Applied Stochastic Process

Assignment 1 - Solutions

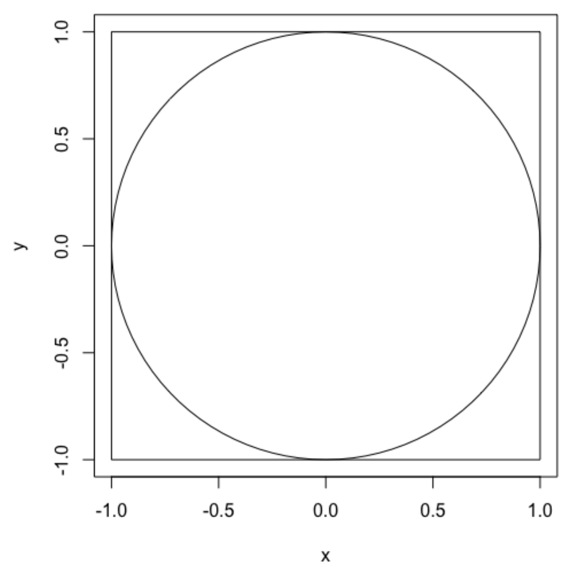
CHEUNG Ying Lun

11/6/2020

Question 1

Find the value of π through simulation.

To find the value of π, we first note that for a circle with with radius $r = 1$, its area is $A_c = \pi r^2 = \pi$. If we draw a square which four sides are all tangent to the circle, we get a square with length 2 and area $A_s = 4$.



To compute the value of π, we can imagin that we are playing darts, and the dart randomly lands on the square. The probability that the dart lands on the circle is given by $P(\text{within circle}) = \pi/4$. Therefore, we can perform a simulation by drawing the x- and y-axes randomly from the uniform distribution $\mathcal{U}(-1, 1)$, and find the probability

$$p = P(x^2 + y^2 \leq 1) = N^{-1} \sum_{i=1}^{N} 1(x_i^2 + y_i^2 \leq 1)$$

where $1(\cdot)$ is the indicator function. The value of π can be approximated by $\pi \approx 4p$.

To begin with, we clear the environment, set seed and initialize some parameters.

```
rm(list = ls())
graphics.off()
```

```r
set.seed(916)
NTrial <- 5000
```

Next, we generate `NTrial` uniformly distributed x_i and y_i, which indicate the x- and y-axes of the location where the i-th dart lands.

```r
X <- runif(NTrial, min = -1, max = 1)
Y <- runif(NTrial, min = -1, max = 1)
```

Finally, we compute the indicator function and approximate π as discussed above.

```r
g <- ( X^2 + Y^2 ) <= 1
pi_hat <- 4 * mean(g)
```

We report the approximation result below.

```r
print(paste0("The value of pi_hat is ", pi_hat))
```

```
## [1] "The value of pi_hat is 3.1168"
```

```r
print(paste0("The value of pi is ", pi))
```

```
## [1] "The value of pi is 3.14159265358979"
```

Question 2

Suppose $X \sim Geo(p)$ with $p = 0.2$. Find $\mathbb{E}[g(X)]$ through simulation, where

- $g(X) = |X|^3$.
- $g(X) = \cos(X)$.
- $g(X) = \exp(X)$.
- $g(X) = \log(X^4 + X^2)$

By the Law of Large Number, expectations can be approximate by the sample mean of independent samples of the random variable. Therefore, to find $\mathbb{E}[g(X)]$, we can generate `NTrial` random samples of $g(X)$, where X is random samples of $Geo(p)$. Again, we first clear the environment and set parameters.

```r
rm(list = ls())
set.seed(1625)
NTrial <- 5000
p <- 0.2
```

Next, we generate random samples $X \sim Geo(p)$ and perform the respective transformations. Note that in `R`, the density function is $P(X = x) = p(1-p)^x$ instead of $P(X = x) = p(1-p)^{x-1}$. In other words, it measures the number of failures before the first success instead of the number of trials until the first success. Therefore, we have to add one back if we mean the latter.

```r
X <- rgeom(NTrial, p)+1
g1 <- X^3
g2 <- cos(X)
g3 <- exp(X-1)
g4 <- log(X)
```

Finally, we can analyze the statistical properties of $g(X)$.

```r
summary(g1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     8.0    64.0   559.6   343.0 68921.0
```

```r
summary(g2)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.99996 -0.65364 -0.14550 -0.06817  0.54030  0.98870
```

```r
summary(g3)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 1.000e+00 3.000e+00 2.000e+01 4.743e+13 4.030e+02 2.354e+17
```

```r
summary(g4)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.6931  1.3863  1.2518  1.9459  3.7136
```

Question 3

Let $X_1, X_2, \ldots$ be a sequence of independent Poisson random variables with parameter $\lambda = 0.2$. Let

$$S_N = \sum_{i=1}^{N} X_i.$$

1. **Find the mean μ_N and variance σ_N^2 of S_N when $N = 10, 20, 100, 1000$.**
2. **Plot the distribution of $\sigma_N^{-1}(S_N - \mu_N)$ when $N = 10, 20, 100, 1000$.**

To find the mean and variance of S_N, we have to generate `NTrial` samples of S_N., each of which is simply the sum of N randomly drawn Poisson distributed random variables. Again, we begin with the initialization of parameters.

```r
rm(list = ls())
NTrial <- 5000
N <- c(10, 20, 100, 1000)
lambda <- 0.2
S <- matrix(nrow = NTrial, ncol = 4)
```

Next, we generate N Poisson distributed random numbders by `rpois(N[i], lambda)`, calculate their sum by `sum(.)`, and create `NTrial` random copies by `replicate(.)`.

```r
for (i in 1:4){
  S[ ,i] <- replicate(NTrial, sum(rpois(N[i], lambda)))
}
```

We then compute the means and standard deviations of S_N as the sample means and sample standard deviations of each column in `S`.

```r
(mu <- colMeans(S))
```

```
## [1]   1.9978   3.9566  19.9538 200.0118
```

```r
(sigma <- apply(S, 2, sd))
```

```
## [1]  1.411876  1.952608  4.447429 14.048556
```

We also plot the histogram of each column in `S` and compare them with the normal density function. We observe that as N increases, they fit better, supporting the central limit theorem.

```r
x <- seq(-5, 5, 0.01)
par(mfrow = c(2,2))
for (i in 1:4) {
  hist((S[ , i] - mu[i]) / sigma[i], freq = FALSE, main = paste0("N = ",N[i]))
```

```
  lines(x, dnorm(x))
}
```