Lab Quiz

Dustin Leatherman

5/21/2020

1

In probability theory and statistics, Gumbel distribution is used to model maximum or minimum of a number of samples of various distributions. The CDF of a the Gubmbel Distribution is

$$F(y) = e^{-e^{-y}}$$

Simulate 1000 samples from Gumbel distribution, print out the first 5. (set the seed for uniform random numbers as 1).

```
set.seed(123456)

n <- 1000
gumbel <- function(y) exp(-exp(-y))

x <- runif(n)
y <- gumbel(x)

head(y, n = 5)
```

```
## [1] 0.6374206 0.6245740 0.5085421 0.4913183 0.4981899
```

2

You and your friend is going to have dinner together. There are two popular restaurants nearby: Chef Ping and Sogongdong. The waiting times to be served in these two restaurants are X_1 and X_2 respectively. X_i's are IID **exponential** randome variables with mean to be 10 minutes. You look at the apps and choose the one with the shorter waiting time. Use simple Monte Carlo method to estimate average waiting time each day, with the estimation error no larger than 0.1 minutes.

```
init.n <- 1000
rate <- 1/10

restaurant1 <- rexp(init.n, rate)
restaurant2 <- rexp(init.n, rate)

# since there are two restaurants, we pick whichever has the lower waiting time.
# pmin gets the pairwise minimum of two vectors
chosenRestaurant <- pmin(restaurant1, restaurant2)

init.waitingTime.sigma <- sd(chosenRestaurant)

est.n <- ceiling((2.58 * 1.1 * init.waitingTime.sigma / 0.1)^2)

wait <- rexp(est.n, rate)
```

```
avg.wait <- mean(wait)

lcl <- avg.wait - 1.96 * sd(wait) / est.n
ucl <- avg.wait + 1.96 * sd(wait) / est.n
```

The number of samples needed to run to achieve an error of 0.1 minutes is 24309.

The average wait time via computed by Monte Carlo methods is 10.0508 minutes.

The Lower and Upper bounds of this estimation are 10.0499 and 10.0516 respectively.

3

Consider the case where an asset price, $S(t)$, with an initial price \$120 is modeled by a geometric Brownian Motion:

$$S(t) = 120e^{-0.23125t+0.75B(t)},\ 0 \le t < \infty$$

where $B(t)$ is a standard Brownian motion. Here the risk-free interest rate is 5% and the stock volatility is 75%. A discretely monitored Asian Arithmetic mean call option with strike price \$120 monitored weekly (52 times per year) and expiring in one year has a discounted payoff of

$$payoff = max(\frac{1}{52}(\sum_{j=1}^{52} S(j/52)) - 120, 0)e^{-0.05}$$

With the same parameters and 10^4 sample paths, estimate the price of the Asian Arithmetic mean call option.

```
n <- 10^4
s0 <- 120
sigma <- 0.75
r <- 0.05
t <- 1
d <- 52
delta <- t/d
grid <- seq(delta, t, length.out = d)

#create a matrix to store asset prices
S <- matrix(rep(0, n * (d + 1)), nrow = n)

#generate nxd pseudo-random normal numbers
x <- matrix(rnorm(n * d), nrow = n)

#generate n sample paths of Brownian Motion
BM <- sqrt(delta) * t(apply(x, 1, cumsum))

S <- cbind(rep(s0, n), s0 * exp(sweep(sigma * BM, MARGIN = 2, (r - sigma^2 / 2) * grid, '+')))

# Calculate payoff for each simulation with pairwise maximums
payoffs <- pmax(apply(S, 1, mean) - s0, 0) * exp(-r)
```

```r
# mean of payoffs gives the estimated call price
est.price <- mean(payoffs)
```

The estimated call price price off an Arithmetic Mean Call Option with r = 0.05, strike price = $120 over the course of a year is $21.25