Module 5.6: Implied Option Parameters
R Commentary

See *Implied GBM Option Parameters Test.R*. The functions are provided in a separate file, *GBMOVM and Extended Functions.R*.

Unlike other modules, we integrate our discussion of different implied parameters within the R code comments.

Implied stock price

Implied stock price is the model stock value that results in the market option price ("Market" denotes observed option market price and not the theoretical "Model" option price). Thus, with implied stock price we have

$$C_{Market} = S_{Model} e^{-\delta T} N\left(d_1\right) - X e^{-rT} N\left(d_2\right) \tag{5R.6.1}$$

$$P_{Market} = X e^{-rT} N\left(-d_2\right) - S_{Model} e^{-\delta T} N\left(-d_1\right) \tag{5R.6.2}$$

$$d_1 = \frac{\ln\left(\frac{S_{Model}}{X}\right) + \left(r - \delta + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \tag{5R.6.3}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{5R.6.4}$$

We illustrate solving for the implied firm value when stock is viewed as an option on the firm. Equity can be viewed as a call option on the firm value (*FV*). The duration of the firm debt is the time to maturity (*T*) and debt par value (*DPV*) will be equivalent to the strike price. Assuming the underlying instrument is the firm value, then the equity value (*EV*) can be represented as follows:

$$EV_{Market} = FV_{Model} N\left(d_1\right) - DPV e^{-rT} N\left(d_2\right), \tag{5R.6.5}$$

where

$$N\left(d\right) = \int_{-\infty}^{d} \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx, \text{ (area under the standard cumulative normal distribution up to d)} \tag{5R.6.6}$$

$$d_1 = \frac{\ln\left(\frac{FV_{Model}}{DPV}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \text{ , and} \tag{5R.6.7}$$

$$d_2 = d_1 - \sigma\sqrt{T}\,. \tag{5R.6.8}$$

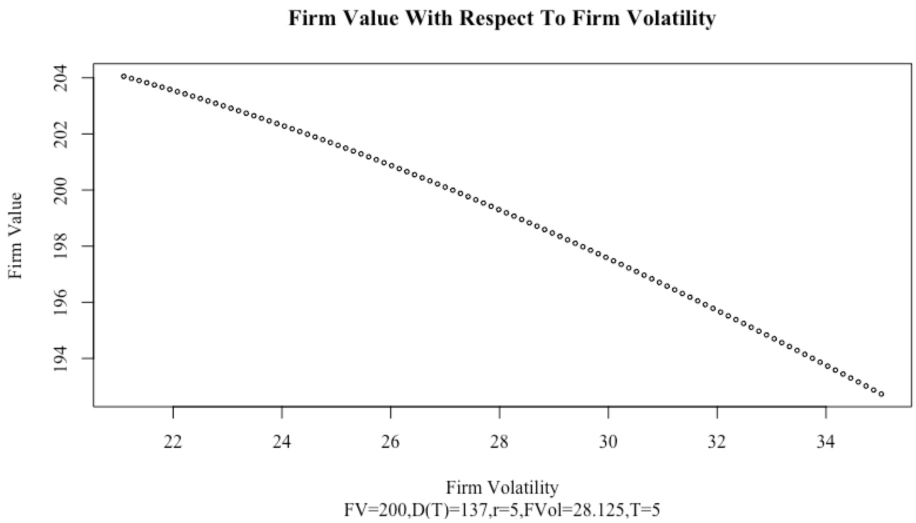
Note that volatility here is the firm volatility.

The following snippet of R code produces an implied firm value of $200.

```
GBMInputData$StockPrice <- -99    # Firm value
GBMInputData$StrikePrice <- 137.0 # Debt par value
GBMInputData$InterestRate <- 5.0  # Risk free rate
GBMInputData$DividendYield <- 0.0
GBMInputData$TimeToMaturity <- 5.0  # Years to debt maturity
GBMInputData$Volatility <- 27.125   # Volatility of firm
GBMInputData$Type <- 1 # Call option
inputEquityValue = 100.0  # Equity value
ImpliedFirmValue = GBMDYOptionImpliedStockPrice(GBMInputData, inputEquityValue)
ImpliedFirmValue
```

Figure 5R.6.1 illustrates the sensitivity of firm value to changes in firm volatility.

Figure 5R.6.1. Illustration of relationship between firm value and firm volatility within option framework



Other implied parameters

We briefly review solving for each of the remaining implied parameters. Although we may not have a good reason for solving for these parameters at this time, quantitative finance challenges often result in seeking unusual, implied parameters for a variety of reasons.

Implied strike price

Implied strike price is the model strike value that results in the market option price ("Market" denotes observed option market price and not the theoretical "Model" option price). Thus, with implied strike price we have

$$C_{Market} = Se^{-\delta T}N(d_1) - X_{Model}e^{-rT}N(d_2) \tag{5R.6.9}$$

$$P_{Market} = X_{Model}e^{-rT}N(-d_2) - Se^{-\delta T}N(-d_1) \tag{5R.6.10}$$

$$d_1 = \frac{\ln\left(\frac{S}{X_{Model}}\right) + \left(r - \delta + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \tag{5R.6.11}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{5R.6.12}$$

In this case, an investor may want to know the strike price that will result from the budgeted amount of funds available for a particular option trade. For example, the R code illustrated below results in a strike price of approximately \$116.11. Thus, the investor would have to purchase a call option about 16% out-of-the-money. Note that the `NewCallValue` is \$8.0.

```
# Test data for strike price
GBMInputData$StockPrice = 100
GBMInputData$StrikePrice = -99
GBMInputData$InterestRate = 5.0
GBMInputData$DividendYield = 0.0
GBMInputData$Volatility = 30.0
GBMInputData$TimeToMaturity = 1
GBMInputData$Type = 1
inputOptionValue = 8.0
GBMInputData$ImpliedLowerBound <- 0.0
GBMInputData$ImpliedUpperBound <- 1000
ImpliedCallStrikePrice = GBMDYOptionImpliedStrikePrice(GBMInputData,
  inputOptionValue)
```

```
GBMInputData$StrikePrice <- ImpliedCallStrikePrice
NewCallValue = GBMOptionValue(GBMInputData)
ImpliedCallStrikePrice; NewCallValue
```

Implied interest rate

Implied interest rate is the model interest rate that results in the market option price ("Market" denotes observed option market price and not the theoretical "Model" option price). Thus, with implied interest rate we have

$$C_{Market} = Se^{-\delta T} N\left(d_1\right) - Xe^{-r_{Model}T} N\left(d_2\right) \tag{5R.6.13}$$

$$P_{Market} = Xe^{-r_{Model}T} N\left(-d_2\right) - Se^{-\delta T} N\left(-d_1\right) \tag{5R.6.14}$$

$$d_1 = \frac{\ln\left(\frac{S}{X}\right) + \left(r_{Model} - \delta + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \tag{5R.6.15}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{5R.6.16}$$

In this case, an investor may want to know how sensitive call prices are to changes in interest rates. For example, the R code illustrated below illustrates that interest rates have to rise 5.89% for the call price to rise 1%.

```
# Interest rates and dividend yields can be zero or negative, sensitive to
#  extreme boundaries
# Test data for interest rates
GBMInputData$StockPrice = 100
GBMInputData$StrikePrice = 100
GBMInputData$InterestRate = -99.0
GBMInputData$DividendYield = 0.0
GBMInputData$Volatility = 30.0
GBMInputData$TimeToMaturity = 1
GBMInputData$Type = 1
GBMInputData$ImpliedLowerBound <- -10
GBMInputData$ImpliedUpperBound <- 10
inputNewOptionValue <- inputOptionValue*1.01
ImpliedCallInterestRate = GBMDYOptionImpliedInterestRate(GBMInputData,
  inputNewOptionValue)
GBMInputData$InterestRate <- ImpliedCallInterestRate
NewCallValue = GBMOptionValue(GBMInputData)
PCCallValue = 100 * (NewCallValue - inputOptionValue)/inputOptionValue
PCInterestRate = 100 * (ImpliedCallInterestRate - inputInterestRate) /
  inputInterestRate
ImpliedCallInterestRate; PCInterestRate; PCCallValue
```

Implied dividend yield

Implied dividend yield is the model interest rate that results in the market option price ("Market" denotes observed option market price and not the theoretical "Model" option price). Thus, with implied dividend yield we have

$$C_{Market} = Se^{-\delta_{Model}T} N\left(d_1\right) - Xe^{-rT} N\left(d_2\right) \tag{5R.6.17}$$

$$P_{Market} = Xe^{-rT} N\left(-d_2\right) - Se^{-\delta_{Model}T} N\left(-d_1\right) \tag{5R.6.18}$$

$$d_1 = \frac{\ln\left(\frac{S}{X}\right) + \left(r - \delta_{Model} + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \tag{5R.6.19}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{5R.6.20}$$

It is well-known that the implied dividend yield for equity index options are seasonal. That is, dividends in the U.S. are paid quarterly and somewhat in the same general time periods. Thus, some short-dated options have a low expected dividend yield (not over a heavy dividend-paying period) while other short-dated options have a high expected dividend yield(over a heavy dividend-paying period). Thus, if one had all the other inputs, it is straightforward to compute the implied dividend yield.

Implied time to maturity

Implied time to maturity is the model time to maturity that results in the market option price ("Market" denotes observed option market price and not the theoretical "Model" option price). Thus, with implied time to maturity we have

$$C_{Market} = Se^{-\delta T_{Model}} N\left(d_1\right) - Xe^{-rT_{Model}} N\left(d_2\right) \tag{5R.6.21}$$

$$P_{Market} = Xe^{-rT_{Model}} N\left(-d_2\right) - Se^{-\delta T_{Model}} N\left(-d_1\right) \tag{5R.6.22}$$

$$d_1 = \frac{\ln\left(\frac{S}{X}\right) + \left(r - \delta + \frac{\sigma^2}{2}\right)T_{Model}}{\sigma\sqrt{T_{Model}}} \tag{5R.6.23}$$

$$d_2 = d_1 - \sigma\sqrt{T_{Model}} \tag{5R.6.24}$$

One could imagine having a budget for hedging and a desire to protect the downside risk at some threshold level (implying a particular X). Given the budget, one could solve for the time to maturity that results in spending this amount of funds. With put options, there are two possible solutions. For some distant maturity, the value of a European-style option begins to decline due to the present value effect.

5.6 Implied GBM Option Parameters Test.R (Selected Excerpts and Output)

The inputs are relatively standard. The unique issue is setting the upper and lower boundaries. If set too wide, you may get unintended consequences and if set too narrow, you may miss the solution.

```
GBMInputData$Type = -1        # Put
# PutValue from GBMOptionValue function computed above
inputOptionValue = PutValue
ImpliedPutVolatility = GBMDYOptionImpliedVolatility(GBMInputData,
  inputOptionValue)
ImpliedPutStockPrice = GBMDYOptionImpliedStockPrice(GBMInputData,
  inputOptionValue)
ImpliedPutStrikePrice = GBMDYOptionImpliedStrikePrice(GBMInputData,
  inputOptionValue)
ImpliedPutTimeToMaturity = GBMDYOptionImpliedTimeToMaturity(GBMInputData,
  inputOptionValue)
ImpliedPutInterestRate = GBMDYOptionImpliedInterestRate(GBMInputData,
  inputOptionValue)
ImpliedPutDividendYield = GBMDYOptionImpliedDividendYield(GBMInputData,
  inputOptionValue)
PutValue; ImpliedPutVolatility; ImpliedPutStockPrice; ImpliedPutStrikePrice
[1] 9.354197
[1] 30
[1] 100
[1] 100
# Note: Put time to maturity has multiple solutions and rate range too large
ImpliedPutTimeToMaturity; ImpliedPutInterestRate; ImpliedPutDividendYield
[1] 20.29694
[1] NA
[1] 5.972726e-05
# Retry with narrower range
# Interest rates and dividend yields can be zero or negative, sensitive to
#  extreme boundaries
GBMInputData$ImpliedLowerBound = 0
GBMInputData$ImpliedUpperBound = 10
ImpliedPutInterestRate = GBMDYOptionImpliedInterestRate(GBMInputData,
  inputOptionValue)
```

```
ImpliedPutTimeToMaturity = GBMDYOptionImpliedTimeToMaturity(GBMInputData,
  inputOptionValue)
GBMInputData$ImpliedLowerBound = inputImpliedLowerBound
GBMInputData$ImpliedUpperBound = inputImpliedUpperBound
ImpliedPutTimeToMaturity; ImpliedPutInterestRate
[1] 0.9999998
[1] 4.999988
```

Note that the implied time to maturity for the put is 20.29694 years although the inputted number was 1.0. For put options there are up to two time to maturities that solve for the current price. Recall the maximum payout is the strike price; hence, with a longer time to maturity the discount effect will eventually dominate the time value effect.

GBM*OVM and Extended Functions.R*

There are numerous ways to handle solving for embedded functions. We illustrate the optimize function within the stats package. First, we build a function that computes the squared error based on some test implied volatility. Second, we solve for the implied volatility that minimizes this test function. Finally, we test whether the solution found is within a tolerable error range (0.01 here). Note we leave abs() in the return function to remind us that an alternative approach would be to base the test function on absolute value rather than squared values.

```
#
# Implied volatility function: May need to pass upper bound as input
#
GBMDYOptionImpliedVolatility <- function(B, inputOptionValue){
  TestFunctionGBMDYOptionImpliedVolatility<-function(testImpliedVolatility,B,
    inputOptionValue){
      B$Volatility = testImpliedVolatility
      return( abs(inputOptionValue - GBMOptionValue(B))^2 )
  }
  solution = optimize(TestFunctionGBMDYOptionImpliedVolatility, B,
    inputOptionValue, interval = c(B$ImpliedLowerBound, B$ImpliedUpperBound),
    tol = .Machine$double.eps^0.25)
  ImpliedVolatility = solution$minimum
  B$Volatility = ImpliedVolatility
  Difference = inputOptionValue - GBMOptionValue(B)
  if (abs(Difference) < 0.01)return(ImpliedVolatility)
  else return(NA)
}
```