APM466 A1

Tianyi Long 1002902889

January 2020

1 Fundamental Questions

1.1

1(a)
The government issue bonds to partially finance its activities and control money supply in the market.

1(b)
It can help government to monitor future liability and whether there will be a recession.

1(c)
By issuing bonds or raising interest rate, government can retrieve money from public.

1.2

CAN 1.5 Mar 1 20, CAN 0.75 Sep 1 20,
CAN 0.75 Mar 1 21, CAN 0.75 Sep 1 21,
CAN 0.5 Mar 1 22, CAN 2.75 Jun 1 22,
CAN 1.75 Mar 1 23, CAN 1.5 Jun 1 23,
CAN 2.25 Mar 1 24, CAN 1.5 Sep 1 24.

I choose 10 bonds which have maturity dates in March and September and recent 5-year bonds since the liquidity issues. Also, avoid the high coupon rate bonds, because the duration and default risk of that bond are too high. For year 2022 and year 2033, we lose the data for September. So I choose bonds mature in June instead.(Nearest Neighbourhood Approach)
Since Canadian government bonds have coupon payment each half year, so we can use them to interpolate the rates in June and December. Also, the number of time variables in bootstrapping algorithm are the same, we can easily deal with that.

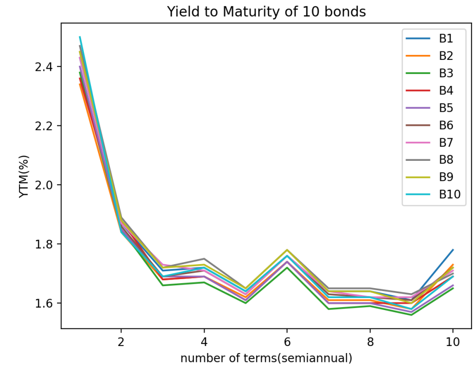
1.3

Covariance matrix represents the direction and spread along variables of data set. Eigenvalues represent how large the variance of data corresponding to each directions represented by the related eigenvectors. The eigenvector with largest eigenvalue determines the first component.

2 Empirical Questions

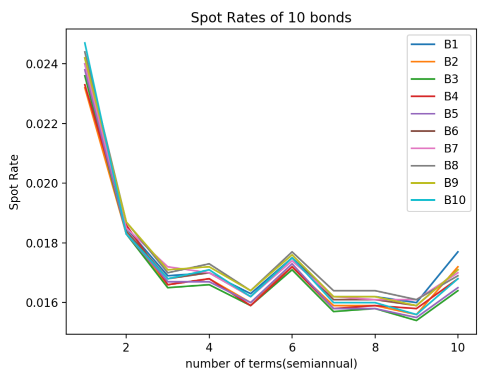
2.1 4 a

First we should calculate the dirty price of each bond on each day by formula $DirtyPrice = AccruedInterest + CleanPrice$ and $AccruedInterest = \frac{n}{365} \times AnnualCoupon$.
Then we discount all the coupon payments and dirty price to the maturity dates, by using formula:
$P \cdot e^{YTM \times t_p} - \sum_{i=1}^{n-1} \frac{C}{2} \cdot e^{YTM \times t_i} - (\frac{C}{2} + FV) = 0$
At last, we can obtain approximated yield to maturity of each bonds by Newton Method.



2.2 4 b

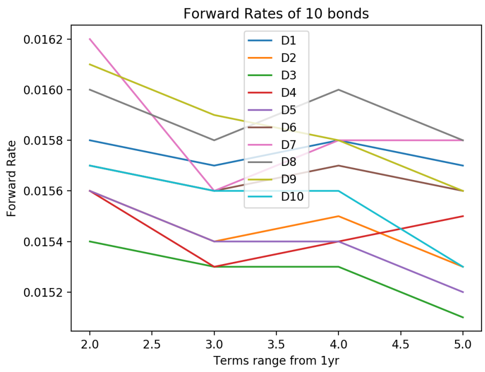
Recall the formula: $P_k = \sum_{k=1}^{k-1} C_k e^{-r_i t_i} + (C_k + Principal)e^{-r_k t_k}$
where P_k is the price of bond k, C_k is semiannual coupon payment, r_i is the zero rate at time t_i, t_k is the duration to maturity of bond.
We can reform the equation to calculate the only unknown variable: r_k.
Note that when facing no coupon payment till maturity, we can pass the first term on the right hand of equation. i.e. $\sum_{k=1}^{k-1} C_k e^{-r_i t_i} = 0$
Pseudo:
Calculate the spot rate of bond which does not have coupon till maturity and store them in list bd1spot
spot rate = - math.log(Dirty Price/ (Coupon/2 + 100)) / (day difference / 365)
Construct 2D array for storing all 10 bonds' info: result = [bd1spot]
for i in range(1, 10):
  temp = []
  Calcualte the spot rate on each day for every bonds.
  for j in range(10):
    pmt = 0
    coupon num = i
    Calculate the total coupon payment base on previous rates
    while coupon num > 0:
      prev sp = result[coupon num - 1][j]
      days = (coupon num - 1) * 184 + 59 - j
      pmt += df['Coupon'][index]/2 * math.exp(-prev sp * days / 365)
      coupon num -= 1
    spot = - math.log((Dirty Price - pmt)/(Coupon/2+100)) / ((i*184+59-j)/365)
    After calculation, store the spot rates in related positions.
    temp.append(round(spot,4))
    result.append(temp)



2.3 4 c

Recall the formula: $r_{i,j} = \frac{r_j \cdot t_j - r_1 \cdot t_1}{t_j - t_1}$
where r_j is the spot rate of year j, t_j is the future time(same as r_1, r_j).
Pseudo:

forward = [[] for i in range(10)]
Calculate forward rate for 10 bonds.
for i in range(10):
We have 5 spot rate for 1 day, so we will get 4 data points as output.
for j in range(1,5):
r_j = spr day[i][j]
r_0 = spr day[i][0]
t_j = (59 - i + 184 * j * 2) / 365
t_0 = (59 - i) /365
Store the forward rates.
forward[i].append(round((r_j * t_j - r_0 * t_0) / (t_j - t_0),4))



2.4 5

Covariance matrix of daily log-returns of yield:

$$\begin{bmatrix} 1.80235000e-04 & 1.13335000e-04 & 9.16362500e-05 & 1.08628750e-04 & 1.50232500e-04 \\ 1.13335000e-04 & 1.78965000e-04 & 1.17226250e-04 & 1.57482500e-04 & 2.00026250e-04 \\ 9.16362500e-05 & 1.17226250e-04 & 9.80525000e-05 & 1.14515000e-04 & 2.11111250e-04 \\ 1.08628750e-04 & 1.57482500e-04 & 1.14515000e-04 & 1.49781944e-04 & 2.09458889e-04 \\ 1.50232500e-04 & 2.00026250e-04 & 2.11111250e-04 & 2.09458889e-04 & 6.77895278e-04 \end{bmatrix}$$

Covariance matrix of forward rates:

$$\begin{bmatrix} 6.45555556e-08 & 4.20000000e-08 & 4.87777778e-08 & 5.30000000e-08 \\ 4.20000000e-08 & 4.26666667e-08 & 4.24444444e-08 & 3.62222222e-08 \\ 4.87777778e-08 & 4.24444444e-08 & 5.12222222e-08 & 5.03333333e-08 \\ 5.30000000e-08 & 3.62222222e-08 & 5.03333333e-08 & 6.32222222e-08 \end{bmatrix}$$

2.5 6

Eigenvalues of daily log-returns of yield:
9.88506597e-04, 2.10732085e-04, 7.70933878e-05, 6.56575021e-06, 2.03190175e-06
Eigenvectors corresponding to above eigenvalues(daily log-return of yield):
$[-0.27166504, -0.52338625, 0.80132495, 0.08253816, 0.05771234]$
$[-0.33943844, -0.46249166, -0.47203707, 0.6485853, -0.16553288]$
$[-0.30063933, -0.15389372, -0.0921631, -0.5310863, -0.77161261]$
$[-0.33477946, -0.36168915, -0.33915881, -0.52559017, 0.60483878]$
$[-0.78007587, 0.59805182, 0.10740853, 0.11927649, 0.08973396]$
Eigenvalues of forward rates:
1.93488327e-07, 1.56377898e-09, 1.07698922e-08, 1.58446684e-08
Eigenvectors corresponding to above eigenvalues(forward rates):
$[0.54301307, 0.10028102, 0.83356102, 0.01601726]$
$[0.41886698, -0.53930621, -0.19445348, -0.70419252]$
$[0.49869035, 0.74572745, -0.41148899, -0.16085808]$
$[0.5300898, -0.37813107, -0.31311449, 0.69136171]$
The eigenvector corresponding the largest eigenvalue represents the direction of the largest variance of data.

3 Reference and Github

Bonds info: https://markets.businessinsider.com/bonds/
PCA info: https://en.wikipedia.org/wiki/Principal_component_analysis
Github link: https://github.com/DuDuBot/longtian-APM466-A1