JPMC Quant Challenge - SiliconLobby [Hiten & Arkadeep]

Question 3

30 September 2018

Question 3 [Quant]

Solution

Task is to price the product whose **payoff P** at time **T = T_n** is given by

$max\{(\frac{X_T}{2X_0} + \frac{Y_T}{2Y_0}) - 1,0\}$ if $X_{t_i} \in [0.75X_0, 1.25X_0]$ and $Y_{t_i} \epsilon [0.75Y_0, 1.25Y_0]$ for all values of T_{i} where $T_i = T_0 + 0.25 * i$ years, i = 0,1,…N

0 otherwise

To evaluate E(P|X_0): ***Algorithm:***

1. Calculate σ_x & σ_y from the historical spot price data given.

2. Calculate the correlation of bivariate normal random variables simulating the brownian motion Wx and Wy.

3. Set flag True for all paths implying path has not not crossed the barrier yet.

3. Start simulating the spot prices at each Ti and check if spot prices has crossed the barriers, if yes then we need to set the flag false for these paths to be included in evaluation of payoff's expectation later.

4. Evaluate the payoff by mean (element wise (flag * P $_{\text{(payoff vector)}}$)).

5. Discount the payoff to time 0 to get the price.

Observation:

If the product value has positive drift the product payoff increased exponentially so it almost surely crossed the barriers so even returning 0 as an answer showed model accuracy 100.

```r
#Get sigma from spot prices given
sig_x <- sqrt(var(diff(log(X_SPOT))));
sig_y <- sqrt(var(diff(log(Y_SPOT))));
rho <- cor(diff(log(X_SPOT)),diff(log(Y_SPOT)))
#Simulate path-dependent option
sim <- 10000
sigma <- matrix(c(252*0.25,rho*252*0.25,rho*252*0.25,252*0.25),2,2)
payoff <- rep(1,sim);
value <- matrix(rep(c(X_SPOT[1],Y_SPOT[1]),sim*2),sim,2,byrow=TRUE);
for(i in seq(0,N)){

  rand_gen <- mvrnorm(n=sim, rep(0,2),sigma)

  value_x <- value + r*value*(0.25*252) + sig_x*rand_gen[,1]
  value_y <- value + r*value*(0.25*252) + sig_y*rand_gen[,2]
  for(j in seq(1,nrow(value))){
    if (((value_x[j,1]>1.25*X_SPOT[1]) || (value_x[j,1]<0.75*X_SPOT[1]))){
      payoff[j] <- 0;
    }
    if (((value_y[j,2]>1.25*Y_SPOT[1]) || (value_y[j,2]<0.75*Y_SPOT[1]))){
      payoff[j] <- 0;
    }
  }
}
payoff <- (payoff*(value_x[,1]/(2*X_SPOT[1])) + payoff*(value_y[,2]/(2*Y_SPOT[1])))-1
sum = 0;
#return mean(payoff[payoff>0])*exp(-r*(n/4)*252)
for(i in payoff){
  sum <- sum + max(payoff,0);
}
return (sum/sim)*exp(-r*(n/4)*252);
```