A6

Mingjia Zhu

10/28/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# 1.a fixed strike look_back option ---------------------------------------

S0=100;dt=1/252;T=1;Path_Length=round(T/dt,0);
r=0.05;
E=rnorm(Path_Length,0,1)
sigma=0.3
strike=95
S= matrix(NA, Path_Length+1,1);
S[1] = S0;
for (j in 1 : Path_Length) { # length of path

  S[j+1] = S[j] + r*S[j]* dt + sigma *S[j]*sqrt(dt)* E[j]

}

S=(S[100:length(S)]);
payoff = max(S)-strike;


MC_Lookback_Fix = function(Type, S0,strike, r, sigma, expire_days, n, m){
  # Type: Put or Call
  # S0: Spot price
  # strike is the predetermined strike price
  # r : interest rate
  # sigma: volatility
  # n: discrete steps frequency
  # m: number of simulations

  dt = 1/n# usually there are 252 trading days
  Path_Length = expire_days # how many days left before expiry it is integer
  payoff_sum = 0      # initialize the sum of payoffs that you will collect over simulated scenarios

  for (i in 1:m){
```

```r
    S= matrix(NA, Path_Length,1)
    S[1] = S0
    E=rnorm(Path_Length,0,1)# generate random noise

    for (j in 1 : Path_Length) { # length of path

      S[j+1] = S[j] + r*S[j]* dt + sigma *S[j]*sqrt(dt)* E[j]

    }

    if (Type == "c" ){ payoff = max(S)-strike }  # Call

    else if (Type == "p" ){ payoff = strike - min(S) }  # Put

    payoff_sum =  payoff_sum + payoff # Accumulate the payoffs from all simulated scenarios.

  }
  OptionValue = (payoff_sum * exp(-r*(expire_days/n)))/m
  return(OptionValue)

}


#try to calculate this kind of call option with expiry days of 125 tradings days
MC_Lookback_Fix("c",100,95,0.05,0.3,125,252,100)
```

```
## [1] 21.06741
```

```r
# 1.b fixed stike look_vack option with time window -----------------------

#we can modify the standard fixed strike lookback option by adding a time window slice of the S path

MC_Lookback_Fix_Partial = function(Type, S0,strike, r, sigma, expire_days,waiting_days, n, m){
  # Type: Put or Call
  # S0: Spot price
  # strike is the predetermined strike price
  # r : interest rate
  # sigma: volatility
  # expire_days is how long this option lasts
  # waiting_days is how many days should be waiting before this option really starts
  # n: discrete steps frequency
  # m: number of simulations

  dt = 1/n# usually there are 252 trading days
  Path_Length = expire_days # how many days left before expiry it is integer
  payoff_sum = 0      # initialize the sum of payoffs that you will collect over simulated scenarios

  for (i in 1:m){

    S= matrix(NA, Path_Length+1,1)
    S[1] = S0
    E=rnorm(Path_Length,0,1)# generate random noise
```

```r
    for (j in 1 : Path_Length) { # length of path

      S[j+1] = S[j] + r*S[j]* dt + sigma *S[j]*sqrt(dt)* E[j]

    }


    #slice the S
    S = S[(waiting_days+1):length(S)]



    #after slicing there exist the possibility that max(s)-strike is negative
    #add the max(0,payoff) to cpnstrain the payoff to be positive
    if (Type == "c" ){ payoff = max(0,(max(S)-strike)) }  # Call

    else if (Type == "p" ){ payoff = max(0,(strike - min(S))) }  # Put

    payoff_sum =  payoff_sum + payoff # Accumulate the payoffs from all simulated scenarios.

  }
  OptionValue = (payoff_sum * exp(-r*(expire_days/n)))/m
  return(OptionValue)

}

MC_Lookback_Fix_Partial("c",100,95,0.05,0.3,125,100,252,10000)
```

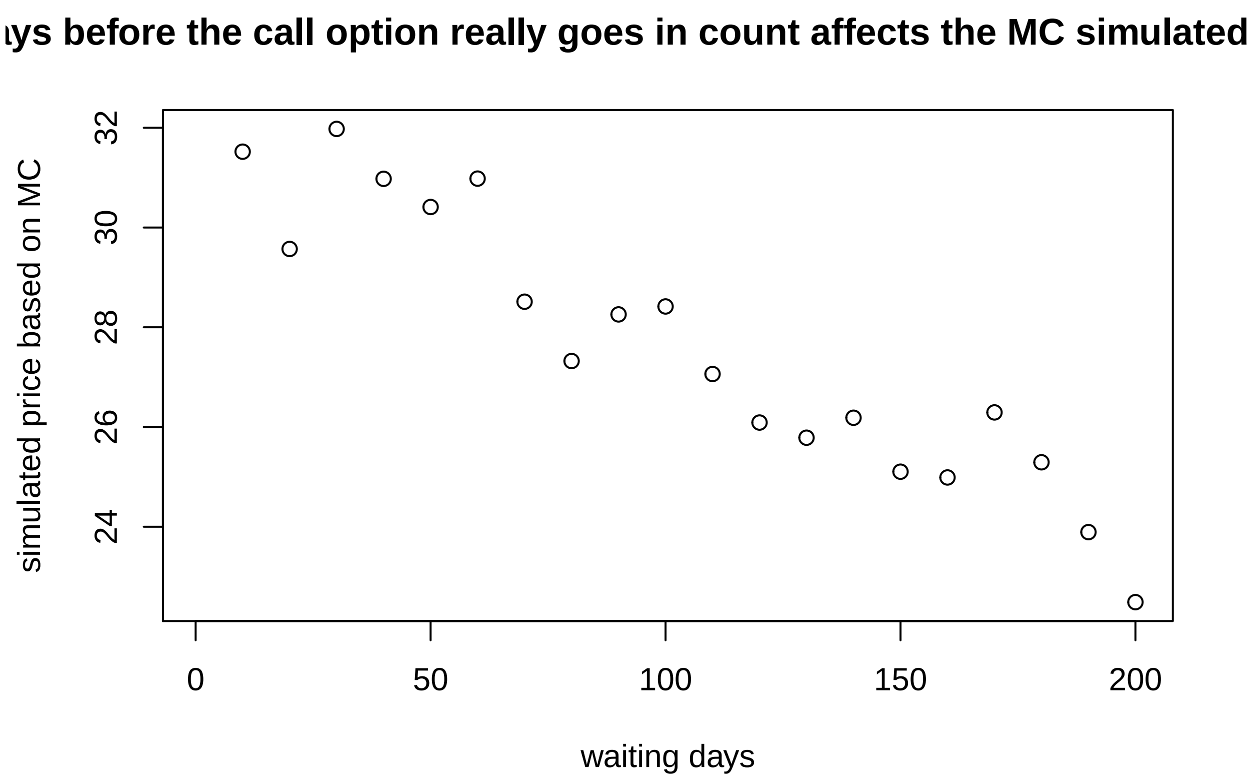
```
## [1] 16.63625
```

```r
# show how The partial-time fixed strike lookback option is cheaper than a similar standard fixed strik


b=seq(0,200,by = 10);
length(b) ;
```

```
## [1] 21
```

```r
#when waiting time is 0 it is equivalent to previous function
Record=matrix(NA,length(b),1)
# when set the simulation number to be 100, we can still see some fluctuation
# setting simulation number to be more than 1000, we can see a generally monetone declining line with f
for (i in seq(0,200,by = 10)){
  Record[i]=MC_Lookback_Fix_Partial("c",100,95,0.05,0.3,252,i,252,1000)
}
plot(Record,main="how waiting days before the call option really goes in count affects the MC simulated
     xlab="waiting days",ylab="simulated price based on MC")
```




```
MC_Lookback_Fix("c",100,95,0.05,0.3,252,252,1000)
```

```
## [1] 30.71263
```

```
library(quantmod)
```

```
## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

## Loading required package: TTR

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

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
# get data
ticker_list = c("^VIX","TSLA")

stock_prices <- NULL

for (ticker in ticker_list){
  stock_prices <- cbind(stock_prices, getSymbols.yahoo(ticker,
                                                       from = '2020-10-14', periodicity= 'daily', auto.a
}

stock_prices$VIX.Close[1];
```

```
##            VIX.Close
## 2020-10-14      26.4
```

```r
stock_prices$TSLA_IV = NA;

df=stock_prices;
# use the previous IV data as initials
#460    -0.37%  59.05   59.28   59.5    58.55   8.1 16.06%  176 974 74.24%

df[1,"TSLA_IV"] = 74.24;
# 7 locates the OCT 22
for (i in 2:7){
  df[i,"TSLA_IV"] = as.numeric(df[i-1,"TSLA_IV"] )*0.95/2+as.numeric(df[i,"VIX.Close"])*2.9/2+10*rnorm(


}
df;
```

```
##            VIX.Close TSLA.Close  TSLA_IV
## 2020-10-14     26.40     461.30 74.24000
## 2020-10-15     26.97     448.88 80.97253
## 2020-10-16     27.41     439.67 68.02964
## 2020-10-19     29.18     430.83 80.33019
## 2020-10-20     29.35     421.94 59.54658
## 2020-10-21     28.65     422.64 67.65784
## 2020-10-22     28.11     425.79 88.37345
## 2020-10-23     27.55     420.63       NA
## 2020-10-26     32.46     420.28       NA
## 2020-10-27     33.35     424.68       NA
```

```r
# 100 contract covers 100*100 shares of stock
library(fOptions)
```

```
## Loading required package: timeDate

## Loading required package: timeSeries

##
## Attaching package: 'timeSeries'
```

```
## The following object is masked from 'package:zoo':
##
##     time<-

## Loading required package: fBasics

##
## Attaching package: 'fBasics'

## The following object is masked from 'package:TTR':
##
##     volatility
```

```
delta1=GBSCharacteristics(TypeFlag = "c",S=461.7,X=460,Time = 65/365,r = 0.008,b=0.008,sigma = 0.7424)$d
# holding the call option is a kind of long delta strategy
# generally we should sell some underlying asset to make the whole portfolio delta hedged

#for the fisrt day of 14th OCT

call_delta = 100*delta1*100;
stock_position = -call_delta;
# call_delta is also the number of share we need to short sell to make the whole portfolio delta neutra

df$TSLA_position=NA;
df;
```

```
##            VIX.Close TSLA.Close  TSLA_IV TSLA_position
## 2020-10-14     26.40     461.30 74.24000            NA
## 2020-10-15     26.97     448.88 80.97253            NA
## 2020-10-16     27.41     439.67 68.02964            NA
## 2020-10-19     29.18     430.83 80.33019            NA
## 2020-10-20     29.35     421.94 59.54658            NA
## 2020-10-21     28.65     422.64 67.65784            NA
## 2020-10-22     28.11     425.79 88.37345            NA
## 2020-10-23     27.55     420.63       NA            NA
## 2020-10-26     32.46     420.28       NA            NA
## 2020-10-27     33.35     424.68       NA            NA
```

```
for (i in 1:7){
  S1=as.numeric(df[i,"TSLA.Close"])
  sigma1 = as.numeric(df[i,"TSLA_IV"])
  delta1=GBSCharacteristics(TypeFlag = "c",S=S1,X=460,Time = ((66-i)/365),r = 0.008,b=0.008,sigma = sig

  df[i,"TSLA_position"] = -100*delta1*100;

}
df;
```

```
##            VIX.Close TSLA.Close  TSLA_IV TSLA_position
## 2020-10-14     26.40     461.30 74.24000     -5675.739
## 2020-10-15     26.97     448.88 80.97253     -5404.221
## 2020-10-16     27.41     439.67 68.02964     -4945.225
```

```
## 2020-10-19     29.18     430.83 80.33019     -4887.369
## 2020-10-20     29.35     421.94 59.54658     -4099.910
## 2020-10-21     28.65     422.64 67.65784     -4337.477
## 2020-10-22     28.11     425.79 88.37345     -4855.567
## 2020-10-23     27.55     420.63       NA            NA
## 2020-10-26     32.46     420.28       NA            NA
## 2020-10-27     33.35     424.68       NA            NA
```

```
#generate daily position change
df$position_change=NA;
df$position_change[1]=df$TSLA_position[1];

for (i in 2:7){
  df$position_change[i]=as.numeric(df$TSLA_position[i])-as.numeric(df$TSLA_position[i-1]);
}
df$position_change;
```

```
##            position_change
## 2020-10-14     -5675.73902
## 2020-10-15       271.51825
## 2020-10-16       458.99553
## 2020-10-19        57.85586
## 2020-10-20       787.45988
## 2020-10-21      -237.56753
## 2020-10-22      -518.08996
## 2020-10-23              NA
## 2020-10-26              NA
## 2020-10-27              NA
```

```
#this vector shows the daily position change of TSLA stock to make the whole portfolio risk neutral
#short the stock can get cash inflow and buy the stock will cause cash outflow

CF=as.vector(df$position_change*df$TSLA.Close*(-1));
CF=CF[1:7];
sum(CF);
```

```
## [1] 2258349
```

```
cat("the delta hedged strategy generates the total cash inflow of ",sum(CF),"\n")
```

```
## the delta hedged strategy generates the total cash inflow of  2258349
```

```
Cost_to_buyback=-df$TSLA_position[7]*-df$TSLA.Close[7]
cat("the whole portfolio still shorts the ",-df$TSLA_position[7],"shares of TSLA and they worths",-df$TS
```

```
## the whole portfolio still shorts the  4855.567 shares of TSLA and they worths -2067452
```

```
cat("the money makes from the short selling stock parts",as.numeric(sum(CF)+Cost_to_buyback),"theoritica
```

```
## the money makes from the short selling stock parts 190897 theoritically it should equals the money lo
```

```r
cat("each call option predicted to lose",as.numeric(sum(CF)+Cost_to_buyback)/10000,"\n")
```

```
## each call option predicted to lose 19.0897
```

```r
#theoritically it should equals the money loses on the price of call option
```