Problem Set 5

Ryan Shea

11/21/2022

I pledge my honor that I have abided by the Stevens Honor System.

Note: I would like to use 1 late day for this assignment.

1: Financial Engineering with Options

```
# Function to price calls and puts using Black Scholes equation
BS <- function(S, K, r, sigma, t, type="c") {
  d1 <- (log(S / K) + (r + sigma^2 / 2) * t) / (sigma * sqrt(t))
  d2 <- (log(S / K) + (r - sigma^2 / 2) * t) / (sigma * sqrt(t))

  if (type == "c" | type == "call") {
    return (
      S * pnorm(d1) - (K * exp(-r * t) * pnorm(d2))
    )
  }

  if (type == "p" | type == "put") {
    return (
      (K * exp(-r * t) * pnorm(-d2)) - S * pnorm(-d1)
    )
  }
}
```

1.1

The trader should write an OTM call and buy an OTM put with a strike $20 away from each other if they want to implement a collar strategy.

```
vol <- 0.4
r <- 0.05
B <- seq(70, 90, 0.5) # Assume strikes are $0.50 apart from each other
A <- B - 20
S_0 <- 70
t <- 0.5

short_c <- BS(S=S_0, K=B, r=r, sigma=vol, t=t, type="c") # CF at t=0
long_p <- BS(S=S_0, K=A, r=r, sigma=vol, t=t, type="p")
difference <- short_c - long_p
df <- data.frame(short_c, B, long_p, A, difference)
colnames(df) <- c("Short Call Price", "Call Strike", "Long Put Price", "Put Strike", "Difference")
df[25:30,]
```

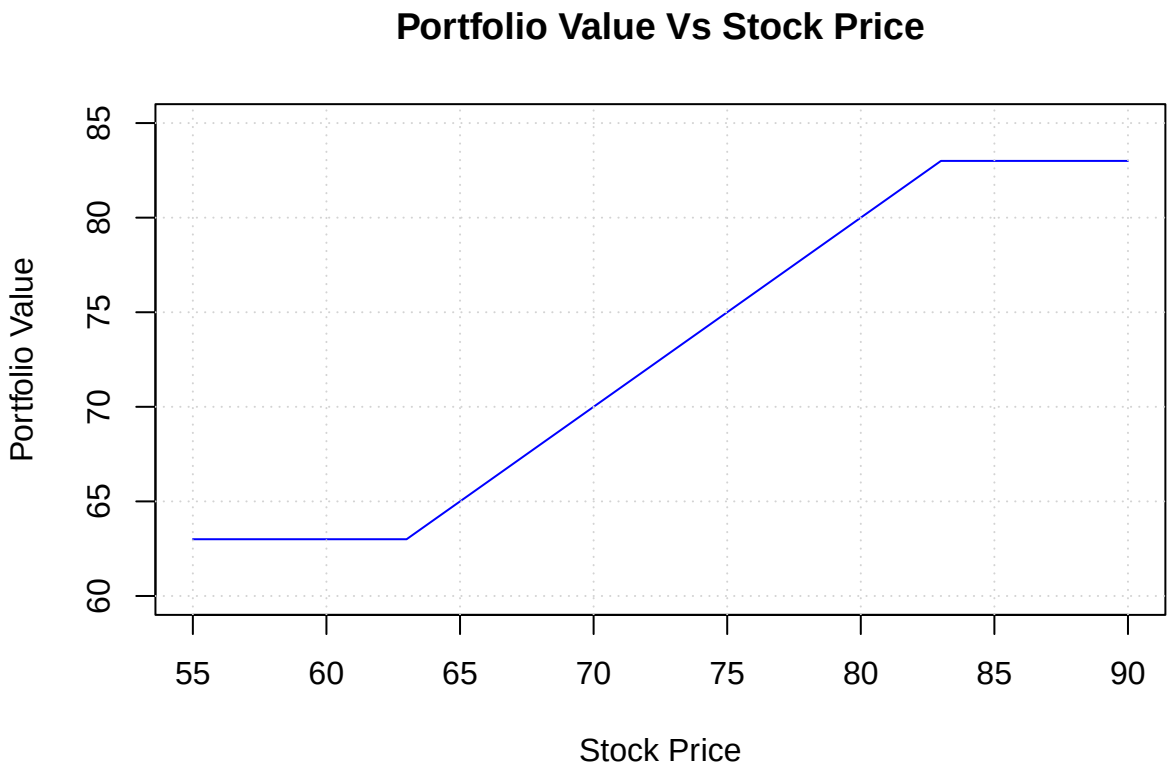
```
##    Short Call Price Call Strike Long Put Price Put Strike Difference
## 25         4.356424        82.0       3.523468       62.0  0.8329562
## 26         4.226424        82.5       3.698412       62.5  0.5280120
## 27         4.099834        83.0       3.878536       63.0  0.2212980
## 28         3.976587        83.5       4.063847       63.5 -0.0872600
## 29         3.856616        84.0       4.254348       64.0 -0.3977321
## 30         3.739855        84.5       4.450039       64.5 -0.7101838
```

```
vol <- 0.4
r <- 0.05
B <- 83
A <- B - 20
```

```r
S_0 <- 70
t <- 0.5
S_t <- 55:90

port <- function(S) {
  return (
    S +
    -max(S - B, 0) + # Short call
    max(A - S, 0) # Long Put
  )
}

portfolio <- cbind(S_t, lapply(S_t, port))
plot(portfolio,
     type='l',
     col='blue',
     ylim=c(60, 85),
     xlab="Stock Price",
     ylab="Portfolio Value",
     main="Portfolio Value Vs Stock Price"
     )
grid()
```



They should write a call at $K = 83$ and buy a put at $K = 63$ since this is the minimum positive difference between the short call and the long put. This allows the written call to pay for the put (while keeping $\sim \$0.22$ as this was the difference) and allows some protection against volatility by capping upside and downside risk.

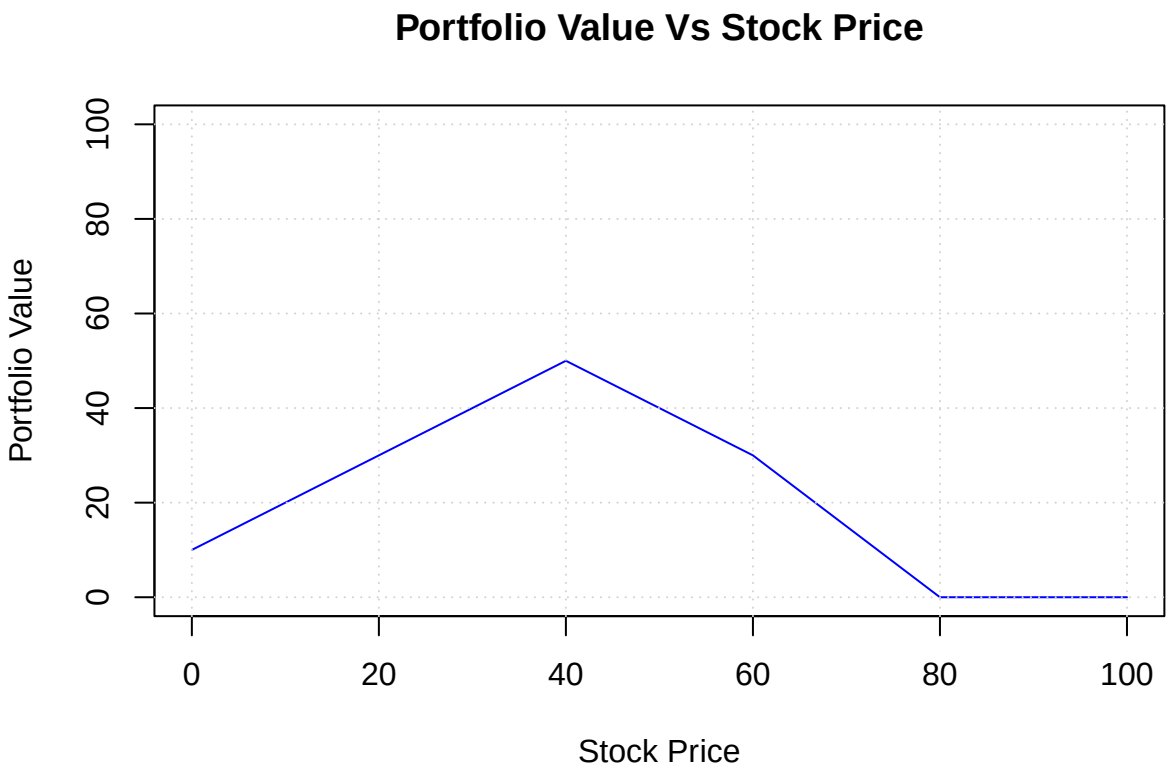
1.2

```
S_t <- 0:100
port <- function(S) {

  put <- function(K){ return ( max(K - S, 0) ) }
  call <- function(K){ return ( max(S - K, 0) ) }

  return (

    3/2 * put(80) +
    - 1/2 * put(60) +
    -2 * put(40)
  )
}
portfolio <- cbind(S_t, lapply(S_t, port))
plot(portfolio,
     type='l',
     col='blue',
     xlab="Stock Price",
     ylab="Portfolio Value",
     ylim = c(0, 100),
     main="Portfolio Value Vs Stock Price"
     )
grid()
```



They buy a bond that matures in 6 months with a face value of 50.

The investor should also buy 1.5 puts with strike 80.

They should also write 0.5 puts with strike 60.

Next, they should write 2 puts with strike 40.

Finally, they should write 0.5 more puts with strike 20.

```
S_0 <- 50
r <- 0.05
t <- 0.5
sigma <- 0.4
#50 = B * exp(r * t) BOND
B <- 50 / exp(r * t) # will equal 50 at 6 months

p80 <-  3/2 * BS(S_0, 80, r, sigma, t, 'p')
p60 <- -1/2 * BS(S_0, 60, r, sigma, t, 'p')
p40 <- -2   * BS(S_0, 40, r, sigma, t, 'p')
p20 <- -1/2 * BS(S_0, 20, r, sigma, t, 'p')


B + p80 + p60 + p40 + p20
```

```
## [1] 83.21709
```

2: Pricing Options with Binomial Trees

2.1

```
S_0 <- 60
u <- S_0 * 1.2
d <- S_0 * 0.8
r <- 0.08
p_K <- 66

# Put Values
pu <- max(p_K - u, 0)
pd <- max(p_K - d, 0)

delta <- (pu - pd) / (u - d)
delta
```

```
## [1] -0.75
```

```
# Portfolio with 1 put, and delta (0.75) stocks, compute value
vu <- pu + u * -delta
vd <- pd + d * -delta
c(vu, vd)
```

```
## [1] 54 54
```

```
V <- 54
pv_V <- V / (1 + r)
pv_V
```

```
## [1] 50
```

```
put_px <- pv_V - (S_0) * -delta # Subtract stocks from portfolio
put_px
```

```
## [1] 5
```

2.2

```
S_0 <- 54
u <- 80
d <- 40
r <- (100 - 96) / 96

# Risk neutral probabilities
q <- (S_0 * exp(r) - d) / (u - d) # P(up)
p <- 1 - q # P(down)

c(q, p)
```

```
## [1] 0.4074383 0.5925617
```

```
# ITM Call Option, K = 50
C <-  (max(u - 50, 0) * q + max(d - 50, 0) * p) * exp(r)
C
```

```
## [1] 12.74321
```

```
# ATM Put Option, K = S_0 = 54
P <- (max(S_0 - u, 0) * q + max(S_0 - d, 0) * p) * exp(r)
P
```

```
## [1] 8.648827
```

2.3

```
S_0 <- 100
r <- 0.05

u <- S_0 * 1.1
d <- S_0 * 0.9

uu <- u * 1.2
ud <- u * 0.8

du <- d * 1.2
dd <- d * 0.8

tree <- data.frame(c(u, NA,  d, NA), c(uu, ud, du, dd))
colnames(tree) <- c("P_1", "P_2") # Price of asset
rownames(tree) <- c("u(uu)", "ud", "d(du)", "dd")
tree
```

```
##       P_1 P_2
## u(uu) 110 132
## ud     NA  88
## d(du)  90 108
## dd     NA  72
```

```
# Risk Neutral Probabilities
p_1 <- (S_0 * exp(r) - d) / (u - d)
q_1 <- 1 - p_1
c(p_1, q_1)
```

```
## [1] 0.7563555 0.2436445
```

```
p_2 <- mean((u * exp(r) - ud) / (uu - ud),
            (d * exp(r) - dd) / (du - dd))
q_2 <- 1 - p_2
c(p_2, q_2)
```

```
## [1] 0.6281777 0.3718223
```

```
K <- 102
r <- 0.05

put_uu <- max(K - uu, 0)
put_ud <- max(K - ud, 0)
put_du <- max(K - du, 0)
put_dd <- max(K - dd, 0)

c(put_uu, put_ud, put_du, put_dd)
```

```
## [1]  0 14  0 30
```

```r
price_put <- function(u, d, p, q, r, t) {
  return (
    (u * p + d * q) * exp(-r * t)
  )
}
# calculate put prices at t = 1
pu <- price_put(put_uu, put_ud, p_2, q_2, r, 1)
pd <- price_put(put_du, put_dd, p_2, q_2, r, 1)
c(pu, pd)
```

```
## [1]  4.951636 10.610648
```

```r
# upward path is OTM, so do not exercise
# downward path is lower than exercise (102 - 90 = 12)
# so you would exercise at step d, making the price 12.
pd <- 12

put_px <- price_put(pu, pd, p_1, q_1, r, 1)
put_px
```

```
## [1] 6.343684
```

3: Stochastic Processes

3.1

$$dX_1 = 5dt + 10d\beta_t$$
$$X_0 = 100,\ P(X_1 > 100)$$
$$\implies X_1 \sim 1 - N_{CDF}(100, \mu = 105, \sigma = 10)$$

```
1 - pnorm(100, mean=105, sd=10)
```

```
## [1] 0.6914625
```

3.2

The gambler's wealth is a martingale because the conditional expectation of the next step in the random walk is always equal to their present wealth. In other words, $E[X_{n+1} \mid X_1, ..., X_n] = X_n$.

In this case, $X_0 = 40$. $E[X_1 \mid X_0] = 0.5 * 41 + 0.5 * 39 = 40 = X_0$. Now let's say we know $X_1 = 41$.

$$E[X_2 \mid X_0, X_1] = 0.5 * 42 + 0.5 * 40 = 41 = X_1$$

This holds true for all valid random walks, so it is martingale.

Now I will simulate using a Monte Carlo to approximate the overall probability.

```
martingaleSim <- function(nsim) {
  set.seed(1) # Reproducible
  count <- 0
  for (i in 1:nsim) {
    value <- 40
    bet <- 1
    while (value > 20 & value <= 100) {
      roll <- rnorm(1) # If positive, wins, else loses
      if (roll >= 0) {
        value <- value + bet
      # bet <- 1 # Reinitialize bet to 1 (Martingale Strategy)
      }
      else {
        value <- value - bet
      # bet <- bet * 2 # Double bet to get money back (Martingale Strategy)
      }
    } # End while loop
    if (value >= 100) { count <- count + 1 }
  } # End for loop
  return(count / nsim)
}
n <- 3000 # Sufficiently large, law of large numbers
# says this approaches the mean (probability)
martingaleSim(n)
```

```
## [1] 0.2533333
```

As the number of simulations increase the probability approaches 0.25, or 1/4. This makes sense as she is 20 away from the bottom but 60 away from the top. This means the gambler has to get 3 times as many wins than losses, making the probability make sense.

If the gambler was able to use the Martingale Strategy commented out in the function above, the probability would increase to over 1/3 (around 0.37 with n = 30000). It is more steady because every win they get will make them a dollar, and they are counting on the idea that losing that many times in a row is highly unlikely in the long run. This allowed me to use ten times as many simulations as they would also have to place fewer bets in order to obtain their result.