Assignment 1 / FE 680

1) Discount Factor values calculated manually in rough sheet and stored the discount values in a list in R, which i further used to calculate the Zero Curve and Forward Curve values.

			discount	zero	forward	par		Bond
		Inputs	curve	curve	curve	curve		Cash Flow
Overnight	0	1.600%			1.6			
Cash	1	1.900%	0.9813	1.9	1.9			4
	2	2.100%	0.9592	2.1	2.3			4
Forwards	3	2.200%	0.9385	2.13	2.2			4
	4	2.450%	0.9161	2.21	2.45			4
	5	2.500%	0.8937	2.27	2.5			4
Swaps	6	2.550%	0.8587	2.57	4.07	2.55		4
	7	2.800%	0.8219	2.84	4.48	2.80		4
	8	2.850%	0.7960	2.89	3,25	2.85		4
	9	3.000%	0.7625	3.05	4.35	3.00		104
	10	3.150%	0.7276	3.23	4.8	3.15		

```
# Discount Curve Values
d <-c(0.9813,0.9592,0.9385,0.9161,0.8937,0.8587,0.8219,0.7960,0.7625,0.7276)
t<-c(1:10)
length(d)
# Zero Curve
z<-c()
for(i in 1:10){
  value<-(1/d[i])^(1/t[i])-1
  z<-append(z,value)
}
z*100

# Forward Curve
f <- c()
for(i in 2:10){
  value<-(d[i-1]/d[i])-1
  print(d[i])
  print(d[i-1])
  print(value)
  print("*****")
  f<-append(f,value)
}
f*100
```

1.2) Computing values of bond cash flows, please refer to R code # Section 1.2
Present value = **$ 107.96;** Multiply discount with future cash flow **CF*discount**

```
# Section 1.2
# Calculating The present Value
cf<-c(4,4,4,4,4,4,4,4,104)

pv=0
for(i in 1:9){
  pv=pv+(cf[i]*d[i])
}
pv
```

1.3)
Calculating discount
#Section 1.3

```
> f1 <- c(0.019,0.02304003,0.02205647,0.02445148,0.02506434,0.04075929,0.04477430,0.
0325,0.0439,0.0479)
> f1=f1*100
> d2<-c()
> temp=1
> f2=(f1)+0.5
> # New Discount Factor
> for(i in 1:10){
+   temp=temp*(1/(1+f2[i]/100))
+   d2<-append(d2, temp)
+ }
> d2
 [1] 0.9765625 0.9499265 0.9249019 0.8984415 0.8722188 0.8340532 0.7945071
 [8] 0.7657900 0.7300887 0.6934074
```

Calculating New **PV = $ 103.9948;** increasing forward rate decreases Present Value

```
> pv=0
> for(i in 1:9){
+   pv=pv+(cf[i]*d2[i])
+ }
> pv
[1] 103.9948
```

Calculating DV101; changing **9th year** forward rate has highest dv01

```
> dv01<-c()
> temp=0
> for(i in 1:9){
+   temp = -(1/1000)*((pv2[i]-pv1[i])/(f2[i]-f1[i]))
+   dv01=append(dv01,temp)
+ }
> dv01*100
[1] 0.00379000 0.01120877 0.02208721 0.03621400 0.05339893 0.07311641
[7] 0.09503072 0.11919872 0.79335464
```

1.4) Please refer to #Section 1.4 - Increasing Forward rates gives lower present values for bonds

```r
# Section 1.4
inc<-c(1,2,3)
for(j in 1:3){
  print("Increase in percetntage by")
  print(inc[j])
  d2<-c()
  temp=1
  f2=(f1)+inc[j]
  # New Discount Factor
  for(i in 1:10){
    temp=temp*(1/(1+f2[i]/100))
    d2<-append(d2, temp)
  }
  print("Discount Factor")
  print(d)

  pv2<-c()
  temp=0
  for(i in 1:9){
    temp=temp+(cf[i]*d2[i])
    pv2=append(pv2,temp)
  }
  print("Present Value")
  print(pv2)
}
```

```
[1] "Increase in percetntage by"
[1] 1
[1] "Discount Factor"
 [1] 0.9718173 0.9407354 0.9115154 0.8811582 0.8513077
 [6] 0.8101833 0.7681106 0.7367967 0.6991145 0.6608512
[1] "Present Value"
[1]   3.887269   7.650211  11.296272  14.820905
[5]  18.226136  21.466869  24.539312  27.486499
[9] 100.194404
[1] "Increase in percetntage by"
[1] 2
[1] "Discount Factor"
 [1] 0.9624639 0.9227488 0.8855075 0.8478206 0.8112616
 [6] 0.7647933 0.7182680 0.6824399 0.6414512 0.6006660
[1] "Present Value"
[1]  3.849856  7.540851 11.082881 14.474163 17.719209
[6] 20.778383 23.651455 26.381214 93.092136
[1] "Increase in percetntage by"
[1] 3
[1] "Discount Factor"
 [1] 0.9532888 0.9052731 0.8604796 0.8160448 0.7734550
 [6] 0.7223425 0.6720876 0.6325531 0.5890242 0.5464553
[1] "Present Value"
[1]  3.813155  7.434248 10.876166 14.140345 17.234165
[6] 20.123535 22.811886 25.342098 86.600614
> 
```

1.5) For this question I made use of cubic spline model to calculate the Yield Maturity for 1.5 years and once known, I use the value to find out forward price of the bond
Refer # Section 1.5

Forward Price of bond **$102.98**

```
#Section 1.5
# Making use of cubic spline model to fund the yield for 1.5 years
t <- c(0,1,2,3,4,5,6,7,8,9,10)
y <- c(1.6, 1.9, 2.1, 2.2, 2.45,2.5, 2.55, 2.8, 2.85,3,3.15)
spl <- smooth.spline(y ~ t)
s<-predict(spl, 1.5)
# Yield Rate is 1.9688
Forwar_Price = 100*exp(.0196*1.5)

# Foward Price Of Bond Is 102.9836
```

1.6) # Section 1.6

Duration is defined as:

$$D = -\frac{1}{P} \times \frac{\Delta P}{\Delta y}$$

Duration for above bond is 0.08088 -
Taken Yield for 1.5 years and 0 year
Duration = **0.08088**

```
duration = - ((-102.9836 + 100)/(1.96888 - 1.6))/100
0.08088267
```

2)The Put value for bond option contract are
Cash Price : **$ 28.86**
Quoted Price : **$ 39.184**
Please refer to screen shots below, no coding was done for this question

Assignment

Answer 2)

$$C = P(0,T)\left[F_B N(d_1) - N(d_2)K\right]$$

$$P = P(0,T)\left[K N(-d_2) - F_B N(-d_1)\right]$$

F_B = Forward bond price.

$$d_1 = \frac{\ln\left(\frac{F_B}{K}\right) + \sigma_B^2 \frac{T}{2}}{\sigma_B\sqrt{T}} \; ; \; d_2 = d_1 - \sigma_B\sqrt{T}$$

European put option.

$T = \frac{8}{12} = \frac{2}{3}$

$B_0 = \$910 \qquad K = \$900$.

$\sigma = 0.08$

Coupon of $35

3 Months

$r = 2\%$.

$$F_B = \frac{B_0 - I}{P(0,T)}$$

$B_0 = \cancel{960}\ 910.$

$$I = 35 \times e^{-0.25 \times 0.02} = 34.8254$$

$$F_B = [910 - 34.825]\, e^{(2/3)\times 0.02} = \$886.9217$$

(a) If strike price is cash price that would be paid for the bond exercise

$F_B = 886.92$

$K = 900$

$P(0,T) = (e)^{-0.02 \times 2/3} = 0.986$

$\sigma = 0.08$

$T = 2/3$

$$d_1 = \frac{\ln\left(\frac{886.92}{900}\right) + (0.08)^2 \times \frac{2}{3} \times \frac{1}{2}}{0.08 \times \sqrt{2/3}}$$

$d_1 = -0.191$

$d_2 = -0.191 - 0.08\sqrt{2/3} = -0.256$

$P = 0.986[900 * N(0.256) - 886.92N(0.191)]$

$p = \$28.86$

(b) 8 Months - 3 Months = 5 Months of interest needs to be paid; and it must be added to K.

$K = 900 + 35 \times \frac{5}{12} = 14.583$

$= 914.583$

$d_1 = -0.437$ $\quad d_2 = -0.437 - 0.08\sqrt{2/3}$

$= -0.502$

$P = 0.986[914.58 * N(0.502) - 886.92 * N(0.437)]$

$= \$39.184$

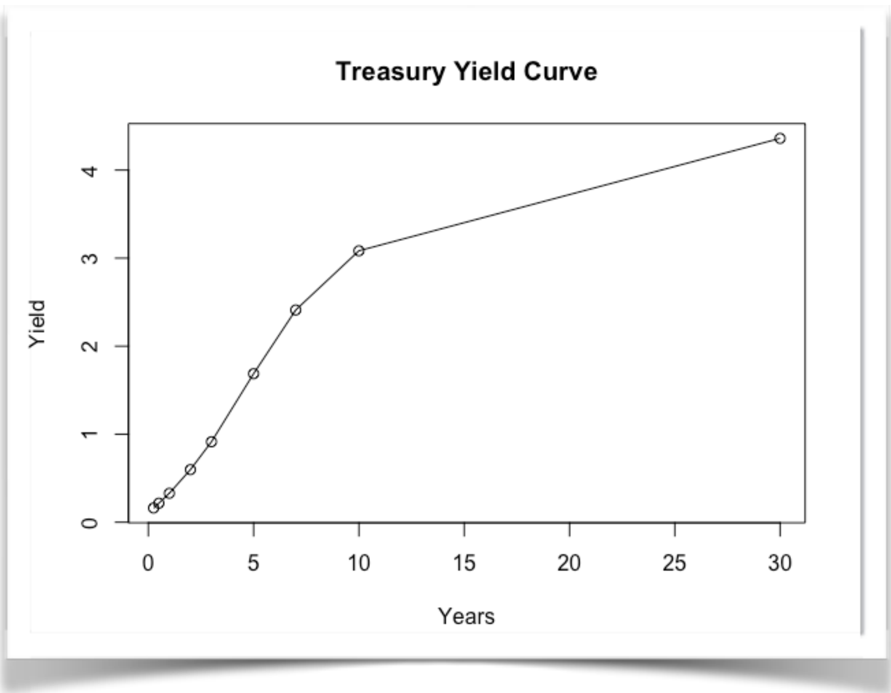
3) The following question was solved in R after understanding the Cubic Spiel model and it's implementation.
Please refer #Section 3 in code for this answer

In Bootstrapping it's difficult to find out what's the yield of a maturity even if we know the yield for some maturities. To solve this issue Spline model is used, which builds cubic differential equations from the given data for a set of interval of times. These equations answer with known amount of equations and variables which helps us to identify where the yield for maturity will lie on the curve.

So i saved the yield and maturity in two different lists and called "smooth.spline" function.
In Mathematical form this function in background would generate a system of cubic equations for different time intervals. These equations will be built from the pre given expiry yield curve data.

```
#Answer 3
# Section 3
t <- c(0.25, 0.5, 1, 2, 3, 5, 7, 10, 30)
y <- c(0.17, 0.23, 0.30, 0.61, 0.90, 1.69, 2.42, 3.08, 4.36)
spl <- smooth.spline(y ~ t)
plot(spl, ylab = 'Yield', xlab = 'Years', main = ' Yield Curve')
lines(spl)
predict(spl, t)
new_t <- seq(from = 0.5, to = 30, by = 0.5)
s<-predict(spl, new_t)
spl <- smooth.spline(s$y ~ s$x)
plot(spl, ylab = 'Yield', xlab = 'Years', main = ' Yield Curve')
lines(spl)
```

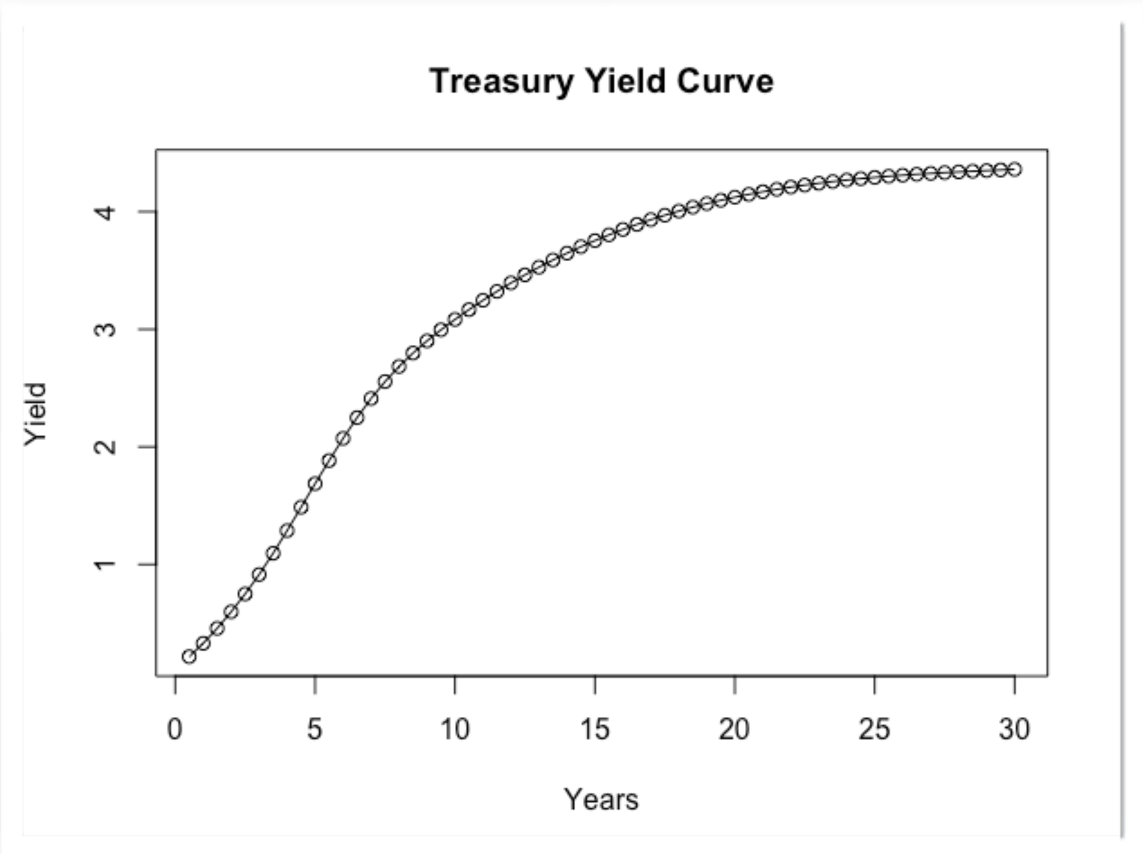
Once we have obtained these equations we go further and input different year values like(0.75 or 1.75) years to obtain the yield.



I further built a new sequence of time from 0.5 to 30 years and calculated yield for every year and further plotted it, to see what kind of curve we get. The above graph just gets more refined.

```
> s<-predict(spl, new_t)
> s
$x
 [1]  0.5  1.0  1.5  2.0  2.5  3.0  3.5  4.0  4.5  5.0  5.5  6.0  6.5  7.0  7.5
[16]  8.0  8.5  9.0  9.5 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0 14.5 15.0
[31] 15.5 16.0 16.5 17.0 17.5 18.0 18.5 19.0 19.5 20.0 20.5 21.0 21.5 22.0 22.5
[46] 23.0 23.5 24.0 24.5 25.0 25.5 26.0 26.5 27.0 27.5 28.0 28.5 29.0 29.5 30.0

$y
 [1] 0.2158917 0.3289346 0.4564993 0.5975559 0.7492567 0.9142589 1.0950223
 [8] 1.2877104 1.4869130 1.6872198 1.8835272 2.0719598 2.2489488 2.4109256
[15] 2.5553207 2.6835616 2.7980753 2.9012884 2.9956279 3.0835206 3.1670055
[22] 3.2465711 3.3223180 3.3943468 3.4627581 3.5276527 3.5891311 3.6472939
[29] 3.7022419 3.7540757 3.8028958 3.8488031 3.8918980 3.9322812 3.9700534
[36] 4.0053152 4.0381673 4.0687103 4.0970448 4.1232715 4.1474910 4.1698040
[43] 4.1903111 4.2091130 4.2263102 4.2420035 4.2562934 4.2692807 4.2810659
[50] 4.2917497 4.3014328 4.3102157 4.3181992 4.3254838 4.3321703 4.3383592
[57] 4.3441512 4.3496469 4.3549470 4.3601521
```



4) #Section 4 in R code

The question was to estimate the beta_0 , beta_1 ,beta_2 and lambda for the given data using Nelson - Siegel Model
To solve the values we needed the Maturity time period and the Yield-Maturity rates.

So the first part is to calculate the maturity-yield.
First I calculated Dirty Price for the bond.

C= Coupon
Market Quote = Clean Price
Delta = Time interval between 2 consecutive coupon payments which is always 0.5 years

Dirty Price = Clean Price + (Coupon * (Maturity date - time to next coupon)/ Delta)

I coded this in R

```
next_p <- c(0.4356, 0.2644, 0.2658, 0.4342, 0.0192, 0.4753, 0.3534, 0.1000, 0.2685, 0.4342, 0.2274,
maturity_t <- c(0.4356, 0.7644, 1.2658, 1.9342, 2.0192, 2.9753, 3.3534, 3.6000, 4.2685, 4.9342, 5.22
coupon <- c(0.875, 0.875, 0.750, 0.625, 0.375, 0.750,1.5, 1.75, 2.125, 1.75, 4.5, 2.375, 2.750, 2.37
clean_price <- c(100.3, 100.48, 100.50, 100.31, 99.78, 100.16, 102.34, 103.08, 104.19, 102.06, 115.9

# Calculate the Dirty Bond Price
Bond_Dirty_Price <- c()
Delta = 0.5
P=100
for(i in 1:29){
    value = clean_price[i]+ (Delta * (coupon[i]/100) * P) * (maturity_t[i]-next_p[i])/Delta
    Bond_Dirty_Price = append(Bond_Dirty_Price, value)
}
Bond_Dirty_Price
```

The Dirty Bond prices are as follows

```
[1] 100.3
[1] 100.9175
[1] 101.25
[1] 101.2475
[1] 100.53
[1] 102.035
[1] 106.84
[1] 109.205
[1] 112.69
[1] 109.935
[1] 138.41
[1] 117.4225
[1] 122.36
[1] 118.4075
[1] 135.03
[1] 142.1525
[1] 125.98
[1] 133.0625
[1] 138.375
[1] 126.1275
[1] 230.08
[1] 245.78
[1] 259.47
[1] 233.315
[1] 241.2119
[1] 196.98
[1] 233.565
[1] 241.545
[1] 220.0571
```

$$B_i = \text{MarketQuote} + C_i \frac{d_i - t_i'}{\delta_i}$$

Once I had the dirty bond prices I calculated the Yield Maruity return for each maturity; I built R code to calculate the yields.

```r
# Calculating the yield to maturity for each maturity

#Calculating r1
r <- c()
for(i in 1:29){
  Sum_of_coupons=0
  V=Bond_Dirty_Price[i]
  Cou=(Delta*(coupon[i]/100)*P)
  for(j in 1:(i-1)){
    if(i==1){
      Sum_of_coupons=0
    }
    else{
      Sum_of_coupons = Sum_of_coupons + Cou/(1+r[j])^j
    }
  }
  value =V-Sum_of_coupons
  r1 = ((Cou+P)/value)^(1/i)-1
  r[i]=r1
}
r
```

```
[1] 0.314992
[1] -0.05610932
[1] -0.0985501
[1] 0.001338283
[1] 0.2020759
[1] 0.07161823
[1] -0.4671387
[1] -0.5965223
[1] -0.7039378
[1] -0.2256074
[1] -2.343555
[1] -0.5079304
[1] -0.6396317
[1] -0.2056461
[1] -1.039283
[1] -1.253336
[1] -0.224814
[1] -0.4597962
[1] -0.5601822
[1] 0.08874383
[1] -2.273634
[1] -2.357679
[1] -2.44741
[1] -1.958055
[1] -1.986532
[1] -1.236964
[1] -1.706856
[1] -1.724418
[1] -1.353744
```

Yield to Maturity and Present Value of a Bond

The yield to maturity is found in the present value of a bond formula:

$$\frac{C}{1+r} + \frac{C}{(1+r)^2} \cdots \frac{C}{(1+r)^t} + \frac{F}{(1+r)^t}$$

For calculating yield to maturity, the price of the bond, or present value of the bond, is already known. Calculating YTM is working backwards from the present value of a bond formula and trying to determine what ***r*** is.

Once I had the Maturity and Yield Return, each bond can be termed as a Zero Coupon bond and we can perform Bootstrap method. I installed package called “YieldCurve” and called the function Nelson.Siegel to find out the values

```
> NS<-Nelson.Siegel(r,maturity_t)
> NS
        beta_0  beta_1   beta_2 lambda
[1,] -1.830215 1.24864 4.806803 0.4356
```

Beta_0= -1.830215
Beta_1 = 1.24864
beta_2= 4.806083
lambda = 0.4356