Assignment 4

Lorenzo Ausiello

2023-12-11

QUESTION 1

1.1

Write this problem into the form on page 9 to find f, ui and ci. Then use function constrOptim() to find the optimal portfolio weights x and the optimal value, you can choose starting point as (2, -2, 0).

```
mu <- c(0.0427, 0.0015, 0.0285)
sigma <- matrix(c(0.01, 0.002, 0.001, 0.002, 0.011, 0.003, 0.001, 0.003, 0.02),nrow=3, ncol=3)

f <- function(x) 0.5*t(x)%*%sigma%*%x

ui <- matrix(c(0.0427, 0.0015, 0.0285, -1, -1, -1), 2, 3, byrow = TRUE)
ci <- c(0.05, -1)

constrOptim(c(2,-2,0), f, grad = NULL, ui = ui, ci = ci)$par
```

```
## [1]  0.9834369 -0.2790816  0.2956445
```

```
constrOptim(c(2,-2,0), f, grad = NULL, ui = ui, ci = ci)$value
```

```
##             [,1]
## [1,] 0.005632476
```

1.2

Write this problem into the form on page 18 to find D, d, A and b. Then use function solve.QP() to find the optimal portfolio weights x and the optimal value.

```
D <- matrix(c(0.01, 0.002, 0.001, 0.002, 0.011, 0.003, 0.001, 0.003, 0.02),nrow=3, ncol=3)
d <- c(0,0,0)
A <- matrix(c(0.0427, 0.0015, 0.0285, -1, -1, -1), 3, 2)
b <- c(0.05, -1)

solve.QP(D, d, A, b)$solution
```

```
## [1]  0.9794907 -0.2811567  0.3016660
```

```r
solve.QP(D, d, A, b)$value
```

```
## [1] 0.005632056
```

QUESTION 2

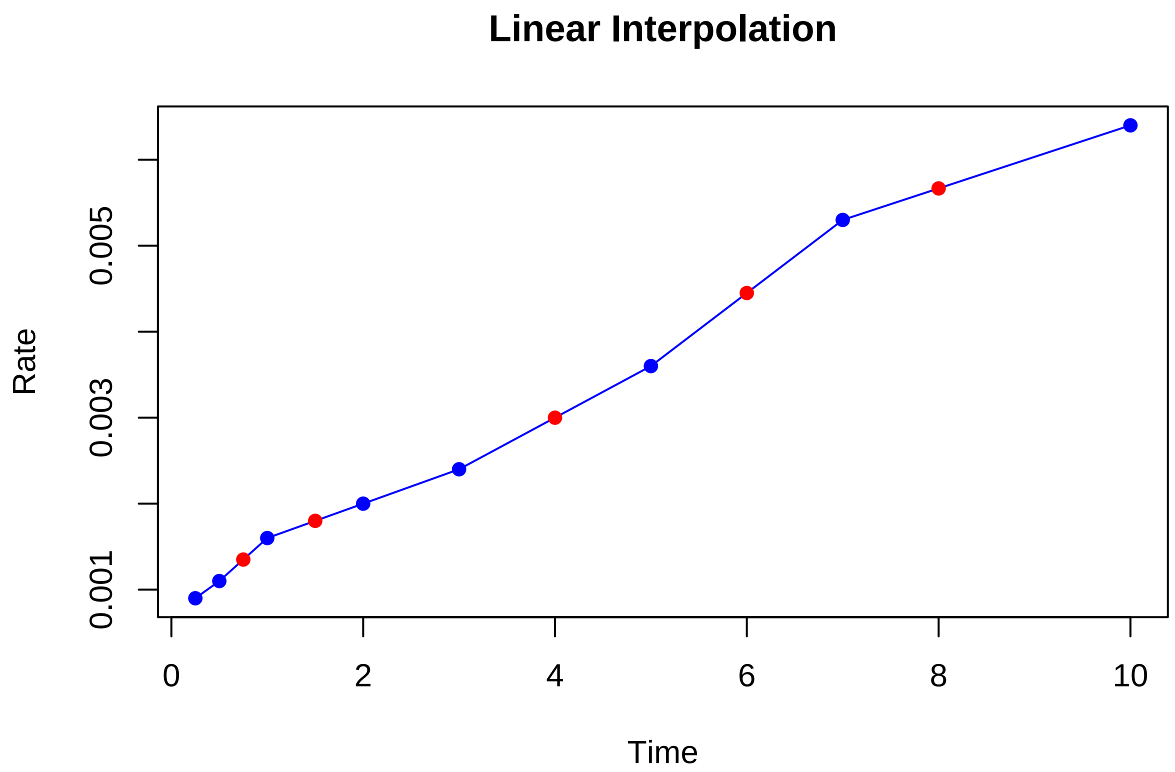
2.1

Use linear interpolation to find r(out).

```r
t <- c(0.25, 0.5, 1, 2, 3, 5, 7, 10)
r <- c(0.0009, 0.0011, 0.0016, 0.002, 0.0024, 0.0036, 0.0053, 0.0064)
tout <- c(0.75, 1.5, 4, 6, 8)
rout <- approx(t, r, tout, method = "linear")
print(rout)
```

```
## $x
## [1] 0.75 1.50 4.00 6.00 8.00
##
## $y
## [1] 0.001350000 0.001800000 0.003000000 0.004450000 0.005666667
```

```r
plot(t, r, type = "o", col = "blue", pch = 16, xlab = "Time", ylab = "Rate", main = "Linear Interpolati
points(tout, rout$y, col = "red", pch = 16)
```



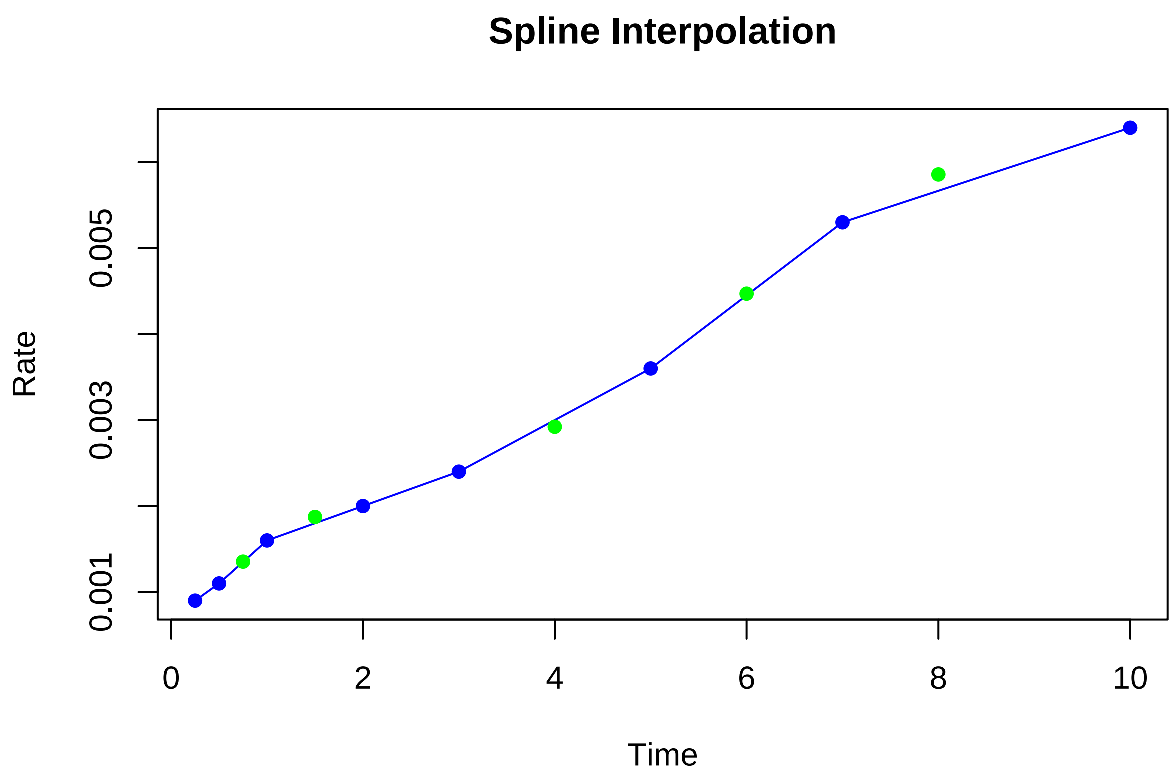
2.2

Use spline interpolation to find rout.

```
rout.s <- spline(t, r, xout = tout, method = "natural")
rout.s$y
```

```
## [1] 0.001353578 0.001873147 0.002922171 0.004470451 0.005856694
```

```
plot(t, r, type = "o", col = "blue", pch = 16, xlab = "Time", ylab = "Rate", main = "Spline Interpolati
points(tout, rout.s$y, col = "green", pch = 16)
```



QUESTION 3

Calculate the value from above equation using numerical integration with S0 = 100, K = 100, T = 1, r = 0.05, sigma = 0.2, compare to the call option price from Black-Scholes formula with the same parameters.

Numerical integration

```
S <- 100
K <- 100
sigma <- 0.2
r <- 0.05
tau <- 1
```

```r
# Retrieving the Call Price
call.option <- function(x){ (S*exp((r-0.5*sigma^2)*tau+sigma*sqrt(tau)*x)-K)*dnorm(x)}
d2 <- (log(S/K)+(r-0.5*sigma^2)*tau)/(sigma*sqrt(tau))
integrate(call.option, -d2, Inf)$value * exp(-r*tau)
```

```
## [1] 10.45058
```

Black-Scholes formula

```r
bs.call <- function(S, K, tau, sigma, r){
  d2 <- (log(S/K) + (r - 0.5 * sigma^2) * tau) / (sigma * sqrt(tau))
  d1 <- d2 + sigma * sqrt(tau)
  option_price <- S * pnorm(d1) - K * exp(-r * tau)  * pnorm(d2)
  return(option_price)
}

call_price <- bs.call(S, K, tau, sigma, r)
print(call_price)
```

```
## [1] 10.45058
```