binomial.option.pricing.models.R

HAG

2020-03-08

```r
# FIN659 - Binomial Option Pricing Models
# Adil Gokturk

# set the working directory and check it!
setwd("~/Desktop/Spring2020/FIN659/Assignments/hw9")
getwd()
```

```
## [1] "/Users/HAG/Desktop/Spring2020/FIN659/Assignments/hw9"
```

```r
# Load the libraries
library(qrmtools)
```

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

```r
library(fOptions) # used this one
```

```
## Loading required package: timeDate

## Loading required package: timeSeries

##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:qrmtools':
##
##     returns

## Loading required package: fBasics
```

```r
library(jrvFinance)
library(derivmkts) # and this one
```

```
##
## Attaching package: 'derivmkts'

## The following object is masked from 'package:jrvFinance':
##
##     duration
```

```r
library(OptionPricing)

# Notes
# Black-Scholes uses continues time rather than discrete

# The binomial tree method is particularly appropriate
# when we're trying to price American options.
```

```r
# Black-Scholes-Merton Model
# Black-Scholes-Merton is useful for pricing only European options.

# But the CRR ( Cox-Ross-Rubenstein) also known as binomial tree
# allows us to work backwards through a tree and modify
# the value at each node if the intrinsic value at that point is
# greater than the discounted expected payoff.

# I used Week9 Asynchronous's example to test R packages accuracy

s=50 # Price of the underlying asset
k=48 #  Strike price
v=0.30 # volatility -sigma
r=0.01 #    Annual continuously-compounded risk-free interest rate
tt=0.50 # Time to maturity in years
d=0 # Dividend yield, annualized, continuously-compounded



# ACCURATE MODEL I
# I used derivmkts R Library

# Black Scholes Call option pricing model
bscall(s, k, v, r, tt, d)
```

```
## [1] 5.344193
```

```r
## To check accuracy
# following returns the same price as previous
assetcall(s, k, v, r, tt, d) - k*cashcall(s, k, v, r, tt, d)
```

```
## [1] 5.344193
```

```r
#Black Scholes Put option pricing model
bsput(s, k, v, r, tt, d)
```

```
## [1] 3.104792
```

```r
## return option prices for multiple  strikes prices!
# very helpfull
bsput(s, k=40:60, v, r, tt, d)
```

```
##  [1]  0.6789755  0.8593617  1.0714436  1.3173928  1.5990277  1.9177762
##  [7]  2.2746551  2.6702640  3.1047924  3.5780402  4.0894467  4.6381277
## [13]  5.2229172  5.8424128  6.4950211  7.1790029  7.8925160  8.6336549
## [19]  9.4004865 10.1910816 11.0035412
```

```r
# Black Scholes call/put option prices
bscall(s, k, v, r, tt, d) # Black-Scholes call  price
```

```
## [1] 5.344193
```

```r
bsput(s, k, v, r, tt, d) #  Black-Scholes put  price
```

```
## [1] 3.104792
```

```r
# # Also tryied some other function of the package
# the prices of binary options
```

```
# that pay one share (the asset options)
assetcall(s, k, v, r, tt, d)
```

```
## [1] 31.31506
```

```
cashcall(s, k, v, r, tt, d)
```

```
## [1] 0.5410598
```

```
assetput(s, k, v, r, tt, d)
```

```
## [1] 18.68494
```

```
cashput(s, k, v, r, tt, d)
```

```
## [1] 0.4539527
```

```
# Binomial European CAL option pricing
# I calculated w9 example's up and down values

# Define UP and DOWN values
(up <- exp(v*sqrt(tt/2))) # 1.1618
```

```
## [1] 1.161834
```

```
(dn <- exp(-v*sqrt(tt/2))) # 0.8607
```

```
## [1] 0.860708
```

```
# European CAL option
binomopt(s, k, v, r, tt, d, nstep = 2, american = FALSE,
         putopt = FALSE,specifyupdn = TRUE,
         jarrowrudd = FALSE, up = up, dn = dn,
         returntrees = FALSE,
         returngreeks = FALSE)
```

```
##    price
## 5.292261
```

```
# European PUT option
binomopt(s, k, v, r, tt, d, nstep = 2, american = FALSE,
         putopt = TRUE,specifyupdn = TRUE,
         jarrowrudd = FALSE, up = up, dn = dn,
         returntrees = FALSE,
         returngreeks = FALSE)
```

```
##   price
## 3.05286
```

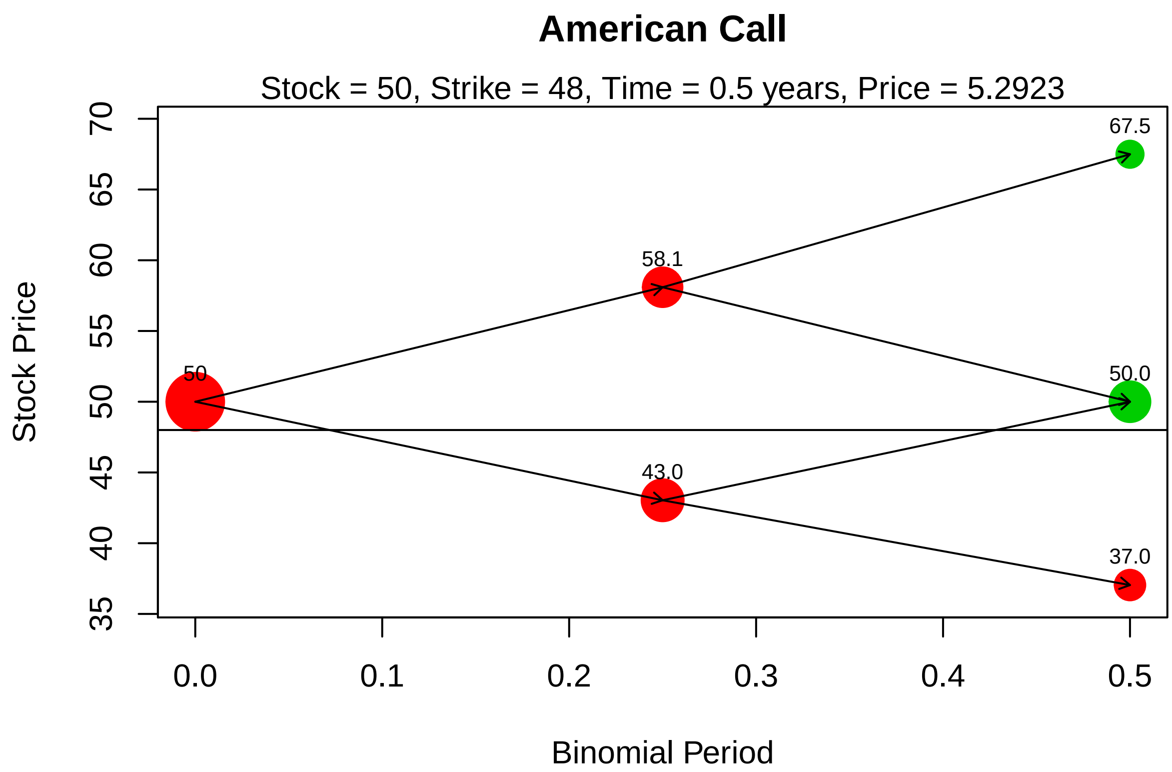
```
# Binomial American (CRR "Cox-Ross-Rubenstein Model") CALL Option
binomopt(s, k, v, r, tt, d, nstep = 2, american = TRUE,
         putopt = FALSE,specifyupdn = TRUE,
         jarrowrudd = FALSE, up = up, dn = dn,
         returntrees = FALSE,
         returngreeks = FALSE)
```

```
##    price
## 5.292261
```

```r
# Binomial American PUT option pricing
binomopt(s, k, v, r, tt, d, nstep = 2, american = TRUE,
         putopt = TRUE,specifyupdn = TRUE,
         jarrowrudd = FALSE, up = up, dn = dn,
         returntrees = FALSE,
         returngreeks = FALSE)
```

```
##   price
## 3.05286
```

```r
# Let's plot the  American CAL option stock Tree
# Again we use  binomial opion pricing (crr model)
# To plot different options manipulate "putop, american and crr" variables!
binomplot(s, k, v, r, tt, d, nstep = 2,
          putopt = FALSE, american = TRUE,
          plotvalues = TRUE, plotarrows = TRUE,
          drawstrike = TRUE, pointsize = 4, ylimval = c(0,0),
          saveplot = FALSE, saveplotfn = 'binomialplot.pfd',
          crr = TRUE, jarrowrudd = FALSE,
          titles = TRUE,
          specifyupdn = TRUE,
          up = up, dn = dn,
          returnprice = FALSE,
          logy = FALSE) # If TRUE, y-axis is plotted on a log scale
```




```r
###################################

## Second model
## fOptions R package
# The generalized Black-Scholes CALL Option Price
```

```r
# b is the annualized cost-of-carry rate
# if no dividend b=r(Annual continuously-compounded risk-free interest rate)
GBSOption(TypeFlag = "c", S = s, X = k, Time = tt,
          r = r, b = r, sigma = v,
          title = NULL, description = NULL)
```

```
##
## Title:
##  Black Scholes Option Valuation
##
## Call:
##  GBSOption(TypeFlag = "c", S = s, X = k, Time = tt, r = r, b = r,
##      sigma = v, title = NULL, description = NULL)
##
## Parameters:
##           Value:
##  TypeFlag c
##  S        50
##  X        48
##  Time     0.5
##  r        0.01
##  b        0.01
##  sigma    0.3
##
## Option Price:
##  5.344188
##
## Description:
##  Sun Mar  8 13:10:00 2020
```

```r
# The generalized Black-Scholes PUT Option Price
# no dividend b=r
GBSOption(TypeFlag = "p", S = s, X = k, Time = tt,
          r = r, b = r, sigma = v,
          title = NULL, description = NULL)@price
```

```
## [1] 3.104787
```

```r
# to compare multiple strike price for BS call or put option
GBSOption(TypeFlag = "p", S = s, X = 40:48, Time = tt,
          r = r, b = r, sigma = v,
          title = NULL, description = NULL)@price
```

```
## [1] 0.6789723 0.8593582 1.0714412 1.3173929 1.5990306 1.9177810 2.2746590
## [8] 2.6702635 3.1047866
```

```r
# CRR ( Cox-Ross-Rubenstein) Binomial Option
# European CALL option
# Accurate result
CRRBinomialTreeOption(TypeFlag = "ce", S = s, X = k,
                      Time = tt,
                      r = r, b = r,
                      sigma = v,n = 2)@price
```

```
## [1] 5.292261
```

```r
# European PUT option
# Accurate result
CRRBinomialTreeOption(TypeFlag = "pe", S = s, X = k,
                      Time = tt,
                      r = r, b = r,
                      sigma = v,n = 2)@price
```

```
## [1] 3.05286
```

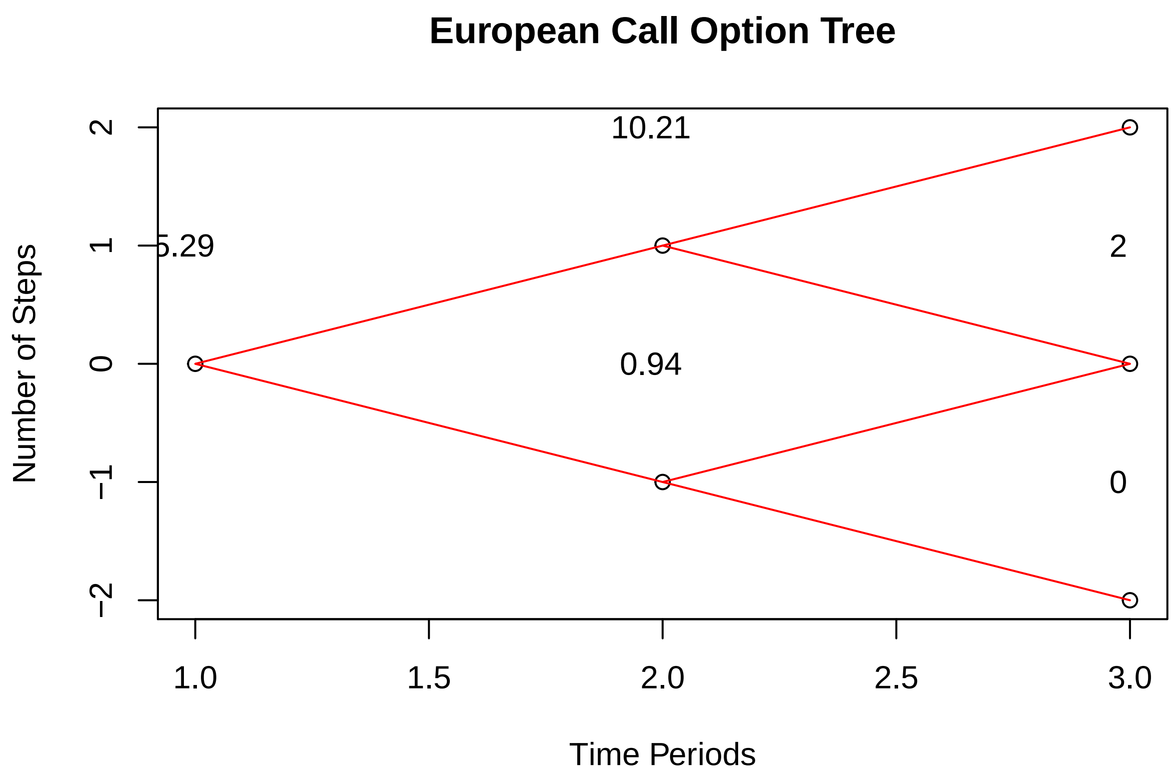
```r
# American CALL option
# Accurate result
CRRBinomialTreeOption(TypeFlag = "ca", S = s, X = k,
                      Time = tt,
                      r = r, b = r,
                      sigma = v,n = 2)@price
```

```
## [1] 5.292261
```

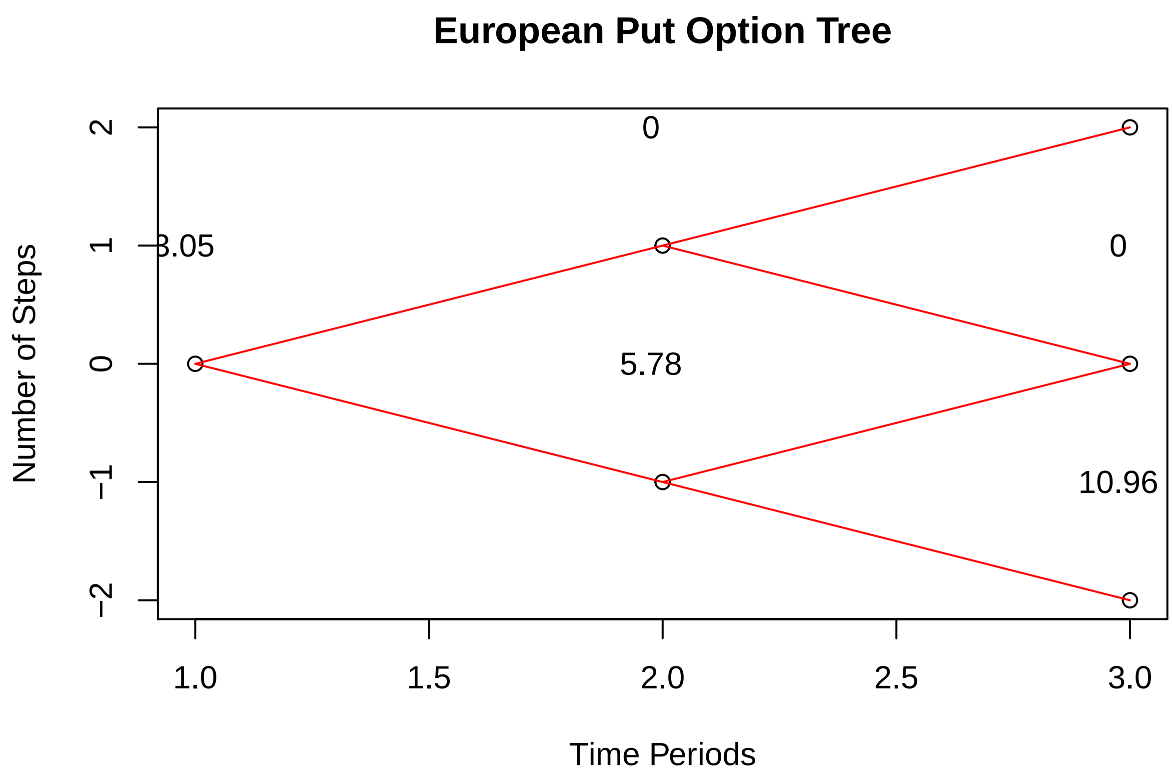
```r
# American PUT option
# Accurate result
CRRBinomialTreeOption(TypeFlag = "pa", S = s, X = k,
                      Time = tt,
                      r = r, b = r,
                      sigma = v,n = 2)@price
```

```
## [1] 3.05286
```

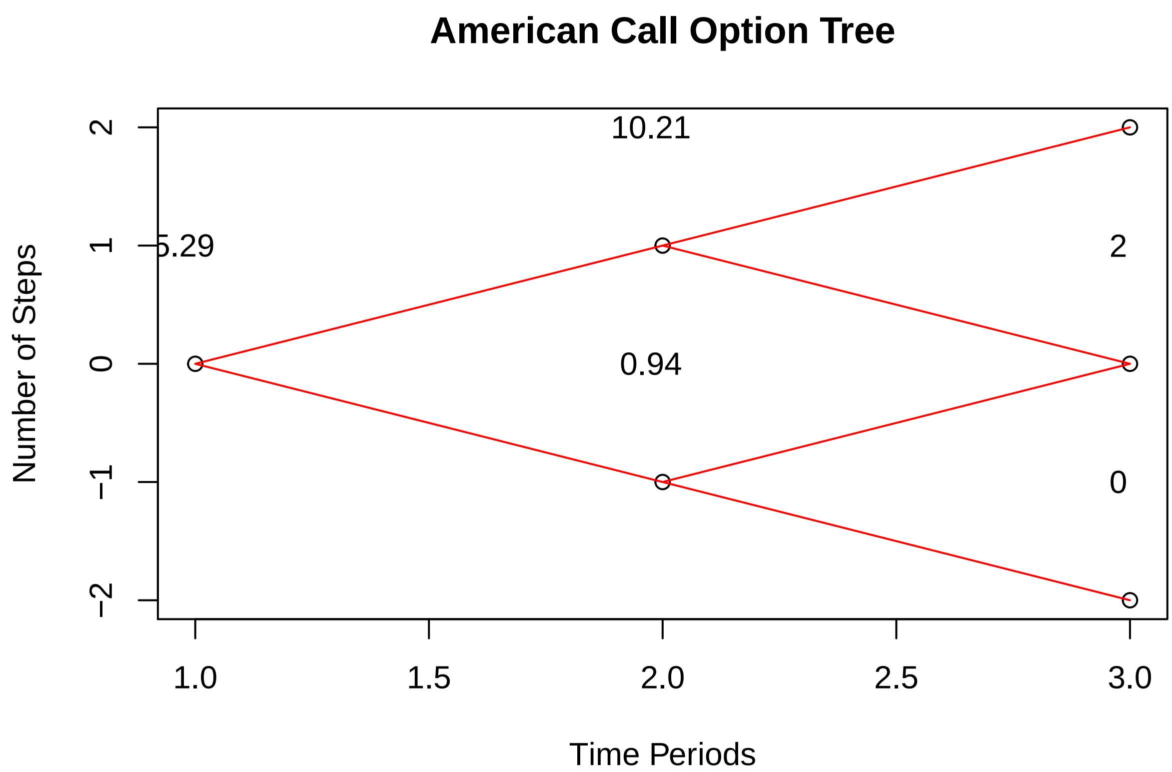
```r
#########################################################
### European p/c option trees
#  Accurate results
# CRR Binomialtree Plots for European CALL Option
bintree <- BinomialTreeOption(TypeFlag = "ce", S = s, X = k,
                              Time = tt,
                              r = r, b = r,
                              sigma = v,n = 2)
# Plot the binomial tree option for European CALL Option
BinomialTreePlot(BinomialTreeValues = bintree, dy = 1,
                 xlab = "Time Periods",
                 ylab = "Number of Steps",
                 main = "European Call Option Tree")
```



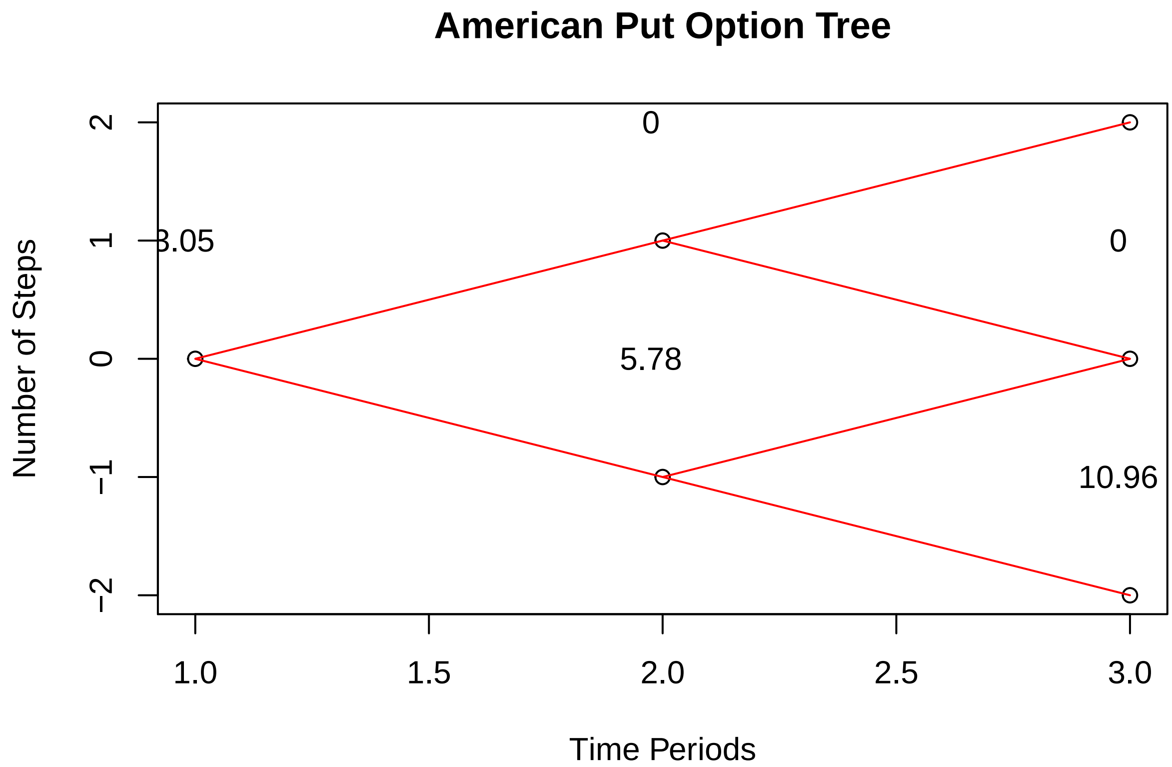

```
# CRR Binomialtree Plot for European PUT Option
bintree <- BinomialTreeOption(TypeFlag = "pe", S = s, X = k,
                              Time = tt,
                              r = r, b = r,
                              sigma = v,n = 2)
# Plot the binomial tree option for European PUT Option
BinomialTreePlot(BinomialTreeValues = bintree, dy = 1,
                 xlab = "Time Periods",
                 ylab = "Number of Steps",
                 main = "European Put Option Tree")
```




```
### American p/c option trees
# CRR Binomialtree Plot for American CALL Option
bintree <- BinomialTreeOption(TypeFlag = "ca", S = s, X = k,
                               Time = tt,
                               r = r, b = r,
                               sigma = v,n = 2)
# Plot the binomial tree option for European CALL Option
BinomialTreePlot(BinomialTreeValues = bintree, dy = 1,
                 xlab = "Time Periods",
                 ylab = "Number of Steps",
                 main = "American Call Option Tree")
```




```r
# CRR Binomialtree Plot for  PUT Option
bintree <- BinomialTreeOption(TypeFlag = "pa", S = s, X = k,
                              Time = tt,
                              r = r, b = r,
                              sigma = v,n = 2)
# Plot the binomial tree option for American PUT Option
BinomialTreePlot(BinomialTreeValues = bintree, dy = 1,
                 xlab = "Time Periods",
                 ylab = "Number of Steps",
                 main = "American Put Option Tree")
```




```
####################################
#### Connection Between the mmodels
# goal is to create a risk free portfolio

# Major difference #
# binomial model is discrete where as
# Black-Scholes-Merton Model is continues

# What is the correct hedging ratio
# partial derivatives of the option value
# with respect to the underlying asset price (DELTA)

# need function!
```