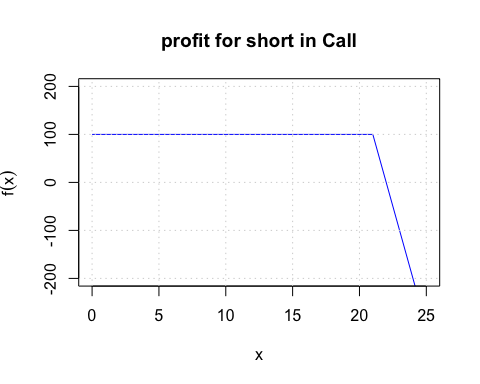
HWK2

Dingxian Cao

September 21, 2015

# 1

size <- 100   
k<-21   
f<-100   
s0<-20   
call\_short <- function(s){  
 100 + sapply(s, function(s) min(c(k-s,0))\*size)  
}   
  
curve(call\_short,0,25, main = "profit for short in Call", col = "blue", lty=1,lwd=1,ylab=expression(f(x)),ylim = c(-200,200))  
grid()



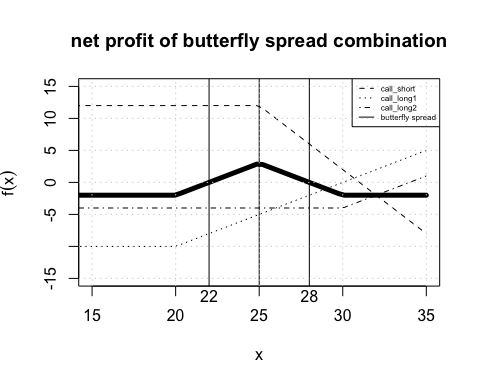
# 2

c=4  
p=3  
r=0.1  
k=22  
s0=20  
t<- 0.5  
parity<-data.frame(p+s0,c+k\*exp(-r\*t))  
names(parity)<-c("p+s0","c+k\*exp(-r\*t)")  
parity

## p+s0 c+k\*exp(-r\*t)  
## 1 23 24.92705

# 3

k1<-20  
c1<-10  
k2<-30  
c2<-4  
  
k3<-25  
c3<-6   
  
call\_short <- function(s){  
 c3\*2 + sapply(s, function(s) min(c(k3-s,0))\*2)  
}   
  
call\_long1 <-function(s){  
 -c1 + sapply(s, function(s) max(c(s-k1,0)))  
}  
call\_long2 <-function(s){  
 -c2 + sapply(s, function(s) max(c(s-k2,0)))  
}  
net\_profit<-function(s){  
 call\_long2(s)+call\_long1(s)+call\_short(s)  
}  
profit<-net\_profit(seq(20,30,by = 0.01))   
  
  
curve(call\_short,0,35,ylim=c(-15,15),xlim=c(15,35),lty=2,main="net profit of butterfly spread combination",ylab=expression(f(x)))  
curve(call\_long1,0,35,lty=3,add = T)  
curve(call\_long2,0,35,lty=4,add = T)  
curve(net\_profit,0,35,lty=1,lwd=5,add = T)  
abline(v=seq(20,30,by = 0.01)[which.max(profit)])  
legend("topright",lty=c(2,3,4,1),c("call\_short","call\_long1","call\_long2","butterfly spread"),cex = .5)   
zero\_point<-seq(20,30,by = 0.01)[which(abs(profit)<0.001)]  
abline(v=zero\_point)  
mtext(text = as.character(zero\_point), at = zero\_point,side = 1)  
grid()



# 4

s0=10  
r=0.12  
k=10.5  
su=11  
sd=9  
  
(delta<-(su-k-0)/(su-sd))

## [1] 0.25

(f<-delta\*s0-(delta\*sd\*exp(-r\*0.25)))

## [1] 0.3164975

# 

# 5

n = 10  
tau= 1  
r = .06  
S0 = 100  
u = 1.1   
d = 0.9  
K = 110   
pu=(exp(r\*tau)-d)/(u-d)  
pd=1-pu  
  
S<- vector("list",10)# market price of the underlying asset  
for(i in seq(10)){  
 for(j in 0:i){  
 S[[i]]<-c(S[[i]],S0\*d^j\*u^(i-j))  
 }  
}  
  
  
V<-vector("list",10)# value for the option  
V[[10]]<-sapply(S[[10]],function(s) max(c(s-K,0)))  
for(i in 9:1){  
 for(j in 1:(i+1)){  
 V[[i]]<-c(V[[i]],exp(-r\*tau)\*(pu\*V[[i+1]][j] +pd\*V[[i+1]][j+1]))  
 }  
}  
V

## [[1]]  
## [1] 46.06298 26.72316  
##   
## [[2]]  
## [1] 53.02244 31.47780 15.22050  
##   
## [[3]]  
## [1] 60.864971 36.947810 18.482341 6.320686  
##   
## [[4]]  
## [1] 69.679457 43.210121 22.365121 8.006382 1.220588  
##   
## [[5]]  
## [1] 79.563788 50.344241 26.959797 10.128538 1.601697 0.000000  
##   
## [[6]]  
## [1] 90.62704 58.43238 32.36021 12.79537 2.10180 0.00000 0.00000  
##   
## [[7]]  
## [1] 102.991987 67.560767 38.658063 16.140121 2.758052 0.000000  
## [7] 0.000000 0.000000  
##   
## [[8]]  
## [1] 116.797633 77.823291 45.935193 20.326142 3.619208 0.000000  
## [7] 0.000000 0.000000 0.000000  
##   
## [[9]]  
## [1] 132.200670 89.328894 54.251986 25.552698 4.749245 0.000000  
## [7] 0.000000 0.000000 0.000000 0.000000  
##   
## [[10]]  
## [1] 149.374246 102.215292 63.630694 32.061477 6.232117 0.000000  
## [7] 0.000000 0.000000 0.000000 0.000000 0.000000

print(exp(-r\*tau\*1)\*(pu\*V[[1]][1] +pd\*V[[1]][2])) #the present fair price for the option. from the view of recursive function.

## [1] 39.90502

p\_bio<-sapply(n:0,function(j) choose(n,j)\*pu^(j)\*pd^(n-j))  
print(exp(-r\*tau\*n)\*(V[[10]]%\*%p\_bio)) #the present fair price for the option. from the view of bionomial model

## [,1]  
## [1,] 39.90502