

hw8 Solution to Ex2, Ex3, Ex5

Yaman Yucel

```
#Q2
u = 1.06
d = 0.95
s0 = 50
r = 0.05
E = 51
months = 3
t1 = months/12
t2 = 2*months/12
sU = u*s0
sU2 = u*sU
sD = d*s0
sD2 = d*sD
sUD = u*d*s0

p = (exp(r*t1) - d)/(u-d)
p_not = 1-p

cu2 = max(sU2 - E, 0)
cud = max(sUD - E, 0)
cd2 = max(sD2 - E, 0)

C = (p^2*cu2 + 2*p*(p_not)*cud + p_not^2*cd2)*exp(-r*t2)
cu = (p*cu2 + p_not*cud)*exp(-r*t1)
cd = (p*cud + p_not*cd2)*exp(-r*t1)
c_2 = (cu*p + cd *p_not)*exp(-r*t1)

print(sU)

## [1] 53
print(sU2)

## [1] 56.18
print(sD)

## [1] 47.5
print(sD2)

## [1] 45.125
print(sUD)

## [1] 50.35
```

```

print(p)

## [1] 0.568895
print(p_not)

## [1] 0.431105
print(cu2)

## [1] 5.18
print(cud)

## [1] 0
print(cd2)

## [1] 0
print(cu)

## [1] 2.910269
print(cd)

## [1] 0
print(C)

## [1] 1.635071
print(c_2)

## [1] 1.635071
#Q3
u = 1.06
d = 0.95
s0 = 50
r = 0.05
E = 51
months = 3
t1 = months/12
t2 = 2*months/12
sU = u*s0
sU2 = u*sU
sD = d*s0
sD2 = d*sD
sUD = u*d*s0

p = (exp(r*t1) - d)/(u-d)
p_not = 1-p

pu2 = max(E - sU2,0)
pud = max(E - sUD ,0)
pd2 = max(E - sD2,0)

P = (p^2*pu2 + 2*p*(p_not)*pud + p_not^2*pd2)*exp(-r*t2)
pu = (p*pu2 + p_not*pud)*exp(-r*t1)

```

```

pd = (p*pud + p_not*pd2)*exp(-r*t1)
P_2 = (pu*p + pd *p_not)*exp(-r*t1)

parity_right = C + E*exp(-r*t2)
parity_left = P + s0
dif = parity_right - parity_left

print(sU)

## [1] 53
print(sU2)

## [1] 56.18
print(sD)

## [1] 47.5
print(sD2)

## [1] 45.125
print(sUD)

## [1] 50.35
print(p)

## [1] 0.568895
print(p_not)

## [1] 0.431105
print(pu2)

## [1] 0
print(pud)

## [1] 0.65
print(pd2)

## [1] 5.875
print(pu)

## [1] 0.2767373
print(pd)

## [1] 2.866468
print(P)

## [1] 1.375877
print(P_2)

## [1] 1.375877

```

```

print(dif)

## [1] -7.105427e-15

u = 1.2
d = 1/u
E = 60
s0 = 50
n = 10
r = 0.1 # for each period, not continuous compounding

k = ceiling(log(E/(d^n*s0))/log(u/d)) # 6

p = ((1+r) - d)/(u-d)
p_not = 1-p

last_price_list = rep(0,n + 1)
call_list = rep(0,n+1)

for (j in 0:n){
  last_price_list[j+1] = u^(j) * d^(n-j) * s0
  call_list[j+1] = max(last_price_list[j+1] - E,0)
}
p_mark = p*u/(1+r)
p_mark_not = 1-p_mark

c = s0*pbinom(k-1,n,p_mark, lower.tail=FALSE) - (E/(1+r)^n)*pbinom(k-1, n, p, lower.tail=FALSE)

get_comb <- function(n,k){ factorial(n)/(factorial(k)*factorial(n-k))}
c_2 = 0
for (i in k:n){
  c_2 = c_2 + get_comb(n,i) * p^i*(1-p)^(n-i)*call_list[i+1]
}
c_2 = c_2 / ((1+r)^n)

dif = c - c_2

print(k)

## [1] 6

print(p)

## [1] 0.7272727

print(p_not)

## [1] 0.2727273

print(last_price_list)

## [1] 8.075279 11.628402 16.744899 24.112654 34.722222 50.000000
## [7] 72.000000 103.680000 149.299200 214.990848 309.586821

print(call_list)

## [1] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 12.0000 43.6800

```

```
## [9] 89.2992 154.9908 249.5868
```

```
print(p_mark)
```

```
## [1] 0.7933884
```

```
print(p_mark_not)
```

```
## [1] 0.2066116
```

```
print(c)
```

```
## [1] 27.48628
```

```
print(c_2)
```

```
## [1] 27.48628
```

```
print(dif)
```

```
## [1] -3.552714e-15
```