Monte Carlo Methods

Dennis Do

11/4/21

1)

```
m <- 1e4
x <- runif(m,0,pi)
g_x <- sin(x)
theta.hat <- mean(g_x)
theta.hat
```

```
## [1] 0.6369917
```

```
theta.hat + 1.96 * c(-1,1) * sd(g_x) / sqrt(m)
```

```
## [1] 0.6309660 0.6430175
```

2)

```
mc_pbeta <- function(q,a,b,m)
{
  x <- runif(m,0,q)
  g_r <- dbeta(x,a,b)
  return(q*mean(g_r))
}

mc_pbeta(.95,3,3,1000)
```

```
## [1] 1.003863
```

```
pbeta(.95,3,3)
```

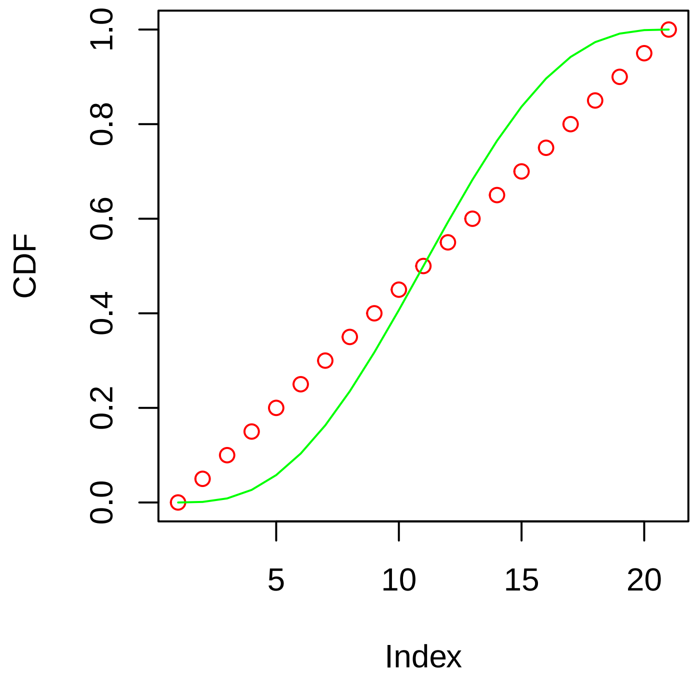
```
## [1] 0.9988419
```

3)

```
q <- seq(0,1, by = 0.05)
mc_pbeta(q,3,3,1000)
```

```
##  [1] 0.00000000 0.03917500 0.07835001 0.11752501 0.15670002 0.19587502
##  [7] 0.23505002 0.27422503 0.31340003 0.35257504 0.39175004 0.43092504
## [13] 0.47010005 0.50927505 0.54845005 0.58762506 0.62680006 0.66597507
## [19] 0.70515007 0.74432507 0.78350008
```

```
q1 <- pbeta(q,3,3)
plot(q, col = 'red', ylab = 'CDF')
lines(q1, col='green')
```



4)

```
g2d <- function(x,y) {
  g2dformula <- (exp(-sqrt(x^2+y^2)))*(1+cos(4*pi*sqrt(x^2+y^2)))
  return(g2dformula)
}
g2d(0,0)
```

```
## [1] 2
```

```
g2d(-1,1)
```

```
## [1] 0.358128
```

```
g2d(0,pi/8)
```

```
## [1] 0.8241773
```

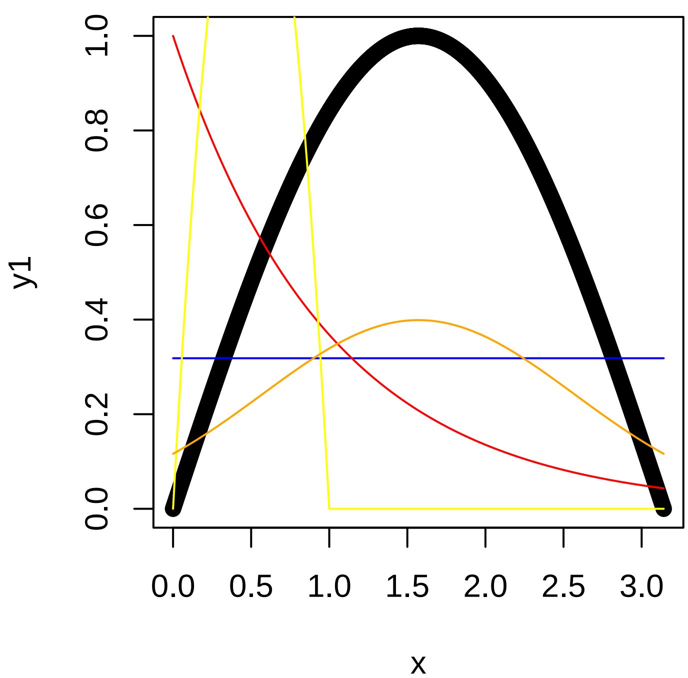
5)

```r
m <- 1e4
x <- runif(n = m, min = -1, max = 1)
y <- runif(n = m, min = -1, max = 1)
g_xy <- g2d(x,y) * 4
pt_est <- mean(g_xy)
pt_est
```

```
## [1] 1.943364
```

6)

```r
x = seq(0,pi,length = 1000)
y1 = sin(x)
plot(x,y1)
lines(x,rep(1/pi,1000), col='blue')
lines(x,exp(-x), col='red')
lines(x,dbeta(x,2,2), col ='yellow')
lines(x, dnorm(x,pi/2,1), col ='orange')
```



The importance function D with a mean of pi/2 and variance of 1 is similar to sin.

7)

```r
means <- pi/2; variance <- 1
n1 <- runif(1e4)
x_norm <- rnorm(n1,means,variance)
g_starsx <- sin(x_norm)/ dnorm(x_norm, means,variance)
mean(g_starsx)
```

```
## [1] -2.526346
```

```r
x1 <- mean(g_starsx)

a <- 2; b <- 2
n <- runif(1e4)
x_beta <- rbeta(n,a,b)
g_star_x <- sin(x_beta) / dbeta(x_beta,a,b)
mean(g_star_x)
```

```
## [1] 0.4745302
```

```r
x2 <- mean(g_star_x)
```

Beta has a narrower CI

8)

```r
m <- 1e4
J <- 4
theta <- rep(0,J)
var <- rep(0, J)
breaks <- seq(0, pi,length.out=5)
for(j in 1:4){
  t<- runif(n = m/J, min = breaks[j], max = breaks[j+1])
  g_t <- sin(t) * (breaks[J=1] - breaks[j])
  theta[j] <- mean(g_t)
  var[j] <- var(g_t) / (m/J)
}
pt_est_strat <- sum(theta)
se_strat <- sqrt(sum(var))
```

9) The standard error in stratified is reduced as stratum represents proportion to size of the population

10)

```r
t <- rgamma(1,10, scale=1/40)
p_z <- mean(t)
p_z
```

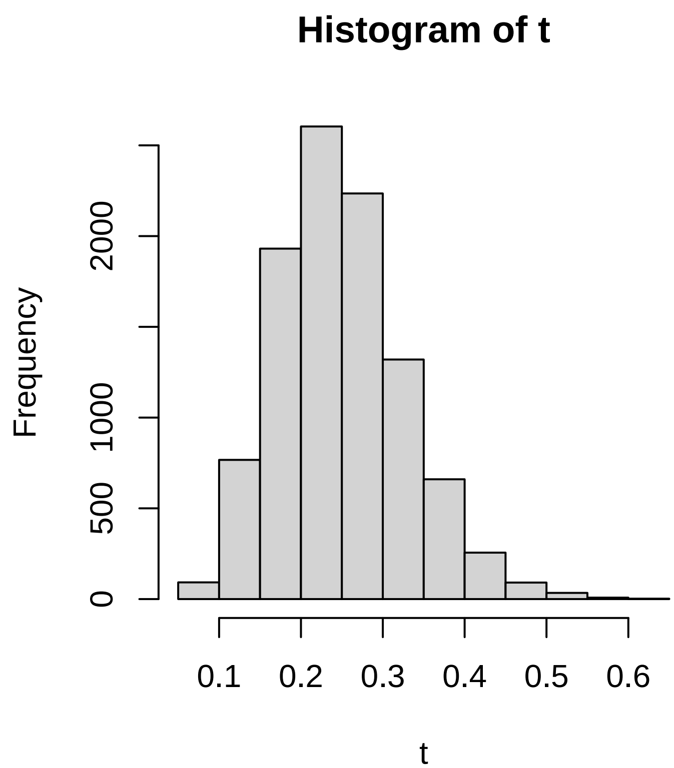
```
## [1] 0.2331408
```

11)

```r
m <- 1e4
t<- rgamma(m,10,scale = 1/40)
p_z <- mean(t)
p_z
```

```
## [1] 0.2493942
```

```r
hist(t)
```



12) The estimated probability is 25% for any given day, this is high.