UNIVERSITY OF PARIS 1 PANTHEON SORBONNE

MASTER 2 FINANCE TECHNOLOGY DATA

2022-2023

Asset Pricing

Empirical Application 5

Binomial Tree to Price Options

Group Members

- [redacted]
- [redacted]
- [redacted]
- Julien DOS REIS - FTD

Part 1

1. **Use the code CRR to calculate the fair price of calls/puts on stocks, by choosing different stocks at a given date and different characteristics of the options. Comment on the results.**

1.1. Cox Ross Rubinstein Model

A useful technique for pricing an option involves constructing a binomial tree. This is a diagram representing different possible paths that might be followed by the stock price over the life of an option. The underlying assumption is that the stock price follows a random walk. In each time step, it has a certain probability of moving up (“u”) by a certain percentage amount and a certain probability of moving down by a certain percentage amount.

Consider the situation of an investor who decided on November 17th 2022 to buy a European call option with a strike price of \$ 183,07 to purchase 100 shares of TESLA's stock. If the current share price (17th November) is \$183.07 with a specified expiration date which will be next year on November 19th 2023, because the option is European, the investor can exercise only on the expiration date. If the stock price on this date is less than \$ 183.07, the investor will decide not to exercise the option. However, if the stock price is above \$ 183 on the expiration date, the investor may exercise the option.

The risk-neutral probability of an up-move and down-move is calculated as follows:

$$\forall t = 0, \dots, T-1, \forall e_{it} q(e_{it}, h) = q(h) = \frac{r-b}{(1+r)(h-b)}$$

$$\forall t = 0, \dots, T-1, q(e_{it}, b) = q(b) = \frac{h-r}{(1+r)(h-b)}$$

If the short rate r is continuously compounded, $r = e^{r\Delta t}$, then the risk neutral probability of a down-move can be written as follows q(b) = 1-q(h)

.

The price of the European call option C0 and the price of a European put option P0 is given by the following formula also known as the CRR model.

$$C_0 = (e^{-r\Delta t})^N \sum_{k=0}^{N} \frac{N!}{k!(N-k)!} (S_0 u^k d^{N-k} - K)^+ q^k (1-q)^{N-K}$$

$$P_0 = (e^{-r\Delta t})^N \sum_{k=0}^{N} \frac{N!}{k!(N-k)!} (K - S_0 u^k d^{N-k})^+ q^k (1-q)^{N-K}$$

Where

S = spot price of Tesla at 18th of November 2022 (at t)

r= risk free rate

K= strike = 180.19

T=expiration date (maturity)

N=number of periods (steps)

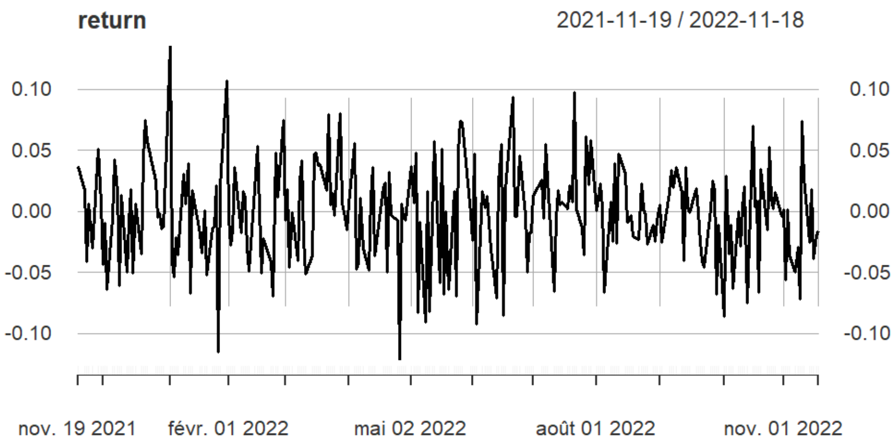
Δt= T/N

u = up move

d =down move

1.2. Data

Before starting the work we will first show the different data that we are going to use :
As mentioned above, we chose TESLA and our dataset starts from November 18th 2021, until November 18th 2022. For the risk free rate we will work with the 10-year US treasury yield. We can plot our data:



	TSLA.Open	TSLA.High	TSLA.Low	TSLA.Close	TSLA.Volume
2021-11-18	368.8500	370.6667	358.3400	365.4600	62696700
2021-11-19	366.2900	379.5733	364.2333	379.0200	64926900
2021-11-22	387.4433	400.6500	377.4767	385.6233	99217500
2021-11-23	389.1700	393.5000	354.2333	369.6767	108515100
2021-11-24	360.1300	377.5900	354.0000	372.0000	67680600
2021-11-26	366.4900	369.5933	360.3333	360.6400	35042700
2021-11-29	366.9967	380.8900	366.7300	378.9967	58393500

1.3. Calculating fair prices of options using CRR Model

1.3.1. Calculation of parameters

Before applying the formula and constructing the binomial tree it would be appropriate to start calculating the different parameter needed:

- volatility (sigma) = 0.643

```
##calcul of volatility
sigma <- sd(return) * sqrt(252)
```

Values	
sigma	0.643957017495721

- risk free rate (r) = 3.818

```
###THE RISK FREE rate 10-year US treasury yield
getSymbols("^TNX",from = "2022-11-16",to = "2022-11-19")
r <- TNX$TNX.Close
r<- r[-1]
```

	TNX.Close
2022-11-18	3.818

- delta-t (Δt) = 0.5

$\Delta t = T/N = 1/2$

(in our case T = 1 year and N= 2)

- down move (d) = 0.634 and up-move (u) = 1.576

```
u = exp(sigma*sqrt(1/2))
d = exp(-sigma*sqrt(1/2))
```

u	1.57671942286033
d	0.634228249808641

- risk-neutral probability

```
##### the proba of risk neutral
q_prob <- function(r, delta_t, sigma,u,d) {

  return(((exp(r*delta_t))-d)/(u-d))
}

q_prob(3.18/100,0.5,sigma,u,d)
```

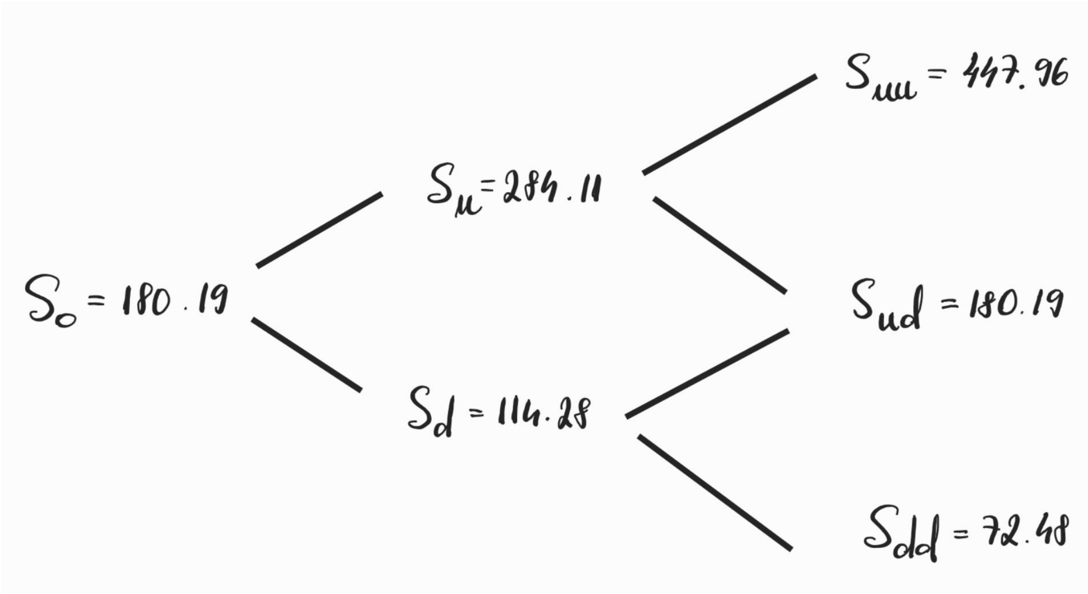
```
> q_prob(3.18/100,0.5,sigma,u,d)
[1] 0.4050954
>
```

1.3.2. Building the stock price binomial tree:

```
build_stock_tree <- function(s, sigma, delta_t, N) {
  tree = matrix(0, nrow=N+1, ncol=N+1)
  for (i in 1:(N+1)) {
    for (j in 1:i) {
      tree[i, j] = s * u^(j-1) * d^((i-1)-(j-1))
    } }
  return(tree)
}
TESLA_tree<-build_stock_tree(s, sigma, delta_t, N)
view(TESLA_tree)
```

	V1	V2	V3
1	180.19000	0.0000	0.0000
2	114.28159	284.1091	0.0000
3	72.48061	180.1900	447.9603

The stock price binomial tree can be represented like this:



Finally, the above functions can be combined and create a complete function to value options in each node of the tree. The following function called value_binomial_option would do that job. So the next step is to get the binomial tree for options pricing.

A call option value at expiration is given by the MAX(S-K,0), where S is the spot price of the underlying asset (the stock) and K is the strike price of the call option. In our example, if the strike price K is USD 180.19, the intrinsic value of the option related to the different stock paths would be: 267.77 , 0 , 0 (for the last one we put directly 0 because we think it wouldn't be interesting to exercise the call option in that case), and we can confirm those result in the option tree later on because the last row of the option_tree matrix is filled with the max the max value between 0 and the difference between the Stock Price and the Strike price if the type is 'call'.

1.3.2. Building the stock price binomial tree:

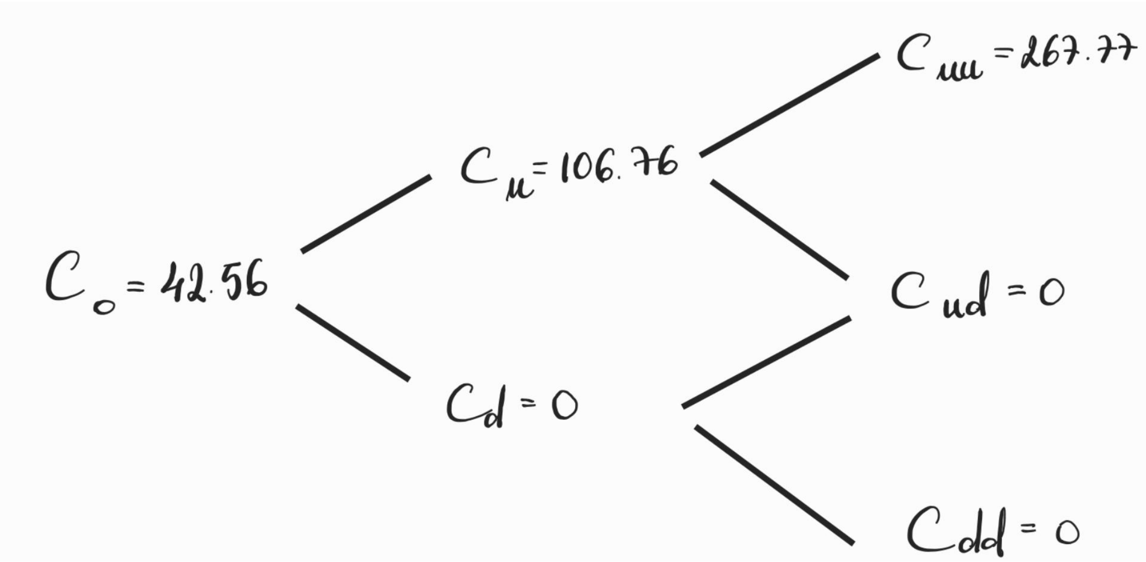
The equivalent R code for the above CRR model which accounts for both the European call and European put options is as follows: (in this code we are going to use the formula we put above of how to call the fair price of an option) :

```
CRR_model_call_put<-function(tree,u,d, delta_t, r, K, type) {
  q = q_prob(r, delta_t,sigma,u,d)
  option_tree <- matrix(0,nrow=nrow(tree), ncol=ncol(tree))
  if(type == 'put') {
    option_tree[nrow(option_tree),]= pmax(K - tree[nrow(tree),]
  } else {
    option_tree[nrow(option_tree),] = pmax(tree[nrow(tree),] -
  }

    for (i in (nrow(tree)-1):1) {
      for(j in 1:i) {
        option_tree[i,j]=((1-q)*option_tree[i+1,j] +q*option_tr
      }
    }
    return(option_tree)
  }
tree=TESLA_tree
CRR_model_call_call(TESLA_tree,u,d,1/2,r,K, 'call')
view(CRR_model_call_put)
```

```
             [,1]       [,2]     [,3]
[1,] 4.256635e+01   0.000000   0.0000
[2,] 7.974106e-07 106.761440   0.0000
[3,] 0.000000e+00   0.000002 267.7703
```

The option (Call) binomial tree can be represented like this:



We can say that the faire price of the call option at t0 is equal to 42.56

It would be interesting for the investor if he exercises his call option after 6 months of the price goes up but since it's an European option he will need to wait till the expiration date and he will gain 267.77* 100 (because we said 100 shares) if the price goes up otherwise he will lose 180.19 which is the strike price.

Part 2

2. Estimation of default probabilities

2.1. Yield curves

Firstly, we consider 2 yield curves as of October 2022, one without risk and the other one for a ranking indicating possible default of the issuing firm.

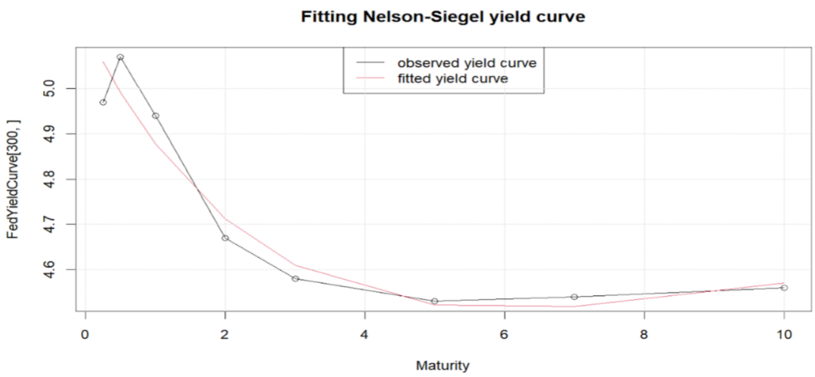
2.1.2. Data

For risk free bonds we chose to work with the FED treasury bonds for different maturities, 3 months, 6 months, 1,2,3,5,7 and 10 years and we chose daily data. For corporate bonds we considered High Quality Market (HQM) Corporate Bond, corporate bonds rated AAA, AA, or A, that are available with monthly probability.

2.1.3. Interpolation of a Yield Curve (Nelson-Siegel-Svensson yield curve fit method):

Since zero-coupon bonds do not exist for all maturities we have to estimate the values we do not observe by those we have. This mens performing an interpolation of a yield curve based on the rates we have for maturities 3 months, 6 months, 1,2,3,5,7 and 10 years. We can estimate them using the Nelson-Siegel approach. We can apply this in R by using the function Nelson.Siegel() and NSrates(). The estimated yield curve for risk-free bonds is presented in Figure 1.

Figure 1. Nelson-Siegel Yield Curve for risk-free bond



So, we can assume that for 1 and 2 years we have the following return rates for risk free and corporate bonds accordingly.

```
#rate for different maturity for corporate bond
Maturity1_corp <- 4.79
Maturity2_corp <- 5.14

#rate for different maturity for risk free bond
Maturity1_free <- 4.55
Maturity2_free <- 4.41
```

2.2. Estimation of the associated default probabilities

As we consider a firm issuing a High-Quality Market Corporate with a maturity of 2 years, we should estimate the associated default probabilities in the first year and in the second year. But to calculate the probabilities of default we should firstly adopt a binomial characterization of the evolution of the return rates of risk free bonds by defining two possible states, up and down, with respective probabilities π and $(1 - \pi)$.

As we cannot reach data of probabilities of returns rate moving up and down and their values, we cannot calculate the price for bonds. So next we continue from a theoretical point of view.

Formula for risk-free bond default probability between period 0 and period 1 is the following:

$$Bf(0,1) = \exp(\text{return rate between dates 0 and 1})$$

Formula for risk-free bond default probability between period 1 and period 2 is the following:

Bf(1,2) = exp(return rate between dates 0 and 1)[0,5*100exp(rd)+0,5*100exp(ru)]

Then to calculate probability of return:

μ(1) denotes the associated default probability over period [1, 2] .

$$B_F(0,1)(1-\mu(0))(1-\mu(1))(\pi * B_{u,F}(1,2)*100+(1-\pi)*B_{d,F}(1,2)*100)$$
$$+B_F(0,1)(1-\mu(0))\,\mu(1)(\pi * B_{u,F}(1,2)*40+(1-\pi)B_{d,F}(1,2)*40)$$
$$+B_F(0,1)\mu(0)(\pi * B_{u,F}(1,2)*40\ +(1-\pi)*B_{d,F}(1,2)*40)$$

$$= 89.7056$$

3. Calculating the fair value of options using the binomial tree

Then we use the tree to assess the fair value of a put on the previous bond. In this step we follow the steps as in part 1 of the assignment.

Code

Part 1

```
library(quantmod)
library(xts)
library(PerformanceAnalytics)
library(urca)
library(forecast)
library(tidyverse)
library(tseries)
library(tsDyn)
library(imputeTS)

### I work with TESLA stocks
getSymbols("TSLA",from = "2021-11-18",to = "2022-11-19")
return <-  CalculateReturns(TSLA$TSLA.Close)
return <- return[-1]
plot(return)
```

```
##calcul of volatility
sigma <- sd(return) * sqrt(252)
###THE RISK FREE rate 10-year US treasury yield
getSymbols("^TNX",from = "2022-11-16",to = "2022-11-19")
r <- TNX$TNX.Close
r<- r[-1]
view(r)

### calcul of u and d
u = exp(sigma*sqrt(1/2))
d = exp(-sigma*sqrt(1/2))
delta_t= 1/2
#calculating the risk neutral probability q
#risk free = r
# stock price =S

# N= maturity
#delta_t = T/N
# N= number of period ( steps)
# K= strike
#s= sport price
s= tail(TSLA$TSLA.Close,1)
N=2
r=0.0318
T=1
##### the proba of risk neutral
q_prob <- function(r, delta_t, sigma,u,d) {
  return(((exp(r*delta_t))-d)/(u-d))
}
q_prob(3.18/100,0.5,sigma,u,d)

#### stock tree ##
build_stock_tree <- function(s, sigma, delta_t, N) {
  tree = matrix(0, nrow=N+1, ncol=N+1)
  for (i in 1:(N+1)) {
    for (j in 1:i) {
      tree[i, j] = s * u^(j-1) * d^((i-1)-(j-1))
    }  }
  return(tree)
}
```

```
TESLA_tree<-build_stock_tree(s, sigma, delta_t, N)
view(TESLA_tree)
#### intrinque value it's way to confirm if we are doing things right##
K <- 180.19 ## strike price of the call option
IntrinsecValue <- TESLA_tree[3,]-K
IntrinsecValue[IntrinsecValue<0] <- 0
View(IntrinsecValue)

#### fisrt try but I dindt like it it's a bit complicated###
CRR_model_ca11_put<-function(tree,u,d, delta_t, r, K, type) {
  q = q_prob(r, delta_t,sigma,u,d)
  option_tree <- matrix(0,nrow=nrow(tree), ncol=ncol(tree))
  if(type == 'put') {
    option_tree[nrow(option_tree),]= pmax(K - tree[nrow(tree),],0)
  } else {
    option_tree[nrow(option_tree),] = pmax(tree[nrow(tree),] - K, 0)
  }

    for (i in (nrow(tree)-1):1) {
      for(j in 1:i) {
        option_tree[i,j]=((1-q)*option_tree[i+1,j] +q*option_tree[i+1,j+1])/exp(r*delta_t)
      }
    }
    return(option_tree)
  }
tree=TESLA_tree
CRR_model_ca11_call(TESLA_tree,u,d,1/2,r,K, 'call')
view(CRR_model_ca11_put)

#### another manner to do i prefer this one####
binomial_option <- function(type, sigma, T, r, K, S, N) {
  q <- q_prob(r=r, delta_t=T/N, sigma=sigma,u,d)
  tree <- build_stock_tree(s=s, sigma=sigma, delta_t=T/N, N=N)
  option <- value_binomial_option(tree, sigma=sigma, delta_t=T/N, r=r, K, type=type)
  return(list(q=q, stock=tree, option=option, price=option[1,1]))
}
results <- binomial_option(type='call', sigma, T, r, K, s, N)
results$price
results$option
```

Part 2

```
#Firstly we consider a yield curve one without risk - in our case US treasury bond
#interpolation
library(YieldCurve)
data(FedYieldCurve)
rate.Fed = c(4.18, 4.51, 4.55, 4.41, 4.38, 4.19, 4.10, 4.02)
maturity.Fed <- c(3/12, 0.5, 1,2,3,5,7,10)
NSParameters <- Nelson.Siegel( rate=FedYieldCurve,            maturity=c(3/12, 0.5, 1,2,3,5,7,10))
y <- NSrates(NSParameters[300,], maturity.Fed)
plot(maturity.Fed,FedYieldCurve[300,],main="Fitting Nelson-Siegel yield curve", xlab=c("Maturity"), type="o")
lines(maturity.Fed,y, col=2)
legend("top",legend=c("observed yield curve","fitted yield curve"),col=c(1,2),lty=1)
grid()

#Secondly we consider a yield curve of a corporate bond
#for a ranking indicating possible default of the issuing firm
#in our case - Moody's Baa Corporate Bond Yield

library(Quandl)

# Obtain Moody's Baa index data
baa <- Quandl("FED/RIMLPBAAR_N_M")

# Identify 11/30/12 yield
baa_yield <- subset(baa, baa$Date == "2012-11-30")

# Convert yield to decimals
baa_yield <- baa_yield$Value / 100
baa_yield

#Now that you know the yield, you can use this information to find the value
#of a Baa-rated bond on October 31, 2015
#with a $100 par value, 5% coupon rate, and 2 years to maturity.

# Create function
#p for par value, r for coupon rate, ttm for time to maturity, and y for yield
bondprc <- function(p, r, ttm, y){
  cf <- c(rep(p * r, ttm - 1), p * (1 + r))
  cf <- data.frame(cf)
```

```
  cf$t <- as.numeric(rownames(cf))
  cf$pv_factor <- 1 / (1 + y)^cf$t
  cf$pv <- cf$cf * cf$pv_factor
  sum(cf$pv)
}

# Value bond
bondprc(p = 100, r = 5, ttm = 2, y = baa_yield)

#value this bond at different levels of yield
# Create prc_yld
prc_yld <- seq(0.02, 0.4, 0.01)

# Convert to data frame
prc_yld <- data.frame(prc_yld)

# Calculate bond price given different yields
for (i in 1:nrow(prc_yld)) {
  prc_yld$price[i] <- bondprc(100, 0.10, 2, prc_yld$prc_yld[i])
}

# Plot P/YTM relationship
plot(prc_yld,
     type = "l",
     col = "blue",
     main = "Price/YTM Relationship")

#we need to find prices
price_V01_corp <- x
price_V12_corp <- x
#we need to find prices
price_V01_free <- x
price_V12_free <- x

#and now we solve the equation to find a default probability
default_1 = x
#Then use the tree to assess the fair value of a put on the previous bond (for the
#given ranking and with maturity 2 years).

#Take from first part code
```