FE515 2022A Assignment 3

Yufu Liao

04/21/2023

Question 1: (50 points)

1.1

Download option prices of ticker ^VIX for all expiration dates and name it VIX.options

```
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 4.2.3
```

```
## Loading required package: xts
```

```
## Warning: package 'xts' was built under R version 4.2.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.2.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Warning: package 'TTR' was built under R version 4.2.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(xts)

VIX.options <- getOptionChain("^VIX", NULL)
```

1.2

Download the current price (last quote price) for ^VIX

```
(VIX.current.price <- getQuote("^VIX")$Last)
```

```
## [1] 16.77
```

1.3

For calls and puts of VIX.options at each expiration calculate the average of Bid and Ask. Create a new column named ‘Price’ to contain the result.

```
for (i in 1:length(VIX.options)) {
  VIX.options[[i]]$calls$Price <- (VIX.options[[i]]$calls$Bid + VIX.options[[i]]$calls$Ask) * 0.
5
  VIX.options[[i]]$puts$Price <- (VIX.options[[i]]$puts$Bid + VIX.options[[i]]$puts$Ask) * 0.5
}
```

1.4

For calls and puts of VIX.options at each expiration, add a column of InTheMoney, which takes value TRUE when it is in-the-money, and FALSE otherwise. Compare it to ITM column to check your results. (Hint. A call option is in-the-money when its strike is less than the current price of underlying. A put option is in-the-money if its strike is greater than the current price of underlying. And the current price of underlying is the last quote price from 1.2)

```
for (i in 1:length(VIX.options)) {
  VIX.options[[i]]$calls$InTheMoney <- ifelse(VIX.options[[i]]$calls$Strike < VIX.current.price,
TRUE, FALSE)
  VIX.options[[i]]$puts$InTheMoney <- ifelse(VIX.options[[i]]$puts$Strike < VIX.current.price, T
RUE, FALSE)
}
```

1.5

For calls and puts of VIX at each expiration, delete all the fields except Strike, Bid, Ask, Price, and In-The-Money, and save them in .csv files with the format “VIXdata2021-09- 26Exp2021-10-08puts.csv”, here 2021-09-26 should be replaced by the date you download the data, and 2021-10-08 should be replaced by the date of expiration.

```
ex <- names(VIX.options)
for (i in 1:length(VIX.options)) {
  VIX.options[[i]]$calls <- VIX.options[[i]]$calls[c("Strike", "Bid", "Ask", "Price", "InTheMoney")]
  VIX.options[[i]]$puts <- VIX.options[[i]]$puts[c("Strike", "Bid", "Ask", "Price", "InTheMoney")]
  write.csv(VIX.options[[i]]$puts, file = paste("VIXdata", Sys.Date(), "Exp", ex[i], "puts.csv", sep = ""))
}
ex
```

```
##  [1] "Apr.26.2023" "May.03.2023" "May.10.2023" "May.17.2023" "May.24.2023"
##  [6] "Jun.21.2023" "Jul.19.2023" "Aug.16.2023" "Sep.20.2023" "Oct.18.2023"
## [11] "Nov.15.2023" "Dec.20.2023"
```

Question 2

2.1

Using Monte-Carlo Simulation to estimate the put option price using S0 = 100, K = 100, T = 1, σ = 0.2, r = 0.05, you can use number of steps n = 252 and number of paths m = 10000

```
S0 <- 100
K <- 100
T1 <- 1
sigma <- 0.2
r <- 0.05
func.mc <- function(S0, K, T1, sigma, r) {
  n = 252
  m = 10000
  h <- T1 / n
  S.vec <- rep(S0, m)
  Z <- matrix(rnorm(n * m), nrow = n)
  for (i in 1:n) {
  S.vec <- S.vec + r * S.vec * h + sigma * S.vec * Z[i,] * sqrt(h)
  }
  return(exp(-r * T1) * mean(pmax(100 - S.vec, 0)))
}

func.mc(S0, K, T1, sigma, r)
```

```
## [1] 5.81996
```

2.2

Implement Black-Scholes formula for pricing the put option

```
func.bs <- function(S0, K, T1, sigma, r) {
  d1 <- (log(S0 / K) + (r + 0.5 * sigma ^ 2) * T1) / (sigma * sqrt(T1))
  d2 <- d1 - sigma * sqrt(T1)
  return (-S0 * pnorm(-d1) + exp(-r * T1) * K * pnorm(-d2))
}
func.bs(S0, K, T1, sigma, r)
```

```
## [1] 5.573526
```