Problem Set 3

Yan Jue

2023-05-14

Contents

Problem Set 3

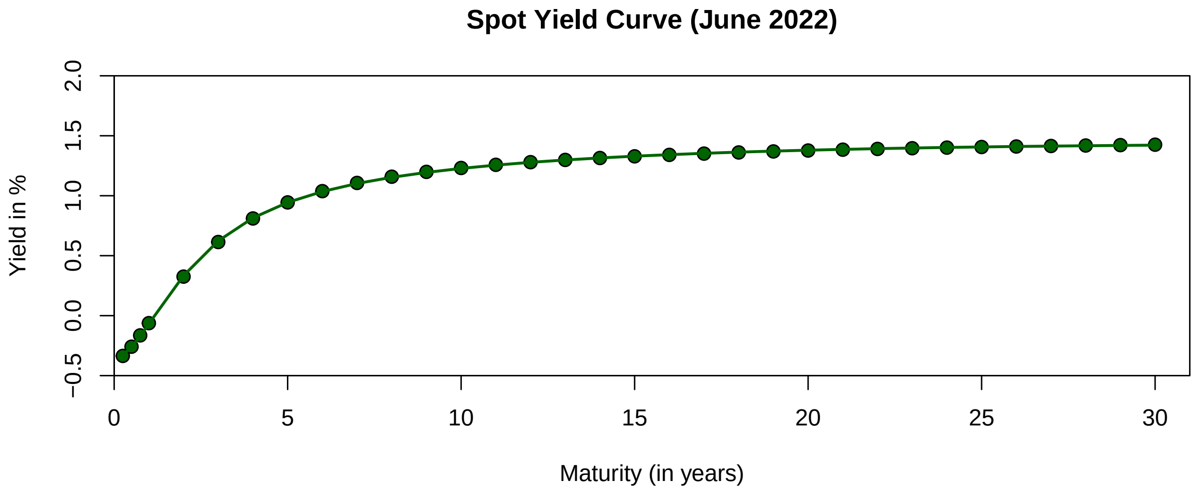
Exercise 1: Nelson-Siegel-Svensson Model

a). Use the interest rate data in the file YieldCurve.csv to estimate the Nelson-Siegel- Svensson yield curve model. Report your parameter estimates

```
##        b0        b1        b2        b3  lambda1   lambda2
## 1 1.38923 -1.729319 -1.886899 0.3965269 1.434619 0.0731952
```

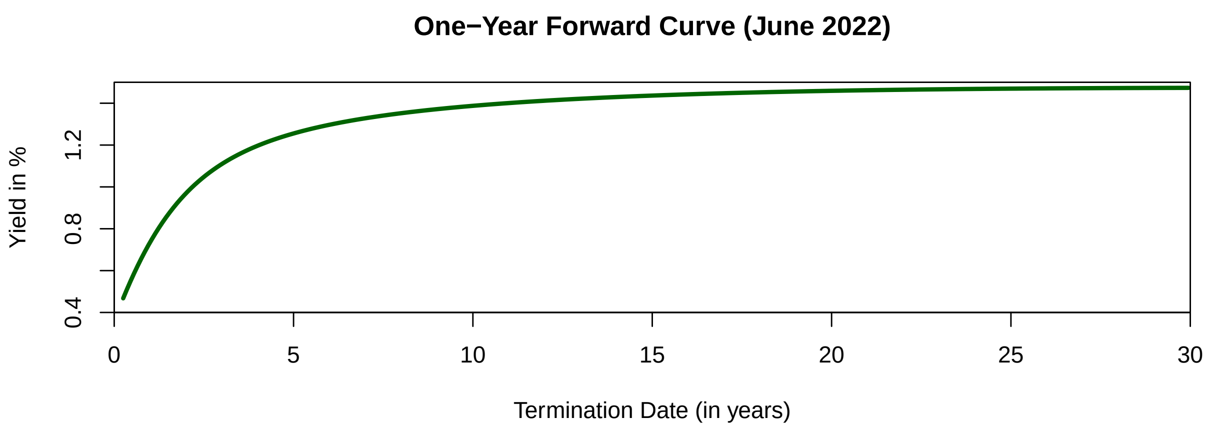
b). Based on your estimated model parameters from a), program an R function that returns the spot interest rate for any maturity T. Plot your spot yield curve

```r
# the R function that returns the spot interest rate for any maturity T
y0 <- function(T){
  F1 <- (1-exp(-lambda1*T))/(lambda1*T)
  F2 <- (1-exp(-lambda1*T))/(lambda1*T) - exp(-lambda1*T)
  F3 <- (1-exp(-lambda2*T))/(lambda2*T) - exp(-lambda2*T)
  b0 + b1*F1 + b2*F2 + b3*F3}

# plot the spot yield curve
plot(maturity, yields, type = "p", cex = 1.25,
     xaxs = "i", yaxs = "i", pch = 21, col = "black", bg = "darkgreen",
     main = "Spot Yield Curve (June 2022)",
     xlim = c(0,31), ylim = c(-0.5,2),
     cex.axis = 1, cex.lab = 1, cex.main = 1.2,
     xlab = "Maturity (in years)",
     ylab = "Yield in %")
lines(maturity, y0(maturity), col = "darkgreen", lwd = 2)
```



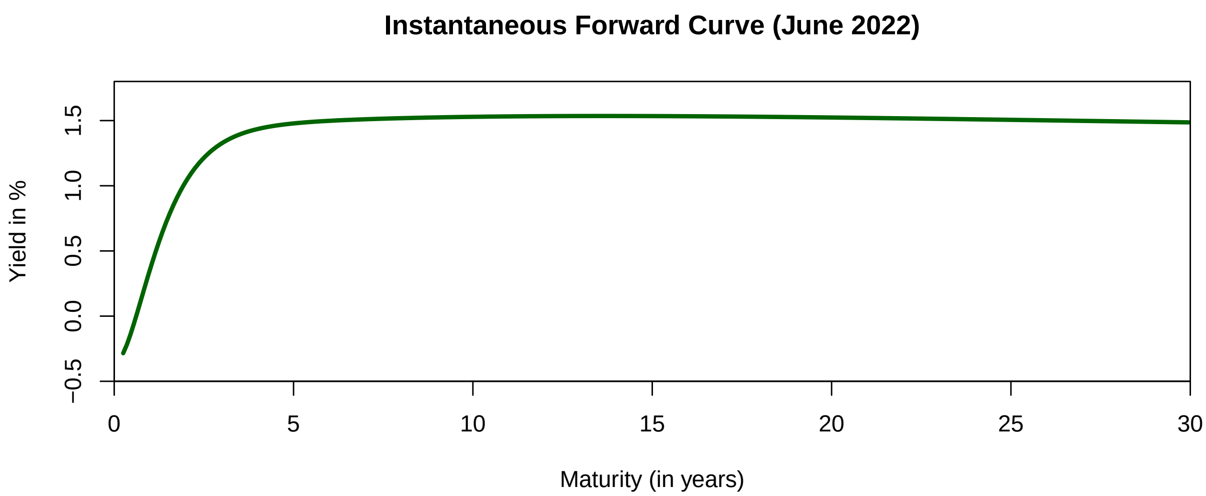
c). Use your spot yield curve function from b) to program an R function for arbitrage- free forward rates with an effective date T1 and a termination date T2. Plot the forward curve for effective date T = 1

```
# the R Function for arbitrage-free forward rates (T2 > T1)
f0 <- function(T1,T2){(y0(T2)*T2-y0(T1)*T1)/(T2-T1)}

# Plot the forward curve for T = 1
m <- seq(0.25,30,0.1)
plot(m, f0(1,1+m), type = "l", cex = 1.25,
     xaxs = "i", yaxs = "i", lwd = 3, col = "darkgreen",
     main = "One-Year Forward Curve (June 2022)",
     xlim = c(0,30), ylim = c(0.4,1.5),
     cex.axis = 1, cex.lab = 1, cex.main = 1.2,
     xlab = "Termination Date (in years)",
     ylab = "Yield in %")
```



d). Based on your estimated model parameters from a), program an R function that returns the instantaneous forward rate for any effective date T. Plot your instantaneous forward curve.

```
# the R function that returns the instantaneous forward rate for any effective date T
f0inst <- function(T){
        F1 <- exp(-lambda1*T)
        F2 <- exp(-lambda1*T)*lambda1*T
        F3 <- exp(-lambda2*T)*lambda2*T
        b0 + b1*F1 + b2*F2 + b3*F3}

# Plot the instantaneous forward curve.
plot(m, f0inst(m), type = "l", cex = 1.25,
     xaxs = "i", yaxs = "i", lwd = 3, col = "darkgreen",
     main = "Instantaneous Forward Curve (June 2022)",
     xlim = c(0,30), ylim = c(-0.5,1.8),
     cex.axis = 1, cex.lab = 1, cex.main = 1.2,
     xlab = "Maturity (in years)",
     ylab = "Yield in %")
```

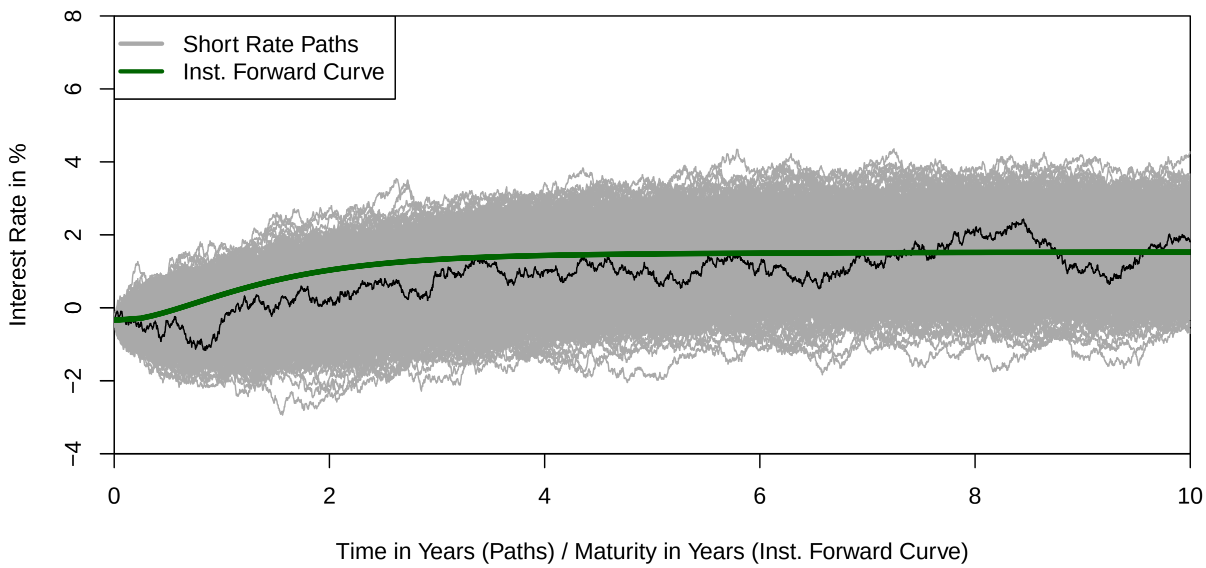


Exercise 2: Hull-White Model

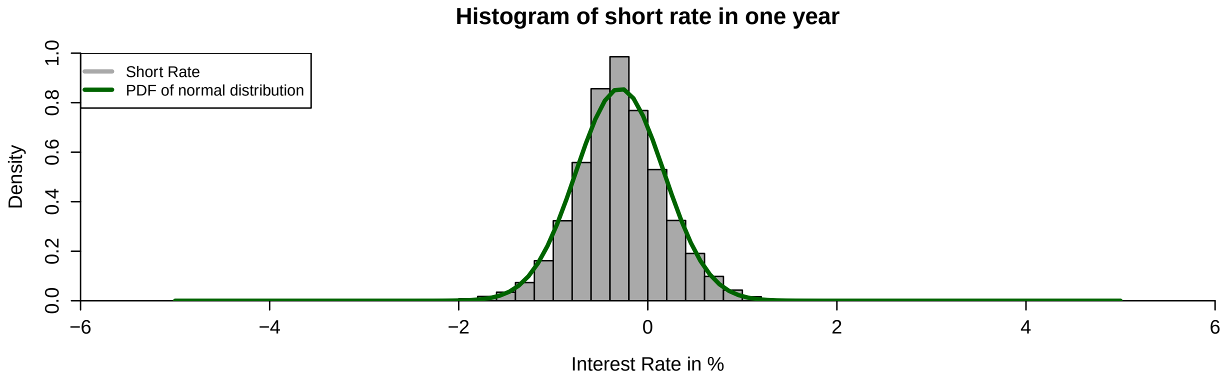
a).Simulate and plot 1000 short rate paths over 10 years with a step size of 365 days per year. Include the instantaneous forward rate curve in your plot.

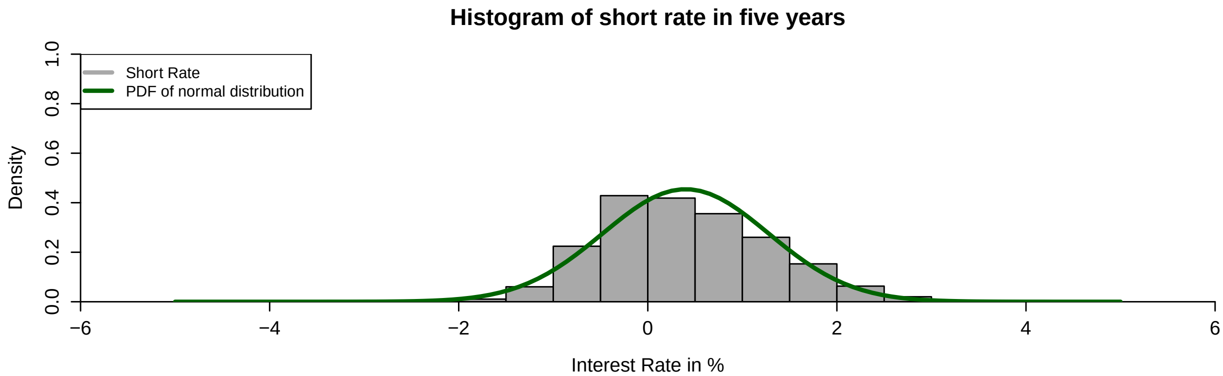
```
alpha <- 0.5
sigma <- 0.8
paths <- 1000
T <- 10
n <- 365
dt <- rep(1/n,T*n)
t <- c(0,cumsum(dt))
r0 <- y0(1/n)
```

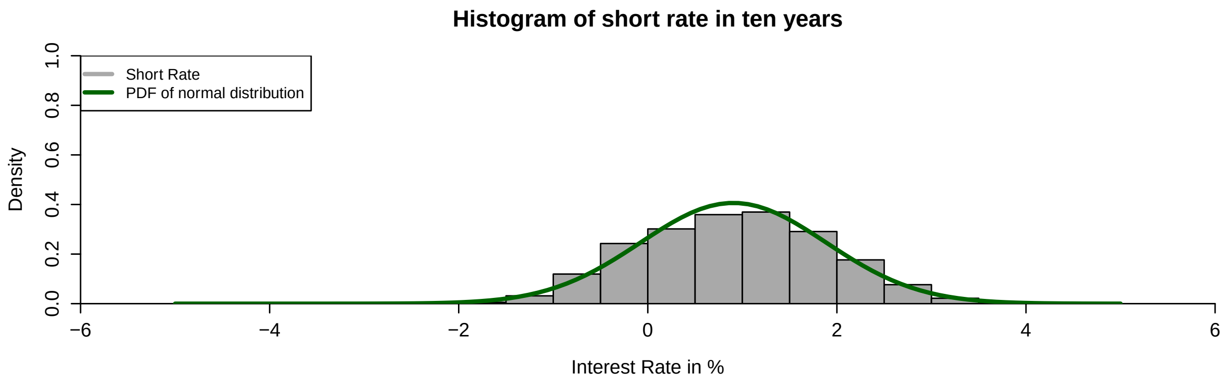
```r
df0inst_dt <- function(T){
  -lambda1*exp(-lambda1*T)*(b1 - b2 + b2*lambda1*T)
  +lambda2*exp(-lambda2*T)*(b3 - b3*lambda2*T)}

set.seed(13)
rt <- matrix(NA, paths, length(dt)+1)
rt[,1] <- r0

for(j in 1:length(rt[,1])){
for(i in 2:length(rt[1,])){
        theta <-  df0inst_dt(t[i-1]) + alpha*f0inst(t[i-1])
        dW <- sqrt(dt[1])*rnorm(n = 1, mean = 0, sd = 1)
        rt[j,i] <- rt[j,i-1] + (theta - alpha*rt[j,i-1])*dt[1] + sigma*dW}}
```



b). Create a histogram of the short rate in one year, in five years and in ten years. Overlay the probability density function (pdf) of a normal distribution with a matching mean and standard deviation.







- The **longer** the T, the **larger** the standard deviation and the mean of the short rate.

c).Compute the *expected* price of a zero-coupon bond with a maturity of one year in three years from now.

```
t = 3
T = 1+3
price <- function(t,T,j){
        # Initial zero bond prices (from current term structure)
```

```r
        P0T <- exp(-y0(T)*T)
        P0t <- exp(-y0(t)*t)
        # B coefficient
        B <- (1 - exp(-alpha*(T-t)))/alpha
        # C coefficient (introduced here to simplify lnA)
        C <- (exp(-alpha*T) - exp(-alpha*t))^2*(exp(2*alpha*t)-1)
        # A coefficient
        lnA <-log(P0T/P0t) + B*f0inst(t) - 1/(4*alpha^3)*sigma^2*C
        # translate t into index to select adequate short rate from matrix rt
        x <- ceiling(t/dt[1])
        # Zero prices at future time t
        PtT <- exp(lnA)*exp(-B*rt[j,x])
        return(PtT)
}
prices = c()
for(i in 1:paths){
  prices[i] = price(t,T,i)
}
paste('the expected price of a zero-coupon bond with maturity T = 1 that pays back $1 is'
      , round(mean(prices),4))
```

```
## [1] "the expected price of a zero-coupon bond with maturity T = 1 that pays back $1 is 0.4252"
```

Use the Hull-White model to price the caplet.

```r
# input
tk1 = 3
tk = 2
deltat = tk1 - tk
N = 1000000
yx = 0.015

y23 = c()
ctk1 = c()
for(i in 1:paths){
  y23[i] = -log(price(2,3,i))/(tk1-tk)
  ctk1[i] = N * deltat * max(y23[i]-yx,0)
}

p = mean(ctk1)/(1+y0(tk1))^tk1

paste('the price of the caplet should be',round(p,4))
```

```
## [1] "the price of the caplet should be 169871.3122"
```