FA_Assignment6_Duo_Zhou

Duo Zhou

8/3/2020

A stock index is currently at 810 and has volatility of 20%. The risk-free rate is 5% per year. Assume that dividend yield equals 1%.

```
library(RQuantLib)
```

1. Price European 6-month call option with strike 800 using EuropeanOption() from RQuantLib.

```
#Assign Inputs
Underlying <- 810
Volatility <- .20
RFRate <- .05
DivYield <- .01
Strike <- 800
Maturity <- .5
# Use EuropeanOption to calculate Call value and greeks
call <- EuropeanOption(type="call", underlying = Underlying, strike=Strike,
                        dividendYield=DivYield, riskFreeRate=RFRate, maturity=Maturity,
                        volatility=Volatility)
call$value
```

```
## [1] 58.73271
```

2. Calculate the same premium manually using the formulas on the last slide of the lecture notes.

$C = S_0N(d_1) - (Kexp(-RT))N(d_2)$

$d_1 = (ln(S_0/K) + (R + .5\sigma^2)T)/(\sigma\sqrt(T))$

$d2 = d_1 - (\sigma\sqrt(T))$

```
d1=(log(Underlying/Strike) + (RFRate + .5*(Volatility^2))*Maturity)/(Volatility*sqrt(Maturity))

d2 = d1 - (Volatility*sqrt(Maturity))

C = (Underlying * pnorm(d1,0,1)) - ((Strike*exp(-RFRate * Maturity))* pnorm(d2,0,1))

C
```

```
## [1] 61.25612
```

Think how dividend yield should affect option price. Experiment with the function EuropeanOption() with zero or non-zero dividend yield and find how the Black-Scholes formula on slide 17 should be modified for dividend yield.

```r
# Use EuropeanOption to calculate Call value  using dvidendYield =0
call.div0 <- EuropeanOption(type="call", underlying = Underlying, strike=Strike,
                            dividendYield=0, riskFreeRate=RFRate, maturity=Maturity,
                            volatility=Volatility)
call.div0$value
```

```
## [1] 61.25612
```

We can see that the Black Scholes Merton formula in the last slide of the lecture did not take divident yields into consideration. As devident yields increase, the call option price will decrease.

The following is the Merton extension on the original formula.

$C = S_0 exp(-qT)N(d_1) - (Kexp(-RT))N(d_2)$

$d_1 = (ln(S_0/K) + (R - q + .5\sigma^2)T)/(\sigma\sqrt{(}T))$

$d2 = d_1 - (\sigma\sqrt{(}T))$

Where q is the divident yield

Let us apply the new formula with divident yield = 0.01

```r
d1n=(log(Underlying/Strike) + (RFRate - DivYield +
.5*(Volatility^2))*Maturity)/(Volatility*sqrt(Maturity))

d2n = d1n - (Volatility*sqrt(Maturity))

Cn = (Underlying *exp(-DivYield*Maturity)*pnorm(d1n,0,1)) -
  ((Strike*exp(-RFRate * Maturity))* pnorm(d2n,0,1))

Cn
```

```
## [1] 58.73271
```

We can see the new formula generates call option price that is identical to our answer in part 1.

3. Calculate the premium of put option with the same expiry and same strike using put-call parity

Put-Call parity without divident yield: $Put = Call + Kexp(-RT) - S_0$

Put-Call parity with divident yield: $Put = Call + Kexp(-RT) - S_0 exp(-qT)$

```r
# put price with divident Yield = 0.01
put.parity <- Cn + (Strike * exp(-RFRate*Maturity)) - (Underlying*exp(-DivYield*Maturity))
put.parity
```

```
## [1] 33.02053
```

Verify the Pu-Call parity formula using EuropeanOption()

```r
put <- EuropeanOption(type="put", underlying = Underlying, strike=Strike,
                        dividendYield=DivYield, riskFreeRate=RFRate,
                      maturity=Maturity, volatility=Volatility)

put$value
```

```
## [1] 33.02053
```