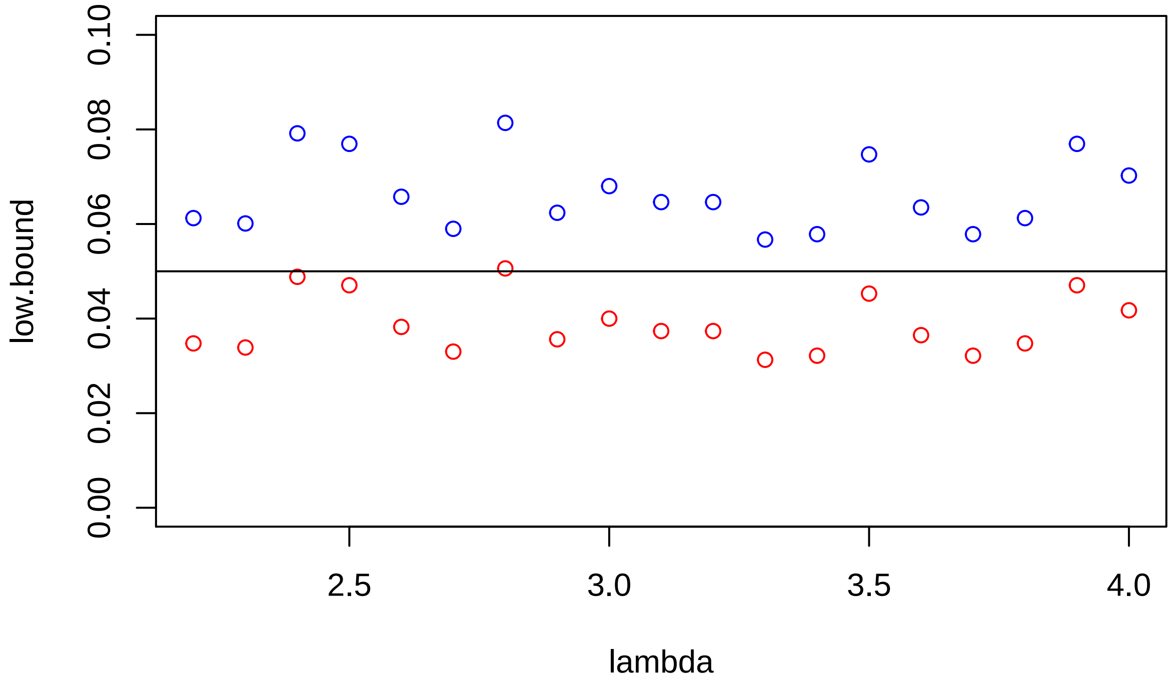
```
## [1] 0.0609209
```

```r
theta.cv+c(1,-1)*sqrt(var.cv)*1.96
```

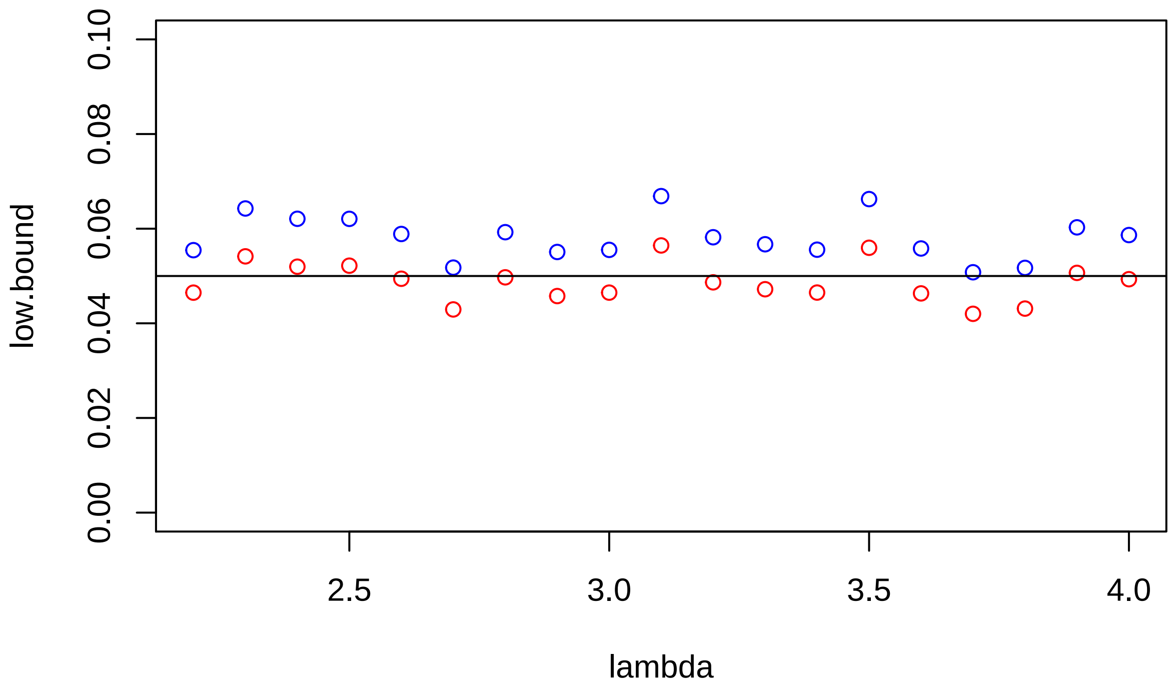
```
## [1] 0.07325799 0.04858382
```

b

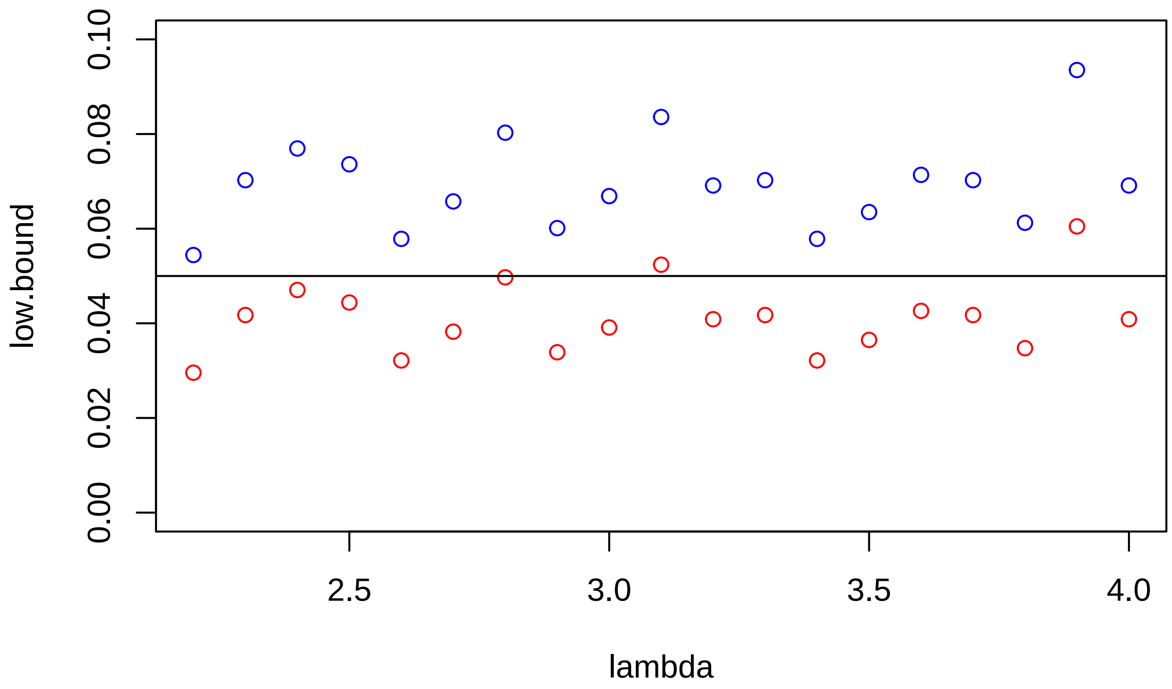
```r
# simple MC
lambda = seq(2.2,4,0.1)
low.bound=c()
up.bound=c()

set.seed(10086)

for (i in lambda){
  x = matrix(rpois(25*M, i), nrow = M, ncol = 25)
  xbar = rowMeans(x)
  deltax = (xbar-i)/(sqrt(i/25)) >= 1.645
  low.bound=c(low.bound, MC.CI(deltax)[1])
  up.bound=c(up.bound, MC.CI(deltax)[2])
}

plot(lambda, low.bound, ylim=c(0,0.1), col='red')
points(lambda, up.bound, col='blue')
abline(h=0.05)
```



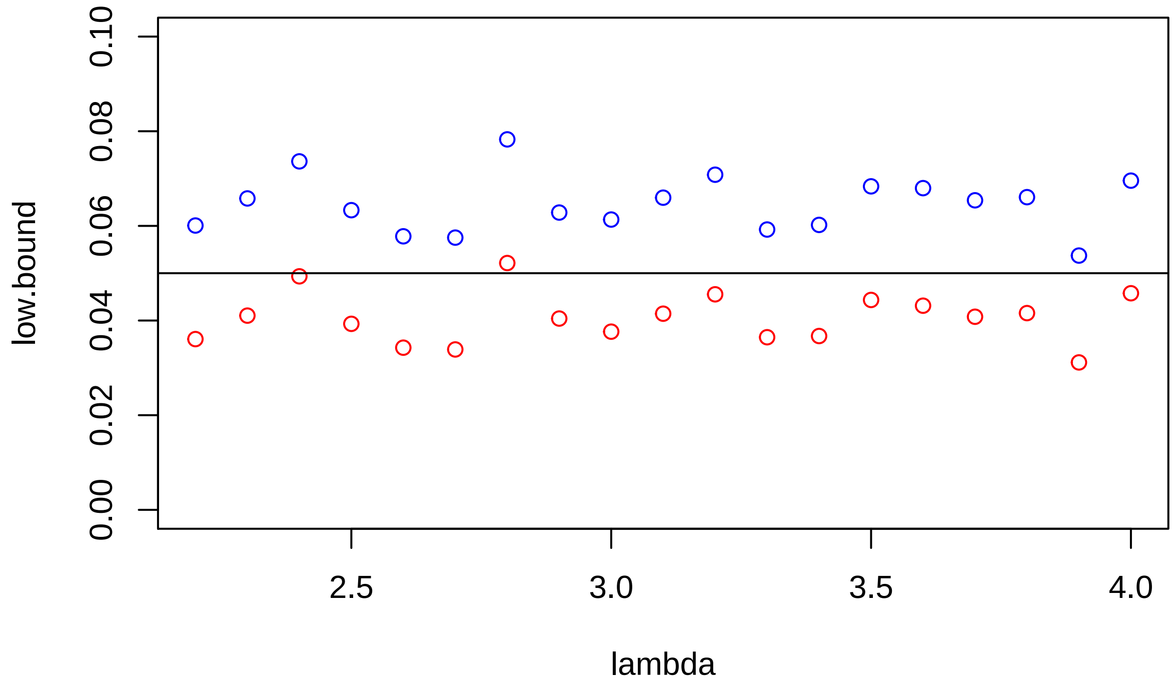
```r
#importance sampling
low.bound=c()
up.bound=c()

for (i in lambda){
  newlambda = 1.645*sqrt(i/25)+i
  x = matrix(rpois(25*M, newlambda), nrow = M, ncol = 25)
  xbar = rowMeans(x)
  deltax = (xbar-i)/(sqrt(i/25)) >= 1.645
  fx = dpois(rowSums(x),i*25)
  gx = dpois(rowSums(x),newlambda*25 )
  deltax = deltax*fx/gx
  low.bound=c(low.bound, MC.CI(deltax)[1])
  up.bound=c(up.bound, MC.CI(deltax)[2])
}

plot(lambda, low.bound, ylim=c(0, 0.1), col='red')
points(lambda, up.bound, col='blue')
abline(h=0.05)
```



```r
# antithetic
low.bound=c()
up.bound=c()

for (i in lambda){
  hM = round(M/2)
  x = matrix(runif(25*hM), nrow = hM, ncol = 25)
  x1 = qpois(x,i)
  x2 = qpois(1-x,i)

  #concatenate 2 groups of data
  x = rbind(x1, x2)
  xbar = rowMeans(x)
  deltax = (xbar-i)/(sqrt(i/25)) >= 1.645
  low.bound=c(low.bound, MC.CI(deltax)[1])
  up.bound=c(up.bound, MC.CI(deltax)[2])
}

plot(lambda, low.bound, ylim=c(0,0.1), col='red')
points(lambda, up.bound, col='blue')
abline(h=0.05)
```



```
# control variate
low.bound=c()
up.bound=c()

for (i in lambda){
  x = matrix(rpois(25*M, i), nrow = M, ncol = 25)
  xbar = rowMeans(x)
  deltax = (xbar-i)/(sqrt(i/25)) >= 1.645

  gx = rowSums(x)


  cov.hat = sum((deltax-mean(deltax))*(gx-mean(gx))/(M*(M-1)))
  var.hat = sum((gx-mean(gx))^2)/(M*(M-1))
  vardelta.hat = sum((deltax-mean(deltax))^2)/(M*(M-1))

  alpha = -cov.hat/var.hat

  real.star=25*i

  delta.cv = deltax + alpha*(gx-real.star)

  var.cv = vardelta.hat+alpha^2*var.hat+2*cov.hat*alpha

  theta.cv = mean(delta.cv)
  theta.cv
  CIs = theta.cv+c(-1,1)*sqrt(var.cv)*1.96

  low.bound=c(low.bound, CIs[1])
  up.bound=c(up.bound, CIs[2])
}

plot(lambda, low.bound, ylim=c(0,0.1), col='red')
points(lambda, up.bound, col='blue')
abline(h=0.05)
```



```
###
```